

SCPI-Befehlsreferenz und Beispiele (mit TestController-Skripten)

Versionshistorie

Version	Datum	Beschreibung
v1.0	2026-05-02	Erstveröffentlichung
v1.1	2026-09-02	Gefährliche Befehle, Fehlercodetabelle, Befehlsindex und Befehlshierarchie hinzugefügt

Kommunikationskonfiguration

- **Baudrate:** 115200
- **Datenbits:** 8
- **Stoppbits:** 1
- **Parität:** Keine
- **Zeilenende:** Geräteabhängig (üblicherweise `\r\n` oder `\n`)
- **Treiber:** SCPIx (TC)
- **Hardware:** CH340C (USB-zu-Seriell)

TestController-Konfigurationsbeispiel:

```
#port comFixedBaud
#baudrate 115200
#driver SCPIx
```

Zusammenfassung gefährlicher Befehle

Die folgenden Befehle **schreiben Flash, führen Reset/Neustart aus, löschen Kalibrierparameter oder ändern Konfigurationsprogramme**. Vor der Ausführung stets den Gerätezustand prüfen und vollständige Backups anlegen.

Befehl	Risiko	Konsequenz
<code>*RST</code>	Hoch	Standard-Reset; springt zum Bootloader
<code>*FWUPD</code> / <code>*FWUPDRST</code>	Sehr hoch	Löscht Produktschlüssel, schreibt Flash, springt zum Bootloader
<code>PACKet:RATIO</code>	Hoch	Parst 14 Kalibrierwerte, schreibt in Kalibrier-Flash
<code>:SYSTem:BACKup:DATA</code>	Hoch	Sichert Datenbereich in Backup-Bereich, schreibt Flash
<code>:SYSTem:RESEt:PROGram</code>	Sehr hoch	Setzt Programme 0–31 auf Werkseinstellungen zurück, springt dann zum Bootloader

Befehl	Risiko	Konsequenz
<code>:SYSTem:RESEt:DATA</code>	Sehr hoch	Stellt Daten aus Backup wieder her, schreibt Flash, springt zum Bootloader
<code>:SYSTem:RESEt:CALibration</code>	Sehr hoch	Stellt Kalibrierparameter aus Backup wieder her, schreibt Flash, springt zum Bootloader
<code>CONTRo1:PROGram:SAVE</code>	Mittel-hoch	Schreibt Programmparameter in Programmspeicherbereich

Hinweis: Die folgenden GSV-Konfigurationsbefehle schreiben ebenfalls Flash — mit Vorsicht verwenden:

`CONFIg:FANS:ENABle`, `CONFIg:BEEPer:KEY:ENABle`, `CONFIg:BEEPer:PROCEsscompletion:ENABle`,
`CONFIg:SCREenflash:ENABle`, `CONFIg:LISPfn:ENABle`, `CONFIg:LDSPfn:ENABle`,
`CONFIg:AUTOstart:ENABle`, `CONFIg:CALibration:ENABle`, `CONFIg:TESTir:ENABle`,
`CONFIg:ACTivatebattery:ENABle`, `CONFIg:DEFAUltprogram`

Inhaltsverzeichnis

- [1. Allgemeine Konventionen](#)
- [2. Allgemeine Befehle](#)
- [3. Messungen und Statusabfragen](#)
- [4. Programmeinstellungen \(PC-Steuerung\)](#)
- [5. Programmsteuerungsablauf](#)
- [6. Supersteuerung](#)
- [7. GSV-Globalkonfiguration](#)
- [8. Kalibrierung, Backup und System-Reset](#)
- [9. Typischer vollständiger Arbeitsablauf](#)
- [10. Fehlercodetabelle](#)

Befehlshierarchie-Übersicht

*IDN?	*RST	*FWUPD	*FWUPDRST	*UPT?
— MEASure				
— ALL?				
— VOLTage?				
— CURRent?				
— CAPAcity?				
— ENERgy?				
— IR?				
— STATus				
— ?				
— CHANnel?				
— LED?				
— CHECK				
— STATus:LED?				

- | |— TIME:STAGE?
- | |— TIME:TOTAL?
- | |— CYCLE:COUNT?
- |
- |— PROGram
 - | |— SERIalNo
 - | |— TYPE
 - | |— TASK
 - | |— CYCLE:MODE
 - | |— CYCLE:COUNTs
 - | |— CAPacity
 - | |— CURRENT:PRECharge / CHARge / TERMination / DISCharge / DISCharge:REDUce
 - | |— VOLTage:TERMination / TARGet / PRECharge to fastcharge / REStart / CUT
 - | |— TIME:PRECharge / CHARge / CUT / RESting:CHARge / RESting:DISCharge
 - | |— DELTaPeak
 - | |— LONGname
 - | |— SHORtname
 - | |— SETUp
- |
- |— CONTRol
 - | |— PROGram:PREProcess / SAVE / COMPleted / RUNNing
 - | |— TRANSferChannelControlToPC
 - | |— PROCess:STARt
 - | |— RELEase
 - | |— RESting
 - | |— ACTIvatebattery:STARt / EXIT
- |
- |— SUPER
 - | |— CONTRol:RUN / STOP
 - | |— PROGram:SETUp / TYPE / TASK
 - | |— PROGram:CURRENT:PRECharge / CHARge / TERMination / DISCharge / DISCharge:REDUce
 - | |— PROGram:VOLTage:TERMination / TARGet / PRECharge to fastcharge / CUT (kein REStart)
 - | |— PROGram:DELTApeak
 - | |— CONTRol:ACTIvatebattery:STARt / EXIT
- |
- |— CONFIg
 - | |— FANS:ENABle
 - | |— BEEPer
 - | | |— KEY:ENABle
 - | | | |— PROCesscompletion:ENABle
 - | |— SCREenflash:ENABle
 - | |— LISPfn:ENABle / LDSPfn:ENABle
 - | |— AUTostart:ENABle
 - | |— CALIbration:ENABle
 - | |— TESTir:ENABle
 - | |— ACTIvatebattery:ENABle
 - | |— DEFaulTprogram
- |
- |— PACKet
 - | |— RATIo
 - | |— RATIo:RUN?
 - | |— RATIo:BACKup?
- |
- |— SYSTem
 - | |— BACKup:DATA
 - | |— RESEt:PROGram
 - | |— RESEt:DATA

Befehlsindextabelle

Befehl	Typ	Abschnitt
*IDN?	Abfrage	2.1
*RST	Aktion	2.2
*FWUPD / *FWUPDRST	Aktion	2.3
*UPT?	Abfrage	2.4
STATUs? / STATUs:CHANne1?	Abfrage	3.2.1
STATUs:LED? / CHECK:STATUs:LED?	Abfrage	3.2.2
MEASure:ALL?	Abfrage	3.3.1
MEASure:VOLTage?	Abfrage	3.3.2
MEASure:CURREnt?	Abfrage	3.3.3
MEASure:CAPAcity?	Abfrage	3.3.4
MEASure:ENERgy?	Abfrage	3.3.5
MEASure:IR?	Abfrage	3.3.6
CHECK:TIME:STAGE?	Abfrage	3.3.7
CHECK:TIME:TOTA1?	Abfrage	3.3.8
CHECK:CYCLE:COUNT?	Abfrage	3.3.9
PROGrama:SERIa1no	Einstellung / Abfrage	4.1.1
PROGrama:TYPE	Einstellung / Abfrage	4.1.2
PROGrama:TASK	Einstellung / Abfrage	4.1.3
PROGrama:CYCLE:MODE	Einstellung / Abfrage	4.1.4
PROGrama:CYCLE:COUNTs	Einstellung / Abfrage	4.1.5
PROGrama:CAPAcity	Einstellung / Abfrage	4.1.6
PROGrama:CURREnt:PRECharge	Einstellung / Abfrage	4.2.1
PROGrama:CURREnt:CHARGE	Einstellung / Abfrage	4.2.2
PROGrama:CURREnt:TERMination	Einstellung / Abfrage	4.2.3
PROGrama:CURREnt:DISCharge	Einstellung / Abfrage	4.2.4
PROGrama:CURREnt:DISCharge:REDUce	Einstellung / Abfrage	4.2.5
PROGrama:VOLTage:TERMination	Einstellung / Abfrage	4.3.1

Befehl	Typ	Abschnitt
PROGrama:VOLTage:TARGET	Einstellung / Abfrage	4.3.2
PROGrama:VOLTage:PREChargeToFastCharge	Einstellung / Abfrage	4.3.3
PROGrama:VOLTage:REStart	Einstellung / Abfrage	4.3.4
PROGrama:VOLTage:CUT	Einstellung / Abfrage	4.3.5
PROGrama:TIME:PRECharge	Einstellung / Abfrage	4.4.1
PROGrama:TIME:CHARGE	Einstellung / Abfrage	4.4.2
PROGrama:TIME:CUT	Einstellung / Abfrage	4.4.3
PROGrama:TIME:RESting:CHARGE	Einstellung / Abfrage	4.4.4
PROGrama:TIME:RESting:DISCharge	Einstellung / Abfrage	4.4.5
PROGrama:DELTApeak	Einstellung / Abfrage	4.5.1
PROGrama:LONGname	Einstellung / Abfrage	4.5.2
PROGrama:SHORTname	Einstellung / Abfrage	4.5.3
CONTRol:PROGrama:PREProcess	Aktion / Abfrage	5.1
CONTRol:PROGrama:SAVE	Aktion	5.2
CONTRol:PROGrama:COMPLeted	Aktion	5.3
PROGrama:SETUp	Aktion	5.4
CONTRol:PROGrama:RUNning	Aktion	5.5
CONTRol:TRANSferChannelControlToPC	Aktion / Abfrage	5.6
CONTRol:PROCEss:START	Aktion	5.7
STATUs:CONTRol:PROCEss:COMPLeted?	Abfrage	5.8
CONTRol:RELEase	Aktion	5.9
CONTRol:RESting	Einstellung	5.10
CONTRol:ACTivatebattery:START / EXIT	Aktion	5.11
SUPER:CONTRol:RUN / STOP	Aktion	6.1
SUPER:PROGrama:SETUp	Einstellung	6.2
SUPER:PROGrama:TYPE	Einstellung	6.3
SUPER:PROGrama:TASK	Einstellung	6.4
SUPER:PROGrama:CURRENT:*	Einstellung	6.5
SUPER:PROGrama:VOLTage:*	Einstellung	6.6
SUPER:PROGrama:DELTApeak	Einstellung	6.7

Befehl	Typ	Abschnitt
<code>SUPER:CONTROL:ACTivatebattery:START / EXIT</code>	Aktion	6.8
<code>CONFig:FANS:ENABle</code>	Einstellung / Abfrage	7.1
<code>CONFig:BEEPer:KEY:ENABle</code>	Einstellung / Abfrage	7.2
<code>CONFig:BEEPer:PROcesscompletion:ENABle</code>	Einstellung / Abfrage	7.3
<code>CONFig:SCREenflash:ENABle</code>	Einstellung / Abfrage	7.4
<code>CONFig:LISPfn:ENABle</code>	Einstellung / Abfrage	7.5
<code>CONFig:LDSPfn:ENABle</code>	Einstellung / Abfrage	7.6
<code>CONFig:AUTOstart:ENABle</code>	Einstellung / Abfrage	7.7
<code>CONFig:CALIbration:ENABle</code>	Einstellung / Abfrage	7.8
<code>CONFig:TESTir:ENABle</code>	Einstellung / Abfrage	7.9
<code>CONFig:ACTivatebattery:ENABle</code>	Einstellung / Abfrage	7.10
<code>CONFig:DEFAUltprogram</code>	Einstellung / Abfrage	7.11
<code>PACKet:RATIo:RUN?</code>	Abfrage	8.1
<code>PACKet:RATIo:BACKup?</code>	Abfrage	8.2
<code>PACKet:RATIo</code>	Einstellung	8.3
<code>:SYSTem:BACKup:DATA</code>	Aktion	8.4
<code>:SYSTem:RESEt:PROGram</code>	Aktion	8.5
<code>:SYSTem:RESEt:DATA</code>	Aktion	8.6
<code>:SYSTem:RESEt:CALIbration</code>	Aktion	8.7

1. Allgemeine Konventionen

1.1 Kanalparameter

Die meisten Befehle benötigen `<ch>`, unterstützt werden:

1, 2, 3, 4
c1, ch1, c2, ch2, c3, ch3, c4, ch4
None, Ignore

In TestController wird der Kanal über die Variable `selectedChannel1` ausgewählt, die vor dem Senden durch `ch1`, `ch2` usw. ersetzt wird.

Kanalvariablendefinition in TC:

```
#scpiCmd channel none
:setvar: selectedChannel=inputValue

#scpiCmd channel? none?
:readmath: selectedChannel
```

1.2 Boolesche Werte

Wahr: T, Y, True, Yes, On
Falsch: F, N, False, No, Off

Schaltflächen in TC verwenden **ON** / **OFF**, zum Beispiel:

```
CONFIg:FANS:ENABLe ON
CONFIg:FANS:ENABLe OFF
```

1.3 Einheiten und interne Umrechnung

Typ	Externe Einheit	Interne Einheit	Behandlung in TC
Spannung	V	mV	V senden, Abfrage liefert mV, readmath: value/1000.0
Strom	A	mA	A senden, Abfrage liefert mA, readmath: value/1000.0
Kapazität	Ah / mAh	mAh	Ah senden, Abfrage liefert mAh, durch 1000 teilen
Zeit	min / s	min	Min senden, Abfrage liefert Minuten
Energie	Wh	Wh	Direkt zurückgegeben
Widerstand	mΩ	Rohe Zahl	Direkt zurückgegeben

1.4 Abfrage und Einstellung

- Mit **?**: Abfrage.
- Ohne **?**: Einstellung oder Aktion.
- Fehler: **SCPI_Error** gibt einen dezimalen Fehlercode aus, z. B. **-224**.

1.5 Kurzform und Langform

Großbuchstaben in SCPI-Befehlen bilden die Kurzform; Kleinschreibung kann weggelassen werden. Zum Beispiel:

Langform	Kurzform
MEASure:VOLTage?	MEAS:VOLT?
PROGram:CURRent:CHARGe	PROG:CURR:CHAR
CONTro1:TRANSferChannelContro1ToPC	CONT:TRAN

Groß-/Kleinschreibung ist nicht relevant; `meas:volt?` ist äquivalent zu `MEAS:VOLT?`. Tatsächliche Zeilenenden und Akzeptanz von Abkürzungen sind geräteabhängig.

2. Allgemeine Befehle

2.1 `*IDN?`

- **Typ:** Abfrage
- **Syntax:** `*IDN?`
- **Rückgabe:** `Hersteller,Modell,Seriennummer,Firmware-Version`
- **Beispiel:**

```
Senden:  *IDN?  
Antwort: ENOVA TECHNOLOGY,GYRFALCON S8000,0,3.0.38
```

2.2 `*RST`

- **Typ:** Aktion
- **Syntax:** `*RST`
- **Beschreibung:** Standard-Reset. ⚠ **Gefährlich.**
- **Beispiel:**

```
Senden:  *RST  
Antwort: (keine)
```

2.3 `*FWUPD` / `*FWUPDRST`

- **Typ:** Aktion
- **Syntax:** `*FWUPD` oder `*FWUPDRST`
- **Beschreibung:** Neustart und Start des Firmware-Update-Programms. ⚠ **Gefährlich.**
- **Beispiel:**

```
Senden:  *FWUPD  
Antwort: (keine)
```

2.4 `*UPT?`

- **Typ:** Abfrage
- **Syntax:** `*UPT?`
- **Rückgabe:** `D|HH:MM:SS`
- **TC-Anwendungsbeispiel:**

```
#cmdSetup infoAsk Total_Runtime Global_Settings  
:read: *UPT?  
:tip:Click "Get" to get the duration of continuous operation from startup to the current  
time.
```

- **Beispiel:**

```
Senden: *UPT?
Antwort: 3D|01:09:59
```

3. Messungen und Statusabfragen

3.1 Messabfragekonfiguration (TC-Anwendungsbeispiel)

```
#askValues MEASure:ALL? 1,2,3,4
#askValuesReadFormat ffffffffffffffff
#readingDelay 5
#interfaceType Charger Charger:2 Charger:3 Charger:4
#interface readVoltage 0 4 8 12
#interface readCurrent 1 5 9 13
#interface readCapacity 2 6 10 14
#interface readEnergy 3 7 11 15
```

Hinweise

- `#askValues MEASure:ALL? 1,2,3,4`: sendet periodisch `MEASure:ALL? 1,2,3,4`.
- `#askValuesReadFormat ffffffffffffffff`: alle 16 Rückgabewerte werden als Fließkommazahlen interpretiert.
- `#interface readVoltage 0 4 8 12`: Rückgabewerte an den Indizes 0, 4, 8 und 12 entsprechen der Spannung von CH1-CH4.

3.2 Statusabfragen

3.2.1 STATUS? / STATUS:CHANNEL?

- **Typ:** Abfrage
- **Syntax:** `STATUS? <ch>` oder `STATUS:CHANNEL? <ch>`
- **Rückgabe:** Statuszeichenkette, z. B. `Charging`, `Done -5mV`, `ErrTimeout`, `ErrMaxCap`, `ErrVOLTover`
- **TC-Anwendungsbeispiel:**

```
#cmdSetup infoAsk Status CH1_Super_Controller
:read: Status? Ch1;
:tip:Click "Get" to retrieve the real-time status of the channel 1.
```

- **Beispiel:**

```
Senden: Status? Ch1
Antwort: CC_Charging
```

3.2.2 STATUS:LED? / CHECK:STATUS:LED?

- **Typ:** Abfrage
- **Syntax:** `STATUS:LED? <ch>`
- **Rückgabe:** `OFF`, `WHITE`, `RED`, `BLUE`, `YELLOW`, `PURPLE BLINKING`, `RED BLINKING`, `GREEN`

- **Beispiel:**

```
Senden:  STATus:LED? Ch1
Antwort:  GREEN
```

3.3 Messbefehle

3.3.1 MEASure:ALL?

- **Typ:** Abfrage
- **Syntax:** MEASure:ALL? <ch1>[,<ch2>...]
- **Beschreibung:** Bis zu 4 Kanäle, sortiert und dedupliziert, nur 1–4 bleiben erhalten. Jeder Kanal liefert 4 Werte.

Aufschlüsselung des Rückgabeformats:

```
MEASure:ALL? 1,2,3,4
Rückgabe: 4.183e+00, 3.510e-01, 2.048e-02, 8.540e-02, 4.202e+00, 3.200e-01, 2.003e-02, 8.411e-02, 4.154e+00, ...
```

↑	↑	↑	↑	↑	↑	↑	↑
CH2 Energie	CH1 Spg	CH1 Strom	CH1 Kap	CH1 Energie	CH2 Spg	CH2 Strom	CH2 Kap

CH3 Spg ...

Jeder Kanal liefert immer 4 Werte in folgender Reihenfolge:

Index	Bedeutung	Einheit	Hinweise
1	Spannung	V	Echtzeit-Spannung
2	Strom	A	Positiv beim Laden, negativ beim Entladen, 0 in Ruhe
3	Kapazität	Ah	Positiv beim Laden, negativ beim Entladen
4	Energie	Wh	Positiv beim Laden, negativ beim Entladen

- **TC-Anwendungsbeispiel:**

```
#askValues MEASure:ALL? 1,2,3,4
#askValuesReadFormat ffffffffffffffffff
```

- **Vollständiges Beispiel:**

```
Senden:  MEASure:ALL? 1,2,3,4
Antwort:  4.183e+00,3.510e-01,2.048e-02,8.540e-02,4.202e+00,3.200e-01,2.003e-02,8.411e-02,4.154e+00,3.510e-01,2.046e-02,8.484e-02,3.000e-03,2.000e-03,0.000e+00,0.000e+00
```

3.3.2 MEASure:VOLTage?

- **Typ:** Abfrage
- **Syntax:** MEASure:VOLTage? <ch>
- **Rückgabe:** %5.3fv, z. B. 4.200v.
- **Beispiel:**

```
Senden: MEASure:VOLTage? Ch1
Antwort: 4.189V
```

3.3.3 MEASure:CURRent?

- **Typ:** Abfrage
- **Syntax:** MEASure:CURRent? <ch>
- **Rückgabe:** %5.3fA. Positiv beim Laden, negativ beim Entladen, 0 in Ruhe.
- **Beispiel:**

```
Senden: MEASure:CURRent? Ch1
Antwort: 0.351A
```

3.3.4 MEASure:CAPAcity?

- **Typ:** Abfrage
- **Syntax:** MEASure:CAPAcity? <ch>
- **Rückgabe:** %5.3fAh. Positiv beim Laden, negativ beim Entladen; in Ruhe wird möglicherweise die letzte Kapazität zurückgegeben.
- **Beispiel:**

```
Senden: MEASure:CAPAcity? Ch1
Antwort: 0.031Ah
```

3.3.5 MEASure:ENERgy?

- **Typ:** Abfrage
- **Syntax:** MEASure:ENERgy? <ch>
- **Rückgabe:** %5.3fWh.
- **Beispiel:**

```
Senden: MEASure:ENERgy? Ch1
Antwort: 0.135Wh
```

3.3.6 MEASure:IR?

- **Typ:** Abfrage
- **Syntax:** MEASure:IR? <ch>
- **Rückgabe:** %4u, 4-stellige Breite, rechtsbündig; kurze Werte haben führende Leerzeichen.
- **Beispiel:**

```
Senden: MEASure:IR? Ch1
Antwort: 54
```

3.3.7 CHECK:TIME:STAGE?

- **Typ:** Abfrage
- **Syntax:** CHECK:TIME:STAGE? <ch>
- **Rückgabe:** HH:MM:SS.
- **Beispiel:**

```
Senden:  CHECK:TIME:STAGE? Ch1
Antwort:  00:06:08
```

3.3.8 CHECK:TIME:TOTAL?

- **Typ:** Abfrage
- **Syntax:** CHECK:TIME:TOTAL? <ch>
- **Rückgabe:** D|HH:MM:SS.
- **Beispiel:**

```
Senden:  CHECK:TIME:TOTAL? Ch1
Antwort:  0D|00:06:24
```

3.3.9 CHECK:CYCLE:COUNT?

- **Typ:** Abfrage
- **Syntax:** CHECK:CYCLE:COUNT? <ch>
- **Rückgabe:** %2u, 2-stellige Breite, rechtsbündig; kurze Werte haben führende Leerzeichen.
- **Beispiel:**

```
Senden:  CHECK:CYCLE:COUNT? Ch1
Antwort:   3
```

4. Programmeinstellungen (PC-Steuerung)

Die folgenden Befehle dienen zur Konfiguration von Programmparametern unter PC-Steuerung. Sie werden in TC häufig verwendet. Detaillierte Anwendungsszenarien finden Sie im vollständigen Konfigurationsskript in TC.

4.1 Grundparameter

4.1.1 PROGram:SERIaIno / PROGram:SERIaIno?

- **Typ:** Einstellung / Abfrage
- **Syntax:**
 - Einstellen: PROGram:SERIaIno <ch> <num|T|Temp|Temporary>
 - Abfragen: PROGram:SERIaIno? <ch oder Nummer>
- **Beschreibung:** num = 0–31; T / Temp / Temporary setzt auf 0xFF.
- **TC-Anwendungsbeispiel:**

```
#scpiCmd SerialNo tx PROGRAM:SERIALno (selectedChannel),(value)
#scpiCmd SerialNo? txrx? PROGRAM:SERIALno? (selectedChannel)
:readmath:
getElement("0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28
,29,30,31,temp",listIndex(value,"0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 255"));
```

- **Beispiel:**

```
Senden: PROGRAM:SERIALno Ch1,0
Senden: PROGRAM:SERIALno? Ch1
Antwort: 2 (gibt die dem aktuellen Kanal zugeordnete Programmnummer zurück)
```

4.1.2 PROGRAM:TYPE / PROGRAM:TYPE?

- **Typ:** Einstellung / Abfrage
- **Syntax:**
 - Einstellen: PROGRAM:TYPE <ch> <type>
 - Abfragen: PROGRAM:TYPE? <ch>
- **Typzuordnung:**

```
Ni / NiMh
Li / LiIon / Lithumion
LF / LiFePO4
Li430 / Liio430
Li435 / Liio435
LD / Li150 / Liio150
```

- **TC-Anwendungsbeispiel:**

```
#scpiCmd Type tx PROGRAM:TYPE (selectedChannel),(value)
#scpiCmd Type? txrx? PROGRAM:TYPE? (selectedChannel)
:readmath: getElement("Ni,Li,LF,Li430,Li435,LD",value-1);
```

- **Beispiel:**

```
Senden: PROGRAM:TYPE Ch1,Li
Senden: PROGRAM:TYPE? Ch1
Antwort: 2 (zugeordnet zu Li)
```

4.1.3 PROGRAM:TASK / PROGRAM:TASK?

- **Typ:** Einstellung / Abfrage
- **Syntax:**
 - Einstellen: PROGRAM:TASK <ch> <mode>
 - Abfragen: PROGRAM:TASK? <ch>
- **Modi:**

C / Charge
D / Discharge
Cy / Cycle
Br / Breakin
Re / Refresh
St / Storage

- **TC-Anwendungsbeispiel:**

```
#scpiCmd Task tx PROGRAM:TASK (selectedChannel),(value)
#scpiCmd Task? txrx? PROGRAM:TASK? (selectedChannel)
:readmath: getElement("C,D,Cy,Br,Re,St",value-1);
```

- **Beispiel:**

```
Senden: PROGRAM:TASK Ch1,C
Senden: PROGRAM:TASK? Ch1
Antwort: 1 (zugeordnet zu C)
```

4.1.4 PROGRAM:CYCLE:MODE / PROGRAM:CYCLE:MODE?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGRAM:CYCLE:MODE <ch> <mode>
- **Modi:** CDC / C>D>C, DCD / D>C>D, DC / D>C, CD / C>D
- **TC-Anwendungsbeispiel:**

```
#scpiCmd CycleMode tx PROGRAM:CYCLE:MODE (selectedChannel),(value)
#scpiCmd CycleMode? txrx? PROGRAM:CYCLE:MODE? (selectedChannel)
:readmath: getElement("CDC,DCD,DC,CD",value-1);
```

- **Beispiel:**

```
Senden: PROGRAM:CYCLE:MODE Ch1,CDC
Senden: PROGRAM:CYCLE:MODE? Ch1
Antwort: 1 (zugeordnet zu CDC)
```

4.1.5 PROGRAM:CYCLE:COUNTS / PROGRAM:CYCLE:COUNTS?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGRAM:CYCLE:COUNTS <ch> <n>, n <= 10
- **TC-Anwendungsbeispiel:**

```
#scpiCmd CycleCounts tx PROGRAM:CYCLE:COUNTS (selectedChannel),(value)
#scpiCmd CycleCounts? txrx? PROGRAM:CYCLE:COUNTS? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGRAM:CYCLE:COUNTS Ch1,3
Senden: PROGRAM:CYCLE:COUNTS? Ch1
Antwort: 3
```

4.1.6 PROGram:CAPAcity / PROGram:CAPAcity?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:CAPAcity <ch> <num>[mAh|mWh], < 10000
- **TC-Anwendungsbeispiel:**

```
#scpiCmd MaxCapacity tx PROGram:CAPAcity (selectedChannel),(value)Ah
#scpiCmd MaxCapacity? txrx? PROGram:CAPAcity? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:CAPAcity Ch1,2.0Ah
Senden: PROGram:CAPAcity? Ch1
Antwort: 2000 (zeigt 2.0Ah an)
```

4.2 Stromparameter

4.2.1 PROGram:CURRent:PRECharge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:CURRent:PRECharge <ch> <num>[A]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd PrechargeCurrent tx PROGram:CURRent:PRECharge (selectedChannel),(value)A
#scpiCmd PrechargeCurrent? txrx? PROGram:CURRent:PRECharge? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:CURRent:PRECharge Ch1,0.1A
Senden: PROGram:CURRent:PRECharge? Ch1
Antwort: 100 (zeigt 0.1A an)
```

4.2.2 PROGram:CURRent:CHARge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:CURRent:CHARge <ch> <num>[A]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd ChargeCurrent tx PROGram:CURRent:CHARge (selectedChannel),(value)A
#scpiCmd ChargeCurrent? txrx? PROGram:CURRent:CHARge? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:CURRent:CHARge Ch1,1.0A
Senden: PROGram:CURRent:CHARge? Ch1
Antwort: 1000 (zeigt 1.0A an)
```

4.2.3 PROGram:CURRENT:TERMination / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** `PROGram:CURRENT:TERMination <ch> <num>[A]`
- **TC-Anwendungsbeispiel:**

```
#scpiCmd EndChargeCurrent tx PROGram:CURRENT:TERMination (selectedChannel),(value)A
#scpiCmd EndChargeCurrent? txrx? PROGram:CURRENT:TERMination? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:CURRENT:TERMination Ch1,0.1A
Senden: PROGram:CURRENT:TERMination? Ch1
Antwort: 100
```

4.2.4 PROGram:CURRENT:DISCharge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** `PROGram:CURRENT:DISCharge <ch> <num>[A]`
- **TC-Anwendungsbeispiel:**

```
#scpiCmd DischargeCurrent tx PROGram:CURRENT:DISCharge (selectedChannel),(value)A
#scpiCmd DischargeCurrent? txrx? PROGram:CURRENT:DISCharge? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:CURRENT:DISCharge Ch1,0.5A
Senden: PROGram:CURRENT:DISCharge? Ch1
Antwort: 500
```

4.2.5 PROGram:CURRENT:DISCharge:REDUce / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** `PROGram:CURRENT:DISCharge:REDUce <ch> <num>[A]`
- **TC-Anwendungsbeispiel:**

```
#scpiCmd DischargeReduceCurrent tx PROGram:CURRENT:DISCharge:REDUce (selectedChannel),
(value)A
#scpiCmd DischargeReduceCurrent? txrx? PROGram:CURRENT:DISCharge:REDUce?
(selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:CURRENT:DISCharge:REDUce Ch1,0.1A
Senden: PROGram:CURRENT:DISCharge:REDUce? Ch1
Antwort: 100
```

4.3 Spannungsparameter

4.3.1 PROGram:VOLTage:TERMination / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:VOLTage:TERMination <ch> <num>[V]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd EndChargeVoltage tx PROGram:VOLTage:TERMination (selectedChannel),(value)V
#scpiCmd EndChargeVoltage? txrx? PROGram:VOLTage:TERMination? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:VOLTage:TERMination Ch1,4.2V
Senden: PROGram:VOLTage:TERMination? Ch1
Antwort: 4200
```

4.3.2 PROGram:VOLTage:TARGet / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:VOLTage:TARGet <ch> <num>[V]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd TargetVoltage tx PROGram:VOLTage:TARGet (selectedChannel),(value)V
#scpiCmd TargetVoltage? txrx? PROGram:VOLTage:TARGet? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:VOLTage:TARGet Ch1,1.5V
Senden: PROGram:VOLTage:TARGet? Ch1
Antwort: 1550
```

4.3.3 PROGram:VOLTage:PREChargeToFastcharge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:VOLTage:PREChargeToFastcharge <ch> <num>[V]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd PrechargeToFastcharge tx PROGram:VOLTage:PREChargeToFastcharge
(selectedChannel),(value)V
#scpiCmd PrechargeToFastcharge? txrx? PROGram:VOLTage:PREChargeToFastcharge?
(selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:VOLTage:PREChargeToFastcharge Ch1,3.0V
Senden: PROGram:VOLTage:PREChargeToFastcharge? Ch1
Antwort: 3000
```

4.3.4 PROGram:VOLTage:REStArt / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:VOLTage:REStArt <ch> <num>[V]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd RestartVoltage tx PROGram:VOLTage:REStArt (selectedChannel),(value)V
#scpiCmd RestartVoltage? txrx? PROGram:VOLTage:REStArt? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:VOLTage:REStArt Ch1,3.0V
Senden: PROGram:VOLTage:REStArt? Ch1
Antwort: 3000
```

4.3.5 PROGram:VOLTage:CUT / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:VOLTage:CUT <ch> <num>[V]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd DischargeCutVoltage tx PROGram:VOLTage:CUT (selectedChannel),(value)V
#scpiCmd DischargeCutVoltage? txrx? PROGram:VOLTage:CUT? (selectedChannel)
:readmath: (value/1000.0)
```

- **Beispiel:**

```
Senden: PROGram:VOLTage:CUT Ch1,3.0V
Senden: PROGram:VOLTage:CUT? Ch1
Antwort: 3000
```

4.4 Zeitparameter

4.4.1 PROGram:TIME:PRECharge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:TIME:PRECharge <ch> <num>[s]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd MaxPreChargeTime tx PROGram:TIME:PRECharge (selectedChannel),(value)Min
#scpiCmd MaxPreChargeTime? txrx? PROGram:TIME:PRECharge? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:TIME:PRECharge Ch1,60Min
Senden: PROGram:TIME:PRECharge? Ch1
Antwort: 60
```

4.4.2 PROGram:TIME:CHARge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:TIME:CHARge <ch> <num>[s]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd MaxChargeTime tx PROGram:TIME:CHARge (selectedChannel),(value)Min  
#scpiCmd MaxChargeTime? txrx? PROGram:TIME:CHARge? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:TIME:CHARge Ch1,120Min  
Senden: PROGram:TIME:CHARge? Ch1  
Antwort: 120
```

4.4.3 PROGram:TIME:CUT / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:TIME:CUT <ch> <num>[s]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd MaxDischargeTime tx PROGram:TIME:CUT (selectedChannel),(value)Min  
#scpiCmd MaxDischargeTime? txrx? PROGram:TIME:CUT? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:TIME:CUT Ch1,120Min  
Senden: PROGram:TIME:CUT? Ch1  
Antwort: 120
```

4.4.4 PROGram:TIME:RESTing:CHARge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:TIME:RESTing:CHARge <ch> <num>[s]
- **TC-Anwendungsbeispiel:**

```
#scpiCmd ChargeRestingTime tx PROGram:TIME:RESTing:CHARge (selectedChannel),(value)Min  
#scpiCmd ChargeRestingTime? txrx? PROGram:TIME:RESTing:CHARge? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:TIME:RESTing:CHARge Ch1,10Min  
Senden: PROGram:TIME:RESTing:CHARge? Ch1  
Antwort: 15
```

4.4.5 PROGram:TIME:RESTing:DISCharge / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** `PROGram:TIME:RESTing:DISCharge <ch> <num>[s]`
- **TC-Anwendungsbeispiel:**

```
#scpiCmd DischargeRestingTime tx PROGram:TIME:RESTing:DISCharge (selectedChannel),  
(value)Min  
#scpiCmd DischargeRestingTime? txrx? PROGram:TIME:RESTing:DISCharge? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:TIME:RESTing:DISCharge Ch1,10Min  
Senden: PROGram:TIME:RESTing:DISCharge? Ch1  
Antwort: 15
```

4.5 Namen und Sonstiges

4.5.1 PROGram:DELTApeak / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** `PROGram:DELTApeak <ch> <Advice|num[mV|V]>`
- **TC-Anwendungsbeispiel:**

```
#scpiCmd DeltaPeak tx PROGram:DELTApeak (selectedChannel),(value)mV  
#scpiCmd DeltaPeak? txrx? PROGram:DELTApeak? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:DELTApeak Ch1,5mV  
Senden: PROGram:DELTApeak? Ch1  
Antwort: 5
```

4.5.2 PROGram:LONGname / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** `PROGram:LONGname <ch> <string>`, maximal 16 Zeichen (ohne Terminator).
- **TC-Anwendungsbeispiel:**

```
#scpiCmd LongName tx PROGram:LONGname (selectedChannel),"(value)"  
#scpiCmd LongName? txrx? PROGram:LONGname? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:LONGname Ch1,"MyCell"  
Senden: PROGram:LONGname? Ch1  
Antwort: CHLI↑1.00A␣4.20V
```

Hinweis: Sonderzeichen in der Rückgabe (↑ ↓ ␣ usw.) können in manchen Terminals als  erscheinen. Beim Schreiben `replaceCharsR()` verwenden, um sie wiederherzustellen.

4.5.3 PROGram:SHORTname / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** PROGram:SHORTname <ch> <string>, maximal 10 Zeichen (ohne Terminator).
- **TC-Anwendungsbeispiel:**

```
#scpiCmd ShortName tx PROGram:SHORTname (selectedChannel),"(value)"  
#scpiCmd ShortName? txrx? PROGram:SHORTname? (selectedChannel)
```

- **Beispiel:**

```
Senden: PROGram:SHORTname Ch1,"MC"  
Senden: PROGram:SHORTname? Ch1  
Antwort: CHLI↑1.00A
```

5. Programmsteuerungsablauf

5.1 CONTro1:PROGram:PREProcess / ?

- **Typ:** Aktion / Abfrage
- **Syntax:**
 - CONTro1:PROGram:PREProcess <ch>
 - CONTro1:PROGram:PREProcess? <ch>
- **Rückgabe:** Ready oder Not ready
- **TC-Anwendungsbeispiel:**

```
#scpiCmd Preprocess tx CONTro1:PROGram:PREProcess (selectedChannel)  
#scpiCmd Preprocess? txrx? CONTro1:PROGram:PREProcess? (selectedChannel)
```

- **Beispiel:**

```
Senden: CONTro1:PROGram:PREProcess Ch1  
Senden: CONTro1:PROGram:PREProcess? Ch1  
Antwort: Not ready
```

5.2 CONTro1:PROGram:SAVE

- **Typ:** Aktion
- **Syntax:** CONTro1:PROGram:SAVE <ch>
- **Voraussetzung:** Vorverarbeitung Ready. Ob in Flash oder in temporäre Parameter geschrieben wird, bestimmt SETUP_SERIAL_NO, das über PROGRAM:SERIALno gesetzt wird.
- **TC-Anwendungsbeispiel:**

```
#scpiCmd SaveCurrentProgram tx CONTro1:PROGram:SAVE (selectedChannel)
```

- **Beispiel:**

Senden: `CONTROL:PROGRAM:SAVE ch1`
Antwort: (keine)

5.3 CONTROL:PROGRAM:COMPLETED

- **Typ:** Aktion
- **Syntax:** `CONTROL:PROGRAM:COMPLETED <ch>`
- **TC-Anwendungsbeispiel:**

```
#scpiCmd ExitProgramConfiguration tx CONTROL:PROGRAM:COMPLETED (selectedChannel)
```

- **Beispiel:**

Senden: `CONTROL:PROGRAM:COMPLETED ch1`
Antwort: (keine)

5.4 PROGRAM:SETUP

- **Typ:** Aktion
- **Syntax:** `PROGRAM:SETUP <ch> <program 0~31>`
- **TC-Anwendungsbeispiel:**

```
#scpiCmd SelectAProgramToConfigure tx PROGRAM:SETUP (selectedChannel),(value)
```

- **Beispiel:**

Senden: `PROGRAM:SETUP ch1,0`
Antwort: (keine)

5.5 CONTROL:PROGRAM:RUNNING

- **Typ:** Aktion
- **Syntax:** `CONTROL:PROGRAM:RUNNING <ch> <program 0~31>`
- **Beispiel:**

Senden: `CONTROL:PROGRAM:RUNNING ch1,0`
Antwort: (keine)

5.6 CONTROL:TRANSFERCHANNELCONTROLTOPC / ?

- **Typ:** Aktion / Abfrage
- **Syntax:**
 - `CONTROL:TRANSFERCHANNELCONTROLTOPC <ch> <bool>`
 - `CONTROL:TRANSFERCHANNELCONTROLTOPC? <ch>`
- **Rückgabe:** `Transferred` oder `Not transferred`
- **TC-Anwendungsbeispiel:**

```
CONTROL:TRANSFERChannelControlToPC (channel),(runNow);[200];  
# if (deviceRead(device,command_TRANSFERChannelControlToPCQ) == "Transferred")
```

- **Beispiel:**

```
Senden: CONTROL:TRANSFERChannelControlToPC Ch1,Yes  
Senden: CONTROL:TRANSFERChannelControlToPC? Ch1  
Antwort: Transferred
```

5.7 CONTROL:PROCESS:START

- **Typ:** Aktion
- **Syntax:** CONTROL:PROCESS:START <ch>
- **TC-Anwendungsbeispiel:**

```
CONTROL:PROCESS:START (channel);[50];
```

- **Beispiel:**

```
Senden: CONTROL:PROCESS:START Ch1  
Antwort: (keine)
```

5.8 STATUS:CONTROL:PROCESS:COMPLETED?

- **Typ:** Abfrage
- **Syntax:** STATUS:CONTROL:PROCESS:COMPLETED? <ch>
- **Rückgabe:** Completed oder CH_x status: n
- **TC-Anwendungsbeispiel:**

```
# while (deviceRead(device,command_ProcessCompletedQ) != "Completed")
```

- **Beispiel:**

```
Senden: STATUS:CONTROL:PROCESS:COMPLETED? Ch1  
Antwort: Completed
```

5.9 CONTROL:RELEASE

- **Typ:** Aktion
- **Syntax:** CONTROL:RELEASE <ch>
- **TC-Anwendungsbeispiel:**

```
CONTROL:RELEASE (channel);[50];
```

- **Beispiel:**

```
Senden: CONTROL:RELEASE Ch1  
Antwort: (keine)
```

5.10 CONTROL:RESTing

- **Typ:** Einstellung
- **Syntax:** CONTROL:RESTing <ch> <time>[s], ≤1000 Minuten.
- **TC-Anwendungsbeispiel:**

```
CONTROL:RESTing (channel),(restingTime)min;[50];
```

- **Beispiel:**

```
Senden: CONTROL:RESTing Ch1,10min  
Antwort: (keine)
```

5.11 CONTROL:ACTivatebattery:START / EXIT

- **Typ:** Aktion
- **Syntax:** CONTROL:ACTivatebattery:START <ch> / CONTROL:ACTivatebattery:EXIT <ch>
- **Beispiel:**

```
Senden: CONTROL:ACTivatebattery:START Ch1  
Senden: CONTROL:ACTivatebattery:EXIT Ch1  
Antwort: (keine)
```

6. Supersteuerung

6.1 SUPER:CONTROL:RUN / STOP

- **Typ:** Aktion
- **Syntax:** SUPER:CONTROL:RUN <ch> / SUPER:CONTROL:STOP <ch>
- **TC-Anwendungsbeispiel (CH1):**

```
#cmdSetup button Start_Process CH1_Super_Controller  
:write:SUPER:CONTROL:RUN Ch1;[200];  
:update:CH1_Super_Controller.Choose_A_Program_To_Run CH1_Super_Controller.Battery_Type  
CH1_Super_Controller.Task CH1_Super_Controller.Nothing  
:tip:The button start the program.  
  
#cmdSetup button Stop_Process CH1_Super_Controller  
:write:SUPER:CONTROL:STOP Ch1;[200];  
:tip:The button stop the program.
```

- **Beispiel:**

```
Senden: SUPER:CONTROL:RUN Ch1  
Senden: SUPER:CONTROL:STOP Ch1  
Antwort: (keine)
```

6.2 SUPER:PROGRAM:SETUP

- **Typ:** Einstellung
- **Syntax:** `SUPER:PROGRAM:SETUP <ch> <program 0~31>`
- **TC-Anwendungsbeispiel:**

```
#cmdSetup combobox Choose_A_Program_To_Run CH1_Super_Controller
:write: SUPER:PROGRAM:SETUP Ch1,#;[200]
:read: PROGRAM:SERIALno? Ch1 ;
:readmath:
getElement("0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28
,29,30,31,temp",listIndex(value,"0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 255"));
:update: CH1_Super_Controller.Battery_Type CH1_Super_Controller.Task
CH1_Super_Controller.Nothing CH1_Super_Controller.End_Charge_Current
CH1_Super_Controller.End_Charge_Voltage CH1_Super_Controller.Target_Voltage
CH1_Super_Controller.Delta_Peak_Voltage
:tip:Select a program to run. Subsequent options allow real-time adjustments to key
configurations, with limited safety checks by the charger.
:string:
Program_0 0
Program_1 1
...
Temporary temp
```

- **Beispiel:**

```
Senden: SUPER:PROGRAM:SETUP Ch1,0
Antwort: (keine)
```

6.3 SUPER:PROGRAM:TYPE

- **Typ:** Einstellung
- **Syntax:** `SUPER:PROGRAM:TYPE <ch> <type>`
- **TC-Anwendungsbeispiel:**

```
#cmdSetup combobox Battery_Type CH1_Super_Controller
:write:SUPE:PROGRAM:TYPE ch1, #;[200];
:read: PROGRAM:TYPE? ch1;
:readmath: getElement("Ni,Li,LF,Li430,Li435,LD",value-1);
:update: CH1_Super_Controller.Task CH1_Super_Controller.Choose_A_Program_To_Run
:tip:Modifying this configuration is not recommended.
:string:
NiMh/Cd Ni
LiionStd Li
LiFePO4 LF
Liion430 Li430
Liion435 Li435
Liion150 LD
```

Hinweis: Das `SUPE:` im TC-Beispiel ist wörtlich aus der Konfigurationsdatei übernommen; es ist äquivalent zu `SUPER:` (der SCPI-Parser akzeptiert Kurzformen).

- **Beispiel:**

```
Senden:  SUPER:PROGrama:TYPE  Ch1,Li
Antwort: (keine)
```

6.4 SUPER:PROGrama:TASK

- **Typ:** Einstellung
- **Syntax:** SUPER:PROGrama:TASK <ch> <mode>
- **TC-Anwendungsbeispiel:**

```
#cmdSetup combobox Task CH1_Super_Controller
:write:SUPE:PROGrama:TASK ch1,#; [200];
:read: PROGrama:TASK?  ch1;
:readmath: getElement("C,D,Cy,Br,Re,St",value-1);
:tip:Modifying this configuration is not recommended.
:string:
Charge C
Discharge D
Cycle Cy
Refresh Re
Storage St
:update:CH1_Super_Controller.Nothing CH1_Super_Controller.Choose_A_Program_To_Run
:visible: inList(CH1_Super_Controller.Battery_Type,"Li LF Li430 Li435")
```

- **Beispiel:**

```
Senden:  SUPER:PROGrama:TASK  Ch1,C
Antwort: (keine)
```

6.5 SUPER:PROGrama:CURRENT:*

Enthält:

- SUPER:PROGrama:CURRENT:PRECharge
- SUPER:PROGrama:CURRENT:CHARGE
- SUPER:PROGrama:CURRENT:TERMination
- SUPER:PROGrama:CURRENT:DISCharge
- SUPER:PROGrama:CURRENT:DISCharge:REDUce
- **Syntax:** SUPER:PROGrama:CURRENT:XXX <ch> <num>[A]
- **TC-Anwendungsbeispiel (teilweise):**

```
#cmdSetup number Precharge_Current CH1_Super_Controller
:write: SUPE:PROGrama:CURRENT:PRECharge ch1,#A; [200];
:read: PROGrama:CURRENT:PRECharge?  ch1;
:readmath: (value/1000.0)
:update:CH1_Super_Controller.Choose_A_Program_To_Run
A 0.030 2.000
:visible:inList(CH1_Super_Controller.Battery_Type, "Ni Li LF Li430 Li435")

#cmdSetup number charge_Current CH1_Super_Controller
```

```
:write: SUPE:PROGrama:CURRENT:CHARGE ch1,#A; [200];
:read: PROGrama:CURRENT:CHARGE? ch1;
:readmath: (value/1000.0)
:update:CH1_Super_Controller.Choose_A_Program_To_Run
A 0.030 2.000
:visible:inList(CH1_Super_Controller.Battery_Type, "Ni Li LF Li430 Li435")
```

- **Beispiel:**

```
Senden: SUPER:PROGrama:CURRENT:CHARGE ch1,1.0A
Senden: SUPER:PROGrama:CURRENT:DISCHARGE ch1,0.5A
Antwort: (keine)
```

6.6 SUPER:PROGrama:VOLTage:*

Enthält:

- SUPER:PROGrama:VOLTage:TERMination
- SUPER:PROGrama:VOLTage:TARGet
- SUPER:PROGrama:VOLTage:PREChargeToFastCharge
- SUPER:PROGrama:VOLTage:CUT

Hinweis: Unter SUPER:PROGrama:VOLTage gibt es **kein** REStart. Um die Neustartspannung einzustellen, verwenden Sie PROGrama:VOLTage:REStart.

- **Syntax:** SUPER:PROGrama:VOLTage:XXX <ch> <num>[V]
- **TC-Anwendungsbeispiel (teilweise):**

```
#cmdSetup number End_Charge_Voltage CH1_Super_Controller
:write: SUPE:PROGrama:VOLTage:TERMination ch1,#V; [200];
:read: PROGrama:VOLTage:TERMination? ch1;
:readmath: (value/1000.0)
:update:CH1_Super_Controller.Choose_A_Program_To_Run
V 0.030 4.850
:visible:inList(CH1_Super_Controller.Battery_Type, "Li LF Li430 Li435")
```

- **Beispiel:**

```
Senden: SUPER:PROGrama:VOLTage:TERMination ch1,4.2V
Senden: SUPER:PROGrama:VOLTage:CUT ch1,3.0V
Antwort: (keine)
```

6.7 SUPER:PROGrama:DELTApeak

- **Typ:** Einstellung
- **Syntax:** SUPER:PROGrama:DELTApeak <ch> <Advice|num[mV|V]>
- **TC-Anwendungsbeispiel:**

```
#cmdSetup combobox Delta_Peak_Voltage CH1_Super_Controller
:write: SUPER:PROGrama:DELTApeak ch1,#; [200];
:read: PROGrama:DELTApeak? ch1;
:update:CH1_Super_Controller.Choose_A_Program_To_Run
:string:
```

```
0dv 0
1_mv 1
2_mv 2
3_mv 3
4_mv 4
5_mv 5
6_mv 6
7_mv 7
8_mv 8
9_mv 9
Advice 10
:visible: CH1_Super_Controller.Battery_Type=="Ni"
```

- **Beispiel:**

```
Senden: SUPER:PROGrama:DELTApeak Ch1,5mV
Senden: SUPER:PROGrama:DELTApeak Ch1,Advice
Antwort: (keine)
```

6.8 SUPER:CONTROL:ACTivatebattery:START / EXIT

- **Typ:** Aktion
- **Syntax:** SUPER:CONTROL:ACTivatebattery:START <ch> / EXIT <ch>
- **TC-Anwendungsbeispiel:**

```
#cmdSetup button Start_Battery_Activation_Process CH1_Super_Controller
:write:SUPER:CONTROL:ACTivatebattery:START CH1;[200];
:tip:Protected cells with Low Voltage Protection triggered are generally safe. But
activation is not recommended for unprotected over-discharged cells. The process lasts 2
minutes.

#cmdSetup button Exit_Battery_Activation_Process CH1_Super_Controller
:write:SUPER:CONTROL:ACTivatebattery:EXIT CH1;[200];
:tip:Terminate the battery activation process early. After 2 minutes, the charger will
automatically end the process.
```

- **Beispiel:**

```
Senden: SUPER:CONTROL:ACTivatebattery:START Ch1
Senden: SUPER:CONTROL:ACTivatebattery:EXIT Ch1
Antwort: (keine)
```

7. GSV-Globalkonfiguration

7.1 CONFIG:FANS:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** CONFIG:FANS:ENABLE <bool> / CONFIG:FANS:ENABLE?
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonson Fans Global_Settings
:write: CONFIG:FANS:ENABLE #; [100]
:read: CONFIG:FANS:ENABLE?;
:readmath: getElement("OFF,ON", value );
:tip: Toggle the fans On or Off
:string:
Off OFF
On ON
:update: Global_Settings.Fans_Status
```

- **Beispiel:**

```
Senden: CONFIG:FANS:ENABLE ON
Senden: CONFIG:FANS:ENABLE?
Antwort: 1 (zeigt ON an)
```

7.2 CONFIG:BEEPer:KEY:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonson Button_Beeper Global_Settings
:write: CONFIG:BEEPer:KEY:ENABLE #; [100]
:read:CONFIG:BEEPer:KEY:ENABLE?;
:readmath: getElement("OFF,ON", value );
:tip: Toggle the key beeper On or Off
:string:
Off OFF
On ON
```

- **Beispiel:**

```
Senden: CONFIG:BEEPer:KEY:ENABLE ON
Senden: CONFIG:BEEPer:KEY:ENABLE?
Antwort: 1
```

7.3 CONFIG:BEEPer:PROcesScompletion:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonson Process_Completion_Beeper Global_Settings
:write: CONFIG:BEEPer:PROcesScompletion:ENABLE #; [100]
:read:CONFIG:BEEPer:PROcesScompletion:ENABLE?;
:readmath: getElement("OFF,ON", value );
:tip: Toggle the Process completion beeper On or Off
:string:
Off OFF
On ON
```

- **Beispiel:**

```
Senden:  CONFIg:BEEPer:PROCEsscompletion:ENABle ON
Senden:  CONFIg:BEEPer:PROCEsscompletion:ENABle?
Antwort: 1
```

7.4 CONFIg:SCREenflash:ENABle / ?

- **Typ:** Einstellung / Abfrage
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonsOn Process_Completion_Screen_Flash Global_Settings
:write: CONFIg:SCREenflash:ENABle #; [100]
:read:CONFIg:SCREenflash:ENABle?;
:readmath: getElement("NO,YES", value );
:tip: Toggle the Process completion beeper On or off
:string:
No NO
Yes YES
```

- **Beispiel:**

```
Senden:  CONFIg:SCREenflash:ENABle YES
Senden:  CONFIg:SCREenflash:ENABle?
Antwort: 1
```

7.5 CONFIg:LISPfn:ENABle / ?

- **Typ:** Einstellung / Abfrage
- **Beschreibung:** Li-Ionen-Starkschutz, invertierte Logik.
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonsOn LiIon&LifeP04_Strong_Protect Global_Settings
:read: CONFIg:LISPfn:ENABle?
:readmath: getElement("OFF ON",value);
:write: CONFIg:LISPfn:ENABle #; [100]
:tip: Charging won't start for liion batteries below 2V when LISP is enabled.
:string:
Off OFF
On ON
:update: Global_Settings.LiIon&LifeP04_Strong_Protect_Status
```

- **Beispiel:**

```
Senden:  CONFIg:LISPfn:ENABle ON
Senden:  CONFIg:LISPfn:ENABle?
Antwort: 1
```

7.6 CONFIG:LDSPfn:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **Beschreibung:** 1,5-V-Li-Ionen-Starkschutz.
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonsOn 1.50V_LiIon_Strong_Protect Global_Settings
:read: CONFIG:LDSPfn:ENABLE?
:readmath: getElement("OFF ON",value);
:write: CONFIG:LDSPfn:ENABLE #; [100]
:tip: Charging won't start for 1.5V liion batteries above 2V when LISP is enabled.
:string:
Off OFF
On ON
:update: Global_Settings.1.50V_LiIon_Strong_Protect_Status
```

- **Beispiel:**

```
Senden: CONFIG:LDSPfn:ENABLE ON
Senden: CONFIG:LDSPfn:ENABLE?
Antwort: 1
```

7.7 CONFIG:AUTOstart:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonsOn Enable_Auto-start_Program Global_Settings
:read: CONFIG:AUTOstart:ENABLE?
:readmath: getElement("NO YES",value);
:write: CONFIG:AUTOstart:ENABLE #; [100]
:tip: Allows auto-start of the program. The program will not auto-start on the first
power-up.
:string:
No NO
Yes YES
:update: Global_Settings.Enable_Auto-start_Program_Status
```

- **Beispiel:**

```
Senden: CONFIG:AUTOstart:ENABLE YES
Senden: CONFIG:AUTOstart:ENABLE?
Antwort: 1
```

7.8 CONFIG:CALibration:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonson Enable_Calibration Global_Settings
:read: CONFIG:CALibration:ENABLE?
:readmath: getElement("NO YES",value);
:write: CONFIG:CALibration:ENABLE #; [100]
:tip: Allow calibration of voltage, charge current, and discharge current.
:string:
No NO
Yes YES
:update: Global_Settings.Enable_Calibration_Status
```

- **Beispiel:**

```
Senden: CONFIG:CALibration:ENABLE YES
Senden: CONFIG:CALibration:ENABLE?
Antwort: 1
```

7.9 CONFIG:TESTir:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonson Enable_Test_IR Global_Settings
:read: CONFIG:TESTir:ENABLE?
:readmath: getElement("NO YES",value);
:write: CONFIG:TESTir:ENABLE #; [100]
:tip: Enable battery internal resistance test? Measurements are for reference only.
:string:
No NO
Yes YES
:update: Global_Settings.Enable_Test_IR_Status
```

- **Beispiel:**

```
Senden: CONFIG:TESTir:ENABLE YES
Senden: CONFIG:TESTir:ENABLE?
Antwort: 1
```

7.10 CONFIG:ACTivatebattery:ENABLE / ?

- **Typ:** Einstellung / Abfrage
- **TC-Anwendungsbeispiel:**

```
#cmdSetup buttonson Enable_0V_Battery_Activation Global_Settings
:read: CONFIG:ACTivatebattery:ENABLE?
:readmath: getElement("NO YES",value);
:write: CONFIG:ACTivatebattery:ENABLE #; [100]
:tip: Enable 0V battery activation?Activating over-discharged lithium-ion batteries is
not recommended.
:string:
No NO
Yes YES
:update: Global_Settings.Enable_0V_Battery_Activation_Status
```

- **Beispiel:**

```
Senden:  CONFIg:ACTIvatebattery:ENABle YES
Senden:  CONFIg:ACTIvatebattery:ENABle?
Antwort:  1
```

7.11 CONFIg:DEFAuLtprogram / ?

- **Typ:** Einstellung / Abfrage
- **Syntax:** CONFIg:DEFAuLtprogram <ch> <program 0~31>
- **TC-Anwendungsbeispiel (CH1):**

```
#cmdSetup combobox CH1_Default_Program Global_Settings
:write: CONFIg:DEFAuLtprogram Ch1,#;[100]
:read: CONFIg:DEFAuLtprogram? Ch1;
:tip: The default program is loaded when the charger is powered on or when the "Restore
Default Program" operation is performed.
:string:
Program_0 0
Program_1 1
...
Program_31 31
```

- **Beispiel:**

```
Senden:  CONFIg:DEFAuLtprogram Ch1,0
Senden:  CONFIg:DEFAuLtprogram? Ch1
Antwort:  0
```

8. Kalibrierung, Backup und System-Reset

8.1 PACKet:RATIo:RUN?

- **Typ:** Abfrage
- **Syntax:** PACKet:RATIo:RUN? <ch>
- **Rückgabe:** PACKet:RATIo #<len><data>,<ch>,<checksum>\r\n
- **Beispiel:**

```
Senden:  PACKet:RATIo:RUN? Ch1
Antwort:  PACKet:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.000
0000000,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,1,28
```

8.2 PACKet:RATIo:BACKup?

- **Typ:** Abfrage
- **Syntax:** PACKet:RATIo:BACKup? <ch>
- **Rückgabe:** PACKet:RATIo #<len><data>,<ch>,<checksum>\r\n
- **Beispiel:**

Senden: PACKeT:RATIo:BACKup? Ch1

Antwort: PACKeT:RATIo

#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,1,28

8.3 PACKeT:RATIo

- **Typ:** Einstellung
- **Syntax:** PACKeT:RATIo <block>
- **Beschreibung:** Parst 14 Kalibrierparameter und schreibt sie in Flash. ⚠ **Gefährlich.**
- **Beispiel eines tatsächlichen Kalibrierparameter-Download-Skripts** (aus Gyrfalcon_S4000Pro_calibration_parameter_download_Script_*.txt):

```
CONTRoL:PROGRaM:PREProceSS (channel);[200];
#iF (deviceRead(device,command_PreprocessBeforeSettingProgramQ) == "Ready")
PACKeT:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,1,28;[200];
PACKeT:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,2,27;[200];
PACKeT:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,3,26;[200];
PACKeT:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,4,25;[200];
CONTRoL:PROGRaM:COMPlETed (channel);[50];
#endiF
```

Hinweis: Für jeden Kanal wird ein Satz Kalibrierparameter geschrieben. Die letzten beiden Felder sind Kanalnummer und Prüfsumme. Die Kanäle 1–4 verwenden die Prüfsummen 28, 27, 26 und 25.

Sonderfall: In diesem Skript ist `channel = "None"`, und `CONTRoL:PROGRaM:PREProceSS None` sowie `CONTRoL:PROGRaM:COMPlETed None` sind beide zulässig (sie akzeptieren `channel_no = 0`).

8.4 :SYSTem:BACKup:DATA

- **Typ:** Aktion
- **Syntax:** :SYSTem:BACKup:DATA
- **Beschreibung:** Sichert Daten in den Backup-Bereich. ⚠ **Gefährlich.**
- **Beispiel:**

Senden: :SYSTem:BACKup:DATA

Antwort: (keine)

8.5 :SYSTem:RESEt:PROGram

- **Typ:** Aktion
- **Syntax:** :SYSTem:RESEt:PROGram
- **Beschreibung:** Setzt Programme 0–31 auf Werkseinstellungen zurück, springt zum Bootloader. ⚠ **Gefährlich.**
- **Beispiel:**

```
Senden: :SYSTem:RESEt:PROGram
Antwort: (keine)
```

8.6 :SYSTem:RESEt:DATA

- **Typ:** Aktion
- **Syntax:** :SYSTem:RESEt:DATA <bank1|bank2|1|2>
- **Beschreibung:** Stellt Backup-Daten wieder her, springt zum Bootloader. ⚠ **Gefährlich.**
- **Beispiel:**

```
Senden: :SYSTem:RESEt:DATA 1
Antwort: (keine)
```

8.7 :SYSTem:RESEt:CALIbration

- **Typ:** Aktion
- **Syntax:** :SYSTem:RESEt:CALIbration
- **Beschreibung:** Stellt Kalibrierparameter aus Backup wieder her und startet neu. ⚠ **Gefährlich.**
- **Beispiel:**

```
Senden: :SYSTem:RESEt:CALIbration
Antwort: (keine)
```

9. Typischer vollständiger Arbeitsablauf

9.1 Steuerungsskript-Vorlage

Anwendungsfall: Mehrere Process-Schritte nacheinander ausführen, z. B. "Entladen → Ruhen → Laden → Ruhen → Entladen".

Wichtige Schritte:

1. `CONTROl:TRANSferChannelContro1ToPC` — Steuerung übertragen
2. Vor jedem Process Konfigurationsbefehle einfügen
3. `CONTROl:PROcess:START` — starten
4. `STATUS:CONTROl:PROcess:COMPLeted?` — auf Abschluss pollen
5. `CONTROl:RESTing` — ruhen
6. `CONTROl:RELEase` — Steuerung freigeben

```
#scriptMenu GyrfaLcon_S4000Pro_Control_Script_Template_20261002124904
```

```
#logcmds 0
```

```
=var device=getDevice("Charger")
```

```
(device):
```

```
=var channel="Ch1";
```

```
=var runNow="Yes";
```

```
=var restingTime=10;
```

```
=var command_TRANSFERChannelControlToPCQ= "CONTROL:TRANSFERChannelControlToPC? "+channel
```

```
=var command_ProcessCompletedQ= "STATUS:CONTROL:PROCESS:COMPLETED? "+channel
```

```
;;Start
```

```
CONTROL:TRANSFERChannelControlToPC (channel),(runNow);[200];
```

```
#if (deviceRead(device,command_TRANSFERChannelControlToPCQ) == "Transferred")
```

```
;;*****Process 1*****
```

```
;;The content generated by the "Configuration File for Control Scripts"
```

```
;;can be directly copied. Please refer to the format specified in the reference
```

```
;;documentation.
```

```
;;- - - - - For example - - - - -
```

```
;;PROGRAM:TYPE (channel),Li;[50];
```

```
;;PROGRAM:TASK (channel),D;[50];
```

```
;;PROGRAM:CURRENT:DISCHARGE (channel),1.0A;[50];
```

```
;;PROGRAM:TIME:CUT (channel),600min;[50];
```

```
;;PROGRAM:CURRENT:DISCHARGE:REDUCE (channel),0.4A;[50];
```

```
;;PROGRAM:VOLTAGE:CUT (channel),3.3V;[50];
```

```
;;CONTROL:PROCESS:START (channel);[50];
```

```
;;- - - - - Example end - - - - -
```

```
;;*****
```

```
*
```

```
#while (deviceRead(device,command_ProcessCompletedQ) != "Completed")
```

```
=sleep(5000)
```

```
#endwhile
```

```
CONTROL:RESTING (channel),(restingTime)min;[50];
```

```
;;*****Process 2*****
```

```
;;The content generated by the "Configuration File for Control Scripts"
```

```
;;can be directly copied. Please refer to the format specified in the reference
```

```
;;documentation.
```

```
;;*****
```

```
*
```

```
#while (deviceRead(device,command_ProcessCompletedQ) != "Completed")
```

```
=sleep(5000)
```

```
#endwhile
```

```

CONTROL:RESTing (channel),(restingTime)min;[50];

;;*****Process 3*****

;;The content generated by the "Configuration File for Control Scripts"
;;can be directly copied. Please refer to the format specified in the reference
;;documentation.

;;*****
*

#while (deviceRead(device,command_ProcessCompletedQ) != "Completed")
=sleep(5000)
#endwhile

CONTROL:RELEase (channel);[50];
#endif

;;Finished

```

9.2 Konfigurationskopie-Skript-Vorlage

Anwendungsfall: Stapelweises Schreiben der Konfigurationen für Programme 0–31. Programm 0 ist Entladen, Programme 1, 2 und 4–31 sind Laden, Programm 3 ist NiMH-Entladen.

```

#scriptMenu Gyrfalcon_S4000Pro_Configuration_Copy_Script_20261002124620

#logcmds 0
=var device=getDevice("Charger")
(device):
=var channel = "None"
=var command_PreprocessBeforeSettingProgramQ = "CONTROL:PROGRAM:PREProcess? " + channel

=popupShowProgressAsync("Configuring GYRFALCON S4000Pro programs",32);

;;Start

CONTROL:PROGRAM:PREProcess (channel);[200];

#if (deviceRead(device,command_PreprocessBeforeSettingProgramQ) == "Ready")

;;*****
;;Details of program 0

;;Battery Type: LiionStd
;;Task: Discharge
;;Discharge Current: 0.95A
;;Max Discharge Time: 600min
;;Discharge Reduce Current: 0.1A
;;Discharge Cut Voltage: 2.8V
;;Program Long Name: DILI↓0.95A↓2.80V
;;Program Short Name: DILI↓0.95A

;;Configuration Script of Program 0

=popupShowProgressSetPosition(0);

```

```

PROGrama:SERIaI no (channel),0;[50];
PROGrama:TYPE (channel),Li;[50];
PROGrama:TASK (channel),D;[50];
PROGrama:CURRENT:DISCharge (channel),0.95A;[50];
PROGrama:TIME:CUT (channel),600min;[50];
PROGrama:CURRENT:DISCharge:REDUce (channel),0.1A;[50];
PROGrama:VOLTage:CUT (channel),2.8V;[50];
program:LongName
(replaceCharsR("DILI↓0.95A↓2.80V","\x7f\u2191\u2193\u29bb\u23b2\u2195\u22A5\u2126\u250d\u2515"
));[50];
program:ShortName
(replaceCharsR("DILI↓0.95A","\x7f\u2191\u2193\u29bb\u23b2\u2195\u22A5\u2126\u250d\u2515"));
[50];

CONTroI:PROGrama:SAVE (channel);[50];

;;*****
;;Details of program 1

;;Battery Type: LiionStd
;;Task: Charge
;;Max Capacity: 5500mAh
;;Precharge Current: 0.1A
;;Max Precharge Time: 180min
;;Precharge To Fast Charge Voltage: 3.0V
;;Charge Current: 1.0A
;;Max Charge Time: 600min
;;End Charge Current: 0.1A
;;End Charge Voltage: 4.2V
;;Restart voltage: 0.0V
;;Program Long Name: CHLI↑1.00A↑4.20V
;;Program Short Name: CHLI↑1.00A

;;Configuration Script of Program 1

=popupShowProgresssetPosition(1);

PROGrama:SERIaI no (channel),1;[50];
PROGrama:TYPE (channel),Li;[50];
PROGrama:TASK (channel),C;[50];
PROGrama:CAPAcity (channel),5500mAh;[50];
PROGrama:CURRENT:PRECharge (channel),0.1A;[50];
PROGrama:TIME:PRECharge (channel),180min;[50];
PROGrama:VOLTage:PREChargetofastcharge (channel),3.0V;[50];
PROGrama:CURRENT:CHARGE (channel),1.0A;[50];
PROGrama:TIME:CHARGE (channel),600min;[50];
PROGrama:CURRENT:TERMination (channel),0.1A;[50];
PROGrama:VOLTage:TERMination (channel),4.2V;[50];
PROGrama:VOLTage:REStArt (channel),0.0V;[50];
program:LongName
(replaceCharsR("CHLI↑1.00A↑4.20V","\x7f\u2191\u2193\u29bb\u23b2\u2195\u22A5\u2126\u250d\u2515"
));[50];
program:ShortName
(replaceCharsR("CHLI↑1.00A","\x7f\u2191\u2193\u29bb\u23b2\u2195\u22A5\u2126\u250d\u2515"));
[50];

```

```

CONTRol:PROGrama:SAVE (channel);[50];

;; Additional programs (2, 3, ... up to 31) can be added similarly

;;*****

CONTRol:PROGrama:COMPleted (channel);[50];
=popupShowProgressSetPosition(100);

#endif

;;Finished

```

9.3 Kalibrierparameter-Download-Skript

Anwendungsfall: Kalibrierparameter in CH1-CH4 schreiben.

```

#scriptMenu Gyrfalcon_s4000Pro_calibration_parameter_download_Script_20261002124528

#logcmds 0
=var device=getDevice("Charger")
(device):
=var channel= "None"
=var command_PreprocessBeforeSettingProgramQ = "CONTRol:PROGrama:PREProcess? " + channel

=popupShowProgressAsync("Download GYRFALCON S4000Pro calibration parameter",4);

;;Start

CONTRol:PROGrama:PREProcess (channel);[200];

#if (deviceRead(device,command_PreprocessBeforeSettingProgramQ) == "Ready")

;;*****Calibration Parameter Download Script*****

=popupShowProgressSetPosition(1);
PACKet:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000
0,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,1,28;[200];

=popupShowProgressSetPosition(2);
PACKet:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000
0,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,2,27;[200];

=popupShowProgressSetPosition(3);
PACKet:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000
0,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,3,26;[200];

=popupShowProgressSetPosition(4);
PACKet:RATIo
#31611.5465085506,0.9604977369,0.3318018615,1.0000000000,3.2911419868,0.0024977922,1.0000000000
0,3.2911419868,0.0024977922,1.0000000000,3.3124411106,-0.0238245726,4,25;[200];

;;*****END*****

```

```

CONTRol:PROGrama:COMPleted (channel);[50];
=popupShowProgressSetPosition(100);

#endif

;;Finished

```

10. Fehlercodetabelle

Die von `SCPI_Error` ausgegebenen dezimalen Fehlercodes folgen dem SCPI-Standard. Häufige Fehlercodes:

Fehlercode	Makro	Bedeutung
-100	<code>SCPI_ERROR_COMMAND_ERROR</code>	Befehl fehlerhaft
-109	<code>SCPI_ERROR_MISSING_PARAMETER</code>	Parameter fehlt
-113	<code>SCPI_ERROR_UNDEFINED_HEADER</code>	Header nicht definiert
-120	<code>SCPI_ERROR_NUMERIC_DATA_ERROR</code>	Numerischer Datenfehler
-140	<code>SCPI_ERROR_CHARACTER_DATA_ERROR</code>	Zeichendatenfehler
-160	<code>SCPI_ERROR_BLOCK_DATA_ERROR</code>	Blockdatenfehler
-161	<code>SCPI_ERROR_INVALID_BLOCK_DATA</code>	Ungültige Blockdaten
-200	<code>SCPI_ERROR_EXECUTION_ERROR</code>	Ausführungsfehler
-221	<code>SCPI_ERROR_SETTINGS_CONFLICT</code>	Einstellungskonflikt
-222	<code>SCPI_ERROR_PARAMETER_VALUE_OUT_OF_RANGE</code>	Parameterwert außerhalb des Bereichs
-224	<code>SCPI_ERROR_ILLEGAL_PARAMETER_VALUE</code>	Unzulässiger Parameterwert
-350	<code>SCPI_ERROR_QUEUE_OVERFLOW</code>	Fehlerwarteschlange überlaufen
-410	<code>SCPI_ERROR_QUERY_INTERRUPTED</code>	Abfrage unterbrochen
-420	<code>SCPI_ERROR_QUERY_UNTERMINATED</code>	Abfrage nicht abgeschlossen
-430	<code>SCPI_ERROR_QUERY_DEADLOCKED</code>	Abfrage blockiert
-440	<code>SCPI_ERROR_QUERY_UNTERMINATED_AFTER_INDEFINITE_RESPONSE</code>	Abfrage nach unbestimmter Antwort nicht beendet

Die Fehlerausgabe erfolgt üblicherweise als Dezimalzahl mit Zeilenumbruch, zum Beispiel:

```

Senden:  PROGRAM:TYPE Ch1,XYZ
Antwort: -224

```