



SCPI

STANDARD COMMANDS FOR PROGRAMMABLE INSTRUMENTS

Befehlsliste für Netzgeräte Command List for Power Supplies

Für den Gebrauch mit / For use with

IF-G1 / IF-E1B / IF-E2B / IF-E1*



* wird nicht mehr vertrieben / discontinued

	Seite
1. Abschlußzeichen bei GPIB.....	3
2. Syntaxformat	3
3. SCPI-Befehle.....	4
3.1 Allgemeine IEEE488.2 Befehle	4
3.2 Ereignissystem	4
3.3 Registerübersicht	5
3.4 Statusbefehle	6
3.5 Systembefehle	6
3.6 Befehle zur Steuerung des Ausgangs.....	9
3.7 Meßbefehle	9
3.8 Sollwertbefehle.....	9
3.9 Sonderbefehle.....	11
4. Fehlermeldungen	12
5. Anhang	13
5.1 SCPI-Befehlsübersicht.....	13

1. Abschlußzeichen bei GPIB

Hinweis

Betrifft nur GPIB (IF-G1)! Bei der Ethernetkarte ist kein Abschlußzeichen dieser Art erforderlich.

Die SCPI-Befehle werden als Klartext gesendet. Es ist ein Abschlußzeichen zu benutzen, das das Ende der Übertragung kennzeichnet:

• **LF (Line Feed, 0xA, ASCII 10)**

• **CR + LF**

• **CR**

Eine Übertragung erfordert zuerst eine Mitteilung vom Host (PC/SPS etc). Die IF-G1 antwortet, wenn der Host eine Antwort erwartet. Das ist immer dann der Fall, wenn am Ende des Befehls ein „?“ steht.

Befehle, die etwas stellen/setzen sollen, bestehen immer aus dem Befehl selbst und einem oder mehreren Werten. Der Befehl kommt zuerst, der oder die Werte durch Kommas getrennt danach:

<BEFEHL> **<Wert 1>**, **<Wert 2>**...

Generell können Befehle in ihrer Kurz- oder Langform gesendet werden. Nachfolgend wird die Kurzform der Befehle in **großgeschrieben** Buchstaben angegeben und ist stets ein Teil der Langform. Befehle können auch in Kleinbuchstaben sein.

2. Syntaxformat

Spezifikation nach „1999 SCPI Command reference“.

Folgende Syntaxformate können in Befehlen bzw. Antworten auftreten:

<Numerisch>

Der Zahlenwert entspricht dem Zahlenformat im Display des Gerätes und ist abhängig von den Nennwerten des Gerätes. Es gilt:

- er wird vom voranstehenden Befehl immer mit einem Leerzeichen getrennt eingegeben
- Anstatt eines Zahlenwertes können alternativ:
MIN (entspricht dem Minimalwert des Parameters)

Hinweis

Sollwerte, die höher sind als die Nennwerte des Gerätes, erzeugen einen Fehler!

<NR1> Zahlenformat ohne Dezimalpunkt

<NR2> Zahlenformat mit Dezimalpunkt

<NR3> Zahlenformat mit Dezimalpunkt und Größenordnung

<NRf> enthält **<NR1>** oder **<NR2>** oder **<NR3>**

Unit	V	Volt
	A	Ampere
	W	Watt
	OHM	Ohm
	s	Sekunden

<CHAR>	0..255: Dezimalzahl
<+INT>	0..32768: positive Integerzahl (Ausgabe)
<B0>	1 oder ON: Funktion ist/wird eingeschaltet 0 oder OFF: Funktion ist/wird ausgeschaltet
<B1>	NONE: lokaler Betrieb, eine Umschaltung auf Fernbedienung ist möglich LOCAL: nur lokaler Betrieb möglich, Auslesen von Daten ist zulässig REMOTE: Fernbedienung des Gerätes ist aktiviert
<B2>	ON oder 1: Automatische Messwerterfassung mit x Messpunkten ONCE oder 0: einmalige Messwerterfassung ausgelöst über *TRG mit x Messpunkten
<ERR>	Fehlernummer (-800 bis 399) und Beschreibung
<SRD>	String
<LF>	Endezeichen (line feed, 0x0 A)
<Time>	[[ddd], [hh], [mm], [s]s.s[s][s][s][s][s] Standardformat ist Sekunden (s.s)
;	Das Semikolon wird verwendet, um innerhalb einer Message mehrere Befehle zu senden.
:	Der Doppelpunkt trennt höherwertige Schlüsselwörter von niederwertigeren Schlüsselwörtern
[]	Kleinbuchstaben und der Inhalt in rechteckigen Klammern sind optional.
?	Das Fragezeichen kennzeichnet eine Abfrage. Die Abfrage kann gleichzeitig mit einer Datensendung verknüpft werden. Hierbei ist darauf zu achten daß, bevor eine neue Datensendung erfolgt, die Antwort des Systems abgewartet werden muss.
->	Antwort vom Gerät

3. SCPI-Befehle

3.1 Allgemeine IEEE488.2 Befehle

- *IDN?** Liest die Geräteidentifikation aus. Antwort: Benutzerdef. Text, Hersteller, Gerätetyp, Geräteserienummer, Gerätefirmwareversion und Firmwareversion der Schnittstellenkarte <LF>
- *RST** Gerät zurücksetzen durch folgende Prozedur:
- Umschaltung in Remote-Betrieb
 - den Ausgang/Eingang auf AUS setzen
 - alle Fehlermeldungen des Gerätes zurücksetzen
- *STB?** Liest das Status Byte Register, das nach dem Lesen gelöscht wird
- Folgende Befehle werden nur von der GPIB-Karte IF-G1 unterstützt:
- *TRG** Triggert einen Messzyklus
- *CLS** Löscht alle Event- und Statusregister des GPIB Controllers
- *ESE <CHAR>** Setzt das Event Status Enable Register
- *ESE?** Liest das Event Status Enable Register
- *ESR?** Liest das Event Status Register, das nach dem Lesen gelöscht wird
- *SRE <CHAR>** Setzt das Service Request Enable Register
- *SRE?** Liest das Service Request Enable Register

3.2 Ereignissystem

Der Signallauf für die verschiedenen Ereignisse wird im Diagramm auf der nächsten Seite verdeutlicht.

Ereignisse können durch Abfrage des Statusregisters STB (GPIB, Ethernet) ausgelesen werden.

Die Bits des Statusregisters STB im Einzelnen:

- Bit 0: nicht verwendet
- Bit 1: nicht verwendet
- Bit 2: **err**, Error Queue (Fehlerliste) ist gefüllt; durch Auslesen der Fehlerliste wird diese gelöscht und das Bit zurückgesetzt. Die Liste kann bis zu 4 Fehler speichern
- Bit 3: **ques**, Questionable Status Register ist aktiv (ein oder mehrere Ereignisse stehen an)
- Bit 4: nicht verwendet
- Bit 5: **esr**, das Standard Event Status Register (ESR), maskiert mit dem Event Status Enable Register (ESE), meldet, daß ein oder mehrere Ereignisse anstehen
- Bit 6: **rsv**, immer aktiv
- Bit 7: **oper**, meldet, daß im Operation Status Register ein oder mehrere Ereignisse anstehen

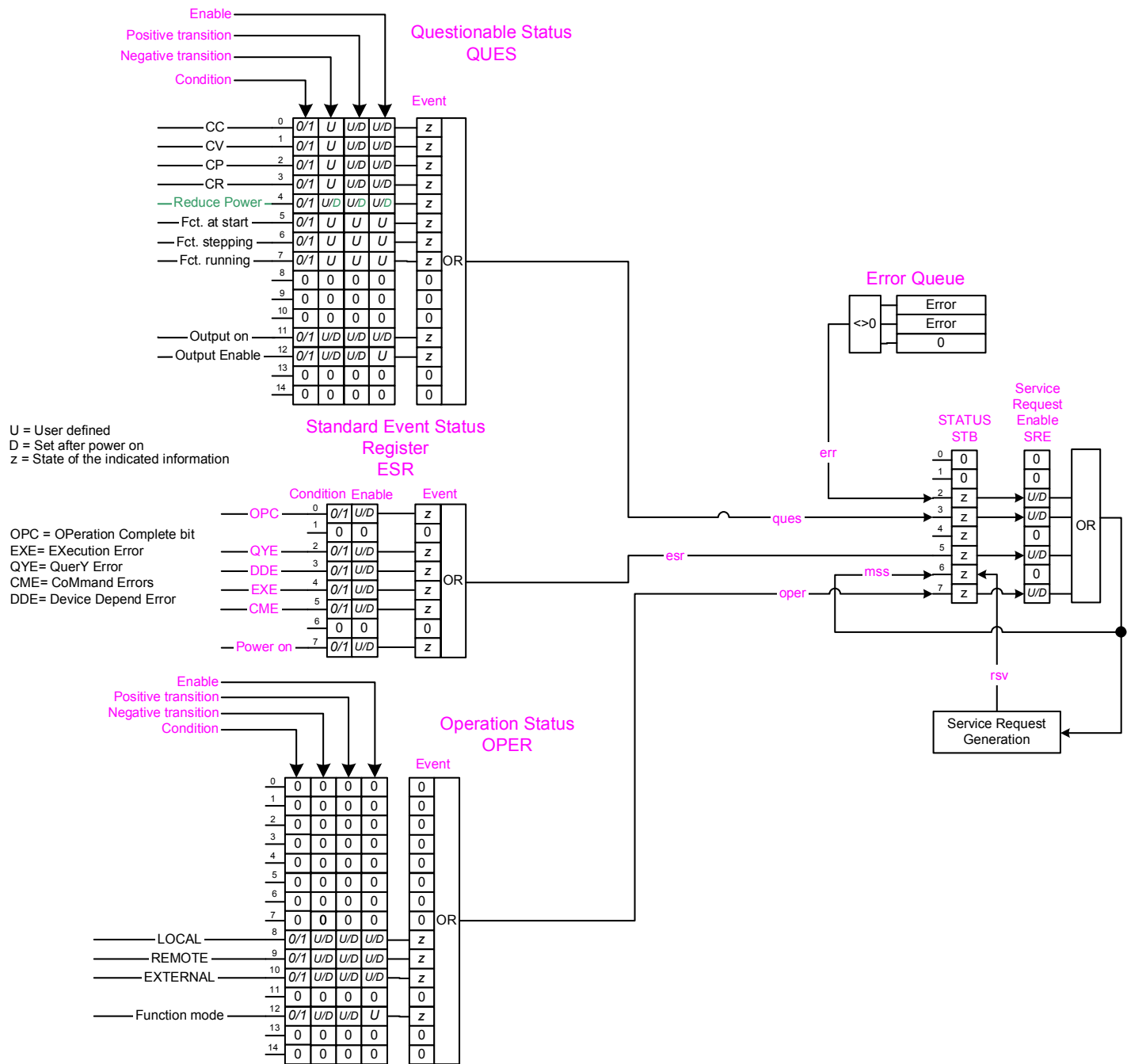
Die Ereignisbits der verschiedenen Register werden zum STB gemeldet, wenn Ereignisse aufgetreten sind, die durch die zugehörigen Bits in den Freigabe-Registern (***ESE**, ***SRE** bzw. **STAT:QUES:ENAB**, **STAT:OPER:ENAB**) zugelassen wurden.

Die Bits des ESR sind im Einzelnen:

- Bit 0: Ausführung beendet
- Bit 1: nicht verwendet
- Bit 2: Anfragefehler (Query error)
- Bit 3: Device Dependent Error (Hardware defekt etc.); Fehler von -399 bis -300 bzw. 100...399
- Bit 4: Execution Error (Strombegrenzung, Grenzwerte überschritten); Fehler von -299 bis -200
- Bit 5: Command Error (falscher Befehl); Fehler von -199 bis -100
- Bit 6: nicht verwendet
- Bit 7: Power On (Gerät wurde eingeschaltet)

Ereignis- und Statusregister können mit dem Befehl ***CLS** gelöscht werden.

3.3 Registerübersicht



Legende:

CC/CV/CP/CR = aktuelle Regelungsart

Reduce Power = Leistungsbegrenzung (nur PSI 9000 Serie)

Fct. at start/running/stepping = Funktionsmanagerstatus

Output on = Eingang bzw. Ausgang des Gerätes ist eingeschaltet

Output enable = Einschaltbereitschaft des Ausganges ist aktiviert

LOCAL = Gerät im Lokalbetrieb, Fernsteuerung ist gesperrt

REMOTE = Gerät wird durch digitale Schnittstellenkarte gesteuert

EXTERNAL = Gerät wird durch analoge Schnittstellenkarte bzw. Analogschnittstelle am Gerät gesteuert

Function mode = Funktionsmanager aktiv

3.4 Statusbefehle

(Spezifikation nach „1999 SCPI Command reference“: 20 System Subsystem)

Das Register Operation Status (OPER) (siehe Diagramm auf der vorherigen Seite) speichert das Auftreten von Zuständen (remote, local usw.) im Unterregister *Condition* zwischen und gibt diese Zustände weiter an das Unterregister *Event*, sofern dieses durch *Enable* freigegeben ist. Die Masken *Positive transition* und *Negative transition* bestimmen, ob die Ereignisse bei einer Low-High-Flanke oder einer High-Low-Flanke ausgegeben werden. Somit kann zum Einen das Auftreten und zum Anderen das Verschwinden eines Zustandes bemerkt werden.

Das gleiche Prinzip gilt für das Questionable Status Register (QUES). In der im Bild gezeigten Konfiguration für das OPER würde das Signal „local“ nur bei einer pos. Flanke (Low->High) ein Ereignis ausgeben, das Signal „Function mode“ dagegen auch bei einer neg. Flanke.

STATus

:OPERation		Betriebsabhängige Meldungen
[:EVENT]?	-><NR1>	Abfrage der Ereignisse im Status Operation Register
:CONDition?	-><NR1>	Zustand der betriebsabhängigen Funktionen abfragen
:ENABle	<NR1>	Freigabe des Ereignisses(Event)
:ENABle?	-><NR1>	Abfrage
:PTRransition	<NR1>	Event wird nur bei Übergang von 0 auf 1 weitergeleitet
:PTRransition?	-><NR1>	Abfrage
:NTRransition	<NR1>	Event wird nur bei Übergang von 1 auf 0 weitergeleitet
:NTRransition?	-><NR1>	Abfrage
:QUESTionable		Geräte und funktionsspezifische Ereignisse
[:EVENT]?	-><NR1>	Abfrage der Ereignisse im Questionable Status Register
:CONDition?	-><NR1>	Zustand der gerätespezifischen Funktionen abfragen
:ENABle	<NR1>	Freigabe des Ereignisses(Event)
:ENABle?	-><NR1>	Abfrage
:PTRransition	<NR1>	Event, nur bei Übergang von 0 auf 1 weitergeleitet
:PTRransition?	-><NR1>	Abfrage
:NTRransition	<NR1>	Event wird nur bei Übergang von 1 auf 0 weitergeleitet
:NTRransition?	-><NR1>	Abfrage

Beispiele:

STAT:OPER? Abfrage des OPERation Status Event Register
STAT:OPER:ENAB_5888 Setzt alle Freigabebits (=Enable) im OPERation Status Event Register

3.5 Systembefehle

(Spezifikation nach „1999 SCPI Command reference“: 19 System Subsystem)

SYSTem:

ERRor:ALL?	-><Err>[,<Err>]...	Abfrage des Fehlerbuffers, Fehlermeldungen aus Fehlerliste lesen, die Bits <i>err</i> , <i>esr</i> sowie <i>ESR:Condition</i> werden gelöscht.
ERRor:NEXT?	-><Err>	Abfrage letzter Fehler, wenn die Fehlerliste leer wird, werden die Bits <i>err</i> , <i>esr</i> sowie <i>ESR:Condition</i> gelöscht.



Hinweis

Die Fehlerabfrage liefert bei einer GPIB-Karte Kommunikations und Gerätefehler zurück, bei einer Ethernetkarte nur Kommunikationsfehler.

LOCK

[:STATe]	<B0>	1 oder ON= Setzt das Gerät in Fernsteuerbetrieb, falls nicht blockiert. 0 oder OFF= Verlässt den Remotebetrieb
-----------------	------	---



Achtung!

Die Geräte können, ohne daß sie in den Fernsteuerbetrieb versetzt wurden, nur überwacht werden. Das bedeutet, man kann nur Istwerte und Zustände abfragen. Um Zustände und Sollwerte zu setzen, müssen sie vorher mit **LOCK:STAT 1** oder **LOCK 1** bzw. ***RST** (siehe auch 3.1) in den Fernsteuerbetrieb gesetzt werden.

Um dies zu können, darf der Fernsteuerbetrieb nicht durch andere Umstände blockiert sein. Über die Bedingungen für Freigabe/Sperre des Fernsteuerbetriebes lesen Sie bitte im Handbuch des Gerätes nach.

Die Freigabe kann über den folgenden Befehl abgefragt werden:

SYSTem:

LOCK

:OWNer? -><B1> Abfrage des Bedienortes
 NONE: Das Gerät kann in den Fernsteuerbetrieb geschaltet werden
 (Bit 8,9,10 =0 in *OPER:Condition*)
 LOCal: Das Gerät ist lokal und für den Fernsteuerbetrieb gesperrt
 (Bit 8=1,9=0,10=0 in *OPER:Condition*)
 Externbetrieb wird als LOCal gedeutet.
 (Bit 8=0,9=0,10=1 in *OPER:Condition*)
 REMote: Das Gerät ist Fernsteuerbetrieb
 (Bit 8=0,9=1,10=0 in *OPER Register*)

VERSION? -><SRD> Abfrage SCPI-Version

Beispiele:

SYST:LOCK:OWN? Fragt den Bedienort ab
SYST:LOCK:STAT_1 Setzt das Gerät in den zur Steuerung erforderlichen Fernsteuerbetrieb
LOCK_ON dito



Achtung!

Achtung! Die zwei folgenden Befehle werden nur von der Netzwerkkarten IF-E1, IF-E2B und IF-E1B unterstützt!

SYSTem:

DATA

:SET <CHAR> Eingeschlossenes Telegramm, aufgebaut nach objektorientiertem Protokoll
 Hier: Daten senden (**SET**) (siehe auch Abschnitt 3.9.1)
 Nähere Informationen zu dem eingeschlossenen Telegramm sind im externen Hand-
 buch „[Programmierung](#)“, sowie den zugehörigen [Objektlisten](#)-Dokumenten zu finden.

Beispiel:

SYST:DATA:SET_50,_100,_0 Schickt das hexadezimale Telegramm 0x32 0x64 0x00 an das Gerät.
 Dieses setzt, wenn das Gerät im Fernsteuerbetrieb ist, die Ausgangsspannung auf 100%.

SYSTem:

DATA

:REQuest <CHAR> Eingeschlossenes Telegramm, aufgebaut nach objektorientiertem Protokoll
 Hier: Daten abfragen (**REQ**) (siehe auch Abschnitt 3.9.1)
 Nähere Informationen zu dem eingeschlossenen Telegramm sind im externen Hand-
 buch „[Programmierung](#)“, sowie den zugehörigen [Objektlisten](#)-Dokumenten zu finden.

Beispiel:

SYST:DATA:REQ_50 Schickt das hexadezimale Telegramm 0x32 an das Gerät.
 Damit wird der zuletzt gesetzte Ausgangsspannungswert abgefragt.
 Entspricht prinzipiell dem SCPI-Befehl **SOUR:VOLT?**
 Die Antwort ist eine Bytefolge aus Dezimalzahlen, z. B. **100,0**. Das entspricht dem
 Hexwert 0x6400 und bedeutet 100% Sollwert.

**Achtung!**

Achtung! Der nachfolgende Befehl wird nur von der Netzwerkkarte IF-E1B unterstützt!

SYSTem:**COMMunicate:****NETwork**

:MAC?	-><SRD>	Abfrage der MAC-Adresse der Ethernetkarte
:IPADdress?	-><SRD>	Abfrage der momentanen IP-Adresse, wird im üblichen Format ausgegeben. Beispiel: 192.168.0.2
:IPADdress	<CHAR>	IP-Adresse über Befehl setzen, mit kommasetrennten Dezimalzahlen. Diese wird erst nach Aus- und Wiedereinschalten des Gerätes aktiv.
:MASK?	-><SRD>	Abfrage der momentanen Subnetzmaske, wird im üblichen Format ausgegeben. Beispiel: 255.0.0.0
:MASK	<CHAR>	Subnetzmarke über Befehl setzen, mit kommasetrennten Dezimalzahlen. Diese wird erst nach Aus- und Wiedereinschalten des Gerätes aktiv.
:GATEway?	-><SRD>	Abfrage der momentanen Gateway-Adresse, wird im üblichen Format ausgegeben. Beispiel: 0.0.0.0
:GATEway	<CHAR>	Gateway-Adresse über Befehl setzen, mit kommasetrennten Dezimalzahlen. Diese wird erst nach Aus- und Wiedereinschalten des Gerätes aktiv.

Beispiel:

SYST:COMM:NET:IPAD_192,_168,_0,_2 Setzt die IP 192.168.0.2, falls sich das Gerät in Fernsteuerung befindet.

3.6 Befehle zur Steuerung des Ausgangs

Leistungsausgang aktivieren/deaktivieren.

OUTPut[:STATe]?	-><B0>	Abfrage Zustand des Leistungsausgangs
OUTPut[:STATe]	<B0>	Schaltet den Leistungsausgang ein oder aus

Beispiele:

OUTP_ON Schaltet den Leistungsausgang ein, setzt aber nicht die Alarmer und Warnungen zurück oder quittiert sie nicht. D.h., steht ein Alarm an, kann der Befehl nicht ausgeführt werden.

3.7 Meßbefehle

Anfrage der aktuellen Istwerte.

MEASure

[:SCALar]

:VOLTage[:DC]?	-><NRf>Unit	Abfrage: Spannungswert
:CURRent[:DC]?	-><NRf>Unit	Abfrage: Stromwert
:POWer[:DC]?	-><NRf>Unit	Abfrage: Leistungswert
:[ARRay]?	-><NRf>Unit, <NRf>Unit ...	Abfrage: Spannungswert, Stromwert, Leistungswert

Beispiele:

MEAS:CURR? Mißt und liefert den aktuellen Strom.

MEAS:ARR? Gibt eine Reihe von Istwerten zurück. Dies sind: U, I, P

3.8 Sollwertbefehle

Hinweis

Durch Anhängen eines Fragezeichens können alle Sollwerte auch ausgelesen werden.

I. Spannungssollwert / Überspannungsgrenze

(Spezifikation nach „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]

VOLTage

[:LEVel]?	-><NRf>Unit	Abfrage letzter Spannungssollwert
[:LEVel]	<NRf+>[Unit]	Spannungssollwert setzen
:PROTect[:LEVel]	<NRf+>[Unit]	OVP-Spannung setzen (nur wenn Ausgang aus)
:PROTect[:LEVel]?	-><NRf>Unit	Abfrage: OVP-Schwelle

Beispiele:

VOLT_5.05 Setzt 5,05 V Ausgangsspannung am Netzgerät

VOLT_6.91_V Setzt 6,91 V Spannung, mit Einheit angegeben

VOLT? Fragt den zuletzt gesetzten Spannungssollwert ab

SOUR:VOLT:PROT_67 Setzt die Überspannungsgrenze (OVP) auf 67 V, wenn der Ausgang ausgeschaltet ist. Ansonsten wird nichts übernommen und ein Fehler erzeugt.

! Hinweis

Der Maximalwert für den Befehl `SOUR:VOLT:PROT` entspricht generell dem Wert, dem man am Gerät für die OVP-Schwelle einstellen kann. Er ist typischerweise 110% der Gerätenennspannung. Bei einem 360 V-Gerät sollte also ein Wert für OVP von 396 V möglich sein (`SOUR:VOLT:PROT 396`). Aufgrund von Umrechnungsfehlern wird der absolute Endwert hier nicht akzeptiert und mit dem Fehler „Data out of range“ returniert werden. Es wird daher empfohlen, einen kleineren Wert zu setzen, z. B. 395 V.

! Hinweis

Betrifft Geräte der Serien PSI 8000 und PSI 9000: beim Setzen der Überspannungsgrenze kann ein gewisser Wert (z.B. 200 V bei einem 360 V-Gerät) nicht akzeptiert werden, wenn am Gerät im Menü „Profile -> Supervision -> U thresholds“ der Wert „U >“ auf einen höheren Wert eingestellt wurde, als der gewünschte OVP-Wert. Siehe auch Gerätehandbuch zur Bedeutung der Überwachungsfunktion (Supervision).

II. Stromsollwert

(Spezifikation nach „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]**CURRent**

[:LEVel]?

-><NRf>[Unit]

Abfrage letzter Stromsollwert

[:LEVel]

<NRf+>Unit

Stromsollwert setzen

Beispiele:

CURR_20.00

Setzt den max. Ausgangsstrom auf 20 A.

III. Leistungssollwert

(Spezifikation nach „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]**POWer**

[:LEVel]?

-><NRf>Unit

Abfrage letzter Leistungssollwert

[:LEVel]

<NRf+>[Unit]

Leistungssollwert setzen

Beispiele:

POW:LEV_2300

Setzt das Gerät auf 2300 W Leistungsbegrenzung, sofern dieser Wert zulässig ist
Wenn ein Gerät darauf nicht reagiert, hat es möglicherweise keine einstellbare Leistung.

IV. Innenwiderstandssollwert

(Spezifikation nach „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]**RESistance**

[:LEVel]?

-><NRf>Unit

Abfrage letzter Widerstandssollwert

[:LEVel]

<NRf+>[Unit]

Widerstandssollwert setzen

Beispiel:

RES_1.300

Setzt den gewünschten Innenwiderstandssollwert auf 1,3Ω.

3.9 Sonderbefehle

3.9.1 SYST:DATA:SET und SYST:DATA:REQ

Hinweis

Diese beiden Befehle werden nur von der Ethernetkarte IF-E1B unterstützt.

Die Netzwerkkarte IF-E1B arbeiten **nur über den Ethernetport** mit SCPI-Befehlen, die ab Abschnitt 3 beschrieben sind. Besonderheit ist hier, daß es zwei SCPI-Befehle gibt, die ein Telegramm, aufgebaut ähnlich dem objektorientierten Kommunikationsprotokoll, wie im externen Handbuch „[Programmierung](#)“ beschrieben, transportieren können.

Der Sinn dieser Befehle ist es, Kommandos an das Gerät zu senden für die es keinen entsprechenden SCPI-Befehl gibt. So kann man über das quasi-binäre Protokoll z. B. den Funktionsmanager der Geräteserien PSI 9000 und PSI 8000 steuern, laden und abfragen, was mit SCPI-Befehlen nicht möglich wäre. Um dies zu tun ist ein Telegramm mit dem Aufbau

SYST:DATA:SET **ON**, **DATEN** bzw.

SYST:DATA:REQ **ON**

zu erstellen und die Werte für **ON** und **DATEN** als Dezimalzahlen an das Gerät zu senden. *Wichtig: Alle Bytes müssen durch Kommas getrennt angegeben werden.*

DATEN ist erforderlich, wenn ein Wert oder mehrere an das Gerät gesendet werden. Die Anzahl der **DATEN** muß stimmen, ansonsten wird ein Fehler erzeugt. Die Datenlänge ergibt sich aus dem jeweiligen Objekt in den Objektlisten (siehe die externen [Objektlisten](#)). Beim Setzen eines Spannungssollwertes wird z. B. ein 16-Bit-Wert geschickt, hier dann zwei kommagetrennte Dezimalzahlen, die das Highbyte und Lowbyte repräsentieren des 16-Bit-Wertes repräsentieren.

Wir unterscheiden hier grundsätzlich zwischen Telegrammen, die nur etwas senden (SYST:DATA:SET) und welche, die etwas abfragen (SYST:DATA:REQ).

Der erste Wert, **ON**, ist die Objektnummer, quasi ein Befehl. Sie ist auch in Spalte 1 der Objektlisten zu finden. Diese Nummer stellt ein Ziel für die nachfolgenden Daten dar. Objektnummer und Daten bilden zusammen einen Stell-Befehl, während die Objektnummer allein einen Abfrage-Befehl bildet.

Beispiel 1:

Es soll die Bytefolge 0x4700 als Spannungssollwert transportiert werden. Laut der [Objektliste](#) für z. B. ein PSI 8000 Gerät ist der Spannungssollwert Objekt 50.

Der sich dann ergebende SCPI-Befehl müßte so aussehen:

SYST:DATA:SET **50**, **71**, **0**

Beispiel 2:

Es sollen die Istwerte (Spannung, Strom, Leistung) angefragt werden. Dazu wird eine Anfrage (Request) gestellt. Laut [Objektliste](#) liefert Objekt 71 alle drei Istwerte auf einmal, die Anzahl der angefragten Bytes ist 6. Die Anfrage sieht dann also so aus:

SYST:DATA:REQ **71**

Das Gerät antwortet dann z. B. 6 Zahlen in dezimaler Form:

71, **67**, **37**, **21**, **127**, **24**, **16**

Der erste Werte ist die Objektnummer, der Rest die Istwerte. Zwei aufeinanderfolgende Zahlen ergeben jeweils einen 16Bit-Wert, der einen Istwert in Prozent darstellt. Die sechs Bytes ergeben in hexadezimaler Form und jeweils zusammengefaßt:

67, **37**, **21**, **127**, **24**, **16**

\ | \ | \ | \ |

0x4325, 0x157F, 0x1810

Die Umrechnung der zwei Werte in einen Dezimal- oder Hexadezimalwerte kann aber auch so erfolgen (VB, C o.ä.):

Prozentwert = Erster Wert * 256 + Zweiter Wert

Für die beiden ersten, 67 und 37, ergäbe das 17189 dezimal, also wieder die 0x4325 hexadezimal.

In der festgelegten Reihenfolge ist der erste der zusammengefaßten Werte der Spannungsiswert, der zweite der Stromiswert und der dritte der Leistungsiswert, jeweils in Prozent vom Nennwert des Gerätes. Für die Umrechnung der Prozentwerte in Realwerte siehe auch Abschnitt 1.7 im externen Handbuch „[Programmierung](#)“.

Um den erhaltenen Prozentwert in einen echten Istwert umzurechnen, muß man wissen, wie die Nennwerte des Netzgerätes sind. Mal angenommen das wären 80 V, 100 A und 3000 W, wie bei einem PSI 9080-100, dann ergäbe sich eine zweite Formel:

Realer Istwert = Nennwert * Prozentwert / 25600

Für das Beispiel oben und den Stromiswert 0x157F ergäbe das:

100 A * 5503 / 25600 = 21,5 A

Hinweis

Dieses Beispiel entspricht dem SCPI-Befehl MEAS:ARR?. Das heißt, der Befehl würde die errechneten Istwerte auch und direkt liefern, eventuell mit leichter Abweichung (meist dritte Nachkommastelle) wegen der Umrechnung.

4. Fehlermeldungen

Fehlermeldungen werden in einer Fehlerliste gesammelt. Das **err** Bit im Statusbyte STB zeigt an, wenn eine neuer Fehler aufgetreten ist. Der oder die aufgetretenen Fehler können einzeln mit ERR:NEXT? oder alle zusammen mit ERR:ALL? abgefragt werden. Das Auslesen aller Fehler löscht das **err** Bit und den Fehlerpuffer.

<ERR>	Meldung	Beschreibung
0	"No error"	Error Queue leer, es liegt kein Fehler vor
-100	"Command error"	Falscher SCPI Befehl
-101	"Invalid character"	Ungültiges Zeichen im Befehlsstring
-102	"Syntax error"	Syntaxfehler
-103	"Invalid separator"	Ungültiges Trennzeichen
-108	"Parameter not allowed"	Parameter nicht erlaubt
-109	"Missing parameter"	Parameter fehlt
-113	"Undefined header"	Befehl unbekannt
-120	"Numeric data error"	Fehler im Zahlenformat
-131	"Invalid suffix"	Falsche Einheit
-141	"Invalid character data"	Nicht erlaubtes Zeichen wurde übertragen
-151	"Invalid string data"	Befehltext nicht korrekt
-200	"Execution error"	Ausführungsfehler
-201	"Invalid while in local"	Gerät ist in Lokalbetrieb (Remote ist gesperrt)
-203	"Command protected"	Zugriff auf Seq.-steuerelementen verweigert
		Freischaltung fehlt
		Zugriff auf Funktionsdefinitionen verweigert
-223	"Too much data"	Zuviele Daten übermittelt
-224	"Illegal parameter value"	nicht zulässiger Parameter empfangen
-225	"Out of memory"	Speicherüberlauf
-240	"Hardware error"	max. CAN-Nodes überschritten / CAN-Node unbekannt / kein Gateway
-241	"Hardware missing"	Hardware fehlt bzw. wurde nicht erkannt
-220	"Parameter error"	Objekt nicht definiert
-221	"Settings conflict"	Schreib-Leserechte verletzt, kein Zugriff
		Zugriff auf Menüparam. nur bei OUTPUT OFF
		Zugriff auf Sollwert verweigert (Gerät im Slave-Betrieb)
-222	"Data out of range"	Wert überschreitet das obere Limit
		Wert überschreitet das untere Limit
-223	"Too much data"	Objektlänge nicht korrekt
-232	"Invalid format"	Zeitformat ist falsch
-350	"Queue overflow"	Fehler- und Ereignispuffer ist übergelaufen
-360	"Communication error"	Zusammenfassung von Kommunikationsfehlern (siehe auch Abschnitt 3.7 des Handbuches „ Programmierung “):
		Framing Fehler
		Prüfsumme nicht korrekt
		etc.
-361	"Parity error in program message"	bei RS232: Parityfehler wurde erkannt
-362	"Framing error in program message"	
-363	"Input buffer overrun"	Empfangsbuffer ist übergelaufen
-365	"Time out error"	Zeitüberschreitung beim Senden/Empfangen

Weitere Fehlermeldungen ergeben sich aus den geräteabhängigen Alarmen, Warnungen und Meldungen (siehe auch das Geräte-Handbuch bezüglich der Definition und das Handbuch zur Schnittstellenkarte wegen der Meldung):

<ERR>	Meldung	Beschreibung
100 - 199	siehe Alarmtabelle im Abschnitt 3.9 des Handbuches „ Programmierung “	nur Anzeige, die Zehner- und Einerstellen des Fehlercodes sind identisch mit den Fehlercodes in der Alarmtabelle oben
200 - 299	siehe Alarmtabelle im Abschnitt 3.9 des Handbuches „ Programmierung “	Warnungen, die Zehner- und Einerstellen des Fehlercodes sind identisch mit den Fehlercodes in der Alarmtabelle oben
300 - 399	siehe Alarmtabelle im Abschnitt 3.9 des Handbuches „ Programmierung “	Alarme, die Zehner- und Einerstellen des Fehlercodes sind identisch mit den Fehlercodes in der Alarmtabelle oben

5. Anhang

5.1 SCPI-Befehlsübersicht

Main	1.Sub	2.Sub	3.Sub	Wert(e)	PS8000T	PS8000DT/2U/3U	PS8000T/DT/2U/3U	PS8000R	PS9000	EL-3000/EL9000	Hinweis
CURR				0...Imax	*	*	*	*	*	*	
CURR?				Sollwert Strom	*	*	*	*	*	*	
CURR: LEV				0...Imax	*	*	*	*	*	*	Bei EL für A oder B, jenachdem was eingestellt wurde
CURR: LEV?				Sollwert Strom	*	*	*	*	*	*	Bei EL von A oder B, jenachdem was eingestellt wurde
CURR: HIGH				CURR:LOW...Imax	*	*	*	*	*	*	Bei EL für AB
CURR: HIGH?				0...Imax					*	*	Bei EL von AB
CURR: LOW				0...CURR:HIGH,					*	*	Bei EL für AB
CURR: LOW?				0...Imax					*	*	Bei EL von AB
ERR: ALL?				Bis zu 3 Fehlertexte mit Nummern	*	*	*	*	*	*	
ERR: NEXT?				1 Fehlertext mit Nummer	*	*	*	*	*	*	
INP				1, 0, ON, OFF					*	*	
INP?				ON, OFF					*	*	
INP: STAT				1, 0, ON, OFF					*	*	
INP: STAT?				ON, OFF					*	*	
LOCK				1, 0, ON, OFF	*	*	*	*	*	*	
LOCK?					*	*	*	*	*	*	
LOCK: STAT				1, 0, ON, OFF	*	*	*	*	*	*	
LOCK: STAT?					*	*	*	*	*	*	
LOCK: OWN?				REM, LOC, NONE	*	*	*	*	*	*	
MEAS: VOLT?				1 Wert	*	*	*	*	*	*	
MEAS: CURR?				1 Wert	*	*	*	*	*	*	
MEAS: POW?				1 Wert	*	*	*	*	*	*	
MEAS: ARR?				3 Werte	*	*	*	*	*	*	
MEAS: VOLT: DC?				1 Wert	*	*	*	*	*	*	
MEAS: CURR: DC?				1 Wert	*	*	*	*	*	*	
MEAS: POW: DC?				1 Wert	*	*	*	*	*	*	
MEAS: SCAL: VOLT?				1 Wert	*	*	*	*	*	*	
MEAS: SCAL: CURR?				1 Wert	*	*	*	*	*	*	
MEAS: SCAL: POW?				1 Wert	*	*	*	*	*	*	
MEAS: SCAL: ARR?				3 Werte	*	*	*	*	*	*	
MEAS: SCAL: VOLT: DC?				1 Wert	*	*	*	*	*	*	
MEAS: SCAL: CURR: DC?				1 Wert	*	*	*	*	*	*	
MEAS: SCAL: POW: DC?				1 Wert	*	*	*	*	*	*	
OUTP				1, 0, ON, OFF	*	*	*	*	*	*	
OUTP?				ON, OFF	*	*	*	*	*	*	
OUTP: STAT				1, 0, ON, OFF	*	*	*	*	*	*	
OUTP: STAT?				ON, OFF	*	*	*	*	*	*	
POW				0...Pmax	*	*	*	*	*	*	
POW?				Sollwert Leistung	*	*	*	*	*	*	
POW: LEV				0...Pmax	*	*	*	*	*	*	
POW: LEV?				Sollwert Leistung	*	*	*	*	*	*	
POW: HIGH				POW:LOW...Pmax					*	*	
POW: HIGH?				0...Pmax					*	*	
POW: LOW				0...POW:HIGH					*	*	
POW: LOW?				0...Pmax					*	*	
PULS: WIDT: LOW				50us...100s					*	*	Bei EL für AB
PULS: WIDT: LOW?				50us...100s					*	*	Bei EL von AB
PULS: WIDT: HIGH				50us...100s					*	*	Bei EL für AB
PULS: WIDT: HIGH?				50us...100s					*	*	Bei EL von AB
PULS: TRAN				30us...200ms					*	*	Bei EL für AB
PULS: TRAN?				30us...200ms					*	*	Bei EL von AB
PULS: TRAN: LEAD				30us...200ms					*	*	Bei EL für AB
PULS: TRAN: LEAD?				30us...200ms					*	*	Bei EL von AB
RES				0...Rmax bzw. R1/R2			*	*	*	*	Bei EL für A oder B, jenachdem was eingestellt wurde
RES?				Sollwert Widerstand			*	*	*	*	Bei EL von A oder B, jenachdem was eingestellt wurde
RES: LEV				0...Rmax bzw. R1/R2			*	*	*	*	Bei EL für A oder B, jenachdem was eingestellt wurde
RES: LEV?				Sollwert Widerstand			*	*	*	*	Bei EL von A oder B, jenachdem was eingestellt wurde
RES: HIGH				RES:LOW...Rmax					*	*	
RES: HIGH?				0...Rmax					*	*	
RES: LOW				0...RES:HIGH					*	*	
RES: LOW?				0...Rmax					*	*	
SOUR: VOLT				0...Umax	*	*	*	*	*	*	
SOUR: VOLT?				Sollwert Spannung	*	*	*	*	*	*	
SOUR: VOLT: LEV				0...Umax	*	*	*	*	*	*	
SOUR: VOLT: LEV?				Sollwert Spannung	*	*	*	*	*	*	
SOUR: VOLT: HIGH				VOLT:LOW...Umax					*	*	
SOUR: VOLT: HIGH?				0...Umax					*	*	
SOUR: VOLT: LOW				0...VOLT:HIGH					*	*	
SOUR: VOLT: LOW?				0...Umax					*	*	

Main	1.Sub	2.Sub	3.Sub	Wert(e)							Hinweis
					PS8000T	PS8000DT/2U/3U	PS18000T/DT/2U/3U	PS1800R	PS19000	EL3000/EL9000	
SOUR: CURR:				dito	*	*	*	*	*	*	
SOUR: CURR?				dito	*	*	*	*	*	*	
SOUR: CURR: LEV				dito	*	*	*	*	*	*	
SOUR: CURR: LEV?				dito	*	*	*	*	*	*	
SOUR: CURR: HIGH				dito						*	
SOUR: CURR: HIGH?				dito						*	
SOUR: CURR: LOW				dito						*	
SOUR: CURR: LOW?				dito						*	
SOUR: POW				dito	*	*	*	*	*	*	
SOUR: POW?				dito	*	*	*	*	*	*	
SOUR: POW: LEV				dito	*	*	*	*	*	*	
SOUR: POW: LEV?				dito	*	*	*	*	*	*	
SOUR: POW: HIGH				dito						*	
SOUR: POW: HIGH?				dito						*	
SOUR: POW: LOW				dito						*	
SOUR: POW: LOW?				dito						*	
SOUR: RES				dito			*	*	*	*	
SOUR: RES?				dito			*	*	*	*	
SOUR: RES: LEV				dito			*	*	*	*	
SOUR: RES: LEV?				dito			*	*	*	*	
SOUR: RES: HIGH				dito						*	
SOUR: RES: HIGH?				dito						*	
SOUR: RES: LOW				dito						*	
SOUR: RES: LOW?				dito						*	
SOUR: VOLT: PROT				0...110% Umax	*	*	*	*	*	*	OVP
SOUR: VOLT: PROT?				0...110% Umax	*	*	*	*	*	*	OVP
SOUR: VOLT: PROT: LEV				0...110% Umax	*	*	*	*	*	*	OVP
SOUR: VOLT: PROT: LEV?				0...110% Umax	*	*	*	*	*	*	OVP
SOUR: PULS: WIDT: LOW				dito						*	
SOUR: PULS: WIDT: LOW?				dito						*	
SOUR: PULS: WIDT: HIGH				dito						*	
SOUR: PULS: WIDT: HIGH?				dito						*	
SOUR: PULS: TRAN				dito						*	
SOUR: PULS: TRAN?				dito						*	
SOUR: PULS: TRAN: LEAD				dito						*	
SOUR: PULS: TRAN: LEAD?				dito						*	
STAT: OPER?					*	*	*	*	*	*	
STAT: OPER: EVENT?				0-32767	*	*	*	*	*	*	
STAT: OPER: COND?				0-32767	*	*	*	*	*	*	
STAT: OPER: ENAB				0-32767	*	*	*	*	*	*	
STAT: OPER: ENAB?				0-32767	*	*	*	*	*	*	
STAT: OPER: PTR				0-32767	*	*	*	*	*	*	
STAT: OPER: PTR?				0-32767	*	*	*	*	*	*	
STAT: OPER: NTR				0-32767	*	*	*	*	*	*	
STAT: OPER: NTR?				0-32767	*	*	*	*	*	*	
STAT: QUES?				0-32767	*	*	*	*	*	*	
STAT: QUES: EVENT?				0-32767	*	*	*	*	*	*	
STAT: QUES: COND?				0-32767	*	*	*	*	*	*	
STAT: QUES: ENAB				0-32767	*	*	*	*	*	*	
STAT: QUES: ENAB?				0-32767	*	*	*	*	*	*	
STAT: QUES: PTR				0-32767	*	*	*	*	*	*	
STAT: QUES: PTR?				0-32767	*	*	*	*	*	*	
STAT: QUES: NTR				0-32767	*	*	*	*	*	*	
STAT: QUES: NTR?				0-32767	*	*	*	*	*	*	
SYST: COMM: NET: IPAD				0, 0, 0, 0 - 255, 255, 255, 255	*	*	*	*	*	*	Nur IF-E1B und IF-E2B
SYST: COMM: NET: IPAD?				0.0.0.0 - 255.255.255.255	*	*	*	*	*	*	Nur IF-E1B und IF-E2B
SYST: COMM: NET: GATE				0, 0, 0, 0 - 255, 255, 255, 255	*	*	*	*	*	*	Nur IF-E1B und IF-E2B
SYST: COMM: NET: GATE?				0.0.0.0 - 255.255.255.255	*	*	*	*	*	*	Nur IF-E1B und IF-E2B
SYST: COMM: NET: MAC?					*	*	*	*	*	*	Nur IF-E1B und IF-E2B
SYST: COMM: NET: MASK				0, 0, 0, 0 - 255, 255, 255, 255	*	*	*	*	*	*	Nur IF-E1B und IF-E2B
SYST: COMM: NET: MASK?				0.0.0.0 - 255.255.255.255	*	*	*	*	*	*	Nur IF-E1B und IF-E2B
SYST: DATA: SET				Objekttelegramm als ASCII	*	*	*	*	*	*	Nur IF-Ex, nur Setzen
SYST: DATA: REQ				Objekttelegramm als ASCII	*	*	*	*	*	*	Nur IF-Ex, Antwort wird gesendet
SYST: ERR: ALL?				dito	*	*	*	*	*	*	
SYST: ERR: NEXT?				dito	*	*	*	*	*	*	
SYST: LOCK				dito	*	*	*	*	*	*	
SYST: LOCK: STAT				dito	*	*	*	*	*	*	
SYST: LOCK: OWN?				dito	*	*	*	*	*	*	
SYST: VERS?				SCPI-Version (1999.0)	*	*	*	*	*	*	

Main	1.Sub	2.Sub	3.Sub	Wert(e)	PS8000T	PS8000DT/2U/3U	PS8000T/DT/2U/3U	PS8000R	PS9000	EL3000/EL9000	Hinweis
VOLT				dito	*	*	*	*	*	*	
VOLT?				dito	*	*	*	*	*	*	
VOLT: LEV				dito	*	*	*	*	*	*	
VOLT: LEV?				dito	*	*	*	*	*	*	
VOLT: HIGH				dito					*		
VOLT: HIGH?				dito					*		
VOLT: LOW				dito					*		
VOLT: LOW?				dito					*		
VOLT: PROT				dito	*	*	*	*	*		OVP
VOLT: PROT?				dito	*	*	*	*	*		OVP
VOLT: PROT: LEV				dito	*	*	*	*	*		OVP
VOLT: PROT: LEV?				dito	*	*	*	*	*		OVP
*RST					*	*	*	*	*	*	
*IDN?				String, max. 128 Zeichen	*	*	*	*	*	*	
*STB?				0...255	*	*	*	*	*	*	
*ESR?				0...255	*	*	*	*	*	*	
*ESE				0...255	*	*	*	*	*	*	
*ESE?				0...255	*	*	*	*	*	*	
*CLS					*	*	*	*	*	*	
*TRG					*	*	*	*	*	*	
*SRE				0...255	*	*	*	*	*	*	
*SRE?				0...255	*	*	*	*	*	*	

	erfordert, daß das Gerät bereits in Fernsteuerbetrieb ist
	nur bei Ethernetkarte IF-E1 oder IF-E2 verfügbar, manche Befehle erfordern Fernsteuerbetrieb
	nur bei GPIB-Karte IF-G1 verfügbar
	nur für bestimmte Schnittstellen

	Page
1. End tokens	18
2. Syntax format	18
3. SCPI commands	19
3.1 Common IEEE488.2 commands	19
3.2 Event system	19
3.3 Register block	20
3.4 Status commands	21
3.5 System commands	21
3.6 Commands to control the output	24
3.7 Measurement commands	24
3.8 Set value commands	24
3.9 Special commands	26
4. Errors	27
5. Appendix	28
5.1 SCPI command overview	28

1. End tokens

Note

This applies only to GPIB (IF-G1)! Using Ethernet, no special end token is required.

SCPI commands are sent as plain text. The end has to be marked with a delimiter:

LF (Line Feed, 0xA, ASCII 10).

A transmission from the device requires to first sent a message from the host (PC/SPC etc). The IF-G1 will reply, if the host expects a reply. This is the case, if the end of the command is a „?“.

Commands that have to set a state or a value always consist of the command itself and one or multiple values, separated by commas. Example:

<COMMAND> „<Numeric value 1>, „<Numeric value 2>...

You can sent any command in its short or complete form. In the following section the short form is given in **capital** letters and is always a part of the complete form.

Changes since IF-G1 firmware version 3.03:

- Additional delimiters accepted:

CR+LF

CR

each together with **EOI**

- Commands can also be given in lowercase letters

2. Syntax format

Specification according „1999 SCPI Command reference“.

Following syntax formats can occur in commands and/or replies:

<Numeric value>

This numeric value corresponds to the value in the display of the device and depends on the nominal values of the device. It applies:

- the value must be sent after the command and separated by a space
- instead of a numeric value you can also use
MIN (corresponds to the minimum value of the parameter)

Note

Set values bigger than nominal values will return an error!

<NR1>	Numeric value without decimal place	
<NR2>	Numeric value with decimal place	
<NR3>	Numeric value with decimal place and scale	
<NRf>	Contains <NR1> or <NR2> or <NR3>	
Unit	V	Volt
	A	Ampere
	W	Watt
	OHM	Ohm
	s	Seconds

<CHAR>	0..255: Decimal number (output)
<+INT>	0..32768: positive integer number (output)
<B0>	1 or ON: Function is activated 0 or OFF: Function is deactivated.
<B1>	NONE: local operation, switching to remote control is possible LOCAL: local operation, reading of data is possible REMOte: remote control of the device is allowed
<B2>	ON or 1: automatic measurement with x rounds ONCE or 0: one-shot measurement with x rounds, triggered by *TRG
<ERR>	Error number (-800 to 399) and description
<SRD>	String
<LF>	End of line token (line feed, 0x0 A)
<Time>	[[ddd], [hh], [mm], [s]s.s[s][s][s][s][s] Standard format is in seconds (s.s)
;	The semicolon is used to separate multiple command within a message
:	The colon is used to separate major keywords from minor keywords
[]	Small letters and items in rectangular brackets are optional
?	The question mark identifies a query. The query can be combined with a data transmission. Here you need to take care to wait for response of the system before sending data
->	Reply from the device

3. SCPI commands

3.1 Common IEEE488.2 commands

- *IDN?** Returns the device identification, consisting of:
User text, device vendor, device name, device serial number, device firmware version, serial number(s) and firmware version(s) of plugged interface cards<LF>
- *RST** Resets the device by doing this:
- setting it into remote control mode (if allowed)
- setting output/input to OFF
- resetting all past alarms of the device
- *STB?** Reads the Status Byte Register, which is cleared after reading
- The following commands are only supported by the GPIB interface IF-G1:
- *TRG** Triggers a measurement
- *CLS** Clears all event and status registers of the GPIB controller
- *ESE <CHAR>** Sets the Event Status Enable Register
- *ESE?** Reads the Event Status Enable Register
- *ESR?** Reads the Event Status Register, which is cleared after reading
- *SRE <CHAR>** Sets the Service Request Enable Register
- *SRE?** Reads the Service Request Enable Register

3.2 Event system

The scheme of the event registration and register assignment is illustrated in the diagram on the next page.

Events can be queried from the device by reading the Status Register STB (GPIB, Ethernet).

The status register STB consists of these bits:

- Bit 0: Not used
- Bit 1: Not used
- Bit 2: **err**, Error queue full; this queue is cleared by reading it and the bit is also reset. The list can hold up to 4 of the last errors
- Bit 3: **ques**, Questionable status register is active (on or more events have occurred)
- Bit 4: Not used
- Bit 5: **esr**, the standard Event Status Register (ESR), masked by the Event Status Enable Register (ESE), is signalling that one or more events have occurred
- Bit 6: **rsv**, always active
- Bit 7: **oper**, signals, that one or more events have occurred and are stored in the Operation Status Register

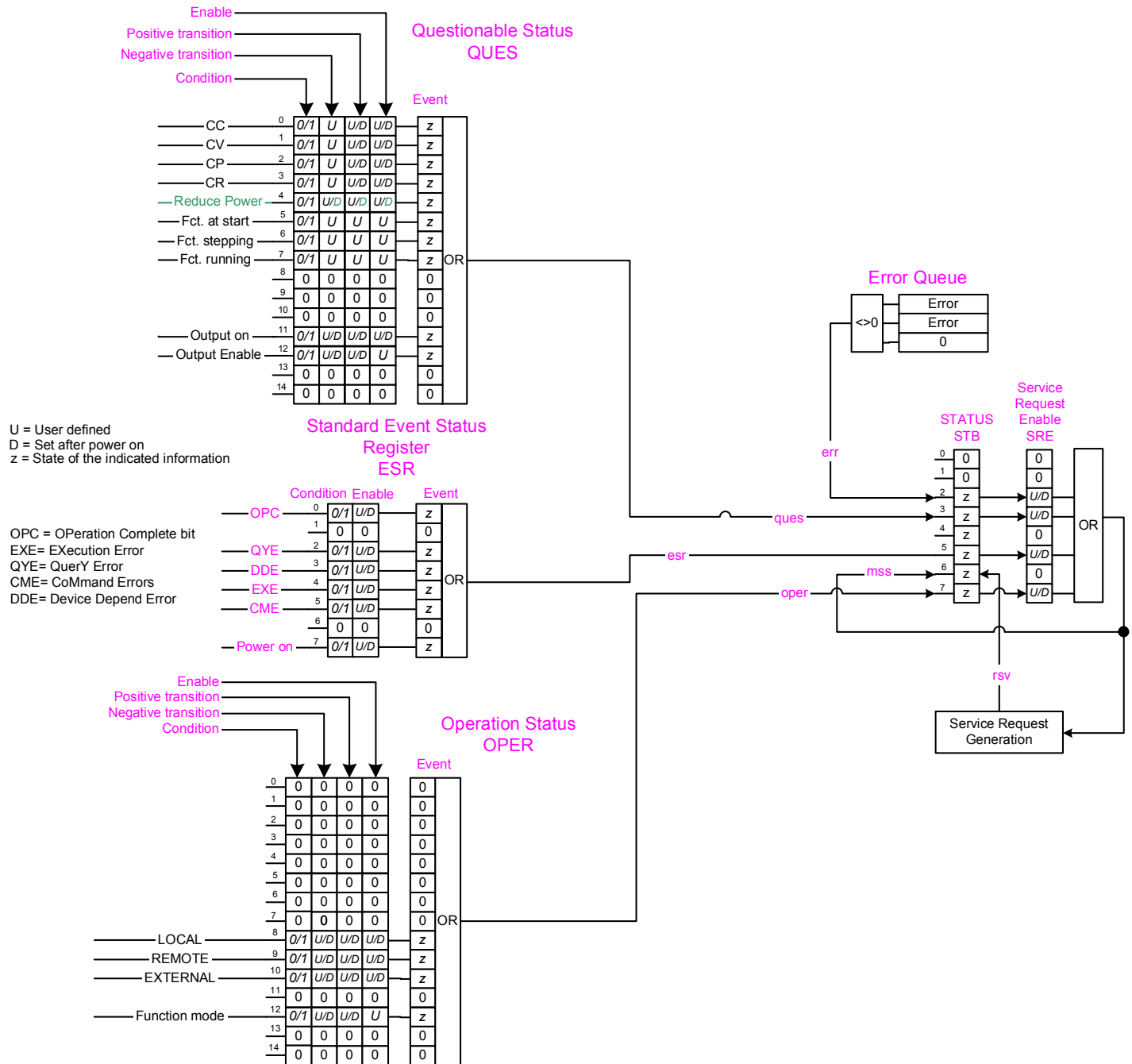
The event bits of the various registers report to the STB, if events have occurred that are enabled to be reported, by the corresponding bits in the enable registers (***ESE**, ***SRE** resp. **STAT:QUES:ENAB**, **STAT:OPER:ENAB**).

The bits of the *ESR* are as follows:

- Bit 0: Operation complete
- Bit 1: Not used
- Bit 2: Query error
- Bit 3: Device Dependent Error (Hardware defective etc.); errors from -399 to -300 resp. 100...399
- Bit 4: Execution Error (current limitation, other limits exceeded); errors from -299 to -200
- Bit 5: Command Error; Errors from -199 to -100
- Bit 6: Not used
- Bit 7: Power On (device was turned on)

Event and status registers can be cleared by using the command ***CLS**.

3.3 Register block



Legend:

CC/CV/CP/CR = currently active regulation mode

Reduce Power = power derating active (PSI 9000 series only)

Fct. at start/running/stepping = function manager status

Output on = Input resp. output of the device is on

Output enable = auto-on for the output is activated

LOCAL = device is in local mode, remote control is not allowed

REMOTE = device is remotely controlled by a digital interface card

EXTERNAL = device is controlled by the analogue interface card resp. the built-in analog interface

Function mode = function manager active

3.4 Status commands

(Specification according to „1999 SCPI Command reference“: 20 Status subsystem)

The Operation Status Register (*OPER*) (see diagram on previous page) stores the several status (remote, local etc.) in register *Condition* and forwards them to the register *Event*, as long as they are enabled by *Enable*. The masks *Positive transition* and *Negative transition* determine if the events are triggered by a rising edge or a falling edge. This can be used to sense the appearance and/or disappearance of a status.

The same applies for the Questionable Status Register (*QUES*). The configuration for the *OPER*, as shown in the diagram, would cause an event only if the signal „local“ changes from low to high. The signal „Function mode“, on the other hand, would cause the event also at a falling edge.

STATus

:OPERation		Operation depending status
[:EVENT]?	-><NR1>	Queries the events in the Status Operation Register
:CONDition?	-><NR1>	Query <i>OPER</i> event conditions
:ENABle	<NR1>	Enable events for <i>OPER</i>
:ENABle?	-><NR1>	Query
:PTRransition	<NR1>	Event will be triggered only at rising edge
:PTRransition?	-><NR1>	Query
:NTRransition	<NR1>	Event will be triggered only at falling edge
:NTRransition?	-><NR1>	Query
:QUESTionable		Device and function specific events
[:EVENT]?	-><NR1>	Query the events in the Questionable Status Register
:CONDition?	-><NR1>	Query <i>QUES</i> event conditions
:ENABle	<NR1>	Enable events for <i>QUES</i>
:ENABle?	-><NR1>	Query
:PTRransition	<NR1>	Event will be triggered only at rising edge
:PTRransition?	-><NR1>	Query
:NTRransition	<NR1>	Event will be triggered only at falling edge
:NTRransition?	-><NR1>	Query

Examples:

STAT:OPER? Queries the OPERation Status Event Register
STAT:OPER:ENAB_5888 Enables all available events for the OPERation Status Event register

3.5 System commands

(Specification according to „1999 SCPI Command reference“: 19 Source subsystem)

SYSTem:

ERRor:ALL?	-><Err>,<Err>...	Queries the error queue, used to read out error descriptions and codes; the bits <i>err</i> , <i>esr</i> and <i>ESR:Condition</i> are cleared
ERRor:NEXT?	-><Err>	Queries only the last error from the queue; if the queue is empty, bits <i>err</i> , <i>esr</i> and <i>ESR:Condition</i> are cleared



Note

When querying errors, a GPIB card returns communication and device errors, an Ethernet card only returns communication errors.

LOCK

[:STATE]	<B0>	1 or ON = puts the device into remote control mode, if not blocked 0 or OFF = exits remote control mode, returns to normal device operation
-----------------	------	--



Attention!

If not in remote mode, the device can only be monitored. It means, you can only query actual values and status. In order to set status, modes and set values, you need to switch the device to remote mode with **LOCK:STATE 1** or **LOCK 1** respectively ***RST** (also see section 3.1).

In order to do so, the remote mode must not be blocked. More information about conditions, blocks and modes can be found in the operating guide of your device.

The lock state can be queried by this command:

SYSTem:

LOCK

:OWNer? -><B1>

Get the current lock state

NONE: if returned, the device can be put to remote mode
(Bits 8,9,10 =0 in *OPER:Condition*)

LOCal: device is in local mode and blocked for remote mode
(Bits 8=1,9=0,10=0 in *OPER:Condition*)

External mode is interpreted as LOCal.

(Bit 8=0,9=0,10=1 in *OPER:Condition*)

REMOte: the device is in remote control mode

(Bit 8=0,9=1,10=0 in *OPER* Register)

VERSION? -><SRD>

Query SCPI version

Examples:

SYST:LOCK:OWN?

Queries the lock state to determine if remote mode is allowed

SYST:LOCK:STAT_1

Puts the device in remote control mode (setting of values allowed now)

LOCK_ON

Ditto



Attention!

The following two commands are only supported by the Ethernet cards IF-E1, IF-E2 and IF-E1B!

SYSTem:

DATA

:SET <CHAR>

Transports an encapsulated telegram in binary format

Here: Send data (**SET**) (also see section 3.9.1)

Further information about the encapsulated telegram can be found in the external guide „[Programming](#)“ and in the related [object list](#) files

Example:

SYST:DATA:SET_50,_100,_0

Sends the hexadecimal telegram 0x32 0x64 0x00 to the device.

If the device is in remote control, it will set the output voltage to 100%.

SYSTem:

DATA

:REQuest <CHAR>

Transports an encapsulated telegram in binary format

Here: REQuest data (also see section 3.9.1)

Further information about the encapsulated telegram can be found in the external guide „[Programming](#)“ and in the related [object list](#) files

Example:

SYST:DATA:REQ_50

Sends the hexadecimal telegram 0x32 to the device.

This requests the voltage set value and corresponds to the

SCPI command **SOUR:VOLT?** The response will be two values,

for example **100,0**. They correspond to the hexadecimal value 0x6400 and translate to 100% set value.

**Attention!**

The following command is only supported by the Ethernet card IF-E1B!

SYSTem:**COMMunicate:****NETwork**

:MAC?	-><SRD>	Query the MAC address of the Ethernet card
:IPADdress?	-><SRD>	Query the actual IP address of the unit, which will be returned in the typical format. Example: 192.168.0.2
:IPADdress	<CHAR>	Set IP address via command with comma separated decimals. This will only be effective if the unit is switched off and then on again.
:MASK?	-><SRD>	Query the actual subnet mask address of the unit, which will be returned in the typical format. Example: 255.0.0.0
:MASK	<CHAR>	Set subnet mask via command with comma separated decimals. This will only be effective if the unit is switched off and then on again.
:GATEway?	-><SRD>	Query the actual gateway address of the unit, which will be returned in the typical format. Example: 0.0.0.0
:GATEway	<CHAR>	Set gateway address via command, with comma separated decimals. This will only be effective if the unit is switched off and then on again.

Example:

SYST:COMM:NET:IPAD_192,_168,_0,_2

Will set IP 192.168.0.2, if device is in remote control.

3.6 Commands to control the output

Activates/deactivates the power output.

OUTPut[:STATe]?	-><B0>	Queries the state of the power supply output
OUTPut[:STATe]	<B0>	Switches the power supply output on/off

Examples:

OUTP_ON Switches the power output on, but does not reset alarms or warnings and also does not quit them. It means, if an alarm is persistent the command can't be executed.

3.7 Measurement commands

Used to read the latest measured results (actual values).

MEASure

[[:SCALar]

:VOLTage[:DC]?	-><NRf>Unit	Query: Actual voltage value
:CURRent[:DC]?	-><NRf>Unit	Query: Actual current value
:POWer[:DC]?	-><NRf>Unit	Query: Actual power value
:[ARRay]?	-><NRf>Unit, <NRf>Unit ...	Query: Actual voltage, actual current, actual power

Examples:

MEAS:CURR? Measures and returns the actual current.

MEAS:ARR? Returns a sequence of actual values. These are: U, I, P

3.8 Set value commands

Note

Set values can also be read back by attaching a question mark to the command.

I. Voltage set value/ Overvoltage threshold

(Specification according to „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]

VOLTage

[[:LEVel] ?	-><NRf>Unit	Queries the last set value for voltage
[[:LEVel]	<NRf+>[Unit]	Set voltage
:PROTectio[n]:[LEVel]	<NRf+>[Unit]	Set overvoltage (OVP) threshold (only if output is off)
:PROTectio[n]:[LEVel]?	-><NRf>Unit	Query the OVP threshold

Examples:

VOLT_5.05 Sets 5,05 V output voltage.

VOLT_6.91_V Set 6.91 V, given with unit.

VOLT? Queries the last set value

SOUR:VOLT:PROT_67 Sets the OVP threshold to 67 V, if the output of the device is off. Else the command is ignored and an error is generated.

Note

The maximum value for the command `SOUR:VOLT:PROT` is generally the same as the OVP threshold maximum you can manually adjust on the device. It is typically 110% of the nominal device voltage. For a 360 V model, a OVP value of 396 V should be accepted (`SOUR:VOLT:PROT 396`). But is not accepted in any case when trying to set the max. value, because of calculation errors. It will return a „Data out of range“ error. It is thus recommended to set a lower value like 395 V, for example.

Note

This applies to devices of series PSI 8000 and PSI 9000: when setting the overvoltage threshold value it may occur that the value is not accepted because it exceeds a certain supervision setting. Check the setting „U >“ in the device menu „Profile -> Supervision -> U thresholds“. If the desired OVP value is lower than U>, it is not accepted by the device. Also refer to the user guide of the power supply for details about the supervision feature.

II. Current set value

(Specification according to „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]**CURRent**

[:LEVel]?

-><NRf>[Unit]

Queries the last set value for current

[:LEVel]

<NRf+>Unit

Set current

Example:

CURR_20.00

Sets the current limit to 20 A.

III. Power set value

(Specification according to „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]**POWer**

[:LEVel]?

-><NRf>Unit

Queries the last set value for power

[:LEVel]

<NRf+>[Unit]

Set power

Examples:

POW:LEV_2300 W

Sets the device to 2300 W power limitation, as long as this value is permitted.

If a device does not react to this command, it might not feature an adjustable power value.

IV. Internal resistance set value

(Specification according to „1999 SCPI Command reference“:19 Source Subsystem)

[SOURce:]**RESistance**

[:LEVel]?

-><NRf>Unit

Queries the last set value for internal resistance

[:LEVel]

<NRf+>[Unit]

Sets the internal resistance set value

Examples:

RES_1.300

Sets the desired, internal resistance set value to 1.3Ω.

3.9 Special commands

3.9.1 SYST:DATA:SET and SYST:DATA:REQ

Note

This section only applies to the Ethernet card IF-E1B!

Only via the Ethernet port, the network card is working with SCPI commands, which are described starting section 3.

In addition, the card provides two extra commands which can be used to send a telegram to the device in hexadecimal form, which is similar to the object orientated binary protocol (as described in the external guide „[Programming](#)“). Purpose of these commands is to control features of the device where no particular SCPI command is available for. So you can even control the PSI 9000 or PSI 8000 series function manager via these two SCPI commands. In order to do this you only need to build a telegram like this:

SYST:DATA:SET_0N, 16 resp.

SYST:DATA:REQ_0N

All 0N and 16 bytes are decimal values and are sent to the device with the SYST:DATA:SET command. *Important: All bytes must be separated by commas!*

16 is required when sending something to the device, like a set value or a status. The number of 16 bytes must be correct, else the device will return an error. The 16 length is defined in the object lists (see external [object lists](#)) and varies from object to object.

For example, when sending a voltage set value, a 16 bit integer is required, which results in two bytes and two comma separated decimal values for the SYST:DATA:SET command.

The first value, 0N, represents the object number (i.e. like a command) from column 1 of the object lists. It defines the target for 16. The combination of object number and data is considered as a set command, while the object number alone is considered as a query command.

Example 1:

The value 0x4700 shall be sent as voltage set value. According to the [object list](#) of e.g. a PSI 8000 this is object 50.

The resulting SCPI command looks like this:

SYST:DATA:SET_50, 47, 0

Here, the value 0x4700 is separated into two bytes 0x47 and 0x00 and each one translated to decimal.

Example 2:

The actual values of voltage, current and power shall be queried from the device. This is done by a REQuest. According to the [object list](#), the object 71 returns all three actual values at once in form of six single bytes. The request would look like this:

SYST:DATA:REQ_71

The device will, for example return, six decimal values:

71, 67, 37, 21, 127, 24, 16

The first value is the object, the rest is the actual values. Two subsequent values build a 16 bit value, which represents the actual value as per cent. So the 6 values result in these hexadecimal, combined values:

67, 37, 21, 127, 24, 16

\\ \\ \\ \\ \\ \\
0x4325, 0x157F, 0x1810

The calculation into decimal or hexadecimal per cent values can also be done like this (VB, C or similar):

Per cent value = First value * 256 + Second value

For the first two bytes of the example above, this will calculate to a decimal of 17198, ie. 67 * 256 + 37, which again represents the hexadecimal value of 0x4325.

The order of the resulting three 16 bit values is defined:

First value is actual voltage, the second the actual current and the third the actual power, each in per cent of the nominal value of the device. After this, they need to be calculated to real values. See section 1.7 of the external guide „[Programming](#)“.

In order to translate the calculated per cent value into a real actual value, the nominal values of the load required as reference. If, for example, the power supply has 80 V, 100 A and 3000 W, like with a PSI 9080-100, a second formula would be this:

Real actual value = Nominal value * per cent value / 25600

For the example above and the per cent value of the current 0x157F, it would result like this:

100 A * 5503 / 25600 = 21.5 A

Note

This example corresponds to the SCPI command MEAS:ARR?. It means, that the command would return the same values as the calculated ones. It may occur that values returned by MEAS:ARR? command slightly differ in the decimal places due to a small translation error.

4. Errors

Errors are collected in an error queue. The **err** bit indicates, if a new error has occurred. It can be queried with a status register byte call (STB?). The error queue is queried separately and is automatically cleared when read.

<ERR>	Message	Description
0	"No error"	Error queue empty, no error present
-100	"Command error"	Invalid command used
-101	"Invalid character"	...in the command string
-102	"Syntax error"	
-103	"Invalid separator"	
-108	"Parameter not allowed"	
-109	"Missing parameter"	
-113	"Undefined header"	Command unknown
-120	"Numeric data error"	
-131	"Invalid suffix"	or unit
-141	"Invalid character data"	
-151	"Invalid string data"	
-200	"Execution error"	General error, used for various situations
-201-	"Invalid while in local"	Device is local, so remote mode is blocked
-203	„Command protected“	Access to sequence control denied or Feature not enabled (R mode) Access to functions parameters denied
-223	"Too much data"	
-224	"Illegal parameter value"	
-225	"Out of memory"	
-240	"Hardware error"	max. CAN nodes exceeded / CAN node unknown/ no gateway
-241	„Hardware missing“	
-220	"Parameter error"	Object not defined
-221	"Settings conflict"	Read-write law violated, no access Access to menu only when OUTPUT OFF Access to set values denied (device is in slave mode)
-222	"Data out of range"	Values exceeds upper or lower limit
-223	"Too much data"	Object length not correct
-232	"Invalid format"	Time format wrong
-350	"Queue overflow"	Error and event buffer overflow
-360	"Communication error"	Various communication errors (also see section 3.7 of the external guide „ Programming “): Framing error Checksum not correct etc.
-361	"Parity error in program message"	at RS232: Parity error
-362	"Framing error in program message"	
-363	"Input buffer overrun"	
-365	"Time out error"	

Further error messages result from device depending alarms, warnings and notifications (see user manual of your device for detailed information and the user manual of the interface card too, for the message):

<ERR>	Message	Description
100 - 199	see alarm table in 3.9 of the external guide „ Programming “	displayed only, the ones and tens columns of the error code are identical to the error code in the table above
200 - 299	see alarm table in 3.9 of the external guide „ Programming “	warnings, the ones and tens columns of the error code are identical to the error code in the table above
300 - 399	see alarm table in 3.9 of the external guide „ Programming “	alarms, the ones and tens columns of the error code are identical to the error code in the table above

5. Appendix

5.1 SCPI command overview

Main	1.Sub	2.Sub	3.Sub	Value(s)	PS8000T	PS8000DT/2U/3U	PS18000T/DT/2U/3U	PS18000R	PS19000	EL3000/EL9000	Remark
CURR				0...I _{max}	*	*	*	*	*	*	
CURR?				Set value of current	*	*	*	*	*	*	
CURR: LEV				0...I _{max}	*	*	*	*	*	*	At EL: Level A or B, depending on what is set
CURR: LEV?				Set value of current	*	*	*	*	*	*	At EL: Level A or B, depending on what is set
CURR: HIGH				CURR:LOW...I _{max}					*	*	At EL: AB level
CURR: HIGH?				0...I _{max}					*	*	At EL: AB level
CURR: LOW				0...CURR:HIGH					*	*	At EL: AB level
CURR: LOW?				0...I _{max}					*	*	At EL: AB level
ERR: ALL?				Up to 3 error strings	*	*	*	*	*	*	
ERR: NEXT?				1 error string	*	*	*	*	*	*	
INP				1, 0, ON, OFF					*	*	
INP?				ON, OFF					*	*	
INP: STAT				1, 0, ON, OFF					*	*	
INP: STAT?				ON, OFF					*	*	
LOCK				1, 0, ON, OFF	*	*	*	*	*	*	
LOCK?					*	*	*	*	*	*	
LOCK: STAT				1, 0, ON, OFF	*	*	*	*	*	*	
LOCK: STAT?					*	*	*	*	*	*	
LOCK: OWN?				REM, LOC, NONE	*	*	*	*	*	*	
MEAS: VOLT?				1 value	*	*	*	*	*	*	
MEAS: CURR?				1 value	*	*	*	*	*	*	
MEAS: POW?				1 value	*	*	*	*	*	*	
MEAS: ARR?				3 values	*	*	*	*	*	*	
MEAS: VOLT: DC?				1 value	*	*	*	*	*	*	
MEAS: CURR: DC?				1 value	*	*	*	*	*	*	
MEAS: POW: DC?				1 value	*	*	*	*	*	*	
MEAS: SCAL: VOLT?				1 value	*	*	*	*	*	*	
MEAS: SCAL: CURR?				1 value	*	*	*	*	*	*	
MEAS: SCAL: POW?				1 value	*	*	*	*	*	*	
MEAS: SCAL: ARR?				3 values	*	*	*	*	*	*	
MEAS: SCAL: VOLT: DC?				1 value	*	*	*	*	*	*	
MEAS: SCAL: CURR: DC?				1 value	*	*	*	*	*	*	
MEAS: SCAL: POW: DC?				1 value	*	*	*	*	*	*	
OUTP				1, 0, ON, OFF	*	*	*	*	*	*	
OUTP?				ON, OFF	*	*	*	*	*	*	
OUTP: STAT				1, 0, ON, OFF	*	*	*	*	*	*	
OUTP: STAT?				ON, OFF	*	*	*	*	*	*	
POW				0...P _{max}	*	*	*	*	*	*	
POW?				Set value of power	*	*	*	*	*	*	
POW: LEV				0...P _{max}	*	*	*	*	*	*	
POW: LEV?				Set value of power	*	*	*	*	*	*	
POW: HIGH				POW:LOW...P _{max}					*	*	
POW: HIGH?				0...P _{max}					*	*	
POW: LOW				0...POW:HIGH					*	*	
POW: LOW?				0...P _{max}					*	*	
PULS: WIDT: LOW				50us...100s					*	*	At EL: AB level
PULS: WIDT: LOW?				50us...100s					*	*	At EL: AB level
PULS: WIDT: HIGH				50us...100s					*	*	At EL: AB level
PULS: WIDT: HIGH?				50us...100s					*	*	At EL: AB level
PULS: TRAN				30us...200ms					*	*	At EL: AB level
PULS: TRAN?				30us...200ms					*	*	At EL: AB level
PULS: TRAN: LEAD				30us...200ms					*	*	At EL: AB level
PULS: TRAN: LEAD?				30us...200ms					*	*	At EL: AB level
RES				0...R _{max}		*		*	*	*	At EL: Level A or B, depending on what is set
RES?				Set value of resistance		*		*	*	*	At EL: Level A or B, depending on what is set
RES: LEV				0...R _{max}		*		*	*	*	At EL: Level A or B, depending on what is set
RES: LEV?				Set value of resistance		*		*	*	*	At EL: Level A or B, depending on what is set
RES: HIGH				RES:LOW...R _{max}					*	*	
RES: HIGH?				0...R _{max}					*	*	
RES: LOW				0...RES:HIGH					*	*	
RES: LOW?				0...R _{max}					*	*	
SOUR: VOLT				0...U _{max}	*	*	*	*	*	*	
SOUR: VOLT?				Set value of value	*	*	*	*	*	*	
SOUR: VOLT: LEV				0...U _{max}	*	*	*	*	*	*	
SOUR: VOLT: LEV?				Set value of value	*	*	*	*	*	*	
SOUR: VOLT: HIGH				VOLT:LOW...U _{max}					*	*	
SOUR: VOLT: HIGH?				0...U _{max}					*	*	
SOUR: VOLT: LOW				0...VOLT:HIGH					*	*	
SOUR: VOLT: LOW?				0...U _{max}					*	*	

Main	1.Sub	2.Sub	3.Sub	Value(s)	PS8000T	PS8000DT/2U/3U	PS18000T/DT/2U/3U	PS18000R	PS19000	EL3000/EL9000	Remark
SOUR:	CURR			dito	*	*	*	*	*	*	
SOUR:	CURR?			dito	*	*	*	*	*	*	
SOUR:	CURR:	LEV		dito	*	*	*	*	*	*	
SOUR:	CURR:	LEV?		dito	*	*	*	*	*	*	
SOUR:	CURR:	HIGH		dito					*	*	
SOUR:	CURR:	HIGH?		dito					*	*	
SOUR:	CURR:	LOW		dito					*	*	
SOUR:	CURR:	LOW?		dito					*	*	
SOUR:	POW			dito	*	*	*	*	*	*	
SOUR:	POW?			dito	*	*	*	*	*	*	
SOUR:	POW:	LEV		dito	*	*	*	*	*	*	
SOUR:	POW:	LEV?		dito	*	*	*	*	*	*	
SOUR:	POW:	HIGH		dito					*	*	
SOUR:	POW:	HIGH?		dito					*	*	
SOUR:	POW:	LOW		dito					*	*	
SOUR:	POW:	LOW?		dito					*	*	
SOUR:	RES			dito		*		*	*	*	
SOUR:	RES?			dito		*		*	*	*	
SOUR:	RES:	LEV		dito		*		*	*	*	
SOUR:	RES:	LEV?		dito		*		*	*	*	
SOUR:	RES:	HIGH		dito					*	*	
SOUR:	RES:	HIGH?		dito					*	*	
SOUR:	RES:	LOW		dito					*	*	
SOUR:	RES:	LOW?		dito					*	*	
SOUR:	VOLT:	PROT		0...110% Umax	*	*	*	*	*	*	OVP
SOUR:	VOLT:	PROT?		0...110% Umax	*	*	*	*	*	*	OVP
SOUR:	VOLT:	PROT:	LEV	0...110% Umax	*	*	*	*	*	*	OVP
SOUR:	VOLT:	PROT:	LEV?	0...110% Umax	*	*	*	*	*	*	OVP
SOUR:	PULS:	WIDT:	LOW	dito					*	*	
SOUR:	PULS:	WIDT:	LOW?	dito					*	*	
SOUR:	PULS:	WIDT:	HIGH	dito					*	*	
SOUR:	PULS:	WIDT:	HIGH?	dito					*	*	
SOUR:	PULS:	TRAN		dito					*	*	
SOUR:	PULS:	TRAN?		dito					*	*	
SOUR:	PULS:	TRAN:	LEAD	dito					*	*	
SOUR:	PULS:	TRAN:	LEAD?	dito					*	*	
STAT:	OPER?				*	*	*	*	*	*	
STAT:	OPER:	EVENT?		0-32767	*	*	*	*	*	*	
STAT:	OPER:	COND?		0-32767	*	*	*	*	*	*	
STAT:	OPER:	ENAB		0-32767	*	*	*	*	*	*	
STAT:	OPER:	ENAB?		0-32767	*	*	*	*	*	*	
STAT:	OPER:	PTR		0-32767	*	*	*	*	*	*	
STAT:	OPER:	PTR?		0-32767	*	*	*	*	*	*	
STAT:	OPER:	NTR		0-32767	*	*	*	*	*	*	
STAT:	OPER:	NTR?		0-32767	*	*	*	*	*	*	
STAT:	QUES?			0-32767	*	*	*	*	*	*	
STAT:	QUES:	EVENT?		0-32767	*	*	*	*	*	*	
STAT:	QUES:	COND?		0-32767	*	*	*	*	*	*	
STAT:	QUES:	ENAB		0-32767	*	*	*	*	*	*	
STAT:	QUES:	ENAB?		0-32767	*	*	*	*	*	*	
STAT:	QUES:	PTR		0-32767	*	*	*	*	*	*	
STAT:	QUES:	PTR?		0-32767	*	*	*	*	*	*	
STAT:	QUES:	NTR		0-32767	*	*	*	*	*	*	
STAT:	QUES:	NTR?		0-32767	*	*	*	*	*	*	
SYST:	COMM:	NET:	IPAD	0, 0, 0, 0 - 255, 255, 255, 255	*	*	*	*	*	*	Only with IF-E1b
SYST:	COMM:	NET:	IPAD?	0.0.0.0 - 255.255.255.255	*	*	*	*	*	*	Only with IF-E1b
SYST:	COMM:	NET:	GATE	0, 0, 0, 0 - 255, 255, 255, 255	*	*	*	*	*	*	Only with IF-E1b
SYST:	COMM:	NET:	GATE?	0.0.0.0 - 255.255.255.255	*	*	*	*	*	*	Only with IF-E1b
SYST:	COMM:	NET:	MAC?		*	*	*	*	*	*	Only with IF-E1b
SYST:	COMM:	NET:	MASK	0, 0, 0, 0 - 255, 255, 255, 255	*	*	*	*	*	*	Only with IF-E1b
SYST:	COMM:	NET:	MASK?	0.0.0.0 - 255.255.255.255	*	*	*	*	*	*	Only with IF-E1b
SYST:	DATA:	SET		Object telegram as ASCII	*	*	*	*	*	*	Only with IF-Ex: Set something
SYST:	DATA:	REQ		Object telegram as ASCII	*	*	*	*	*	*	Only with IF-Ex: Request something
SYST:	ERR:	ALL?		dito	*	*	*	*	*	*	
SYST:	ERR:	NEXT?		dito	*	*	*	*	*	*	
SYST:	LOCK			dito	*	*	*	*	*	*	
SYST:	LOCK:	STAT		dito	*	*	*	*	*	*	
SYST:	LOCK:	OWN?		dito	*	*	*	*	*	*	
SYST:	VERS?			SCPI version (1999.0)	*	*	*	*	*	*	

Main	1.Sub	2.Sub	3.Sub	Value(s)	PS8000T	PS8000DT/2U/3U	PS18000T/DT/2U/3U	PS1800R	PS19000	EL3000/EL9000	Remark
VOLT				dito	*	*	*	*	*	*	
VOLT?				dito	*	*	*	*	*	*	
VOLT: LEV				dito	*	*	*	*	*	*	
VOLT: LEV?				dito	*	*	*	*	*	*	
VOLT: HIGH				dito					*		
VOLT: HIGH?				dito					*		
VOLT: LOW				dito					*		
VOLT: LOW?				dito					*		
VOLT: PROT				dito	*	*	*	*	*		OVP
VOLT: PROT?				dito	*	*	*	*	*		OVP
VOLT: PROT: LEV				dito	*	*	*	*	*		OVP
VOLT: PROT: LEV?				dito	*	*	*	*	*		OVP
*RST					*	*	*	*	*	*	
*IDN?				String, max. 128 characters	*	*	*	*	*	*	
*STB?				0...255	*	*	*	*	*	*	
*ESR?				0...255	*	*	*	*	*	*	
*ESE				0...255	*	*	*	*	*	*	
*ESE?				0...255	*	*	*	*	*	*	
*CLS					*	*	*	*	*	*	
*TRG					*	*	*	*	*	*	
*SRE				0...255	*	*	*	*	*	*	
*SRE?				0...255	*	*	*	*	*	*	

	requires the device to be already in remote control mode
	only available with Ethernet card IF-E1 or IF-E2, some commands require remote control mode
	only available with GPIB card IF-G1
	only for specific interface cards



Elektro-Automatik

EA-Elektro-Automatik GmbH & Co. KG

Entwicklung - Produktion - Vertrieb

Development - Production - Sales

Helmholtzstraße 31-33

41747 Viersen

Germany

Fon: 02162 / 37 85-0

Fax: 02162 / 16 230

ea1974@elektroautomatik.de

www.elektroautomatik.de