

Höcherl & Hackl GmbH  
Industriestr. 13  
94357 Konzell  
Germany

**Handbuch Version/**

**Manual Version: ZS 0320-35**

**gültig für/valid for Firmware:**  
**ZS\_SCP1\_05.00 ff**  
**ZS\_AI\_05.15 ff**

Ihr Ansprechpartner / your contact:



## Höcherl & Hackl GmbH Elektronische Last ZS / Electronic Load ZS Programmieranleitung / Programming Manual

### Inhalt

<b>1</b>	<b>Allgemeines</b> .....	<b>6</b>
1.1	Ein- und Ausschalten.....	6
1.2	Remote und Local.....	6
<b>2</b>	<b>Serielle Schnittstelle</b> .....	<b>7</b>
2.1	Einstellen der RS232-Parameter.....	7
2.2	USB-Schnittstelle.....	10
2.3	Datenformat bei RS232 und USB.....	10
2.4	Auslesen von Daten.....	10
<b>3</b>	<b>GPIB Interface</b> .....	<b>12</b>
3.1	Einstellen der GPIB-Geräteadresse.....	12
3.2	Datenformat bei GPIB.....	13
3.3	Auslesen von Daten.....	15
<b>4</b>	<b>Ethernet-RS-232-Konverter (Option ZS25)</b> .....	<b>16</b>
<b>5</b>	<b>Systembus</b> .....	<b>17</b>
<b>6</b>	<b>Unteradressen</b> .....	<b>19</b>
6.1	Einzeladressierung.....	19
6.2	Blockadressierung.....	21
6.3	Systemadressierung.....	23
<b>7</b>	<b>SCPI-Syntax Einführung</b> .....	<b>24</b>
7.1	Common Commands.....	24
7.2	Gerätespezifische Befehle.....	24
7.2.1	Aufbau des Headers.....	24
7.2.1.1	Einrückungen.....	25
7.2.1.2	Aliases.....	25

7.2.2	White Space .....	26
7.2.3	Lang- und Kurzform, Groß- und Kleinschreibung .....	26
7.2.4	Wahlweise einfügbare Schlüsselwörter .....	27
7.2.5	Parameter .....	27
7.2.5.1	Zahlenwerte <NRF> .....	28
7.2.5.2	Einheiten und Multiplizierer .....	28
7.2.5.3	Zahlenwerte und Extremwerte <num> .....	29
7.2.5.4	Boolesche Parameter .....	30
7.2.5.5	Text .....	31
7.2.6	Benutzung des Semikolons .....	31
7.2.7	Abfragebefehle (Queries) .....	33
7.2.8	Auslesen von Daten .....	33
7.3	SCPI Befehlsübersicht .....	34
7.3.1	Common Commands .....	34
7.3.2	Gerätespezifische Befehle der Serie ZS .....	35
7.4	SCPI Befehlsbeschreibung .....	41
7.4.1	Common Commands .....	41
7.4.2	Gerätespezifische Befehle .....	45
7.4.2.1	Schnelleinstieg .....	45
7.4.2.2	Subsystem CHANnel INSTrument .....	47
7.4.2.3	Subsystem CURRent .....	49
7.4.2.4	Subsystem DATA TRACe .....	58
7.4.2.5	Subsystem DELay .....	61
7.4.2.6	Subsystem GTL .....	62
7.4.2.7	Subsystem INPut OUTPut .....	63
7.4.2.8	Subsystem LIST .....	64
7.4.2.9	Subsystem MEASure .....	77
7.4.2.10	Subsystem MODE FUNctIon .....	82
7.4.2.11	Subsystem PCYCLE .....	87
7.4.2.12	Subsystem POWer .....	96
7.4.2.13	Subsystem PROGram .....	101
7.4.2.14	Subsystem RESistance .....	105
7.4.2.15	Subsystem SETup .....	112
7.4.2.16	Subsystem SFUNctIon .....	115
7.4.2.17	Subsystem STATus .....	124
7.4.2.18	Subsystem SYSTem .....	136
7.4.2.19	Subsystem TRIGger .....	151
7.4.2.20	Subsystem VOLTage .....	155
7.5	Benutzerorientierte Befehlsfolge .....	165
7.5.1	Gerät in Grundzustand versetzen .....	165
7.5.2	Auswahl der Betriebsart .....	165
7.5.3	Sollwert einstellen .....	165
7.5.4	Geräteeingang zu- und abschalten .....	166
7.5.5	Messen von Strom-, Spannungs-, Widerstands- und Leistungswerten .....	166
7.5.6	Programmierbare Kurvenform .....	166
7.5.7	Messdatenerfassung .....	167
7.5.8	Batterieprüf-Funktion .....	168
7.5.9	Externe Ansteuerung .....	169
7.5.10	Volle Lüfterleistung .....	169
7.5.11	Regelgeschwindigkeit umschalten .....	169
7.5.12	Software-Watchdog einstellen .....	170

---

7.5.13	Steuerung der Power-I/O-Karte (Option ZS07).....	170
7.5.14	Funktionen mit Option ZS13 (Data Acquisition).....	171
7.5.14.1	Abfrage der Option ZS13 .....	171
7.5.14.2	Schnellen AD-Wandler verwenden .....	171
7.5.14.3	Externspannung messen .....	171
7.5.14.4	Maximum Power Point Tracking .....	172
7.5.14.4.1	Funktion des MPP Trackings.....	173
7.5.14.4.2	Grenzen des MPP-Betriebes .....	176
7.5.14.4.3	Messdatenerfassung während des laufenden MPP-Betriebes .....	177
7.5.14.4.4	Einstellen des passenden Gerätestrombereiches für den Prüfling .....	178
7.5.14.4.5	Ermitteln des Delta_P_Min-Parameters für ein Panel .....	179
7.5.14.4.6	MPP Tracking Programmierbeispiel .....	180
7.5.14.4.7	Messen von Ampere-stunden und Wattstunden.....	181
7.5.14.5	Exponentialfunktion.....	181
<b>8</b>	<b>Software-Tools .....</b>	<b>183</b>
<b>9</b>	<b>Hersteller.....</b>	<b>184</b>

## Table Of Contents

<b>1</b>	<b>General Information .....</b>	<b>6</b>
1.1	Power On and Off .....	6
1.2	Remote and Local .....	6
<b>2</b>	<b>Serial Interface .....</b>	<b>7</b>
2.1	Setting the RS232 Parameters .....	7
2.2	USB Interface .....	10
2.3	Data Format of the RS232 Interface .....	10
2.4	Reading Data .....	10
<b>3</b>	<b> GPIB Interface .....</b>	<b>12</b>
3.1	Setting the GPIB Device Address.....	12
3.2	Data Format of the GPIB Interface.....	13
3.3	Reading Data .....	15
<b>4</b>	<b>Ethernet-RS-232 Converter (Option ZS25) .....</b>	<b>16</b>
<b>5</b>	<b>System Bus .....</b>	<b>17</b>
<b>6</b>	<b>Sub Addresses.....</b>	<b>19</b>
6.1	Single Addressing .....	19
6.2	Block Addressing .....	21
6.3	System Addressing .....	23
<b>7</b>	<b>SCPI Syntax Introduction .....</b>	<b>24</b>
7.1	Common Commands .....	24
7.2	Device Dependent Commands .....	24
7.2.1	Header Construction.....	24
7.2.1.1	Indentations.....	25
7.2.1.2	Aliases.....	25
7.2.2	White Space.....	26
7.2.3	Long and Short Form, Upper and Lower Case .....	26
7.2.4	Optional Keywords.....	27
7.2.5	Parameters .....	27
7.2.5.1	Numeric Values <NRf> .....	28
7.2.5.2	Units and Multipliers .....	28
7.2.5.3	Numerical Values and Extreme Values <num> .....	29
7.2.5.4	Boolean Parameter .....	30
7.2.5.5	Text .....	31
7.2.6	The Semicolon .....	31
7.2.7	Queries .....	33
7.2.8	Reading Data.....	33
7.3	SCPI Command Overview .....	34
7.3.1	Common Commands.....	34
7.3.2	Device Dependent Commands for ZS Series .....	35
7.4	SCPI Commands – Detailed Description .....	41
7.4.1	Common Commands.....	41
7.4.2	Device Dependent Commands .....	45
7.4.2.1	First Steps .....	45
7.4.2.2	Subsystem CHANnel INSTrument .....	47

7.4.2.3	Subsystem CURRent .....	49
7.4.2.4	Subsystem DATA TRACe .....	58
7.4.2.5	Subsystem DELay .....	61
7.4.2.6	Subsystem GTL .....	62
7.4.2.7	Subsystem INPut OUTPut .....	63
7.4.2.8	Subsystem LIST.....	64
7.4.2.9	Subsystem MEASure .....	77
7.4.2.10	Subsystem MODE FUNCTION.....	82
7.4.2.11	Subsystem PCYCLE .....	87
7.4.2.12	Subsystem POWer.....	96
7.4.2.13	Subsystem PROGram .....	101
7.4.2.14	Subsystem RESistance .....	105
7.4.2.15	Subsystem SETUp .....	112
7.4.2.16	Subsystem SFUNction.....	115
7.4.2.17	Subsystem STATus.....	124
7.4.2.18	Subsystem SYSTem.....	136
7.4.2.19	Subsystem TRIGger .....	151
7.4.2.20	Subsystem VOLTage.....	155
7.5	User-Oriented Command Sequence.....	165
7.5.1	Resetting the Device.....	165
7.5.2	Selecting the Operating Mode .....	165
7.5.3	Setting a Nominal Value .....	165
7.5.4	Load Input On and Off.....	166
7.5.5	Measuring Current, Voltage, Resistance and Power Values.....	166
7.5.6	Programmable Waveform .....	166
7.5.7	Data Acquisition .....	167
7.5.8	Battery Test Function .....	168
7.5.9	External Control .....	169
7.5.10	Full Fan Power .....	169
7.5.11	Switching Control Speed .....	169
7.5.12	Setting the Software Watchdog .....	170
7.5.13	Controlling the Power I/O Card (Option ZS07).....	170
7.5.14	Functions with ZS13 Option (Data Acquisition).....	171
7.5.14.1	Query for Option ZS13 .....	171
7.5.14.2	Using the Fast A/D Converter .....	171
7.5.14.3	Measuring External Voltage .....	171
7.5.14.4	Maximum Power Point Tracking .....	172
7.5.14.4.1	How does MPP Tracking work? .....	173
7.5.14.4.2	Limits of the MPP Mode .....	176
7.5.14.4.3	Data Acquisition during running MPP Mode .....	177
7.5.14.4.4	Determination of the suitable current range for the device under test.....	178
7.5.14.4.5	Evaluation of the „delta_P_min“ parameter of the panel .....	179
7.5.14.4.6	MPP Tracking Programming Example.....	180
7.5.14.4.7	Measuring Ah and Wh .....	181
7.5.14.5	Exponential Function .....	181
8	Software Tools.....	183
9	Manufacturer .....	184

## 1 Allgemeines

### 1.1 Ein- und Ausschalten

Nach dem Einschalten durchläuft die Last eine Initialisierungsroutine. Erst danach können am Gerät Einstellungen vorgenommen werden. Diese Initialisierung dauert ca. 10 Sekunden, in dieser Zeit blinkt die Anzeige „Limit“. Nach Abschluss der Initialisierung wechselt die Anzeige für die Regelzeitkonstante auf „Medium“. Das Gerät ist jetzt betriebsbereit.

Warten Sie nach dem Ausschalten des Gerätes ca. 5 Sekunden, bis Sie es wieder einschalten.

### 1.2 Remote und Local

Jeder Befehl, der über eine der Schnittstellen im Gerät ankommt – mit Ausnahme von MEAS:xx-Befehlen (s. SubSystem MEASure) – schaltet die Last in Remotezustand, bei dem keine manuellen Einstellungen möglich sind. Das Gerät kann entweder durch Drücken der Local-Taste oder durch Senden des Befehls *GTL* wieder in Handbetrieb gebracht werden.

## 1 General Information

### 1.1 Power On and Off

After switching the device on, the electronic load runs an initialisation routine which lasts approximately 10 seconds. During this initialisation the "Limit" display at the front panel blinks and no settings can be made at the load (even no remote settings). After the init routine the Control Speed is set to medium and the device is ready for operation.

After switching the device off, please wait 5 seconds before reactivating it.

### 1.2 Remote and Local

Any command the device receives by one of the interfaces – *MEAS:xx* commands excepted (see SubSystem *MEASure*) – sets the load to remote operation mode where no manual settings are possible. The device can be set to manual operation by pressing the Local key or by sending the *GTL* command.

## 2 Serielle Schnittstelle

Die optionale Serielle Schnittstelle erlaubt die Programmierung der Elektronischen Last in der Standard-Programmiersprache SCPI (Standard Commands for Programmable Instruments).

In der seriellen Schnittstelle werden die SCPI-Befehle auf das Format des Systembus umgewandelt und im 9Bit-Adressierungsformat an das Analog-Interface (nicht zu verwechseln mit der Analog-I/O-Schnittstelle) gesendet, welches die Befehle letztendlich erst ausführt.

Es sind verschiedene programmierbare Datenflusssteuerungs-Methoden (Handshake) verfügbar. Beschreibung s. Subsystem SYSTEM:COMMunication.

### 2.1 Einstellen der RS232-Parameter

Die Einstellungen der RS232-Schnittstelle werden über die zugehörige DIP-Schalterreihe konfiguriert. Nachdem ein oder mehrere DIP-Schaltereinstellungen verändert worden sind, muss das Gerät aus- und eingeschaltet werden, um die neuen Einstellungen zu übernehmen.

## 2 Serial Interface

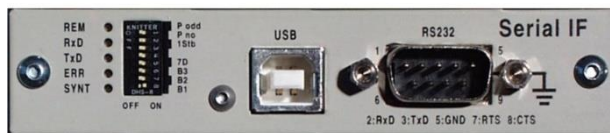
The optional serial interface allows the programming of the electronic load with standard programming language SCPI (Standard Commands for Programmable Instruments).

In the serial interface the SCPI command strings are converted to the system bus format and transferred to the Analog Interface (don't mix up with Analog I/O Interface) using a 9Bit addressing protocol. The Analog Interface executes the commands.

Several programmable flow control methods (handshake) are available. Description see subsystem SYSTEM:COMMunication.

### 2.1 Setting the RS232 Parameters

The RS232 interface is configured by the corresponding DIP switches. After adjusting one or more of the RS232 DIP switches the device has to be switched off and on, so that the new settings will be accepted.



Die mit B1 bis B3 bezeichneten Schalter dienen zur Einstellung der Baudrate. Diese wird folgendermaßen codiert:

Baudrate	B1	B2	B3
4800	Off	Off	Off
9600	On	Off	Off
14400	Off	On	Off
19200	On	On	Off
28800	Off	Off	On
38400	On	Off	On
57600	Off	On	On

Zur Einstellung der Datenlänge ist der Schalter 7D vorhanden.

Datenlänge	7D
7 Datenbits	On
8 Datenbits	Off

Die Anzahl der Stop-Bits wird durch den Schalter 1Stb festgelegt:

Stop Bits	1Stb
1	On
2	Off

Ob eine Prüfung der Parität erfolgt, wird mit dem Schalter P no festgelegt.

Parity	P no
Parity on	Off
Parity off	On

Wie das Paritätsbit zu bewerten ist, bestimmt Schalter P odd:

Parity	P odd
odd	On
even	Off

Bei der Auslieferung der Geräte werden folgende Werkseinstellungen für die RS232-Schnittstelle gemacht:

The switches B1 to B3 allow to set the baud rate. They should have one of the following settings:

Baud Rate	B1	B2	B3
4800	Off	Off	Off
9600	On	Off	Off
14400	Off	On	Off
19200	On	On	Off
28800	Off	Off	On
38400	On	Off	On
57600	Off	On	On

Switch 7D allows the setting of the data length.

Data Length	7D
7 data bit	On
8 data bit	Off

The amount of stop bits is determined by the switch 1Stb:

Number of Stop Bits	1Stb
1	On
2	Off

Switch Pno determines, whether a parity check occurs.

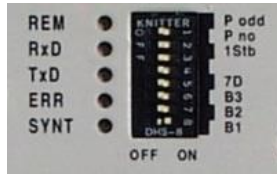
Parity	P no
Parity on	Off
Parity off	On

Switch S2 sets the parity bit:

Parity	P odd
odd	On
even	Off

When leaving the factory, the following settings are activated at the RS232 interface DIPs:



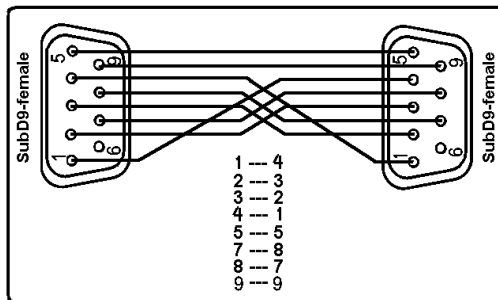


B1 = ON, alle anderen OFF, d. h.:  
9600 Baud, 8 Datenbits, 2 Stopbits, Even Parity

B1 = ON, all others OFF, that means:  
9600 Baud, 8 data bits, 2 stop bits, even parity.

Als RS232-Kabel ist das mitgelieferte H&H Standard Nullmodem-Kabel mit RTS-CTS-Handshake zu verwenden oder ein nach folgendem Verdrahtungsschema gefertigtes Kabel (Buchsen in Verdrahtungsansicht):

With RS232 communication the shipped H&H standard nullmodem cable with RTS-CTS handshake or a cable with the following pin assignment must be used (sockets in wiring view):



H&H Standard Nullmodem Cable

Die RxD- und TxD-Leitung (Pin 2 und 3) werden ausgekreuzt, d. h. Pin 2 der linken Buchse wird mit Pin 3 der rechten verbunden, Pin 3 der linken Buchse wird mit Pin 2 der rechten verbunden. Ebenso werden jeweils die RTS- und CTS-Leitungen (Pin 7 und 8) sowie die DCD- und DTR-Leitungen (Pin 1 und 4) ausgekreuzt verdrahtet. Pin 5 wird durchverdrahtet.

The RxD and TxD lines (pin 2 and 3) are cross-over, i.e. pin 2 of the left socket is wired to pin 3 of the right socket and pin 3 of the left socket is wired to pin 2 of the right socket. Also the RTS and CTS lines (pin 7 and 8) as well as the DCD and DTR lines (pin 1 and 4) are wired cross-over. Pin 5 is straightly connected from the left to the right connector.

## 2.2 USB-Schnittstelle

Um eine ZS Last über die USB-Schnittstelle ansteuern zu können, muss ein USB-Treiber, der von [www.hoecherl-hackl.de](http://www.hoecherl-hackl.de) heruntergeladen werden kann, auf dem betreffenden PC installiert werden. Dazu wird die eingeschaltete Last über ein handelsübliches USB-AB-Kabel an den laufenden PC angesteckt. Der Computer erkennt die neue Hardware und fragt nach dem entsprechenden Treiber. Geben Sie den Pfad auf den heruntergeladenen und entpackten Treiber an.

Das Installationsprogramm installiert einen "Virtuellen COM-Port" auf dem PC, dessen COM-Port-Nummer im Gerätemanager der Systemsteuerung ersichtlich ist. S. auch Application Note FTDI\_Drivers\_Installation\_Guide\_for\_Windowsxx.pdf im heruntergeladenen ZIP File.

## 2.3 Datenformat bei RS232 und USB

Die serielle Schnittstelle erwartet als Terminierungszeichen (Endekennung) das Zeichen LF (10 dez.).

Das Gerät sendet als Terminierungszeichen ebenfalls ein LF (10dez.).

## 2.4 Auslesen von Daten



Daten müssen nach einem Abfragebefehl unbedingt komplett ausgelesen werden bevor der nächste Abfragebefehl gesendet wird.

## 2.2 USB Interface

To control a ZS load via the USB interface a USB driver available from [www.hoecherl-hackl.com](http://www.hoecherl-hackl.com) must be installed on the concerning PC. To do this just connect the running load to the running PC with a usual USB-AB cable. The computer detects the new hardware and asks for the concerning driver. Set the path to the downloaded and unzipped driver.

The installation program installs a "Virtual COM Port" on the PC which COM port number is displayed in the Windows Control Panel's Device Manager. See also FTDI\_Drivers\_Installation\_Guide\_for\_Windowsxx.pdf in the downloaded ZIP file.

## 2.3 Data Format of the RS232 Interface

The RS232 interface expects the character <LF> (10dec.) as end of string sign.

After a measurement data request there is also sent <LF> (10dec.) as end of string sign.

## 2.4 Reading Data



Requested data have to be read completely before the next request for data is sent.

Folgende Leuchtdioden signalisieren den Zustand der seriellen Schnittstelle:

LED Name	Bedeutung
REM	Remote: Daten-Interface ist im Fernsteuerzustand, leuchtet beim ersten Startbit
RxD	Daten-Interface ist beim Datenempfang
TxD	Daten-Interface sendet Daten
ERR	Error: Fehler bei der Datenübertragung (nur bei RS232)
SYNT	SCPI Syntaxfehler (gilt auch im GPIB-Betrieb)

The following LEDs show the status of the serial interface:

LED Name	Function
REM	Remote: Data Interface is in remote state, lights up after first start bit
RxD	Data Interface is receiving data
TxD	Data Interface is sending data
ERR	Error in data transmission (only at RS232)
SYNT	SCPI syntax error (also for GPIB operation)

### 3 GPIB Interface

Zusätzlich zum RS232-Interface kann das GPIB-Interface in die Last eingesteckt werden, welche ebenfalls die Programmierung in SCPI erlaubt. Die GPIB-Schnittstelle beinhaltet lediglich die erforderliche GPIB Hardware, die Verarbeitung der SCPI-Befehle erfolgt im RS232-Interface, weshalb zur Verwendung der GPIB-Schnittstelle immer auch das RS232-Interface nötig ist.



#### Einstellen der Interfaceparameter

Die GPIB-Schnittstellenkarte kann über DIP Schalter auf der Geräterückseite eingestellt werden.

#### 3.1 Einstellen der GPIB-Geräteadresse

Die Einstellung der GPIB Geräteadresse erfolgt im Binärsystem.

### 3 GPIB Interface

Additionally to the RS232 interface the GPIB interface can be inserted into the load which also allows the programming in SCPI syntax. The GPIB interface merely contains the required GPIB hardware. The procession of the SCPI commands happens in the RS232 interface. Therefore you always need also the RS232 interface with the GPIB interface.

#### Setting the Interface Parameters

The setting of the GPIB interface functions is done by using the DIP switches at the back panel of the device.

#### 3.1 Setting the GPIB Device Address

The setting of the GPIB device address is done binary-coded.



Schalter	Wert
A1	1
A2	2
A3	4
A4	8
A5	16

Um eine bestimmte Adresse einzustellen, muss anhand des Wertes die Schalterstellung ermittelt werden.

Beispiel: Adresse 10  
Schalter A4 und A2 auf ON

Bei der Auslieferung ist die GPIB Adresse auf "7" eingestellt, d.h. A1, A2, A3 auf ON, A4 und A5 auf OFF.

Nachdem ein oder mehrere DIP-Schaltereinstellungen verändert worden sind, muss das Gerät aus- und eingeschaltet werden, um die neuen Einstellungen zu übernehmen.

### 3.2 Datenformat bei GPIB

Bei Betrieb über die GPIB-Schnittstelle können beliebige ASCII-Zeichen gesendet werden.

Beim Empfang von Daten erwartet das GPIB Interface eine der folgenden Kombinationen:

DB EOI	DB+LF	DB+LF EOI
-----------	-------	--------------

DB = Datenbyte, LF = Line Feed, EOI = End or Identify-Leitung

Neben den 5 Adressschaltern sitzen 3 weitere Schalter zur Einstellung der Endezeichen, die das Gerät beim Senden (Talken) verwenden soll.

Switch	Value
A1	1
A2	2
A3	4

A4	8
A5	16

To set a specific address, the switch position has to be determined depending on the value.

Example: Address 10  
Switch A4 and A2 are set ON

The default address when leaving the factory is "7", i.e. A1, A2, A3 at ON position, A4 and A5 at OFF position. After adjusting the GPIB device address the device has to be switched off and on, so that the new device address will be accepted.

### 3.2 Data Format of the GPIB Interface

When operating with the GPIB interface any ASCII characters may be sent.

As end of string sign the GPIB interface expects one of the following combinations:

DB EOI	DB+LF	DB+LF EOI
-----------	-------	--------------

DB = Data Byte, LF = Line Feed, EOI = End or Identify

Besides the 5 address switches there are 3 further switches which are used to set the termination characters when the device sends data (Talker).

Bei der Auslieferung sind alle Endezeichen aktiviert (ON).

Wenn in dieser Konfiguration Messwerte vom Gerät angefordert wurden, so liefert das Gerät die Daten in folgendem Format:

**SD.DDDDDDESDD<CR><LF>**  
**EOI**

S: Vorzeichen, + oder -  
D: numerische Stellen  
E: Exponent-Zeichen  
<CR> Carriage Return  
<LF> Line Feed  
<EOI> End Or Identify- Leitung

Mit dem letzten aktivierten Endezeichen (CR oder LF) wird die Leitung EOI gesetzt, wenn der Schalter "EOI" auf "ON" steht. Ist kein Endezeichen aktiviert und EOI auf "ON", wird die EOI-Leitung mit dem letzten Datenbyte gesetzt.

Ist weder ein Endezeichen noch EOI aktiviert, muss das Lesen vom Gerät abhängig von der Anzahl der zu erwartenden Zeichen beendet werden (nicht zu empfehlen).



Wenn über das GPIB Interface im Gerät gespeicherte Messdatensätze gelesen werden, muss der LF Schalter auf ON gestellt werden.

Nachdem ein oder mehrere DIP-Schaltereinstellungen verändert worden sind, muss das Gerät aus- und eingeschaltet werden, um die neuen Einstellungen zu übernehmen.

When leaving the factory, all termination characters are activated (ON).

When measurement data have been requested in this configuration, the device supplies the data in the following format:  
**SD.DDDDDDESDD<CR><LF>**  
**EOI**

S: Sign, + or -  
D: Numeric digits  
E: Exponent  
<CR> Carriage Return  
<LF> Line Feed  
<EOI> End Or Identify Line

The <EOI> line - if activated - is set with the last activated termination character (CR or LF).

If no termination character is activated and the EOI switch is "ON", the EOI line will be set with the last data byte.

If you have activated neither a termination character nor EOI you must terminate the reading from the device after the last data byte (not recommended!).



If you want to read data sets saved in the electronic load then you must set the LF switch to ON position.

After adjusting one of the GPIB DIP switches the device has to be switched off and on, so that the new setting will be accepted.

### 3.3 Auslesen von Daten



Daten müssen nach einem Abfragebefehl unbedingt komplett ausgelesen werden bevor der nächste Abfragebefehl gesendet wird.

Folgende Leuchtdioden signalisieren den Zustand der GPIB-Schnittstelle:

LED Name	Bedeutung
REM	Remote: Daten-Interface ist im Fernsteuerzustand
LI	Listen: Daten-Interface ist beim Datenempfang
TA	Talk: Daten-Interface sendet Daten
SRQ	Service Request: Bedieneranforderung gesetzt
ERR	Error: Fehler bei der Datenübertragung

### 3.3 Reading Data



Requested data have to be read completely before the next request for data is sent.

The following LEDs show the status of the GPIB interface:

LED Name	Function
REM	Remote: Data Interface was addressed and is in remote state
LI	Listen: Data Interface is receiving data
TA	Talk: Data Interface is sending data
SRQ	Service Request is set
ERR	An error in the data transmission occurred or command not allowed

## **4 Ethernet-RS-232- Konverter (Option ZS25)**

Die Installations- und Konfigurationsanweisungen für Option ZS25 finden Sie auf dem mitgelieferten USB-Stick.

## **4 Ethernet-RS-232 Converter (Option ZS25)**

The installation and configuration instructions for Option ZS25 can be found on the supplied USB flash drive.



## 5 Systembus

Dieser Punkt ist nur maßgebend bei der Steuerung von mehreren Geräten über eine gemeinsame Datenschnittstelle.



Wenn im Gerät ein Systembus-Interface eingebaut ist, muss das Gerät mit der am Systembus-Interface eingestellten Unteradresse adressiert werden, selbst wenn der Systembus gar nicht verwendet wird.

Bei Verwendung eines Einzelgerätes in Verbindung mit einer RS-232- oder GPIB-Schnittstelle ohne Systembus-Interface ist dieses Kapitel für das Verständnis der Gerätefunktionen nicht erforderlich. Das Handbuch kann ab Punkt 7 weiterverfolgt werden.

Bei Verwendung der Systembus-Option ZS04 oder ZS05 können bis zu 192 Geräte über eine gemeinsame GPIB- oder RS232-Schnittstelle adressiert werden.

## 5 System Bus

This section applies only for the control of several devices by the System Bus.

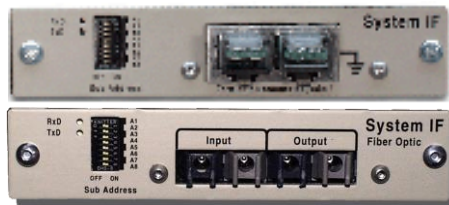


If there is a system bus interface installed in the device it must be addressed with the sub address set at the system interface's DIP switches even when the system bus is not used.

For using a single device with RS-232 or GPIB interface without system bus interface the following section is not necessary for understanding the device functions.

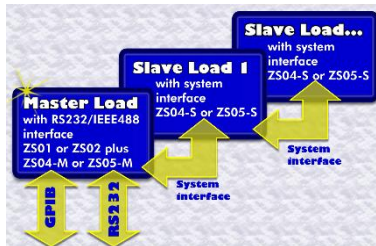
In this case continue the manual with section 7.

When operating with the System Bus Option ZS04 or ZS05 up to 192 devices can be addressed by one single GPIB or RS232 interface.



Die vom Steuerrechner per Dateninterface ankommenden Daten werden im Interface auf das Format des Systembusses umgeformt und seriell durch alle Geräte geschickt.

The data received from the controlling computer via GPIB or RS-232 interface are transformed into the format of the System Bus and are sent serially through all devices.



Messwerte, die von den Geräten über den Systembus kommen, werden wiederum vom Dateninterface auf ASCII-Format gebracht und an den Steuerrechner übergeben.

Durch ein Kabel wird der SYS-BUS-OUTPUT mit dem SYS-BUS-INPUT des nachfolgenden Gerätes verbunden.



#### ACHTUNG!

Bei der Systembusverschaltung mit LWL (Fiber Optic) darf der SYS-BUS-OUTPUT nur mit dem SYS-BUS-INPUT des nachfolgenden Gerätes verbunden werden.

Measurement values from the devices sent on the system bus are transferred by the SCPI interpreter located on the RS232 board into ASCII format and sent to the controlling PC.

The System IF Output of one device is connected to the System IF Input of the following device by the enclosed system bus cable.



#### ATTENTION!

The SYS-BUS Output of a fiber optic system may be connected only with the SYS-BUS Input of the following device.

## 6 Unteradressen

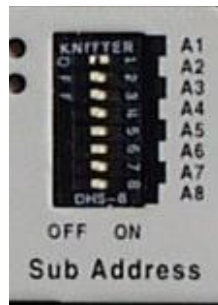
### 6.1 Einzeladressierung

Um bei der Programmierung die einzelnen Geräte voneinander unterscheiden zu können, ist jedem Gerät eine Unteradresse zugewiesen, die an der DIP-Schalterreihe "Sub Address" eingestellt wird. Die Unteradresse ist binär zu decodieren mit A1 als niederwertigstes Bit. Bei einem Einzelgerät muss Unteradresse 1 eingestellt werden (A1 ON, alle anderen OFF):

## 6 Sub Addresses

### 6.1 Single Addressing

To distinguish the different devices for programming a sub address is allocated for every device which is set at the DIP switches "Sub Address" of the System IF card. The sub address is binary coded with A1 as least significant bit. When a device is used as a stand-alone device then sub address 1 has to be set (A1 ON, all others OFF):



Die Unteradresse wird nur einmal nach dem Einschalten vom DIP-Schalter gelesen. Wird die Unteradresse beim laufenden Gerät verändert, so muss das Gerät aus- und eingeschaltet werden, um die neue Unteradresse zu übernehmen.



**Unteradresse 0 (alle Sub Address DIPs OFF) ist nicht erlaubt.**

Um ein bestimmtes Gerät zu programmieren, wird dem Gerätebefehl die entsprechende Unteradresse vorangestellt.

Um dem Gerät anzuzeigen, daß es sich dabei um eine Unteradresse handelt, wird die Zeichenkette "CHANnel" oder "INSTrument" vor die Nummer gesetzt.

Beispiel zum Einschalten des Geräteeinganges:

**CHAN 3;INP ON**

Lasteingang von Gerät mit der Unteradresse 3 wird eingeschaltet

**CHAN 22;INP ON**

Lasteingang von Gerät mit Unteradresse 22 wird eingeschaltet

Sollen mehrere Kommandos an dasselbe Gerät gesendet werden, so braucht die Unteradresse nur einmal zu Beginn der Befehlskette gegeben zu werden.

Beispiel zur Programmierung des Laststromes von 1A mit Einschalten des Geräteeinganges:

**CHAN 3;CURR 1;INP ON** oder

**CHAN 3::CURR 1::INP ON**

Gerät mit der Unteradresse 3 stellt 1A ein und schaltet den Eingang ein.

The device reads the sub address from the DIPs when it is switched on. When a device's sub address is changed while it is switched on then the device has to be switched off and on to recognize the new sub address.



**Sub address 0 (all DIPs OFF) is not allowed!**

To program a specific device the sub address is put first to the device command.

To indicate that this is a sub address the string "CHANnel" or "INSTrument" is put first.

Example: Switching on the device input, command "INPut ON" is used.

**CHAN 3;INP ON**

Load input of device with address 3 is switched on

**CHAN 22;INP ON**

Load input of device with sub address 22 is switched on

When there are several commands to be sent to one device the secondary address only has to be specified at the beginning of the command string.

Example:

Programming of the load current 1A, switching the device input on:

**CHAN 3;CURR 1;INP ON** or

**CHAN 3::CURR 1::INP ON**

Device with sub address 3 sets 1A and switches the input on.

Eine Zeichenkette kann maximal 512 Zeichen lang sein.

In der Zeichenkette kann auch die Unteradresse von weiteren Geräten enthalten sein.

Beispiel:

**CHAN 1;;INP ON;;CHAN 2;INP OFF**

Gerät #1 Eingang ein

Gerät #2 Eingang aus

Wenn ein Gerät durch Erkennen der eigenen Unteradresse einmal adressiert wurde, so bleibt dieser Zustand erhalten, bis ein anderes Gerät adressiert wird. Das heißt, dass die Befehle vom adressierten Gerät solange ausgeführt werden, bis durch die Adressierung eines anderen Gerätes das erste entadressiert wird.

Beispiel:

**CHAN 3;INP ON**

Gerät #3 wird adressiert  
#3 Eingang ein

**CURR 1.2**

Gerät #3 1.2 A

**INP OFF** Gerät #3 Eingang aus

**CHAN 7;INP ON**

Gerät #7 wird adressiert  
#3 wird entadressiert  
#7 Eingang ein

**CURR 0.15**

Gerät #7 0.15 A

## 6.2 Blockadressierung

Bei Verwendung einer größeren Anzahl von Geräten kommt es häufig vor, dass mehrere Geräte mit der gleichen Einstellung versehen werden müssen. Dazu ist es sehr umständlich, jedes Gerät über die Einzeladressierung zu programmieren.

A character string may contain maximum 512 characters.

The secondary addresses from further devices may be contained in the same character string.

Example:

**CHAN 1;;INP ON;;CHAN 2;;INP OFF**

device #1 input on

device #2 input off

When a device has been addressed by a secondary address, this status remains constant until another device will be addressed.

That means, the commands from the addressed device are executed until another device is addressed so that the first device is de-addressed.

Example:

**CHAN 3;INP ON**

device #3 Input on  
#3 is addressed

**CURR 1.2**

device #3 1.2 A

**INP OFF** device #3 input off

**CHAN 7;INP ON**

device #7 input on  
#3 is de-addressed  
#7 is addressed

**CURR 0.15**

device #7 0.15 A

## 6.2 Block Addressing

For use of several devices it often occurs that a group of devices have to be adjusted with the same setting. It is very toilsome to address every single device by discrete addressing.

Hier ist die Blockadressierung eine komfortablere Methode, einer umgrenzten Gruppe von Geräten die gleichen Befehle ausführen zu lassen. Dazu werden eine Anfangs- und eine Endadresse angegeben, die durch einen Doppelpunkt (:) getrennt sind.

Beispiel:

#### **CHAN 3:15;INP ON**

Gerät 3 bis Gerät 15 Eingang einschalten

#### **CHAN 20:50;INP OFF**

Gerät 20 bis Gerät 50 Eingang ausschalten

Werden die Blockgrenzen nicht aufsteigend programmiert, dreht der SCPI-Interpreter im Digital-Interface die Reihenfolge automatisch um, so dass beide folgende Blockangaben möglich sind:

**CHAN 8:3**

**CHAN 3:8**

Entsprechend wie bei der Einzeladressierung bleibt auch der Zustand der Adressierung für einen Block erhalten, bis einzelne Geräte oder ein anderer Block neu adressiert werden.

Die Blockadressierung ist bei Kommandos, die eine Antwort des Gerätes auslösen (z.B. Messfunktionen), wegen möglicher Kollision der Messdaten unzulässig. Deshalb werden bei Abfragen in Blockadressierung keine Daten vom Gerät gesendet.

Group addressing is a comfortable way to instruct a group of devices to execute the same commands. For this purpose, a start and end address is set, separated by a colon (:).

Example:

#### **CHAN 3:15;INP ON**

Device 3 to device  
15 input on

#### **CHAN 20:50;INP OFF**

Device 20 to device  
50 input off

The SCPI interpreter automatically re-orders block boundaries if they are not programmed in ascending order. Therefore both of the following commands are valid:

CHAN 8:3

CHAN 3:8

Analog to the discrete addressing the status of the addressing for a group remains constant, until single devices or other groups of devices are addressed.

The group addressing is not allowed for commands that require an answer of a device (measurement and status functions), because the measurement data could collide with each other.

### 6.3 Systemadressierung

Zum Adressieren aller angeschlossenen Geräte ist die Systemadresse 0 vorgesehen.

Wird einem Kommando die Systemadresse 0 vorangestellt, so wird das Kommando von allen am Systembus angeschlossenen Geräten ausgeführt. Besonders nützlich ist die Systemadresse, wenn z.B. alle Geräte in den Grundzustand versetzt werden sollen.

Beispiel:

**CHAN 0;\*RST** System-Reset

Die Systemadressierung ist bei Kommandos, die eine Antwort des Gerätes auslösen (z.B. Messfunktionen), wegen möglicher Kollision der Messdaten unzulässig. Deshalb werden bei Abfragen bei Systemadressierung keine Daten vom Gerät gesendet.

### 6.3 System Addressing

To address all connected devices, the system address 0 is provided.

When a command puts first the system address 0, the command will be executed from all devices that are connected to the system bus. This is especially useful when all devices have to be reset.

Example:

**CHAN 0;\*RST** System Reset

The system addressing is not allowed for commands that require an answer of a device (measurement and status functions), because the measurement data could collide with each other. Therefore a device won't send any answer in system addressing mode.

## 7 SCPI-Syntax Einführung

Der SCPI-Standard (Standard Commands for Programmable Instruments) beschreibt einen einheitlichen Befehlsatz zur Programmierung von Geräten, unabhängig von Gerätetyp und Hersteller. Es sollen damit die gerätespezifischen Befehle vereinheitlicht werden.

### 7.1 Common Commands

Common Commands sind geräteunabhängige Befehle, die in der Norm IEEE 488.2 definiert sind. Sie bestehen aus einem Stern (\*) und drei Buchstaben mit evtl. folgendem Parameter. Abfragebefehle werden durch Anhängen eines Fragezeichens gebildet.

#### Beispiele:

\*RST     Reset; Gerät in Grundzustand versetzen  
\*ESE 9    Bits 0 und 3 in ESE-Register setzen  
\*IDN?     Identifikationsstring lesen

### 7.2 Gerätespezifische Befehle

#### 7.2.1 Aufbau des Headers

Die Struktur der gerätespezifischen Befehle ist hierarchisch aufgebaut. Ein Befehl besteht aus einem sog. Header und eventuell einem oder mehreren folgenden Parametern, durch ein "White Space" (s.u.) vom Header getrennt.

Der Header besteht aus einem oder mehreren Schlüsselwörtern, die wiederum durch einen Doppelpunkt (:) voneinander getrennt werden.

## 7 SCPI Syntax Introduction

The SCPI Standard (Standard Commands for Programmable Instruments) includes a standardized command set for programming devices, independent of device type and manufacturer. In this way the device dependent commands are unified.

### 7.1 Common Commands

Common Commands are device independent commands that are defined in the standard IEEE 488.2. They include an asterisk (\*) and three letters with optional parameter.

Query commands are built by post-fixing a question mark.

#### Examples:

\*RST     Reset  
\*ESE 9    Set Bits 0 and 3 in ESE  
\*IDN?     Read identification string

### 7.2 Device Dependent Commands

#### 7.2.1 Header Construction

The device dependent commands are hierarchically structured.

A command contains a so called Header as well as one or more parameters, separated by a white space from the header.

The header contains one or more keywords, that are separated by a colon (:).



### 7.2.1.1 Einrückungen

Die verschiedenen Ebenen der Befehlshierarchie sind in der Übersicht durch Einrücken nach rechts dargestellt. Je tiefer die Ebene liegt, desto weiter wird nach rechts eingerückt.

Beispiel: Befehlssystem CURRent :

```
CURRent
[:LEVel]
  [:IMMEDIATE] <num>
  [:IMMEDIATE]?
  :TRIGgered <num>
  :TRIGgered?
:PROT <num>
:PROT?
:RANGe <num>
:AUTO <Boolean>
:RANGe?
```

Um beispielsweise einen Strombereich von 10A einzustellen, muss folgender String an das Gerät gesendet werden:  
CURR:RANG 10

### 7.2.1.2 Aliases

Für einige Befehle existiert eine Auswahl an Schlüsselwörtern mit identischer Wirkung. Diese Schlüsselwörter werden in der Befehlsübersicht in der gleichen Zeile angegeben, durch einen senkrechten Strich (|) getrennt.

Nur eines der alternativen Schlüsselwörter darf im Befehlsstring angegeben werden. Die Wirkung des Befehls ist unabhängig von der Auswahl der Alternative.

Beispiel: Befehlssystem INPut:

```
INPut|OUTPut
[:STATe]
[:STATe]? <Boolean>
```

Der Befehl

```
INPut ON
```

führt zum gleichen Ergebnis wie

```
OUTP ON          oder
```

```
OUTP 1
```

### 7.2.1.1 Indentions

The levels of the command hierarchy are identified by indention to the right. The deeper the level, the more it is indented to the right.

Example: Command System CURRent :

```
CURRent
[:LEVel]
  [:IMMEDIATE] <num>
  [:IMMEDIATE]?
  :TRIGgered <num>
  :TRIGgered?
:PROT <num>
:PROT?
:RANGe <num>
:AUTO <Boolean>
:RANGe?
```

To set a current range of 10A, the following string has to be sent to a device:  
CURR:RANG 10

### 7.2.1.2 Aliases

For some commands there are several keywords with identical effect. These keywords are shown in the command syntax within one line, separated through a vertical bar (|).

In a command string only one of the alternative keywords may be specified. The result of the command is not dependent of using a particular alternative.

Example: Command System INPut:

```
INPut|OUTPut
[:STATe]
[:STATe]? <Boolean>
```

The command

```
INPut ON
```

has the same result as

```
OUTP ON          or
```

```
OUTP 1
```

Im Parameterfeld der Befehlsübersicht dient der senkrechte Strich (|) zur Angabe von erlaubten Parametern.

### 7.2.2 White Space

Zum "White Space" gehören alle Zeichen mit dem ASCII-Code von 0 bis 9 dez. und von 11 bis 32 dezimal.

Das Zeichen LineFeed (10dez.) ist also vom White Space ausgeschlossen. Dieses dient zur Erkennung des Stringendes.

Das White Space wird benutzt, um einen Parameter vom Header zu trennen. Es dürfen mehrere White Spaces nacheinander folgen.

### 7.2.3 Lang- und Kurzform, Groß- und Kleinschreibung

Es gibt bei den Schlüsselwörtern eine Kurz- und eine Langform (soweit das Wort aus mehr als vier Zeichen besteht). Es kann entweder nur die Kurzform oder die vollständige Langform eines Schlüsselwortes angegeben werden. Andere Abkürzungen sind nicht erlaubt und verursachen einen Syntaxfehler.

Zur Unterscheidung wird in diesem Handbuch die Kurzform in Großbuchstaben geschrieben. Der restliche String, der zusammen mit der Kurzform die Langform ergibt, wird in Kleinbuchstaben an die Kurzform angehängt.

Das Gerät selbst unterscheidet nicht zwischen Groß- und Kleinschreibung.

Es gibt z.B. folgende Möglichkeiten, einen getriggerten Strom von 5A zu programmieren:  
**CURRENT:TRIG 5**  
*curr:triggered 5*  
**Curr:TRig 5**

In the parameter field of the syntax the vertical bar (|) describes allowed parameters.

### 7.2.2 White Space

"White Space" includes all characters with ASCII code from 0 to 9 dec. and from 11 to 32 dec.

The character LineFeed (10dec) is not part of white space. It determines the end of the string.

White Space is used to separate the parameters from the header. Several white space characters may be combined.

### 7.2.3 Long and Short Form, Upper and Lower Case

Keywords are provided in long and short format (if the word contains more than four characters).

Both formats are allowed. All other abbreviations are not supported and result in a syntax error.

This manual shows the short form in upper case, to allow a distinction. The remaining string, that builds in combination with the short form the long form, is appended to the short form.

The device itself doesn't distinguish between upper case and lower case letters.

To program a triggered current of 5A there are several methods:

**CURRENT:TRIG 5**  
*curr:triggered 5*  
**Curr:TRig 5**

nicht jedoch: *CURR:TRIGGER 5*

Für möglichst kurze Ausführzeiten ist die Kurzform zu bevorzugen.

#### 7.2.4 Wahlweise einfügbare Schlüsselwörter

In manchen Befehlssystemen ist es aus Gründen der SCPI-Konformität möglich, bestimmte Schlüsselwörter wahlweise in den Header einzufügen oder auszulassen.

In dieser Beschreibung sind solche Wörter durch eckige Klammern gekennzeichnet.

Beachten Sie, dass sich der Befehlsstring durch Weglassen der wahlweisen Schlüsselwörter erheblich verkürzen kann.

Beispiel: Laststrom 10A  
*CURRent[:LEVel][:IMMEDIATE] 10*  
lässt sich verkürzen zu:  
*CURR 10*

#### 7.2.5 Parameter

Zu den meisten Befehlen muss an den Header ein Parameter angehängt werden (mit White Space getrennt, s.o.).

Je nach dem erkannten Header wird vom Gerät ein bestimmter Parametertyp erwartet. Dieser Typ kann sein: Zahlenwert, Boolean, Text

Werden bei bestimmten Befehlen mehrere Parameter verlangt, so werden diese durch ein Komma (,) voneinander getrennt.

Beispiel:  
*TRANSient:MODE PULSe,5*

but not: *CURR:TRIGGER 5*

For shortest possible execution times you should use the short form.

#### 7.2.4 Optional Keywords

In some command systems it is possible to optionally use certain keywords in the header, to guarantee SCPI conformity.

These words are marked using brackets ([]).

Note that the command string can be considerably shortened by omitting the optional keywords.

Example: Load Current 10A  
*CURRent[:LEVel][:IMMEDIATE] 10*  
can be reduced to:  
*CURR 10*

#### 7.2.5 Parameters

For most commands parameters have to be appended to the header (separated through white space).

Depending on the recognized header the device expects a certain parameter type: Numeric, Boolean, String

If a command needs several parameters, they are separated by comma (,).

Example:  
*TRANSient:MODE PULSe,5*

### 7.2.5.1 Zahlenwerte <NRf>

Zahlenwerte können grundsätzlich in jeder gebräuchlichen dezimalen Form angegeben werden: als Ganzzahl, Float, Engineering Format.

Dezimaltrennzeichen ist der Punkt (.), kein Komma!

Als Platzhalter für Zahlenwerte steht in der Befehlsübersicht <NRf>.

Beispiel (Widerstand 0.558 Ohm):

RESistance 55.8E-2

RES .558

Die Länge des gesamten Zahlenstrings (Vorzeichen, Vor-, Nachkommastellen, Dezimalpunkt, Exponent) ist auf 16 Zeichen begrenzt:

+1.234567890E+01

255.000000000000

### 7.2.5.2 Einheiten und Multiplizierer

Nach den meisten Zahlenwerten kann die Einheit (Suffix) mit angegeben werden.

Außerdem kann vor die Einheit ein Multiplizierer gesetzt werden.

Gebräuchliche Multiplizierer sind bei den elektronischen Lasten:

Mnemonic	Definition	Multiplikator
M	Milli	$10^{-3}$
K	Kilo	$10^3$
MA	Mega	$10^6$

In Bezug auf die physikalische Größe sind bei den elektronischen Lasten grundsätzlich folgende Einheiten erlaubt:

### 7.2.5.1 Numeric Values <NRf>

Numeric values may be provided in every common decimal format: as integer, float or engineering format.

The decimal separator is the dot (.), no comma!

In the syntax the dummy <NRf> is used for numerical values.

Example (Resistance 0.558 Ohms):

RESistance 55.8E-2

RES .558

The length of the entire number string must not exceed 16 characters (sign, digits, decimal point, exponent):

+1.234567890E+01

255.000000000000

### 7.2.5.2 Units and Multipliers

For most numerical values the unit may be specified (suffix).

In front of the unit a multiplier can be set. Common multipliers for electronic loads are:

Mnemonic	Definition	Multiplier
M	Milli	$10^{-3}$
K	Kilo	$10^3$
MA	Mega	$10^6$

For the physical dimension the following units are supported for electronic loads:

Größe	Einheit	Beschreibung
Strom	A	Ampere
	MA	Milliampere
Widerstand	OHM	Ohm
	KOHM	Kiloohm
	MOHM	Megohm (!) <sup>*)</sup>
Leistung	W	Watt
	MW	Milliwatt
	KW	Kilowatt
Spannung	V	Volt
	MV	Millivolt
Zeit	S	Sekunde
	MS	Millisekunde

<sup>\*)</sup> Um zwischen den Multiplizierern "Milli" ( $10^{-3}$ ) und "Mega" ( $10^6$ ) zu unterscheiden, ist normalerweise die Abkürzung für Milli 'M' und für Mega 'MA'. Eine Ausnahme gibt es bei der Widerstandseinheit. Hier gibt es keine Einheit für 'Milliohm'. Die Einheit 'MOHM' bedeutet immer MegaOhm!

Beispiel (Laststrom 520mA):

```
CURR 520MA
CURR 0.52
CURR 520E-3
```

### 7.2.5.3 Zahlenwerte und Extremwerte <num>

Bei den meisten Befehlen, die einen Zahlenwert als Parameter haben, können außer eines spezifischen Zahlenwertes die Werte MIN und MAX angegeben werden.

MIN bezeichnet den kleinstmöglichen Wert, den ein Parameter annehmen kann (meist 0).

MAX bezeichnet den größtmöglichen Wert eines Parameters.

Als Platzhalter für Zahlenparameter, die auch MIN oder MAX enthalten dürfen, steht in der Befehlsübersicht <num>.

Dimension	Unit	Description
Current	A	Amp
	MA	Milliamp
Resistance	OHM	Ohm
	KOHM	Kiloohm
	MOHM	Megohm(!) <sup>*)</sup>
Power	W	Watt
	MW	Milliwatt
	KW	Kilowatt
Voltage	V	Volt
	MV	Millivolt
Time	S	Second
	MS	Millisecond

<sup>\*)</sup> To distinguish between the multipliers "Milli" ( $10^{-3}$ ) and "Mega" ( $10^6$ ), the abbreviations "M" for Milli and "MA" for Mega are used. One exception is the resistance unit. There is no unit for "Milliohm". The unit "MOHM" always means MegaOhm!

Example (Load Current 520mA):

```
CURRENT 520MA
CURRENT:IMM 0.52
```

### 7.2.5.3 Numerical Values and Extreme Values <num>

For most commands that use a numeric value as parameter, the values MIN and MAX can be specified.

MIN describes the smallest possible value for a parameter (mostly 0).

MAX describes the highest possible value for a parameter.

As dummy for a numeric parameter, that can contain MIN and MAX, the syntax uses <num>.

Beispiel: maximalen Strom einstellen:  
*CURRent MAX*

An MIN und MAX darf kein Suffix angehängt werden.

Minimal- und Maximalwert eines Zahlenparameters können durch Abfrage ermittelt werden. Dazu wird nach dem Fragezeichen ein White Space und MIN bzw. MAX angehängt.

Beispiel: Ermittlung des maximalen Laststromes:  
*CURR? MAX* liefert bei ZS2606  
im größten Bereich:  
 $+3.000000E+02$

#### 7.2.5.4 Boolesche Parameter

Einige Befehle verlangen einen Booleschen Parameter, z.B. der Befehl zum Schalten des Geräteeinganges:  
*INPut ON*

Boolesche Parameter haben zwei logische Zustände. Der logische Zustand "TRUE" wird durch den Parameter ON oder den Zahlenwert 1 repräsentiert. Entsprechend steht für den Zustand "FALSE" der Parameter OFF oder 0.

Bei der Programmierung eines booleschen Parameters ist es egal, ob die Zahlenform oder die Textform gewählt wird.

So hat z.B. der Befehl  
*INPut ON* die gleiche Wirkung wie  
*INPut 1*

Bei der Abfrage von booleschen Zuständen wird immer der boolesche Zahlenwert geliefert.

Beispiel:  
*INPut?* (Antwort: 1)

Example: Set maximal current:  
*CURRent MAX*

MIN and MAX must not be followed by a suffix.

The minimal and maximal value of a numeric parameter can be determined by query. To do so, a white space as well as MIN or MAX are appended after the question mark.

Example: Determining the maximal load current:  
*CURR? MAX* results for ZS2606  
in the high range:  
 $+3.000000E+02$

#### 7.2.5.4 Boolean Parameter

For some commands a Boolean parameter has to be provided, for example to switch the device input:  
*INPut ON*

Boolean parameters can take two logic values. The logic value "TRUE" is represented by the parameter ON or the numeric value 1. The state "FALSE" is represented by the parameter OFF or 0.

For programming a boolean parameter it doesn't matter whether the numeric form or the text form is used:

The command  
*INPut ON* has the same result as  
*INPut 1*

For the query of boolean states always the boolean numeric values are returned.

Example:  
*INPut?* (Response: 1)

### 7.2.5.5 Text

Textparameter folgen den syntaktischen Regeln für Schlüsselwörter, besitzen also eine Lang- und eine Kurzform. Die Trennung vom Header erfolgt wie bei jedem Parameter durch ein White Space.

Beispiel:

*TRANSient:MODE CONTInuous*

Bei der Abfrage von Textparametern erhält man die Kurzform.

Beispiel:

*TRANSient:MODE?*      Antwort z.B.:  
*CONT*

### 7.2.6 Benutzung des Semikolons

Es gibt verschiedene Möglichkeiten, mehrere Befehle in einem einzigen Befehlsstring zu kombinieren.

Ein Semikolon (;) am Ende des ersten Befehls kehrt zum letzten Doppelpunkt (:) zurück, und es kann ein weiterer Befehl der selben Hierarchiestufe eines Befehlssystems hinzugefügt werden.

Beispiel:

Die beiden einzelnen Anweisungen  
*CURRent:IMMEDIATE 15* und  
*CURRent:TRIGgered 10*  
können zu einem String  
zusammengefasst werden:  
*CURRent:IMMEDIATE 15;TRIGgered 10*

Mit dem Semikolon kann man immer nur eine Stufe im Hierarchiesystem zurückgehen.

An den Anfang der Hierarchie (Root level) gelangt man, indem man an das Semikolon direkt einen Doppelpunkt anhängt (:;).

### 7.2.5.5 Text

Text parameter obey the syntactic rules for keywords and provide a short and a long form. The separation from the header is realized by white spaces.

Example:

*TRANSient:MODE CONTInuous*

For the query of text parameters the short form is returned.

Example:

*TRANSient:MODE?*      Result (ex.):  
*CONT*

### 7.2.6 The Semicolon

There are several possibilities to combine commands in one command string.

A semicolon (;) at the end of the first command returns to the last colon (:), and another command of the same hierarchical level of a command system can be appended.

Example:

The two single commands  
*CURRent:IMMEDIATE 15* and  
*CURRent:TRIGgered 10*  
can be combined to one string:  
*CURRent:IMMEDIATE 15;TRIGgered 10*

Using the semicolon only one level of the hierarchical system can be rolled back.

The beginning of the hierarchy (root level) is reached by appending a colon to the semicolon (:;).

Beispiel:

```
CURR:LEV:IMM 15;TRIG 10::INP ON
```

Hat der erste Befehl nur eine Hierarchiestufe, kann man den Doppelpunkt nach dem Semikolon weglassen, da man sich nach dem Semikolon ohnehin wieder im Root level befindet.

Beispiel:

```
CURR 15::INP ON ergibt dasselbe wie  
CURR 15;INP ON
```

Jedoch bei

```
MODE:RES::INP ON
```

muss die Folge :: angegeben werden.

Wenn das Ende einer Zeichenkette erlangt ist, wird automatisch wieder zum Root level gewechselt.

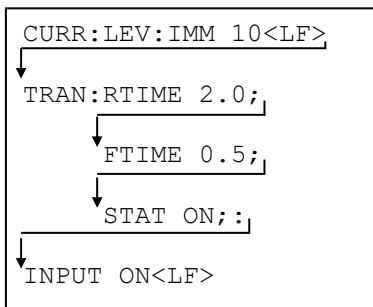
Das Stringende wird erkannt bei einem der folgenden Fälle:

GPIB-Betrieb (s. Kap. 1.2):

- Zeichen <LF> (10dez.)
- EOI

RS232-Betrieb:

- Zeichen <LF> (10dez.)

Einige Beispiele:Example:

```
CURR:LEV:IMM 15;TRIG 10::INP ON
```

If the first command has got only one hierarchical level, the colon behind the semicolon can be omitted, because one semicolon switches back to the root level in such a case.

Example:

```
CURR 15::INP ON has the same result as  
CURR 15;INP ON
```

But for

```
MODE:RES::INP ON
```

the characters :: must be specified.

When the end of a character string is reached, an automatic change to the root level happens.

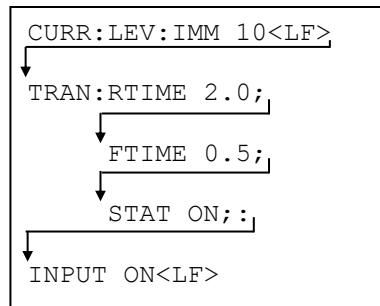
The string end is recognized in one of the following cases:

Operating Mode GPIB (see 1.2):

- Character <LF> (10dec.)
- EOI

Operating Mode RS232:

- Character <LF> (10dec.)

Some Examples:



### 7.2.7 Abfragebefehle (Queries)

Zu den meisten Befehlen gibt es einen zugehörigen Abfragebefehl, der die momentane Einstellung ermittelt. Dazu wird dem Header ein Fragezeichen (?) angehängt.

**Beispiel:** Ermittlung des eingestellten Laststrom-Sollwertes:

`CURRent?`           Antwort z.B.  
                          `+1.000000E+01`

Die vom Gerät gesendete Zahl erscheint im Exponentialformat mit Vorzeichen, einer Vorkomma-, sechs Nachkommastellen, Exponent, Vorzeichen, zwei Exponentstellen.

Das Gerät sendet grundsätzlich keine Einheiten nach Zahlenwerten.

Zur Ermittlung des minimal und maximal möglichen Zahlenwertes wird dem Fragezeichen ein White Space und MIN bzw. MAX angehängt. Als Antwort erhält man den Zahlenwert ohne Einheit.

**Beispiel:** Ermittlung des Maximalstromes  
`CURRent? MAX`    Antw. bei ZS2606  
                          im größten Bereich:  
                          `+3.000000E+02`

Innerhalb eines Befehlsstrings darf immer nur maximal ein Abfragebefehl enthalten sein.

Die Antwort auf diesen Abfragebefehl muss erst ausgelesen werden, bevor der nächste Befehl an das Gerät geschickt wird.

### 7.2.8 Auslesen von Daten



Daten müssen nach einem Abfragebefehl unbedingt komplett ausgelesen werden bevor der nächste Abfragebefehl gesendet wird.

### 7.2.7 Queries

For most commands there is a corresponding query, that determines the actual setting. For the query a question mark (?) is appended to the header.

**Example:** Determine the actual set point for the load current:

`CURR?`                   Result (example)  
                          `+1.000000E+01`

The numeric value that is sent from the device is presented in the exponential format with sign, one digit before the comma, as default six digits after the comma, exponent, sign, two exponent digits.

The device never sends units appended to the numeric values.

To determine the minimum and maximum numeric value the question mark is followed by a white space and MIN or MAX. The result is a numeric value without unit.

**Example:** Determine the maximum current

`CURRent? MAX`        Result for ZS2606 in  
                          max. range:  
                          `+3.000000E+02`

A command string may only include one query. The result for this query must be read before the next query can be sent to the device.

### 7.2.8 Reading Data



Requested data have to be read completely before the next request for data is sent.

## 7.3 SCPI Befehlsübersicht

### 7.3 SCPI Command Overview

#### 7.3.1 Common Commands

#### 7.3.1 Common Commands

Cmd	Parameter	Description	Duration
*CLS		Clear Status	
*ESE	<NRf>	Set Bits in Standard Event Status Enable Register	
*ESE?		Read Standard Event Status Enable Register	
*ESR?		Read Standard Event Status Register	
*IDN?		Identify Device. Response: "Manufacturer,Model,Serial No,Firmware Revision"	40ms
*OPC		Operation Complete Event Bit Command	
*OPC?		Operation Complete Query	
*RST		Device Reset	100ms
*SRE	<NRf>	Set Bits in Service Request Enable Register	
*SRE?		Read Service Request Enable Register	
*STB?		Read Status Byte	
*TRG		Trigger Command	
*TST?		Selftest Query	
*WAI		Wait until all commands have been executed	

### 7.3.2 Gerätespezifische Befehle der Serie ZS

### 7.3.2 Device Dependent Commands for ZS Series

Command	Parameter	Unit <sup>1)</sup>	Description	Duration <sup>2)</sup>
CHANnel INSTrument	<Nrf>[:<Nrf>]		Select sub address or block	
CURRent [:LEVe] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE  :MODE? :PROTection [:LEVe] [:HIGH] [:HIGH]? :TRIPped? :RANGe :AUTO :RANGe?	<num> [MIN MAX] <num> [MIN MAX] FIXed LIST        <num> [MIN MAX]  <num> <Boolean> [MIN MAX]	[A MA]  [A MA]       [A MA]	Set load current Query nominal current Set triggered current Query triggered current Start list or set triggered static current at trigger Query current mode    Set current protection Query current protection Query curr. prot. state Set fixed current range Autorange on off Query current range	15ms 15ms 15ms 15ms         15ms 15ms 20ms 20ms 20ms 15ms
DATA TRACe :POINts?  :REMOve?	  <Nrf>		Query number of measured data sets Read and remove defined number of data sets	
DElay	<Nrf>		Set delay between two commands in program mode (see subsystem PROGram)	
GTL			Set manual operation mode	
INPUt OUTPUt INPUt OUTPUt?	<Boolean>		Load input on off Query load input state	15ms
LIST :CURRent [:LEVe] :RTIME :DWELl :STRamp  :STDWell  :POWEr [:LEVe] :RTIME :DWELl :STRamp  :STDWell  :RESistance	<Nrf>{,<Nrf>} <Nrf>{,<Nrf>} <Nrf>{,<Nrf>} <Nrf>{,<Nrf>}  <Nrf>{,<Nrf>}  <Nrf>{,<Nrf>} <Nrf>{,<Nrf>} <Nrf>{,<Nrf>}  <Nrf>{,<Nrf>} <Nrf>{,<Nrf>} <Nrf>{,<Nrf>}  <Nrf>{,<Nrf>}  <Nrf>{,<Nrf>}		Define current list Define current ramp list Define current dwell list Define current ramp measurement list Define current dwell measurement list  Define power list Define power ramp list Define power dwell list Define power ramp measurement list Define power dwell measurement list	

[:LEVe] :RTIME :DWELL :STRamp  :STDWell  :VOLTage [:LEVe] :RTIME :DWELL :STRamp  :STDWell  :COUNT :STATE :STATE?	<NRf>{,<NRf>} <NRf>{,<NRf>} <NRf>{,<NRf>} <NRf>{,<NRf>}  <NRf>{,<NRf>}  <NRf>{,<NRf>} <NRf>{,<NRf>} <NRf>{,<NRf>} <NRf>{,<NRf>}  <NRf> INfInity <Boolean>		Define resistance list Define resistance ramp list Define resistance dwell list Define resistance ramp measurement list Define resistance dwell measurement list  Define voltage list Define voltage ramp list Define voltage dwell list Define voltage ramp measurement list Define voltage dwell measurement list  Define number of loops Start/stop list execution Query list execution state	
MEASure :CHARge [:DC]? :ENERgy [:DC]? :CURRent [:DC]? :EXTernal [:DC]? :MPP [:DC]? :POWer [:DC]? :VOLTage [:DC]? :RESistance [:DC]?			Measure charge  Measure energy  Measure current  Measure external voltage  Measure maximum power  Measure power  Measure voltage  Measure resistance	  20ms  20ms   20ms  20ms  20ms
MODE FUNCTION :CURRent [:DC] :RESistance [:DC] :VOLTage [:DC] :POWer [:DC] :MPP MODE FUNCTION?			Constant current mode  Constant resistance mode  Constant voltage mode  Constant power mode MPP Tracking mode Query mode	20ms  20ms  20ms  20ms 20ms 15ms
PCYCLE :CURRent  :RESistance  :POWer  :VOLTage	<row>,<NRf>  <row>,<NRf>  <row>,<NRf>  <row>,<NRf>	[A MA]  [OHM KOHM  MOHM] [MW W KW]  [MW W KW]	Programmable waveform Set table row (Par1) with current (Par2) Set table row (Par1) with resistance (Par2) Set table row (Par1) with power (Par2)	20ms  20ms  20ms  20ms

:TIME	<row>, <Nrf>	[S MS]	Set table row (Par1) with voltage (Par2) Set table row (Par1) with time (Par2)	20ms
:MODE	CONTInuous  PULSe, <Nrf>		Set continuous or pulse mode	20ms
:MODE?			Query waveform mode	15ms
:TRIGgered	<Boolean>		Enable/disable waveform triggering	40ms
:STATe	<Boolean>		Start/Stop programmed waveform	15ms
:STATe?			Query waveform state	
POWer [:LEVe] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE :MODE? :RANGe :RANGe?	<num> [MIN MAX] <Nrf>  MAX [MIN MAX] FIXed LIST  <num> [MIN MAX]	[MW W KW]  [MW W KW]  [MW W KW]	Set nominal power Query nominal power Set triggered power Query triggered power At trigger start list or set triggered static power Query power mode Set power range Query power range	15ms 20ms 20ms 20ms  20ms 15ms
PROGram [:SELected] :BEGin :DELete [:SELected] :ALL :END :NAME :STATe :STATe?	0 1 2 RUN ON 1		Start programming mode  Delete active program Delete all programs End programming mode Select active program Start active program Query running state	
RESistance [:LEVe] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE :MODE? :RANGe :AUTO :RANGe?	<Nrf>  MAX  [MIN MAX] <Nrf>  MAX  [MIN MAX] FIXed LIST  <num>  <Boolean> [MIN MAX]	[OHM KOHM  MOHM]  [OHM KOHM  MOHM]  [OHM KOHM  MOHM]	Set nominal resistance  Query nominal resistance Set triggered resistance  Query triggered resistance At trigger start list or set triggered static resistance Query resistance mode Set resistance range Autoranging on off Query resistance range	15ms 20ms 20ms 20ms  20ms 20ms 20ms
SETup :ADC :ADC?	SLOW FAST  SLOW FAST		Activate slow or fast ADC (def. slow) Query active ADC	15ms 15ms

SETup?			Query device parameters	50ms
SFUNction :BATTery :ENABLE  :ENABLE? :STATe? :TEVoltage :TEVoltage? :EXPNontial  :ENABLE :ENABLE? :STATe?	<Boolean>    <NRf>  <NRf>, <NRf>, <NRf>, <NRf>, <NRf> <Boolean>		Enable/disable battery test function Query enable state Query active state Define test-end voltage Query test-end voltage  Define parameters for exponential function Enable/disable e-function Query enable state Query active state	
STATus :OPERation [:EVENT]? :CONDition? :ENABLE :ENABLE? :QUEStionable [:EVENT]? :CONDition? :ENABLE :ENABLE? :PRESet	<NRf>         <NRf>		Query Op. Event Reg. Query Op. Condition Reg. Set Op. Enable Bits Query Op. Enable Register  Query Ques. Event Reg. Query Ques. Cond. Reg. Set Ques. Enable Bits Query Ques. Enable Reg. Reset Status	20ms 20ms 20ms 20ms  20ms 20ms 20ms 20ms 15ms

SYSTem :COMMunicate :SERial :CONTRol :RTS [:RECeive] :PACE  :CONTRol  :CONTRol? :ERRor? :FAN :LANGUage :LANGUage? :PARameter :PARameter <NRf> ? :PROTectioN [:LEVe] [:LEVe]? :STATe :TRIPped? :SPEEd  :SPEEd? :STRing :VERSion?	IBFull RFR  XON  EXTeRnal   INTeRnal  AUTO FULL SCPI  <NRf> , <NRf>  <NRf>  <Boolean>  SLOW FAST  MEDium  250,"string"	[S MS]	Enable RTS/CTS handshake  Enable XON/XOFF flow control external/internal control  Query control source Query last error message Set fan speed auto or full Set programming syntax Query programming syntax Set device parameter Query device parameter  Set SW watchdog time Query SW watchdog time Enable/disable SW WD Query SW watchdog state Set control rate  Query control rate Set format string Query SCPI-Version	20ms  15ms 15ms  20ms  15ms
TRIGger [:SEQuence] :SOUrce  :SOUrce? :TIMer :TIMer?	BUS EXTeRnal  IMMEDIATE TIMer  <NRf>		Set trigger source  Query trigger source Set trigger timer value Query trigger timer value	
VOLTage :CRANge :CRANge?  [:LEVe] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE  :MODE?  :PROTectioN [:LEVe] [:LOW] [:LOW]? :TRIPped? :RANge :AUTO :RANge?	<num> [MIN MAX]  <num> [MIN MAX] <num> [MIN MAX] FIXed LIST  <num> [MIN MAX]  <num> <Boolean> [MIN MAX]	[A MA]  [V MV]  [V MV]	Current range in V mode Query current range in V mode (reserved)  Set nominal voltage Query nominal voltage Set triggered voltage Query triggered voltage Start list or set triggered static voltage at trigger Query voltage mode	15ms 20ms 20ms 20ms  15ms 15ms 20ms 20ms 20ms 20ms

Bemerkung <sup>(1)</sup>:

Um zwischen den Multiplizierern "Milli" ( $10^{-3}$ ) und "Mega" ( $10^6$ ) zu unterscheiden, ist normalerweise die Abkürzung für Milli 'M' und für Mega 'MA'.

Eine Ausnahme gibt es bei der Widerstandseinheit. Hier gibt es keine Einheit für 'Milliohm'. Die Einheit 'MOHM' bedeutet immer MegOhm!

Bemerkung <sup>(2)</sup>:

Die in der Spalte "Duration" angeführte Zeit gibt die ungefähre Dauer zwischen dem Empfang des ersten Bytes auf dem GPIB Bus und der Ausführung des Befehls im Analoginterface an. Die Zeit zum Senden eines evtl. Antwortstrings ist hier nicht enthalten, da diese bei GPIB vom Controller abhängt und bei serieller Datenübertragung von den RS232-Einstellungen wie Baudrate etc.

Comment <sup>(1)</sup>:

To distinguish between the multipliers "Milli" ( $10^{-3}$ ) and "Mega" ( $10^6$ ), the abbreviations "M" for Milli and "MA" for Mega are used.

One exception is the resistance unit. There is no unit for "Milliohm". The unit "MOHM" always means MegaOhm!

Comment <sup>(2)</sup>:

The time represented in the column "Duration" defines the approximate time between the first received byte on the GPIB bus and the execution of the command in the analog interface. The time for sending a response is not included in this time since it depends on the speed of the GPIB controller and, respectively, at serial communication on RS232 settings like baud rate etc.



## 7.4 SCPI Befehlsbeschreibung

### 7.4.1 Common Commands

#### \*CLS

**Clear Status** löscht den Inhalt folgender Statusregister:

Questionable Status Event, Operation Status Event, Standard Event, Statusbyteregister.

Alle anderen Statusregister (Condition, Enable) bleiben unverändert.

Der Ausgabepuffer wird gelöscht.

#### \*ESE 0...255

setzt das Standard **Event Status Enable** Register auf den angegebenen Wert (Belegung siehe Kap. 5.4.2.13).

#### \*ESE?

liest den Inhalt des Standard **Event Status Enable** Registers als dezimale Ganzzahl zurück (Belegung siehe Kap. 5.4.2.13).

#### \*ESR?

liest den Inhalt des Standard **Event Status Registers** als dezimale Ganzzahl zurück und löscht dieses.

#### \*IDN?

**Identification Query** fragt die Geräteerkennung ab und liest String mit folgender Zusammensetzung zurück: Hersteller, Gerätebezeichnung, Seriennummer, Firmwareversion (evtl. mit Hardware-Info).

Ist die Seriennummer nicht verfügbar, wird an deren Stelle eine 0 geliefert.

Die Antwort einer elektronischen Last der Serie ZS könnte beispielsweise lauten:

*HOECHERL&HACKL,ZS2606,2525B-0804,ZS\_AI\_02.21-E-04AI04-R00*

## 7.4 SCPI Commands – Detailed Description

### 7.4.1 Common Commands

#### \*CLS

**Clear Status** deletes the contents of the following status registers:

Questionable Status Event, Operation Status Event, Standard Event, Statusbyte Register.

All other status registers (Condition, Enable) remain unchanged.

The output buffer is deleted.

#### \*ESE 0...255

Event Status Enable sets the standard register **Event Status Enable** Register to the specified value (see chapter 11).

#### \*ESE?

Reads the contents of the standard register **Event Status Enable** back as decimal integer (see chapter 11).

#### \*ESR?

Reads the contents of the standard register **Event Status Registers** back as decimal integer and deletes it.

#### \*IDN?

**Identification Query** queries the device identification and reads a string with the following contents back: Manufacturer, Device Name, Serial Number, Firmware Version (possibly with hardware information).

If no serial number is provided, 0 is used.

The response of an electronic ZS load could be:

*HOECHERL&HACKL,ZS2606,2525B-0804,ZS\_AI\_02.21-04AI04-R00*

Der letzte Eintrag ist möglicherweise nochmals mit Bindestrichen unterteilt in folgende Subinformationen:

ZS\_AI\_02.21 = Firmwareversion Analoginterface (02.21)

E = EEPROM Nichtflüchtiger Speicher (F wäre FRAM)

-04 = Firmwareversion Subcontroller (04)

-AI04 = Hardwareversion Analoginterface (04)

-R00 = Hardwareversion Regelungsplatine (00)

Der ID-String des RS232-Interfaces kann durch Voranstellen der reservierten Unteradresse 255 ausgelesen werden:

CHAN 255;\*IDN?

Nach dem Auslesen des ID-Strings von CHANnel 255 muss das Gerät entweder aus- und eingeschaltet oder einmal der Befehl CHAN x (x=Unteradresse des Gerätes) gesendet werden.

#### \*OPC

**Operation Complete** setzt das Bit 0 im Event Status Register, wenn alle vorausgegangenen Befehle abgearbeitet sind.

(Bemerkung: Bit 0 im Event Status Register ist bei den elektronischen Lasten immer high, da die Befehle nicht im Overlapped-Modus ausgeführt werden, sondern immer sequenziell.)

#### \*OPC?

**Operation Complete Query** schreibt die Nachricht '1' in den Ausgabepuffer, sobald alle vorangegangenen Befehle ausgeführt sind.

(Bemerkung: Die Befehlsabarbeitung bei den elektronischen Lasten von H&H erfolgt sequenziell. Daher erhält man als Antwort immer eine '1'.)

The last entry is possibly subdivided into the following information:

ZS\_AI\_02.21 = Firmware version analog interface (02.21)

E = EEPROM waveform memory (F means FRAM)

-04 = Firmware version subcontroller (04)

-AI04 = Hardware version analog interface (04)

-R00 = Hardware version controller board (00)

The firmware version of the RS232 interface can be read by preceding the reserved sub address 255:

CHAN 255;\*IDN?

After the ID string has been read you must either switch the device off and on or once send the command CHAN 1.

#### \*OPC

**Operation Complete** sets bit 0 in the Event Status Register, if all commands ahead have been executed.

(Comment: Bit 0 in the Event Status Register is always high for electronic loads, because the commands aren't executed in the overlapped mode, but always sequential.)

#### \*OPC?

**Operation Complete Query** writes the message '1' into the output buffer, if all commands ahead have been executed.

(Comment: The command execution of the electronic loads from H&H happens sequentially. The response is always '1'.)

**\*RST**

**Reset** versetzt das Gerät in die Grundeinstellung.

Bei den elektronischen Lasten der Serie ZS sind dies:

**\*RST**

**Reset** resets the device to its standard settings.

For the ZS series electronic loads these are:

```

CURRent:RANGe:AUTO ON
CURRent 0
CURRent:TRIGgered 0
CURRent:MODE FIXEd
MODE:CURRent
INPut OFF
LIST:CURRent 0,0,..
LIST:CURRent:RTIMe 0,0,...
LIST:CURRent:DWELl 0,0,...
LIST:CURRent:STRamp 0,0,...
LIST:CURRent:STDWELL 0,0,... // equal for P/R/V lists
LIST:COUNT INF
LIST:STATe OFF
PCYClE:CURRent <row>,0
PCYClE:TIME <row>,0 // <row>= {0...2000}
PCYClE:MODE CONTInuous
PCYClE:STATe OFF
POWer:RANGe MAX
POWer 0
POWer:MODE FIXEd
RESistance:RANGe:AUTO ON
RESistance MAX
RESistance:TRIGgered MAX
RESistance:MODE FIXEd
SFUNction:BATTeRy ENABle OFF
SFUNction:BATTeRy:TEVOLTage 0
SFUNction:EXPOntial 0,0,0,0,0
SFUNction:EXPOntial:ENABle OFF
SYSTem:PROTection:STATe OFF
SYSTem:PROTection[:LEVel] 60s
TRIGger:SOURce IMM
TRIGger:TIMer 0.0002
VOLTage:RANGe MAX
VOLTage MAX
VOLTage:TRIGgered MAX
VOLTage:MODE FIXEd

```

**\*SRE 0...255**

setzt das **Service Request Enable** Register auf den angegebenen Wert.

**\*SRE?**

liest den Inhalt des **Service Request Enable** Registers als dezimale Ganzzahl zurück.

**\*STB?**

liest den Inhalt des **Statusbytes** als dezimale Ganzzahl zurück.

**\*TRG**

**Trigger** löst Aktionen, die auf ein Triggerereignis warten, aus, wenn TRIGger:SOURce auf BUS gesetzt ist.

**\*TST?**

**Selbsttest** Abfrage löst Selbsttest des Gerätes aus und gibt dezimale Ganzzahl zurück. Rückgabewert ungleich 0 bedeutet Fehler.

**\*WAI**

**Wait to Continue** erlaubt die Abarbeitung der nachfolgenden Befehle erst, nachdem alle vorhergehenden Befehle durchgeführt worden sind.

(Bemerkung: Die Befehlsabarbeitung bei den elektronischen Lasten von H&H erfolgt sequenziell. Daher ist dieser Befehl nur zum Zwecke der SCPI-Konformität implementiert.)

**\*SRE 0...255**

Sets the register **Service Request Enable** to the specified value.

**\*SRE?**

Reads the contents of the register **Service Request Enable** back as a decimal integer.

**\*STB?**

Reads the contents of the **Status Byte** back as decimal integer.

**\*TRG**

**Trigger** triggers actions waiting for a trigger event. TRIGger:SOURce must be set to BUS.

**\*TST?**

**Selftest Query** triggers the selftest of the device and returns a decimal integer. A return value of unequal to 0 identifies an error.

**\*WAI**

**Wait to Continue** allows the execution of following commands, after all commands ahead have been executed. (Comment: The command execution in the electronic loads from H&H happens sequential. This command has been implemented for SCPI conformity.)

## 7.4.2 Gerätespezifische Befehle

### 7.4.2.1 Schnelleinstieg

Die wichtigsten Befehlssysteme bei der Programmierung der elektronischen Lasten der ZS Serie sind die Systeme

- CURRent
- INPut
- MEASure
- MODE
- POWer
- RESistance
- VOLTage

Die Defaulteinstellungen bei Reset des Gerätes sind `MODE:CURRent;:INPut OFF;:CURRent 0;:POWer MIN;:RESistance MAX;:VOLTage MAX.`

Um eine bestimmte Belastung (z.B. 12.5A) im Konstantstrombetrieb einzustellen, geben Sie den Laststrom an und schalten den Geräteeingang ein:  
`CURR 12.5;:INP ON`

Um eine bestimmte Belastung im Konstantwiderstandsbetrieb (z.B.  $1\Omega$ ) einzustellen, geben Sie den gewünschten Widerstandswert an, wechseln in den Widerstandsbetrieb (Annahme: Eingang ist bereits ein):  
`RES 1;:MODE:RES`

Wenn Sie jetzt mit `MODE:CURR` wieder in Konstantstrombetrieb wechseln, wird der letzte gültige Stromwert eingestellt, in unserem Beispiel also 12.5A.

Die Messwerte für Strom, Spannung, Widerstand und Leistung können mit den Befehlen

## 7.4.2 Device Dependent Commands

### 7.4.2.1 First Steps

The main command systems for the programming of the electronic loads of series ZS are

- CURRent
- INPut
- MEASure
- MODE
- POWer
- RESistance
- VOLTage

The default settings after a Reset of the device are `MODE:CURRent;:INPut OFF;:CURRent 0;:POWer MIN;:RESistance MAX;:VOLTage MAX.`

To set a particular load (for example 12.5 A) in the operating mode constant current, specify the load current and activate the device input:  
`CURR 12.5;:INP ON`

To set a particular load in the operating mode resistance (for example  $1\Omega$ ), specify the desired resistance value and change into the operating mode resistance (assumption: the input is activated):  
`RES 1;:MODE:RES`

If you change back to the operating mode constant current using `MODE:CURR` the last valid current value is set, in our example 12.5A.

The measuring values for current, voltage and power are queried using the following commands:

MEAS:CURR?

MEAS:VOLT?

MEAS:RES?

MEAS:POW?

abgefragt werden.

Der angeforderte Messwert wird vom Gerät im Exponentialformat bereitgestellt:

SD.DDDDDDESDD

Vorzeichen, 1 Vorkommastelle, Dezimalpunkt, 6 Nachkommastellen, 'E', Vorzeichen, 2 Stellen für Exponent

Hinweis:

Im RS232-Betrieb muss zwischen Senden eines Abfragebefehls und dem Lesen des Rückgabewertes eine Wartezeit von mindestens 100ms eingefügt werden.

Im folgenden werden die einzelnen Befehlssysteme in alphabetischer Reihenfolge beschrieben.

MEAS:CURR?

MEAS:VOLT?

MEAS:RES?

MEAS:POW?

The device provides the required measuring value in exponential format:

SD.DDDDDDESDD

Sign, 1 digit before the comma, decimal separator, 6 digits after the comma, 'E', sign, 2 digits for the exponent.

Note:

In the operating mode RS232 a wait of about 100ms has to be inserted between sending a query and reading of the return value.

The following sections describe the single command systems in alphabetic order.

### 7.4.2.2 Subsystem CHANnel|INSTrument

Befehl	Parameter	Einheit	Bemerkung
CHANnel INSTrument	<NRF>[:<NRF>]		Unter-/Blockadresse

### 7.4.2.2 Subsystem CHANnel|INSTrument

Command	Parameter	Unit	Comment
CHANnel INSTrument	<NRF>[:<NRF>]		Sub/Group Address

Das Subsystem CHANnel ist vorhanden, um beim Betrieb von mehreren Geräten mit einer gemeinsamen GPIB- Adresse bzw. über die RS232- Schnittstelle die einzelnen Geräte voneinander unterscheiden zu können (s. Kap. 6). Der Parameter dieses Befehls selektiert die sog. Unteradresse eines Gerätes.

Wenn keine Unteradresse programmiert worden ist, selektiert das GPIB- bzw. RS232-Interface defaultmäßig Kanal 1.

Wurden ein oder mehrere Geräte als System bestellt (d.h. mind. ein Gerät hat einen Systembus-Eingang), so werden werksseitig - falls nicht anders geordert - beginnend bei 1 aufsteigende Unteradressen an den rückseitigen Sub Address DIP-Schaltern eingestellt.

Wird ein Gerät als Einzelgerät eingesetzt (im Normalfall), ist das Subsystem CHANnel nicht von Bedeutung.

Anstelle des Schlüsselwortes CHANnel darf auch das Wort INSTrument stehen.

#### **CHANnel <0...192[:1...192],255>**

Adressiert ein Gerät am Systembus. CHANnel 255 ist reserviert für das Dateninterface (s. Kap. 7.4.1: \*IDN?).

The sub system CHANnel is provided to distinguish between the devices when operating several devices with common GPIB address or via the RS232 interface (see chapter 6).

If no sub address is programmed before a SCPI command is given the device's SCPI interpreter selects sub address 1.

If one or more devices have been ordered as system (i.e. at least one device has got a system bus input), the sub addresses are assigned beginning with 1 (if not specified otherwise) at the rear sub address DIP switches.

If a device is used as single device (standard) the sub system CHANnel is not relevant.

Instead of the keyword CHANnel the keyword INSTrument may be used.

#### **CHANnel <0...192[:1...192],255>**

Addresses a device via the H&H system bus.

CHANnel 255 is reserved for the data interface (see chapter 7.4.1: \*IDN?).

Folgt dem Header nur ein Zahlenparameter, muss die im Gerät gespeicherte Unteradresse genau mit dem Parameter übereinstimmen, damit die nachfolgenden Befehle ausgeführt werden.

Beispiel:

CHANnel 5

adressiert Gerät mit Unteradresse 5

Der Parameter 0 adressiert alle am Systembus angeschlossenen Geräte.

Beispiel:

CHAN 0;\*RST Systemreset

**Blockadressierung:**

Folgt dem ersten Parameter ein Doppelpunkt mit einem weiteren Zahlenparameter, so werden damit alle jene Geräte adressiert, deren Unteradressen größer/gleich dem ersten Parameter und kleiner/gleich dem zweiten Parameter sind (Blockadressierung).

Beispiel:

CHAN 6:10::INP ON

Geräte Nr. 6, 7, 8, 9 und 10 schalten den Lasteingang ein

Hinweis:

Bei Block- und Systemadressierung sind Mess- und Abfragebefehle unzulässig. Daher werden von keinem Gerät Daten zurückgeschickt, wenn es mit System- oder Blockadressierung adressiert worden ist.

If only one numeric parameter follows the header, the sub address at the device's DIPs has to match exactly with the parameter, so that following commands can be executed.

Example:

CHANnel 5

addresses a device with the sub address 5.

The parameter 0 addresses all devices connected to the system bus.

Example:

CHAN 0;\*RST System Reset

**Block addressing:**

If the first parameter is followed by a colon and a further numeric parameter, all devices are addressed, where the subaddress is greater/equal the first parameter and smaller/equal the second parameter.

Example:

CHAN 6:10::INP ON

The devices 6, 7, 8, 9 and 10 activate the load input

Note:

For block and system addressing measuring and query commands are not allowed. No device sends data back if it has been addressed using system or block addressing.



## 7.4.2.3 Subsystem CURRent

Befehl	Parameter	Einheit	Bemerkung
CURRent [:LEVel] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE  :MODE? :PROTection [:LEVel] [:HIGH] [:HIGH]? :TRIPped? :RANGe :AUTO :RANGe?	<num> [MIN   MAX] <num> [MIN   MAX] FIXed   LIST      <num> [MIN   MAX]  <num> <Boolean> [MIN   MAX]	[A   MA]  [A   MA]       [A   MA]	Laststrom einstellen Laststrom-Sollwert abfr. getriggertter Laststrom Laststrom-Triggerwert abfr. Bei Trigger Liste starten oder statischen Triggerstrom einst. Current Mode abfragen    Strombegrenzung einstellen Strombegr.wert abfragen Strombegr.zustand abfr. fester Strombereich Autorange ein   aus Strombereich abfragen

## 7.4.2.3 Subsystem CURRent

Command	Parameter	Unit	Comment
CURRent [:LEVel] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE  :MODE? :PROTection [:LEVel] [:HIGH] [:HIGH]? :TRIPped? :RANGe :AUTO :RANGe?	<num> [MIN   MAX] <num> [MIN   MAX] FIXed   LIST      <num> [MIN   MAX]  <num> <Boolean> [MIN   MAX]	[A   MA]  [A   MA]       [A   MA]	Set load current Query nominal current Set triggered current Query triggered current Start list or set triggered static current at trigger Query current mode    Set current protection Query current protection Query curr. prof. state Set fixed current range Autorange on   off Query current range

Das Befehlssystem CURRent dient zur Einstellung und Abfrage des Laststrom-Sollwertes sowie zur Einstellung des Strombegrenzungs-Limits.

The command system CURRent is used for setting and querying the load current set point and the current protection limit.

**CURRent[:LEVel][:IMMediate]****<num>**

Stellt einen neuen Laststrom ein. Befindet sich das Gerät im Strombetrieb, wird der neue Wert sofort eingestellt, sofern dieser im gültigen Wertebereich liegt.

Der Einstellbereich ist den Technischen Daten des jeweiligen Modells zu entnehmen.

Bei Überschreitung des Wertebereichs leuchtet Err2 an der Last und es wird ein "Data out of range"-Error ausgelöst, der mit *SYSTEM:ERRor?* ausgelesen werden kann. In diesem Fall wird der maximal mögliche Wert eingestellt und der programmierte Wert als Sollwert gespeichert. Wird später in einen ausreichend hohen Strombereich gewechselt, so stellt die Last den Sollwert ein.

Befindet sich das Gerät nicht im Konstantstrombetrieb, wird der neue Einstellwert gespeichert und beim Wechsel in den Strombetrieb (mit *MODE:CURRent*) eingestellt.

Als Parameter sind alle Zahlenwerte innerhalb des momentanen Strombereiches erlaubt.

Außerdem sind die speziellen Zahlenparameter MIN und MAX erlaubt. Im Autoranging-Betrieb stellt der Parameter MAX den Endwert im höchsten Strombereich ein, ansonsten den Endwert im momentanen Bereich.

Beispiele:*CURR:LEV 15.23**CURRent:IMM 0**CURR MAX*

Als Dezimaltrennzeichen erwartet das Gerät einen Punkt (.), kein Komma!

**CURRent[:LEVel][:IMMediate]****<num>**

Sets a new load current. If the device is in the operating mode current, the new value will be set immediately if it is within the valid range.

The setting range is specified in the technical data of the particular device type.

When exceeding the current range the Err2 lights up and a "Data out of range" Error is triggered, that can be read using *SYSTEM:ERRor?*. In this case the maximum possible value is set and the programmed value is kept as nominal value. If a suited current range is set at a later time the load sets the saved nominal value.

If the device is not in the operating mode constant current, the new setting value is saved and set when changing into the operating mode current (using *MODE:CURRent*).

As parameters all numeric values within the current range of the particular model are allowed.

The specific numeric parameters MIN and MAX are allowed. In autoranging mode the parameter MAX sets the end value in the highest current range, otherwise the end value in the active range.

Examples:*CURR:LEV 15.23**CURRent:IMM 0**CURR MAX*

As decimal separator the device expects a point (.), no comma!

**CURRent[:LEVel][:IMMediate]?**

Abfrage des momentan eingestellten Sollwertes im Strombetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Einstellwert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Als Rückgabewert erhält man den im momentanen Strombereich minimal bzw. maximal möglichen Einstellwert.

Beispiele:

CURR? (Antw. z.B.: +1.850000E+01)

CURR? MAX

(Antwort von ZS2606: +3.000000E+02)

**CURRent[:LEVel]:TRIGgered****<num>**

Stellt einen neuen Wert für den getriggerten Laststrom ein.

S. auch CURRent:RANGe und Subsystem TRIGger.

Bei Überschreitung des zulässigen Wertebereichs wird ein "Data out of range"-Error ausgelöst, der mit SYSTem:ERRor? ausgelesen werden kann. In diesem Fall wird der maximal mögliche Wert eingestellt und der programmierte Wert als Sollwert gespeichert. Wird später in einen ausreichend hohen Strombereich gewechselt, so stellt die Last den Sollwert ein.

**CURRent[:LEVel][IMMediate]?**

Queries the actual set point in the operating mode current.

A numeric value in exponential format is returned:

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible setting value is queried appending a white space and the parameter MIN or MAX to the question mark.

As response the load sends the minimum or maximum possible setting value in the respective setting range.

Examples:

CURR? (Response for example:  
+1.850000E+01)

CURR? MAX

(Resp. from ZS2606: +3.000000E+02)

**CURRent[:LEVel]:TRIGgered****<num>**

Sets a new value for the triggered load current.

See also CURRent:RANGe and Subsystem TRIGger.

When exceeding the allowed value range the error "Data out of range" is triggered, that can be read with SYSTem:ERRor? In this case the maximum possible value is set and the programmed value is kept as nominal value. If a suited current range is set at a later time the load sets the saved nominal value.

Das Triggerereignis wird mit dem Befehl **TRIGger:SOURce** definiert.

Tritt das Triggerereignis ein und Konstantstrombetrieb ist eingestellt, stellt das Gerät den vorher programmierten Trigger-Laststrom ein, wenn nicht eine dynamische Betriebsart triggerfähig gemacht wurde (s. SubSystem TRIGger).

Als Parameter sind alle Zahlenwerte innerhalb des momentanen Strombereiches erlaubt.

Außerdem sind die speziellen Zahlenparameter MIN und MAX erlaubt.

Im Autoranging-Betrieb stellt der Parameter MAX den Endwert im höchsten Strombereich ein, ansonsten den Endwert im momentanen Bereich.

Beispiele: bei Trigger 0A einstellen

**CURR:TRIG 0.0**

**CURRent:LEVEL:TRIGGERED 0**

**CURR:TRIG MIN**

Als Dezimaltrennzeichen erwartet das Gerät einen Punkt (.), kein Komma!

### **CURRent[:LEVel][TRIGgered]?**

Abfrage des triggerbaren Sollwertes im Strombetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

**SD.DDDDDDESDD** S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert im momentanen Strombereich wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

The trigger event is defined using the command **TRIGger:SOURce**.

If the trigger event takes place and the operating mode constant current is set, the device sets the programmed trigger load current unless there is a dynamic mode waiting for a trigger (see subsystem TRIGger).

Allowed parameters are all numeric values within the current range of the particular device type.

The special numeric values MIN and MAX are allowed.

In autoranging mode the parameter MAX sets the end value in the highest current range, otherwise the end value in the active range.

Examples: Set 0A at trigger

**CURR:TRIG 0.0**

**CURRent:LEVEL:TRIGGERED 0**

**CURR:TRIG MIN**

The device expects a point (.) as decimal separator, no comma!

### **CURRent[:LEVel][TRIGgered]?**

Queries the triggerable set point for operating mode current.

The return value is a numeric value in exponential form:

**SD.DDDDDDESDD** S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible value is queried by appending a question mark and the parameters MIN or MAX.

Beispiele:

**CURR:TRIG?** (Antw. nach \*RST:  
+0.000000E+00)  
**CURR:LEVEL:TRIG? MAX**  
(Antwort von ZS2606: +3.000000E+02)

**CURRent:MODE FIXed | LIST**

Bestimmt, ob bei einem Triggerereignis der statische Triggerstrom (CURRent:TRIGgered) oder eine programmierte Stromliste (LIST:CURRent (Kurvenform)) eingestellt werden soll. Nach dem Einschalten der Last ist CURRent:MODE FIXed gewählt. S. auch Subsysteme LIST und TRIGger.

Beispiel:

**CURR:MODE LIST**

**CURRent:MODE?**

Strom-Trigger-Betriebsart abfragen. Als Rückgabewert wird die Kurzform des entsprechenden Parameters bereitgestellt (FIX, LIST).

Beispiel:

**CURR:MODE?** (Antwort nach  
Einschalten: FIX)

**CURRent:PROTEction[:LEVel]  
[:HIGH] <num>**

Stellt die Strombegrenzung ein. Die Strombegrenzung kann nicht durch einen separaten Befehl aktiviert bzw. deaktiviert werden. Durch die Programmierung eines Strombegrenzungswertes ist die Strombegrenzung auch aktiviert. Soll die programmierte Strombegrenzung deaktiviert werden, muss der Sollwert wieder auf den Maximalwert (default) programmiert werden.

Examples:

**CURR:TRIG?** (Resp. after \*RST:  
+0.000000E+00)  
**CURR:LEVEL:TRIG? MAX**  
(Resp. of ZS2606: +3.000000E+02)

**CURRent:MODE FIXed | LIST**

Determines if the static trigger current (CURRent:TRIGgered) or a programmed current list (LIST:CURRent (waveform)) shall be set when a trigger event occurs. After power-on CURRent:MODE FIXed is set.

Example:

**CURR:MODE LIST**

See also Subsystems LIST and TRIGger.

**CURRent:MODE?**

Query current trigger mode. The return value is the short form of the corresponding parameters (FIX, LIST).

Example:

**CURR:MODE?** (Response after  
power-on: FIX)

**CURRent:PROTEction[:LEVel]  
[:HIGH] <num>**

Sets the current limitation for all operating modes. The current limitation can not be separately activated/deactivated but is always active. If the current limitation shall be "deactivated" the current limitation value shall be set to maximum value (default).

Beispiele:

**CURR:PROT 10** (Strombegrenzung  
10A wird eingestellt)  
**CURR:PROT MAX** (maximale Strombe-  
grenzung wird  
eingestellt)

Hinweis:

Der Parameter für die Strombegrenzung kann nur verändert werden, wenn keine dynamische Betriebsart (TRAN, PCYC) aktiv ist.

Der Parameter MAX (CURR:PROT MAX) stellt einen um ca. 2...5% höheren Wert als den Strombereichsendwert ein. Wird CURR:PROT MAX im Autorangingbetrieb programmiert und das Gerät vollzieht einen automatischen Bereichswechsel, stellt es die Strombegrenzung auch auf den Maximalwert des neuen Bereiches. Ansonsten gelten die gleichen Regeln wie bei Stromsettings.

**CURR:PROT[:LEVEL]  
[:HIGH]?**

Abfrage des momentan eingestellten Strombegrenzungswertes.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

**SD.DDDDDDESDD** S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden. Dann sendet das Gerät im Autorangingbetrieb den Maximalwert im größten Bereich, ansonsten den Maximalwert im momentanen Bereich.

Beispiele:

**CURR:PROT?** (Antw. z.B.:  
+2.650000E+01)  
**CURR:PROT? MAX** (Antwort z.B.:  
+3.150000E+01)

Examples:

**CURR:PROT 10** (current limitation  
10A is set)  
**CURR:PROT MAX** (maximum current  
limitation is set)

Note:

The parameter for the current limitation is only programmable when no dynamic function(TRAN, PCYC) is active.

Parameter MAX (CURR:PROT MAX) sets the current limit to approximately 2...5% above the current range end value. If CURR:PROT MAX is programmed in autoranging mode and the device performs an automatic range switch then the current limitation is set to the maximum of the new range. Additionally, the same rules as with current settings are valid.

**CURR:PROT[:LEVEL]  
[:HIGH]?**

Queries the actual set current limitation value.

The return value is a numeric value in exponential form:

**SD.DDDDDDESDD** S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible value is queried by appending a question mark and the parameters MIN or MAX. If autoranging is active the load sends the maximum value of the highest range, otherwise the maximum value of the active range.

Examples:

**CURR:PROT?** (Resp. (ex.):  
+2.650000E+01)  
**CURR:PROT? MAX** (Return:  
+3.150000E+01)

**CURRent:PROTection:TRIPped?**

Abfrage des Strombegrenzungsstatus.  
Als Rückgabewert wird eine Boolesche Zahl (0 oder 1) zur Verfügung gestellt.

Dabei bedeutet:

0: Strom wird momentan nicht begrenzt.  
1: Strom wird begrenzt, Status OC aktiv.

Beispiel:

**CURR:PROT:TRIP?** Antwort: 0

**CURRent:RANGe <num>**

Wählt einen festen Einstellbereich im Strombetrieb.

Der numerische Parameter muss innerhalb des Strombereichs des betreffenden Gerätes sein (Technische Daten).

Hinweis:

Die meisten Geräte der ZS Serie verfügen zwar nur über einen Strombereich, auf den sich auch alle Genauigkeitsangaben und Messsignale beziehen, zur feineren Einstellung für kleine Werte gibt es jedoch die "Erweiterte Einstellauflösung", was in diesem Programmierhandbuch zumeist mit Einstellbereich bezeichnet wird.

Der numerische Parameter wird in Ampere angegeben und muss innerhalb des Strombereiches des jeweiligen Modells liegen (Technische Daten).

Die ZS Last versucht, den eingestellten Sollwert aufrechtzuerhalten, d.h. wenn ein höherer Strombereich programmiert wird, wird der letzte Laststrom nach dem Bereichswechsel wieder eingestellt. Wird ein kleinerer Bereich programmiert und der zuletzt eingestellte Sollwert ist zu groß für den kleineren Bereich, wird der maximal mögliche Strom eingestellt und ERR2 am Gerät angezeigt. Der Sollwert (CURR?) bleibt unverändert.

**CURRent:PROTection:TRIPped?**

Queries the status of the current limitation. The response is a Boolean number (0 or 1).

There means:

0: Current is not being limited.  
1: Current is being limited, Status OC active.

Example:

**CURR:PROT:TRIP?** Response: 0

**CURRent:RANGe <num>**

Sets a fixed setting range for the operating mode current.

The numeric parameter has to be within the current range of the particular device type (technical data).

Note:

Most ZS series devices have got only one current range which all accuracy and measurement signals correspond to. Nevertheless, for exact setting of small values there is the "extended resolution" which is called operating range or setting range in this manual because of better understanding.

The numeric parameter is in Amps and must be within the current range of the concerning model (technical data).

The ZS load tries to keep the topical nominal value. That means if a higher range is programmed the last setting is kept. If a lower range is programmed and the last setting is too high for this range then the maximum possible value is set and ERR2 is shown on the load's front panel. The nominal value (CURR?) does not change in this case.

Die speziellen Zahlenparameter MIN und MAX sind ebenfalls erlaubt.

Beispiele:

**CURR:RANG 10** (10A-Bereich oder nächsthöheren einstellen)

**CURRENT:RANGE MAX**

Wenn ein Bereichsparameter angegeben wird, der bei dem betreffenden Gerät nicht vorhanden ist, wird der nächsthöhere Strombereich eingestellt.

Beispiel ZS1430:

**CURR:RANG 20**

Da das Gerät über keinen 20A-Bereich verfügt, wird der nächsthöhere Strombereich von 30A eingestellt.

**CURRent:RANGe:AUTO**

**ON|1|OFF|0**

Schaltet den Autoranging-Modus im Strombetrieb ein oder aus.

Die Parameter ON oder 1 schalten Autoranging ein, die Parameter OFF oder 0 schalten Autoranging aus, und es muss der gewünschte Strombereich gewählt werden.

Im Autoranging-Betrieb stellt das Gerät je nach programmiertem Stromwert den optimalen Strombereich ein.

Beim Einschalten des Gerätes oder bei Reset ist im Strombetrieb die Autoranging-Funktion aktiviert.

Beispiel:

**CURR:RANG:AUTO ON**

The special numeric parameters MIN and MAX are allowed.

Examples:

**CURR:RANG 10** (Set 10A range or next higher range)

**CURRENT:RANGE MAX**

If a range parameter is set which is not available at the respective device the next possible higher range is set.

Example ZS1430:

**CURR:RANG 20**

Since the device has no 20A range the next higher range of 30A is set.

**CURRent:RANGe:AUTO**

**ON|1|OFF|0**

Switches autoranging in constant current mode on or off.

The parameter ON or 1 switches autoranging on, the parameter OFF or 0 switch autoranging off and the desired current range has to be programmed.

In autoranging mode due to the programmed setting value the device sets the optimum range.

At power-on or reset the autoranging function in current mode is activated.

Example:

**CURR:RANG:AUTO ON**



**Achtung!**

Autoranging ist nicht erlaubt bei einer der folgenden Betriebsarten:

bei der Definition von Modulationsparametern (TRANsient-Subsystem)

- bei der Definition von frei programmierbaren Kurvenformen (PCYClE-Subsystem)
- im MPP Tracking Mode

In diesen Fällen muss vorher ein fester Strombereich gewählt werden.

**CURRent:RANGe?**

Abfrage des Strombereiches in Ampere. Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt.

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Beispiele:

CURR:RANG?

(Antwort von ZS1406: +5.000000E+01)

CURR:RANG? MAX

(Antwort von ZS1406: +1.500000E+02)

**Important!**

Autoranging is not allowed in one of the following items:

when modulation parameters are defined (TRANsient Subsystem)

- when programmable load cycles are defined (PCYClE Subsystem)
- in MPP Tracking mode

In these cases you have to pre-set a fixed current range.

**CURRent:RANGe?**

Queries the current range. The returned value is a numeric value in exponential form.

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible value is queried by appending a white space and the parameters MIN or MAX to the question mark.

Examples:

CURR:RANG?

(Resp. of ZS1406: +5.000000E+01)

CURR:RANG? MAX

(Resp. of ZS1406: +1.500000E+02)

**7.4.2.4 Subsystem DATA|TRACe**

Befehl	Parameter	Einheit	Bemerkung
DATA TRACe :POINts?			Anzahl der gespeicherten Datensätze abfragen
:REMove?	<Nrf>		Definierte Anzahl gespeicherter Datensätze lesen und löschen

**7.4.2.4 Subsystem DATA|TRACe**

Command	Parameter	Unit	Comment
DATA TRACe :POINts?			Query number of stored data sets
:REMove?	<Nrf>		Read and remove defined number of data sets

Das Subsystem DATA dient zum Auslesen der im internen Messdatenspeicher abgelegten Datensätze.

Ein Datensatz besteht aus einem Zeit-, Spannungs-, und Stromwert.

Der Zeitstempel ist immer relativ zum Start bzw. Trigger des entsprechenden Vorgangs (z.B. LIST:STAT ON oder TRIGGER:SOURce TIMER).

**DATA|TRACe:POINts?**

Abfrage der Anzahl der bereits im internen Messwertspeicher abgelegten Messwert-Datensätze.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt.

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Beispiel:

DATA:POIN?

(Antwort z.B.: +5.240000E+01)

**DATA|TRACe:REMove? <Nrf>**

Auslesen und Löschen der angegebenen Anzahl von Messdatensätzen aus dem lastinternen Messwertspeicher.

The subsystem DATA is used to read the measurement data records which are stored in the device-internal memory.

A data record consists of a timestamp, voltage, and current value.

The time is always relative to the start or trigger, respectively, of the corresponding operation (e.g. LIST:STAT ON or TRIGGER:SOURce TIMER).

**DATA|TRACe:POINts?**

Queries the number of data records already stored in the internal measurement memory.

The returned value is a numeric value in exponential form.

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

Example:

DATA:POIN?

(Response e.g.: +5.240000E+01)

**DATA|TRACe:REMove? <Nrf>**

Queries and removes the defined number of data records from the internal measurement buffer.

Wenn der Befehl nach dem Fragezeichen 0 oder keine Zahl enthält, heißt das, dass alle Datensätze angefordert werden.

Die gesendeten Messdaten werden von der Last automatisch gelöscht.

Als erstes werden immer die ältesten Daten gesendet.

Nachdem die Last bereits eine definierte Anzahl von Datensätzen gesendet hat, werden bei erneuter Anforderung von Datensätzen die nächsten im Speicher abgelegten Daten gesendet.

Übersteigt die Anzahl der auszulsenden Datensätze die der tatsächlich verfügbaren, so sendet die Last nur die vorhandenen Datensätze.

Die Last kann maximal 2000 Messdatensätze speichern. Wenn das Ende des Speichers erreicht wird und keine Messdaten von der Last abgeholt worden sind, wird in einer dynamischen Betriebsart (z.B. LIST, SFUNCTION) die Speicherung beendet und ein Bit im Questionable Status Register gesetzt. Im statischen Betrieb wird bei Erreichen des Speicherendes wieder bei 0 begonnen und die älteren Daten werden mit den neuen überschrieben (Ringpuffer).

Die Antwort wird von der Last in folgendem Format gesendet:

```
<Zeit1>,<Spannung1>,<Strom1>,<Zeit2>,<Spannung2>,<Strom2>,...
```

Die Zahlenwerte sind durch Kommata ohne Leerzeichen voneinander getrennt.



Durch eine ungünstige Konstellation der Geschwindigkeiten von Messwert-Lesen und Messwert-Schreiben kann es, abhängig von den Speicherintervallen und der verwendeten Schnittstelle, zu einer Endlosschleife der Messwertspeicherung kommen, wenn der Befehl `DATA:REMove?` oder `DATA:REMove? 0` verwendet wird oder

If the command contains 0 or no number after the question mark all records are requested.

The load automatically deletes the measurement data after they have been transferred.

The oldest values are transferred first.

After the load has sent a defined number of data records it will send the next data located in the memory when a new read occurs.

If the number of requested records exceeds the number of the available records the load will only send the available records.

The load can store up to 2000 data records. If the load reaches the end of memory and no data have been read from the load the storage is quit and a bit in the Questionable Status Register is set when a dynamic mode (e.g. LIST, SFUN) is active. In static mode the beginning of the data memory is overwritten when the end is reached (ring puffer).

The load sends the response in the following format:

```
<Time1>,<Voltage1>,<Current1>,<Time2>,<Voltage2>,<Current2>,...
```

The numbers are separated by commas without any space.



An unfavourable constellation of the speeds of writing and reading stored measurements can cause an endless loop of the data storage which depends on the sample rates and the interface type. This can occur when the command `DATA:REMove?` or `DATA:REMove? 0` is used or when more samples shall be read out than being available.

mehr Datensätze angefordert werden als verfügbar sind.



Wir empfehlen daher, immer die verfügbare Anzahl der gespeicherten Datensätze mit `DATA:POINt?` abzufragen und diese dann mit `DATA:REMove? <Anzahl>` auszulesen.

Beispiel:

`DATA:POIN?` (Antwort: 633)  
`DATA:REM? 633`



Therefore we recommend always to query the number of stored samples using the command `DATA:POINt?` and afterwards to read out this number of samples using the command `DATA:REMove? <NRf>`.

Example:

`DATA:POIN?` (Response: 633)  
`DATA:REM? 633`

**7.4.2.5 Subsystem DELay**

Befehl	Parameter	Einheit	Bemerkung
DElay	<NRf>		Verzögerung zwischen zwei Befehlen im Programmdefinitions-Mode

**7.4.2.5 Subsystem DELay**

Command	Parameter	Unit	Comment
DElay	<NRf>		Set delay between two commands in program mode

**DElay <0...65.5>**

Der DElay-Befehl ist nur gültig im Programmdefinitions-Mode, d.h. wenn vorher mit PROG:BEG die Definition eines Programmes angekündigt worden ist.

Er bestimmt eine Verzögerungszeit bei der Ausführung eines Programms zwischen dem vorhergehenden und dem folgenden Befehl.

Beispiel:

DElay 0.5 (0.5s Delay)

Siehe auch Subsystem PROGRAM.

**DElay <0...65.5>**

The Delay command is only valid in program definition mode, i.e. when previously the definition of a program has been announced by the PROG:BEG command.

It determines the delay between the execution of two commands in a program.

Example:

DElay 0.5 (0.5s delay)

See also subsystem PROGRAM.

**7.4.2.6 Subsystem GTL**

Befehl	Parameter	Einheit	Bemerkung
GTL			in Handbetrieb wechseln

**7.4.2.6 Subsystem GTL**

Command	Parameter	Unit	Comment
GTL			Change to manual operating mode.

**GTL**

Goto Local. Wechselt in den Handsteuerbetrieb. Die Anzeige "Remote" an der Frontplatte des Gerätes erlischt.

Sobald ein neuer Befehl im Gerät ankommt, geht das Gerät wieder in Fernsteuerbetrieb über.

**GTL**

Goto Local. Changes into the manual operating mode. The LED "Remote" goes off.

When a new command arrives in the device, it changes back to the remote controlled mode.

**7.4.2.7 Subsystem INPut|OUTPut**

Befehl	Parameter	Einheit	Bemerkung
INPut OUTPut [:STATE] [:STATE]?	<Boolean>		Lasteingang ein aus Zustand d. Lasteingangs abfragen

**7.4.2.7 Subsystem INPut|OUTPut**

Command	Parameter	Unit	Comment
INPut OUTPut [:STATE] [:STATE]?	<Boolean>		Load Input on  off Query the state of the load input

Das Subsystem INPut|OUTPut dient zum Ein- und Ausschalten des Lasteinganges der elektronischen Last.

The subsystem INPut|OUTPut activates and deactivates the load input of the electronic load.

**INPut|OUTPut[:STATE]  
ON|1|OFF|0**

Lasteingang ein/aus.

**INPut|OUTPut[:STATE]  
ON|1|OFF|0**

Load Input on|off.

Beispiel:

*INP ON* Belastung einschalten

*INP OFF* Belastung ausschalten

Example:

*INP ON* Activate load

*INP OFF* Deactivate load

Die Spannungsmessung kann auch bei abgeschaltetem Lasteingang durchgeführt werden.

The voltage can be measured also for deactivated load input.

**INPut[:STATE]?**

Frägt den Schaltzustand des Lasteinganges ab. Als Rückgabewert wird eine 1 bereitgestellt, wenn der Eingang eingeschaltet ist.

Bei ausgeschaltetem Eingang wird mit 0 geantwortet.

**INPut[:STATE]?**

Queries the state of the load input. The return value is 1, if the input is activated. The return value is 0, if the input is deactivated.

Beispiel:

*INP?* (Antwort bei eingeschaltetem Eingang: 1)

Example:

*INP?* (Resp. for activated input: 1)

7.4.2.8 Subsystem LIST

Befehl	Parameter	Einheit	Bemerkung
LIST			
:CURRent			
[:LEVel]	<NRf>{,<NRf>}		Liste der Strom-Sollwerte definieren
:RTIME	<NRf>{,<NRf>}		Liste der Strom-Rampenzeiten definieren
:DWELL	<NRf>{,<NRf>}		Liste der Strom-Verweilzeiten definieren
:STRamp	<NRf>{,<NRf>}		Liste der Messintervalle für Stromrampen
:STDWell	<NRf>{,<NRf>}		Liste der Messintervalle für Stromverweilzeiten
:POWer			
[:LEVel]	<NRf>{,<NRf>}		Liste der Leistungs-Sollwerte definieren
:RTIME	<NRf>{,<NRf>}		Liste der Leistungs-Rampenzeiten definieren
:DWELL	<NRf>{,<NRf>}		Liste der Leistungs-Verweilzeiten definieren
:STRamp	<NRf>{,<NRf>}		Liste der Messintervalle für Leistungsrampen
:STDWell	<NRf>{,<NRf>}		Liste der Messintervalle für Leistungsverweilzeiten
:RESistance			
[:LEVel]	<NRf>{,<NRf>}		Liste der Widerstands-Sollwerte definieren
:RTIME	<NRf>{,<NRf>}		Liste der Widerstands-Rampenzeiten definieren
:DWELL	<NRf>{,<NRf>}		Liste der Widerstands-Verweilzeiten definieren
:STRamp	<NRf>{,<NRf>}		Liste der Messintervalle für Widerstandsrampen
:STDWell	<NRf>{,<NRf>}		Liste der Messintervalle für Wdst.verweilzeiten
:VOLTage			
[:LEVel]	<NRf>{,<NRf>}		Liste der Spannungs-Sollwerte definieren
:RTIME	<NRf>{,<NRf>}		Liste der Spannungs-Rampenzeiten definieren
:DWELL	<NRf>{,<NRf>}		Liste der Spannungs-Verweilzeiten definieren
:STRamp	<NRf>{,<NRf>}		Liste der Messintervalle für Spannungsrampen
:STDWell	<NRf>{,<NRf>}		Liste der Messintervalle für Spannungsverweilzeiten
:COUNt	<NRf> INFINITY		Anzahl der Listendurchläufe definieren
:STATe	<Boolean>		Ausführung der Kurve starten/stoppen
:STATe?			Ausführungszustand der Liste abfragen

7.4.2.8 Subsystem LIST

Command	Parameter	Unit	Comment
LIST			
:CURRent			
[:LEVel]	<NRf>{,<NRf>}		Define current list
:RTIME	<NRf>{,<NRf>}		Define current ramp list
:DWELL	<NRf>{,<NRf>}		Define current dwell list
:STRamp	<NRf>{,<NRf>}		Define current ramp measurement list
:STDWell	<NRf>{,<NRf>}		Define current dwell measurement list
:POWer			
[:LEVel]	<NRf>{,<NRf>}		Define power list
:RTIME	<NRf>{,<NRf>}		Define power ramp list
:DWELL	<NRf>{,<NRf>}		Define power dwell list
:STRamp	<NRf>{,<NRf>}		Define power ramp measurement list
:STDWell	<NRf>{,<NRf>}		Define power dwell measurement list
:RESistance			
[:LEVel]	<NRf>{,<NRf>}		Define resistance list
:RTIME	<NRf>{,<NRf>}		Define resistance ramp list
:DWELL	<NRf>{,<NRf>}		Define resistance dwell list
:STRamp	<NRf>{,<NRf>}		Define resistance ramp measurement list
:STDWell	<NRf>{,<NRf>}		Define resistance dwell measurement list
:VOLTage			
[:LEVel]	<NRf>{,<NRf>}		Define voltage list
:RTIME	<NRf>{,<NRf>}		Define voltage ramp list
:DWELL	<NRf>{,<NRf>}		Define voltage dwell list
:STRamp	<NRf>{,<NRf>}		Define voltage ramp measurement list



:STDWell	<NRF>{,<NRF>}	Define voltage dwell measurement list
:COUNT	<NRF> INFINITY	Define number of loops
:STATe	<Boolean>	Start/stop list execution
:STATe?		Query list execution state

Das Subsystem LIST ist zur Definition von beliebigen Lastkurvenformen in Form von aneinandergereihten ansteigenden oder abfallenden Geradenteilstücken vorhanden.

Dazu werden die Parameter Sollwert (CURRent/POWer/RESistance/VOLTage), Anstiegs-/Abfallzeit (ramp time RTIME) und Verweildauer (dwell DWELL) jeweils in einer Liste an das Gerät übertragen. Dies ist in allen vier Betriebsarten möglich.

Alle drei Listen (LEVEL, RTIME und DWELL) müssen die gleiche Anzahl von Parametern haben, wobei der erste Einstellwert der Sollwertliste jeweils mit dem ersten der RTIME- sowie der DWELL-Liste korrespondiert, der zweite mit dem zweiten usw.

#### Beispiel:

LIST:CURR 10,20,30,40

LIST:CURR:RTIM 0,0,0,2.5

LIST:CURR:DWEL 0.5,0.5,1,2

#### **Bei installierter Option ZS13:**

Genauso wie die Settings und Einstellzeiten kann für jedes Geradenteilstück ein eigenes Messintervall in Sekunden programmiert werden, mit dem Zeitstempel, Spannung und Strom im geräteinternen Messwertspeicher abgelegt werden.

The LIST subsystem is available to define arbitrary load characteristics in the form of stringing together rising or falling straight lines.

To do this, the parameters nominal value (CURRent/POWer/RESistance/VOLTage), rise/fall time (ramp time RTIME) and dwell (DWELL) are sent to the load as three lists. This is possible in all four operating modes.

The three lists (Setting, RTIME and DWELL) must have equal numbers of parameters, in which the first value of the nominal value list corresponds to the first of the RTIME as well as with the first of the DWELL list, the second with the second and so on.

#### Example:

LIST:CURR 10,20,30,40

LIST:CURR:RTIM 0,0,0,2.5

LIST:CURR:DWEL 0.5,0.5,1,2

#### **With installed ZS13 Option:**

In the same way you can define an own sample rate in seconds for each linear part of the curve. Timestamp, voltage, and current are stored in the device-internal measurement storage with the given sample rate.

Dafür werden die Schlüsselwörter *STRamp* (Sample Time Ramp) und *STDWell* (Sample Time Dwell) in Verbindung mit der Betriebsart verwendet, z.B. im Strombetrieb:

```
LIST:CURR:STR 0,0,0,0.02
LIST:CURR:STDW 0.01,0.01,0.02,0.04
```

Details s.u.

Mit dem Befehl  
*LIST:COUNT <Nrf> |INFinity*  
wird die Anzahl der Zyklen definiert, welche nach dem Start mit  
*LIST:STATe ON*  
ausgeführt werden soll. Beim Parameter *INF* wird die Liste solange ausgeführt, bis sie mit *LIST:STATe OFF* oder *\*RST* gestoppt wird.

Soll die Kurve mittels eines Triggerereignisses gestartet werden, muss außerdem die Triggerquelle definiert werden (s. *TRIGger:SOURce*) und der entsprechende Mode der Betriebsart auf *LIST* programmiert werden, z.B.:  
*CURRent:MODE LIST*  
(s. Subsysteme *CURR*, *POW*, *RES*, *VOLT*)

```
LIST:CURRent[:LEVel] <Nrf> {,<Nrf>}
Definiert eine Stromsollwert-Liste.
LIST:POWer[:LEVel] <Nrf> {,<Nrf>}
Definiert eine Leistungssollwert-Liste.
LIST:RESistance[:LEVel] <Nrf>
{,<Nrf>}
Definiert eine Widerstandssollwert-Liste.
LIST:VOLTAge[:LEVel] <Nrf> {,<Nrf>}
Definiert eine Spannungssollwert-Liste.
```

Die Sollwerte sind durch Kommata voneinander getrennt. Die Parameter *MIN* und *MAX* sind nicht zulässig.

The sample rate is defined using the keywords *STRamp* (sample time ramp) and *STDWell* (sample time dwell) in conjunction with the corresponding operating mode, e.g. for current mode:

```
LIST:CURR:STR 0,0,0,0.02
LIST:CURR:STDW 0.01,0.01,0.02,0.04
```

See below for details.

By the command  
*LIST:COUNT <Nrf> |INFinity*  
the number of cycles is defined which shall be executed after being started by *LIST:STATe ON*.  
If parameter *INF* has been programmed the list is executed as long as it isn't stopped explicitly by one of the commands *LIST:STATe OFF* or *\*RST*.

If the curve shall be started by a trigger event you have to define on one hand the trigger source (see *TRIGger:SOURce*) and on the other hand you must set the mode of the respective operating mode to *LIST*, e.g.:  
*CURRent:MODE LIST*  
(see subsystems *CURR*, *POW*, *RES*, *VOLT*)

```
LIST:CURRent[:LEVel] <Nrf> {,<Nrf>}
Defines a current list.
LIST:POWer[:LEVel] <Nrf> {,<Nrf>}
Defines a power list.
LIST:RESistance[:LEVel] <Nrf>
{,<Nrf>}
Defines a resistance list.
LIST:VOLTAge[:LEVel] <Nrf> {,<Nrf>}
Defines a voltage list.
```

The nominal values are separated by commas. The special parameters *MIN* and *MAX* are not allowed.

Die Sollwerte werden bei der Ausführung nacheinander in der Reihenfolge eingestellt, wie sie in der Liste programmiert worden sind, und zwar mit den dazugehörigen Rampen- und Verweilzeiten der beiden korrespondierenden Listen (s.u.).

Es können maximal **50** Sollwerte in einer Liste definiert werden.

Beispiele:

*LIST:CURR 20.55,77,3.4,0*

*LIST:VOLT:LEV 6E1,5E1, 4E1*

**LIST:CURRent:RTIME <NRf>{,<NRf>}**

Definiert eine Liste für Stromrampen.

**LIST:POWer:RTIME <NRf>{,<NRf>}**

Definiert eine Liste für Leistungsrampen.

**LIST:RESistance:RTIME <NRf>{,<NRf>}**

Definiert eine Liste für Widerstandsrampen.

**LIST:VOLTage:RTIME <NRf>{,<NRf>}**

Definiert eine Liste für Spannungsrampen.

<NRf> = 0 ... 2000 (in s)

Die Rampenzeit gibt an, in welcher Zeit die Last von einem momentan eingestellten Listensetting zum nächsten wechseln soll.

Bei einer Rampenzeit=0 für die nächste Einstellung stellt die Last den nächsten Wert sprunghaft ein, d.h. mit der schnellstmöglichen Anstiegszeit, die die Hardware zulässt.

Beispiel:

*LIST:CURR RTIM 0.05,0,1.0,0*

**LIST:CURRent:DWELL <NRf>{,<NRf>}**

Definiert eine Liste für die Verweilzeiten der Stromliste.

**LIST:POWer:DWELL <NRf>{,<NRf>}**

Definiert eine Liste für die Verweilzeiten der Leistungsliste.

At execution, the nominal values are set in the order they have been programmed in the list, in fact with the necessary ramp and dwell times of the two corresponding lists (see below).

There may be up to **50** nominal values in one list.

Examples:

*LIST:CURR 20.55,77,3.4,0*

*LIST:VOLT:LEV 6E1,5E1, 4E1*

**LIST:CURRent:RTIME <NRf>{,<NRf>}**

Defines a current ramp list.

**LIST:POWer:RTIME <NRf>{,<NRf>}**

Defines a power ramp list.

**LIST:RESistance:RTIME <NRf>{,<NRf>}**

Defines a resistance ramp list.

**LIST:VOLTage:RTIME <NRf>{,<NRf>}**

Defines a voltage ramp list.

<NRf> = 0 ... 2000 (in s)

The ramp time determines in what time the load shall change from the instantaneous list setting to the next.

If the ramp time is 0 for the next setting the load sets a jump to this point with the fastest possible rise time the hardware allows.

Example:

*LIST:CURR RTIM 0.05,0,1.0,0*

**LIST:CURRent:DWELL <NRf>{,<NRf>}**

Defines a list for the dwells of the current list.

**LIST:POWer:DWELL <NRf>{,<NRf>}**

Defines a list for the dwells of the power list.

**LIST:RESistance:DWELL <NRf>****{,<NRf>}**

Definiert eine Liste für die Verweilzeiten der Widerstandsliste.

**LIST:VOLTage:DWELL <NRf> {,<NRf>}**

Definiert eine Liste für die Verweilzeiten der Spannungsliste.

&lt;NRf&gt; = 0.0002 ... 2000 (in s)

Die Verweilzeit gibt an, wie lange das momentane Lastsetting der Liste eingestellt bleibt, bis der Wechsel zum nächsten Wert vollzogen wird.

Eine Verweilzeit=0 z.B. für dreieckförmige Lastverläufe ist praktisch nicht möglich, daher muss anstatt 0 ein kleiner Dwellwert programmiert werden.

**LIST:CURREnt:STRamp <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Rampenzeiten der Stromliste.

**LIST:POWer:STRamp <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Rampenzeiten der Leistungsliste.

**LIST:RESistance:STRamp <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Rampenzeiten der Widerstandsliste.

**LIST:VOLTage:STRamp <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Rampenzeiten der Spannungsliste.

**LIST:CURREnt:STDWell <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Verweilzeiten der Stromliste.

**LIST:POWer:STDWell <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Verweilzeiten der Leistungsliste.

**LIST:RESistance:DWELL <NRf>****{,<NRf>}**

Defines a list for the dwells of the resistance list.

**LIST:VOLTage:DWELL <NRf> {,<NRf>}**

Defines a list for the dwells of the voltage list.

&lt;NRf&gt; = 0.0002 ... 2000 (in s)

The dwell determines how long the instantaneous setting shall be kept until the change to the next one is started.

A dwell=0 which might be wanted for triangular characteristics cannot be realized in practice. Therefore a short dwell value must be programmed instead of 0.

**LIST:CURREnt:STRamp <NRf>****{,<NRf>}**

Defines a list for the sample rate of ramp times in the current list.

**LIST:POWer:STRamp <NRf>****{,<NRf>}**

Defines a list for the sample rate of ramp times in the power list.

**LIST:RESistance:STRamp <NRf>****{,<NRf>}**

Defines a list for the sample rate of ramp times in the resistance list.

**LIST:VOLTage:STRamp <NRf>****{,<NRf>}**

Defines a list for the sample rate of ramp times in the voltage list.

**LIST:CURREnt:STDWell <NRf>****{,<NRf>}**

Defines a list for the sample rate of dwell times in the current list.

**LIST:POWer:STDWell <NRf>****{,<NRf>}**

Defines a list for the sample rate of dwell times in the power list.

**LIST:RESistance:STDWell <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Verweilzeiten der Widerstandsliste.

**LIST:VOLTage:STDWell <NRf>****{,<NRf>}**

Definiert eine Liste für das Messintervall der Verweilzeiten der Spannungsliste.

Folgendes Bild soll die Einsetzbarkeit von verschiedenen Messraten verdeutlichen:

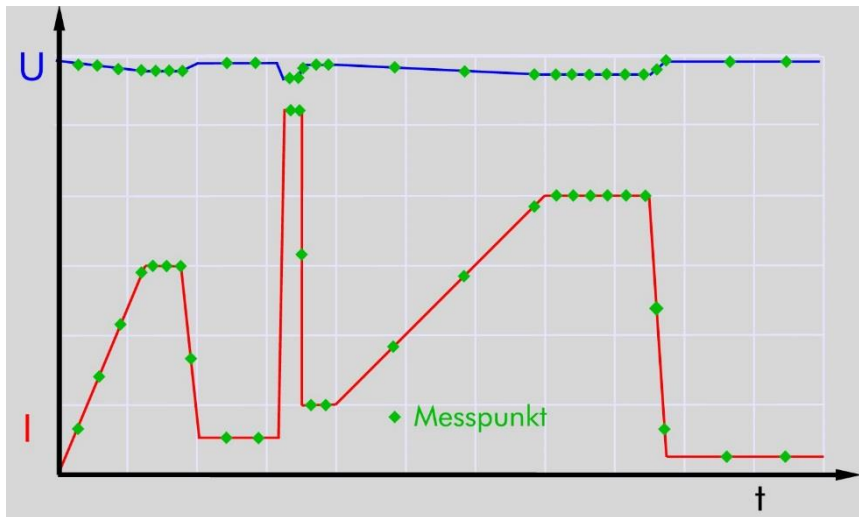
**LIST:RESistance:STDWell <NRf>****{,<NRf>}**

Defines a list for the sample rate of dwell times in the resistance list.

**LIST:VOLTage:STDWell <NRf>****{,<NRf>}**

Defines a list for the sample rate of dwell times in the voltage list.

The following picture shall clarify the use of different sample rates for the lines:

**Hinweis:**

Die Messdatenspeicherung ist immer aktiv, auch wenn diese nicht explizit mit *LIST:CURRent:STRamp* bzw. *LIST:CURRent:STDWell* (entsprechend für VOLT/RES/POW) programmiert wird. Die Messintervall-Listen sind mit  $200\mu\text{s}$  vorbelegt. 0 wird intern zum Minimalintervall  $200\mu\text{s}$  umgewandelt.

**Note:**

Sampling and storing records is always active, also if they haven't been programmed by *LIST:CURRent:STRamp* or *LIST:CURRent:STDWell* (respectively for VOLT/POW/RES). The sample rate lists are pre-defined with  $200\mu\text{s}$ . 0 is internally converted to the minimum interval of  $200\mu\text{s}$ .

Wenn keine Option ZS13 (Data Acquisition) installiert ist, macht eine Messrate unter 330ms keinen Sinn, da der Standard-AD-Wandler nicht schneller messen kann.

Das Auslesen der Messdatensätze erfolgt über die Befehle im Subsystem *DATA*.

### **LIST:COUNt <0..65535> |INFinity**

Bestimmt die Anzahl der beim Start der Kurve auszuführenden Perioden.

Es können nur ganze Perioden ausgeführt werden. Wenn als Parameter eine Nicht-Ganzzahl programmiert wird, rundet die Last den Wert zur nächsten Ganzzahl auf oder ab.

Ist der Parameter dieses Befehls > 1, dann wird nach dem letzten Wert der Liste diese wieder von vorne gestartet, und zwar so lange, bis die Anzahl der programmierten Zyklen abgelaufen ist. Der letzte Wert der Liste bleibt dann statisch eingestellt.

Mit dem Parameter *INFinity* erreicht man eine Endlosschleife, d.h. die Periode wird dauernd wiederholt und kann nur durch den Befehl *LIST:STATe OFF* oder *\*RST* abgebrochen werden. Bei Abbruch mit *LIST:STATe OFF* stellt die Last den letzten statischen Einstellwert ein, der vor dem Start der Liste eingestellt war.

#### Beispiele:

```
LIST:COUN 2
LIST:COUN +2.0E+01
LIST:COUN INF
```

### **LIST:STATe ON|OFF|1|0**

Startet oder stoppt die programmierte Liste.

If no ZS13 option is installed the shortest measurement interval is 330ms since the standard AD converter cannot take data faster.

The reading of the measured data sets is done by the commands in the *DATA* subsystem.

### **LIST:COUNt <0..65535> |INFinity**

Determines the number of periods being generated when the list is started.

Only complete periods can be executed. If a non-integer is programmed as parameter the load will round the parameter to the next integer.

If this command's parameter is > 1 the list will be restarted with the first value after the dwell of the last value has expired.

When the programmed periods have been executed the last list value is kept as static value.

Programming *INFinity* as parameter generates an endless loop when the list is started. In this case the list execution can only be stopped with one of the commands *LIST:STATe OFF* or *\*RST*. At termination with *LIST:STATe OFF* the load will set the last static setting value which had been set before the list execution.

#### Examples:

```
LIST:COUN 2
LIST:COUN +2.0E+01
LIST:COUN INF
```

### **LIST:STATe ON|OFF|1|0**

Starts or stops the list.

Der Parameter ON oder 1 startet die Ausführung der Liste. OFF oder 0 stoppt die Ausführung und der letzte statische Wert, der vor der Kurvenausführung eingestellt war, wird wieder gesetzt.

Das Starten einer Liste setzt den Messwertspeicher-Index automatisch auf 0 zurück, auch wenn zuvor gespeicherte Datensätze noch nicht ausgelesen worden sind.

Beispiele:

LIST:STAT ON  
LIST:STAT 1  
LIST:STAT OFF

**LIST:STATE?**

Abfrage über den Ausführungszustand der Liste.

Als Rückgabewert wird eine 1 bereitgestellt, wenn die programmierte Liste gerade ausgeführt wird.

Bei momentan nicht ausgeführter Liste wird mit 0 geantwortet.

Beispiel:

LIST:STATE? (Antwort bei  
laufender Liste: 1)

Programmierbeispiele s. folgende Seiten.

The parameter ON or 1 starts the list execution. OFF or 0 stops the execution and the last static value which had been set before the list execution is set again.

Starting a list automatically resets the storage index for measurements to 0, even if previously saved records haven't been read out.

Examples:

LIST:STAT ON  
LIST:STAT 1  
LIST:STAT OFF

**LIST:STATE?**

Queries the execution state of the list. The return value is 1, if the list is currently being executed.

The return value is 0, if the list is not being executed.

Example:

LIST:STATE? (Resp. for running  
list: 1)

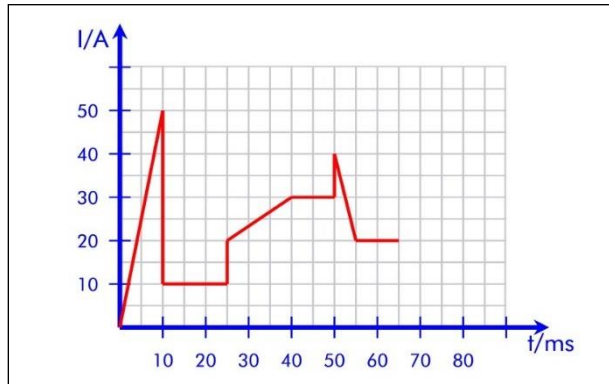
See following pages for programming examples.

**Beispiel 1 zur LIST Programmierung**

Zur Veranschaulichung der Zusammenhänge von Sollwerten und Zeiten soll folgende Stromkurve programmiert und genau einmal ausgeführt werden.

**LIST Programming Example 1**

To illustrate the cohesion of setting values and times the following current curve shall be programmed as a list and executed once.



List Programming Example 1

**Programmierung:****Programming:**

```

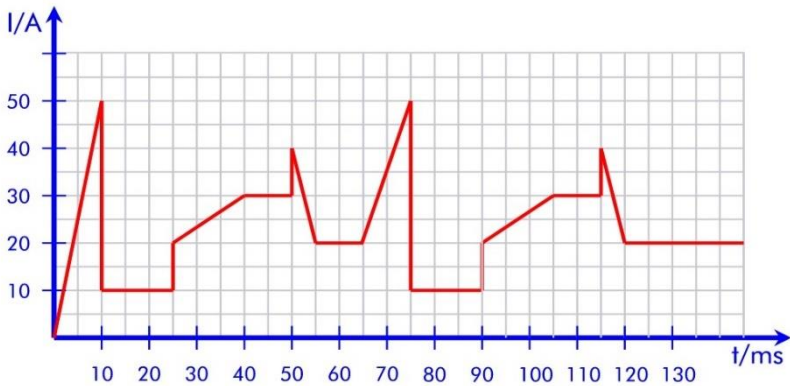
MODE:CURR
CURR:RANG 50::CURR 0
LIST:CURR 50,10,20,30,40,20
LIST:CURR:RTIM 0.01,0,0,0.015,0,0.005
LIST:CURR:DWEL 0.001,0.015,0.001,0.01,0.001,0.01
LIST:COUN 1
INP ON
LIST:STAT ON

```

Die Lastkurve wird einmal durchgeführt. Am Ende bleibt der letzte Listenwert statisch stehen.

The load curve is executed once. Then the last list setting is kept statically.



**Beispiel 2 mit LIST:COUNT 2 :****Example 2 with LIST:COUN 2 :**

List Programming Example 2

**Programmierung:****Programming:**

```

MODE:CURR
CURR:RANG 50,;CURR 0
LIST:CURR 50,10,20,30,40,20
LIST:CURR:RTIM 0.01,0,0,0.015,0,0.005
LIST:CURR:DWEL 0.001,0.015,0.001,0.01,0.001,0.01
LIST:COUN 2
INP ON
LIST:STAT ON

```

Nach Ablauf des letzten Listenwertes wird wieder der erste Wert mit der zugehörigen Rampe angefahren.

Am Ende der programmierten COUNT-Zahl bleibt der letzte Listenwert eingestellt.

Wenn eine laufende Kurvenform an einer beliebigen Stelle mit dem Befehl *LIST:STAT OFF* abgebrochen wird, stellt die Last den vor der Liste programmierten statischen Sollwert ein.

After the last list setting has expired the setting will again run to the first value with the first ramp time.

At the end of the programmed COUNT number the last list setting is kept.

If a running waveform is stopped by the command *LIST:STAT OFF* the load sets the last static setting value programmed before the list.

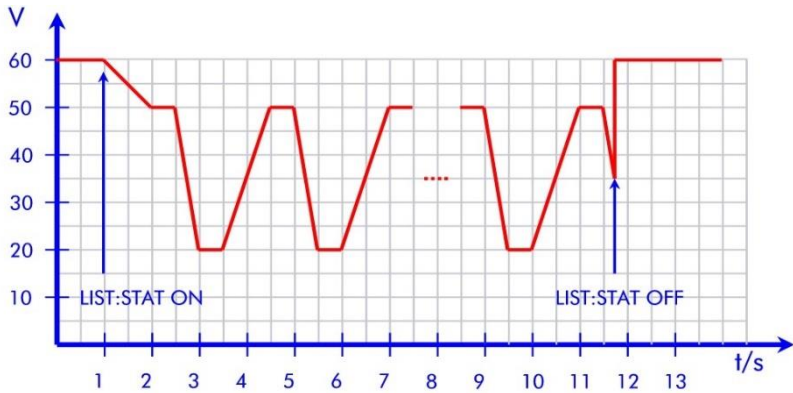
**Beispiel 3:** Kontinuierliches Spannungsprofil mit statischem Wert von 60V

**Example 3:** Continuous voltage profile with 60V static setting

**Programmierung:**

**Programming:**

```
MODE:VOLT  
VOLT 60  
LIST:VOLT 50,20  
LIST:VOLT:RTIM 1,0.5  
LIST:VOLT:DWEL 0.5,0.5  
LIST:COUN INF  
INP ON  
LIST:STAT ON
```



List Programming Example 3

**Beispiel 4:** Kontinuierliches Spannungsprofil von Beispiel 3 mit Speicherung von U und I. Im Anstieg soll alle 333ms ein Datensatz gespeichert werden, im Abfall alle 125ms. In der oberen und unteren Verweildauer soll jeweils nach 250ms ein Datensatz gespeichert werden.

--> Hier ist der Einsatz des schnellen AD-Wandlers erforderlich; Option ZS13 muss in der Last sein!

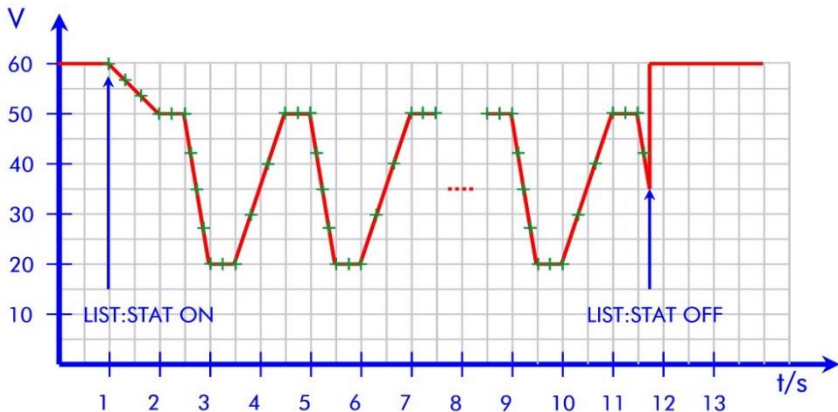
**Example 4:** Continuous voltage profile of example 3 with recording of V and I. Records shall be taken every 330ms in the rising ramp and every 125ms in the falling ramp. In both dwell parts every 250ms a record shall be taken.

--> In this case the fast A/D converter must be used; Option ZS13 must be installed in the load!

#### Programmierung:

```
SET:ADC FAST
MODE:VOLT
VOLT 60
LIST:VOLT 50,20
LIST:VOLT:RTIM 1,0.5
LIST:VOLT:DWEL 0.5,0.5
LIST:VOLT:STR 0.333,0.125
LIST:VOLT:STDW 0.25,0.25
LIST:COUN INF
INP ON
LIST:STAT ON
```

#### Programming:



Die gespeicherten Messdatensätze können mit dem Befehl DATA:REMove? {<NRf>} ausgelesen werden (s. Subsystem DATA).

The saved records may be read out using the command DATA:REMove? {<NRf>}. See subsystem DATA.

**Beispiel 5: Extern getriggerte Kurve**

Eine definierte Kurvenform (Sollwertliste) kann auch durch einen externen Trigger gestartet werden. Dazu muss die Triggerquelle definiert werden und der Mode der betreffenden Betriebsart auf LIST programmiert werden.

Rechteckstrom 15A/1s, 0A/2s durch externen Triggerimpuls starten:

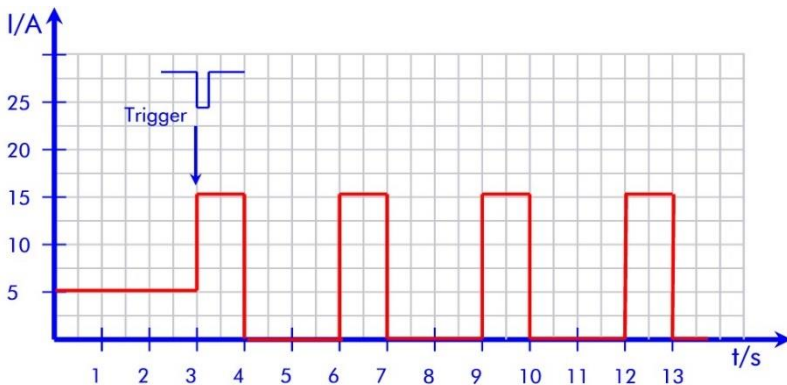
```
MODE:CURR
CURR:RANG 15;LEV 5
CURR:MODE LIST
LIST:CURR 15,0
LIST:CURR:RTIM 0,0
LIST:CURR:DWEL 1,2
LIST:COUN INF
TRIG:SOUR EXT
INP ON
```

**Example 5: Externally triggered waveform**

A predefined waveform (list) may also be started by an external trigger. For that, the trigger source must be defined and the mode of the concerning operating mode must be set to LIST.

Rectangular current with 15A/1s, 0A/2s shall be started by an external trigger:

```
MODE:CURR
CURR:RANG 15;LEV 5
CURR:MODE LIST
LIST:CURR 15,0
LIST:CURR:RTIM 0,0
LIST:CURR:DWEL 1,2
LIST:COUN INF
TRIG:SOUR EXT
INP ON
```



List Programming Example 5

Ein erneuter Trigger in der laufenden Liste startet diese von vorn. Die Funktion ist also nachtriggerbar. Mit dem Befehl `LIST:STAT OFF` kann die Kurvenform jederzeit gestoppt werden.

A new trigger in the running list restarts the list, that means the function is retriggerable. By the command `LIST:STAT OFF` the waveform can be stopped.

## 7.4.2.9 Subsystem MEASure

Befehl	Parameter	Einheit	Bemerkung
MEASure :CHARge [:DC]?			Ladung abfragen (MPPT)
:ENERgy [:DC]?			Energie abfragen (MPPT)
:CURRent [:DC]?			Strommesswert abfragen
:EXTernal [:DC]?			Ext. Messeingang abfragen
:MPP [:DC]?			Maximalen Leistungspunkt abfragen
:POWer [:DC]?			Leistungsmesswert abfragen
:VOLTage [:DC]?			Spannungsmesswert abfr.
:RESistance [:DC]?			Widerstandsmesswert abfr.

## 7.4.2.9 Subsystem MEASure

Command	Parameter	Unit	Comment
MEASure :CHARge [:DC]?			Query accumulated charge (MPPT)
:ENERgy [:DC]?			Query accumulated energy (MPPT)
:CURRent [:DC]?			Measure current
:EXTernal [:DC]?			Measure external voltage
:MPP [:DC]?			Measure Maximum Power Point
:POWer [:DC]?			Measure power
:VOLTage [:DC]?			Measure input voltage
:RESistance [:DC]?			Measure resistance

Neben den Einstellmöglichkeiten für die verschiedenen Belastungsarten bietet das Gerät auch die Möglichkeit, Spannung, Strom, Widerstand und Leistung sowie bei Verwendung eines optional schnelleren ADC ein externes Signal zu messen und die Messwerte an den Steuerrechner zu übergeben.

Apart from the settings for the different load states the device offers the possibility to measure voltage, current and power as well as in use with Option ZS13 an external signal, and to pass the measuring values to a controlling computer.

Es handelt sich dabei um eine vom Einstellkreis unabhängige Messung über einen 18-Bit AD-Wandler (incl. Vorzeichen), welcher laufend die messbaren Größen scannt und den jeweils letzten Wert speichert.



Der zuletzt gespeicherte Wert wird bei der Messdatenabfrage übertragen und kann bis zu 330ms alt sein. D.h. um sicher zu sein, einen nach einer geänderten Einstellung aktuellen Messwert zu erhalten, sollte nach einer Setting- oder Bereichsänderung erst nach 330ms ein neuer Messwert angefordert werden.

Für schnelle Anwendungen kann ein zweiter, schnellerer 14-Bit ADC in das Analoginterface eingebaut werden, dessen Messwerte max. 50ms alt sind (Option ZS13). Um die Messwerte mit diesem schnellen ADC zu erfassen, muss das Kommando SET:ADC FAST gegeben werden (s. SubSystem SETup).



Wie die Last programmiert wird, um automatisch in bestimmten Zeitintervallen Messdatensätze im internen Datenspeicher abzulegen, sehen Sie im Subsystem TRIGger.

#### Hinweis:

Die ZS Lasten gehen in Remotezustand, sobald irgendein Befehl über eine der Schnittstellen im Gerät ankommt. Bei Messbefehlen gilt eine Ausnahme. War das Gerät bisher im Handbetrieb, so schaltet ein MEAS:xx Befehl das Gerät nicht in Remotebetrieb. Dies ist z.B. nützlich, wenn man die Last extern ansteuert und die Messwerte von der Last abfragt.

The measurement is independent of the setting circuit and takes place via a 18 Bit A-D converter (incl. sign) which continuously scans the measurable values and stores the most recent ones.



The topical value is sent by the load when the MEASure command is received. This value may be up to 330ms old. That means, if you want to be sure to get a refreshed measurement value after a setting or a range change has taken place the measurement shall be requested 330ms after the new setting. For more dynamic applications a further, faster ADC can be implemented into the analog interface whose scanned values may be maximum 50ms old (Option ZS13). To measure all values with this fast ADC you have to send once the command SET:ADC FAST to the load (see also SubSystem SETup).



See how the load is programmed to automatically store measurement records in internal memory in subsystem TRIGger.

#### Note:

ZS loads go to Remote operation as soon as any command is received by one of the interfaces. There is an exception with measurement commands. If the device has previously been in manual operation a MEAS:xx command does not set remote operation active. This is suitable e.g. when you want to control the load analogously but take the measurements from the load.

Da der Remote- vom Handbetrieb unabhängig ist, muss man bei Mehrbereichsgeräten evtl. den Einstell-/Messbereich voreinstellen (z.B. CURR:RANG xx) und dann mit GTL wieder in Localbetrieb wechseln.

Mit den Kommandos für die Messungen wird das Gerät dazu veranlasst, **einen** Messwert zur Ausgabe bereitzuhalten. Innerhalb eines Befehlsstrings darf immer nur maximal ein Abfragebefehl enthalten sein.

Die Antwort auf diesen Abfragebefehl muss erst ausgelesen werden, bevor ein nächster Abfragebefehl an das Gerät geschickt wird.

Als Rückgabewert wird eine einheitslose Zahl im Exponentialformat bereitgestellt (Ausnahme: MEAS:MPP?, s.u.):

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Das Format des Messwertstrings kann verändert werden (s. SYSTem:STRing).

### MEASure:CHARge[:DC]?

Kumulierte Ladung in Ah abfragen.

Diese Funktion ist nur im MPP Tracking Mode und im Battery Test Mode verfügbar. Ist MPP Tracking oder Battery Test Mode aktiv und der Lasteingang wird eingeschaltet, wird der interne Ah-Zähler zurückgesetzt und die Ladungsmenge aufkumuliert, bis der Lasteingang ausgeschaltet wird.

Der Ah-Zähler wird durch INP ON (im MPP Mode) zurückgesetzt.

#### Beispiel:

MEAS:CHAR? (Antwort z.B.:  
+9.550500E+00)

Since remote operation settings are independent from manual settings at multi-range devices you may have to preset the range (e.g. CURR:RANG xx) and then change to manual operation with GTL.

With the measurement commands the device will send **one** measured value. A command string may only include one query command.

The answer for this query command has to be read before the next query command can be sent to the device.

The return value is a numeric value without unit in exponential form (Exception: MEAS:MPP?, see below):

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

The format of the measurement string may be changed (see SYSTem:STRing).

### MEASure:CHARge[:DC]?

Query accumulated charge in Ah.

This function is only available in MPP Tracking or Battery Test mode. If MPP Tracking or Battery Test mode is active and the load input is switched on then the internal Ah counter is reset and the charge is added up until the load input is switched off.

The Ah counter is reset by the command INP ON (in MPP mode).

#### Example:

MEAS:CHAR? (Response e.g.:  
+9.550500E+00)

**MEASure:ENERgy[:DC]?**

Kumulierte Energie in Wh abfragen.

Diese Funktion ist nur im MPP Tracking Mode und im Battery Test Mode verfügbar. Ist MPP Tracking oder Battery Test Mode aktiv und der Lasteingang wird eingeschaltet, wird der interne Wh-Zähler zurückgesetzt und die Energiemenge aufkumuliert, bis der Lasteingang ausgeschaltet wird.

Der Wh-Zähler wird durch INP ON (im MPP Mode) zurückgesetzt.

Beispiel:

MEAS:ENER? (Antwort z.B.:  
+3.580500E+01)

**MEASure:CURRent[:DC]?**

Aktuellen Laststrom messen.

Beispiel:

MEAS:CURR? (Antwort z.B.:  
+1.550700E+01)

**MEASure:EXTErnal[:DC]?**

Nur bei Option ZS13 "Data Acquisition". Aktuelle Externspannung an Analog-I/O-Stecker messen (s. Kap. 7.4.2.15 SET:ADC und Hardwareteil Kap. Analog-I/O-Stecker).

Beispiel:

MEAS:EXT? (Antwort z.B.:  
+8.550700E+00)

**MEASure:MPP[:DC]?**

Maximum Power Point (Punkt der maximalen Leistung des Prüflings) messen.

Bei diesem Messbefehl sendet die Last vor dem eigentlichen Messwert ein Flag als Boolesche 0 oder 1, mit Komma vom Messwert getrennt.

Dabei bedeutet:

**MEASure:ENERgy[:DC]?**

Query accumulated charge in Wh.

This function is only available in MPP Tracking or Battery Test mode. If MPP Tracking or Battery Test mode is active and the load input is switched on then the internal Wh counter is reset and the energy is added up until the load input is switched off.

The Wh counter is reset by the command INP ON (in MPP mode).

Example:

MEAS:ENER? (Response e.g.:  
+3.580500E+01)

**MEASure:CURRent[:DC]?**

Measure actual load current.

Example:

MEAS:CURR? (Response e.g.:  
+1.550700E+01)

**MEASure:EXTErnal[:DC]?**

Only with option ZS13 "Data Acquisition". Measure actual external voltage at analog I/O connector (see also chapter 7.4.2.15 hardware manual chapter Analog I/O connector).

Example:

MEAS:EXT? (Response e.g.:  
+1.550700E+00)

**MEASure:MPP[:DC]?**

Measure Maximum Power Point of the UUT.

At this measurement command, the load sends a flag as boolean 0 or 1 in front of the measurement value itself, separated by a comma.

There means:



- 0 Maximum Power Point nicht gefunden. Möglich, wenn keine MPP-Betriebsart aktiv oder die Last den MPP gerade nachregeln muss.
- 1 Maximum Power Point gefunden und aktiv.

Beispiel:

MEAS:MPP? (Antwort z.B.:  
0,+1.155000E+02)

S. auch Subsystem *MODE:MPP*.

**MEASure:POWer[:DC]?**

Aktuelle Leistung messen.

Beispiel:

MEAS:POW? (Antwort z.B.:  
+1.155000E+02)

**MEASure:RESistance[:DC]?**

Aktuellen Widerstand messen.

Beispiel:

MEAS:RES? (Antwort z.B.:  
+1.155000E+00)

**MEASure:VOLTage[:DC]?**

Aktuelle Eingangsspannung messen.

Beispiel:

MEAS:VOLT? (Antwort z.B.:  
+1.155000E+02)

- 0 Maximum Power Point not found. Possible when no MPP mode is active or the load is adjusting the MPP.

- 1 Maximum Power Point found and active.

Example:

MEAS:MPP? (Response e.g.:  
0,+1.155000E+02)

See also Subsystem *MODE:MPP*.

**MEASure:POWer[:DC]?**

Measure actual power.

Example:

MEAS:POW? (Response e.g.  
+1.155000E+02)

**MEASure:RESistance[:DC]?**

Measure actual resistance.

Example:

MEAS:RES? (Response e.g.  
+1.155000E+00)

**MEASure:VOLTage[:DC]?**

Measure actual input voltage.

Example:

MEAS:VOLT? (Response e.g.  
1.155000E+02)

**7.4.2.10 Subsystem MODE|FUNCTION**

Befehl	Parameter	Einheit	Bemerkung
MODE FUNCTION :CURRent [:DC] :RESistance [:DC] :POWER [:DC] :VOLTage [:DC] :MPP MODE FUNCTION?			Betriebsart Konstantstrom  Betriebsart Konstantwiderstand  Betriebsart Konstantleistung  Betriebsart Konst.spannung Betriebsart MPP Tracking aktuelle Betriebsart abfragen

**7.4.2.10 Subsystem MODE|FUNCTION**

Command	Parameter	Unit	Comment
MODE FUNCTION :CURRent [:DC] :RESistance [:DC] :POWER [:DC] :VOLTage [:DC] :MPP MODE FUNCTION?			Operating Mode Constant Current Operating Mode Constant Resistance Operating Mode Constant Power Operating Mode Constant Voltage Operating Mode MPP Tracking Query actual operating mode

Die Geräte der Serie ZS können in den Betriebsarten Konstant-Strom (CURR), Konstant-Leistung (POW), Konstant-Widerstand (RES) und Konstant-Spannung (VOLT) arbeiten. Außerdem gibt es in Verbindung mit Option ZS13 in der Betriebsart VOLTage noch die Möglichkeit, den Punkt der maximalen Leistung zu finden und nachzuregeln: "MPP Tracking" (Maximum Power Point Tracking).

The electronic loads can be operated in the modes constant current (CURR), constant resistance (RES), constant power (POW) and constant voltage (VOLT). Furthermore, with option ZS13 there is the possibility to find and control the point with the most power in the mode VOLTage mode: "MPP Tracking" (Maximum Power Point Tracking).

Durch Senden der entsprechenden Kommandos schaltet die Elektronische Last automatisch den Lasteingang aus, stellt die Betriebsart um und schaltet anschließend den Lasteingang wieder ein, sofern dieser vorher eingeschaltet war. Eine Ausnahme gilt bei MPP Tracking. Bevor das Kommando MODE:MPP an die Last gegeben wird, muss der Lasteingang ausgeschaltet sein. Dann muss MODE:VOLT gewählt werden, MODE:MPP gesendet und als letztes erst der Eingang eingeschaltet werden.

Für jede Betriebsart wird der jeweilige Einstellbereich sowie der Sollwert gespeichert, und beim Umschalten der Betriebsart automatisch eingestellt. Wurde in der jeweiligen Betriebsart zuvor noch kein Einstellwert programmiert, so wird der Default-Wert im Default-Range eingestellt (s. \*RST).

Die Default-Betriebsart nach dem Einschalten ist CURRent.

### **MODE:CURRent[:DC]**

In Konstantstrombetrieb wechseln und zuletzt programmierten Wert einstellen.

Beispiel:

MODE:CURR

### **MODE:POWer[:DC]**

In Konstantleistungsbetrieb wechseln und zuletzt programmierten Wert einstellen.

Beispiel:

MODE:POW

### **MODE:RESistance[:DC]**

In Konstantwiderstandsbetrieb wechseln und zuletzt programmierten Wert einstellen.

Beispiel:

MODE:RES

By sending the corresponding command the electronic load automatically switches the input off, changes the operating mode and afterwards switches the input on if previously the input has been on. There is an exception with MPP Tracking. Before the command MODE:MPP is sent to the load the input must be switched off. Then MODE:VOLT must be chosen, MODE:MPP be sent and last the input must be switched on.

For each operating mode the setting range and the nominal value are saved and set when the mode is changed.

If no setting value has been programmed for this operating mode, the default value in the default range (see \*RST) is set.

The default mode after activation is CURRent.

### **MODE:CURRent[:DC]**

Change to operating mode constant current and set the last programmed value.

Example:

MODE:CURR

### **MODE:POWer[:DC]**

Changes to the operating mode constant power and set the last programmed value.

Example:

MODE:POW

### **MODE:RESistance[:DC]**

Change to the operating mode constant resistance and set the last programmed value.

Example:

MODE:RES

**MODE:VOLTage[:DC]**

In Konstantspannungsbetrieb wechseln und zuletzt programmierten Wert einstellen.

Beispiel:

MODE:VOLT

**MODE:MPP**

In Maximum Power Point Tracking Mode wechseln. Nur möglich mit Option ZS13!

Diese "Betriebsart" ist nur im Spannungsbetrieb möglich, d.h. die Last muss sich zum Zeitpunkt des Befehls MODE:MPP im VOLT Mode befinden, ansonsten leuchtet die Err2-Meldung am Gerät auf. Bei Mehrbereichsgeräten muss außerdem der Strombereich im Spannungsbetrieb (VOLTage:CRANge) vorgewählt werden.

Der Wechsel in MPP Tracking Mode ist nur bei ausgeschaltetem Eingang möglich. Erst nach dem Befehl MODE:MPP kann der Eingang mit INP ON eingeschaltet werden. Die Last schaltet dann automatisch in den SLOW-Regelbetrieb.

Beim MPP Start (im Spannungs-Mode) wird die Spannung solange verringert, bis sich eine negative Leistungsänderung ergibt.

Dann erfolgt das eigentliche MPP Tracking: die Spannung wird jeweils solange erhöht/verringert, bis sich eine negative Leistungsänderung ergibt und dann eine Richtungsänderung vorgenommen.

Um Richtungsänderungen der Spannungsvariation aufgrund von Messungenauigkeiten zu vermeiden, muss sich die Leistung um einen bestimmten Wert ändern, bevor entschieden wird, in welche Richtung die Spannung geändert werden muss.

**MODE:VOLTage[:DC]**

Change to the operating mode constant voltage and set the last programmed value.

Example:

MODE:VOLT

**MODE:MPP**

Change to Maximum Power Point Tracking Mode. Only possible with option ZS13!

This "Operating Mode" is only possible in constant voltage mode, i.e. the load has to be in VOLT mode when it receives the MODE:MPP command. Otherwise, the Err2 message will appear at the device. Furthermore, at devices with more than one current setting range the current range in voltage mode must be pre-selected (VOLTage:CRANge).

Changing to MPP Tracking is only possible when the input is switched off. After MODE:MPP has been programmed, the input may be switched on with INP ON. Then the load automatically switches to SLOW regulation speed.

At MPP start (in voltage mode) the voltage is reduced until the resulting power decreases.

Then MPP Tracking is controlled in the way that the voltage is continuously increased/decreased when a certain power difference is present. If the power difference becomes negative then the voltage direction will be reversed.

To avoid direction changes of the voltage variation because of measurement inaccuracies the power difference must be of a certain amount to decide in which direction the voltage adjustment must be made.

Das rührt daher, dass bei den meisten Lasten der theoretische Bereich der Leistungsmessung viel größer ist als die Maximalleistung der Last ( $U_{max} \cdot I_{max} \gg P_{max}$ ). Daher würde ein Promille Leistungsänderung schon in der Messgenauigkeit untergehen. Der Mindestwert dieser Leistungsänderung wird durch den Systemparameter `Delta_P_min` (Nr. 53) vorgegeben.

S. auch Subsystem `SYSTEM:PARAMeter`.

Der MPP Tracking Mode wird mit jedem anderen `MODE`-Befehl (`MODE:CURR`, `MODE:VOLT`, `MODE:POW`, `MODE:RES`) wieder deaktiviert, vorher muss jedoch der Lasteingang ausgeschaltet werden (`INP OFF`).

#### Beispiel nach Power on:

<code>MODE:VOLT</code>	Spannungsbetrieb
<code>MODE:MPP</code>	MPP Tracking
<code>INP ON</code>	Lasteingang ein
...	
<code>INP OFF</code>	Lasteingang aus
<code>MODE:CURR</code>	Konst.-Strombetrieb
<code>CURR 10</code>	10A
<code>INP ON</code>	Lasteingang ein

Bei MPP Tracking kumuliert die Last automatisch die Ladung in Ah und die Energie in Wh. Diese beiden Werte können mit `MEASure:CHARge?` bzw. `MEASure:ENERgy?` ausgelesen werden (s. Subsystem `MEASure`).

#### Hinweis:

MPP Tracking erfordert die Angabe einiger Parameter des Prüflings (Solarpanel etc.). Diese werden mit `SYSTEM:PARAMeter` (s. Subsystem `SYSTEM`) programmiert und sind mit Werkseinstellungen vorbelegt.

The reason for this is that at most loads the theoretical range for the power measurement is much higher than the maximum power of the load ( $U_{max} \cdot I_{max} \gg P_{max}$ ). Therefore one part per thousand would be lost in the measurement accuracy. The minimum level of this power difference is determined by the system parameter `Delta_P_min` (No. 53).

See also subsystems `SYSTEM:PARAMeter`.

The MPP Tracking mode is deactivated with any of the other `MODE` commands (`MODE:CURR`, `MODE:VOLT`, `MODE:POW`, `MODE:RES`), but previously the input must be switched off (`INP OFF`).

#### Example after Power on:

<code>MODE:VOLT</code>	Voltage Mode
<code>MODE:MPP</code>	MPP Tracking
<code>INP ON</code>	Load Input on
...	
<code>INP OFF</code>	Load Input off
<code>MODE:CURR</code>	Const. Current
<code>CURR 10</code>	10A
<code>INP ON</code>	Load Input on

In MPP Tracking mode the load automatically calculates the charge in Ah and the energy in Wh. These two values can be read from the device with `MEASure:CHARge?` or `MEASure:ENERgy?`, respectively (see subsystem `MEASure`).

#### Note:

MPP Tracking requires the definition of a few parameters of the UUT (solar panel etc.). These parameters are programmed with `SYSTEM:PARAMeter` and predefined with factory settings.

Spezialfunktionen sollten immer explizit beendet werden, bevor in einer anderen Betriebsart weitergearbeitet wird.

**MODE?**

Momentane Betriebsart abfragen.  
Als Rückgabewert wird die Kurzform der entsprechenden Betriebsart bereitgestellt (*CURR*, *POW*, *RES* oder *VOLT*).

Im MPP Tracking Mode liefert die Last als Antwort *VOLT\_MPP*.

Beispiel:

*MODE?* (Antwort z.B.: *RES*)

Special functions shall explicitly be deactivated before operating the device in a different operating mode.

**MODE?**

Queries the topical operating mode.  
As response the short form of the respective SCPI mode keyword is returned (*CURR*, *POW*, *RES* or *VOLT*).

In MPP Tracking mode the load returns *VOLT\_MPP*.

Example:

*MODE?* (Response for example: *RES*)

## 7.4.2.11 Subsystem PCYCLE

Befehl	Parameter	Einheit	Bemerkung
PCYCLE			programmierbare Kurve
:CURRent	<NRf>,<NRf>	[A MA]	Tabellenzeile (Par1) mit Stromwert (Par2) beschr.
:POWer	<NRf>,<NRf>	[MW W KW]	Tabellenzeile (Par1) mit Leistungswert (Par2) beschr.
:RESistance	<NRf>,<NRf>	[OHM KOHM MOHM]	Tabellenzeile (Par1) mit Widerstandswert (Par2) beschr.
:VOLTage	<NRf>,<NRf>	[MV V]	Tabellenzeile (Par1) mit Spannungswert (Par2) beschreiben
:TIME	<NRf>,<NRf>	[S MS]	Tabellenzeile (Par1) mit Zeitwert (Par2) beschreiben
:TRIGgered	<Boolean>		Triggerfunktion f. Kurvenform ein   aus
:MODE	CONTinuous PULSe,<NRf>		Kontinuierliche Kurvenform oder bestimmte Anzahl von Perioden
:MODE?			Kurvenformmodus abfragen
:STATe	<Boolean>		Ausgabe der Kurvenform ein aus
:STATe?			Zustand der Kurvenausgabe abfragen

## 7.4.2.11 Subsystem PCYCLE

Command	Parameter	Unit	Command
PCYCLE			Programmable waveform
:CURRent	<NRf>,<NRf>	[A MA]	Set table row (Par1) with current value (Par2)
:POWer	<NRf>,<NRf>	[MW W KW]	Set table row (Par1) with power value (Par2)
:RESistance	<NRf>,<NRf>	[OHM KOHM MOHM]	Set table row (Par1) with resistance value (Par2)
:VOLTage	<NRf>,<NRf>	[MV V]	Set table row (Par1) with voltage value (Par2)
:TIME	<NRf>,<NRf>	[S MS]	Set table row (Par1) with time value (Par2)
:MODE	CONTinuous PULSe,<NRf>		Set continuous or pulse mode
:MODE?			Query waveform mode
:TRIGgered	<Boolean>		Enable/disable waveform triggering
:STATe	<Boolean>		Start/Stop waveform
:STATe?			Query waveform state

Die Geräte der Serie ZS können mit einem beliebigen Lastkurvenverlauf (Programmable Load Cycle) programmiert werden, indem die jeweiligen Einstellzeiten mit den

zugehörigen Einstellwerten in Tabellenform angegeben werden.

Diese Funktion ist in allen Betriebsarten möglich. Die Betriebsart und der

Einstellbereich müssen vor der Programmierung der Sollwerte fest eingestellt werden. Eine Kurvenform kann also nicht im Autorangingbetrieb programmiert werden.

Bei der Ausführung einer PCYCLE-Kurve ist keine synchrone Messung möglich.

Die Indizierung der Tabellenzeile beginnt bei 0.

Der letzte Wert wird durch die zugehörige Dauer=0 (PCYC:TIME x,0) definiert. Das Lastsetting der Zeile x bleibt nach Ende der Kurvenformausgabe statisch gesetzt (s. Bsp. u.).

Der Kurvenformspeicher der Last wird für alle dynamischen Betriebsarten (PCYCLE, LIST, SFUNction) verwendet, d.h. die Programmierung einer LIST-Kurve überschreibt eine evtl. zuvor programmierte andere Kurve. Die Befehle PCYC:STAT ON und LIST:STAT ON starten also gleichberechtigt dieselbe zuletzt programmierte Kurve.

S. auch Subsystem LIST und SFUNction.

**PCYCLE:CURRENT** <0...100>,  
<NRF>

Setzt Strom-Einstellwert <num> an bestimmter Stelle <NRF>.

Es gelten die gleichen Regeln für die Einstellwerte wie im Befehlssystem CURRENT.

The electronic loads can be programmed with any waveform (programmable load cycle) by setting the particular times to the corresponding values in tabular form.

This function is valid in all operating modes. The operating mode and the setting range must be pre-selected before any waveform parameters are programmed. This means that a waveform may be neither programmed nor executed in autoranging mode.

When executing a PCYCLE curve no synchronous measurement will be possible.

The indexing begins at row 0.

The last value is defined by the corresponding duration=0 (PCYC:TIME x,0). The load setting of this row is kept after the waveform generation is stopped (see example below).

See also WAVEform subsystem.

The waveform memory of the load is used for all dynamic modes (PCYCLE, LIST, FUNCTION), that means programming a LIST curve overwrites a previously programmed curve even of another subsystem. The starting commands PCYC:STAT ON and LIST:STAT ON start the last programmed curve with equal rights.

See also subsystem LIST and SFUNction.

**PCYCLE:CURRENT** <0...100>,  
<NRF>

Sets the current setting value <num> at a specified position <NRF>.

There are the same rules for the setting the values as in the command system CURRENT.



**PCYcle:POWer <0...100>, <NRf>**

Setzt Leistungs-Einstellwert <num> an bestimmter Stelle <NRf>.

Es gelten die gleichen Regeln für die Einstellwerte wie im Befehlssystem POWer.

**PCYcle:RESistance <0...100>, <NRf>**

Setzt Widerstands-Einstellwert <num> an bestimmter Stelle <NRf>.

Es gelten die gleichen Regeln für die Einstellwerte wie im Befehlssystem RESistance.

**PCYcle:VOLTage <0...100>, <NRf>**

Setzt Spannungs-Einstellwert <num> an bestimmter Stelle <NRf>.

Es gelten die gleichen Regeln für die Einstellwerte wie im Befehlssystem VOLTage.

**PCYcle:TIME <0...100>, <NRf>**

Setzt Einstelldauer <num> an bestimmter Stelle <NRf>.

Die Einstelldauer ist mindestens 200 $\mu$ s lang, aufgelöst mit 20 $\mu$ s. Eine Ausnahme gilt bei Zeitwert=0: das Ende der Tabelle wird vom Gerät erkannt, sobald der erste Zeitwert=0 auftritt. Der Lastwert an dieser Stelle bleibt statisch eingestellt.

Die Zeiten werden immer in Sekunden programmiert, es sei denn, der Suffix MS (Millisekunden) wird an den Zahlenwert angehängt.

Die längste Einstellzeit beträgt **2147s**.

Beispiel: Programmierung einer Treppenfunktion gemäß folgendem Bild:

**PCYcle:POWer <0...100>, <NRf>**

Sets the power setting value <num> at a specified position <NRf>.

There are the same rules for the setting the values as in the command system POWer.

**PCYcle:RESistance <0...100>, <NRf>**

Sets the resistance setting value <num> at a specified position <NRf>.

There are the same rules for the setting values as in the command system RESistance.

**PCYcle:VOLTage <0...100>, <NRf>**

Sets the voltage setting value <num> at a specified position <NRf>.

There are the same rules for the setting the values as in the command system VOLTage.

**PCYcle:TIME <0...100>, <NRf>**

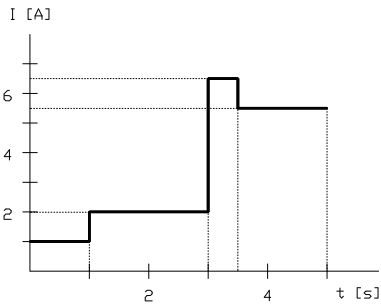
Sets the setting time <num> at a specified position <NRf>.

The time is minimum 200 $\mu$ s long with a resolution of 20 $\mu$ s. An exception is at time=0: the device recognizes the end of the table as soon as a time value = 0 is read. The load value of this row is kept statically.

Times are always programmed in seconds, excepted the suffix MS (milliseconds) is post-fixed to the numeric parameter.

The longest allowed time is **2147s**.

Example: Programming of a step function as shown in the following figure:



Die Tabelle für diese Funktion sieht folgendermaßen aus:

Zeitdauer in sec.	Einstellwert (in A)
1	1
2	2
0.5	6.5
1.5	5.5
0	0

Befehle an el. Last:

```
PCYC:CURR 0,1
PCYC:TIME 0,1
PCYC:CURR 1,2
PCYC:TIME 1,2
PCYC:CURR 2,6.5
PCYC:TIME 2,0.5
PCYC:CURR 3,5.5
PCYC:TIME 3,1.5
PCYC:CURR 4,0
PCYC:TIME 4,0
```

oder:

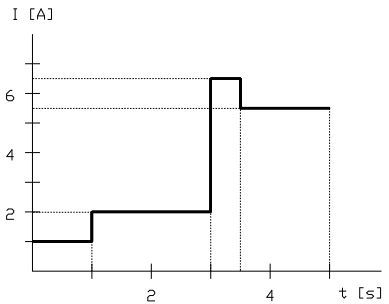
```
PCYC:CURR 0,1;TIME 0,1;CURR 1,2;
TIME 1,2;CURR 2,6.5;TIME 2,0.5;CURR
3,5.5;TIME 3,1.5;CURR 4,0;TIME 4,0
```

### PCYCle:MODE

#### CONTInuous|PULSe,< 0..65535>

Mit diesem Befehl kann man zwischen einer laufenden Wiederholung des programmierten Lastzyklus (CONTInuous) oder einer bestimmten Anzahl von Pulsen (PULSe,0...65535) wählen.

Beispiel s.u.



The table for this function looks like that:

Time in sec.	Setting value (in A)
1	1
2	2
0.5	6.5
1.5	5.5
0	0

Commands to load:

```
PCYC:CURR 0,1
PCYC:TIME 0,1
PCYC:CURR 1,2
PCYC:TIME 1,2
PCYC:CURR 2,6.5
PCYC:TIME 2,0.5
PCYC:CURR 3,5.5
PCYC:TIME 3,1.5
PCYC:CURR 4,0
PCYC:TIME 4,0
```

or:

```
PCYC:CURR 0,1;TIME 0,1;CURR 1,2;
TIME 1,2;CURR 2,6.5;TIME 2,0.5;CURR
3,5.5;TIME 3,1.5;CURR 4,0;TIME 4,0
```

### PCYCle:MODE

#### CONTInuous|PULSe,< 0..65535>

This command allows to select between a continuous repetition of the programmed load cycle (CONTInuous) or a fixed number of cycles (PULSe,0...65535).

Example follows.

**PCYcle:TRIGgered ON|1|OFF|0**

Enablt bzw. disablt den Start der dynamischen Betriebsart durch externen Trigger oder Bustrigger.

Anforderungen an das Triggersignal: siehe SubSystem TRIGger.

**PCYcle:STATe ON|1|OFF|0**

Parameter ON aktiviert die Ausgabe der programmierten Kurve (ohne Trigger). Je nach eingestelltem MODE wird die Kurve dauernd wiederholt oder nur die definierte Anzahl von Pulsen ausgeführt.

**PCYcle:STATe ON**

Beim Starten der Kurvenform leuchtet die Anzeige "Dynamic" an der Frontplatte der Last auf.

Eine im kontinuierlichen Mode gestartete Kurve wird mit Parameter OFF abgebrochen.

**PCYcle:STATe OFF**

Die Anzeige "Static" erscheint an der Frontplatte und die Last stellt den Lastwert ein, der beim Zeitwert=0 steht.

Beispiel 1:

Die vorher programmierte Kurvenform soll kontinuierlich ausgeführt werden, bis das Kommando zum Abbruch kommt. Als statischer Wert wird jetzt 0A gesetzt, weil bei Zeit=0 in Zeile 4 Strom=0 programmiert wurde.

**PCYcle:TRIGgered ON|1|OFF|0**

Enables or disables the Start of the programmed waveform by an external or bus trigger.

Requirements for the trigger signal: see sub system TRIGger.

**PCYcle:STATe ON|1|OFF|0**

The parameter ON activates the output of the programmed curve. Depending on the set MODE the curve will be repeated continuously or the defined number of cycles is executed.

**PCYcle:STATe ON**

At start, the "Dynamic" mode is shown at the front panel.

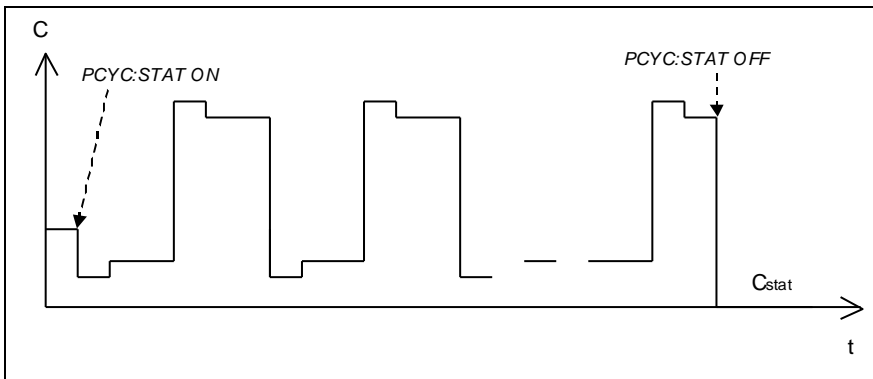
A curve that has been started in the continuous mode, is stopped with the parameter OFF:

**PCYcle:STATe OFF**

"Static" mode is shown at the front panel and the load sets the setting corresponding to time=0.

Example 1:

The programmed curve form shall be executed continuously until the command for stop is sent. Afterwards the load sets the value programmed with time=0 because in row 4 current = 0 is set.

Beispiel 2:

Der vorher programmierte Laststromverlauf soll genau zweimal

ausgeführt werden und dann automatisch stoppen. Danach sollen 8A statisch gesetzt werden.

Befehlsfolge:

```
PCYC:CURR 0,1;TIME 0,1;CURR 1,2;
TIME 1,2;CURR 2,6.5;TIME 2,0.5;CURR
3,5.5;TIME 3,1.5;CURR 4,8.0;TIME 4,0;
MODE PULS,2;STAT ON
```

Außerdem könnte man auch hier die gestartete Kurvenform mit PCYCl:STATe OFF vorzeitig abbrechen.

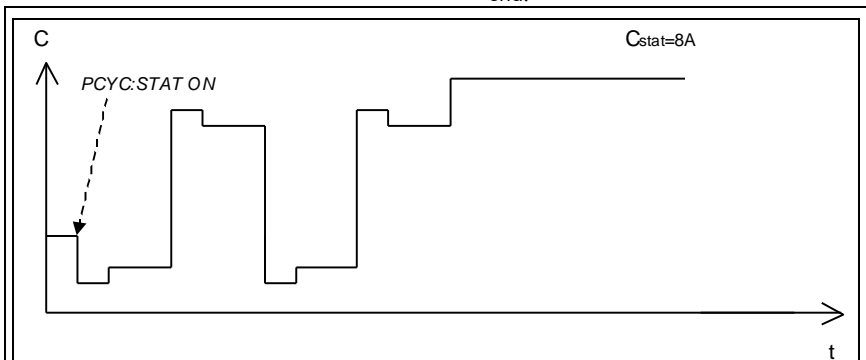
Example 2:

The programmed load current course shall be executed twice and then stopped automatically. After the cycle end 8A shall be statically set.

Commands to ZS:

```
PCYC:CURR 0,1;TIME 0,1;CURR 1,2;
TIME 1,2;CURR 2,6.5;TIME 2,0.5;CURR
3,5.5;TIME 3,1.5;CURR 4,8.0;TIME 4,0;
MODE PULS,2;STAT ON
```

The running waveform could be stopped with PCYCl:STATe OFF before its normal end.



**PCYCl:MODE?**

Abfrage des aktuellen Ausgabemodus der programmierbaren Kurvenform. Liefert als Antwort die Kurzform des entsprechenden Parameters, also entweder *CONT* oder *PULS*.

Beispiel:

*PCYC:MODE?*      Antwort z.B.:  
*CONT*

**PCYCl:STATe?**

Abfrage des aktuellen Ausgabezustandes der programmierbaren Kurvenform.

**PCYCl:MODE?**

Queries the actual output mode of the programmable waveform. Returns as answer the abbreviation of the corresponding parameter, i.e. *CONT* or *PULS*.

Example:

*PCYC:MODE?*      Response e.g.:  
*CONT*

**PCYCl:STATe?**

Queries the actual output mode of the programmable curve form.

Liefert als Antwort die Zahlenform des entsprechenden Parameters, also entweder 0 (für inaktiv) oder 1 (für "Kurve läuft").

Beispiel:

PCYC:STAT?            Antwort z.B.: 0

**Zu beachten:**

Bei der Programmierung einer Kurvenform ist auf folgende Punkte zu achten:

• Einstellintervall für Zeiten

Die Zeiten werden immer in Sekunden programmiert, es sei denn, der Suffix MS (Millisekunden) wird an den Zahlenwert angehängt.

Ein Zeitintervall ist mindestens 200 $\mu$ s lang, mit einer Auflösung von 20 $\mu$ s. Die längste Einstellzeit beträgt **85000s**.

• Erster Wert in der Tabelle

Die Zeilen-Nummerierung der Tabelle beginnt mit Nummer 0 (nicht mit Nummer 1!).

Sind keine Tabellenwerte für Zeile 0 vorhanden, wird die Kurve bei Kommando „PCYC:STAT ON“ nicht ausgeführt.

• Zu wählender Modus und Bereich

Das Gerät wechselt beim Start der Kurvenform nicht automatisch in den richtigen Modus.

Das heißt: Wenn Sie beispielsweise eine VOLTage-Kurve programmiert haben, müssen Sie auch in den VOLT-Modus wechseln, bevor Sie das Kommando PCYCLe:STATe ON geben (und diesen dann auch beibehalten). Das Gerät stellt sonst falsche Einstellwerte ein. Außerdem muss der Einstellbereich vorher gewählt werden. Kurvenformen im Autoranging-Betrieb sind nicht möglich.

Returns as answer the numeric value of the parameter, i.e. 0 (for not active) or 1 (for "curve active").

Example:

PCYC:STAT?            Answer for example:  
0

**Attention:**

For the programming of a waveform the following aspects have to be taken into account:

• Setting interval for time values

The times are programmed in seconds, if the suffix MS (Millisecond) is not appended to the numeric value.

The shortest time interval amounts to 200 $\mu$ s with a resolution of 20 $\mu$ s.

The longest setting time amounts to **85000s**.

• First value in the table

The row numbering in the table begins with 0 (not with 1!).

If there are no table values for row 1, the curve will not be executed for the command "PCYC:STAT ON".

• Selected Mode

When starting the waveform the device does not change automatically into the right mode.

That is, if you have programmed a VOLTage curve, you have to change into the VOLT mode, before you execute the command PCYCLe:STATe ON (and keep it). Otherwise the device will set wrong values. Furthermore, the setting range must be pre-set. Waveforms are not executable in autoranging mode.

Beispiel:

Befehlsfolge für eine Rechteckfunktion im Spannungsbetrieb ( 1s 10V, 1s 1V ) mit statischem Endwert von 5V:

```

MODE:VOLT           Spannungs-
                       betrieb
VOLT:RANG 20       fester Bereich
INP ON              Eingang ein
PCYC:TIME 0,1;     je 1s für beide
TIME 1,1;TIME 2,0 Settings
PCYC:VOLT 0,10;   erster Wert 10V,
VOLT 1,1;VOLT 2,5 zweiter Wert 1V
PCYC:STAT ON      Start      der
                       Rechteckfunktion
.
.
PCYC:STAT OFF     Ende      der
                       Rechteckfunktion,
                       der statische Wert
                       5V      wird
                       eingestellt

```

Example:

Command suite for a square-wave function in constant voltage mode (1s 10V, 1s 1V) with static end value of 5V:

```

MODE:VOLT           Operating Mode
                       Voltage
VOLT:RANG 20       fixed   voltage
                       range
INP ON              Input on
PCYC:TIME 0,1;     1s for each
TIME 1,1;TIME 2,0 setting
PCYC:VOLT 0,10;   first value 10V,
VOLT 1,1;VOLT 2,5 second value 1V
PCYC:STAT ON      Start of the
                       square-wave
                       function
.
.
PCYC:STAT OFF     Stopping the
                       square-wave
                       function, the
                       static value 5V is
                       set.

```

## 7.4.2.12 Subsystem POWER

Befehl	Parameter	Einheit	Bemerkung
Power [:LEVel] [:IMMediate] [:IMMediate]? :MODE  :MODE? :RANGe :RANGe?	<num> [MIN MAX] FIXed LIST    <num> [MIN MAX]	[MW W KW]      [MW W KW]	Konstantleistung einstellen Leistungs-Sollwert abfragen Bei Trigger Liste starten oder stat. Triggerleistung einst. Power Mode abfragen Leistungsbereich einstellen Leistungsbereich abfragen

## 7.4.2.12 Subsystem POWER

Command	Parameter	Unit	Comment
Power [:LEVel] [:IMMediate] [:IMMediate]? :MODE  :MODE? :RANGe :RANGe?	<num> [MIN MAX] FIXed LIST    <num> [MIN MAX]	[MW W KW]      [MW W KW]	Set nominal power Query nominal power At trigger start list or set triggered static power Query power mode Set power range Query power range

Das Befehlssystem POWER dient zur Einstellung und Abfrage des Leistungs-Sollwertes.

Das POWER Subsystem entspricht im wesentlichen dem CURRent Subsystem, jedoch ist die Vorgabe eines triggerbaren Sollwertes nicht möglich.

**Power[:LEVel][:IMMediate] <num>**  
Stellt einen neuen Leistungswert ein. Befindet sich das Gerät im Leistungsbetrieb, wird der neue Wert sofort eingestellt, sofern dieser im gültigen Wertebereich liegt.  
Die Einstellbereiche sind den Technischen Daten des jeweiligen Modells zu entnehmen.

The command system POWER sets and queries the power set point.

The subsystem POWER is almost identical with the subsystem CURRent, however, programming a triggerable setting value is not possible.

**Power[:LEVel][:IMMediate] <num>**  
Sets a new power value. If the device operates in the mode power, the new value will be set immediately, provided that it is contained in the valid range.  
The setting ranges are specified in the technical data of the particular device type.



Bei Überschreitung des zulässigen Wertebereichs wird ein "Data out of range"-Error ausgelöst, der mit `SYSTEM:ERRor?` ausgelesen werden kann.

In diesem Fall wird der maximal mögliche Wert eingestellt und der programmierte Wert als Sollwert gespeichert. Wird später in einen ausreichend hohen Leistungsbereich gewechselt, so stellt die Last den Sollwert ein.

Befindet sich das Gerät nicht im Konstantleistungsbetrieb, wird der neue Einstellwert gespeichert und beim Wechsel in den Leistungsbetrieb (mit `MODE:POWer`) eingestellt.

Als Parameter sind alle Zahlenwerte innerhalb des momentanen Leistungsbereiches erlaubt.

Außerdem sind die speziellen Zahlenparameter MIN und MAX erlaubt.

#### Beispiele:

`POW:LEV 150.23`

`POWer:IMM 0`

`POW MAX`

Als Dezimaltrennzeichen erwartet das Gerät einen Punkt (.), kein Komma!

#### **POWer[:LEVel][:IMMediate]?**

Abfrage des momentan eingestellten Sollwertes im Leistungsbetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

`SD.DDDDDDESDD` S: Sign(Vorzeichen),

D: Digit,

E: Exponent

Der größt- bzw. kleinstmögliche Einstellwert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

When exceeding the allowed scope the error "Data out of range" is triggered and can be read with `SYSTEM:ERRor?`

In this case the maximum possible value is set and the programmed value is kept as nominal value. If a suited power range is set at a later time the load sets the saved nominal value.

If the device doesn't operate with constant power the new setting is saved and set when changing into the operating mode power (using `MODE:POWer`).

Allowed parameters are all numeric values within the power range of the particular device type.

The special numeric parameters MIN and MAX are allowed.

#### Examples:

`POW:LEV 150.23`

`POWer:IMM 0`

`POW MAX`

The device expects a point (.) as decimal separator, no comma!

#### **POWer[:LEVel][:IMMediate]?**

Queries the actual set point of the operating mode power.

The return value is a numerical value in exponent form:

`SD.DDDDDDESDD` S: Sign,

D: Digit,

E: Exponent

The highest or lowest possible setting is determined appending a white space and the parameters MIN or MAX to the question mark.

Als Rückgabewert erhält man den im momentanen Leistungsbereich minimal bzw. maximal möglichen Einstellwert.

Beispiele:

POW? (Antw. z.B.: +1.850000E+01)

POW? MAX

(Antwort von ZS1406: +1.400000E+03)

### POW:MODE FIXEd | LIST

Bestimmt, ob bei einem Triggerereignis die statische Triggerleistung (POW:TRIGgered) oder eine programmierte Leistungsliste (LIST:POWer (Kurvenform)) eingestellt werden soll.

Nach dem Einschalten der Last ist POW:MODE FIXEd gewählt.

S. auch Subsysteme LIST und TRIGger.

Beispiel:

POW:MODE LIST

### POW:MODE?

Leistungs-Trigger-Betriebsart abfragen.

Als Rückgabewert wird die Kurzform des entsprechenden Parameters bereitgestellt (FIX, LIST).

Beispiel:

POW:MODE? (Antwort nach Einschalten: FIX)

### POW:RANGe <num>

Stellt Einstellbereich im Leistungsbetrieb ein.

Im Konstant-Leistungsbetrieb ist keine Autoranging-Funktion verfügbar.

Der numerische Parameter muss innerhalb des größtmöglichen Leistungsbereiches liegen (Technische Daten).

As response the device sends the maximum or minimum possible power setting in the topical power range.

Examples:

POW? (Response for example: +1.850000E+01)

POW? MAX (Response from ZS1406: +1.400000E+03)

### POW:MODE FIXEd | LIST

Determines if the static triggered power (POW:TRIGgered) or a programmed power list (LIST:POWer (waveform)) shall be set when a trigger event occurs.

After power-on POW:MODE FIXEd is set.

Example:

POW:MODE LIST

See also Subsystems LIST and TRIGger.

### POW:MODE?

Query power trigger mode.

The return value is the short form of the corresponding parameters (FIX, LIST).

Example:

POW:MODE? (Response after power-on: FIX)

### POW:RANGe <num>

Sets the setting range in the operating mode power.

There is no autoranging function in constant power mode.

The numeric parameter has to be in the actual power range of the particular device (technical data).

Hinweis:

Die meisten Geräte der ZS Serie verfügen zwar nur über einen Strom- und damit auch einen Leistungsbereich, auf den sich auch alle Genauigkeitsangaben und Messsignale beziehen, zur feineren Einstellung für kleine Werte gibt es jedoch die "Erweiterte Einstellauflösung", was in diesem Programmierhandbuch oft mit Einstellbereich bezeichnet wird.

Die ZS Last versucht, den eingestellten Sollwert aufrechtzuerhalten, d.h. wenn ein höherer Leistungsbereich programmiert wird, wird der letzte Leistungswert nach dem Bereichswchsel wieder eingestellt. Wird ein kleinerer Bereich programmiert und der zuletzt eingestellte Sollwert ist zu groß für den kleineren Bereich, wird die maximal mögliche Leistung eingestellt und ERR2 am Gerät angezeigt. Der Sollwert (POWer?) bleibt unverändert.

Die speziellen Zahlenparameter MIN und MAX sind ebenfalls erlaubt.

Wenn ein Bereichsparameter angegeben wird, der bei dem betreffenden Gerät nicht vorhanden ist, wird der nächsthöhere Leistungsbereich eingestellt.

Beispiele ZS3612:

POWer:RANG 1000 (3600W-Bereich wird eingestellt)

POW:RANGE MAX (10800W-Bereich wird eingestellt)

**POWer:RANGe?**

Abfrage des Leistungsbereiches. Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt.

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Note:

Most ZS series devices have got only one current range and therefore one power range which all accuracy and measurement signals correspond to. Nevertheless, for exact setting of small values there is the "extended resolution" which is called operating range or setting range in this manual because of better understanding.

The ZS load tries to keep the topical nominal value. That means if a higher range is programmed the last setting is kept. If a lower range is programmed and the last setting is too high for this range then the maximum possible value is set and ERR2 is shown on the load's front panel. The nominal value (POWer?) does not change in this case.

The special numeric parameters MIN and MAX are allowed.

If a range parameter is set which is not available at the respective device the next possible higher range is set.

Examples for ZS3612:

POWer:RANG 1000 (set 3600W range)

POW:RANGE MAX (set 10800W range)

**POWer:RANGe?**

Queries the power range. A numeric value in exponent form is returned.

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Beispiele:

POW:RANG?

(Antwort z.B.: +3.600000E+03)

POW:RANGE? MAX

(Antwort von ZS3612: +1.080000E+04)

The highest or lowest possible setting is determined appending a white space and the parameters MIN or MAX to the question mark.

Examples:

POW:RANG?

(Response e.g. +3.600000E+03)

POW:RANGE? MAX

(Resp. from ZS3612: +1.080000E+04)

## 7.4.2.13 Subsystem PROGRAM

Befehl	Parameter	Einheit	Bemerkung
PROGram [:SElEcted] :BEGin :DELeTe [:SElEcted] :ALL :END :NAME :STATe	0 1 2 RUN ON 1		Programm-Mode starten  Aktives Programm löschen Alle Programme löschen Programm-Mode beenden Aktives Programm auswählen Aktives Programm starten

## 7.4.2.13 Subsystem PROGRAM

Command	Parameter	Unit	Comment
PROGram [:SElEcted] :BEGin :DELeTe [:SElEcted] :ALL :END :NAME :STATe	0 1 2 RUN ON 1		Start programming mode  Delete active program Delete all programs End programming mode Select active program Start active program

Das Subsystem PROGRAM dient zur Definition und Ausführung von bis zu drei vordefinierten Programmen.

**Programmdefinition nur möglich bei ausgeschaltetem Lasteingang!**

Dazu wird zunächst mit *PROGram:NAME* die Speichernummer ausgewählt und sodann die Last mit *PROGram:BEGin* in den Programm-definitionsmodus versetzt. Die Last speichert dann alle folgenden Befehle im ausgewählten Speicher, ohne diese auszuführen. Das geschieht so lange, bis der Programm-definitionsmodus mit *PROGram:END* beendet wird.

Im Programmdef.modus kann außer den üblichen Befehlen eine Verzögerung zwischen zwei Befehlen definiert werden (s. Subsystem DELay).

The subsystem PROGRAM is available to define and execute up to three pre-programmed programs.

**Program definition only possible when load input is off!**

For this purpose, first the storage number is selected by the command *PROGram:NAME* and then the load is set to program definition mode by the command *PROGram:BEGin*.

After this, the load saves all following commands in the selected memory without executing the commands. This happens until the program definition mode is finished by the command *PROGram:END*.

In program definition mode, in addition to the usual commands you can define a delay between two commands (see subsystem DELay).

Es gibt drei Programmspeicher (0, 1, 2), die mit *PROG:NAME 0|1|2* aktiviert werden. Speicher 0 wird beim Einschalten der Last automatisch ausgeführt. Ist Speicher 0 nicht programmiert oder gelöscht, stellt die Last die \*RST-Zustände wie bei der Auslieferung ein.

#### Hinweis:

Speicher 1 und 2 stehen erst ab Baureihe D zur Verfügung!

Ein Schachteln von Programmen ist nicht möglich.

In einem Programm dürfen keine Abfragebefehle enthalten sein.

Ein vordefiniertes Programm kann mit *PROG:NAME xx* selektiert und mit *PROG:STATE ON* gestartet werden.

Während das Analog-Interface die Befehle eines Programmes abarbeitet, werden ankommende Befehle ignoriert. D.h. ein laufendes Programm kann auch nicht unterbrochen oder gestoppt werden.

#### **PROG[:SElected]:BEGin**

Startet den Programmdefinitions-Modus.

Nachdem die Last diesen Befehl erhalten hat, speichert sie alle folgenden Befehle in den Speicherblock im nichtflüchtigen Speicher, welcher mit *PROG[:SElected]:NAME* ausgewählt wurde. Die Befehle werden nur gespeichert, nicht ausgeführt!

#### Beispiel:

*PROG:BEG*

#### **PROG[:SElected]:DElete[:SElected]**

#### **PROG[:SElected]:DElete:ALL**

Löscht den aktiven bzw. alle Programmspeicher.

There are three program memories (0, 1, 2) which are activated by *PROG:NAME 0|1|2*. Memory 0 is executed automatically at power-on. If memory 0 is deleted or not programmed the load sets the reset state at power-on as it does at ex-works state.

#### Note:

Memory 1 and 2 are available with production series D of the load.

Nesting several programs is not possible.

A program must not contain any query commands.

A pre-defined program can be selected with *PROG:NAME xx* and started with *PROG:STATE ON*.

While the analog interface is working off the commands of a program any commands sent to the load are ignored. I.e. a running program can not be interrupted or stopped.

#### **PROG[:SElected]:BEGin**

Starts program definition mode.

After the load has received this command, it will save all following commands into the memory block in non-volatile memory which was selected by *PROG[:SElected]:NAME*. The commands are only stored, not executed!

#### Example:

*PROG:BEG*

#### **PROG[:SElected]:DElete[:SElected]**

#### **PROG[:SElected]:DElete:ALL**

Deletes the active program or all program memories, respectively.

Mit `PROGRAM:DELeTe` oder `PROGram:DELeTe:SELeCted` wird der Programmspeicher gelöscht, der zuvor mit `PROGRAM:NAME` selektiert worden ist.

Mit `PROGRAM:DELeTe:ALL` werden alle Programmspeicher (0, 1 und 2) gelöscht.

Beispiel:

`PROG:NAME 1;DEL`  
(Prog.speicher 1 löschen)  
`PROG:DEL:ALL` (alle Programme  
löschen)

**PROGRAM[:SELeCted]:END**

Beendet den Programmdefinitions-Modus.

Die folgenden Kommandos werden nicht mehr im Programmspeicher abgelegt, sondern direkt ausgeführt.

Beispiel:

`PROG:END`

**PROGRAM[:SELeCted]:NAME {0..2}**

Wählt den aktiven Programmplatz aus.

Es sind drei Programmspeicher verfügbar: Speicher 0, 1 und 2. Speicher 1 und 2 sind ab Baureihe D der Last verfügbar.

Wenn ein Speicher mit `PROG:NAME x` selektiert worden ist, bleibt dieser solange selektiert, bis ein anderer Speicherplatz mit `PROG:NAME y` selektiert wird.

Speicher 0 nimmt eine Sonderstellung ein: er wird automatisch beim Einschalten der Last ausgeführt. So kann man z.B. Lasteinstellungen, die sich nie verändern, in Speicher 0 ablegen. Diese sind nach dem Aus- und Einschalten oder nach einem Netzausfall sofort wieder verfügbar, ohne einen PC anschließen zu müssen.

The command `PROGRAM:DELeTe` or `PROGram:DELeTe:SELeCted` deletes the program which has been selected by `PROGRAM:NAME`.

`PROGram:DELeTe:ALL` deletes all program memories (0, 1, and 2) simultaneously.

Example:

`PROG:NAME 1;DEL`  
(delete program memory 1)  
`PROG:DEL:ALL` (delete all  
programs)

**PROGRAM[:SELeCted]:END**

Exits the program definition mode.

The following commands won't be saved in the program storage but are directly executed.

Example:

`PROG:END`

**PROGRAM[:SELeCted]:NAME {0..2}**

Selects the active program storage.

There are three program storages: storage 0, 1, and 2. Storage 1 and 2 are available with production series D (and higher) of the load.

If a storage has been selected with `PROG:NAME x` this storage stays selected until a different storage is selected with `PROG:NAME y`.

Storage 0 has got a special meaning: it is automatically executed at power-on of the load. For example, you can store load settings which will always be the same in storage 0. These settings will automatically be available after power-on or after a mains failure without having to connect a computer to make the settings.

Speicher 1 und 2 sind ebenfalls nichtflüchtig. Diese müssen jedoch mit `PROG:NAME` selektiert und mit `PROG:STATe ON` explizit gestartet werden (s.u.).

Beispiel:

```
PROG:NAME 1
```

**PROGRAM[:SElected]:STATE  
1|ON|RUN**

Startet das aktive Programm.

Da ein gestartetes Programm nicht gestoppt werden kann, gibt es für diesen Befehl nur startende Parameter.

Während die Last ein gespeichertes Programm abarbeitet, werden ankommende Befehle ignoriert. Daher kann der `PROG:STATe` auch nicht abgefragt werden.

Examples:

```
PROG:STAT 1
PROG:STATe ON
PROG:STAT RUN
```

Beispiel 1: Programm definieren und auf Speicherplatz 1 speichern, das den Lasteingang einschaltet und 200ms später 20A einstellt.

```
PROG:NAME 1
PROG:BEG
CURR:RANG 20;LEV 0
INP ON;DEL 0.2;CURR 20
PROG:END
```

Zu einem beliebigen Zeitpunkt kann der Ablauf gestartet werden:  
`PROG:NAME 1;STAT ON`

Storage 1 and 2 are also non-volatile. But these must be selected with `PROG:NAME` and explicitly started with `PROG:STATe ON` (see below).

Example:

```
PROG:NAME 1
```

**PROGRAM[:SElected]:STATE  
1|ON|RUN**

Starts the active program.

Since a running program cannot be stopped, there are only starting parameters for this command.

While the load is working off a saved program it ignores arriving commands. Therefore `PROG:STATe` cannot be queried.

Examples:

```
PROG:STAT 1
PROG:STATe ON
PROG:STAT RUN
```

Example 1: Define and store in memory 1 a program which switches the load input on and 200ms later sets 20A.

```
PROG:NAME 1
PROG:BEG
CURR:RANG 20;LEV 0
INP ON;DEL 0.2;CURR 20
PROG:END
```

The process may be started at any time:  
`PROG:NAME 1;STAT ON`



## 7.4.2.14 Subsystem RESistance

Befehl	Parameter	Einheit	Bemerkung
RESistance [:LEVe] [:IMMediate]	<num>	[OHM KOHM   MOHM]	Konstantwiderstand einstellen
[:IMMediate] ? :TRIGgered	[MIN MAX] <NRf>	[OHM KOHM   MOHM]	Widerstands-Sollwert abr. getriggerten Widerstand einstellen
:TRIGgered? :MODE	[MIN MAX]		Widerstand-Triggerwert abfragen Bei Trigger Liste starten oder statischen Triggerwrdst. einst. Resistance Mode abfragen
:MODE?			
:RANGe	<num>	[OHM KOHM   MOHM]	Widerstandsbereich einstellen
:AUTO :RANGe?	<Boolean> [MIN MAX]		Autoranging ein aus Widerstandsbereich abr.

## 7.4.2.14 Subsystem RESistance

Command	Parameter	Unit	Comment
RESistance [:LEVe] [:IMMediate]	<num>	[OHM KOHM   MOHM]	Set nominal resistance
[:IMMediate] ? :TRIGgered	[MIN MAX] <NRf>	[OHM KOHM   MOHM]	Query nominal resistance Set triggered resistance
:TRIGgered? :MODE	[MIN MAX]		Query triggered resistance Start list or set triggered static resistance at trigger
:MODE?			Query resistance mode
:RANGe	<num>	[OHM KOHM   MOHM]	Set resistance range
:AUTO :RANGe?	<Boolean> [MIN MAX]		Autoranging on off Query resistance range

Das Befehlssystem RESistance dient zur Einstellung und Abfrage des Widerstands-Sollwertes. Es entspricht im wesentlichen dem Subsystem CURRent.

The command system RESistance sets and queries the resistance set point. It is almost identical with the subsystem CURRent.

**RESistance[:LEVel][:IMMediate]****<num>**

Stellt einen neuen Widerstandswert ein. Befindet sich das Gerät im Widerstandsbetrieb, wird der neue Wert sofort eingestellt, sofern dieser im gültigen Wertebereich liegt.

Die Einstellbereiche sind den Technischen Daten des jeweiligen Modells zu entnehmen.

Bei Überschreitung des zulässigen Wertebereichs wird ein "Data out of range"-Error ausgelöst, der mit *SYSTEM:ERRor?* ausgelesen werden kann. In diesem Fall wird der maximal mögliche Wert eingestellt und der programmierte Wert als Sollwert gespeichert. Wird später in einen ausreichend hohen Widerstandsbereich gewechselt, so stellt die Last den Sollwert ein.

Befindet sich das Gerät nicht im Konstantwiderstandsbetrieb, wird der neue Einstellwert gespeichert und beim Wechsel in den Widerstandsbetrieb (mit *MODE:RESistance*) eingestellt.

Als Parameter sind alle Zahlenwerte innerhalb des momentanen Widerstandsbereiches erlaubt.

Außerdem sind die speziellen Zahlenparameter MIN und MAX erlaubt. Im Autoranging-Betrieb stellt der Parameter MAX den Endwert im höchsten Widerstandsbereich ein, ansonsten den Endwert im momentanen Bereich.

Beispiele:*RES:LEV 15.23**RESistance:IMM 0**RES MAX*

Als Dezimaltrennzeichen erwartet das Gerät einen Punkt (.), kein Komma!

**RESistance[:LEVel][:IMMediate]****<num>**

Sets a new resistance. If the device operates in the mode resistance, the new value will be set immediately, provided that it is contained in the valid scope.

The setting ranges are specified in the technical data of the particular device.

When exceeding the valid scope the error "Data out of range" is triggered and can be read with *SYSTEM:ERRor?* In this case the maximum possible value is set and the programmed value is kept as nominal value. If a suited resistance range is set at a later time the load sets the saved nominal value.

If the device is not operating in constant resistance mode, the new setting value is saved and set when changing into the operating mode resistance (using *MODE:RESistance*).

All numeric values within the actual resistance range of the particular device are allowed as parameters.

The special numeric parameters MIN and MAX are also allowed.

In autoranging mode the parameter MAX sets the end value in the highest resistance range, otherwise the end value in the active range.

Examples:*RES:LEV 15.23**RESistance:IMM 0**RES MAX*

The device expects a point (.) as decimal separator, no comma!

**RESistance[:LEVel][:IMMediate]?**

Abfrage des momentan eingestellten Sollwertes im Widerstandsbetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

*SD.DDDDDDESDD* S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Einstellwert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Als Rückgabewert erhält man den im momentanen Widerstandsbereich minimal bzw. maximal möglichen Einstellwert.

Beispiele:

*RES?* (Antwort z.B.: +1.850000E+01)

*RES? MAX*

(Antwort z.B.: +1.780000E+01)

**RESistance[:LEVel]:TRIGgered**

**<NRF>**

Stellt einen neuen Wert für den getriggerten Widerstand ein.

S. auch RESistance:RANGe und SubSystem TRIGger.

Bei Überschreitung des zulässigen Wertebereichs wird ein "Data out of range"-Error ausgelöst, der mit *SYSTEM:ERRor?* ausgelesen werden kann. In diesem Fall wird der maximal mögliche Wert eingestellt und der programmierte Wert als Sollwert gespeichert. Wird später in einen ausreichend hohen Widerstandsbereich gewechselt, so stellt die Last den Sollwert ein.

Das Triggerereignis wird mit dem Befehl *TRIGger:SOURce* definiert.

**RESistance[:LEVel][IMMediate]?**

Queries the actual set point in the operating mode resistance.

A numeric value in exponent form is returned:

*SD.DDDDDDESDD* S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible setting is determined appending a white space and the parameters MIN or MAX to the question mark.

As response the minimum or maximum resistance setting in the topical range is sent.

Example:

*RES?* (Resp. e.g.: +1.850000E+01)

*RES? MAX*

(Resp. e.g.: +1.780000E+01)

**RESistance[:LEVel]:TRIGgered**

**<NRF>**

Sets a new value for the triggered resistance.

See also RESistance:RANGe and SubSystem TRIGger.

When exceeding the allowed scope the error "Data out of range" is triggered and can be read with *SYSTEM:ERRor?* In this case the maximum possible value is set and the programmed value is kept as nominal value. If a suited current range is set at a later time the load sets the saved nominal value.

The trigger event is defined using the command *TRIGger:SOURce*.

Tritt das Triggerereignis ein und Konstantwiderstandsbetrieb ist eingestellt, stellt das Gerät den vorher programmierten Trigger-Widerstand ein, wenn nicht eine dynamische Betriebsart triggerfähig gemacht wurde (s. SubSystem TRIGger).

Als Parameter sind alle Zahlenwerte innerhalb des momentanen Widerstandsbereiches erlaubt.

Beispiele: bei Trigger 10Ω einstellen  
RES:TRIG 10.0  
RESISTANCE:LEVEL:TRIGGERED 1.0E1

Als Dezimaltrennzeichen erwartet das Gerät einen Punkt (.), kein Komma!

### RESistance[:LEVEl][TRIGgered]?

Abfrage des triggerbaren Sollwertes im Widerstandsbetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert im momentanen Widerstandsbereich wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Beispiel:  
RES:TRIG? MIN  
(Antw.: +0.000000E+00)

If the trigger event happens and the operating mode constant resistance is set, the device sets the programmed trigger resistance unless a dynamic function is waiting for a trigger (see subsystem TRIGger).

All numeric values within the resistance range of the particular device type are allowed as parameters.

Examples: set 10Ω at trigger  
RES:TRIG 10.0  
RESISTANCE:LEVEL:TRIGGERED 1.0E1

The device expects a point (.) as decimal separator, no comma!

### RESistance[:LEVEl][TRIGgered]?

Queries the triggerable set point in the operating mode resistance.

A numeric value in exponent form is returned:

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible setting is determined appending a white space and the parameters MIN or MAX to the question mark.

Example:  
RES:TRIG? MIN (Response:  
+1.500000E+01)

**RESistance:MODE FIXEd | LIST**

Bestimmt, ob bei einem Triggerereignis der statische Triggersiderstand (RESistance:TRIGgered) oder eine programmierte Widerstandsliste (LIST:RESistance (Kurvenform)) eingestellt werden soll.

Nach dem Einschalten der Last ist RESistance:MODE FIXEd gewählt.

S. auch Subsysteme LIST und TRIGger.

Beispiel:

RES:MODE LIST

**RESistance:MODE?**

Widerstands-Trigger-Betriebsart abfragen.

Als Rückgabewert wird die Kurzform des entsprechenden Parameters bereitgestellt (FIX, LIST).

Beispiel:

RES:MODE? (Antwort nach Einschalten: FIX)

**RESistance:RANGe <num>**

Stellt Einstellbereich im Widerstandsbetrieb ein.

Der numerische Parameter muss innerhalb des größtmöglichen Widerstandsbereiches des jeweiligen Modells liegen (Technische Daten).

Hinweis:

Die meisten Geräte der ZS Serie verfügen zwar nur über einen Strom- und damit auch einen Widerstandsbereich, auf den sich auch alle Genauigkeitsangaben und Messsignale beziehen, zur feineren Einstellung gibt es jedoch die "Erweiterte Einstellauflösung", was in diesem Programmierhandbuch oft mit Einstellbereich bezeichnet wird.

**RESistance:MODE FIXEd | LIST**

Determines if the static trigger resistance (RESistance:TRIGgered) or a programmed resistance list (LIST:RESistance (waveform)) shall be set when a trigger event occurs.

After power-on RESistance:MODE FIXEd is set.

Example:

RESistance:MODE LIST

See also Subsystems LIST and TRIGger.

**RESistance:MODE?**

Query resistance trigger mode.

The return value is the short form of the corresponding parameters (FIX, LIST).

Example:

RES:MODE? (Response after power-on: FIX)

**RESistance:RANGe <num>**

Sets the setting range in the operating mode resistance.

The numeric parameter must be within the resistance range of the particular device type (technical data).

Note:

Most ZS series devices have got only one current range and therefore one resistance range which all accuracy and measurement signals correspond to. Nevertheless, for exact setting there is the "extended resolution" which is called operating range or setting range in this manual because of better understanding.

Die ZS Last versucht, den eingestellten Sollwert aufrechtzuerhalten, d.h. wenn ein höherer Widerstandsbereich programmiert wird, wird der letzte Widerstandswert nach dem Bereichswechsel wieder eingestellt.

Wird ein kleinerer Bereich programmiert und der zuletzt eingestellte Widerstand ist zu groß für den kleineren Bereich, wird der maximal mögliche Wert eingestellt und ERR2 am Gerät angezeigt. Der Sollwert (RESistance?) bleibt unverändert. Die speziellen Zahlenparameter MIN und MAX sind ebenfalls erlaubt.

Wenn ein Bereichsparameter angegeben wird, der bei dem betreffenden Gerät nicht vorhanden ist, wird der nächsthöhere Widerstandsbereich eingestellt.

Beispiele:

RESistance:RANG 10

RES:RANGE MAX

**RESistance:RANGe:AUTO**

**ON|1|OFF|0**

Schaltet den Autoranging-Modus im Widerstandsbetrieb ein oder aus.

Die Parameter ON oder 1 schalten Autoranging ein, die Parameter OFF oder 0 schalten Autoranging aus, und es muss der gewünschte Widerstandsbereich gewählt werden.

Im Autoranging-Betrieb stellt das Gerät je nach programmiertem Widerstandswert den optimalen Widerstandsbereich ein.

Beim Einschalten des Gerätes ist im Widerstandsbetrieb die Autoranging-Funktion aktiviert.

Beispiel:

RES:RANG:AUTO ON

The ZS load tries to keep the topical nominal value. That means if a higher range is programmed the last setting is kept.

If a lower range is programmed and the last setting is too high for this range then the maximum possible value is set and ERR2 is shown on the load's front panel. The nominal value (RESistance?) does not change in this case.

The special numeric parameters MIN and MAX are allowed.

If a range parameter is set which is not available at the respective device the next possible higher range is set.

Examples:

RES:RANG 10

RES:RANGE MAX

**RESistance:RANGe:AUTO**

**ON|1|OFF|0**

Switches autoranging in constant resistance mode on or off.

The parameter ON or 1 switches autoranging on, the parameter OFF or 0 switch autoranging off and the desired current range has to be programmed.

In autoranging mode due to the programmed setting value the device sets the optimum range.

At power-on the autoranging function in resistance mode is activated.

Example:

RES:RANG:AUTO ON

Achtung!

Autoranging ist nicht erlaubt bei einer der folgenden Betriebsarten:

- wenn ein Widerstands-Triggerwert gesetzt wird
- bei der Definition von Modulationsparametern (TRANSient-Subsystem)
- bei der Definition von frei programmierbaren Kurvenformen (PCYCLE-Subsystem)

In diesen Fällen muss vorher ein fester Widerstandsbereich gewählt werden.

**RESistance:RANGe?**

Abfrage des momentanen Widerstandsbereiches.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt.

*SD.DDDDDDESDD* S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert im momentanen Widerstandsbereich wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Beispiele:

*RES:RANG?*

*RES:RANG? MAX*

Important!

Autoranging is not allowed in one of the following items:

- if a resistance trigger value is set
- when modulation parameters are defined (TRANSient Subsystem)
- when programmable load cycles are defined (PCYCLE Subsystem)

In these cases you have to pre-set a fixed resistance range.

**RESistance:RANGe?**

Queries the resistance range.

A numeric value in exponent form is returned.

*SD.DDDDDDESDD* S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible setting is determined by appending a white space and the parameters MIN or MAX to the question mark.

Examples:

*RES:RANG?*

*RES:RANG? MAX*

**7.4.2.15 Subsystem SETup**

Befehl	Parameter	Einheit	Bemerkung
SETup :ADC :ADC? SETup?	SLOW FAST		AD-Wandler selektieren Aktiven AD-Wandler abfragen Geräte-Einstellungen abfragen

**7.4.2.15 Subsystem SETup**

Command	Parameter	Unit	Comment
SETup :ADC  :ADC? SETup?	SLOW FAST		Activate slow or fast ADC (def. slow) Query active ADC Query device parameters

Das Subsystem SETup bedient im wesentlichen geräteinterne Funktionen wie die Auswahl des AD-Wandlers für Messungen.

**SETup:ADC SLOW|FAST**

Wählt den aktiven ADC zur Messwerterfassung. Nur möglich mit Option ZS13.

Die Elektronischen Lasten der Serie ZS können optional mit einem zweiten, schnelleren AD-Wandler (ADC) bestückt werden (Option ZS13). Um Messwerte auch mit diesem schnelleren ADC zu erfassen, muss dieser erst aktiviert werden (Parameter FAST). Der schnelle ADC kann außerdem auch eine Externspannung messen (s. Subsystem MEASure). Beim Einschalten des Gerätes und nach Reset wird defaultmäßig immer der langsame (höherauflösende) ADC aktiviert.

Soll also bei einem Gerät mit der Option "Schneller ADC" grundsätzlich dieser für Messungen eingesetzt werden, muss beim Start eines entsprechenden Steuerprogrammes immer der Befehl SET:ADC FAST an die Last gesendet werden.

In general, the subsystem SETup handles device-internal functions such as the selection of the A/D converter for measurements.

**SETup:ADC SLOW|FAST**

Selects the active ADC for measurements. Only possible with option ZS13.

The electronic loads of ZS series can be optionally (Option ZS13) equipped with an additional fast A/D converter (ADC). To acquire measurements with this ADC it must be pre-selected (parameter FAST). With the fast ADC it is also possible to measure an external voltage (see subsystem MEASure). At power-on and after reset the slow (higher-resolved) ADC is selected as default.

That means if the fast ADC shall be used always with a device including the optional fast ADC then the controlling program has to send the command SET:ADC FAST to the load at the beginning of the test procedure.



Beispiele:

```
SET:ADC FAST
SET:ADC SLOW
```

**SETup:ADC?**

Abfrage des momentan aktiven ADC.

Als Rückgabewert wird der momentan aktive Parameter gesendet.

Beispiel:

```
SET:ADC?           (Antwort z.B.:
                   SLOW)
```

**SETup?**

Abfrage der Geräteeinstellungen.

Die Last antwortet mit einem String, der aus folgenden Informationen besteht: Unteradresse, Strombereiche, Spannungsbereiche, Widerstandsbereiche, Leistungsbereiche.

Antwortsyntax (hier bei einem Einbereichsgerät mit erweiterter Einstellung):

```
=A:a,C1:c1,C2:c2,V1:v1,V2:v2,R1:r1,R2:r2,P1:p1,P2:p2;
```

wobei:

*a* Unteradresse  
*c1* kleiner Strombereich mit erweiterter Einstellauflösung  
*c2* Strombereich  
*v1* kleiner Spannungsbereich mit erweiterter Einstellauflösung  
*v2* Spannungsbereich  
*r1* Widerstandsbereich  
*r2* kleiner Widerstandsbereich mit erweiterter Einstellauflösung  
*p1* kleiner Leistungsbereich mit erweiterter Einstellauflösung  
*p2* Leistungsbereich

Der Setstring ist mit einem Semikolon (;) abgeschlossen.

Examples:

```
SET:ADC FAST
SET:ADC SLOW
```

**SETup:ADC?**

Queries the currently active ADC.

The parameter string of the currently active ADC is returned.

Example:

```
SET:ADC?           (Response e.g.:
                   SLOW)
```

**SETup?**

Queries the device settings.

The load responds with a string containing the following information: Sub address, current ranges, voltage ranges, resistance ranges, power ranges.

Response syntax (here at a single-range device with extended setting resolution):  
 =A:a,C1:c1,C2:c2,V1:v1,V2:v2,R1:r1,R2:r2,P1:p1,P2:p2;

where:

*a* Sub address  
*c1* Low current range with extended resolution  
*c2* Current range  
*v1* Low voltage range with extended resolution  
*v2* Voltage range  
*r1* Resistance range  
*r2* Low resistance range with extended resolution  
*p1* Low power range with extended resolution  
*p2* Power range

The set string is terminated by a semicolon (;).

Beispiel:

SET? Antwort ZS1406:

```
=A:1,C1:50.0000,C2:150.0000,V  
1:20.0000,V2:60.0000,R1:13.300  
0,R2:4.4300,P1:4200.0000,P2:14  
00.0000;
```

Bei einem Mehrbereichsgerät werden entsprechend der Anzahl der Einstellbereiche alle Strombereiche von C1 bis C4 hintereinander ausgegeben, gefolgt von Spannungs-, Widerstands- und Leistungsbereichen.

Example:

SET? Response ZS1406:

```
=A:1,C1:50.0000,C2:150.0000,V  
1:20.0000,V2:60.0000,R1:13.300  
0,R2:4.4300,P1:4200.0000,P2:14  
00.0000;
```

A multi-range device will output its corresponding current ranges (C1 to C4) in ascending order, followed by voltage, resistance and power ranges.

7.4.2.16 Subsystem SFUNction

Befehl	Parameter	Einheit	Bemerkung
SFUNction			
:BATTery			
:ENABLE	<Boolean>		Batterieprüffunktion enabeln/disablen
:ENABLE?			Enablezustand abfragen
:STATE?			Aktivzustand abfragen
:TEVoltage	<NRf>		Testendspannung der Batterie def.
:TEVoltage?			Testendspannungs-Setting abfr.
:EXPonential	<NRf>, <NRf>, <NRf>		Parameter für Exponentialfunktion definieren
:ENABLE	<Boolean>		Exp.funktion enablen/disablen
:ENABLE?			Enablezustand abfragen
:STATE?			Aktivzustand abfragen

7.4.2.16 Subsystem SFUNction

Command	Parameters	Unit	Comment
SFUNction			
:BATTery			
:ENABLE	<Boolean>		Enable/disable battery test function
:ENABLE?			Query enable state
:STATE?			Query active state
:TEVoltage	<NRf>		Define battery's test-end voltage
:TEVoltage?			Query TEV setting
:EXPonential	<NRf>, <NRf>, <NRf>		Define parameters for exponential function
:ENABLE	<Boolean>		Enable/disable exp. function
:ENABLE?			Query enable state
:STATE?			Query active state

Das Subsystem SFUNction (Special Function) wurde geschaffen, um besondere Funktionen der Elektronischen Last, wie z.B. eine automatische Batterieprüffunktion oder die Generierung eines Einschaltstrompulses, bedienen und parametrisieren zu können.

The SFUNction (Special Function) Subsystem was created to provide special functions of the electronic load, such as an automatic battery test function or the generation of a turn-on current peak.

## A) Batterieprüf-Funktion

Die Batterieprüffunktion stellt ein einfaches Mittel zur Kapazitäts- bzw. Energieermittlung von Batterien, Akkus etc. zur Verfügung.

Dazu wird die Entladeschlussspannung des Prüflings programmiert (*SFUN:BATT:TEV*) und die Prüffunktion mit *SFUN:BATT:ENAB ON* (s.u.) erlaubt. Sobald der Lasteingang eingeschaltet wird, beginnt die Last, die Ladung (Ah) und die Energie (Wh) zu kumulieren, bis die Eingangsspannung unter die programmierte Endspannung fällt oder der Lasteingang ausgeschaltet wird. Die Last schaltet dann automatisch den Eingang aus. Kapazität und Energie können jederzeit mit den Befehlen *MEASure:CHARge?* und *MEASure:ENERgy?* gelesen werden. Bei erneutem Einschalten des Lasteinganges starten die Ah-/Wh-Kumulatoren wieder bei 0.

Um Spannung und Strom mit entsprechendem Zeitstempel von der Last erfassen und speichern zu lassen, kann der interne Timer verwendet werden (s. Subsystem *TRIGger*).

### **SFUNction:BATTery:ENABLE ON|OFF|1|0**

Erlaubt oder stoppt die Ausführung der Batterieprüffunktion.

Der Parameter ON bzw. 1 erlaubt den Start (der Start selbst erfolgt erst durch Einschalten des Lasteingangs). Der Parameter OFF bzw. 0 verhindert die Funktion, d.h. wenn der Lasteingang eingeschaltet wird, werden Ah und Wh nicht kumuliert.

Beispiel:

*SFUN:BATT:ENAB ON*

## A) Battery Test Function

The battery test function is a simple way to determine the capacity or energy, respectively, of batteries, accumulators etc.

For this purpose, the test-end voltage of the DUT (*SFUN:BATT:TEV*) must be programmed and the battery test function must be enabled (*SFUN:BATT:ENAB ON*). As soon as the load input is switched on the load starts accumulating the charge (Ah) and the energy (Wh) until the input voltage decreases below the programmed test-end voltage. Then the load automatically switches the input off. The capacity and energy may be read any time using the commands *MEASure:CHARge?* and *MEASure:ENERgy?*.

When the load input is reactivated the Ah/Wh cumulators restart with 0.

To acquire and save voltage and current with the corresponding timestamp the internal timer can be used (see subsystem *TRIGger*).

### **SFUNction:BATTery:ENABLE ON|OFF|1|0**

Enables or disables the battery test function.

The parameter ON or 1 enables the start (the start itself is done by switching the input on). The parameter OFF or 0 disables the function, that means Ah and Wh are not accumulated when the input is switched on.

Example:

*SFUN:BATT:ENAB ON*

**SFUNCTION:BATTERY:ENABLE?**

Abfrage des Aktivierungszustandes der Batterieprüf-Funktion.

Als Rückgabewert wird eine 1 bereitgestellt, wenn die Ausführung der Funktion freigegeben ist.

Bei deaktivierter Ausführung der Batterietest-Funktion wird mit 0 geantwortet.

Beispiel:

*SFUN:BATT:ENAB?* (Antwort bei freigegebener Ausführung: 1)

**SFUNCTION:BATTERY:STATE?**

Abfrage des Ausführungszustandes der Batterietest-Funktion.

Als Rückgabewert wird eine 1 bereitgestellt, wenn die Batterietest-Funktion gerade ausgeführt wird.

Mit 0 wird dann geantwortet, wenn die Funktion momentan nicht ausgeführt wird.

Beispiel:

*SFUN:BATT:STAT?* (Antwort bei laufender Ausführung: 1)

**SFUNCTION:BATTERY:TEVoltage****<NRf>**

Definiert die Entladeschlussspannung (Test End Voltage) in V für die Batterietest-Funktion.

Die Endspannung ist mit 0 vorbelegt und kann auch bei laufender Batterietestfunktion verändert werden.

Beispiel:

*SFUN:BATT:TEV 1.2020*

**SFUNCTION:BATTERY:ENABLE?**

Queries the activation state of the battery test function.

The load responds with 1 if the execution of the battery function is enabled.

At disabled execution the load responds with 0.

Example:

*SFUN:BATT:ENAB?* (Response at enabled execution: 1)

**SFUNCTION:BATTERY:STATE?**

Queries whether the programmed battery test function is running.

The load returns 1 if the function is currently running.

It returns 0 if the battery test function is not running.

Example:

*SFUN:BATT:STAT?* (Response at enabled execution: 1)

**SFUNCTION:BATTERY:TEVoltage****<NRf>**

Defines the test end voltage in volts for the battery test function.

The end voltage is pre-defined with 0 and may be changed while the battery test function is running.

Example:

*SFUN:BATT:TEV 1.2020*

**SFUNction:BATTery:TEVoltage?**

Abfrage der programmierten Entladeschlussspannung. Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt.

Beispiel:

SFUN:BATT:TEV?

(Antwort z.B.: +1.202000E+00)

Beispiel Befehlsfolge:

CURR:RANG 20;LEV 10

SFUN:BATT:TEV 1.25;ENAB ON

INP ON

...

SFUN:BATT:STAT?

MEAS:CHAR?

**B) Exponentialfunktion**

Diese Funktion ist nur verfügbar in Verbindung mit der Option "Data Acquisition Tool" (ZS13). Außerdem muss der "schnelle ADC" (siehe Subsystem Setup) ausgewählt werden. Mit der Exponentialfunktion kann das Einschaltverhalten von Prüflingen untersucht werden. Dazu wird mit dem folgenden Befehl der Parametersatz der E-Funktion definiert.

**SFUNction:EXponential <Peak>, <End>, <Tau>, <Dwell>, <SampleRate>**

Definiert die Parameter für eine exponentiell abfallende Lastspitze.

<Peak>, <End>, <Tau>

Eine Exponentialfunktion kann prinzipiell in allen Betriebsarten (CC, CR, CV, CP) programmiert und ausgeführt werden. Die Last muss vor der Parametrierung in der entsprechenden Betriebsart sein und der Einstellbereich muss fest eingestellt sein (kein Autoranging!).

**SFUNction:BATTery:TEVoltage?**

Queries the programmed value of the test end voltage. A numeric value in exponent form is returned.

Example:

SFUN:BATT:TEV?

(Response e.g.: +1.202000E+00)

Example Command Set:

CURR:RANG 20;LEV 10

SFUN:BATT:TEV 1.25;ENAB ON

INP ON

...

SFUN:BATT:STAT?

MEAS:CHAR?

**B) Exponential Function**

This function is only available with the "Data Acquisition Tool" Option (ZS13). Additionally the "fast ADC" (see Subsystem Setup) has to be chosen. With the exponential function you can investigate the turn-on characteristics of UUTs. To define the e-function the parameters of the following command are used.

**SFUNction:EXponential <Peak>, <End>, <Tau>, <Dwell>, <SampleRate>**

Defines the parameters of an exponential decreasing peak load.

<Peak>, <End>, <Tau>

An exponential function can in principle be executed in all operating modes (CC, CR, CV, CP). The electronic load must be in the corresponding mode before the e-function's parameters are programmed, furthermore, a fixed range must be set (no autoranging!).

Die Parameter <Peak> und <End> geben Settingwerte für die entsprechende Betriebsart an, werden jedoch einheitslos programmiert.

Der Parameter <Tau> ist die Zeit, nach welcher der Spitzenwert um 63% im Bezug auf den Endwert abgesunken ist.

Es wird folgende Funktion zugrundegelegt:

$$\text{Peak} \cdot e^{(-t/\tau)}$$

Soll die E-Funktion nicht gegen 0, sondern gegen einen von Null verschiedenen Endwert laufen, wird das durch einen von Null verschiedenen Parameter <End> definiert. Damit ergibt sich ein Funktionsverlauf nach folgender Formel:

$$(\text{Peak}-\text{End}) \cdot e^{(-t/\tau)} + \text{End}$$

Zum Starten der E-Funktion muss eine Triggerspannung programmiert werden (s. *VOLTage:PROTection*). Sobald die Eingangsspannung der Last diese Triggerspannung überschreitet, startet die programmierte Lastkurve. Nach  $8 \tau$  (Tau) gilt die Kurve als ausgeführt und der Endwert bleibt statisch eingestellt, bis entweder ein erneutes Triggerereignis stattfindet oder ein neuer statischer Sollwert programmiert wird.

The parameters <Peak> and <End> define the settings in the corresponding operating mode and must be programmed without any unit.

The parameter <Tau> is the time after which the peak value has decreased to 63% relating to the end value.

The function bases on the following:

$$\text{Peak} \cdot e^{(-t/\tau)}$$

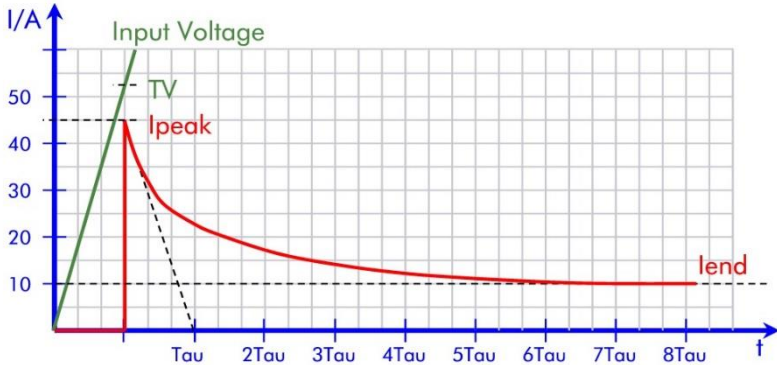
If the e-function shall not run to 0 at the end but to a value not equal to 0 you can define this value by programming the parameter <End>. This causes a function behaving like the following formula:

$$(\text{Peak}-\text{End}) \cdot e^{(-t/\tau)} + \text{End}$$

To start the generation of the e-function a trigger voltage must be programmed (see *VOLTage:PROTection*). As soon as the input voltage becomes higher than this trigger voltage the programmed load peak is started. After  $8 \tau$  (Tau) the function is fully executed and the end value is statically kept until either a new trigger event occurs or a new static value will be programmed.

**Beispiel:** Strompuls mit 45A Spitzwert (Peak=45), 10A Endwert (End=10) und  $\tau=15\text{ms}$  (Spitzenverweildauer und SampleRate =0):  
SFUN:EXP 45,10,0.015,0,0  
Programmierbeispiel mit TV s.u.

**Example:** Current pulse with 45A peak, 10A end value and  $\tau=15\text{ms}$  (peak dwell and sample rate =0):  
SFUN:EXP 45,10,0.015,0,0  
Programming Example with TV see below.



#### <DWELL>

Da jede Elektronische Last eine gewisse Anstiegszeit hat, kann es sein, dass ein hoher Spitzenstrom hardwaremäßig noch nicht erreicht ist, jedoch die Ansteuerung schon den exponentiellen Abfall generiert.

Um diese Totzeit zu kompensieren, kann man mit dem Parameter *Dwell* angeben, wie lange die Ansteuerung bei dem Spitzwert verweilen soll, bevor der Abfall beginnt. Defaultwert ist 0.

#### <SampleRate>

Mit dem Parameter *SampleRate* wird das Messintervall festgelegt, mit dem bei der Ausführung der E-Funktion Spannung und Strom gemessen und mit dem entsprechenden Zeitstempel im internen Messwertspeicher abgelegt werden (s. Subsystem *DATA*).

Nach 8 Tau gilt die e-Funktion und somit das Messwertspeichern als beendet.

#### <DWELL>

Since any electronic load has got a certain rise time it may be possible that the power-stage hasn't reached a high peak current although the control unit already performs the exponential fall.

To compensate this dead time you can define the *dwell* time the control unit shall wait at the peak time until the beginning of the fall.

Default value is 0.

#### <SampleRate>

The parameter *SampleRate* defines the time interval which is used for measuring and storing voltage and current with the corresponding timestamp in the internal measurement memory (see subsystem *DATA*).

After 8 Tau the e-function and therefore the measurement sampling is terminated.



Die Messdatenspeicherung ist immer aktiv, auch wenn diese nicht explizit mit dem Parameter *SampleRate* programmiert wird. Die *SampleRate* ist mit  $200\mu\text{s}$  vorgelegt. 0 wird intern zum Minimalintervall  $200\mu\text{s}$  umgewandelt.

### Start durch Triggerspannung

Das Starten der Exponentialfunktion wird ausschließlich durch die Eingangsspannung der Last bestimmt, die den zuvor programmierten Triggerwert (*VOLTage:PROTection*) überschreiten muss.

Die E-Funktion wird sodann 8 Tau lang ausgeführt.

Beim Start der E-Funktion beginnt auch die Messdatenspeicherung wieder bei 0, d.h. nicht ausgelesene Datensätze werden überschrieben.

### Nachtriggern

Eine laufende E-Funktion ist nachtrIGGERbar. Um ein prellfreies Aufschalten des Prüflings zu ermöglichen, kann die NachtrIGGERbarkeit verhindert werden. Die Zeit, in der ab dem Trigger nicht nachgetriggert werden kann, wird mit dem Systemparameter 13 programmiert.

SYSTem:PARAmeter 13, <NRf>

Der Zeitparameter wird in Sekunden angegeben und ist nichtflüchtig. Als Defaultwert sind 10ms eingestellt.

**Beispiel:** Bei einmal erfolgtem Trigger soll die E-Funktion mindestens 25ms lang nicht nachgetriggert werden können:  
SYST:PAR 13,0.025

Sampling and storing records is always active, also if they haven't been programmed by the *SampleRate* parameter. The *SampleRate* is pre-defined with  $200\mu\text{s}$ . 0 is internally converted to the minimum interval of  $200\mu\text{s}$ .

### Start by Trigger Voltage

The starting of the exponential function is exclusively started by the load's input voltage which must exceed the previously programmed trigger voltage (*VOLTage:PROTection*).

Then the e-function is executed for 8 Tau. At start of the e-function the measurement recording restarts at 0, that means data which haven't been read out are overwritten.

### Retrigger

A running e-function is retriggerable. To enable a debounced switching of the device under test one can prevent this retriggerability. The time which the load does not retrigger since the first trigger is set by the system parameter 13.

SYSTem:PARAmeter 13, <NRf>

The time parameter is set in seconds. It is non-volatile. 10ms are set as default.

Example: When the trigger has occurred the e-function shall not be retriggerable for 25ms:  
SYST:PAR 13,0.025

**SFUNction:EXPonential:ENABLE  
ON|OFF|1|0**

Erlaubt oder verhindert die Ausführung der e-Funktion bei Überschreiten der Triggerspannung.

Der Parameter ON bzw. 1 erlaubt die E-Funktion bei  $U_{in} > TV$  (die Funktion selbst wird durch den Befehl noch nicht gestartet). Der Parameter OFF bzw. 0 verhindert die Ausführung, d.h. auch wenn die Eingangsspannung die programmierte Triggerspannung überschreitet, wird die E-Funktion nicht ausgeführt.

Beispiel:

*SFUN:EXP:ENAB ON* (E-Fkt. enabeln)

**SFUNction:EXPonential:ENABLE?**

Abfrage des Aktivierungszustandes der E-Funktion.

Als Rückgabewert wird eine 1 bereitgestellt, wenn die Ausführung der E-Funktion freigegeben ist.

Bei deaktivierter Ausführung der E-Funktion wird mit 0 geantwortet.

Beispiel:

*SFUN:EXP:ENAB?* (Antwort bei freigegebener Ausführung: 1)

**SFUNction:EXPonential:STATE?**

Abfrage des Ausführungszustandes der E-Funktion.

Als Rückgabewert wird eine 1 bereitgestellt, wenn die E-Funktion gerade ausgeführt wird.

Mit 0 wird dann geantwortet, wenn die E-Funktion momentan nicht ausgeführt wird oder nach mehr als 8 Tau nach dem Start der Funktion.

Beispiel:

*SFUN:EXP:STAT?* (Antwort bei laufender Ausführung: 1)

**SFUNction:EXPonential:ENABLE  
ON|OFF|1|0**

Enables or disables the execution of the e-function when the trigger voltage is exceeded.

The parameter ON or 1 enables the e-function at  $U_{in} > TV$  (the function itself is not yet started by this command). The parameter OFF or 0 disables the execution, that means even when the input voltage exceeds the programmed trigger voltage the e-function won't be executed.

Example:

*SFUN:EXP:ENAB ON* (enable e-function)

**SFUNction:EXPonential:ENABLE?**

Queries the activation state of the exponential function.

The load responds with 1 if the execution of the e-function is enabled.

At disabled execution the load responds with 0.

Example:

*SFUN:EXP:ENAB?* (Response at enabled execution: 1)

**SFUNction:EXPonential:STATE?**

Queries whether the programmed exponential function is running.

The load returns 1 if the e-function is currently being executed.

It returns 0 if the e-function is not being executed or after more than 8 Tau since the start of the function.

Example:

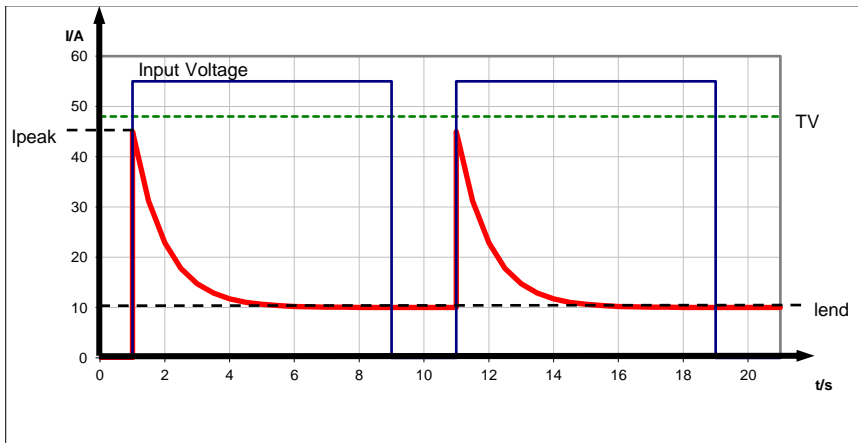
*SFUN:EXP:STAT?* (Response at running execution: 1)

Beispiel: Programmierung einer e-Funktion mit 45A Spitzenstrom, 10A Endstrom, 1s Tau, 50ms Samplerate, 5V Triggerspannung.

Befehlsfolge:  
SET:ADC FAST  
CURR:RANG 50;:CURR 0;MODE:CURR  
VOLT:PROT 5  
SFUN:EXP 45,10,1,0,0.05  
SFUN:EXP:ENAB ON  
INP ON

Example: Programming an e-function with 45A peak value, 10A end value, 1s Tau, 50ms sample rate, 5V trigger voltage.

Commands:  
SET:ADC FAST  
CURR:RANG 50;:CURR 0;MODE:CURR  
VOLT:PROT 5  
SFUN:EXP 45,10,1,0,0.05  
SFUN:EXP:ENAB ON  
INP ON





Bit	Wert/Value	Bit	Wert/Value
0	1	8	256
1	2	9	512
2	4	10	1024
3	8	11	2048
4	16	12	4096
5	32	13	8192
6	64	14	16384
7	128	15	32768

Beim Einschalten sind alle Bits sämtlicher Statusregister FALSE - bis auf Bit 7 (PON) im Standard Event Register (s.u.).

Das Statusmodell gliedert sich in folgende Gruppen (s.u.):

- Questionable Status
- Operation Status
- Standard Event Status
- Status Byte

Außerdem sind einige Gruppen unterteilt in:

#### • Condition Register

repräsentiert den momentanen Zustand bestimmter Signale. Der Bitzustand eines Condition Registers wird durch das Lesen nicht verändert. Ein Zustand/Fehler ist aktiv, wenn das zugehörige Bit TRUE ist. Ist die entsprechende Bedingung nicht mehr erfüllt, wird auch das jeweilige Bit im zugehörigen Condition Register wieder gelöscht.

#### • Event Register

speichert Informationen über bestimmte Zustände. Jedes Bit eines Event Registers korrespondiert mit einem Bit im Condition Register (beim Questionable Status und Operation Status) oder direkt mit bestimmten Ereignissen (Standard Event Status).

Ein Ereignis, d.h. ein Bit im Event Register, wird TRUE, wenn die zugehörige Bedingung einen Wechsel von FALSE nach TRUE gemacht hat.

After the activation of the device all bits of all status registers are FALSE - except Bit 7 (PON) in the Standard Event Register (see following sections).

The status model contains the following groups:

- Questionable Status
- Operation Status
- Standard Event Status
- Status Byte

Moreover some groups contain of:

#### • Condition Register

Represents the state of particular signals. The bit state of a Condition Register is not changed by reading it. A state/error is active, if the corresponding bit is TRUE. If the condition is no longer valid, the bit in the corresponding Condition Register is set to 0.

#### • Event Register

Saves information about particular states. Every bit of an Event Register corresponds to a bit in the Condition Register (for Questionable Status and Operation Status) or directly to special events (Standard Event Status).

An event, i.e. a bit in the Event Register, is set to TRUE, when the corresponding condition has changed from FALSE to TRUE.

Das Ereignis bleibt so lange gesetzt, bis das entsprechende Event Register gelesen worden ist. Beim Lesen werden alle Bits im betreffenden Event Register zurückgesetzt.

- **Enable Register**

bestimmt, welche Bits des zugehörigen Event Registers logisch zu einem Summenbit geODERT werden.

Das Enable Register wirkt sozusagen wie ein Filter auf das zugehörige Event Register.

Der Bitzustand eines Enable Registers wird durch das Lesen nicht verändert.

Auf der folgenden Seite ist das Statusmodell mit seinen einzelnen Registern dargestellt.

The event is set until the corresponding Event Register has been read. After reading all bits in the Event Registers are reset.

- **Enable Register**

Determines which bits of the corresponding Event Registers are combined to a total bit using OR.

The Enable Register acts as filter for the corresponding Event Register.

The bit state of an Enable Register is not changed by reading it.

On the following page the status model is shown with its single registers.



**Questionable Status**

Die Questionable Status Register informieren über bestimmte Fehler- bzw. Überlastzustände.

**Beispiel:** Überlastsignal der Leistungseinheit setzt Bits 0, 1, 3 und 4.  
Dezimalwert:  $1+2+8+16=27$ .

**Questionable Status**

The Questionable Status Register inform about particular error or overload states.

**Example:** The overload signal from the power unit sets the bits 0, 1, 3 and 4.  
Decimal value:  $1+2+8+16=27$ .

Bit	Wert Value	Bedeutung	Description
0 VOLT	1	Überspannungsfehler. LED OVP an der Frontplatte des Gerätes leuchtet.	Over-voltage error. LED OVP at front panel lights up.
1 CURR	2	Überstromfehler. LED OCP an der Frontplatte des Gerätes leuchtet.	Over-current error. LED OCP at front panel lights up.
3 POW	8	Überleistungsfehler. LED OPP an der Frontplatte des Gerätes leuchtet.	Over-power error. LED OPP at front panel lights up.
4 TEMP	16	Übertemperatur. LED OTP an der Frontplatte des Gerätes leuchtet.	Over-temperature error. LED OTP at front panel lights up.
9 WD	512	Watchdog. Wird gesetzt, wenn aktivierter Software-Watchdog den Lasteingang abgeschaltet hat.	Watchdog. Is set when the software watchdog shuts off the load input.
10 UV	1024	Unterspannung. Wird gesetzt, wenn die Eingangsspannung nicht ausreicht, um die eingestellte Belastung aufrechtzuerhalten.	Under voltage. Is set when the input voltage is not high enough to maintain the desired load setting.
11 TV	2048	Triggerspannung. Wird gesetzt, wenn die Eingangsspannung unterhalb der programmierten Triggerspannung ( <i>VOLTage:PROTection</i> ) liegt.	Trigger voltage. Is set when the input voltage is below the programmed trigger voltage ( <i>VOLTage:PROTection</i> )
12 DATA	4096	Datenspeicher. Wird gesetzt, wenn die maximale Anzahl der im internen Messwertspeicher abzulegenden Daten erreicht ist.	Data memory. Is set when the maximum number of data records is stored in the internal measurement memory.



**STATus:QUEStionable:CONDition?**

Abfrage des Inhaltes des Questionable Status Condition Registers.

Als Rückgabewert wird ein ganzzahliger Dezimalwert bereitgestellt, der den aktuellen Zustand der Schutzeinrichtungen beschreibt.

Die Dezimalzahl kann nach obiger Tabelle decodiert werden.

Beispiel:

STAT:QUES:COND?            Antwort: 27

**STATus:QUEStionable[:EVENTi]?**

Abfrage des Inhaltes des Questionable Status Event Registers.

Als Rückgabewert wird ein ganzzahliger Dezimalwert bereitgestellt, der erkennen lässt, ob ein Questionable Status seit dem letzten Lesen des Event Registers aktiv war. Ein Bit im Event Register wird nicht automatisch wieder gelöscht, wenn das Ereignis nicht mehr ansteht, sondern bleibt TRUE, bis das Event Register gelesen wird.

Erst nach dem Lesen des Event Registers wird dieses auf 0 zurückgesetzt.

Beispiel:

STAT:QUES?            Antwort: 27

**STATus:QUEStionable:ENABLE  
<0...65535>**

Setzt die durch den dezimalen Parameter festgelegte Bitkombination im Questionable Status Enable Register.

Damit wird festgelegt, welche Bits aus dem Questionable Event Register in der Auswertung für das QUES Summenbit relevant sind.

Beispiel: Bit TEMP und WD setzen

STAT:QUES:ENAB 528

**STATus:QUEStionable:CONDition?**

Queries the contents of the Questionable Status Condition Register.

Return value is an integer decimal value that describes the actual state of the protection facilities.

The decimal value is coded corresponding to the table.

Example:

STAT:QUES:COND?            Resp.: 27

**STATus:QUEStionable[:EVENTi]?**

Queries the contents of the Questionable Status Event Register.

Return value is an integer decimal value that determines, whether a Questionable Status has been active since the last reading of the Event Register. A bit in the Event Register is not automatically deleted, if the event is no more valid, but stays TRUE, until the Event Register is read.

After reading the Event Register the bit is reset to 0.

Example:

STAT:QUES?            Response: 27

**STATus:QUEStionable:ENABLE  
<0...65535>**

Sets the bit pattern for the Questionable Status Enable Register, that is determined by the decimal parameter.

Determines which bits from the Questionable Event Register are relevant for the interpretation of the QUES sum bit.

Example: Set bits TEMP and WD

STAT:QUES:ENAB 528

**STATus:QUEStionable:ENABle?**

Abfrage des Inhaltes des Questionable Status Enable Registers.

Als Rückgabewert wird der selbe Dezimalwert als Ganzzahl zur Verfügung gestellt, der zuvor programmiert worden ist.

Beispiel:

**STAT:QUES:ENAB?**                      Antwort: 528

Ist zuvor noch kein Wert für das Enable Register programmiert worden, liefert das Gerät als Antwort 0.

**STATus:QUEStionable:ENABle?**

Queries the contents of the Questionable Status Enable Register.

The decimal value that has been programmed previously is returned.

Example:

**STAT:QUES:ENAB?**                      Response:  
528

If no value has been programmed for the Enable Register, the device returns 0.

**Operation Status**

Die Operation Status Register geben Auskunft über den Betriebszustand der elektronischen Last.

**Operation Status**

The Operation Status Registers provide information about the operating state of the electronic load.

Bit	Wert Value	Bedeutung	Description
5 TRG	32	Trigger (reserviert)	Trigger (reserved)
8 PCYC	256	Programmable Cycles. Wird gesetzt, wenn das Gerät eine programmierte Kurvenform ausführt.	Programmable Cycles. Is set when the device executes a programmed waveform.

**STATus:OPERation:CONDition?**

Abfrage des Inhaltes des Operation Status Condition Registers.

Als Rückgabewert wird ein ganzzahliger Dezimalwert bereitgestellt, der den aktuellen Betriebszustand der elektronischen Last beschreibt.

Die Dezimalzahl kann nach obiger Tabelle decodiert werden.

Beispiel:

STAT:OPER:COND?      Antwort: 256

**STATus:OPERation:CONDition?**

Queries the contents of the Operation Status Condition Register.

Return value is an integer decimal value that describes the actual state of the electronic load.

The decimal value can be coded corresponding the table.

Example:

STAT:OPER:COND?      Resp.: 256

**STATus:OPERation[:EVENTi]?**

Abfrage des Inhaltes des Operation Status Event Registers.

Als Rückgabewert wird ein ganzzahliger Dezimalwert bereitgestellt, der erkennen lässt, ob ein Operation Status seit dem letzten Lesen des Event Registers aktiv war. Ein Bit im Event Register wird nicht automatisch wieder gelöscht, wenn das Ereignis nicht mehr ansteht, sondern bleibt TRUE, bis das Event Register gelesen wird.

**STATus:OPERation[:EVENTi]?**

Queries the contents of the Operation Status Event Register.

Return value is an integer decimal value that determines, whether an Operation Status has been active since the last reading of the Event Register. A bit in the Event Register is not automatically deleted if the event is no more valid, but stays TRUE until the Event Register is read.

Erst nach dem Lesen des Event Registers wird dieses auf 0 zurückgesetzt.

Beispiel:

STAT:OPER?                      Antwort: 256

**STATus:OPERation:ENABle**  
**<0...65535>**

Setzt die durch den dezimalen Parameter festgelegte Bitkombination im Operation Status Enable Register.

Damit wird festgelegt, welche Bits aus dem Operation Event Register in der Auswertung für das OPER Summenbit relevant sind.

Beispiel: Bit PCYC und TRAN setzen

STAT:QUES:ENAB 528

**STATus:OPERation:ENABle?**

Abfrage des Inhaltes des Operation Status Enable Registers.

Als Rückgabewert wird der selbe Dezimalwert (als Ganzzahl) zur Verfügung gestellt, der zuvor programmiert worden ist.

Beispiel:

STAT:QUES:ENAB?                      Antwort:  
528

Ist zuvor noch kein Wert für das Enable Register programmiert worden, liefert das Gerät als Antwort 0.

**STATus:PRESet**

setzt das Questionable Status Enable und das Operation Status Enable Register zurück auf 0.

After reading the Event Register the bit is reset to 0.

Example:

STAT:OPER?                      Response: 256

**STATus:OPERation:ENABle**  
**<0...65535>**

Sets the bit pattern for the Parameter Operation Status Enable Register, that is determined by the decimal parameter.

Determines which bits from the Operation Event Register are relevant for the interpretation of the OPER sum bit.

Example: Set Bits PCYC and TRAN

STAT:QUES:ENAB 528

**STATus:OPERation:ENABle?**

Queries the contents of the Operation Status Enable Register.

The decimal value that has been previously programmed is returned (as integer).

Example:

STAT:QUES:ENAB?                      Response:  
528

If no value has been programmed for the Enable Register, the device returns 0.

**STATus:PRESet**

Resets the Questionable Status Enable and the Operation Status Enable Register to 0.

**Standard Event Status**

Das Standard Event Status Register enthält Informationen über Standardereignisse, die in der Norm IEEE 488.2 definiert sind.

**Standard Event Status**

The Standard Event Status Register contains information about the standard events, that are defined in the standard IEEE 488.2.

Bit	Wert Value	Bedeutung	Description
0 OPC	1	Operation Complete. Das Gerät hat alle anstehenden Befehle ausgeführt. Bei den Geräten der Serie ZS immer TRUE, da die Befehle nicht im Overlapped Modus ausgeführt werden, sondern immer nacheinander.	Operation Complete. The device has executed all pending commands. For the devices of the series ZS this bit is always TRUE, because the commands are executed serially and not in overlapped mode.
2 QYE	4	Query Error. Errors im Bereich von -400 bis -499 können dieses Bit setzen.	Query Error. Errors in the range from -400 to -499 can set this bit.
3 DDE	8	Device Dependent Error. Errors im Bereich von -399 bis 300 können dieses Bit setzen.	Device Dependent Error. Errors in the range from -399 to -300 can set this bit.
4 EXE	16	Execution Error. Errors im Bereich von -299 bis -200 können dieses Bit setzen.	Execution Error. Errors in the range from -299 to -200 can set this bit.
5 CME	32	Command Error. Errors im Bereich von -199 bis -100 können dieses Bit setzen.	Command Error. Errors in the range from -199 to -100 can set this bit.
7 PON	128	Power On. Zeigt an, dass seit dem letzten Lesen ein OFF → ON Wechsel stattgefunden hat.	Power On. Indicates that the load has been switched off and on since the last reading.

Zum Lesen des Standard Event Status Registers wird das Common Command **\*ESR?** verwendet.

For reading the Standard Event Status Register the common command **\*ESR?** is used.

Der Befehl  
**\*ESE <0...255>**

setzt die durch den dezimalen Parameter festgelegte Bitkombination im Standard Event Status Enable Register.

The command  
**\*ESE <0...255>**

sets the bit pattern in the Standard Event Status Enable Register that is determined by the decimal parameter.

Damit wird festgelegt, welche Bits aus dem Standard Event Register in der Auswertung für das ESB Summenbit relevant sind.

Beispiel: Bit CME setzen

\*ESE 32

Mit

**\*ESE?**

findet die Abfrage des Inhaltes des Standard Event Status Enable Registers statt.

Als Rückgabewert wird der selbe Dezimalwert zur Verfügung gestellt, der zuvor programmiert worden ist.

Beispiel:

\*ESE?

Antwort: 32

### Status Byte

Im Status Byte Register sind die Status Events aller Status Register summiert.

Das Statusbyte wird gelesen mit dem Common Command

**\*STB?**

Determines which bits from the Standard Event Register are relevant for the interpretation of the ESB sum bit.

Example: Set Bit CME

\*ESE 32

**\*ESE?**

queries the contents of the Standard Event Status Enable Registers.

The decimal value (as integer) that has been programmed is returned.

Example:

\*ESE?

Response: 32

### Status Byte

In the Status Byte Register the Status Events of all Status Registers are combined.

The status byte is read using the command

**\*STB?**

Bit	Wert Value	Bedeutung	Description
QUES	8	Questionable. Ein enabledes Questionable Event ist eingetreten.	Questionable. An enabled Questionable Event has happened.
ESB	32	Event Status Bit. Ein enabledes Standard Event ist eingetreten.	Event Status Bit. An enabled Standard Event has happened.
MSS	64	Master Summary Status. Reserviert.	Master Summary Status. Reserved.
OPER	128	Operation. Ein enabledes Operation Event ist eingetreten.	Operation. An enabled Operation Event has happened.

Das Statusbyte wird nach dem Lesen automatisch auf 0 zurückgesetzt.

The status byte is reset to 0 after being read.

## 7.4.2.18 Subsystem SYSTEM

Befehl	Parameter	Einheit	Bemerkung
SYSTem :COMMunication :SERial :CONTrol :RTS [:RECeive] :PACE :CONTrol  :CONTrol? :ERRor? :FAN :LANG :LANG? :PARAmeter :PARAmeter <NRf>? :PROTection [:LEVel] [:LEVel]? :STATe :TRIPped? :SPEEd  :SPEEd? :STRing :VERSion?	IBFull   RFR  XON EXTernal   INTernal  AUTO   FULL SCPI  <NRf> <NRf>  <NRf>  <Boolean>  SLOW   FAST   MEDium  250, "string"	[S   MS]	RTS/CTS-Handshake ein  XON/XOFF-Flusssteuerung ein Externe oder interne Ansteuerung setzen Ansteuerquelle abfragen letzte Fehlermeldung lesen Lüftergeschw. auto oder voll Programmiersyntax Programmiersyntax abfragen Systemparameter programmieren Systemparameter lesen  SW-Watchdog Zeit setzen SW-Watchdog Zeit abfragen Software-Watchdog ein   aus Watchdog-Zustand abfragen Regelzeitkonstante umschalten  Regelgeschwindigkeit abfragen Formatstring definieren SCPI-Version abfragen

## 7.4.2.18 Subsystem SYSTEM

Command	Parameter	Unit	Comment
SYSTem :COMMunication :SERial :CONTrol :RTS [:RECeive] :PACE :CONTrol  :CONTrol? :ERRor? :FAN :LANG :LANG? :PARAmeter :PARAmeter <NRf>? :PROTection [:LEVel] [:LEVel]?	IBFull   RFR  XON EXTernal   INTernal  AUTO   FULL SCPI  <NRf> <NRf>  <NRf>	[S   MS]	Activate RTS/CTS handshake  Activate XON/XOFF flow control Set control source to external or internal Query control source Read last error message Fan speed auto or full Programming language Query programming language Set system parameter Query system parameter  Set SW-Watchdog time Query SW-Watchdog time



:STATe	<Boolean>	Software-Watchdog on   off
:TRIPped?		Query Watchdog state
:SPEEd	SLOW   FAST   MEDIUm	Change control time constant
:SPEEd?		Query regulation time constant
:STRing	250,"string"	Define format string
:VERSion?		Query SCPI version

### SYSTem:COMMunication:SERial :CONTRol:RTS IBFull | RFR

Aktiviert den Hardware-Handshake RTS/CTS der RS232-Schnittstelle mit einem der Parameter *IBFull* oder *RFR* (ready for receive). Es gibt keinen Parameter zur Deaktivierung.

In diesem Modus sendet die Last nur Daten, wenn dies durch die aktivierte CTS-Leitung der RS232-Schnittstelle ermöglicht wird. Umgekehrt deaktiviert die Last die RTS-Leitung, wenn ihr Dateneingangspuffer nahezu voll ist. Ist wieder genügend Speicherplatz freigeworden, aktiviert sie die RTS-Leitung wieder.

Dieser Befehl ist flüchtig, d.h. nach dem Aus- und Einschalten der Last ist das RTS/CTS-Handshake immer deaktiviert, ebenso nach einem Reset (\*RST).

#### Beispiel:

SYST:COMM:SER:CONT:RTS RFR

### SYSTem:COMMunication:SERial [:RECeive]:PACE XON

Aktiviert durch den Parameter *XON* die softwaremäßige Datenflusssteuerung XON/XOFF. Es gibt keinen Parameter zur Deaktivierung.

Es werden bestimmte Sonderzeichen zur Datenflusssteuerung benutzt: XON (11hex) und XOFF (13hex).

### SYSTem:COMMunication:SERial :CONTRol:RTS IBFull | RFR

Activates the hardware handshake RTS/CTS of the RS232 interface by one of the parameters *IBFull* or *RFR* (ready for receive). There is no parameter for deactivation.

In this mode the load only sends data if this is enabled by an activated CTS line of the RS232 interface. The other way, the load deactivates the RTS line if its data buffer is nearly full. When enough memory has been freed the load re-activates its RTS line.

This command is volatile, i.e. after power-on of the load RTS/CTS handshake is always deactivated, also after a reset (\*RST).

#### Example:

SYST:COMM:SER:CONT:RTS RFR

### SYSTem:COMMunication:SERial [:RECeive]:PACE XON

Activates the software-controlled data flow XON/XOFF by the parameter *XON*. There is no parameter for deactivation.

Special characters are used to control the data flow: XON (11hex) and XOFF (13hex).

Wenn der Dateneingangspuffer der Last nahezu voll ist, sendet sie das Zeichen XOFF, um den Sender zu stoppen. Ist wieder viel genug Speicher vorhanden, schickt sie XON.

Umgekehrt wird die Last durch ein ankommendes XOFF vom Senden weiterer Daten abgehalten, und zwar solange, bis ein XON im Eingangspuffer ankommt.

Dieser Befehl ist flüchtig, d.h. nach dem Aus- und Einschalten der Last ist das XON/XOFF-Handshake immer deaktiviert, ebenso nach einem Reset (\*RST).

Beispiel:

SYST:COMM:SER:PACE XON

### **SYSTEM:CONTROL EXTERNAL|INTERNAL**

Schaltet auf externe oder interne Ansteuerung. Nach dem Einschalten ist die interne Ansteuerung eingestellt.

Bei externer Ansteuerung kann über den Analog-I/O-Stecker mit einem Analogsignal von 0...10V angesteuert werden (s. Hardwareteil).

Beispiel:

SYST:CONT EXT

### **SYSTEM:CONTROL ?**

Abfrage der Ansteuerquelle. Als Antwort wird die Kurzform des entsprechenden Parameters gegeben.

Beispiel:

SYST:CONT? (Antwort: INT)

When the load's input buffer is getting nearly full the load sends the XOFF character to stop the sender. If there is again enough memory available XON is sent.

The other way, the load is prevented from sending by an XOFF character coming into the data buffer. The load stops sending until an XON character is detected.

This command is volatile, i.e. after power-on XON/XOFF handshake is always deactivated, also after a reset (\*RST).

Example:

SYST:COMM:SER:PACE XON

### **SYSTEM:CONTROL EXTERNAL|INTERNAL**

Switches to external or internal control. At power-on internal control is set.

At external control the load can be controlled via the analog I/O connector. The control signal has to be from 0...10V (see hardware manual).

Example:

SYST:CONT EXT

### **SYSTEM:CONTROL ?**

Queries the control source. The device returns the short-form of the corresponding parameter.

Example:

SYST:CONT? (Response: INT)

**SYSTem:ERRor?**

Abfrage der letzten Fehlermeldung.  
Das Gerät speichert einen Error solange, bis dieser aus der Error Queue ausgelesen worden ist. Danach wird der gelesene Error aus der Queue entfernt. Treten mehrere Fehler hintereinander auf, ohne dass ein Error gelesen worden ist, werden die Fehlernachrichten in der Queue um eine Stelle nach hinten gerückt. Der neue Fehler wird immer an den Anfang der Queue gestellt, der älteste wird aus der Queue hinausgeschoben. Zur Erkennung wird in diesem Fall der Fehlercode -350 (Queue Overflow) ans Ende der Queue gehängt.

**SYSTem:ERRor?**

Queries the last error message.  
The device saves an error until it has been read from the Error Queue. After reading the error it is deleted from the queue. If there are several errors without reading the queue, the error messages are saved in the queue. The new error is written to the beginning of the queue and the oldest error is deleted from the queue. This is signalized by the error code -350 (Queue overflow).

Bei der Serie ZS sind folgende  
Errornachrichten möglich:

Nachricht	Bedeutung
0, "No error"	Alle Befehle konnten fehlerlos ausgeführt werden.
102, "Syntax Error"	In einem Befehlsstring war ein Fehler, der nicht genauer definiert werden kann.
103, "Invalid separator"	Ein Trennzeichen wurde nicht richtig erkannt, z.B. wenn ein ':' zwischen zwei Schlüsselwörtern erwartet wird, jedoch ein ';' steht.
-110, "Command header error"	Ein ungültiges Schlüsselwort wurde erkannt
-200, "Execution error"	Ausführfehler. Wird benutzt, wenn keiner der Codes von -201 bis -294 für die Fehlerbeschreibung paßt
-220, "Parameter error"	Ein ungültiger Parameter wurde erkannt, z.B. Zahl wird erwartet, aber es wird keine Zahl als Parameter erkannt
-221, "Settings conflict"	Ein Befehl/Parameter war zwar korrekt, konnte jedoch aufgrund der momentanen Geräteeinstellungen nicht ausgeführt werden.
-222, "Data out of range"	Ein Parameter ist außerhalb des zulässigen Bereiches, z.B. "RES 0".
-223, "Too much data"	Im Gerät sind mehr Daten angekommen als verarbeitet werden können.
-224, "Illegal parameter value"	Für einige Befehle stehen nur ganz bestimmte Parameter aus einer Liste zur Verfügung. Keiner dieser Parameter konnte erkannt werden. Z.B. bei "TRAN:MODE CONT PULS,<NRf> TOGG"
-300, "Device specific error"	Ein Gerätezustand wurde erkannt, der nie eingenommen werden kann.
-340, "Calibration failed"	Nur bei der Inbetriebnahme von Bedeutung
-350, "Queue Overflow"	Es sind mehr Errors aufgetreten als in der Error Queue gespeichert werden können. Die "ältesten" Errors sind aus der Queue entfernt worden und können nicht mehr gelesen werden.
-360, "Communication error"	Fehler in der Datenübertragung. Kann z.B. sein: Framing Error (Start/Stopbits fehlerhaft), Parity Error
-363, "Input buffer overrun"	Es wurden mehr Zeichen an das Gerät gesendet, als im Befehlsdatenpuffer gespeichert werden können.

The ZS series supports the following error messages:

Message	Description
"0, No error"	All commands could be executed correctly.
"102, Syntax Error"	There was an unknown error in a command string.
"103, Invalid separator"	A separator hasn't been recognized, for example a ':' between two keywords was expected, but a ';' was provided.
"-110, Command header error"	Invalid keyword.
"-200, Execution error"	Execution error. Is used if none of the codes from -201 to -294 offer a useful error description.
"-220, Parameter error"	Invalid parameter, for example a number, was expected but not provided as parameter.
"-221, Settings conflict"	A command/parameter was correct, but couldn't be executed because of the actual device state.
"-222, Data out of range"	A parameter is not contained in a valid range, for example "RES 0".
"-223, Too much data"	The device has received more data than can be proceeded.
"-224, Illegal parameter value"	For some commands only special parameters can be used. None of these parameters have been recognized, for example "TRAN:MODE CONT PULS,<NRf> TOGG".
"-300, Device specific error"	A device state has been recognized that can't be set.
"-340, Calibration failed"	Reserved.
"-350, Queue overflow"	There have been more errors than can't be saved in the error queue. The "oldest" errors have been deleted from the queue and can't no longer be read.
"-360, Communication error"	Data transmission error, for example Framing Error (Start/Stopbits faulty), Parity Error
"-363, Input buffer overrun"	There have been sent more characters to the device than can be saved in the command data buffer.

**SYSTem:FAN AUTO|FULL**

Schaltet bei Parameter AUTO den/die Gerätelüfter auf automatischen (strom- und temperaturgesteuerten) Betrieb oder bei Parameter FULL auf dauernde Maximalgeschwindigkeit.

Nach dem Einschalten und nach Reset (\*RST) ist die automatische Lüftersteuerung aktiv.

Beispiel:

SYST:FAN FULL      volle Lüfter-  
geschwindigkeit

**SYSTem:FAN?**

Abfrage der eingestellten Lüftergeschwindigkeit. Als Antwort wird AUTO oder FULL gegeben.

Beispiel:

SYST:FAN?      (Antwort: AUTO)

**SYSTem:LANGuage SCPI**

Nur zur Konformität mit anderen H&H-Lasten. Die Programmiersyntax kann bei der Serie ZS nicht umgestellt werden, sondern ist immer SCPI.

Beispiel:

SYST:LANG SCPI

**SYSTem:LANGuage?**

Abfrage der eingestellten Programmiersyntax. Als Antwort wird immer SCPI gegeben.

Beispiel:

SYST:LANG?      (Antwort: SCPI)

**SYSTem:FAN AUTO|FULL**

Switches the fans to automatic (current and temperature-controlled) operation at parameter AUTO or to continuous maximum speed at parameter FULL. After power-on and after reset (\*RST) the automatic fan control is active.

Example:

SYST:FAN FULL      maximum fan  
speed

**SYSTem:FAN?**

Queries the fan speed. The device returns AUTO or FULL.

Example:

SYST:FAN?      (Response: AUTO)

**SYSTem:LANGuage SCPI**

Just for compatibility with other H&H loads. The programming syntax cannot be changed at ZS devices but is always SCPI.

Example:

SYST:LANG SCPI

**SYSTem:LANGuage?**

Queries the programming syntax. The device always returns SCPI.

Example:

SYST:LANG?      (Response: SCPI)

**SYSTEM:PARAMeter <Nrf>,<Nrf>  
SYSTEM:PARAMeter <NRF>?**

Programmieren und Lesen der Geräteparameter. Nur möglich bei ausgeschaltetem Lasteingang (INP OFF). Ausnahme Parameter 82 und 83. Hier ist die Programmierung bzw. das Lesen auch bei eingeschaltetem Lasteingang (INP ON) möglich

Der erste Parameter des Befehls gibt die Parameternummer an, der zweite den neuen Wert, den der Parameter annehmen soll.

**ZS System Parameter, die für den Anwender relevant sind:**

**SYSTEM:PARAMeter <Nrf>,<Nrf>  
SYSTEM:PARAMeter <NRF>?**

Programming and reading of the device parameters. Only possible when the load input is off (INP OFF).

Exception parameter 82 and 83. Programming and reading of the parameters also possible, if the load input is on.

The command's first parameter defines the parameter number, the second one defines the new value the parameter shall take.

**ZS System Parameters relevant for the programmer:**

Parameter Number	Name	Explanation	Factory Setting
48	Pmax	Maximum power of UUT	max. Short-time Power of Load <sup>1)</sup>
49	VOC	Open-circuit voltage of UUT	max. Voltage of Load <sup>1)</sup>
50	ISC	Short-circuit current of UUT	max. Current of Load <sup>1)</sup>
51	VPM	Voltage at MPP	max. Voltage of Load <sup>1)</sup>
52	IPM	Current at MPP	max. Current of Load <sup>1)</sup>
53	Delta_P_min	Minimum Delta Power	$0.0002 * U_{\text{Range}} * I_{\text{Range}}$ <sup>1)</sup>
...			
60	Device_Options	binary coded, second lowest Bit means "ZS13 installed"	
...			
81	Wave_min_steptime	Minimum waveform step width	$2.0E-4$ (200 $\mu$ s) <sup>2)</sup>
82	Power_IO_card_out	8 bit relay port of Power I/O card	all open <sup>2)</sup>
83	Power_IO_card_in	8 bit input port of Power I/O card	

<sup>1)</sup> non-volatile. Value is kept after power-off/power-on

<sup>2)</sup> volatile. Value is set to factory setting after power-off/power-on

**MPP Parameter (Nr. 48..53)**

Maximum Power Point Tracking wird vorwiegend beim Solarpaneltest angewendet.

Die Technischen Daten des Solarpanels werden in Systemparameter 48 bis 52 programmiert und dienen im wesentlichen zur Optimierung des Regelbereiches. Die Leerlaufspannung (Par. 49) kann angegeben werden, um den MPP schneller zu finden.

*Delta\_P\_min* definiert die Leistungsänderung, die sich mindestens ergeben muss, damit die Last die Spannung variiert. Um den optimalen Wert für diesen Parameter zu finden, sollte man mit einem nicht zu kleinen Wert beginnen und diesen sukzessive verkleinern, sonst besteht die Gefahr, dass der Regler "hängenbleibt", wenn *Delta\_P\_min* in der Messgenauigkeit untergeht.

**Device\_Options Parameter (Nr. 60)**

Das zweitniederwertigste Bit von Parameter 60 gibt Aufschluss darüber, ob die für einige Spezialfunktionen notwendige Option ZS13 im Gerät installiert ist. Die Antwort erfolgt als binär codierte Dezimalzahl, welche im Exponentialformat ausgegeben wird.

**MPP Parameters (No. 48..53)**

Maximum Power Point Tracking is mainly used for solar panel test. The Technical Data of the solar panel are programmed in parameters 48 to 52 and essentially serve to optimize the regulation range. The open-circuit voltage (par. 49) is specified to find the MPP faster.

*Delta\_P\_min* defines the power difference which must result at minimum that the load will vary the voltage. To find the best value for this parameter one shall start with a value not too small since with *Delta\_P\_min* below the measurement accuracy the controller may hang up.

**Device\_Options Parameter (Nr. 60)**

The second lowest bit of parameter 60 gives information if the option ZS13 required for a few special functions is available in the device. The answer is given as binary coded decimal number which is sent in exponential format.



Beispiel:

SYST:PAR 60? (Antwort: +2.00000E+00  
(ZS13 installiert))

**Waveform Parameter (Nr. 81)**

Beeinflusst die Auflösung eines Teilstücks einer programmierbaren Kurvenform. Damit kann man schnellere Kurvenformen erreichen.

Wird dieser Parameter  $< 200\mu\text{s}$  programmiert, ist jedoch keine synchrone Messung mehr möglich.

Der Wert wird in Sekunden programmiert und muss zwischen  $100\mu\text{s}$  und  $200\mu\text{s}$  liegen.

**Power I/O Card Parameter (Nr. 82..83)**

Bei diesen beiden Parameter ist das Lesen bzw. Schreiben auch bei eingeschaltetem Lasteingang (INP ON) möglich.

SYSTEM:PARAMeter wird u.a. auch zur Steuerung der optionalen Power-I/O-Karte verwendet. Mit der Power-I/O-Karte werden 8 Relais gesteuert und 8 Logikeingänge eingelesen. Die Steckerbelegung und technische Daten sind der Beschreibung im Hardwareteil zu entnehmen.

**Parameternummer 82 ist mit dem 8Bit-Relaisport der Power-I/O-Karte belegt, Parameternummer 83 mit dem 8Bit-Inputport.**

Es wird immer der ganze Port gelesen bzw. geschrieben, d.h. es kann z.B. nicht ein einzelner Relaiskontakt unabhängig von den übrigen Kontakten geschlossen werden, sondern der Sollzustand aller Kanäle muss programmiert werden.

Die Portbelegung wird als binär codierte Dezimalzahl angegeben. Relais 1 und Logikeingang 1 sind die niederwertigsten, Relais 8 und Logikeingang 8 die hochwertigsten Bits.

Example:

SYST:PAR 60? (Response:  
+2.000000E+00  
(ZS13 installed))

**Waveform Parameter (No. 81)**

Affects the resolution of a waveform's part line. Thereby faster waveforms can be achieved.

If this parameter is programmed  $< 200\mu\text{s}$  no synchronized measurement is possible.

The value must be programmed in seconds and has to be within  $100\mu\text{s}$  and  $200\mu\text{s}$ .

**Power I/O Card Parameters (No. 82..83)**

Programming and reading of the parameters also possible, if the load input (INP ON) is on.

Among others, SYSTEM:PARAMeter is used to control the optional Power I/O card. The Power I/O card controls 8 relays and 8 logic inputs. Refer to the description in the hardware manual for the pin assignment and technical characteristics.

**Parameter number 82 is occupied with the 8Bit relay port of the Power I/O card, parameter number 83 with the 8Bit input port.**

There is always the whole port written or read, respectively, that means, it is not possible to close for example one relay contact independently from the other contacts' states. The nominal state of all contacts must be programmed.

The port assignment is programmed as binary coded decimal number. Relay 1 and logic input 1 are the least significant, relay 8 and logic input 8 the most significant bits.

Bitzustand 1 zeigt einen geschlossenen Relaiskontakt bzw. einen logisch high Eingang an. Entsprechend zeigt Bitzustand 0 einen geöffneten Relaiskontakt bzw. einen logisch low Eingang an.

Mit SYSTem:PARAmeter <Nrf>? wird der Parameter der entsprechenden Stelle gelesen. Die Antwort wird in Exponentialform wie bei Messwerten gegeben.

#### Beispiele:

SYST:PAR 82,15 (15d=00001111b)  
Relais 1 bis 4 ein,  
Relais 5 bis 8 aus

SYST:PAR 82,65 (65d=01000001b)  
Relais 1 und 7 ein,  
alle anderen aus

SYST:PAR 82?  
Relaisport lesen.  
Antwort nach dem  
letzten Bsp.befehl:  
+6.500000E+01

SYST:PAR 83?  
Logikeingangs-Port  
lesen. Antwort z.B.  
+1.700000E+02  
(170d=10101010b)  
Bits 2,4,6,8 high,  
Bits 1,3,5,7 low

### **SYSTem:PROTectio[n]:[LEVEL] <0...4290000>**

Programmiert den geräteinternen Timer mit dem angegebenen Wert in Sekunden.

Die Geräte der Serie ZS sind mit einer watchdog-ähnlichen Software-Funktion ausgestattet, die bei eventuellem Absturz des Steuerrechners, Fehlbedienung der Steuersoftware o.ä. die Elektronische Last in einen sicheren Betriebszustand bringt.

#### Sicherer Betriebszustand heißt:

Wenn das Gerät über einen bestimmten Zeitraum hinweg (Defaulteinstellung oder programmierter Zeitraum) keinen

Bit state 1 means a closed relay contact or a logic high input. Bit state 0 means an open relay contact or a logic low input.

With SYSTem:PARAmeter <Nrf>? the corresponding parameter number is read. The response is given in exponential form, like it is with measurement values.

#### Examples:

SYST:PAR 82,15 (15d=00001111b)  
Relais 1 to 4 on,  
Relais 5 to 8 off

SYST:PAR 82,65 (65d=01000001b)  
Relais 1 and 7 on,  
all others off

SYST:PAR 82?  
Read relay port.  
Response after last  
command example:  
+6.500000E+01

SYST:PAR 83?  
Read logic input  
port. Resp. e.g.:  
+1.700000E+02  
(170d=10101010b)  
Bits 2,4,6,8 high,  
Bits 1,3,5,7 low

### **SYSTem:PROTectio[n]:[LEVEL] <0...4290000>**

Sets the device internal timer to the specified value, provided in seconds.

The devices of ZS series provide a watchdog-like software function, that sets the electronic load in a safe operating mode, if the controlling computer breaks down or if the controlling software is not correctly handled.

#### Safe operating mode means:

If the device hasn't received a command from the controlling computer for a defined time (default value or

Befehl vom steuernden Rechner erhalten hat, schaltet es den Lasteingang ab.

Die zu überwachende Zeit wird mit dem Befehl

`SYSTem:PROTection[:LEVel] <NRf>`

definiert. Der Parameter steht für den zu überwachenden Zeitraum in Sekunden.

Beispiel s.u.

### **SYSTem:PROTection[:LEVel]?**

Abfrage der programmierten Watchdogzeit in Sekunden.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt.

`SD.DDDDDDESDD` S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

### **SYSTem:PROTection:STATe**

**ON|1|OFF|0**

Aktiviert/deaktiviert den Software-Watchdog.

Wird der Watchdog aktiviert (`SYST:PROT:STAT ON`) und die programmierte Zeit läuft ab, ohne dass ein Befehl vom Steuerrechner im Gerät eintrifft, schaltet die elektronische Last den Lasteingang ab und der Watchdog wird deaktiviert. Alle anderen Einstellungen am Gerät bleiben erhalten. Dieser Zustand ist im Questionable Status ersichtlich.

Der Lasteingang kann dann wieder eingeschaltet und der Watchdog neu aktiviert werden.

Mit dem Befehl

`SYST:PROT:STAT OFF`

wird die Überwachung der programmierten Zeit deaktiviert, d.h. auch wenn die definierte Sekundenzahl abgelaufen ist, lässt die Last den Geräteeingang in seinem bisherigen Zustand.

programmed value), it deactivates the load input.

The watched time is defined using the command

`SYSTem:PROTection[:LEVel] <NRf>`

The parameter specifies the watched time interval, represented in seconds.

See example below.

### **SYSTem:PROTection[:LEVel]?**

Queries the programmed watchdog time in seconds.

A numeric value in exponent form is returned.

`SD.DDDDDDESDD` S: Sign,  
D: Digit,  
E: Exponent

### **SYSTem:PROTection:STATe**

**ON|1|OFF|0**

Activates/deactivates the software watchdog.

If the watchdog is activated (`SYST:STAT ON`) and the programmed time has been expired without a command being received from the controlling computer, the electronic load deactivates the load input and the watchdog gets deactivated. All other settings of the device are kept. This state can be queried from the Questionable Status Register.

The load input as well as the watchdog can be reactivated.

The command

`SYST:PROT:STAT OFF`

deactivates the watching of the programmed time, i.e. the load keeps the activated device input in its state, despite the programmed seconds have expired.

Beispiel:

Nach 15 Minuten ohne Dateneingang soll Eingang abgeschaltet werden:  
SYST:PROT 900;PROT:STAT ON

**SYSTem:PROTEction:STATe?**

Abfrage des Aktivierungszustandes des Software-Watchdogs.

Als Rückgabewert wird eine Boolesche Zahl (0 oder 1) zur Verfügung gestellt.

Dabei bedeutet:

0: Watchdogfunktion nicht aktiviert

1: Watchdogfunktion aktiviert

Beispiel:

SYST:PROT:STAT? Antwort: 0

**SYSTem:PROTEction:TRIPped?**

Abfrage des Auslösezustandes des Software-Watchdogs.

Als Rückgabewert wird eine Boolesche Zahl (0 oder 1) zur Verfügung gestellt.

Dabei bedeutet:

0: Watchdog hat nicht eingegriffen

1: Watchdog hat Lasteingang abgeschaltet.

Beispiel:

SYST:PROT:TRIP? Antwort: 0

**SYSTem:SPEEd SLOW|FAST|MEDIum**

Schaltet die Regelzeitkonstante des Gerätes um.

Bei schwingenden Systemen kann es von Vorteil sein, die Regelzeitkonstante auf SLOW oder MEDIUM zu schalten.

Beim Einschalten und bei Reset ist die mittlere Regelzeitkonstante eingestellt.

Beispiel:

SYST:SPEED SLOW (langsame Regelzeitkonstante setzen)

Example:

The input shall be deactivated, if there have been no data for 15 minutes:  
SYST:PROT 900;PROT:STAT ON

**SYSTem:PROTEction:STATe?**

Queries the activation state of the software watchdog.

Return value is a boolean number (0 or 1).

0: Watchdog is deactivated

1: Watchdog is activated

Example:

SYST:PROT:STAT? Response: 0

**SYSTem:PROTEction:TRIPped?**

Queries the trigger state of the software watchdog.

Return value is a boolean number (0 or 1).

0: Watchdog has not been triggered

1: Watchdog has deactivated the load input.

Example:

SYST:PROT:TRIP? Response: 0

**SYSTem:SPEEd SLOW|FAST|MEDIum**

Changes the control time constant of the device.

At oscillating systems switching to slow or medium regulation time constant may cause stabilization of the system.

At reset, the medium time constant is set.

Example:

SYST:SPE SLOW (set slow regulation time constant)

**SYSTem:SPEed?**

Abfrage der Regelgeschwindigkeit.

Als Rückgabewert wird der entsprechende momentane Parameter des Befehls gegeben.

Beispiel:

SYST:SPEED?                      Antwort: MED

**SYSTem:STRing 250,"formatstring"**

Mit dem Formatstring kann das Format der Messwerte, die die Last an den PC sendet, verändert werden. Dafür muss die Speicherstelle 250 benutzt werden.

Der Formatstring muss in Anführungszeichen (" ") stehen.

Dabei gilt die allgemein in der Programmiersprache 'C' übliche Syntax:

**%[flags][width][.precision][modifiers]type**

**SYSTem:SPEed?**

Queries the regulation speed.

The device returns the corresponding parameter of the command.

Example:

SYST:SPEED?                      Response: MED

**SYSTem:STRing 250,"formatstring"**

With the format string you can modify the format of measurement values the load sends to the PC. For this, storage position 250 must be used.

The format string must be set between quotes (" ").

The syntax usually valid in the programming language 'C' is used:

**%[flags][width][.precision][modifiers]type**

<i>type</i>	Ausgabe/Output	Beispiel/Example
<b>e</b>	Wissenschaftliche Notation (Mantisse/Exponent) mit <b>e</b> Scientific notation (mantise/exponent) using <b>e</b> character	3.9265e2
<b>E</b>	Wissenschaftliche Notation (Mantisse/Exponent) mit <b>E</b> Scientific notation (mantise/exponent) using <b>E</b> character	3.9265E2
<b>f</b>	Dezimale Gleitpunktzahl / Decimal floating point	392.65
<b>g</b>	Kürzestes aus %e oder %f / Use shorter %e or %f	392.65
<b>G</b>	Kürzestes aus %E oder %f / Use shorter %E or %f	392.65

<i>flags</i>	Meaning
-	Linksausrichtung mit der angegebenen Weite (default: Rechtsausrichtung) Left align within the given width (right align is the default)
+	Ergebnis beginnt zwingend mit einem Vorzeichen (+ oder -) (ansonsten wird nur - (Minus) ausgegeben) Forces to precede the result with a sign (+ or -) type (otherwise only - (minus) is printed)
blank	Wenn das Argument positiv ist, wird ein Leerzeichen vor die Zahl gesetzt. If the argument is a positive signed value, a blank is inserted before the number.

<i>width</i>	Meaning
<i>number</i>	Minimale Anzahl von auszugebenden Zeichen. Ist der auszugebende Wert kürzer als <i>number</i> , wird das Ergebnis mit Leerzeichen aufgefüllt. Der Wert wird nicht trunziert, auch nicht, wenn das Ergebnis länger ist. Minimum number of characters to be printed. If the value to be printed is shorter than this <i>number</i> the result is padded with blanks. The value is never truncated even if the result is larger.
<b>0number</b>	Wie oben, jedoch mit <b>0</b> en anstelle von Leerzeichen aufgefüllt. Same as above but filled with <b>0</b> s instead of blanks.

<i>.precision</i>	Meaning
<i>.number</i>	Bei <b>e</b> , <b>E</b> , <b>f</b> Typen: Anzahl der Stellen, die nach dem Dezimalpunkt ausgegeben werden sollen. Bei <b>g</b> , <b>G</b> Typen: maximale Anzahl auszugebender signifikanter Stellen For <b>e</b> , <b>E</b> , <b>f</b> types: number of digits to be printed after the decimal point (if nothing specified default is <b>6</b> ). For <b>g</b> , <b>G</b> types : maximum number of significant digits to be printed.

modifier	Meaning
	modifier nicht benutzt. / modifier not used.

**Beispiel:**  
Der Messwert 120.665 in verschiedenen  
Formaten:

**Example:**  
Value 120.665 in different formats:

Format String	Ergebnis / Result
SYST:STR 250,"%+9.6E"	+1.206650E+02
SYST:STR 250,"%9.3e"	1.207e+02
SYST:STR 250,"%f"	120.665000
SYST:STR 250,"%+f"	+120.665000
SYST:STR 250,"%08.2f"	00120.67
SYST:STR 250,"% 8.2f"	120.67

Nach dem Einschalten des Gerätes ist  
folgendes Default-Zahlenformat  
eingestellt:

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Dies entspricht folgendem Formatstring:  
"%+9.6E"

Die Formatstringinformation ist flüchtig,  
d.h. nach dem Aus- und Einschalten des  
Gerätes ist wieder das Defaultformat  
eingestellt.

### SYSTem:VERSion?

Abfrage der SCPI-Version, zu der das  
Gerät kompatibel ist.

**Beispiel:**

SYST:VERS?            Antwort: 1995.0

After power-on the following default  
format is set:

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

This corresponds to following format  
string:  
"%+9.6E"

The format string information is volatile,  
that means after switching the device off  
and on the default format is set.

### SYSTem:VERSion?

Queries the SCPI version that the device  
conforms.

**Example:**

SYST:VERS?            Response: 1995.0

## 7.4.2.19 Subsystem TRIGger

Befehl	Parameter	Einheit	Bemerkung
TRIGger [:SEQUence] :SOURce	BUS EXTernal  TIMer IMMediate		Triggerquelle festlegen
:SOURce?			Triggerquelle abfragen
:TIMer	<NRf>		Timerwert festlegen
:TIMer?			Timerwert abfragen

## 7.4.2.19 Subsystem TRIGger

Command	Parameter	Unit	Comment
TRIGger [:SEQUence] :SOURce	BUS EXTernal  TIMer IMMediate		Set trigger resource
:SOURce?			Query trigger resource
:TIMer	<NRf>		Set timer value
:TIMer?			Query timer value

Das Subsystem *TRIGger* ist zur Definition und Abfrage der aktuellen Triggerquelle vorhanden.

Das heißt, damit wird das Ereignis festgelegt, bei welchem ein mit *CURRENT[:LEVel]:TRIGgered <NRf>*, *RESistance[:LEVel]:TRIGgered <NRf>* oder *VOLTage[:LEVel]:TRIGgered <NRf>* programmierter Triggerwert aktiviert bzw. ein dynamischer Vorgang gestartet wird.

Beim Einschalten des Gerätes und bei Reset ist die Triggerquelle *IMM* eingestellt.

**TRIGger[:SEQUence]:SOURce  
BUS|EXTernal|IMMediate|TIMer**  
Legt die Triggerquelle für getriggerte Einstellwerte in allen Betriebsarten fest.

The subsystem *TRIGger* defines and queries the actual trigger source.

That means, an event is determined, for which the command *CURRENT[:LEVel]:TRIGgered <NRf>*, *RESistance[:LEVel]:TRIGgered <NRf>* or *VOLTage[:LEVel]:TRIGgered <NRf>* activates a programmed trigger value or starts a dynamic operation, respectively.

At the activation of the device, the trigger source *IMM* is set.

**TRIGger[:SEQUence]:SOURce  
BUS|EXTernal|IMMediate|TIMer**  
Sets the trigger resource for triggered setting values in the operating modes current and resistance.

Mit

**TRIGger[:SEQuence]:SOURce BUS**

wartet die elektronische Last auf ein Triggersignal vom steuernden Bus.

Dieses wird erzeugt durch

- das Common Command **\*TRG** (bei GPIB und RS232 möglich)
- die GPIB-Mehrdrahtnachricht **GET** (Group Execute Trigger, nur bei GPIB möglich)

Mit

**TRIGger[:SEQuence]:SOURce EXTernal**

wartet die elektronische Last auf ein Triggersignal vom Analog I/O-Stecker an der Geräterückseite.

Als externer Trigger muss ein Low-Puls von mindestens 50  $\mu$ s (TTL-Signal) an der Analog-I/O-Buchse angelegt werden (s. Steckerbelegung Analog-I/O-Stecker in Hardware-Bedienungsanleitung).

Mit

**TRIGger[:SEQuence]:SOURce IMMEDIATE**

wird das Warten auf ein Triggerereignis deaktiviert.

Mit

**TRIGger[:SEQuence]:SOURce TIMER**

wird der Ablauf des lastinternen Timers, dessen Wert mit **TRIGger[:SEQuence]:TIMer <NRf>** (s.u.) programmiert wird, als Triggerquelle zur Messdatensatz-speicherung festgelegt. Wenn noch kein Timerintervall (mit **TRIGger:TIMer <NRf>**) definiert bzw. auf 0 gesetzt worden ist, speichert jeder Befehl **TRIGger:SOURce TIMER** sozusagen manuell einen Messdatensatz mit Zeitstempel im geräteinternen Messdatenspeicher.

Using

**TRIGger[:SEQuence]:SOURce BUS**

the electronic load waits for a trigger signal from the controlling bus.

The bus trigger is produced by

- the Common Command **\*TRG** (for GPIB and RS232)
- the GPIB multi channel message **GET** (Group Execute Trigger, only for GPIB communication)

Using

**TRIGger[:SEQuence]:SOURce EXTernal**

the electronic load waits for a trigger signal from the Analog I/O Port at the back panel.

As external trigger signal a falling edge TTL signal must be provided to the Analog I/O connector (see nin assignment in hardware manual).

With

**TRIGger[:SEQuence]:SOURce IMMEDIATE**

waiting for a trigger is disabled.

With

**TRIGger[:SEQuence]:SOURce TIMER**

the expiry of the load-internal timer is defined as trigger source for measurement record savings. The timer value is defined using the command **TRIGger[:SEQuence]:TIMer <NRf>** (see below).

If no timer interval has been programmed yet or the timer interval has been set to 0 (with **TRIGger:TIMer <NRf>**) each command **TRIGger:SOURce TIMER** saves one measurement record with timestamp in the internal measurement storage quasi-manually.



**TRIGger[:SEQUence]:SOURce?**

Abfrage der aktiven Triggerquelle.

Als Rückgabewert wird die Kurzform des jeweiligen Parameters bereitgestellt.

Beispiel:

```
TRIG:SOUR?           Antwort: IMM
TRIG:SOUR EXTERNAL
TRIG:SOUR?           Antwort: EXT
```

**TRIGger[:SEQUence]:TImer <NRf>**

Legt das Timerintervall in sec. z.B. zur Speicherung von Messwertsätzen im **statischen** Betrieb fest. Die Triggerquelle (TRIGger:SOURce) muss dazu auf TImer gesetzt werden.

Bereits beim Einschalten des Gerätes läuft die lastinterne Zeit von 0s los bis max. 23.86h, d.h. wenn die Speicherung der Messdaten mit dem Befehl TRIGger:SOURce TImer gestartet wird, hat der erste gespeicherte Zeitstempel bereits den Wert, der seit Einschalten der Last verstrichen ist.

Der Zahlenparameter <NRf> dieses Befehls muss zwischen 200 $\mu$ s ... 23.86h liegen und ist mit 50 $\mu$ s aufgelöst. Der Wert muss in Sekunden programmiert werden.

Bei einem Standardgerät ohne Option ZS13 machen Zeitwerte unter 300ms keinen Sinn, da der Standard-ADC 300ms für einen neuen Messwert braucht. Wird dennoch eine kürzere Timerzeit programmiert, speichert das Gerät ggf. denselben Messdatensatz öfters hintereinander ab.

Beispiel:

```
VOLT 12.5;MODE:VOLT;:INP ON
TRIG:TIM 5
TRIG:SOUR TIM
...
DATA:REM?
```

**TRIGger[:SEQUence]:SOURce?**

Query the active trigger resource.

The return value is the short form of the corresponding parameter.

Example:

```
TRIG:SOUR?           Response: IMM
TRIG:SOUR EXT
TRIG:SOUR?           Response: EXT
```

**TRIGger[:SEQUence]:TImer <NRf>**

Defines the timer interval in sec., for example for the storage of measurement sets in **static** operation. For this purpose, the trigger source (TRIGger:SOURce) must be set to TImer.

At power-on the internal time runs from 0s up to maximum 23.86h. That means when the data recording is started by the command TRIGger:SOURce TImer the first timestamp stored will get already the value elapsed since the load had been powered-on.

The NRf parameter of this command must be set within 200 $\mu$ s ... 23.86h and is resolved with 50 $\mu$ s. The value must be in seconds.

With a standard device without ZS13 option time values below 300ms make no sense because of the conversion time of the standard ADC. If nevertheless a shorter time is programmed the device possibly saves the same data set repeatedly.

Example:

```
VOLT 12.5;MODE:VOLT;:INP ON
TRIG:TIM 5
TRIG:SOUR TIM
...
DATA:REM?
```

Bei *TRIG:SOUR TIM* wird der erste Messdatensatz mit Zeitmarke seit Einschalten gespeichert, die folgenden in diesem Beispiel immer nach 5s. Mit *DATA:REM?* werden alle im Speicher abgelegten Datensätze ausgelesen.

Die Speicherung der Messdatensätze erfolgt solange, bis die Triggerquelle auf einen anderen Parameter als *TIMER* gesetzt oder der Lasteingang ausgeschaltet wird.

S. auch Subsystem *DATA*.

### **TRIGger[:SEQUence]:TIMer?**

Abfrage des programmierten Timerwertes.

Als Rückgabewert wird eine einheitslose Zahl im Exponentialformat (in sec.) bereitgestellt.

#### Beispiel:

*TRIG:TIM?*

(Antwort z.B.: +5.000000E-01)

At *TRIG:SOUR TIM* the first data set with timestamp since power-on is stored, the following always after a period of 5s. The command *DATA:REM?* reads all records saved in memory.

The records are saved until the trigger source is set to a parameter different from *TIMER* or until the load input is switched off.

SEE also Subsystem *DATA*.

### **TRIGger[:SEQUence]:TIMer?**

Queries the programmed trigger timer value.

A numeric value without unit in exponent form is returned (in sec.).

#### Beispiel:

*TRIG:TIM?*

(Antwort z.B.: +5.000000E-01)

## 7.4.2.20 Subsystem VOLTage

Befehl	Parameter	Einheit	Bemerkung
VOLTage :CRANge :CRANge? [:LEVel] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE  :MODE? :PROTection [:LEVel] [:LOW] [:LOW]? :TRIPped? :RANGe :AUTO :RANGe?	<num>   <num> [MIN MAX] <num> [MIN MAX]      <num> [MIN MAX]  <num> <Boolean> [MIN MAX]	[A MA]   [V MV] [V MV]      [V MV]	Strombereich im Spannungsbetrieb Strombereich im V-Betrieb abfragen  Spannungswert einstellen Sollspannung abfragen getriggerte Sollspannung Spannungs-Triggerwert abfragen Bei Trigger Liste starten oder statischen Triggerstrom einst. Current Mode abfragen    Triggerspannung einstellen Triggersp.wert abfragen Triggersp.zustand abfragen fester Spannungsbereich Autorange ein aus Spannungsbereich abfragen

## 7.4.2.20 Subsystem VOLTage

Command	Parameter	Unit	Comment
VOLTage :CRANge :CRANge? [:LEVel] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE  :MODE? :PROTection [:LEVel] [:LOW] [:LOW]? :TRIPped? :RANGe :AUTO :RANGe?	<num> [MIN MAX]  <num> [MIN MAX] <num> [MIN MAX]      <num> <Boolean> [MIN MAX]	[A MA]   [V MV] [V MV]      [V MV]	Current range in V mode Query current range in V mode  Set nominal voltage Query nominal voltage Set triggered voltage Query triggered voltage Start list or set triggered static current at trigger Query current mode    Set trigger voltage Query nom. trigger voltage Query trigger voltage state Set voltage range Autorange on/off Query voltage range

Das Befehlssystem VOLTage dient zur Einstellung und Abfrage des Lastspannungs-Sollwertes und der Triggerspannung.

The command system VOLTage is used for setting and querying the voltage set point and the trigger voltage.

Bei Mehrbereichsgeräten (z.B. ZS506-4, ZS512-4, ZS530-3, ZS560-3) kann im Spannungsbetrieb ein Strombereich programmiert werden (VOLTage:CRANge xx), d.h. der Strombereich kann im Spannungsbetrieb nicht überschritten werden, auch wenn dadurch die Sollspannung evtl. nicht aufrechterhalten werden kann. Defaultmäßig ist im Spannungsbetrieb der größte Strombereich eingestellt.

### **VOLTage:CRANge <num>**

Stellt den Strombereich im Spannungsbetrieb ein. Nur möglich bei Mehrbereichsgeräten (z.B. ZS506-4, ZS512-4, ZS530-3, ZS560-3).

Damit wird der Strom im Spannungsbetrieb auf den jeweiligen Strombereich begrenzt (ein spezieller Strombegrenzungswert wird mit CURR:PROT programmiert).

Wenn ein Strombereich angegeben wird, der bei dem betreffenden Gerät nicht vorhanden ist, wird der nächsthöhere Strombereich eingestellt.

#### Beispiel ZS530-3 (120mA/1.2A/12A):

VOLT:CRAN 10

Da das Gerät über keinen 10A-Bereich verfügt, wird der nächsthöhere Strombereich von 12A im Spannungsbetrieb eingestellt.

Defaultmäßig ist der maximale Strombereich im Spannungsbereich aktiv, d.h. wenn VOLT:CRAN nicht programmiert wird, ist der volle Nennstrom im Spannungsbetrieb möglich. Durch entsprechende Auswahl von VOLT:CRAN kann dann auch die Strombegrenzung CURR:PROT mit angepasster Auflösung programmiert werden.

At multi-range devices such as ZS506-4, ZS512-4, ZS530-3, ZS560-3, there can be set a current range for constant voltage mode (VOLTage:CRANge xx). That means the current range can not be exceeded even if this causes the nominal voltage not to be set or kept.

As default, the maximum current range is set in constant voltage mode.

### **VOLTage:CRANge <num>**

Sets the current range in voltage mode. Only possible at multi-range devices (e.g. ZS506-4, ZS512-4, ZS530-3, ZS560-3).

This way the current in constant voltage mode is limited to the respective current range (a special current protection is programmed with CURR:PROT).

If a current range is programmed which is not present at the concerning load model the load sets the next higher current range in voltage mode.

#### Example ZS530-3 (120mA/1.2A/12A):

VOLT:CRAN 10

Since the device has no 10A range it sets the next higher current range of 12A in voltage mode.

As default, the maximum current range is active in constant voltage mode, i.e. if VOLT:CRAN is not programmed the full nominal current is possible in voltage mode. By choosing a proper current range with VOLT:CRAN xx the current protection CURR:PROT is programmed with the corresponding resolution.

**VOLTage[:LEVel][:IMMediate]****<num>**

Stellt eine neue Lastspannung ein. Befindet sich das Gerät im Spannungsbetrieb, wird der neue Wert sofort eingestellt, sofern dieser im gültigen Wertebereich liegt.

Die Einstellbereiche sind den Technischen Daten des jeweiligen Modells zu entnehmen.

Bei Überschreitung des zulässigen Wertebereichs wird ein "Data out of range"-Error ausgelöst, der mit `SYSTEM:ERRor?` ausgelesen werden kann. In diesem Fall wird der maximal mögliche Wert eingestellt und der programmierte Wert als Sollwert gespeichert. Wird später in einen ausreichend hohen Spannungsbereich gewechselt, so stellt die Last den Sollwert ein.

Befindet sich das Gerät nicht im Konstantspannungsbetrieb, wird der neue Einstellwert gespeichert und beim Wechsel in den Spannungsbetrieb (mit `MODE:VOLTage`) eingestellt.

Als Parameter sind alle Zahlenwerte innerhalb des momentanen Spannungsbereiches erlaubt.

Außerdem sind die speziellen Zahlenparameter MIN und MAX erlaubt. Im Autoranging-Betrieb stellt der Parameter MAX den Endwert im höchsten Strombereich ein, ansonsten den Endwert im momentanen Bereich.

Beispiele:

VOLT:LEV 15.23

VOLT:IMM 0

VOLT MAX

Als Dezimaltrennzeichen erwartet das Gerät einen Punkt (.), kein Komma!

**VOLTage[:LEVel][:IMMediate]****<num>**

Sets a new load voltage. If the device is in the operating mode voltage the new value will be set immediately if it is within the valid range.

The setting ranges are specified in the technical data of the particular device type.

When exceeding the allowed scope a "Data out of range"-Error is triggered, which is read using `SYSTEM:ERRor?`. In this case the maximum possible value is set and the programmed value is kept as nominal value. If a suited voltage range is set at a later time the load sets the saved nominal value.

If the device is not in the operating mode constant voltage, the new setting value is saved and set when changing to the operating mode voltage (using `MODE:VOLTage`).

As parameters all numeric values within the actual range of the particular model are allowed.

The specific numeric parameters MIN and MAX are also allowed.

In autoranging mode the parameter MAX sets the end value in the highest voltage range, otherwise the end value in the active range.

Examples:

VOLT:LEV 15.23

VOLT:IMM 0

VOLT MAX

As decimal separator the device expects a point (.), no comma!

**VOLTage[:LEVel][:IMMediate]?**

Abfrage des momentan eingestellten Sollwertes im Spannungsbetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

*SD.DDDDDDESDD* S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Einstellwert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Als Rückgabewert erhält man den im momentanen Spannungsbereich minimal bzw. maximal möglichen Einstellwert.

Beispiele:

VOLT? (Antw. z.B.: +1.850000E+01)

VOLT? MAX

(Antwort von ZS1406: +6.000000E+01)

**VOLTage[:LEVel]:TRIGgered****<num>**

Stellt einen neuen Wert für den getriggerten Spannungssollwert ein.

Das Gerät darf sich dabei nicht im Autoranging-Modus befinden. Der gewünschte Spannungsbereich muss vorher fest eingestellt werden.

Bei Überschreitung des zulässigen Wertebereichs wird ein "Data out of range"-Error ausgelöst, der mit *SYSTEM:ERRor?* ausgelesen werden kann. In diesem Fall wird der maximal mögliche Wert eingestellt und der programmierte Wert als Sollwert gespeichert. Wird später in einen ausreichend hohen Spannungsbereich gewechselt, so stellt die Last den Sollwert ein.

**VOLTage[:LEVel][IMMediate]?**

Queries the actual set point in the operating mode voltage.

A numeric value in exponential format is returned:

*SD.DDDDDDESDD* S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible setting value is queried appending a white space and the parameter MIN or MAX, respectively, to the question mark.

The load returns the minimum or maximum possible setting value for the actual voltage range.

Examples:

VOLT? (Response for example:  
+1.850000E+01)

VOLT? MAX

(Response from ZS1406:  
+6.000000E+01)

**VOLTage[:LEVel]:TRIGgered****<num>**

Sets a new value for the triggered load voltage. Not possible in autoranging mode! The desired voltage range must be preset.

When exceeding the allowed setting range the error "Data out of range" is triggered which can be read with *SYSTEM:ERRor?*

In this case the maximum possible value is set and the programmed value is kept as nominal value. If a suited voltage range is set at a later time the load sets the saved nominal value.

Das Triggerereignis wird mit dem Befehl `TRIGger:SOURce` definiert.

Tritt das Triggerereignis ein und Konstantspannungsbetrieb ist eingestellt, stellt das Gerät die vorher programmierte Trigger-Spannung ein, wenn nicht eine dynamische Betriebsart triggerfähig gemacht wurde (s. Subsystem TRIGger).

Als Parameter sind alle Zahlenwerte innerhalb des momentanen Spannungsbereiches erlaubt.

Außerdem sind die speziellen Zahlenparameter MIN und MAX erlaubt. Im Autoranging-Betrieb stellt der Parameter MAX den Endwert im höchsten Strombereich ein, ansonsten den Endwert im momentanen Bereich.

Beispiele: bei Trigger 10V einstellen  
`VOLT:TRIG 5.89`  
`VOLTage:LEVEL:TRIGGERED 20`

Als Dezimaltrennzeichen erwartet das Gerät einen Punkt (.), kein Komma!

### **VOLTage[:LEVel]:TRIGgered?**

Abfrage des triggerbaren Sollwertes im Spannungsbetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

`SD.DDDDDDESDD` S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert im momentanen Spannungsbereich wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Beispiele:

`VOLT:TRIG?` (Antw. z.B.:  
`+1.500000E+01`)  
`VOLT:LEVEL:TRIG? MAX`  
(Antwort von ZS1406: `+6.000000E+01`)

The trigger event is defined using the command `TRIGger:SOURce`.

If the trigger event takes place and the operating mode constant voltage is set, the device sets the programmed trigger voltage unless there is a dynamic mode waiting for a trigger (see subsystem TRIGger).

Allowed parameters are all numeric values within the current range of the particular device type.

The special numeric values MIN and MAX are also allowed.

In autoranging mode the parameter MAX sets the end value in the highest voltage range, otherwise the end value in the active range.

Examples: Set 10V at Trigger  
`VOLT:TRIG 10.0`  
`VOLTage:LEVEL:TRIGGERED 10`

The device expects a point (.) as decimal separator, no comma!

### **VOLTage[:LEVel]:TRIGgered?**

Queries the triggerable set point for operating mode voltage.

The return value is a numeric value in exponential form:

`SD.DDDDDDESDD` S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible value is queried by appending a question mark and the parameters MIN or MAX.

Examples:

`VOLT:TRIG?` (Response e.g.:  
`+1.500000E+01`)  
`VOLT:LEVEL:TRIG? MIN`  
(Response: `+0.000000E+00`)

**VOLTage:MODE FIXed|LIST**

Bestimmt, ob bei einem Triggerereignis die statische Triggerspannung (RESistance:TRIGgered) oder eine programmierte Spannungsliste (LIST:VOLTage (Kurvenform)) eingestellt werden soll.

Nach dem Einschalten der Last ist VOLTage:MODE FIXed gewählt.

S. auch Subsysteme LIST und TRIGger.

Beispiel:

VOLT:MODE LIST

**VOLTage:MODE?**

Spannungs-Trigger-Betriebsart abfragen.

Als Rückgabewert wird die Kurzform des entsprechenden Parameters bereitgestellt (FIX, LIST).

Beispiel:

VOLT:MODE? (Antwort nach Einschalten: FIX)

**VOLTage:PROTection[:LEVel][:LOW] <num>**

Stellt die Trigger-Eingangsspannung ein. Die Triggerspannung kann nicht durch einen separaten Befehl aktiviert bzw. deaktiviert werden.

Durch die Programmierung eines Triggerspannungswertes ist die Triggerspannung auch aktiviert. Soll die programmierte Triggerspannung deaktiviert werden, muss der Sollwert wieder auf den Minimalwert (default) programmiert werden.

Beispiele:

VOLT:PROT 10 (Triggerspannung 10V wird eingestellt)  
VOLT:PROT MIN (minimale Triggerspannung 0V wird eingestellt)

**VOLTage:MODE FIXed|LIST**

Determines if the static trigger voltage (VOLTage:TRIGgered) or a programmed voltage list (LIST:VOLTage (waveform)) shall be set when a trigger event occurs. After power-on VOLTage:MODE FIXed is set.

Example:

VOLT:MODE LIST

See also Subsystems LIST and TRIGger.

**VOLTage:MODE?**

Query voltage trigger mode.

The return value is the short form of the corresponding parameters (FIX, LIST).

Example:

VOLT:MODE? (Response after power-on: FIX)

**VOLTage:PROTection[:LEVel][:LOW] <num>**

Sets the input trigger voltage. The trigger voltage cannot be activated or deactivated by a separate command but is always active.

If the trigger voltage shall be deactivated the nominal value has to be set to the minimum value (default).

Examples:

VOLT:PROT 10 (Trigger voltage 10V is set)  
VOLT:PROT MIN (minimum trigger voltage 0V is set)



Hinweis:

Der Parameter für die Triggerspannung kann nur verändert werden, wenn keine dynamische Betriebsart (LIST, PCYC) aktiv ist.

Der Parameter MAX (VOLT:PROT MAX) stellt einen um ca. 2...5% höheren Wert als den Spannungsendwert ein. Wird VOLT:PROT MAX im Autorangingbetrieb programmiert und das Gerät vollzieht einen automatischen Bereichswchsel, stellt es den Triggerspannungswert auch auf den Maximalwert des neuen Bereiches. Ansonsten gelten die gleichen Regeln wie bei Spannungssettings.

**VOLTage:PROT[:LEVel][:LOW]?**

Abfrage des momentan eingestellten Triggerspannungswertes.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

SD.DDDDDDESDD S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden. Dann sendet das Gerät im Autorangingbetrieb den Maximalwert im größten Bereich, ansonsten den Maximalwert im momentanen Bereich.

Beispiele:

VOLT:PROT? (Antw. nach \*RST:  
+0.000000E+00)

VOLT:PROT? MAX (Antwort z.B.:  
+3.000000E+02)

**VOLTage:PROT:TRIPped?**

Abfrage des Triggerspannungsstatus.

Als Rückgabewert wird eine Boolesche Zahl (0 oder 1) zur Verfügung gestellt.

Note:

The parameter for the trigger voltage is only programmable when no dynamic function (LIST, PCYC) is active.

Parameter MAX (VOLT:PROT MAX) sets the trigger voltage to approximately 2...5% above the voltage range end value. If VOLT:PROT MAX is programmed in autoranging mode and the device performs an automatic range switch then the trigger voltage is set to the maximum of the new range. Additionally, the same rules as with voltage settings are valid.

**VOLTage:PROT[:LEVel][:LOW]?**

Queries the nominal value for the trigger value (no measurement!).

The return value is a numeric value in exponential form:

SD.DDDDDDESDD S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible value is queried by appending a space question mark and the parameters MIN or MAX. If autoranging is active the load sends the maximum value of the highest range, otherwise the maximum value of the active range.

Examples:

VOLT:PROT? (Resp. after \*RST:  
+0.000000E+00)

VOLT:PROT? MAX (Response e.g.:  
+3.000000E+02)

**VOLTage:PROT:TRIPped?**

Queries the status of the trigger voltage.

The response is a Boolean number (0 or 1).

Dabei bedeutet:

0: Spannung oberhalb Triggerspannung  
1: Spannung unterhalb Triggerspannung, Status TV aktiv.

Beispiel:

VOLT:PROT:TRIP? Antwort: 0

**VOLTage:RANGe <num>**

Wählt einen festen Einstellbereich im Spannungsbetrieb.

Der numerische Parameter wird in Volt angegeben und muss innerhalb des Spannungsbereiches des jeweiligen Modells liegen (Technische Daten).

Hinweis:

Die meisten Geräte der ZS Serie verfügen zwar nur über einen Strom- und einen Spannungsbereich, auf den sich auch alle Genauigkeitsangaben und Messsignale beziehen, zur feineren Einstellung für kleine Werte gibt es jedoch die "Erweiterte Einstellauflösung", was in diesem Programmierhandbuch oft mit Einstellbereich bezeichnet wird.

Die ZS Last versucht, den eingestellten Sollwert aufrechtzuerhalten, d.h. wenn ein höherer Spannungsbereich programmiert wird, wird der letzte Spannungswert nach dem Bereichswechsel wieder eingestellt.

Wird ein kleinerer Bereich programmiert und der zuletzt eingestellte Sollwert ist zu groß für den kleineren Bereich, wird die maximal mögliche Spannung eingestellt und ERR2 am Gerät angezeigt. Der Sollwert (VOLTage?) bleibt unverändert.

Die speziellen Zahlenparameter MIN und MAX sind ebenfalls erlaubt.

There means:

0: Input voltage higher than trigger voltage.  
1: Input voltage lower than trigger voltage, Status TV active.

Example:

VOLT:PROT:TRIP? Response: 0

**VOLTage:RANGe <num>**

Sets the setting range for the operating mode voltage.

The numeric parameter has to be within the highest voltage range of the particular device type (technical data).

Note:

Most ZS series devices have got only one current and voltage range which all accuracy data and measurement signals correspond to. Nevertheless, for exact setting of small values there is the "extended resolution" which is called operating range or setting range in this manual because of better understanding.

The ZS load tries to keep the topical nominal value. That means if a higher range is programmed the last setting is kept.

If a lower range is programmed and the last setting is too high for this range then the maximum possible value is set and ERR2 is shown on the load's front panel. The nominal value (VOLTage?) does not change in this case.

The special numeric parameters MIN and MAX are also allowed.

Beispiele:

VOLT:RANG 120 (120V-Bereich  
einstellen)  
VOLTAGE:RANGE MAX

Wenn ein Bereichsparameter angegeben wird, der bei dem betreffenden Gerät nicht vorhanden ist, wird der nächsthöhere Spannungsbereich eingestellt.

Beispiel ZS1406:

VOLT:RANG 10

Da das Gerät über keinen 10V-Bereich verfügt, wird der nächsthöhere Spannungsbereich von 20V eingestellt.

**VOLTage:RANGe:AUTO**

**ON|1|OFF|0**

Schaltet den Autoranging-Modus im Spannungsbetrieb ein oder aus. Die Parameter ON oder 1 schalten Autoranging ein, die Parameter OFF oder 0 schalten Autoranging aus, und es muß der gewünschte Spannungsbereich gewählt werden.

Im Autoranging-Betrieb stellt das Gerät je nach programmiertem Sollwert den optimalen Spannungsbereich ein.

Beim Einschalten des Gerätes ist im Spannungsbetrieb die Autoranging-Funktion deaktiviert.

Beispiel:

VOLT:RANG:AUTO ON

Achtung!

Autoranging ist nicht erlaubt bei einer der folgenden Betriebsarten:

- bei der Definition von Modulationsparametern (TRANsient-Subsystem)
- bei der Definition von frei programmierbaren Kurvenformen (PCYCLE-Subsystem)
- im MPP Tracking Mode

Examples:

VOLT:RANG 120 (Set 120V range)  
VOLTAGE:RANGE MAX

If a range parameter is set which is not available at the respective device the next possible higher range is set.

Example ZS1406:

VOLT:RANG 10

Since the device has no 10V range it sets the next higher voltage range of 20V.

**VOLTage:RANGe:AUTO**

**ON|1|OFF|0**

Switches autoranging in constant voltage mode on or off.

The parameter ON or 1 switches autoranging on, the parameter OFF or 0 switch autoranging off and the desired voltage range has to be programmed.

In autoranging mode due to the programmed setting value the device sets the optimum range.

At power-on the autoranging function in voltage mode is deactivated.

Example:

VOLT:RANG:AUTO ON

Important!

Autoranging is not allowed in one of the following situations:

- when modulation parameters are defined (TRANsient Subsystem)
- when programmable load cycles are defined (PCYCLE Subsystem)
- in MPP Tracking mode

In diesen Fällen muss vorher ein fester Spannungsbereich gewählt werden.

### **VOLTage:RANGe?**

Abfrage des Spannungsbereiches in Volt. Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt.

*SD.DDDDDDESDD* S: Sign(Vorzeichen),  
D: Digit,  
E: Exponent

Der größt- bzw. kleinstmögliche Wert wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

#### Beispiele:

*VOLT:RANG?*

(Antwort z.B.: +3.000000E+02)

*VOLT:RANGe? MAX*

(Antwort von ZS1406: +6.000000E+01)

In these cases you have to preset a fixed voltage range.

### **VOLTage:RANGe?**

Queries the voltage range. The return value is a numeric value in exponential form.

*SD.DDDDDDESDD* S: Sign,  
D: Digit,  
E: Exponent

The highest or lowest possible value is queried by appending a white space and the parameters MIN or MAX to the question mark.

#### Example:

*VOLT:RANG?*

(Response e.g.: +2.000000E+01)

## 7.5 Benutzerorientierte Befehlsfolge

Nachdem die Fernsteuerfunktionen in alphabetischer Reihenfolge aufgeführt worden sind, werden hier nochmals die wichtigsten Befehle in der Reihenfolge dargelegt, wie sie für den Benutzer sinnvoll sind. Genaue Informationen entnehmen Sie bitte der jeweiligen Befehlsbeschreibung in Kap. 7.

### 7.5.1 Gerät in Grundzustand versetzen

**\*RST**

### 7.5.2 Auswahl der Betriebsart

Es kann eine der vier Betriebsarten ausgewählt werden.

Betriebsart Konstant-Strom:

**MODE:CURR**

Betriebsart Konstant-Leistung:

**MODE:POW**

Betriebsart Konstant-Widerstand:

**MODE:RES**

Betriebsart Konstant-Spannung:

**MODE:VOLT**

Nach dem Einschalten des Gerätes ist die Betriebsart Konstant-Strom mit 0A Einstellwert aktiv.

### 7.5.3 Sollwert einstellen

Die Last speichert für jede Betriebsart den zuletzt programmierten Einstellwert, so dass beim Wechsel in eine andere Betriebsart der in dieser Betriebsart zuletzt programmierte Sollwert bzw. der Defaultwert (bei Reset \*RST) eingestellt wird.

Setzen eines Stromwertes:

**CURR <Nrf>**

Setzen eines Leistungswertes:

**POW <Nrf>**

## 7.5 User-Oriented Command Sequence

After the remote operation commands have been presented in alphabetic order they are shown again in a more user-friendly order which may be helpful for programming beginners. Detailed information is given in the respective command description in chapter 7.

### 7.5.1 Resetting the Device

**\*RST**

### 7.5.2 Selecting the Operating Mode

One can choose between constant current or constant resistance mode.

Selecting constant current mode:

**MODE:CURR**

Selecting constant power mode:

**MODE:POW**

Selecting constant resistance mode:

**MODE:RES**

Selecting constant voltage mode:

**MODE:VOLT**

After power-on operating mode current is active with 0A.

### 7.5.3 Setting a Nominal Value

In each operating mode the load saves the corresponding setting. That means if the operating mode is changed the load sets the value which has been programmed last in this mode, or if no value has been programmed yet in this mode it sets the default value (as it is with \*RST).

Setting a nominal current:

**CURR <Nrf>**

Setting a nominal power:

**POW <Nrf>**

Setzen eines Widerstandswertes:

**RES <NRf>**

Setzen eines Spannungswertes:

**VOLT <NRf>**

<NRf> steht für einen Zahlenwert innerhalb des gültigen Bereiches der entsprechenden Betriebsart.

Damit die Last die programmierte Belastung einstellen kann, muss der Eingang noch zugeschaltet werden (s.u.).

#### 7.5.4 Geräteingang zu- und abschalten

Eingang zuschalten:

**INP ON**

Eingang abschalten.

**INP OFF**

#### 7.5.5 Messen von Strom-, Spannungs-, Widerstands- und Leistungswerten

Der momentane Strom, die Spannung, der Widerstand oder die Leistung können gemessen werden. Die ausgelesenen Leistungs- und Widerstandswerte werden aus den gemessenen Strom- und Spannungswerten errechnet.

Messen des Stromes:

**MEAS:CURR?**

Messen der Spannung:

**MEAS:VOLT?**

Messen des Widerstandes:

**MEAS:RES?**

Messen der Leistung:

**MEAS:POW?**

#### 7.5.6 Programmierbare Kurvenform

Mit dem Subsystem LIST können beliebige Lastkurvenformen in Form von aneinandergereihten ansteigenden oder abfallenden Geradenteilstücken definiert werden.

Setting a nominal resistance:

**RES <NRf>**

Setting a nominal voltage:

**VOLT <NRf>**

<NRf> is the numeric value within the valid range of the corresponding operating mode.

The load input must be switched on to enable the device adjusting the current (see below).

#### 7.5.4 Load Input On and Off

Switching input on:

**INP ON**

Switching input off:

**INP OFF**

#### 7.5.5 Measuring Current, Voltage, Resistance and Power Values

The current, voltage, resistance and power value can be measured.

The read power and resistance values are calculated from current and voltage measured.

Current measurement:

**MEAS:CURR?**

Voltage measurement:

**MEAS:VOLT?**

Resistance measurement:

**MEAS:RES?**

Power measurement:

**MEAS:POW?**

#### 7.5.6 Programmable Waveform

The LIST subsystem is available to define arbitrary load characteristics for DC current in form of stringing together rising or falling straight lines.

Zum Programmieren solcher Kurven werden Listen mit Stromwerten, Anstiegs- und Verweilzeiten an das Gerät geschickt.

Diese drei Listen müssen die gleiche Anzahl von Parametern haben.

Senden der Liste mit Stromwerten:

**LIST:CURR <NRf1>,<NRf2>, ...**

Senden der Liste mit Anstiegs- bzw. Abfallzeiten:

**LIST:CURR:RTIM <NRf1>,<NRf2>, ...**

Senden der Liste mit Verweildauern:

**LIST:CURR:DWEL <NRf1>,<NRf2>, ...**

Es kann noch angegeben werden, ob diese Kurve dauernd laufen soll:

**LIST:COUN INF**

oder nur eine bestimmte Anzahl von Zyklen:

**LIST:COUN <NRf>**

(<NRf> steht für die Anzahl der Zyklen)

Lasteingang ein, Starten der Kurvenform:

**INP ON**

**LIST:STAT ON**

### 7.5.7 Messdatenerfassung

Das Gerät verfügt über eine zeitgetriggerte, programmierbare Messwertaufnahme und einen internen Messdatenspeicher für Zeit, Spannung und Strom.

Festlegen des Messintervalls:

**TRIG:TIM <NRf>**

<NRf> steht für einen Zahlenwert [s], der dem Messintervall entspricht.

Starten der automatischen

Messwertaufnahme:

**TRIG:SOUR TIM**

Stoppen der automatischen

Messwertaufnahme:

**TRIG:SOUR IMM**

For programming such curves, lists with current values, rise and dwell times are sent to the device.

The three lists must contain a matching number of parameters.

Sending a list with current values:

**LIST:CURR <NRf1>,<NRf2>, ...**

Sending a list with rise or fall times:

**LIST:CURR:RTIM <NRf1>,<NRf2>, ...**

Sending a list with dwell times:

**LIST:CURR:DWEL <NRf1>,<NRf2>, ...**

Moreover, the list can be run continuously:

**LIST:COUN INF**

or only a certain number of cycles:

**LIST:COUN <NRf>**

(<NRf> is the number of cycles)

Input on, Waveform start:

**INP ON**

**LIST:STAT ON**

### 7.5.7 Data Acquisition

The device integrates a time-triggered, programmable data acquisition with an internal data logger for time, voltage and current.

Determining the measurement interval:

**TRIG:TIM <NRf>**

<NRf> is the measurement interval in s.

Starting data logging:

**TRIG:SOUR TIM**

Stopping data logging:

**TRIG:SOUR IMM**

Anzahl der Messdatensätze abfragen:

**DATA:POIN?**

Die Last antwortet mit einem Zahlenwert, der die Anzahl der abgelegten Datensätze angibt.

Auslesen der Daten und Löschen des Speichers:

**DATA:REM? <NRf>**

(<NRf> gibt die Anzahl der zu lesenden Datensätze an.)

### 7.5.8 Batterieprüf-Funktion

Zur Kapazitäts- bzw. Energieermittlung von Batterien steht die Batterieprüffunktion zur Verfügung.

Sobald der Laststrom eingeschaltet wird, beginnt die Last, die Ladung (Ah) und die Energie (Wh) zu kumulieren, bis die Eingangsspannung unter die programmierte Endspannung fällt oder der Lasteingang ausgeschaltet wird. In ersterem Fall schaltet die Last den Lasteingang automatisch aus.

Bei erneutem Einschalten des Lasteingangs beginnt die Kumulation wieder bei 0.

**SFUN:BATT:TEV <NRf>**

(<NRf> gibt die Entladeschlussspannung in V an.)

**SFUN:BATT:ENAB ON|OFF**

(Erlauben bzw. Verboten der Batterietestfunktion bei INP ON)

**MEAS:CHAR?**

(Momentan entladene Kapazität abfragen)

**MEAS:ENER?**

(Momentan entladene Energie abfragen)

Query number of measuring records:

**DATA:POIN?**

The load responds with the number of stored data records.

Reading records and deleting memory:

**DATA:REM? <NRf>**

(<NRf> is the number of records being read.)

### 7.5.8 Battery Test Function

To determine the capacity or energy of batteries the battery test function is available.

As soon as the load input is switched on the load starts accumulating the charge (Ah) and the energy (Wh) until the input voltage decreases below the programmed test-end voltage or until the input is switched off. In the first case the load automatically switches the input off. When the load input is reactivated the Ah/Wh cumulators restart with 0.

**SFUN:BATT:TEV <NRf>**

(<NRf> defines the test-end voltage in V.)

**SFUN:BATT:ENAB ON|OFF**

(Enables or disables the battery test function at INP ON)

**MEAS:CHAR?**

(Queries currently discharged capacity)

**MEAS:ENER?**

(Queries currently discharged energy)



Beispiel:

CURR:RANG 20  
 CURR 5  
 SFUN:BATT:TEV 1.25  
 TRIG:TIM 30  
 TRIG:SOUR TIM  
 SFUN:BATT:ENAB ON  
 INP ON  
 ...  
 SFUN:BATT:STAT?  
 MEAS:CHAR?  
 SFUN:BATT:ENAB OFF  
 DATA:POIN?  
 DATA:REM? 655

**7.5.9 Externe Ansteuerung**

Bei externer Ansteuerung kann über den Analog-I/O-Stecker mit einem Analogsignal von 0...10V angesteuert werden (s. Hardwareteil).

**SYST:CONT EXT**

**7.5.10 Volle Lüfterleistung**

Im Normalbetrieb werden die Gerätelüfter strom- und temperaturabhängig geregelt.

Der Befehl

**SYST:FAN FULL**

schaltet auf dauernde Maximalleistung der Lüfter. Mit

**SYST:FAN AUTO**

oder Reset (\*RST) wird wieder auf automatische Lüfterregelung geschaltet.

**7.5.11 Regelgeschwindigkeit umschalten**

Bei schwingenden Systemen kann es von Vorteil sein, die Regelzeitkonstante der Last auf SLOW oder MEDIUM (default) zu schalten.

**SYST:SPE SLOW**

**SYST:SPE MED**

**SYST:SPE FAST**

Example:

CURR:RANG 20  
 CURR 5  
 SFUN:BATT:TEV 1.25  
 TRIG:TIM 30  
 TRIG:SOUR TIM  
 SFUN:BATT:ENAB ON  
 INP ON  
 ...  
 SFUN:BATT:STAT?  
 MEAS:CHAR?  
 SFUN:BATT:ENAB OFF  
 DATA:POIN?  
 DATA:REM? 655

**7.5.9 External Control**

In external control mode the load can be controlled with an analogous voltage of 0...10V via the Analog I/O connector (see hardware manual).

**SYST:CONT EXT**

**7.5.10 Full Fan Power**

In normal mode the load's fans are controlled current and temperature-controlled.

The command

**SYST:FAN FULL**

switches to maximum fan power.

With

**SYST:FAN AUTO**

or reset (\*RST) the load is switched to automatic fan control again.

**7.5.11 Switching Control Speed**

It may be an advantage to switch the control speed of the load to SLOW or MEDIUM (default) speed if there is an oscillating system.

**SYST:SPE SLOW**

**SYST:SPE MED**

**SYST:SPE FAST**

### 7.5.12 Software-Watchdog einstellen

Die Geräte der Serie ZS sind mit einer watchdog-ähnlichen Software-Funktion ausgestattet. Diese bringt bei eventuellem Absturz des Steuerrechners, Fehlbedienung der Steuersoftware o.ä. die Elektronische Last in einen sicheren Betriebszustand, indem es den Lasteingang abschaltet, wenn nach einer definierten Zeit kein gültiges Kommando im Gerät angekommen ist.

Die zu überwachende Zeit wird mit dem Befehl

**SYST:PROT <NRf>**

definiert. Der Parameter <NRf> steht für den zu überwachenden Zeitraum in Sekunden.

Die Aktivierung erfolgt mit

**SYST:PROT:STAT ON**

Deaktivierung mit

**SYST:PROT:STAT OFF**

Abfrage des Auslösezustandes des Software-Watchdogs:

**SYST:PROT:TRIP?**

Als Rückgabewert wird eine Boolesche Zahl (0 oder 1) zur Verfügung gestellt.

Dabei bedeutet:

0: Watchdog hat nicht eingegriffen

1: Watchdog hat Lasteingang abgeschaltet.

### 7.5.13 Steuerung der Power-I/O-Karte (Option ZS07)

Zur Steuerung der optionalen Power-I/O-Karte werden Systemparameter-Befehle verwendet. Der 8Bit-Relaisport wird mit Parameter 82 gesteuert und gelesen:

**SYST:PAR 82, <NRf>**

**SYST:PAR 82?**

<NRf> steht für eine binär codierte Dezimalzahl und programmiert den ganzen Port.

Der 8Bit-Inputport wird mit Parameter 83 gelesen:

**SYST:PAR 83?**

### 7.5.12 Setting the Software Watchdog

The ZS series Electronic Loads are suited with a watchdog-like software function. This brings the load into a safe mode by switching the input off in the case that the load doesn't receive a valid command within a defined time. This may be caused by a PC hang-up or a program's misapplication.

The time being monitored for a valid command is set with the command

**SYST:PROT <NRf>**

<NRf> is the time in seconds within which a valid command shall be received.

The watchdog is activated by

**SYST:PROT:STAT ON**

Deactivation with

**SYST:PROT:STAT OFF**

Querying the trip state of the software watchdog:

**SYST:PROT:TRIP?**

A Boolean number (0 or 1) is returned as response.

There means:

0: Watchdog has not tripped

1: Watchdog has tripped and switched the input off

### 7.5.13 Controlling the Power I/O Card (Option ZS07)

System parameter commands are used to control the optional Power I/O Card. The 8Bit relais port is written and read by parameter 82:

**SYST:PAR 82, <NRf>**

**SYST:PAR 82?**

<NRf> is a binary coded decimal number and programs the whole port.

The 8Bit input port is read with parameter 83:

**SYST:PAR 83?**

Beispiel:

SYST:PAR 82,19 (Relais 1, 2 und 5 ein,  
alle anderen aus)  
SYST:PAR 83? (Antw. z.B.:  
+3.200000E+01  
Bit 6 high,  
alle anderen low)

Example:

SYST:PAR 82,19 (Relais 1, 2 and 5 on,  
all others off)  
SYST:PAR 83? (Resp. e.g.:  
+3.200000E+01  
Bit 6 high,  
all others low)

**7.5.14 Funktionen mit Option ZS13  
(Data Acquisition)****7.5.14.1 Abfrage der Option ZS13**

Ob die Option ZS13 im betreffenden Gerät installiert ist, kann durch Abfrage des Geräteparameters 60 ermittelt werden:

**SYST:PAR 60?**

Ist das zweitniederwertigste Bit der binär codierten Antwort-Dezimalzahl gesetzt, ist Option ZS13 vorhanden.

**7.5.14.2 Schnellen AD-Wandler verwenden**

Nur mit Option ZS13!

Die Option ZS13 (Data Acquisition) beinhaltet einen schnellen AD-Wandler, der anstelle des defaultmäßig verwendeten langsameren ADC aktiviert werden kann:

**SET:ADC FAST**

Nach dem Einschalten und nach Reset ist wieder der langsame ADC aktiv.

**7.5.14.3 Externspannung messen**

Nur mit Option ZS13!

Wenn der schnelle AD-Wandler aktiv ist, kann außer den Standard-Größen Strom, Spannung, Leistung, Widerstand noch eine Externspannung von 0 ... 10V gemessen werden.

**SET:ADC FAST****MEAS:EXT?****7.5.14 Functions with ZS13 Option  
(Data Acquisition)****7.5.14.1 Query for Option ZS13**

To find out if ZS13 option is installed in the concerning device parameter 60 can be read out:

**SYST:PAR 60?**

If the second lowest bit of the binary-coded decimal number sent as response is set then ZS13 is available.

**7.5.14.2 Using the Fast A/D Converter**

Only with ZS13 option!

Option ZS13 (Data Acquisition) includes a fast A/D converter which may be used instead of the slower one used by default:

**SET:ADC FAST**

After power-on and after reset the slow ADC becomes active again.

**7.5.14.3 Measuring External Voltage**

Only with Option ZS13!

If the fast ADC is active apart from the standard magnitudes current, voltage, power and resistance, an external voltage between 0 ... 10V may be measured.

**SET:ADC FAST****MEAS:EXT**

#### 7.5.14.4 Maximum Power Point Tracking

Die Elektronischen Lasten der Serie ZS können durch die Option ZS13 mit der Funktion MPP Tracking (Maximum Power Point Tracking) ausgerüstet werden.

MPP Tracking wird beim Betrieb von Solarpanels verwendet, um dem Panel die größtmögliche Leistung zu entnehmen.

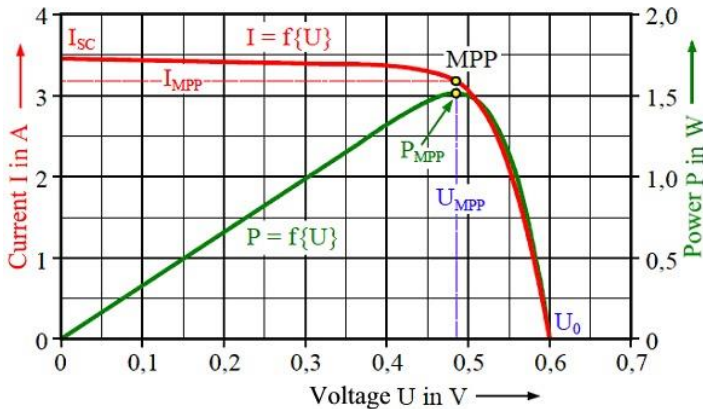
Durch die Kennlinie eines Solarpanels ergibt sich dieser Betriebspunkt (MPP). Die MPP Funktion der Elektronischen Last sucht diesen Punkt und hält ihn auch bei veränderlicher Sonneneinstrahlung bei.

#### 7.5.14.4 Maximum Power Point Tracking

The Electronic Loads of Series ZS can be equipped with Option ZS13 for MPP (Maximum Power Point) Tracking.

MPP Tracking is used for operating solar panels to get the maximum possible power from the panel.

The MPP results from the diagram of a solar panel. The MPP function of the load seeks the MPP and holds it also at variable solar radiation.



#### 7.5.14.4.1 Funktion des MPP Trackings

Um die MPP Funktion der Elektronischen Last richtig nutzen zu können, ist es erforderlich, das Regelprinzip zu verstehen.

Die Elektronische Last arbeitet im MPP Betrieb mit Konstant-Spannungs-Betrieb. Wird die Funktion aktiviert, startet die Last mit einer höheren Spannung als die max. Spannung des Solarpanels und regelt die Spannung nach unten.

Wenn die vorgegebene Spannung an der Elektronischen Last die aktuelle Spannung des Solarpanels unterschreitet, beginnt Strom zu fließen.

Die Elektronische Last misst kontinuierlich Spannung und Strom und errechnet die aufgenommene Leistung. Solange die Leistung steigt, wird die Richtung der Spannungsänderung beibehalten. Fällt die Leistung (wenn die Einstellungen das Leistungsmaximum überschritten haben), wird die Richtung der Spannungsänderung umgekehrt, so dass die Leistung erneut steigt.

Die Elektronische Last regelt dadurch immer über den MPP.

Wie weit die Elektronische Last um den MPP regelt, kann durch Eingabe des System Parameters Nr. 53 „Delta\_P\_min“ bestimmt werden.

Weiterhin benötigt die Elektronische Last noch Angaben über:

#### 7.5.14.4.1 How does MPP Tracking work?

To use the MPP function of the load it is necessary to understand how MPP Tracking works.

In MPP mode the Electronic Load works in constant voltage mode.

When MPP is activated the load begins with setting a higher voltage than the output voltage of the panel and reduces the voltage stepwise downwards.

When the voltage setting is lower than the voltage of the solar panel current flow will begin.

The Electronic Load continuously measures current and voltage and calculates the resulting power. As long as the power increases the direction of the voltage change is retained. When the power begins to decrease (the settings have passed the MPP) the direction of the voltage run is reversed to cause power to increase again.

In this way the Electronic Load is regulating continuously over the MPP.

How far the Electronic Load deviates from the MPP can be set by system parameter No. 53 "Delta\_P\_Min".

In addition the Electronic Load needs some information about:

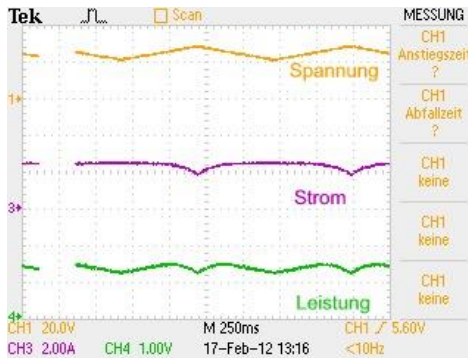
			System Parameter
Nennleistung des Panels	nominal panel power	"Pmax"	48
Leerlaufspannung des Panels	unloaded panel voltage	"VOC"	49
Max. Kurzschlussstrom	max. short circuit current	"ISC"	50
Spannung im MPP Punkt	expected voltage at MPP	"VPM"	51
auftretender Strom im MPP Punkt	expected current at MPP	"IPM"	52
Leistungsregelbereich	Power regulation range	„Delta_P_min“	53

Diese Parameter werden benötigt, damit die Elektronische Last den passenden Strombereich einstellt und den MPP Punkt möglichst schnell findet.

Die Parameter werden nichtflüchtig im Gerät gespeichert und stehen nach dem Aus- und Einschalten der Netzspannung wieder zur Verfügung.

The parameters are required that the load can select the suitable current range and that it can find the MPP as fast as possible.

These parameters are stored in the non-volatile memory of the Electronic Load and are also available after power off and on.



Im Diagramm sind Spannung (oben), Strom (mitte) und Leistung (unten) aufgezeichnet.

Die Elektronische Last verstellt die Spannung dauernd nach oben oder unten bis die Leistung um den angegebenen „Delta\_P\_min“ Wert abweicht. Dann ändert die Spannung die Richtung, bis die gleiche Abweichung in der anderen Richtung auftritt.

Bei hohen „Delta\_P\_min“-Werten ist deutlich der dreieckförmige Spannungsverlauf zu sehen.

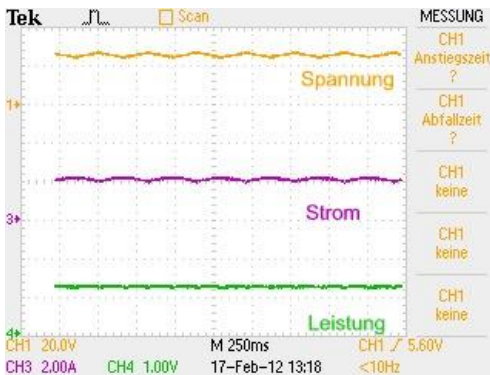
Die obere Spitze der Leistungskurve stellt den MPP Wert dar.

The diagram shows voltage (upper curve), current (in the middle) and power (lower curve).

The Electronic load varies the voltage continuously up or down till the power deviates from the set “delta\_P\_min” value. Then the voltage changes the direction until the same deviation is achieved in the opposite direction.

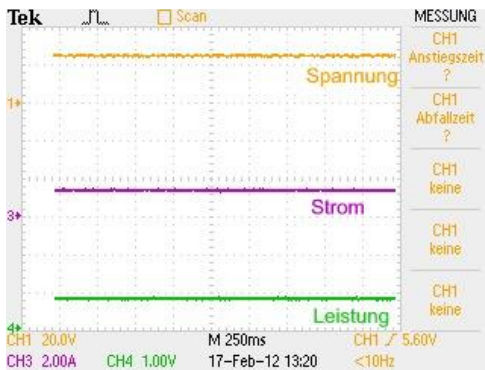
At high numbers for “delta\_P\_min” the triangle shaped voltage is recognisable.

The upper peak of the power diagram is the MPP.



Wird der „Delta\_P\_min“-Wert reduziert (hier um Faktor 10 gegenüber dem Diagramm zuvor), reduziert sich die Amplitude der Dreiecksspannung und die Leistungskurve wird ebenfalls glatter. Die Elektronische Last arbeitet sehr nahe am MPP Wert.

When “delta\_P\_min” is reduced (here by factor 10 against the diagram before), the amplitude of the triangle shaped voltage will be reduced and the power curve will also get smoother. Now the Electronic Load is working very close to the MPP.

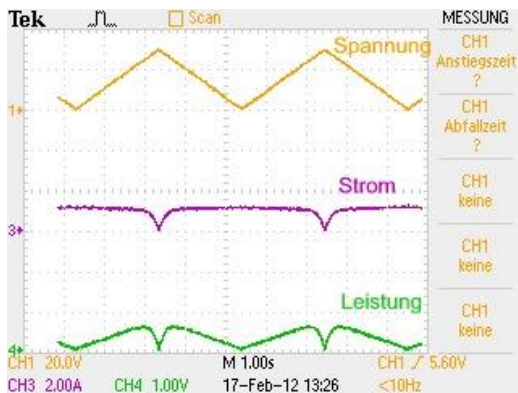


Bei einer weiteren Verringerung von „Delta\_P\_min“ treten an der Spannung kaum mehr Änderungen durch den Regelvorgang auf und die Leistung wird glatt auf dem MPP Punkt gehalten.

At further reduction of “delta\_P\_min” there is nearly nothing to see from the regulation process and the power is kept exactly on the MPP.

7.5.14.4.2 Grenzen des MPP-  
Betriebes

## 7.5.14.4.2 Limits of the MPP Mode



Wenn „Delta\_P\_min“ für ein Panel zu groß eingestellt wird, so dass das Panel diese Leistung nicht liefern kann, so wechselt die MPP Regelung dauernd zwischen Leerlauf und Kurzschluss des Panels.

Dieser Fall kann auch auftreten, wenn das Panel abgeschattet wird oder nur eine geringe Einstrahlung vorhanden ist.

Der MPP-Betrieb kann auch nicht aufrecht erhalten werden, wenn die Elektronische Last an ihre Leistungsgrenzen kommt (max. Spannung, max. Strom des jeweiligen Bereiches, oder max. Leistung).

When “delta\_P\_min” is set too high so that the panel can not supply sufficient power, MPP regulation will permanently change between open circuit and short circuit.

This can happen when the panel is shadowed or when there is only a low radiation available.

The MPP Mode can also not be kept when the load is working at its limits (max. voltage, max. current of the selected range, or max. power)



#### 7.5.14.4.3 Messdatenerfassung während des laufenden MPP-Betriebes

Werden während des MPP-Betriebes am Panel Spannung und Strom gemessen, so ist dabei zu beachten, dass aufgrund der MPP-Regelfunktion Spannung und Strom nicht dauernd konstante Werte aufweisen, sondern in Abhängigkeit von der „Delta\_P\_min“-Einstellung schwanken. (Siehe Diagramme zum Kurvenverlauf bei verschiedenen Delta\_P\_min-Werten).

Ist „Delta\_P\_min“ sehr groß eingestellt (in Relation zu der tatsächlichen Panelleistung), so verschlechtern sich die gemessenen Leistungswerte.

#### 7.5.14.4.3 Data Acquisition during running MPP Mode

When voltage, current and power are measured during MPP Mode one to consider that resulting from the running regulation process the numbers are not constant but change depending on the setting of "delta\_P\_min". (See waveform diagrams for different settings of "delta\_P\_min")

When "delta\_P\_min" is very high (referred to the actual power of the panel) the measured numbers for power will grow worse.

#### 7.5.14.4.4 Einstellen des passenden Gerätestrombereiches für den Prüfling

Die ZS Elektronischen Lasten verfügen teilweise über bis zu 4 Einstellbereiche. Beim Betrieb des MPP Trackings muss der Einstellbereich so gewählt werden, dass der max. Kurzschlussstrom bei höchster Einstrahlung noch abgedeckt wird. Dies geschieht durch Programmieren des Parameters 50 „ISC“ (Kurzschlussstrom des Panels). Das Gerät stellt dann bei Start des MPP Trackings den geeigneten Strombereich ein und behält diesen bei.

Bei geringer Einstrahlung auf das Panel kann es jedoch vorkommen, dass ein deutlich geringerer Strom fließt und damit die Wahl eines kleineren Strombereiches sinnvoll wäre. In diesem Fall muss der „IPM“ Wert entsprechend geringer programmiert werden, damit das Gerät einen kleineren Strombereich auswählt. Überprüfen Sie dazu, welche Strombereiche die Elektronische Last zur Verfügung stellt.

Wenn in diesem Fall jedoch unerwartet die volle Einstrahlung auftritt, kann es vorkommen, dass die Elektronische Last einen höheren Strom als den möglichen Maximalstrom des eingestellten Bereiches einstellen müsste, um den MPP zu halten.

Die Elektronische Last signalisiert dann Überstrom „OCP“ (Überstrombegrenzung). Der „OCP“ Status sollte deswegen vom Steuerprogramm laufend ausgelesen werden, um richtig reagieren zu können (Wechsel auf einen höheren Strombereich).

#### 7.5.14.4.4 Determination of the suitable current range for the device under test

The ZS Electronic Loads have up to 4 setting ranges.

At MPP mode that setting range has to be chosen which will cover the max short circuit current at highest radiation.

This is done by programming parameter 50 “ISC” (short circuit current of the panel).

When starting MPP mode the Electronic Load will set the appropriate current range and keep it while MPP is running.

At low radiation it can happen that the current is much less and that the selection of a smaller current range would be useful.

In this case the value for “IPM” has to be reprogrammed to select a smaller current range. Please check what current ranges the Electronic load provides.

When suddenly a full radiation appears then a higher current is required to keep the MPP than that of the selected current range.

Then the Electronic Load will signalize “OCP” (over current protection).

OCP should therefore be continuously checked by the control software to response by changing the current range.

#### 7.5.14.4.5 Ermitteln des Delta\_P\_Min-Parameters für ein Panel

Der „Delta\_P\_Min“ Parameter gibt an, wie weit die Regelung um den zuletzt ermittelten MPP Punkt abweichen darf, um den MPP bei jedem Regelvorgang neu zu ermitteln.

Bei einem Neugerät wurde Delta\_P\_Min werksseitig auf den Wert  $\Delta_{P\_Min} = 0.0002 *$  *Gerätesspannung* \* *max. Gerätestrom* gestellt.

Beispiel: ZS530-3:  
 $0.0002 * 300V * 12A = 0,72W$

Diese Einstellung ist für ein Panel ab 50W gut geeignet. Beim MPP Tracking weicht der tatsächliche Wert dann nur 0.72W vom MPP Punkt ab.

Bei leistungsschwächeren Panels sollte aber ein kleinerer Wert eingestellt werden, da sich die MPP Regelung sonst zu weit vom tatsächlichen MPP Punkt entfernt und die Leistung im Mittel dann deutlich darunter liegt.

Der Wert für „Delta\_P\_Min“ sollte sinnvollerweise auch geändert werden, wenn dauerhaft nur eine schwache Einstrahlung gegeben ist. Wenn ein Panel nur mehr einen Bruchteil seiner Leistung liefert, sollte der „Delta\_P\_Min“ Wert angepasst werden. Entsprechend ist auch bei einem Strombereichswechsel wegen zu geringer Einstrahlung der Wert für Delta\_P\_Min zu korrigieren.

Wird Delta\_P\_Min auf 0 gestellt, so benutzt das Gerät intern den technisch kleinstmöglichen Wert.

Der Delta\_P\_Min Wert ist nichtflüchtig im Gerät gespeichert und bleibt nach dem Aus- und Einschalten der Netzspannung erhalten.

#### 7.5.14.4.5 Evaluation of the „delta\_P\_min“ parameter of the panel

The „Delta\_P\_min“ parameter defines the permitted deviation to the MPP being determined at last to calculate the present MPP.

For a new device „delta\_P\_min“ is set to:

$$\Delta_{P\_min} = 0.0002 * \text{max. voltage} * \text{max. current range}$$

Example: ZS530-3:  
 $0.0002 * 300V * 12A = 0,72W$

This setting is well suited for a panel with about 50W. In MPP mode the regulation will deviate 0.72W from the MPP.

For panels with lower power a smaller number for „Delta\_P\_min“ should be chosen because the MPP Regulation would work too far from the real MPP point and the mean value of the power would be below.

The number for „Delta\_P\_min“ should also be changed when there is only low radiation

When a panels supplies only a small part of its nominal power „Delta\_P\_min“ should be matched.

The current range should also be matched when possible.

When „Delta\_P\_min“ is set to 0, the unit works at the smallest possible number for „delta\_P\_min“.

The „Delta\_P\_min“ setting is kept in the non volatile storage of the Electronic Load.

**7.5.14.4.6 MPP Tracking  
Programmierbeispiel**

Technische Daten des Panels:

Nennleistung P: 224 [W]

Spannung bei  
Nennleistung: 29,2 [V]Strom bei  
Nennleistung: 7,68 [A]

Leerlaufspannung: 36,8 [V]

Kurzschlussstrom: 8,09 [A]

Befehle:

```

SYSTEM:PARAmeter 48,224
SYSTEM:PARAmeter 49,36.8
SYSTEM:PARAmeter 50,8.09
SYSTEM:PARAmeter 51,29.2
SYSTEM:PARAmeter 52,7.68
SYSTEM:PARAmeter 53,0.5
MODE:VOLT
MODE:MPP
INP ON

```

**Hinweise:**

Zur Programmierung der Parameter  
siehe auch:  
Subsystem *SYSTEM:PARAmeter*

Der MPP Tracking Mode wird mit jedem  
anderen MODE-Befehl (MODE:CURR,  
MODE:VOLT, MODE:POW, MODE:RES)  
wieder deaktiviert, vorher muss jedoch  
der Lasteingang ausgeschaltet werden  
(*INP OFF*).

**7.5.14.4.6 MPP Tracking  
Programming Example**

Technical Data of the panel:

Nominal Power: 224 [W]

Voltage at  
Nominal Power: 29.2 [V]Current at  
Nominal Power: 7.68 [A]

Voltage unloaded: 36.8 [V]

Short Circuit  
Current: 8.09 [A]

Commands:

```

SYSTEM:PARAmeter 48,224
SYSTEM:PARAmeter 49,36.8
SYSTEM:PARAmeter 50,8.09
SYSTEM:PARAmeter 51,29.2
SYSTEM:PARAmeter 52,7.68
SYSTEM:PARAmeter 53,0.5
MODE:VOLT
MODE:MPP
INP ON

```

**Hints:**

For programming the Parameters see  
also:  
Subsystem *SYSTEM:PARAmeter*

The MPP Tracking mode is deactivated  
with any of the other MODE commands  
(MODE:CURR, MODE:VOLT,  
MODE:POW, MODE:RES), but  
previously the input must be switched off  
(*INP OFF*).

#### 7.5.14.4.7 Messen von Ampere- stunden und Wattstunden

Bei MPP Tracking kumuliert die Last automatisch die Ladung in Ah und die Energie in Wh. Diese beiden Werte können mit `MEASure:CHARge?` bzw. `MEASure:ENERgy?` ausgelesen werden (s. Subsystem MEASure).

#### 7.5.14.5 Exponentialfunktion

Nur mit Option ZS13!

Mit der Exponentialfunktion kann das Einschaltverhalten von Prüflingen untersucht werden. Dazu wird mit dem folgenden Befehl der Parametersatz der E-Funktion definiert.

**SFUN:EXP <Peak>,<End>,<Tau>,  
<Dwell>,<SampleRate>**

Definiert die Parameter für eine exponentiell abfallende Lastspitze.

Das Starten der Exponentialfunktion wird ausschließlich durch die Eingangsspannung der Last bestimmt, die den zuvor programmierten Triggerwert überschreiten muss.

Programmieren der Triggerspannung:

**VOLT:PROT <NRF>**

Damit die Last bei Überschreitung der Triggerspannung die Exponentialfunktion ausführt, muss dies erst noch erlaubt werden:

**SFUN:EXP:ENAB ON**

Der Parameter ON bzw. 1 erlaubt die E-Funktion bei  $U_{in} > TV$  (die Funktion selbst wird durch den Befehl noch nicht gestartet). Der Parameter OFF bzw. 0 verhindert die Ausführung, d.h. auch wenn die Eingangsspannung die programmierte Triggerspannung überschreitet, wird die E-Funktion nicht ausgeführt.

#### 7.5.14.4.7 Measuring Ah and Wh

In MPP Tracking mode the load automatically calculates the charge in Ah and the energy in Wh. These two values can be read from the device with `MEASure:CHARge?` or `MEASure:ENERgy?`, respectively (see subsystem MEASure).

#### 7.5.14.5 Exponential Function

Only with ZS13 option!

With the exponential function you can investigate the turn-on characteristics of UUTs. To define the e-function the parameters of the following command are used.

**SFUN:EXP <Peak>,<End>,<Tau>,  
<Dwell>,<SampleRate>**

Defines the parameters of an exponential decreasing peak load.

The starting of the exponential function is exclusively started by the load's input voltage which must exceed the previously programmed trigger voltage.

Programing the trigger voltage:

**VOLT:PROT <NRF>**

To cause the load to generate the exponential function when the trigger voltage is exceeded you must allow this by the following command:

**SFUN:EXP:ENAB ON**

The parameter ON or 1 enables the e-function at  $U_{in} > TV$  (the function itself is not yet started by this command). The parameter OFF or 0 disables the execution, that means even when the input voltage exceeds the programmed trigger voltage the e-function won't be executed.

Beispiel: Programmierung einer e-Funktion mit 45A Spitzenstrom, 10A Endstrom, 1s Tau, 50ms Samplerate, 5V Triggerspannung.

Befehlsfolge:

```
CURR:RANG 50;:CURR 0;MODE:CURR  
VOLT:PROT 5  
SFUN:EXP 45,10,1,0,0.05  
SFUN:EXP:ENAB ON  
INP ON
```

Details entnehmen Sie bitte der ausführlichen Befehlsbeschreibung.

Example: Programming an e-function with 45A peak value, 10A end value, 1s Tau, 50ms sample rate, 5V trigger voltage.

Commands:

```
CURR:RANG 50;:CURR 0;MODE:CURR  
VOLT:PROT 5  
SFUN:EXP 45,10,1,0,0.05  
SFUN:EXP:ENAB ON  
INP ON
```

Please see the comprehensive command description for details.

## 8 Software-Tools

Sie finden die aktuellen Software-Tools, Treiber und Handbücher zum Download auf unserer Homepage unter:

**[www.hoecherl-hackl.de](http://www.hoecherl-hackl.de)**

Drücken Sie den „Downloads“ Button und wählen Sie das Feld „Software/Treiber“ aus.

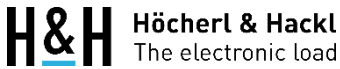
## 8 Software Tools

You will find the actual software tools, drivers and manuals for download on our website under:

**[www.hoecherl-hackl.com](http://www.hoecherl-hackl.com)**

Select the “Software/Manuals/Driver” button on the “Downloads” Section

## 9 Hersteller/Kontakt:



Höcherl & Hackl GmbH  
Industriestr. 13  
94357 Konzell  
Germany

## 9 Manufacturer/Contact

Phone: (+49) 9963 94301 - 0  
Fax: (+49) 9963 94301 - 84  
E-Mail: [support@hoecherl-hackl.com](mailto:support@hoecherl-hackl.com)  
Web: [hoecherl-hackl.com](http://hoecherl-hackl.com)

Ihr Ansprechpartner / your contact:

The Schulz-Electronic logo consists of a stylized graphic of overlapping triangles in blue, white, and orange, followed by the text 'Schulz Electronic' in a bold, sans-serif font. Below the logo, the company name 'Schulz-Electronic GmbH' is listed, along with its address 'Dr.-Rudolf-Eberle-Straße 2, D-76534 Baden-Baden', phone number '+49 7223 9636 0', fax number '+49 7223 9636 90', email 'vertrieb@schulz-electronic.de', and website 'www.schulz-electronic.de'.

**Schulz-Electronic GmbH**  
Dr.-Rudolf-Eberle-Straße 2  
D-76534 Baden-Baden  
Fon + 49 7223 9636 0  
Fax + 49 7223 9636 90  
[vertrieb@schulz-electronic.de](mailto:vertrieb@schulz-electronic.de)  
[www.schulz-electronic.de](http://www.schulz-electronic.de)