



Höcherl & Hackl GmbH
Industriestr. 13
94357 Konzell
Germany

Phone: (++49) 9963 94301 - 0
Fax: (++49) 9963 94301 - 84
eMail: support@hoecherl-hackl.com
Internet: www.hoecherl-hackl.com

Handbuch/Manual Revision: ZSAC 0517-11
gültig für/valid for Firmware: ZS_SCPI_04.00 ff
ZS_AI_04.03 ff

Höcherl & Hackl GmbH
Elektronische Lasten Serie ZSAC
Electronic Loads ZSAC Series
Programmierung/Programming Manual



Inhalt

1	Allgemeines	6
1.1	Ein- und Ausschalten.....	6
1.2	Remote und Local	6
2	Serielle Schnittstelle	7
2.1	Einstellen der RS232-Parameter	7
2.2	USB-Schnittstelle	10
2.3	Datenformat bei RS232 und USB	10
3	IEEE 488 Interface	12
3.1	Einstellen der IEEE 488-Geräteadresse	12
3.2	Datenformat bei IEEE 488.....	13
4	LAN Interface.....	15
4.1	Einsatzmöglichkeit	15
4.2	Anschluss	15
4.3	Konfiguration.....	16
5	Systembus.....	24
6	Unteradressen	25
6.1	Einzeladressierung	25
6.2	Blockadressierung	27
6.3	Systemadressierung	29
7	SCPI-Syntax Einführung	30
7.1	Common Commands.....	30

7.2	Gerätespezifische Befehle.....	30
7.2.1	Aufbau des Headers.....	30
7.2.1.1	Einrückungen	31
7.2.1.2	Aliases	31
7.2.2	White Space.....	32
7.2.3	Lang- und Kurzform, Groß- und Kleinschreibung.....	32
7.2.4	Wahlweise einfügbare Schlüsselwörter	33
7.2.5	Parameter.....	33
7.2.5.1	Zahlenwerte <NRf>	34
7.2.5.2	Einheiten und Multiplizierer.....	34
7.2.5.3	Zahlenwerte und Extremwerte <num>	35
7.2.5.4	Boolesche Parameter.....	36
7.2.5.5	Text	37
7.2.6	Benutzung des Semikolons	37
7.2.7	Abfragebefehle (Queries).....	39
7.3	SCPI Befehlsübersicht.....	41
7.3.1	Common Commands	41
7.3.2	Gerätespezifische Befehle der Serie ZSAC.....	42
7.4	SCPI Befehlsbeschreibung.....	46
7.4.1	Common Commands	46
7.4.2	Gerätespezifische Befehle	50
7.4.2.1	Schnelleinstieg.....	50
7.4.2.2	Subsystem CHANnel INSTrument.....	52
7.4.2.3	Subsystem CURRent	54
7.4.2.4	Subsystem DATA TRACe	60
7.4.2.5	Subsystem DELay.....	63
7.4.2.6	Subsystem GTL	64
7.4.2.7	Subsystem INPut OUTPut	65
7.4.2.8	Subsystem LIST.....	66
7.4.2.9	Subsystem MEASure.....	75
7.4.2.10	Subsystem MODE FUNctIon.....	79
7.4.2.11	Subsystem PROGram.....	81
7.4.2.12	Subsystem RESistance.....	85
7.4.2.13	Subsystem SETUp.....	91
7.4.2.14	Subsystem SFUNctIon.....	92
7.4.2.15	Subsystem STATus	102
7.4.2.16	Subsystem SYSTem.....	112
7.4.2.17	Subsystem TRIGger.....	125
7.5	Benutzerorientierte Befehlsfolge.....	129
7.5.1	Gerät in Grundzustand versetzen.....	129
7.5.2	Auswahl der Funktionsart AC, DC, LINE.....	129
7.5.3	Auswahl der Betriebsart Strom, Widerstand.....	129
7.5.4	Stromwert einstellen.....	129
7.5.5	Widerstandswert einstellen	130
7.5.6	Geräteeingang zu- und abschalten.....	130
7.5.7	Messen von Strom-, Spannungs-, Widerstands- und Leistungswerten.....	130
7.5.8	Freiprogrammierbare AC-Kurvenform.....	131
7.5.9	Programmspeicher.....	131
7.5.10	Stromkurve mit Phasenanschnitt	133

7.5.11	Stromkurve mit Oberwellen (Fouriersynthese).....	133
7.5.12	Stromkurve mit Crestfaktor.....	133
7.5.13	Phasenverschiebung der Stromkurve.....	134
7.5.14	Speichern einer Kurvenform in den Internspeicher	134
7.5.15	Programmierbare Kurvenform im DC- oder Hüllkurve im AC-Betrieb.....	135
7.5.16	Messdatenerfassung	136
8	Hersteller.....	137

Table Of Contents

1	General Information	6
1.1	Power On and Off	6
1.2	Remote and Local.....	6
2	Serial Interface	7
2.1	Setting the RS232 Parameters.....	7
2.2	USB Interface	10
2.3	Data Format of the RS232 Interface.....	10
3	IEEE 488 Interface	12
3.1	Setting the IEEE488 Device Address.....	12
3.2	Data Format of the IEEE 488 Interface.....	13
4	LAN Interface.....	15
4.1	Possible Application	15
4.2	Connection	15
4.3	Configuration.....	16
5	System Bus.....	24
6	Sub Addresses	25
6.1	Single Addressing.....	25
6.2	Block Addressing	27
6.3	System Addressing	29
7	SCPI Syntax Introduction	30
7.1	Common Commands.....	30
7.2	Device Dependent Commands.....	30
7.2.1	Header Construction.....	30
7.2.1.1	Indentations	31
7.2.1.2	Aliases.....	31
7.2.2	White Space.....	32
7.2.3	Long and Short Form, Upper and Lower Case	32
7.2.4	Optional Keywords.....	33
7.2.5	Parameters	33
7.2.5.1	Numeric Values <Nrf>.....	34
7.2.5.2	Units and Multipliers	34
7.2.5.3	Numerical Values and Extreme Values <num>	35
7.2.5.4	Boolean Parameter.....	36
7.2.5.5	Text.....	37
7.2.6	The Semicolon	37
7.2.7	Queries.....	39
7.3	SCPI Command Overview.....	41
7.3.1	Common Commands	41
7.3.2	Device Dependent Commands for ZSAC Series.....	42
7.4	SCPI Commands – Detailed Description.....	46

7.4.1	Common Commands	46
7.4.2	Device Dependent Commands	50
7.4.2.1	First Steps	50
7.4.2.2	Subsystem CHANnel INSTrument	52
7.4.2.3	Subsystem CURRent	54
7.4.2.4	Subsystem DATA TRACe	60
7.4.2.5	Subsystem DELay	63
7.4.2.6	Subsystem GTL	64
7.4.2.7	Subsystem INPut OUTPut	65
7.4.2.8	Subsystem LIST	66
7.4.2.9	Subsystem MEASure	75
7.4.2.10	Subsystem MODE FUNctIon	79
7.4.2.11	Subsystem PROGram	81
7.4.2.12	Subsystem RESistance	85
7.4.2.13	Subsystem SETup	91
7.4.2.14	Subsystem SFUNctIon	92
7.4.2.15	Subsystem STATus	102
7.4.2.16	Subsystem SYSTem	113
7.4.2.17	Subsystem TRIGger	125
7.5	User-Oriented Command Sequence	129
7.5.1	Resetting the Device	129
7.5.2	Selection of Function AC, DC, LINE	129
7.5.3	Selection of Mode Current, Resistance	129
7.5.4	Current Setting	129
7.5.5	Resistance Setting	130
7.5.6	Load Input On and Off	130
7.5.7	Measuring Current, Voltage, Resistance and Power Values	130
7.5.8	User-Programmable AC Waveform	131
7.5.9	Program Memory	131
7.5.10	Current Waveform With Phase Control	133
7.5.11	Current Waveform With Harmonics (Fourier Synthesis)	133
7.5.12	Current Waveform With Crest Factor	133
7.5.13	Phase Shift of Current Waveform	134
7.5.14	Storing a Waveform into Internal Memory	134
7.5.15	Programmable Waveform in DC or Envelope Curve in AC Mode	135
7.5.16	Data Acquisition	136
8	Manufacturer	137

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, während dieser Zeit sollten keine Befehle an das Gerät geschickt werden.

Warten Sie nach dem Ausschalten des Gerätes ca. 5s, 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, no commands should be sent to the device during this time.

After switching off the device, please wait 5s before switching it on again.

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

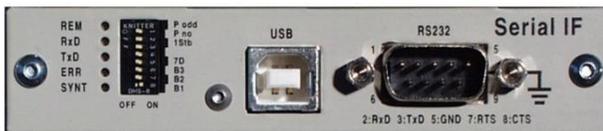
The optional serial interface allows the programming of the electronic load with standard programming language SCPI (Standard Commands for Programmable Instruments).

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

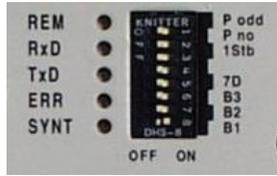
Switch Pno determines, whether a parity check occurs.

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:

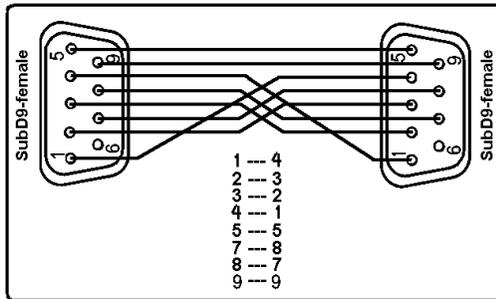


B1 = ON, alle anderen OFF, d. h.:
9600 Baud, 8 Datenbits, 2 Stopbits, Even Parity

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 zu verwenden (Buchse 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 (socket 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 die RTS- und CTS-Leitungen (Pin 7 und 8) 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) are wired cross-over. Pin 5 is straightly connected from the left to the right connector.

2.2 USB-Schnittstelle

Um eine ZSAC Last über die USB-Schnittstelle ansteuern zu können, muss ein USB-Treiber, der sich auf der mitgelieferten CD ROM befindet, 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 folgenden Pfad an:

```
<CD>:  
\USB\usb_driver_ZSAC_series_xx\win2k_xp
```

Das Installationsprogramm installiert einen "Virtuellen COM-Port" auf dem PC, dessen COM-Port-Nummer im Gerätemanager der Systemsteuerung ersichtlich ist. S. auch COMPORT.PDF und FTDI Driver Installation Ap Note for Win2000.pdf im o.g. Handbuch-CD-Pfad.

2.3 Datenformat bei RS232 und USB

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

Bei serieller Messdatenabfrage muss zwischen dem Absenden des Abfragebefehls und dem Auslesen der Daten eine Wartezeit von mindestens

100ms

eingehalten werden.

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

2.2 USB Interface

To control a ZSAC load via the USB interface a USB driver located on the shipped CD ROM 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. It is located in the following CD ROM directory:

```
<CD>:  
\USB\usb_driver_ZSAC_series_xx\win2k_xp
```

The installation program installs a "virtual COM port" on the PC which COM port number can be seen in the Windows Control Panel's Device Manager. See also COMPORT.PDF and FTDI Driver Installation Ap Note for Win2000.pdf in the above CD path.

2.3 Data Format of the RS232 Interface

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

When commands are sent to the device causing an answer (e.g. measurements) a delay of at least

100ms

must be kept between sending and receiving data.

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

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 IEEE 488 Interface

Zusätzlich zum RS232-Interface kann das IEEE 488-Interface in die Last eingesteckt werden, welche ebenfalls die Programmierung in SCPI erlaubt. Die IEEE 488-Schnittstelle beinhaltet lediglich die erforderliche GPIB Hardware, die Verarbeitung der SCPI-Befehle erfolgt im RS232-Interface, weshalb zur Verwendung der IEEE 488-Schnittstelle immer auch das RS232-Interface nötig ist.



Einstellen der Interfaceparameter

Die IEEE 488-Schnittstellenkarte kann über DIP Schalter auf der Geräte-rückseite eingestellt werden.

3.1 Einstellen der IEEE 488-Geräteadresse

Die Einstellung der IEEE 488 Geräteadresse erfolgt im Binärsystem.

3 IEEE 488 Interface

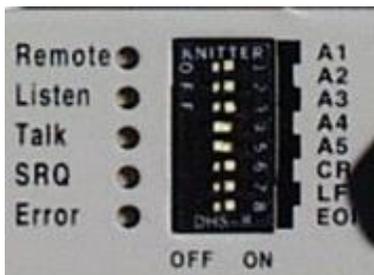
Additionally to the RS232 interface the IEEE 488 interface can be inserted into the load which also allows the programming in SCPI syntax. The IEEE 488 interface merely contains the required GPIB hardware. The procession of the SCPI commands happens in the RS232 interface. Therefore you always need also the RS232 interface with the IEEE 488 interface.

Setting the Interface Parameters

The setting of the IEEE488 interface functions is done by using the DIP switches at the back panel of the device.

3.1 Setting the IEEE488 Device Address

The setting of the IEEE 488 device address is provided in the binary system.



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

Um eine bestimmte Adresse einzustellen, muss anhand des Wertes die Schalter-stellung ermittelt werden.

Beispiel: Adresse 10
Schalter A4 und A2 auf ON

Bei der Auslieferung ist die IEEE 488 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 IEEE 488

Bei Betrieb über die IEEE 488-Schnittstelle können beliebige ASCII-Zeichen gesendet werden.

Beim Empfang von Daten erwartet das IEEE 488 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 IEEE 488 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 IEEE 488 Interface

When operating with the IEEE 488 interface any ASCII characters may be sent.

As end of string sign the IEEE 488 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).

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

Folgende Leuchtdioden signalisieren den Zustand der IEEE488-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

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!).

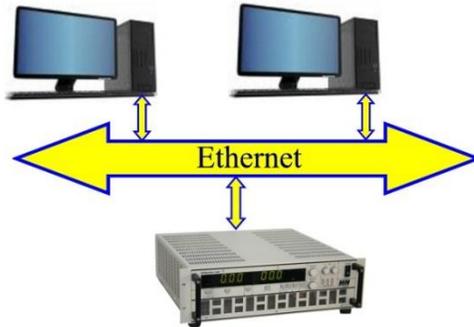
After adjusting one of the IEEE 488 DIP switches the device has to be switched off and on, so that the new setting will be accepted.

The following LEDs show the status of the IEEE 488 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 LAN Interface

4.1 Einsatzmöglichkeit



4.2 Anschluss

Verbinden Sie die Ethernet=>RS232 Karte mithilfe eines Standard-Netzwerkkabels mit einem Ethernet Switch (Patchkabel oder Crossover-Kabel) oder direkt mit der Netzwerkkarte eines PCs (Crossover-Kabel).

Verbinden Sie den RS232-Out-Stecker der LAN-Konverter-Karte über das mitgelieferte RS-232-Kabel mit dem RS232-Stecker der Serial IF Karte der Elektronischen Last.

Im Auslieferungszustand sind folgende Einstellungen vorgegeben:

IP Adresse: 192.168.111.80
Subnetzmaske: 255.255.255.0

4 LAN Interface

4.1 Possible Application

4.2 Connection

Connect the Ethernet=>RS232 card via a standard network cable with an Ethernet switch (patch cable or cross-over cable) or with the network card of a PC (cross-over cable).

Connect the RS232-Out plug of the LAN converter card with the RS232 plug of the Serial Interface card using the included RS232 cable.

The following settings are valid examples:

IP Address: 192.168.111.80
Subnet mask: 255.255.255.0

4.3 Konfiguration

Allgemein: Die LAN Schnittstellenkarte hat die Funktion eines Embedded Geräteservers für serielle Geräte. Verschiedene Betriebsarten stehen für die Verwendung zur Verfügung. H&H empfiehlt die Nutzung des „Real Com Modes“. In dieser Betriebsart ist die weitere Verwendung von bestehender Software, die für den Einsatz mit RS232 entwickelt wurde, möglich (z.B. H&H ZS Tools). Dabei wird die LAN Schnittstelle auf einen virtuellen COM Port abgebildet und es kann eine konventionelle serielle Verbindung aufgebaut werden.

- Installieren Sie das Tool „NPort Search Utility“. Dieses Tool finden Sie auf der mitgelieferten CD (MOXA Software CD) im Verzeichnis „D:\Software\Windows\NPort Search Utility“ oder unter www.hoecherl-hackl.de
- Öffnen Sie „NPort Search Utility“ und klicken Sie auf die Schaltfläche „Search“. Nach der erfolgreichen Suche finden Sie eine Auflistung aller sich im Netzwerk befindlichen LAN Module.



- Markieren Sie das zu konfigurierende LAN Modul und wählen Sie anschließend die Schaltfläche „Console“.

4.3 Configuration

General: The LAN interface converter acts as an Embedded Serial Device Server. The LAN interface converter offers different operation modes. H&H recommends the usage of the "Real Com Mode". This operation mode allows users to continue using software that was written for RS232 applications (e.g. H&H ZS Tools). For that purpose, the LAN interface converter is mapped to a Virtual COM Port to allow a conventional serial connection

- Install the „NPort Search Utility“ tool which can be found on the provided product CD (MOXA Software CD) „D:\Software\Windows\NPort Search Utility“ or at <http://www.hoecherl-hackl.com>.
- Open the „NPort Search Utility“ and click the button „Search“. After a short search you will get a list of all found NPort modules.

- Mark the desired module and click the button „Console“.

- Es öffnet sich ein Browserfenster mit der Startseite der Modulkonfiguration. Klicken Sie anschließend auf „Go to configuration“. Jetzt befinden Sie sich im Konfigurationsmenü.

- A browser window that shows the start page opens. Click on „Go to configuration“ to go to the configuration section.

Welcome to MiNePort web console

Model name	MiNePort E3
Serial No.	6517
Device name	MiNePort_E3_6517
Firmware version	1.0 Build 11071409
Ethernet IP address	192.168.127.254
Ethernet MAC address	00:90:E8:2C:6B:F7
Up time	0 days 00h:04m:33s
Serial communication status	Data Mode

- Navigieren Sie zu „Basic Settings“ -> „Network Settings“. Hier können Sie die Netzwerkeinstellungen der LAN Schnittstelle konfigurieren (Default IP address: 192.168.111.80). Kontaktieren Sie hierzu bitte Ihren Netzwerkadministrator, um Netzwerkkonflikte zu vermeiden. Um Änderungen dauerhaft zu speichern, quittieren Sie Ihre Eingaben mit der Schaltfläche „Submit“.

- Navigate to „Basic Settings“ -> „Network Settings“. Here you can configure the LAN settings of the LAN interface converter (Default IP address: 192.168.111.80). Please contact your administrator for a suitable network configuration to avoid network conflicts. To save the configuration click on „Submit“.

Network Settings

Device name:

IP configuration:

IP address:

Netmask:

Gateway:

DNS server 1:

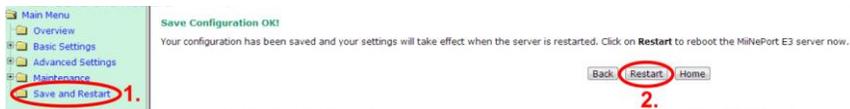
DNS server 2:

- Navigieren Sie nun zu „Basic Settings“-> „Operation Modes“. H&H empfiehlt die Verwendung des „Real COM“ Modus, da in dieser Betriebsart die vorhandenen ZS/NL Tools weiterverwendet werden können. Dieser Modus benötigt einen speziellen Treiber, der im Anschluss installiert wird. Natürlich kann auch ein anderer Funktionsmodus verwendet werden. Nähere Informationen hierzu finden sie auf der mitgelieferten CD im Verzeichnis „D:\Document\User's Manual“ oder unter www.hoecherl-hackl.de. Wählen Sie den Modus „Real COM“ aus der Drop-Down-Liste. Schreiben Sie in das Feld „Delimiter 1“ den Wert „0A“ (Linefeed LF) und setzen Sie den Haken bei „Enable“. Quittieren Sie die Eingaben mit „Submit“.
- Navigate to „Basic Settings“-> „Operation Modes“. H&H recommends the usage of the „Real COM Mode“. This operation mode allows users to continue using the ZS/NL tools. This operation mode needs a special driver, which will be installed at the step 11. It is also possible to choose a different operation mode. Detailed information can be found on the provided product CD in the folder „D:\Document\User's Manual“ or at www.hoecherl-hackl.com. Choose the „Real COM“ mode from the drop down list. Write „0A“ (Linefeed LF) into the „Delimiter 1“ field and set the „Enable“ checkbox. Quit the configuration by clicking on „Submit“.

The screenshot displays the 'Operation Modes' configuration window. On the left, a navigation tree lists various settings, with 'Operation Modes' circled in red. The main configuration area includes the following fields and options:

- Mode:** Real COM (selected in a dropdown menu)
- TCP alive check time:** 7 (range: 0 - 99 min)
- Max connection:** 1 (dropdown menu)
- Ignore jammed IP:** Disable (radio button selected)
- Allow driver control:** Enable (radio button selected)
- Data Packing:**
 - Packet length:** 0 (range: 0 - 1024)
 - Delimiter 1:** 0A (Hex) [checked] Enable
 - Delimiter 2:** 00 (Hex) [unchecked] Enable
 - Delimiter process:** Do nothing (dropdown menu)
 - Force transmit:** 0 (range: 0 - 65535 ms)
- Submit:** A button at the bottom of the configuration area.

- Klicken Sie nun auf „Save and Restart“ (1.) und anschließend auf „Restart“ (2.).
- Navigate to „Save and Restart“ (1.) and click on „Restart“ button (2.).



- Nun ist die LAN Schnittstellenkarte konfiguriert. Um Veränderungen der Konfiguration vorzunehmen, tippen Sie die IP Adresse der LAN Schnittstellenkarte in die Adresszeile Ihres Browsers oder verwenden sie das „NPort Search Utility“, wie bereits oben beschrieben.

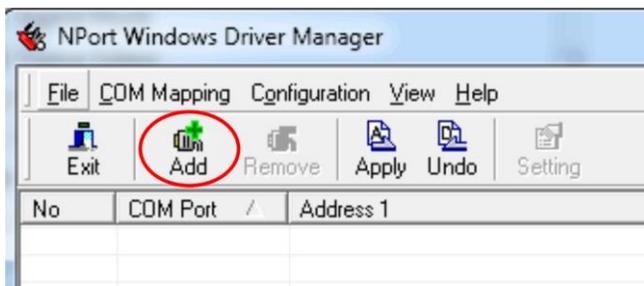
Installation des PC-Treibers für den „Real COM Mode“:

- Installieren Sie nun den „NPort Windows Driver Manager“. Dieses Tool finden Sie auf der mitgelieferten CD im Verzeichnis „D:\Software\Windows\NPort Windows Driver Manager“ oder unter www.hoecherl-hackl.de.
- Öffnen Sie den „NPort Windows Driver Manager“ und drücken sie die Schaltfläche „Add“.

- Now the LAN interface is configured. To change or review the settings insert the converters actual IP address into the address bar of your browser or use the “NPort Search Utility” as described above.

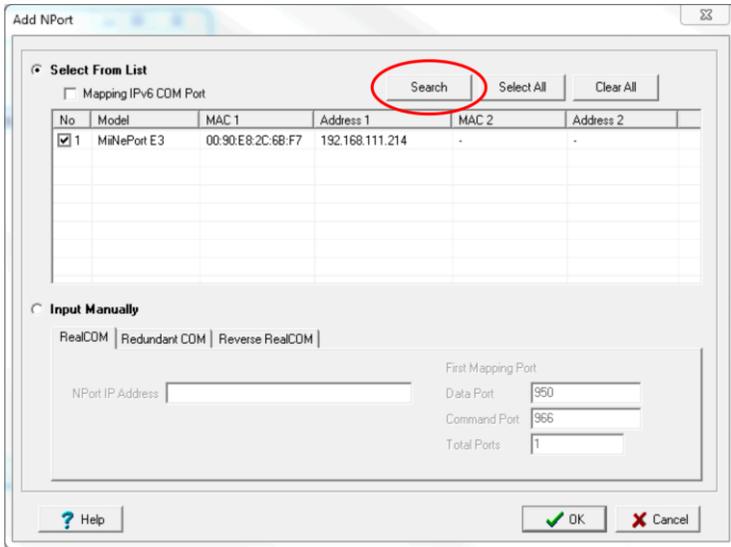
Installation of the PC driver for “Real COM Mode”

- Install the „NPort Windows Driver Manager“ from the delivered product CD „D:\Software\Windows\NPort Windows Driver Manager“ or from www.hoecherl-hackl.com.
- Open the „NPort Windows Driver Manager“ and click on the „Add“ button.

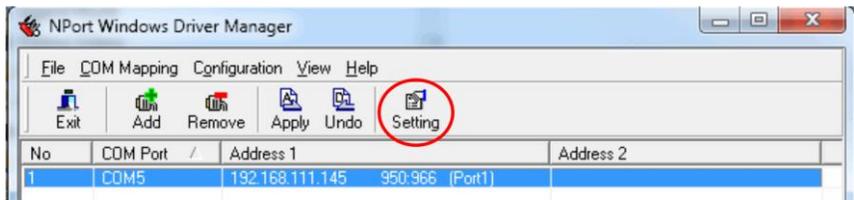


- Es öffnet sich das Fenster „Add NPort“. Klicken Sie auf die Schaltfläche „Search“, wählen Sie anschließend das gewünschte LAN Modul aus und verlassen Sie das Fenster mit der Schaltfläche „OK“.

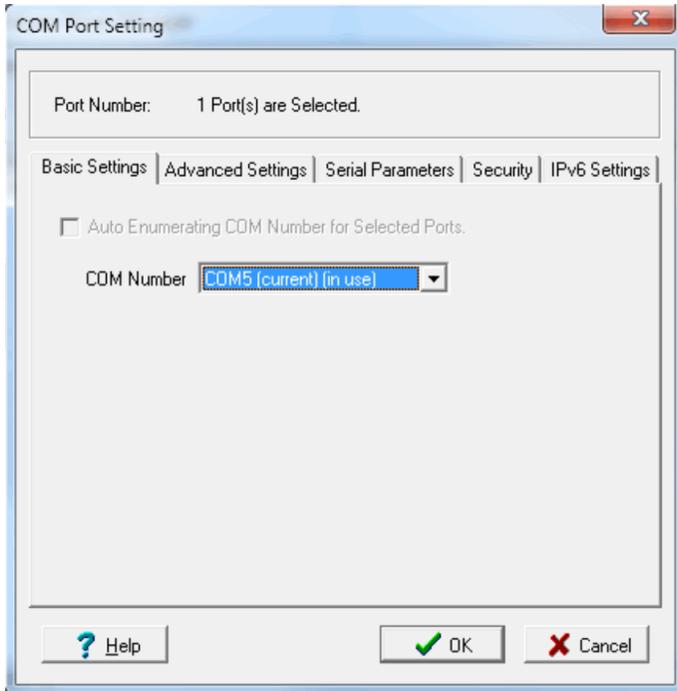
- The window „Add NPort“ opens. Click on the „Search“ button, choose the desired LAN converter module and click the „OK“ button.



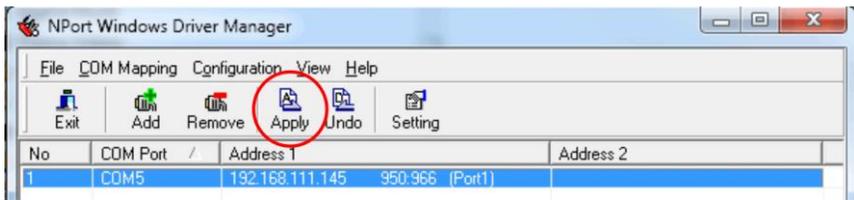
- Quittieren Sie das nachfolgende PopUp-Fenster mit der Frage „Do you want to activate the COM Port now?“ mit „No“.
- Markieren Sie anschließend den gewünschten NPort in der Liste und drücken Sie die Schaltfläche „Setting“.
- Confirm the following pop up window „Do you want to activate the COM Port now?“ with „No“.
- Mark the desired LAN converter module and click the “Setting” button.



- Anschließend öffnet sich das Fenster „COM Port Settings“. Wählen Sie hier die gewünschte COM Port Nummer aus.
- The window „COM Port Settings“ opens. Choose the desired COM port number.



- Verlassen Sie das Fenster „COM Port Setting“ mit der Schaltfläche „OK“ und wählen Sie die Schaltfläche „Apply“.
- Quit window „COM Port Setting“ with the „OK“ button and afterwards choose the „Apply“ button.



- Bestätigen Sie die folgenden Popup Fenster mit „Yes“ und „OK“.
- Schließen sie den „NPort Windows Driver Manager“.
- Nun ist die LAN Schnittstellenkarte betriebsbereit.
- Bei der Kommunikation mit einem Terminalprogramm oder Ihrer Applikation ist darauf zu achten, dass deren serielle Einstellungen denen der Elektronischen Last entsprechen.
- Confirm the following pop up windows with „Yes“ and „OK“.
- Exit „NPort Windows Driver Manager“ tool.
- Now the LAN converter module is ready to use.
- Please ensure that the serial settings of your application or terminal program match with the serial settings of the Electronic Load.

5 Systembus

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

Bei Verwendung eines Einzelgerätes in Verbindung mit einer RS232- oder GPIB-Schnittstelle ist dieser Punkt 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 IEEE488- oder RS232-Schnittstelle adressiert werden.

5 System Bus

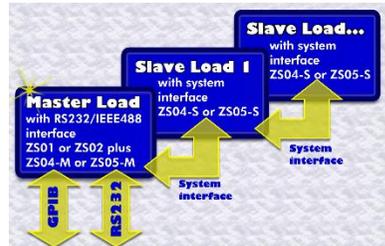
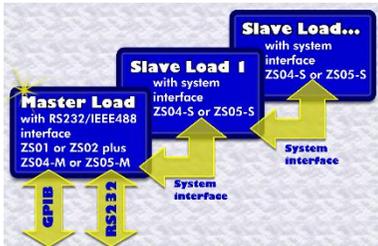
This section applies only for the control of several devices by the System Bus. For using a single device with RS232 or GPIB interface this section is not necessary for understanding the device functions. In this case continue the manual with section 7.

When operating with the System Bus Option ZS04 or ZS05 up to 192 devices can be addressed by one single IEEE488 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 IEEE488 or RS232 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

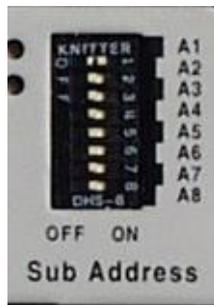
6.1 Einzeladressierung

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.

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.

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

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!

If a systembus interface is installed in the device the device must be addressed with the sub address set at the systembus interface, even when the systembus is not used at all.

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 sub 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

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

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.

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

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 Comm-ands for Programmable Instruments) beschreibt einen einheitlichen Befehls-satz zur Programmierung von Geräten, unabhängig von Gerätetyp und Hersteller. Es sollen damit die geräte-spezifischen Befehle vereinheitlicht werden.

7.1 Common Commands

Common Commands sind geräte-unabhängige Befehle, die in der Norm IEEE488.2 definiert werden. 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 IEEE488.2. 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?

: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 ONoder

OUTP 1

7.2.1.1 Indentions

The levels of the command hierarchy are identified by indentation 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?

: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 ONor

OUTP 1

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

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

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. Dies-es dient zur Erkennung des Stringendes.

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

7.2.2 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 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, ein-en getriggerten Strom von 5A zu programmieren:

CURRENT:TRIG 5

curr:triggered 5

Curr:TRig 5

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:
LIST: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 use certain keywords optionally 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:
LIST: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 multiplier 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 (!) ^{*)}
Leistung	W	Watt
	MW	Milliwatt
	KW	Kilowatt
Spannung	V	Volt
	MV	Millivolt
Zeit	S	Sekunde
	MS	Millisekunde

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

^{*)} Um zwischen den Multiplizierern "Milli" (10⁻³) und "Mega" (10⁶) 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!

^{*)} To distinguish between the multipliers "Milli"(10⁻³) and "Mega" (10⁶), 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!

Beispiel (Laststrom 520mA):

CURR 520MA
CURR 0.52
CURR 520E-3

Example (Load Current 520mA):

CURRENT 520MA
CURRENT:IMM 0.52

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>.

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 ZSAC2826
+2.000000E+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 ZSAC2826
+2.000000E+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:

LIST:MODE CONTInuous

Bei der Abfrage von Textparametern erhält man die Kurzform.

Beispiel:

LIST: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:

LIST:MODE CONTInuous

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

Example:

LIST: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.

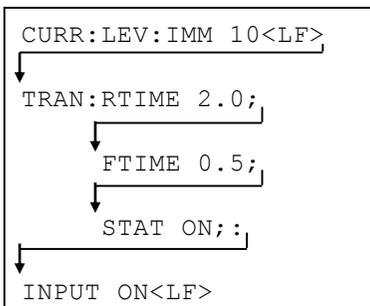
Das Stringende wird erkannt bei einem der folgenden Fälle:

IEEE488-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.

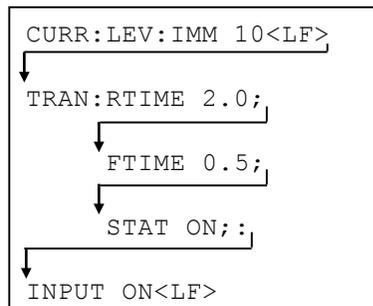
The string end is recognized in one of the following cases:

Operating Mode IEEE488 (see 1.2):

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

Operating Mode RS232:

- Character <LF> (10dec.)

Some Examples:

Abfragebefehle im RS232-Betrieb

Bei der Datenübertragung über die serielle Schnittstelle RS232 muss zwischen dem Absenden des Abfragebefehls und dem Auslesen der Daten eine Wartezeit von mindestens

100ms

erfolgen.

Query Commands for RS232 Operation

For the data transmission over the serial interface RS232 the wait time between the sending of the query command and the reading of the data has to amount at least to

100ms.

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 Std Event Status Enable Register	
*ESE?		Read Std Event Status Enable Register	
*ESR?		Read Std Event Status Register	
*IDN?		Identify Device. Response: "Manufacturer,Model,Serial No,Firmware Version"	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 ZSAC
7.3.2 Device Dependent Commands for ZSAC Series

Command	Parameter	Unit ¹⁾	Description	Duration ²⁾
CHANnel INSTrument	<NRf>[:<NRf>]		Select sub address or block	
CURRent [:LEVel] [:IMMediate] [:IMMediate]? :TRIGGered :TRIGGered? :MODE :MODE? :RANGe?	<num> [MIN MAX] <num> [MIN MAX] FIXed LIST [MIN MAX]	[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 Query current range	15ms 15ms 15ms 15ms 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 [:LEVel] :RTIME :DWELL :STRamp :STDWell :RESistance [:LEVel] :RTIME :DWELL :STRamp :STDWell :COUNT :STATE	<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> INfInity <Boolean>		Define current list Define current ramp list Define current dwell list Define current ramp measurement list Define current dwell measurement list Define resistance list Define resistance ramp list Define resistance dwell list Define resistance ramp measurement list Define resistance dwell measurement list Define number of loops Start/stop list execution	

:STATE?			Query list execution state	
MEASure :CURRent [:DC]? :EXternal [:DC]? :POWer [:DC]? :VOLTage [:DC]? :RESistance [:DC]?			Measure current Measure external voltage Measure power Measure voltage Measure resistance	20ms 20ms 20ms 20ms 20ms
MODE FUNction :CURRent [:DC] :RESistance [:DC] MODE FUNction?			Constant current mode Constant resistance mode Query mode	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 [:LEVel] [:IMMediate] [:IMMediate]? :TRIGgered :TRIGgered? :MODE :MODE? :RANGe?	<NRf> MAX [MIN] MAX <NRf> MAX [MIN] MAX FIXed LIST [MIN] MAX	[OHM] KOHM MOHM [OHM] KOHM MOHM	Set nominal resistance Query nominal resistance Set triggered resistance Query triggered resistance Start list or set triggered static resistance at trigger Query resistance mode Query resistance range	15ms 20ms 20ms 20ms 20ms
SETup?			Query device parameters	50ms
SFUNction :BATTery :ENABle :ENABle? :STATe?	<Boolean>		Enable/disable battery test function Query enable state Query active state	

:TEVoltage :TEVoltage? :AC :FSYNthesis :PHControl :PHShift :CRESt :WAVEform :WAVEform:RECall :WAVEform:SAVE	<Nrf> <Nrf1>,...,<Nrf8> <Nrf1>,<Nrf2> <Nrf> <Nrf> <Nrf1>,...,<Nrf33> <Nrf> <Nrf>		Define test-end voltage Query test-end voltage Fourier-Synthesis Phasecontrol Phaseshift Crest factor programming AC waveform programming Load AC waveform Save AC waveform	
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 :FUNction :LANGuage :LANGuage? :PARAmeter :PARAmeter <Nrf> ? :PROtection [:LEVel] [:LEVel]? :STATe :TRIPped? :STRing :VERsion?	IBFull RFR XON EXTernal INTernal AUTO FULL AC LINE DC SCPI <Nrf>,<Nrf> <Nrf> <Boolean> 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 Function AC,LINE oder DC 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 format string Query SCPI-Version	20ms 15ms 15ms

TRIGger [:SEQUence] :SOURce :SOURce? :TIMer :TIMer?	BUS EXTerنال IMMediate TIMer <NRf>		Set trigger source Query trigger source Set trigger timer value Query trigger timer value	
--	---	--	--	--

Bemerkung ⁽¹⁾:

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 ⁽²⁾:

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.

Comment ⁽¹⁾:

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 ⁽²⁾:

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.

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

Event Status Enable 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 ZSAC könnte beispielsweise lauten:

```
HOECHERL&HACKL,ZSAC2826,2525B-0804,ZSAC_AI_02.21-E-04A104-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 ZSAC load could be:

```
HOECHERL&HACKL,ZSAC2826,2525B-0804,ZSAC_AI_02.21-04A104-R00
```

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

ZS_AI_04.10 = Firmwareversion Analoginterface (04.10)

E = EEPROM Nichtflüchtiger Speicher (F wäre FRAM)

-01 = Firmwareversion Subcontroller (01)

-AI01 = Hardwareversion Analog-interface (01)

-R01 = Hardwareversion Regelungsplatine (01)

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 subdivided into the following information:

ZS_AI_04.10 = Firmware version ana-log interface (04.10)

E = EEPROM waveform memory (F means FRAM)

-01 = Firmware version subcontroller (01)

-AI01 = Hardware version analog interface (01)

-R00 = Hardware version controller board (01)

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 ZSAC sind dies:

***RST**

Reset resets the device to its standard settings.

For the ZSAC series electronic loads these are:

```

CURRent:RANGe:AUTO OFF
CURRent:RANGe MAX
CURRent 0
CURRent:TRIGgered 0
CURRent:MODE FIXed
MODE:CURRent
INPut OFF
LIST:CURRent 0,0,..
LIST:CURRent:RTIME 0,0,..
LIST:CURRent:DWELI 0,0,..
LIST:CURRent:STRamp 0,0,..
LIST:CURRent:STDWell 0,0,.. // equal for P/R/V lists
LIST:COUNt INF
LIST:STATe OFF
RESistance:RANGe:AUTO OFF
RESistance MAX
RESistance:TRIGgered MAX
RESistance:MODE FIXed
SFUNction:BATTery ENABle OFF
SFUNction:BATTery:TEVoltage 0
SYSTem:PROTection:STATe OFF
SYSTem:PROTection[:LEVel] 60s
TRIGger:SOURce IMM
TRIGger:TIMer 0.0002

```

***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 ZSAC Serie sind die Systeme

- CURRent
- INPut
- MEASure
- MODE
- RESistance

Die Defaulteinstellungen bei Reset des Gerätes sind `MODE:CURRent;;INPut OFF;;CURRent0;;RESistance 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. 1Ω) 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 ZSAC are

- CURRent
- INPut
- MEASure
- MODE
- RESistance

The default settings after a Reset of the device are `MODE:CURRent;;INPut OFF;;CURRent 0;; RESistance 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 1Ω), 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:CURRE?

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:CURRE?

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 IEEE488- 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 IEEE488- 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 IEEE488 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]	<num>	[A MA]	Laststrom einstellen
[:IMMediate]?	[MIN MAX]		Laststrom-Sollwert abfr.
:TRIGgered	<num>	[A MA]	getriggertes Laststrom
:TRIGgered?	[MIN MAX]		Laststrom-Triggerwert abfr.
:MODE	FIXed LIST		Bei Trigger Liste starten oder statischen Triggerstrom einst.
:MODE?			Current Mode abfragen
:RANGe	<num>		(nur für Kompatibilität)
:AUTO	<Boolean>		(nur für Kompatibilität)
:RANGe?	[MIN MAX]		Strombereich abfragen

7.4.2.3 Subsystem CURRent

Command	Parameter	Unit	Comment
CURRent			
[:LEVel]			
[:IMMediate]	<num>	[A MA]	Set load current
[:IMMediate]?	[MIN MAX]		Query nominal current
:TRIGgered	<num>	[A MA]	Set triggered current
:TRIGgered?	[MIN MAX]		Query triggered current
:MODE	FIXed LIST		Start list or set triggered static current at trigger
:MODE?			Query current mode
:RANGe	<num>		(only for compatibility)
:AUTO	<Boolean>		(only for compatibility)
:RANGe?	[MIN MAX]		Query current range

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

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

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 ZSAC2826: $+2.000000E+02$)

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 ZSAC2826: $+2.000000E+02$)

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.

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 wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Beispiele:

CURR:TRIG? (Antw. nach *RST:
+0.000000E+00)

CURR:LEVEL:TRIG? MAX

(Antw. v. ZSAC2606: +3.000000E+02)

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.

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.

Examples:

CURR:TRIG? (Resp. after *RST:
+0.000000E+00)

CURR:LEVEL:TRIG? MAX

(Resp. of ZSAC2606: +3.000000E+02)

CURRent:MODE FIXed|LIST

Bestimmt, ob bei einem Triggerereignis der statische Triggerstrom (CURRent:TRIGger) 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:RANGe <num>

Wählt einen festen Einstellbereich im Strombetrieb.

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

Hinweis:

Dieser Befehl ist ausschließlich aus Konformitätsgründen implementiert und ist für die Serie ZSAC nicht nötig, da diese nur über einen einzigen Strombereich verfügen.

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

Die speziellen Zahlenparameter MIN und MAX sind ebenfalls erlaubt.

CURRent:MODE FIXed|LIST

Determines if the static trigger current (CURRent:TRIGger) or a programmed current list (LIST:CURRent (waveform)) shall be set when a trigger event occurs.

After power-on CURRent:MODE FIXed is set.

See also Subsystems LIST and TRIGger.

Example:

CURR:MODE LIST

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: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:

This command is only implemented because of conformity reasons. It is not necessary for ZSAC series devices since these only have got one single current range.

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

The special numeric parameters MIN and MAX are allowed.

Beispiele:

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

CURRENT:RANGE MAX

CURRent:RANGe:AUTO ON|1|OFF|0

Schaltet den Autoranging-Modus im Strombetrieb ein oder aus.

Beispiel:

CURR:RANG:AUTO ON

Hinweis:

Dieser Befehl ist ausschließlich aus Konformitätsgründen implementiert und ist für die Serie ZSAC nicht nötig, da diese nur über einen einzigen Strombereich verfügen.

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 (bei ZSAC sind beide Werte gleich).

Beispiele:

CURR:RANG?

(Antw. v. ZSAC2826: +2.000000E+01)

CURR:RANGE? MAX

(Antw. v. ZSAC2826: +2.000000E+01)

Examples:

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

CURRENT:RANGE MAX

CURRent:RANGe:AUTO ON|1|OFF|0

Switches autoranging in constant current mode on or off.

Example:

CURR:RANG:AUTO ON

Note:

This command is only implemented because of conformity reasons. It is not necessary for ZSAC series devices since these only have got one single 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 (at ZSAC both values are identical).

Examples:

CURR:RANG?

(Resp. of ZSAC2826: +2.000000E+01)

CURR:RANGE? MAX

(Resp. of ZSAC2826: +2.000000E+01)

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*)

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>

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

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 auszulesenden 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 die Speicherung beendet und ein Bit im Questionable Status Register gesetzt.

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.

DATA|TRACe:REMOve? <NRF>

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

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.

The load sends the response in the following format:

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

The numbers are separated by commas without any space.



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 mehr Datensätze angefordert werden als verfügbar sind. Wir empfehlen daher, immer die verfügbare Anzahl der gespeicherten Datensätze mit *DATA:POINts?* abzufragen und diese dann mit *DATA:REMove? <Anzahl>* auszulesen.

Beispiel:

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



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. Therefore we recommend always to query the number of stored samples using the command *DATA:POINts?* 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]?

Fragt 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
: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
: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
:RESistance			
[:LEVel]	<NRf>{,<NRf>}		Define resistance list
:RTIME	<NRf>{,<NRf>}		Define resistance ramp list
:DWELL	<NRf>{,<NRf>}		Define resistance dwell 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/RESistance*), Anstiegs-/Abfallzeit (ramp time *RTIME*) und Verweildauer (dwell *DWELL*) jeweils in einer Liste an das Gerät übertragen. Dies ist in beiden Betriebsarten möglich.

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/RESistance*), rise/fall time (ramp time *RTIME*) and dwell (*DWELL*) are sent to the load as three lists. This is possible in both operating modes R and C.

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

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, RES)



Wenn FUNC:AC bzw. FUNC:LINE programmiert wurde, arbeitet die LIST-Funktion (siehe folgende Abb.) als Hüllkurven-Generator für die AC-Effektivwerte. Mit dem LIST-Subsystem wird die mit dem SFUN:AC-Subsystem programmierte AC-Kurvenform amplitudenmoduliert. Beide Kurven-Generatoren arbeiten voneinander unabhängig. Es ergibt sich beispielsweise folgender AC-Kurvenverlauf:

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

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, RES)

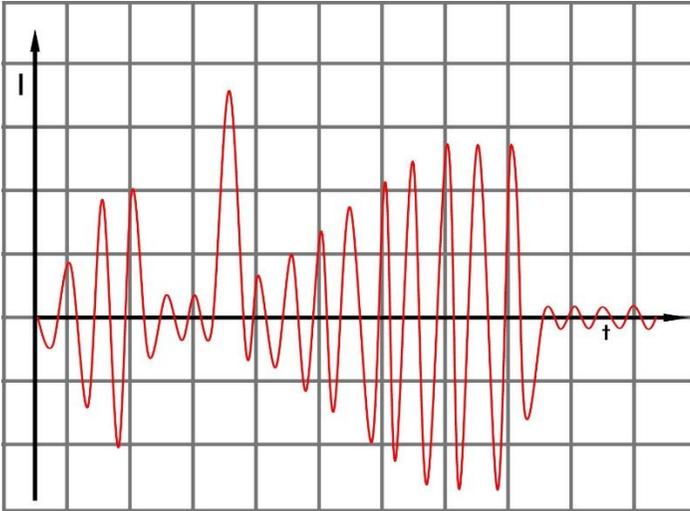


If FUNC:AC or FUNC:LINE was programmed, the LIST function works as an envelope generator for the AC RMS values (see following picture).

With the LIST subsystem the AC waveform programmed with the SFUN:AC subsystem is amplitude modulated.

Both curve generators work independently of each other.

For example, the following AC course is the result:

**LIST:CURRent[:LEVel] <NRf>{,<NRf>}**

Definiert eine Stromsollwert-Liste.

LIST:RESistance[:LEVel] <NRf>

{,<NRf>}

Definiert eine Widerstandssollwert-Liste.

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

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:RES:LEV 6E1,5E1, 4E1

LIST:CURRent[:LEVel] <NRf>{,<NRf>}

Defines a current list.

LIST:RESistance[:LEVel] <NRf>

{,<NRf>}

Defines a resistance list.

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

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:RES:LEV 6E1,5E1, 4E1

LIST:CURRENT:RTIME <NRf>{,<NRf>}

Definiert eine Liste für Stromrampen.

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

Definiert eine Liste für Widerstandsrampen.

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

Definiert eine Liste für die Verweilzeiten der Widerstandsliste.

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

Bei einer Verweilzeit=0 wird sofort beim Erreichen des entsprechenden Settings die nächste Rampe angefahren. Damit kann man z.B. dreieckförmige Ströme erreichen.

LIST:CURRENT:RTIME <NRf>{,<NRf>}

Defines a current ramp list.

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

Defines a resistance 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:RESistance:DWELL <NRf>{,<NRf>}

Defines a list for the dwells of the resistance list.

The dwell determines how long the instantaneous setting shall be kept until the change to the next one is started. At a dwell=0 the load immediately starts the next ramp when the corresponding setting is reached. This way you can make for example triangular currents.

LIST:COUNT <NRF>|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.

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.

LIST:COUNT <NRF>|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.

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.

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 laufender
Liste: 1)

Programmierbeispiele s. folgende Seiten.

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.

Wenn FUNC:AC bzw. FUNC:LINE programmiert wurde, arbeitet die LIST-Funktion ähnlich einem Hüllkurven-Generator, jedoch nicht für Spitzen-, sondern für Effektivwerte.

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.

If FUNC:AC or FUNC:LINE was programmed, the LIST function works similar to an envelope generator, not for peak values but for RMS values.



List Programming Example 1

Programmierung:

```

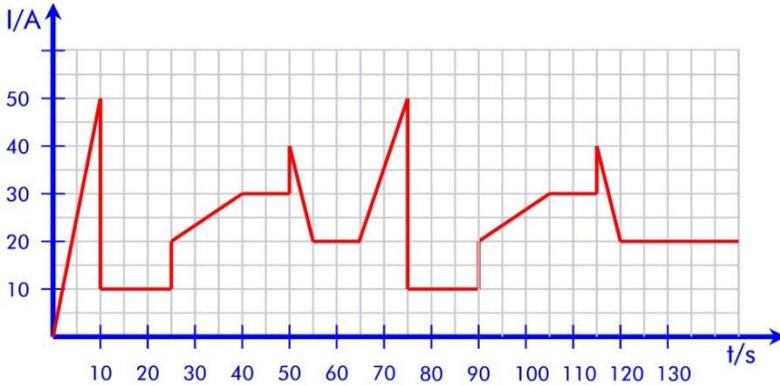
SYST:FUNC:DC
MODE:CURR
CURR 0
LIST:CURR 50,10,20,30,40,20
LIST:CURR:RTIM 1,0,0,15,0,5
LIST:CURR:DWEL 0,15,0,1,0,1
LIST:COUN 1
INP ON
LIST:STAT ON

```

Programming:

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:COUNT 2 :**

List Programming Example 2

Programmierung:**Programming:**

```

SYST:FUNC:DC
MODE:CURR
CURR:RANG 50::CURR 0
LIST:CURR 50,10,20,30,40,20
LIST:CURR:RTIM 1,0,0,15,0,5
LIST:CURR:DWEL 0,15,0,1,0,1
LIST:COUN 2
INP ON
LIST:STAT ON

```

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

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

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.

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: 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:

```

SYST:FUNC:DC
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 3: Externally triggered waveform

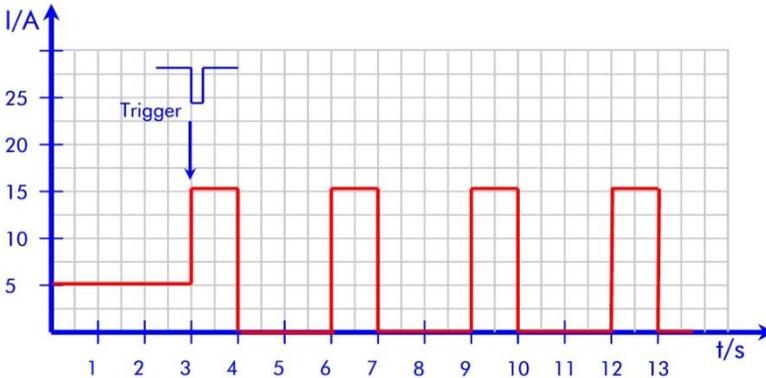
A predefined waveform (list) may also be started by an external trigger. For that, the trigger source must be defined and the mode of the concerning operating mode must be set to LIST.

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

```

SYST:FUNC:DC
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 :CURRent [:DC]? :EXTernal [:DC]? :POWer [:DC]? :RESistance [:DC]? :VOLTage [:DC]?			Strommesswert abfragen Ext. Messeingang abfragen Leistungsmesswert abfragen Widerstandsmesswert abfragen Spannungsmesswert abfragen

7.4.2.9 Subsystem MEASure

Command	Parameter	Unit	Comment
MEASure :CURRent [:DC]? :EXTernal [:DC]? :POWer [:DC]? :RESistance [:DC]? :VOLTage [:DC]?			Measure current Measure external voltage Measure power Measure resistance Measure voltage

Neben den Einstellmöglichkeiten für die verschiedenen Belastungsarten bietet das Gerät auch die Möglichkeit, Spannung, Strom, Widerstand und Leistung messen und die Messwerte an den Steuerrechner zu übergeben.

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.

Apart from the settings for the different load states the device offers the possibility to measure voltage, current and power, and to pass the measuring values to a controlling computer. 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.



Der zuletzt gespeicherte Wert wird bei der Messdatenabfrage übertragen und kann bis zu 300ms 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 300ms ein neuer Messwert angefordert werden.



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

Hinweis:

Die ZSAC 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. 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 kann das Gerät dazu veranlasst werden, einen Messwert zur Ausgabe bereitzuhalten.

Zur Datenaufbereitung benötigt das Gerät ca. 100 ms.

Bei der Kommunikation über die RS232-Schnittstelle kann der Messwert erst nach dieser Wartezeit gelesen werden.

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



The topical value is sent by the load when the MEAS command is received. This value may be up to 300ms old. That means, if you want to be sure to get a refreshed measurement value after a setting or a range change had taken place the measurement shall be requested 300ms after the new setting.



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

Note:

ZSAC 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. 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.

The device needs about 100ms to prepare the data.

Using the communication via RS232 interface the measuring value can be read after this delay.

A command string may only include one query command.

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

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

MEASure:CURRent[:DC]?

Aktuellen Laststrom messen.

Beispiel:

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

MEASure:POWer[:DC]?

Aktuelle Leistung messen.

Beispiel:

MEAS:POW? (Response e.g.
 $+1.79000+03$)



Der zurückgelesene Leistungswert entspricht dem Produkt aus Effektivstrom und Effektivspannung.

MEASure:VOLTage[:DC]?

Aktuelle Spannung messen.

Beispiel:

MEAS:VOLT? (Response e.g.
 $1.155000E+02$)

MEASure:RESistance[:DC]?

Aktuellen Widerstand messen.

Beispiel:

MEAS:RES? (Response e.g.
 $7.482500E+00$)

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

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

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

MEASure:CURRent[:DC]?

Measure actual load current.

Example:

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

MEASure:POWer[:DC]?

Measure actual power.

Example:

MEAS:POW? (Response e.g.
 $+1.79000+03$)



The read power value corresponds to the product of effective voltage and effective current.

MEASure:VOLTage[:DC]?

Measure actual input voltage.

Example:

MEAS:VOLT? (Response e.g.
 $1.155000E+02$)

MEASure:RESistance[:DC]?

Measure actual resistance.

Example:

MEAS:RES? (Response e.g.
 $7.482500E+00$)



Der zurückgelesene Widerstandswert entspricht dem Quotienten aus Effektivspannung durch Effektivstrom.



The read resistance value corresponds to the quotient of effective voltage and effective current.

7.4.2.10 Subsystem MODE|FUNCTION

Befehl	Parameter	Einheit	Bemerkung
MODE FUNCTION :CURRent [:DC] :RESistance [:DC] MODE FUNCTION?			Betriebsart Konstantstrom Betriebsart Konstantwiderstand aktuelle Betriebsart abfragen

7.4.2.10 Subsystem MODE|FUNCTION

Command	Parameter	Unit	Comment
MODE FUNCTION :CURRent [:DC] :RESistance [:DC] MODE FUNCTION?			Operating Mode Constant Current Operating Mode Constant Resistance Query actual operating mode

Die Geräte der Serie ZSAC können in den Betriebsarten Konstant-Strom (CURR) und Konstant-Widerstand (RES) arbeiten.

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.

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 eingestellt (s. *RST).

Die Default-Betriebsart nach dem Einschalten ist CURRent.

The electronic loads can be operated in the modes constant current (CURR) and constant resistance (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.

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 (see *RST) is set.

The default mode after power-on is CURRent.

MODE:CURRENT[:DC]

In Konstantstrombetrieb wechseln und zuletzt programmierten Wert einstellen.

Beispiel:

MODE:CURR

MODE:RESistance[:DC]

In Konstantwiderstandsbetrieb wechseln und zuletzt programmierten Wert einstellen.

Beispiel:

MODE:RES

MODE?

Momentane Betriebsart abfragen.

Als Rückgabewert wird die Kurzform der entsprechenden Betriebsart bereitgestellt (*CURR* oder *RES*).

Beispiel:

MODE?

(Antwort z.B.: RES)

MODE:CURRENT[:DC]

Change to constant current mode and set the last programmed value.

Example:

MODE:RES

MODE:RESistance[:DC]

Change to the operating mode constant resistance and set the last programmed value.

Example:

MODE:RES

MODE?

Queries the topical operating mode.

As response the short form of the respective SCPI mode keyword is returned (*CURR* or *RES*).

Example:

MODE? (Response for example: RES)

7.4.2.11 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

7.4.2.11 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-defined 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:

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:

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 *PROG:DELeTe* oder *PROG:DELeTe:SELeCted* wird der Programmspeicher gelöscht, der zuvor mit *PROG:NAME* selektiert worden ist. Mit *PROG: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)
```

PROG:SELeCted]:END

Beendet den Programmdefinitions-Modus.

Die folgenden Kommandos werden nicht mehr im Programmspeicher abgelegt, sondern direkt ausgeführt.

Beispiel:

```
PROG:END
```

PROG:SELeCted]:NAME {0..2}

Wählt den aktiven Programmplatz aus.

Es sind drei Programmspeicher verfügbar: Speicher 0, 1 und 2. verfügbar. Wenn ein Speicher mit *PROG:NAME <NRf>* selektiert worden ist, bleibt dieser solange selektiert, bis ein anderer Speicherplatz mit *PROG:NAME <NRf>* 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 *PROG:DELeTe* or *PROG:DELeTe:SELeCted* deletes the program which has been selected by *PROG:NAME*.

PROG: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)
```

PROG: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
```

PROG:SELeCted]:NAME {0..2}

Selects the active program storage.

There are three program storages: storage 0, 1, and 2.

If a storage has been selected with *PROG:NAME x* this storage stays selected until a different storage is selected with *PROG:NAME <NRf>*.

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

**PROG[[: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 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

**PROG[[: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:LEV 0

INP ON;DEL 0.2;CURR 20

PROG:END

The process may be started at any time:

PROG:NAME 1;STAT ON

7.4.2.12 Subsystem RESistance

Befehl	Parameter	Einheit	Bemerkung
RESistance [:LEVel] [:IMMediate]	<NRf> MAX	[OHM KOHM MOHM]	Konstantwiderstand einstellen
[:IMMediate] ? :TRIGgered	[MIN MAX] <NRf> MAX	[OHM KOHM MOHM]	Widerstands-Sollwert abr. getriggerten Widerstand einstellen
:TRIGgered? :MODE	[MIN MAX] FIXed LIST		Widerstand-Triggerwert abfragen Bei Trigger Liste starten oder statischen Triggersollwert einst.
:MODE?			Resistance Mode abfragen
:RANGe	<num>		Resistance Mode abfragen
:AUTO	<Boolean>		(nur für Kompatibilität)
:RANGe?	[MIN MAX]		(nur für Kompatibilität) Widerstandsbereich abfragen

7.4.2.12 Subsystem RESistance

Command	Parameter	Unit	Comment
RESistance [:LEVel] [:IMMediate]	<NRf> MAX	[OHM KOHM MOHM]	Set nominal resistance
[:IMMediate] ? :TRIGgered	[MIN MAX] <NRf> MAX	[OHM KOHM MOHM]	Query nominal resistance Set triggered resistance
:TRIGgered? :MODE	[MIN MAX] FIXed LIST		Query triggered resistance Start list or set triggered static resistance at trigger
:MODE?			Query resistance mode
:RANGe	<num>		(only for compatibility)
:AUTO	<Boolean>		(only for compatibility)
:RANGe?	[MIN MAX]		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]**<NRF>|MAX**

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.

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 Widerstandsbereiches erlaubt.

Außerdem ist der spezielle Zahlenparameter MAX erlaubt. Der Parameter MIN ist für Widerstandseinstellwerte nicht erlaubt.

Beispiele:*RES:LEV 15.23**RESistance:IMM 0**RES MAX*

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

RESistance[:LEVel][:IMMEDIATE]?

Abfrage des momentan eingestellten Sollwertes im Widerstandsbetrieb.

Als Rückgabewert wird eine Zahl im Exponentialformat bereitgestellt:

RESistance[:LEVel][:IMMEDIATE]**<NRF>|MAX**

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 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 parameter MAX is also allowed. The parameter MIN is not allowed for resistance settings.

Examples:*RES:LEV 15.23**RESistance:IMM 0**RES MAX*

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

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(Vorzeichen),
D: Digit,
E: Exponent

SD.DDDDDDESDD S: Sign,
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 mini-mal bzw. maximal möglichen Einstellwert.

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 is sent.

Beispiele:

RES? (Antwort z.B.: *+1.850000E+01*)

RES? MAX

(Antwort z.B.: *+1.780000E+01*)

Example:

RES? (Resp. e.g.: *+1.850000E+01*)

RES? MAX

(Resp. e.g.: *+1.780000E+01*)

RESistance[:LEVel]:TRIGgered <NRf>|MAX

Stellt einen neuen Wert für den getriggerten Widerstand ein.

S. auch SubSystem TRIGger.

RESistance[:LEVel]:TRIGgered <NRf>|MAX

Sets a new value for the triggered resistance.

See also 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.

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.

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

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).

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

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).

Als Parameter sind alle Zahlenwerte innerhalb des Widerstandsbereiches erlaubt.

All numeric values within the resistance range of the particular device type are allowed as parameters.

Außerdem ist der spezielle Zahlenparameter MAX erlaubt. Der Parameter MIN ist für Widerstandseinstellwerte nicht 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 wird angefordert, indem an das Fragezeichen ein Leerzeichen und der Parameter MIN bzw. MAX angehängt werden.

Beispiel:

RES:TRIG? MIN
(Antw.: +0.000000E+00)

RESistance:MODE FIXEd|LIST

Bestimmt, ob bei einem Triggerereignis der statische Triggerwiderstand (RESistance:TRIGger) 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

The special numeric parameter MAX is also allowed. The parameter MIN is not allowed for resistance settings.

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

Determines if the static trigger resistance (RESistance:TRIGger) or a programmed resistance list (LIST:RESistance (waveform)) shall be set when a trigger event occurs.

After power-on RESistance:MODE FIXEd is set.

See also Subsystems LIST and TRIGger.

Example:

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>

Wählt einen festen Einstellbereich im Widerstandsbetrieb.
Der numerische Parameter muss innerhalb des Widerstandsbereichs des betreffenden Gerätes sein (Technische Daten).

Hinweis:

Dieser Befehl ist ausschließlich aus Konformitätsgründen implementiert und ist für die Serie ZSAC nicht nötig, da diese nur über einen einzigen Widerstandsbereich verfügen.

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

Die speziellen Zahlenparameter MIN und MAX sind ebenfalls erlaubt.

Beispiele:

RES:RANG 10 (10Ω-Bereich oder
nächsthöheren
einstellen)

RESISTANCE:RANGE MAX

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 a fixed setting range for the operating mode resistance.

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

Note:

This command is only implemented because of conformity reasons. It is not necessary for ZSAC series devices since these only have got one single resistance range.

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

The special numeric parameters MIN and MAX are allowed.

Examples:

RES:RANG 10 (Set 10Ω range or
next higher range)

RESISTANCE:RANGE MAX

**RESistance:RANGe:AUTO
ON|1|OFF|0**

Schaltet den Autoranging-Modus im Widerstandsbetrieb ein oder aus.

Beispiel:

RES:RANG:AUTO ON

Hinweis:

Dieser Befehl ist ausschließlich aus Konformitätsgründen implementiert und ist für die Serie ZSAC nicht nötig, da diese nur über einen einzigen Widerstandsbereich verfügen.

**RESistance:RANGe:AUTO
ON|1|OFF|0**

Switches autoranging in resistance mode on or off.

Example:

RES:RANG:AUTO ON

Note:

This command is only implemented because of conformity reasons. It is not necessary for ZSAC series devices since these only have got one single resistance range.

7.4.2.13 Subsystem SETup

Befehl	Parameter	Einheit	Bemerkung
SETup?			Geräte-Einstellungen abfragen

7.4.2.13 Subsystem SETup

Command	Parameter	Unit	Comment
SETup?			Query device parameters

SETup?

Abfrage der Geräteeinstellungen.

Die Last antwortet mit einem String, der aus folgenden Informationen besteht: Unteradresse, Strombereiche, Spannungsbereich, Widerstandsbereiche, Leistungsbereich.

Antwortsyntax (Einbereichsgerät:
=A:a,C1:c1,V1:v1,R1:r1,P1:p1;

wobei:

- a* Unteradresse
- c1* Strombereich
- v1* Spannungsbereich
- r1* Widerstandsbereich
- v1* Spannungsbereich

Der Setstring ist mit einem Semikolon (;) abgeschlossen.

Beispiel:

SET? Antwort ZSAC2826:
=A:1,C1:20.0000,V1:260.000,
R1:600.000,P1:2800.0000;

SETup?

Queries the device settings.

The load responses with a string containing the following information:
Sub address, current ranges, voltage range, resistance ranges, power range.

Response syntax (single-range device):
=A:a,C1:c1,V1:v1,R1:r1,P1:p1;

where:

- a* Sub address
- c1* Current range
- v1* Voltage range
- r1* Resistance range
- p2* Power range

The set string is terminated by a semicolon (;).

Example:

SET? Response ZSAC2826:
=A:1,C1:20.0000,V1:260.000
,R600.000,P1:2800.000;

7.4.2.14 Subsystem SFUNction

Befehl	Parameter	Einheit	Bemerkung
SFUNction :BATTery :ENABLE :ENABLE? :STATe? :TEVoltage :TEVoltage?	<Boolean>		Batterieprüffunktion enabeln/disabeln Enablezustand abfragen Aktivzustand abfragen Testendspannung der Batterie def. Testendspannungs-Setting abfr.
:AC :PHControl :PHSHift :CRESt :FSYNthesis :WAVE :SAVe :RECall	<NRf1>, <NRf2> <NRf> <NRf> <NRf>, ..., <NRf8> <NRf>, ..., <NRf33> <NRf> <NRf>		AC-Funktionen Phasenanschnitt generieren Phasenverschiebung Crestfaktor einstellen Fourier-Synthese AC-Kurvenform programmieren AC-Kurvenform speichern AC-Kurvenform laden

7.4.2.14 Subsystem SFUNction

Command	Parameters	Unit	Comment
SFUNction :BATTery :ENABLE :ENABLE? :STATe? :TEVoltage :TEVoltage?	<Boolean>		Battery test functions Enable/disable battery test function Query enable state Query active state Define battery's test-end voltage Query TEV setting
:AC :PHControl :PHSHift :CRESt :FSYNthesis :WAVE :SAVe :RECall	<NRf1>, <NRf2> <NRf> <NRf> <NRf1>, ..., <NRf8> <NRf1>, ..., <NRf33> <NRf> <NRf>		AC functions Phase control Phase shifting Crest factor programming Fourier-Synthesis Program AC waveform Store AC waveform Load AC waveform

Das Subsystem SFUNction (Special Function) wurde geschaffen, um besondere Funktionen der Elektronischen Last, wie z.B. eine automatische Batterieprüffunktion bedienen und parametrisieren zu können.

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).

The SFUNction (Special Function) Subsystem was created to provide special functions of the electronic load, such as an automatic battery test function.

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

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

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: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?

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>**

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 1.202

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 10

SFUN:BATT:TEV 1.25;ENAB ON

INP ON

..

SFUN:BATT:STAT?

MEAS:CHAR?

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 1.202

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 10

SFUN:BATT:TEV 1.25;ENAB ON

INP ON

..

SFUN:BATT:STAT?

MEAS:CHAR?

SFUNction:AC:PHControl

<NRf1>,<NRf2>

Phasenanschnitt generieren:

<NRf1> entspricht dem Phasenwinkel der ersten Halbwelle in Grad; <NRf2> entspricht dem Phasenwinkel der zweiten Halbwelle in Grad, beide Phasenwinkel müssen im Bereich 0°...180° liegen.

Beispiel:

SFUN:AC:PHC 40,40

Wichtig:

Das Variieren der Phasenwinkel ist nur in Richtung größer werdender Winkel möglich, ein Phasenanschnitt kann nur durch Laden des Default-Sinus rückgängig gemacht werden.

SFUNction:AC:PHControl

<NRf1>,<NRf2>

Phase control:

<NRf1> corresponds to the phase angle of the first half-wave in degrees; <NRf2> corresponds to the second half-wave in degrees. Both phase angles must be in the range of 0°...180°.

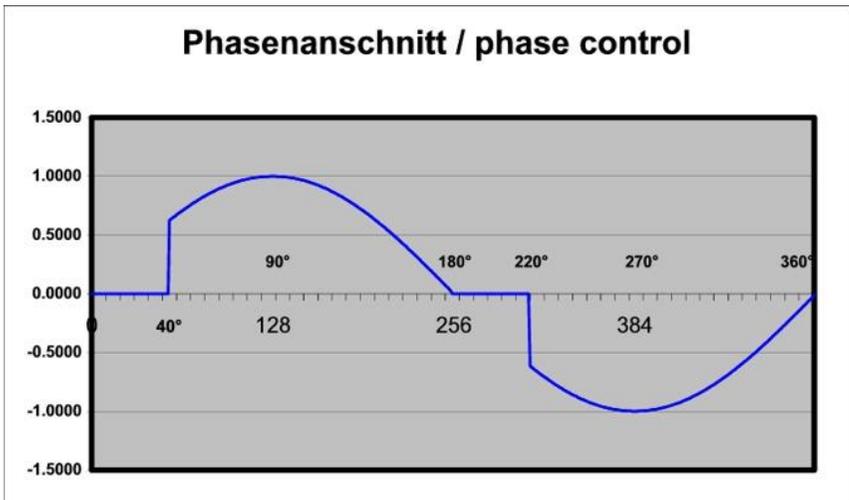
Example:

SFUN:AC:PHC 40,40

Important:

Its only possible to increase the phase angle, decrementing the phase angle does not change the wave.

A phase control can only be undone by recalling the default sine curve.



SFUNction:AC:CRESt <NRf>

Crestfaktor programmieren

<NRf> entspricht dem Crestfaktor. Der Crestfaktor muss im Bereich 0 ... 10.0 liegen.

Beispiel:

SFUN:AC:CRESt 2.8

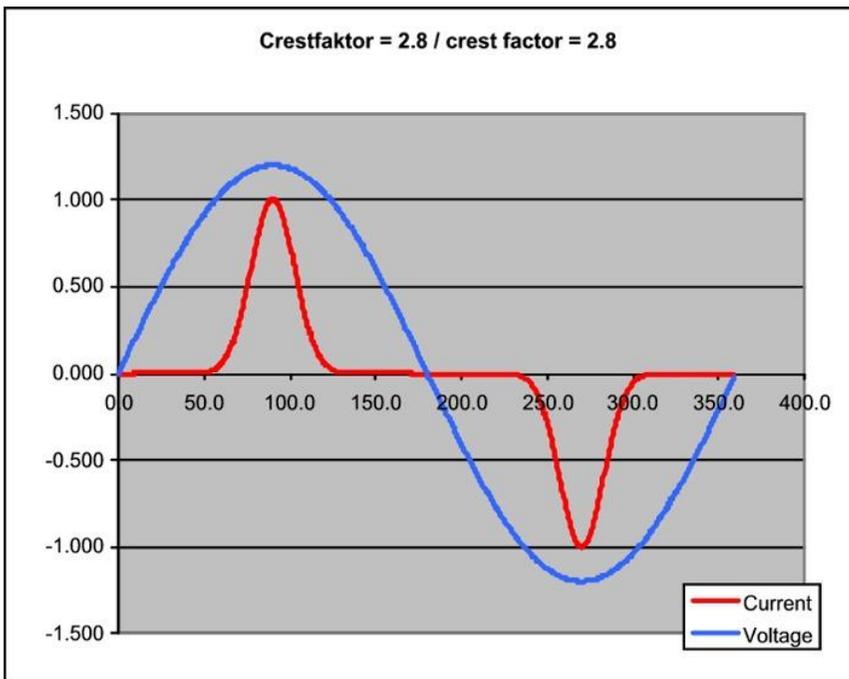
SFUNction:AC:CRESt <NRf>

Crestfactor programing

<NRf1> corresponds to the crest factor. The Crest factor must be in the range of 0.0 ... 10.0.

Beispiel:

SFUN:AC:CRESt 2.8



SFUNction:AC:PHShift <NRf>**Phasenverschiebung**

<NRf> entspricht dem Phasenwinkel, um den die Stromkurve in bezug auf die Spannung verschoben werden soll.

Der Phasenwinkel muss im Bereich -90° ... $+90^\circ$ liegen.

Beispiel:

SFUN:AC:PHS -45

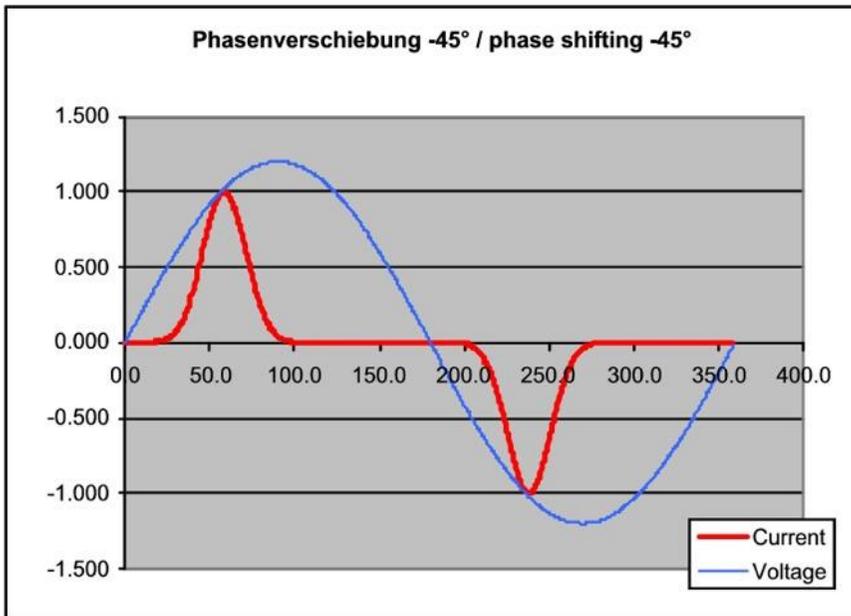
SFUNction:AC:PHShift <NRf>**Phase shifting**

<NRf> corresponds to the phase angle between input voltage and current.

Valid Values for phase angles are between -90° and $+90^\circ$.

Example:

SFUN:AC:PHS -45



SFUNction:AC:FSYNthesis

<NRf1>, ..., <NRf8>

Fouriersynthese

Die Fouriersynthese beschreibt die Erzeugung beliebiger Signale aus Sinusfunktionen.

<NRf1>...<NRf8> geben die Amplituden der ungeradzahigen Harmonischen im Bereich von 1...15 an. Es müssen immer alle 8 Amplitudenwerte programmiert werden. Die Kurvenform wird nach Berechnung der Fourier-Synthese auf den Wert 1.0 skaliert.

In diesem Beispiel ist die Fouriersynthese eines Rechtecksignals dargestellt:
SFUNction:AC:FSYNthesis 1.000,0.333, 0.200,0.143,0.111,0.091,0.077,0.067;

SFUNction:AC:FSYNthesis

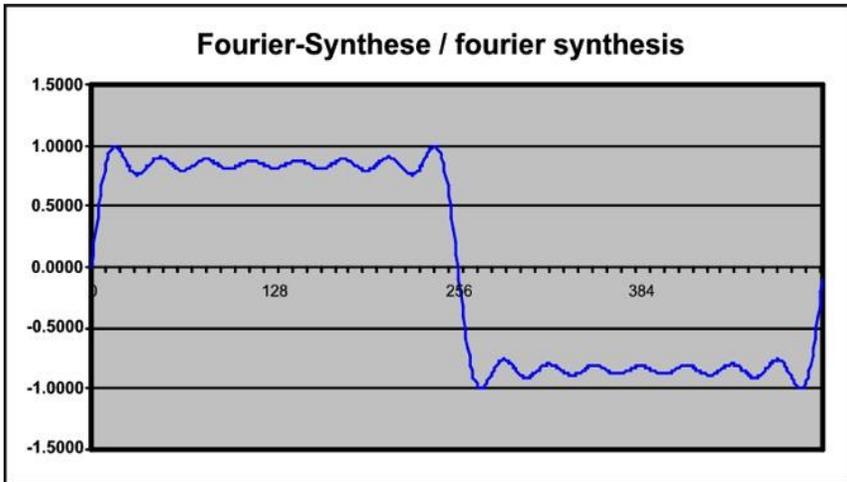
<NRf1>, ..., <NRf8>

Fourier Synthesis

The fourier synthesis describes the generation of arbitrary signals from sine functions.

<NRf1>...<NRf8> correspond to the amplitude values of the uneven harmonics in the range of 1...15. All 8 amplitude values must be programmed. The curve form is scaled to the value 1.0 after calculating the Fourier Synthese.

In this example the fouriersynthesis of a rectangle signal is represented:
SFUNction:AC:FSYNthesis 1.000,0.333, 0.200,0.143,0.111,0.091,0.077,0.067;



SFUNction:AC:WAVE**<NRf1>, ..., <NRf33>****AC-Kurvenform programmieren**

Eine AC-Kurvenform wird intern mit 512 einzelnen Punkten abgebildet. Um eigene Kurvenformen generieren zu können, kann jeder einzelne Punkt im Bereich von 0 bis 511 mit einem Wert zwischen 0.0 und 1.0 beschrieben werden.

Mit einem SCPI-Befehl können maximal 33 Parameter (Index + 32 Werte) gleichzeitig an die Last gesendet werden.

Beispiel:**Spannungseinbruch:**

SFUN:AC:WAVE:REC 1

SFUN:AC:WAVE 55,0.25,0.25,0.25,0.25;

SFUN:AC:WAVE 59,0.25,0.25,0.25,0.25;

SFUN:AC:WAVE 63,0.25,0.25,0.25,0.25;

SFUN:AC:WAVE 67,0.25;

SFUN:AC:WAVE 310,0.25,0.25,0.25,0.25;

SFUNction:AC:WAVE**<NRf1>, ..., <NRf33>****AC-wave programming**

An AC curve form is shown internally with 512 single points. To program your own waves, it is possible to set each single point (between 0 and 511) to a value between 0.0 and 1.0.

Within one SCPI command maximum 33 parameters can be sent to the load at the same time (index +32 values).

Example:**Glitch:**

SFUN:AC:WAVE:REC 1

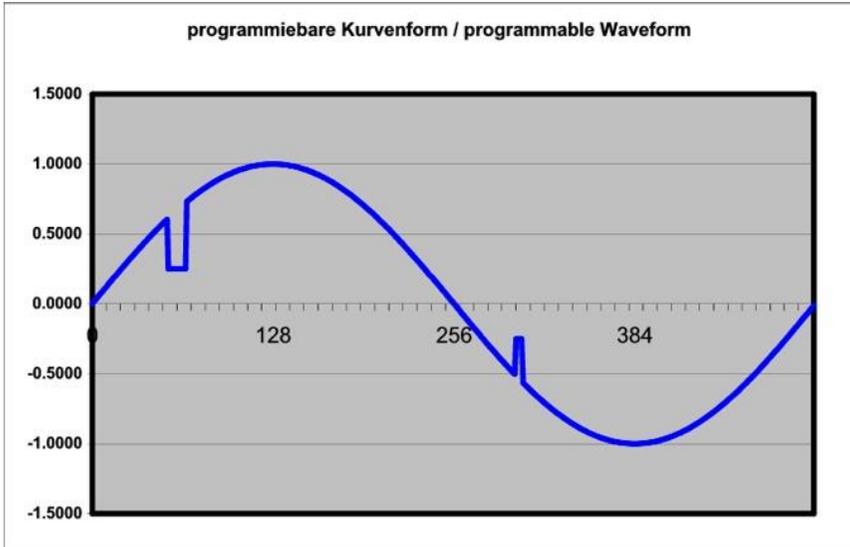
SFUN:AC:WAVE 55,0.25,0.25,0.25,0.25;

SFUN:AC:WAVE 59,0.25,0.25,0.25,0.25;

SFUN:AC:WAVE 63,0.25,0.25,0.25,0.25;

SFUN:AC:WAVE 67,0.25;

SFUN:AC:WAVE 310,0.25,0.25,0.25,0.25;

**SFUNction:AC:WAVE:SAVE <NRf>****Kurvenform speichern**

Speichert die aktuelle AC-Kurvenform unter der angegebenen Nummer <NRf> ins nicht-flüchtige EEPROM.

Gültige Werte für <NRf>: 2...6

Hinweis:

Unter Nummer 1 ist der Default-Sinus gespeichert. Dieser kann nicht überschrieben werden.

SFUNction:AC:WAVE:RECall <NRf>**Kurvenform laden**

Eine unter Nummer <NRf> gespeicherte Kurvenform in den AC-Kurvenform-Generator laden.

Hinweis:

Unter Nummer 1 ist der Default-Sinus gespeichert. Dieser kann nicht überschrieben werden.

SFUNction:AC:WAVE:SAVE**<NRf>****Save AC waveform**

Stores the actual AC waveform at memory <NRf> in non-volatile EEPROM. Valid values for <NRf>: 2...6

Note:

Memory 1 is reserved for the default sine-wave which is not writeable.

SFUNction:AC:WAVE:RECall**<NRf>****recall/load AC waveform**

Recalls/loads the waveform stored in memory <NRf> to the AC waveform generator.

Note:

The default sine-waveform is stored at memory 1 which is not writeable.

7.4.2.15 Subsystem STATus

Befehl	Parameter	Einheit	Bemerkung
STATus :OPERation [:EVENT]? :CONDition? :ENABLE :ENABLE? :QUESTionable [:EVENT]? :CONDition? :ENABLE :ENABLE? :PRESet	<NRf> <NRf>		Operation Event Reg. abfragen Op. Condition Reg. abfragen Operation Enable Bits setzen Op. Enable Register abfragen Ques. Event Reg. abfragen Ques. Condition Reg. abfragen Ques. Enable Bits setzen Ques. Enable Register abfragen Status rücksetzen

7.4.2.15 Subsystem STATus

Command	Parameter	Unit	Comment
STATus :OPERation [:EVENT]? :CONDition? :ENABLE :ENABLE? :QUESTionable [:EVENT]? :CONDition? :ENABLE :ENABLE? :PRESet	<NRf> <NRf>		Query Operation Event Reg. Query Op. Condition Reg. Set Operation Enable Bits Query Op. Enable Register Query Ques. Event Reg. Query Ques. Condition Reg. Set Ques. Enable Bits Query Ques. Enable Register Status Reset

Das Subsystem STATus dient zum einen zur Ermittlung von bestimmten Zuständen im Gerät und zum anderen zur Festlegung der relevanten Zustände für das Status Byte. Der Inhalt eines Statusregisters wird durch eine Dezimalzahl repräsentiert, die sich aus den Werten der gesetzten Bits zusammensetzt:

The subsystem STATus determines special states in the devices on one hand and sets the relevant values for the status byte on the other hand.

The contents of a status register is represented by a decimal number that is built of the values of the set bits:

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 Le-sen nicht verändert. Ein Zustand/Fehler ist aktiv, wenn das zugehörige Bit WAHR 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).

After the activation of the device all bits of all status registers are FALSE - except 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).

Ein Ereignis, d.h. ein Bit im Event Register, wird WAHR, wenn die zugehörige Bedingung einen Wechsel von FALSCH nach WAHR gemacht hat.

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.

An event, i.e. a bit in the Event Register, is set to TRUE, when the corresponding condition has changed from FALSE to TRUE.

The event is set until the corresponding Event Register has been read. After reading all bits in the Event Registers are reset.

• **Enable Register**

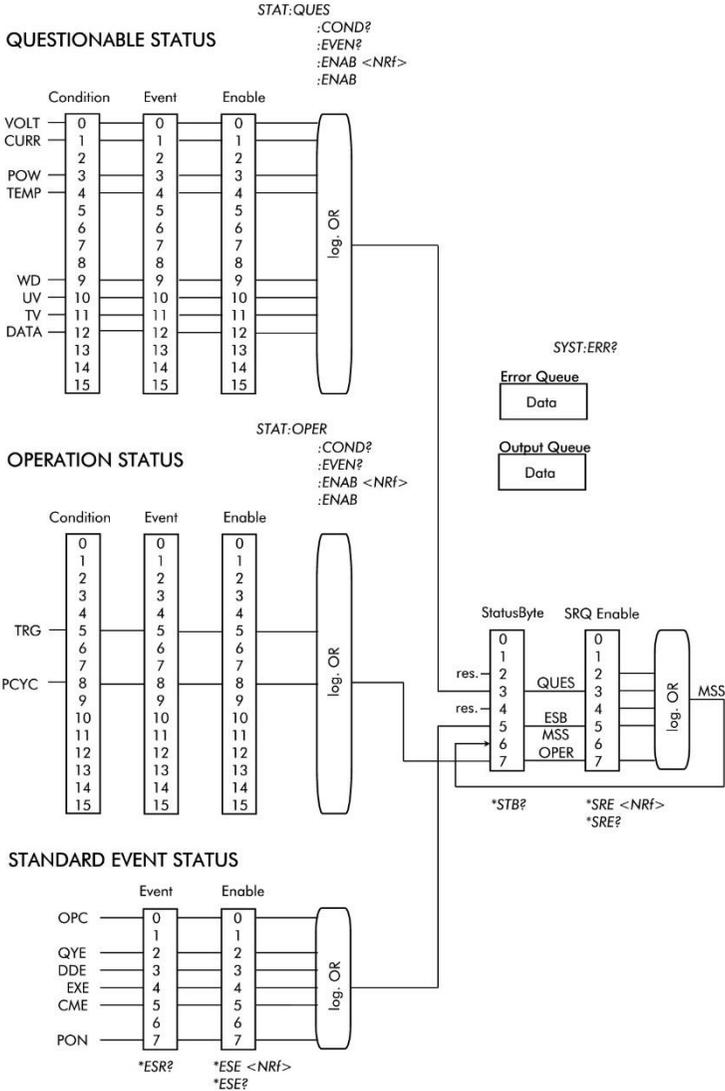
Determines which bits of the corresponding Event Registers are combined to a total bit using OR.

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.

Statusmodell ZSAC:



Questionable Status

Die Questionable Status Register informieren über bestimmte Fehler- bzw. Überlastzustände.

Bit 0, 1, 3 und 4 sind Überlastsignale der Leistungseinheit.

Beispiel: Überleistung und Übertemperatur: 8+16=24.

Questionable Status

The Questionable Status Register inform about particular error or overload states.

Bit 0, 1, 3 and 4 are overload singals from the power stage.

Example: Overpower and overtemperature: 8+16=24.

Bit	Wert Value	Bedeutung	Description
0 VOLT	1	Überspannungsfehler. LED OVP an der Frontplatte des Gerätes leuchtet.	Overvoltage error. LED OVP at front panel lights up.
1 CURR	2	Überstromfehler. LED OCP an der Frontplatte des Gerätes leuchtet.	Overcurrent error. LED OCP at front panel lights up.
3 POW	8	Überleistungsfehler. LED OPP an der Frontplatte des Gerätes leuchtet.	Overpower error. LED OPP at front panel lights up.
4 TEMP	16	Übertemperatur. LED OTP an der Frontplatte des Gerätes leuchtet.	Overtemperature 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.	Undervoltage. 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 (VOLTage:PROTection) liegt.	Trigger voltage. Is set when the input voltage is below the programmed trigger voltage (VOLTage:PROTection)
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.

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.

Beispiel:

STAT:QUES:COND?

Antwort: 24

STATus:QUESTionable[:EVENT]?

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: 24

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: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.

Example:

STAT:QUES:COND?

Resp.: 24

STATus:QUESTionable[:EVENT]?

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: 24

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?

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.
11 EXT	2048	Externe Ansteuerung ist aktiviert.	External control is activated (by command SYST:CONT EXT).

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[:EVENT]?

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[:EVENT]?

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 TRG setzen

STAT:QUES:ENAB 288

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: 288

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 TRG

STAT:QUES:ENAB 288

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: 288

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 ZSAC 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 ZSAC 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. Shows, that a read has taken place since the last change from OFF → ON.

Zum Lesen des Standard Event Status Registers wird das Common Command ***ESR?** verwendet.

Der Befehl ***ESE <0...255>** setzt die durch den dezimalen Parameter festgelegte Bitkombination im Standard Event Status Enable Register.

Damit wird festgelegt, welche Bits aus dem Standard Event Register in der Auswertung für das ESB Summenbit relevant sind.

For reading the Standard Event Status Register the common command ***ESR?** is used.

The command ***ESE <0...255>** sets the bit pattern in the Standard Event Status Enable Register that is determined by the decimal parameter.

Determines which bits from the Standard Event Register are relevant for the interpretation of the ESB sum bit.

Beispiel: Bit CME setzen

*ESE 32

Mit

***ESE?**

wird der Inhalt des Standard Event Status Enable Registers abgefragt.

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?**

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 enabeltes Questionable Event ist eingetreten.	Questionable. An enabled Questionable Event has happened.
ESB	32	Event Status Bit. Ein enabeltes 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 enabeltes 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.

7.4.2.16 Subsystem SYSTEM

Befehl	Parameter	Einheit	Bemerkung
SYSTem			
:COMMunication			
:SERial			
:CONTrol			
:RTS	IBFull RFR		RTS/CTS-Handshake ein
[:RECeive]			
:PACE	XON		XON/XOFF-Flusssteuerung ein
:CONTrol	EXTernal INTernal		Externe oder interne Ansteuerung setzen
:CONTrol?			Ansteuerquelle abfragen
:ERRor?			letzte Fehlermeldung lesen
:FAN	AUTO FULL		Lüftergeschw. auto oder voll
:FUNCTion	AC LINE DC		Function AC,LINE oder DC
:LANG	SCPI		Programmiersyntax
:LANG?			Programmiersyntax abfragen
:PARAMeter	<NRf> <NRf>		Systemparameter programmieren
:PARAMeter <NRf>?			Systemparameter lesen
:PROTection		[S MS]	
[:LEVe]	<NRf>		SW-Watchdog Zeit setzen
[:LEVe]?			SW-Watchdog Zeit abfragen
:STATe	<Boolean>		Software-Watchdog ein aus
:TRIPped?			Watchdog-Zustand abfragen
:STRing	<NRf>,"string"		Formatstring definieren
:VERSion?			SCPI-Version abfragen

7.4.2.16 Subsystem SYSTEM

Command	Parameter	Unit	Comment
SYSTem			
:COMMunication			
:SERial			
:CONTrol			
:RTS	IBFull RFR		Activate RTS/CTS handshake
[:RECeive]			
:PACE	XON		Activate XON/XOFF flow control
:ERRor?			Read last error message
:FAN	AUTO FULL		Fan speed auto or full
:FUNCTion	AC LINE DC		Function AC,LINE or DC
:LANG	SCPI		Programming language
:LANG?			Query programming language
:PARAmeter	<NRf> <NRf>		Set system parameter
:PARAmeter <NRf>?			Query system parameter
:PROTection			
[:LEVe]	<NRf>		Set SW-Watchdog time
[:LEVe]?			Query SW-Watchdog time
:STATe	<Boolean>	[S MS]	Software-Watchdog on off
:TRIPped?			Query Watchdog state
:STRing	<NRf>, "string"		Query regulation time constant
:VERSion?			Define format string
			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.

**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.

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).

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

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).

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: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 signaled by the error code -350 (Queue overflow).

Bei der Serie ZSAC 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 de 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 ZSAC 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 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:FUNCTION AC|LINE|DC

Das Gerät kann an verschiedenen Spannungen betrieben werden.

Mit dem Befehl SYSTEM:SFUNCTION kann die Funktion des Gerätes eingestellt werden.

Dabei ist:

AC: Wechselspannung
LINE: Wechselspannung mit
Netzfrequenz
DC: Gleichspannung

Siehe Kapitel *Bedienung/Einstellen der Gerätefunktion* im Hardwareteil der Bedienungsanleitung.

SYSTEM:LANGUAGE SCPI

Nur zur Konformität mit anderen H&H-Lasten. Die Programmiersyntax kann bei der Serie ZSAC 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: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.

SYSTEM:FUNCTION AC|LINE|DC

The load can work at different input voltages.

The function of the load can be selected with the command SYSTEM:SFUNCTION.

There is:

AC: AC voltage
LINE: voltage synchronized to the
mains supply
DC: DC voltage

See chapter Operating/Setting the device function in the hardware part of the operating manual.

SYSTEM:LANGUAGE SCPI

Just for compatibility with other H&H loads. The programming syntax cannot be changed at ZSAC devices but is always SCPI.

Example:

SYST:LANG SCPI

SYSTEM:LANGUAGE?

Queries the actual programming syntax. The device always returns SCPI.

Example:

SYST:LANG? (Response: SCPI)

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.

ZSAC System Parameter, die für den Anwender relevant sind:**ZSAC System Parameters relevant for the programmer:**

Parameter Number	Name	Explanation	Factory Setting
48	Pmax	Maximum power of UUT	max. Short-time Power of Load ¹⁾
49	VOC	Open-circuit voltage of UUT	max. Voltage of Load ¹⁾
50	ISC	Short-circuit current of UUT	max. Current of Load ¹⁾
51	VPM	Voltage at MPP	max. Voltage of Load ¹⁾
52	IPM	Current at MPP	max. Current of Load ¹⁾
53	Delta_P_min	Minimum Delta Power	$0.0002 * U_{\text{Range}} * I_{\text{Range}}$ ¹⁾
54	Delta_U_min	Min. Voltage level difference for readjusting MPP in C mode	$0.02 * U_{\text{previous}}$ ¹⁾
...			
81	Wave_min_steptime	Minimum waveform step width	$2.0E-4$ (200 μ s) ²⁾
82	Power_IO_card_out	8 bit relay port of Power I/O card	all open ²⁾
83	Power_IO_card_in	8 bit input port of Power I/O card	

¹⁾ non-volatile. Value is kept after power-off/power-on

²⁾ volatile. Value is set to Factory Setting after power-off/power-on

Waveform Parameter (Nr. 81)

Beeinflusst die Auflösung eines Teilstücks einer programmierbaren Kurvenform. Damit kann man schnellere Kurvenformen erreichen.

Wird dieser Parameter $< 200\mu$ s programmiert, ist jedoch keine synchrone Messung mehr möglich.

Der Wert wird in Sekunden programmiert und muss zwischen 100μ s und 200μ 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.

Waveform Parameter (No. 81)

Affects the resolution of a waveform's part line. Thereby faster waveforms can be achieved.

If this parameter is programmed $< 200\mu$ s no synchronized measurement is possible.

The value must be programmed in seconds and has to be within 100μ s and 200μ s.

Power I/O Card Parameters (Nr. 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.

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.

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

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.

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? Logikeingangs-Port lesen. Antwort z.B. +1.700000E+02 (170d=10101010b) Bits 2,4,6,8 high, Bits 1,3,5,7 low	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:PROTection[:LEVEL]**<0...4290000>**

Programmiert den geräteinternen Timer mit dem angegebenen Wert in Sekunden.

Die Geräte der Serie ZSAC 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 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[:LEVEL]**<0...4290000>**

Sets the device internal timer to the specified value, provided in seconds.

The devices of ZSAC 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 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**

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 Geräteingang in seinem bisherigen Zustand.

Beispiel:

Nach 15 Minuten ohne Dateneingang soll Eingang abgeschaltet werden:

SYST:PROT 900;PROT:STAT ON

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:PROTection:STATe
ON|1|OFF|0**

Activates/deactivates the software watchdog.

If the watchdog is activated (*SYST:PROT: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.

Example:

The input shall be deactivated, if there have been no data for 15 minutes:

SYST:PROT 900;PROT:STAT ON

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: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: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' must be used:

`%[flags][width][.precision][modifiers]type`

<i>type</i>	Ausgabe/Output	Beispiel/Example
e	Wissenschaftliche Notation (Mantisse/Exponent) mit e Scientific notation (mantise/exponent) using e character	3.9265e2
E	Wissenschaftliche Notation (Mantisse/Exponent) mit E Scientific notation (mantise/exponent) using E character	3.9265E2
f	Dezimale Gleitpunktzahl / Decimal floating point	392.65
g	Kürzestes aus %e oder %f / Use shorter %e or %f	392.65
G	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.
0number	Wie oben, jedoch mit 0 en anstelle von Leerzeichen aufgefüllt. Same as above but filled with 0 s instead of blanks.

<i>.precision</i>	Meaning
<i>.number</i>	Bei e , E , f Typen: Anzahl der Stellen, die nach dem Dezimalpunkt ausgegeben werden sollen. Bei g , G Typen: maximale Anzahl auszugebender signifikanter Stellen For e , E , f types: number of digits to be printed after the decimal point (if nothing specified default is 6). For g , G types : maximum number of significant digits to be printed.

<i>modifier</i>	Meaning
	<i>modifier</i> nicht benutzt. / <i>modifier</i> 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"

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"

Die Formatstringinformation ist flüchtig,
d.h. nach dem Aus- und Einschalten des
Gerätes ist wieder das Defaultformat
eingestellt.

The format string information is volatile,
that means after switching the device off
and on the default format is set.

SYSTEM:VERSion?

Abfrage der SCPI-Version, zu der das Gerät
kompatibel ist.

Beispiel:

SYST:VERS? Antwort: 1995.0

SYSTEM:VERSion?

Queries the SCPI version that the device
conforms.

Example:

SYST:VERS? Response: 1995.0

7.4.2.17 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

7.4.2.17 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>*, oder *RESistance[: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>* or *RESistance[: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 IEEE 488 und RS232 möglich)
- die IEEE 488-Mehrdrahtnachricht **GET**
(Group Execute Trigger, nur bei IEEE
488 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 eine abfallende
Flanke (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 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 IEEE 488 and RS232)
- the IEEE 488 multi channel message
GET (Group Execute Trigger, only for
IEEE 488 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 pin 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. 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]: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>

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 200µs ... 23.86h liegen und ist mit 50µs aufgelöst. Der Wert muss in Sekunden programmiert werden. Unter 300ms machen 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:

CURR 12.5;MODE:CURR;:INP ON

TRIG:TIM 5

TRIG:SOUR TIM

DATA:REM?

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 200µs ... 23.86h and is resolved with 50µs. The value must be in seconds.

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:

CURR 12.5;MODE:CURR;: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.

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.

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)

7.5 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.

7.5.1 Gerät in Grundzustand versetzen

***RST**

7.5.2 Auswahl der Funktionsart AC, DC, LINE

Es kann zwischen den Funktionsarten Wechselstrom (AC), Gleichstrom (DC) und Wechselstrom netzsynchronisiert (LINE) gewählt werden.

Auswahl AC:

SYST:FUNC AC

Auswahl DC:

SYST:FUNC DC

Auswahl LINE:

SYST:FUNC LINE

7.5.3 Auswahl der Betriebsart Strom, Widerstand

Es kann zwischen den beiden Betriebsarten Konstant-Strom oder Konstant-Widerstand ausgewählt werden.

Auswahl Konstant-Strom CURR

MODE:CURR

Auswahl Konstant-Widerstand RES

MODE:RES

7.5.4 Stromwert einstellen

Vergewissern Sie sich, dass Sie die richtige Funktion (AC, DC oder LINE) ausgewählt haben und sich in der Betriebsart Konstant-Strom CURR befinden.

Setzen eines Stromwertes:

CURR <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.

7.5.1 Resetting the Device

***RST**

7.5.2 Selection of Function AC, DC, LINE

One can choose between function alternating current (AC), direct current (DC) and alternating current line synchronized.

Selection AC:

SYST:FUNC AC

Selection DC:

SYST:FUNC DC

Selection LINE:

SYST:FUNC LINE

7.5.3 Selection of Mode Current, Resistance

One can choose between constant current or constant resistance mode.

Selecting constant current mode:

MODE:CURR

Selecting constant resistance mode:

MODE:RES

7.5.4 Current Setting

Make sure that the correct function (AC, DC, or LINE) and constant current mode is set.

Setting a current value:

CURR <Nrf>

<NRf> steht für einen Zahlenwert innerhalb des Gerätestrombereiches. Damit die Last den Strom einstellen kann, muss der Eingang noch zugeschaltet werden (s.u.).

7.5.5 Widerstandswert einstellen

Vergewissern Sie sich, dass Sie die Betriebsart Konstant-Widerstand ausgewählt haben.

Die Funktionsart (AC, DC, LINE) spielt für den Widerstandsbetrieb keine Rolle.

Einstellen eines Widerstandswertes:

RES <NRf>

<NRf> steht für einen Zahlenwert innerhalb des Gerätewiderstandsbereiches. Damit die Last den Strom einstellen kann, muss der Eingang noch zugeschaltet werden.

7.5.6 Geräteeingang zu- und abschalten

Eingang zuschalten:

INP ON

Eingang abschalten.

INP OFF

7.5.7 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 Effektivstrom- und -spannungswerten errechnet und sind nur richtig bei sinusförmiger Kurvenform.

Messen des Stromes

MEAS:CURR?

Messen der Spannung

MEAS:VOLT?

Messen des Widerstandes

MEAS:RES?

Messen der Leistung

MEAS:POW?

<NRf> is the numeric value within the device's current range.

The load input must be switched on to enable the device adjusting the current.

7.5.5 Resistance Setting

Make sure that constant resistance mode is set.

The function (AC, DC, LINE) does not have influence on the resistance mode.

Setting of a resistance value:

RES <NRf>

<NRf> is the numeric value within the device resistance range.

The load input must be switched on to enable the device adjusting the current.

7.5.6 Load Input On and Off

Switching input on:

INP ON

Switching input off:

INP OFF

7.5.7 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 effective current and voltage.

These are only correct at a sinusoidal curve.

Current measurement:

MEAS:CURR?

Voltage measurement:

MEAS:VOLT?

Resistance measurement:

MEAS:RES?

Power measurement:

MEAS:POW?

7.5.8 Freiprogrammierbare AC-Kurvenform

Es besteht die Möglichkeit, die Stromkurve frei zu programmieren.

Eine AC-Kurvenform wird intern mit 512 einzelnen Punkten abgebildet. Es kann jeder Punkt von 0...511 mit einem Faktor beschrieben werden.

Der Faktor 1 entspricht dem Spitzenwert der späteren Kurvenform.

Es kann ab jedem Punkt der Kurve mit dem Beschreiben begonnen werden. Der erste Parameter des Kommandos bestimmt den Index.

Es können maximal 32 Punkte in einem String beschrieben werden. Sind mehr Punkte zu beschreiben, müssen mehrere Strings generiert werden. Die erste Zahl <NRF1> im Kommando gibt an, ab welchem Punkt (0...511) die Kurve abgeändert wird, die zweite Zahl gibt den Faktor für den mit der ersten Zahl benannten Kurvenpunkt an, alle weiteren Faktorangaben beziehen sich dann auf die nachfolgenden Punkte.

SFUN:AC:WAVE <NRF1>, ..., <NRF33>

7.5.9 Programmspeicher

Das Gerät hat 3 interne Programmspeicher (NAME 0 ... 2). In jeden dieser Speicher können SCPI-Befehle abgelegt werden. Die Speicher können einzeln aufgerufen und gestartet werden, das eingespeicherte Programm wird dann selbständig abgearbeitet. Eine Besonderheit liegt darin, dass das Gerät nach dem Einschalten den Speicherplatz 0 automatisch abarbeitet. So können bestimmte Prüfanforderungen auch ohne angeschlossenen PC ausgeführt werden, wenn vorher eine entsprechende Befehlssequenz in der Last gespeichert worden ist.

7.5.8 User-Programmable AC Waveform

It is possible to program the current waveform arbitrarily.

An AC curve is internally represented with 512 single points. Every point from 0...511 can be written with a factor.

The factor 1 represents the peak of the resulting waveform.

One can start with writing at any point of the curve. The first parameter of the corresponding command gives the index.

At most 32 points can be described in one single string. If more points shall be written one must generate several command strings.

The first number <NRF1> in the command indicates the index (0...511) from where the curve is amended, the second number indicates the factor for the index point, all following factors obtain to following points.

SFUN:AC:WAVE <NRF1>, ..., <NRF33>

7.5.9 Program Memory

The device has 3 internal program memories (name 0 ... 2). SCPI commands can be stored in each of these memories. The memories can be called, then the stored program is worked-off in standalone. A special feature is that after switching on the device the storage 0 is executed automatically.

So certain tests can be done without a connected PC if the corresponding command sequence has been programmed into the program memory of the load.

Programm erstellen

Auswählen eines Speicherplatzes:

PROG:NAME <NRf>

(<NRf> ist Speicherplatz 0, 1 oder 2)

Speicherplatz in den Programmiermodus versetzen: **PROG:BEG**

Alle nachfolgenden Befehle werden jetzt in diesen Speicher geschrieben, ohne abgearbeitet zu werden.

Hinweis:

Während der Speicherbeschreibung erfolgt keine Syntaxprüfung durch die Last.

Programmiermodus abschließen

PROG:END

Starten eines gespeicherten Programms

Auswählen eines Speicherplatzes:

PROG:NAME <NRf>

(<NRf> ist Speicherplatz 0, 1 oder 2)

Programm starten:

PROG:STAT ON

Hinweis:

Die Abarbeitung der Befehle eines Programms kann nicht gestoppt werden.

Löschen von Programmspeichern

Auswählen eines zu löschenden Speicherplatzes

PROG:NAME <NRf>

(<NRf> ist Speicherplatz 0, 1 oder 2.)

PROG:DEL

Löschen aller Speicher

PROG:DEL:ALL

Generating a program

Selecting a storage:

PROG:NAME <NRf>

(<NRf> is storage 0, 1 or 2.)

Putting storage into programming mode:

PROG:BEG

All following orders are written into this memory without working off immediately.

Note:

The load performs no syntax check during memory writing.

Completing programming mode:

PROG:END

Starting a stored program

Selecting a storage:

PROG:NAME <NRf>

(<NRf> is storage 0, 1 or 2)

Starting program:

PROG:STAT ON

Note:

Running a program's commands cannot be stopped.

Deleting storage

Selecting a storage to be deleted

PROG:NAME <NRf>

(<NRf> is storage 0, 1 or 2)

PROG:DEL

Deleting all storages:

PROG:DEL:ALL

7.5.10 Stromkurve mit Phasenanschnitt

Mit der ZSAC Last können phasenangeschnittene Ströme erzeugt werden. Der Anschnittswinkel muss für die positive UND für die negative Halbwelle getrennt angegeben werden.

SFUN:AC:PHC <Nrf1>,<Nrf2>

<Nrf1> steht für den Winkel in der ersten Halbwelle, <Nrf2> für den in der zweiten. Ein Anschnittswinkel kann nur in größer werdende Richtung verändert werden. Soll ein bereits programmierter Phasenwinkel verkleinert werden, muss zunächst die reine Sinuskurve geladen und danach der Phasenanschnitt neu programmiert werden.

SFUN:AC:WAVE:REC 1

SFUN:AC:PHC <Nrf1>,<Nrf2>

7.5.11 Stromkurve mit Oberwellen (Fouriersynthese)

Die Stromkurve kann mit Oberwellen überlagert werden. Die folgenden ungeraden Harmonischen können mit einem Faktor, 0-1 der sich auf den Spitzenwert bezieht, angegeben werden: 1., 3., 5., 7., 9., 11., 13., 15. harmonische. Anschließend wird die Kurvenform auf 1 skaliert.

SFUN:AC:FSYN <Nrf1>,...<Nrf8>

Es müssen immer die Faktoren für alle 8 Harmonischen angegeben werden.

7.5.12 Stromkurve mit Crestfaktor

Mit der nachfolgend beschriebenen Funktion kann der Crestfaktor einer Sinuskurve verändert werden. Crestfaktoren von 0 bis 10 sind möglich, der Spitzenwert der Kurve bleibt dabei gleich.

SFUN:AC:CRES <Nrf>

<Nrf> steht für einen Crestfaktor von 0 bis 10.

7.5.10 Current Waveform With Phase Control

With the ZSAC load phase-controlled currents can be produced.

The angle for the phase control must be programmed for the positive AND negative half wave.

SFUN:AC:PHC <Nrf1>,<Nrf2>

<Nrf1> is the angle in the first half wave, <Nrf2> the angle in the second half wave.

A phase control angle can only be changed in increasing direction. If an already programmed angle shall be decreased one must previously reload the pure sinus curve and program the new angle afterwards.

SFUN:AC:WAVE:REC 1

SFUN:AC:PHC <Nrf1>,<Nrf2>

7.5.11 Current Waveform With Harmonics (Fourier Synthesis)

The current waveform can be superimposed with harmonic waves.

The following odd harmonics can be defined which factors 0-1 relate to the peak value: 1., 3., 5., 7., 9., 11., 13., 15. Then the waveform is scaled to 1.

SFUN:AC:FSYN <Nrf1>,...<Nrf8>

All 8 factors must be programmed.

7.5.12 Current Waveform With Crest Factor

With the function described below, the crest factor of a sine waveform can be changed.

Crest factors from 0 to 10 are possible, the peak value is kept.

SFUN:AC:CRES <Nrf>

<Nrf> is the crest factor from 0 to 10.

7.5.13 Phasenverschiebung der Stromkurve

Die Stromkurve kann gegen die Spannungskurve in nacheilende (positive Winkelangabe) oder in voreilende Richtung (negative Winkelangabe) verschoben werden. Da die Last im Nulldurchgang der Spannung weder Strom aufnehmen noch liefern kann, ist diese Funktion nur sinnvoll, wenn die Kurvenform so abgeändert wurde, dass zu diesem Zeitpunkt kein Stromfluss auftritt.

SFUN:AC:PHS <NRf>

(<NRf> steht für den Winkel der Phasenverschiebung von $-90^\circ \dots +90^\circ$)

7.5.14 Speichern einer Kurvenform in den Internspeicher

Eine erzeugte oder abgeänderte Kurvenform kann in einen der fünf (Wave 2...6) frei zur Verfügung stehenden Kurvenformspeichern abgelegt werden und bei Bedarf wieder aufgerufen werden. Im Handbetrieb ist die Auswahl einer abgelegten Kurve durch einen Taster möglich.

Der Kurvenformspeicher 1 kann nicht überschrieben werden, in diesem ist immer die reine Sinuskurve hinterlegt.

Abspeichern einer Kurve:

SFUN:AC:WAVE:SAVE <NRf>

(<NRf> steht für Speicherplatz 2...6)

Jede Kurve kann aus dem Kurvenformspeicher in den Arbeitsspeicher zurückgeholt werden.

Rückholen einer Kurvenform aus dem Internspeicher:

SFUN:AC:WAVE:REC <NRf>

(<NRf> steht für Speicherplatz 1...6)

Achtung: Wird eine Kurvenform bearbeitet oder verändert, muss vor dem Speichern in einen Kurvenform-

7.5.13 Phase Shift of Current Waveform

The current waveform can be moved against the input voltage curve in leading (positive angle) or trailing (negative angle) direction. Since in zero-crossing of the input voltage the load is not able to pass current to or take current from the DUT this function only makes sense if the curve waveform previously has been modified that no current appears at this time.

SFUN:AC:PHS <NRf>

(<NRf> is the phase shift in degrees $-90^\circ \dots +90^\circ$)

7.5.14 Storing a Waveform into Internal Memory

A generated or modified waveform can be stored into one of five waveform storages (2...6) and recalled again if required. In manual operation a stored waveform can be selected by a front panel key.

The waveform storage 1 cannot be overwritten, the pure sine waveform is saved in this storage.

Storing a curve:

SFUN:AC:WAVE:SAVE <NRf>

(<NRf> is storage 2 ... 6)

Any waveform can be recalled out of the waveform storage into the working memory.

Reloading a waveform from internal memory:

SFUN:AC:WAVE:REC <NRf>

(<NRf> is storage 1 ... 6)

Attention: Before storing a processed or changed waveform into a curve memory, 2000ms waiting time must be

speicher eine Wartezeit von 2000ms zwischen den Befehlen eingehalten werden.

7.5.15 Programmierbare Kurvenform im DC- oder Hüllkurve im AC-Betrieb

Mit dem Subsystem LIST können beliebige Lastkurvenformen für den DC-Betrieb in Form von aneinandergereihten ansteigenden oder abfallenden Geradenteilstücken definiert werden. Im AC- und Line-Betrieb dient diese als Hüllkurve für die gewählte AC-Kurve. 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

kept between the commands.

7.5.15 Programmable Waveform in DC or Envelope Curve in AC Mode

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

In AC and LINE mode this curve works as an envelope curve for AC amplitude modulation.

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>, ...

Sendig 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

7.5.16 Messdatenerfassung

Das Gerät verfügt über eine zeitgetriggerte, programmierbare Messwertaufnahme und einen internen Messdatenspeicher Datensätzen 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

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.)

7.5.16 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.

Start data logging:

TRIG:SOUR TIM

Stop data logging:

TRIG:SOUR IMM

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.)

8 Hersteller

8 Manufacturer



Your technical support and sales partner:
Schulz-Electronic GmbH supports you for all kind of
questions on these products.

The image contains the Schulz-Electronic logo, which consists of a blue triangle pointing up and an orange triangle pointing down, meeting at a point. To the right of the triangles, the text 'Schulz Electronic' is written in blue, with 'Professional Power Supplies' in a smaller font below it. Below the logo, the following contact information is provided:

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
www.schulz-electronic.de