

Programmers guide

Last changed on: 2021-02-16

iseq SCPI general instruction set

Applies to the HV power supply devices

- NHS
- NHR
- SHR
- MICC
- HPS
- FPS

Document history

Version	Date	Major changes
2.3	2021-02-16	Added more general information the SCPI command line handling Added more information the serial echo handling Smaller editorial changes
2.2	2020-11-10	Added description for HFS devices Smaller editorial changes
2.1	2020-11-02	Added the missing channel bits Arc and Arc Error Added description for Arc, ADC Sample Rate, Digital Filter and Fine Adjustment Added description for Module Event Channel Status
2.0	2020-10-07	Added HPS and FPS devices Added more examples to the existing devices
1.3	2020-05-05	Change Inhibit action of channel
1.2	2020-02-04	Added CDC-ACM USB driver Link
1.1	2019-11-07	Troubleshooting
1.0	2019-09-16	Initial version

Disclaimer / Copyright

Copyright © 2021 by iseg Spezialelektronik GmbH / Germany. All Rights Reserved.

This document is under copyright of iseg Spezialelektronik GmbH, Germany. It is forbidden to copy, extract parts, duplicate for any kind of publication without a written permission of iseg Spezialelektronik GmbH. This information has been prepared for assisting operation and maintenance personnel to enable efficient use.

The information in this manual is subject to change without notice. We take no responsibility for any mistake in the document. We reserve the right to make changes in the product design without reservation and without notification to the users. We decline all responsibility for damages and injuries caused by an improper use of the device.





Safety

This section contains important security information for the installation and operation of the device. Failure to follow safety instructions and warnings can result in serious injury or death and property damage.

Safety and operating instructions must be read carefully before starting any operation.

We decline all responsibility for damages and injuries caused which may arise from improper use of our equipment.

Depiction of the safety instructions

DANGER!	
 DANGER!	<p>“Danger!” indicates a severe injury hazard. The non-observance of safety instructions marked as “Danger!” will lead to possible injury or death.</p>
WARNING!	
 WARNING!	<p>“Warning!” indicates an injury hazard. The non-observance of safety instructions marked as “Warning!” could lead to possible injury or death.</p>
CAUTION!	
 CAUTION!	<p>Advices marked as “Caution!” describe actions to avoid possible damages to property.</p>
INFORMATION	
 INFORMATION	<p>Advices marked as “Information” give important information.</p>



Read the manual.



Attention high voltage!

HIGH VOLTAGE



Important information.

Intended use

The device may only be operated within the limits specified in the data sheet. The permissible ambient conditions (temperature, humidity) must be observed. The device is designed exclusively to control high voltage systems as specified in the data sheet. It must only be used specified in Technical data. Any other use not specified by the manufacturer is not intended. The manufacturer is not liable for any damage resulting from improper use.

Qualification of personnel

A qualified person is someone who is able to assess the work assigned to him, recognize possible dangers and take suitable safety measures on the basis of his technical training, his knowledge and experience as well as his knowledge of the relevant regulations.

General safety instructions

- Observe the valid regulations for accident prevention and environmental protection.
- Observe the safety regulations of the country in which the product is used.
- Observe the technical data and environmental conditions specified in the product documentation.
- You may only put the product into operation after it has been established that the high-voltage device complies with the country-specific regulations, safety regulations and standards of the application.
- The high-voltage power supply unit may only be installed by qualified personnel.

DANGER!



DANGER!

This device is part of a high voltage supplying systems.
High voltages are dangerous and may be fatal.

USE CAUTION WHILE WORKING WITH THIS EQUIPMENT.
BE AWARE OF ELECTRICAL HAZARDS.

Always follow at the minimum these provisions:

- High voltages must always be grounded
- Do not touch wiring or connectors without securing
- Never remove covers or equipment
- Always observe humidity conditions
- Service must be done by qualified personnel only

WARNING!



WARNING!

Before connecting or disconnecting HV cables or any operation on the HV output or the application, the unit has to be switched off and discharge of residual voltage has to be finished. Depending on application, residual voltages can be present for long time periods.

CAUTION!



CAUTION!

When controlling the high voltage systems with software, make sure that nobody is near the high voltage or can be injured.

INFORMATION



INFORMATION

Please check the compatibility with the devices used.

Table of Contents

Document history	2
Safety	3
Depiction of the safety instructions	3
Intended use	4
Qualification of personnel	4
General safety instructions	4
1 General information to high voltage devices	8
1.1. Channel and Module	8
1.2. Devices covered by this manual	8
1.3. High voltage device operation modes	9
1.3.1 Terminology	9
1.3.2 Channel operation modes	10
1.4. Status and Event generation	11
1.5. Hardware options	11
1.5.1 Option VCT (Voltage Compensation by Temperature)	11
2 Introduction to SCPI commands	12
2.1. Introduction	12
2.2. General Command Syntax	12
3 Interfaces	15
3.1. USB and RS-232 interface	16
3.1.1 Programming	16
3.1.2 Serial Echo Handling	16
3.1.3 RS-232 interface	17
3.1.4 USB interface	17
3.1.5 Windows USB driver installation	18
3.1.6 Linux USB driver installation	19
3.2. Ethernet	21
3.2.1 Hardware Implementations	21
3.2.2 Connecting over Ethernet	21
3.2.3 Ethernet programming example	22
4 Common instruction set	24
5 NHR, SHR, NHS and MICC	25
5.1. SCPI Instruction Set	25
5.2. Hardware Options	35
5.2.1 Option VCT	35
6 HPS and FPS	36
6.1. SCPI Instruction Set	36
6.2. ARC Management configuration for HPS ≥ 1.5 kW	42
6.3. Controlling HFS devices	42
6.3.1 Introduction	42
6.3.2 FPS Heating and Emission Regulation parameters	43
6.3.3 Direct Heating and Emission Regulation	43
7 Output formats for voltage and current	44
7.1. Examples	45
7.1.1 Simple examples	45
7.1.2 Advanced examples	45
7.1.3 Commands for NHR and SHR	45

8	Description of control, status, event, and mask registers	46
8.1.	Status and control register decoding	46
8.2.	Channel Status	46
8.3.	Channel Event Status	48
8.4.	Channel Event Mask	50
8.5.	Channel Control	52
8.6.	Module Status	53
8.7.	Module Event Status	55
8.8.	Module Event Mask	56
8.9.	Module Control	57
8.10.	Module Event Channel Status	58
8.11.	Module Event Channel Mask	58
9	iseq Terminal	59
10	iseq SCPI Control	60
11	ASCII character table	61
12	References	62
13	Manufacturer contact	62

1 General information to high voltage devices

1.1. Channel and Module

A high voltage channel is a single high voltage output and measurement circuit. The channel has different operation modes and provides different measurements and status flags.

A module is the combination of one or more high voltage channels in a common housing. Examples are a HPS with one channel or a NHS with six channels. Beside the combination of all channels, the module provides some more measurement and status information for the whole device.

Module commands set a module-wide function or return a module-wide status or measurement value. Channel commands, in contrast, operate on a specific channel. In the SCPI command set, this channel is specified as command suffix, for example (@1).

1.2. Devices covered by this manual

All devices described in this manual have in common, that they support the EDCP (Enhanced device communication protocol) in combination with the SCPI command set. Some devices also support other command sets, these are not covered by this manual and can be looked up directly in the specific device's manual.

- SHR is a high precision desktop high voltage power supply with four channels of switchable polarity
- NHR is a high precision, high voltage power supply in NIM format, with four channels of switchable polarity
- NHS is a high precision, high voltage power supply in NIM format, with six channels of fixed polarity
- MICC is a multipurpose input/output controller to build modular HV systems in 3U MMC format
- EHQ is an one channel high voltage power supply in MME format
- HPS is a series of industrial high voltage devices, typically with one output channel and power greater 300 W
- FPS is a potential-free low voltage power device, typically with one channel and currents starting at 6 A

1.3. High voltage device operation modes

1.3.1 Terminology

Syntax	Declaration
V_{nom}	Voltage nominal, the maximum possible output voltage
I_{nom}	Current nominal, the maximum possible output current
V_{set}	Voltage Set, the user-controllable demanded output voltage
V_{meas}	Voltage Measure, the actual measured output voltage
V_{lim}	Voltage Limit. Can be a hardware or a software limit and serves two purposes: <ol style="list-style-type: none"> 1. It limits the upper value of V_{set} to the given limit value: $V_{set} \leq V_{lim}$ 2. It generates the Channel Status is Voltage Limit if V_{meas} exceeds the limit value (the exact threshold value is device dependent)
I_{set}	Current Set, the user-controllable demanded output current
I_{meas}	Current Measure, the actual measured output current
I_{lim}	Current Limit. Can be a hardware or a software limit and serves two purposes: <ol style="list-style-type: none"> 1. It limits the upper value of I_{set} to the given limit value: $I_{set} \leq I_{lim}$ 2. It generates the Channel Status is Current Limit if I_{meas} exceeds the limit value (the exact threshold value is device dependent)
V_{bounds}	Voltage bounds, a tolerance tube $V_{set} \pm V_{bounds}$ around V_{set} . If V_{meas} exceeds this tolerance in either direction, the Channel Status is Voltage Bounds is generated. (condition: no ramping)
I_{bounds}	Current bounds, a tolerance tube $I_{set} \pm I_{bounds}$ around I_{set} . If I_{meas} exceeds this tolerance in either direction, the Channel Status is Current Bounds is generated. (condition: no ramping)

Table 1: Terminology

1.3.2 Channel operation modes

Operation Mode	Description	
Off	The channel is off, it does not generate high voltage. If all status conditions are satisfied, the channel can be turned on.	
On	The channel is actively generating high voltage.	
Ramping	The channel ramps to the voltage set V_{set} if turned on or is turns off with the programmed ramp speed. NHR and SHR devices provide voltage and current ramp speed setting per channel in V/s resp. A/s. All other devices have a common voltage and current ramp speed for all channels in %/s. A ramp speed of 1 V/s means, the voltage changes with 1 Volt every second until it reaches the destination. A ramp speed of 1 %/s means, that for a voltage nominal of 1000 V, the voltage changes with 10 Volt every second until it reaches the destination. ($1000\text{ V} \cdot 1\text{ %/s} = 10\text{ V/s}$)	
Emergency Off	The channel is shut down without ramp. Afterwards, it stays in Emergency Off until Emergency Clear is given.	
Emergency Clear	The channel leaves the state Emergency Off and goes to state Off. If the channel is not in Emergency Off, nothing happens.	
Kill Enable	The mode Kill Enable provides a higher safety. This mode is module-wide and therefore affects all channels. The channel will go to Trip and shut down without ramp when any of the following conditions occur:	
	<ul style="list-style-type: none"> • $V_{meas} > V_{lim}$ OR • $V_{meas} > V_{set} + V_{bounds}$ • $I_{meas} > I_{lim}$ OR • $V_{meas} < V_{set} - V_{bounds}$ • $I_{meas} > I_{set}$ OR • $I_{meas} < I_{set} - I_{bounds}$ 	
Constant Voltage	The channel operates as constant voltage source, that means $V_{meas} \approx V_{set}$ and $I_{meas} < I_{set}$	
Constant Current	The channel operates as constant current source, that means $V_{meas} < V_{set}$ and $I_{meas} \approx I_{set}$	
Delayed Trip	This is a special mode of Constant Current. If this mode is activated, and I_{meas} reaches or exceeds I_{set} , the Channel Status Is Current Trip is generated. Depending on the trip configuration, the channel may stay in Constant Current, or turn off with ramp or shut down without ramp. Trip also happens when Kill Enable is active and any of the Kill conditions occur.	
External Inhibit	External Inhibit is a hardware line to control the high voltage generation. Depending on the device, there might be one External Inhibit per channel or one External Inhibit for all channels. Some devices always shut down the high voltage without ramp on External Inhibit, some devices allow to configure this function, as shown below.	
	External Inhibit individual per channel	NHS, NHR, SHR
	External Inhibit for all channels	MICC, HPS, FPS
	External Inhibit Action is configurable	NHS, NHR, SHR
	External Inhibit shuts down channel without ramp	MICC, HPS, FPS
Input Error	An input error occurs, if an invalid command or parameter is given, or the parameter of a command exceeds the allowed range. Example: setting a V_{set} of 4000 V for a channel with V_{nom} of 3000 V.	
Fine Adjustment	The fine adjustment allows a long-term stabilization of the output voltage. Since the DAC typically drifts more with temperature changes, it is readjusted based on the ADC measurement values.	
ADC Sample Rate	The ADC sample rate determines the resolution and noise of the measurement. Lower sample rates typically lead to better results but slower updating.	

Operation Mode	Description
Digital Filter	The digital filter allows a smoothing of the measured values. Higher digital filter values lead to less fluctuating measured values, but also to slower acceptance of changes. For highest speed the digital filter can be deactivated with the value 1.
Arc	An electrical arc is an electrical breakdown of a gas that produces a prolonged electrical discharge.
Output Mode	NHR and SHR devices provide switchable output modes with different voltage and current combinations, e.g. 6 kV/1 mA and 2 kV/3 mA.
Output Polarity	NHR and SHR devices provide switchable output polarities, positive and negative.

Table 2: Channel operation modes

1.4. Status and Event generation

Channel as well as Module have status and event registers. Both registers contain similar condition bits. The difference between both register types is, that status bits are set and cleared by the device according to the current conditions. Event bits, however, are only set by the device and must be cleared explicitly by the user (like a latch).

For example, the Status bit is Constant Current indicates that the channel *is now* in constant current mode. The Event bit Event Constant Current in contrast indicates, that the channel has been (or still is) in constant current mode since the last clearing of this bit.

It is thereby possible to clear all status flags at once or to just clear individual bits. In general it is not possible to clear an event bit if the corresponding status bit is still set. The status and event registers are described in detail in section 8 Description of control, status, event, and mask registers.

1.5. Hardware options

Some devices can be attached with hardware options that provide extra functionality. These hardware options can be controlled with additional SCPI commands. These commands can only be used with a device that has the according hardware option installed.

A short overview of hardware options is given below. For more information, refer to the [devices hardware manuals](#), see 12 References.

1.5.1 Option VCT (Voltage Compensation by Temperature)

This option can be added to devices NHS or NHR. An external temperature sensor allows a V_{set} correction according a VCT factor and the temperature difference. The VCT features can be accessed with the 5.2.1 Option VCT.

2 Introduction to SCPI commands

2.1. Introduction

The high voltage devices can be connected to a control computer by the USB, RS-232, Ethernet, or GPIB interface. All these interfaces use the same general format for commands and query responses. The commands are transmitted as plain text, encoded in the standard 7-bit ASCII character table (section 11 ASCII character table).

Some special ASCII characters are notably highlighted in this document. These characters have to be replaced with their correct counterpart:

<CR> (0x0D or 13) the carriage return character

<LF> (0x0A or 10) the line feed character

␣ (0x20 or 32) the space character

Example: The command `:MEAS_VOLT?_ (@3) <CR><LF>` would give the following string in the C programming language:

```
":MEAS:VOLT? (@3) \r\n"
```

Two general types of commands exists:

- **Order commands** which have no answer. These always cause the HV device to perform any action. Most often, these commands ends with a parameter that specifies the action. Example: `:VOLT_ON`
It is possible to force an answer from an order command by appending the command operation complete (OPC).
Example: `:VOLT:ON;*OPC?`
- **Query commands** that returns a response. These generally do not cause the HV device to perform an action other than returning the response. All these commands have a keyword that ends with the "?" character. Example:
`:MEAS:VOLT?`

2.2. General Command Syntax

Each command is formed by a set of one or more fields. Each field is separated from the next by a field separator. The first field is always the command keyword, the remaining fields and their syntax depend on the command keyword. Commands are always executed in the same order in which they are received. If an error is found in an instruction set, processing of the instruction set is aborted and the remainder of the instruction set is not decoded or executed. For requests containing an error, no answer is sent back.

COMMAND LINES

All command lines sent to the device or received from the device end with `<CR><LF>`. For the sake of simplicity, `<CR><LF>` will not be explicitly mentioned later in the command tables, but nevertheless needs to be given when sending commands to the device.

A command line can contain one or multiple SCPI commands. More information about command grouping can be found in the subsection COMMAND HIERARCHY.

The device has only a limited buffer size to receive command lines. If more characters are sent to the device, the behavior is undefined. Similar, the device has only a limited transmit buffer. When a request generates an answer that exceeds the transmit buffer length, the behavior is also undefined.

Device	Receive buffer size	Transmit buffer size
NHS	80	200
NHR/SHR	80	220
MICC/MICCETH	80	400
EHQ 1-channel	80	120
HPS compact 350 W	80	140
HPS 19" 300/800 W	80	140
HPS 19" \geq 1.5 W	80	140
FPS	80	140

Table 3: Receive and transmit buffer sizes of the different devices

SERIAL ECHO

The RS-232 and USB interfaces have a special feature: they confirm all commands sent to the device with an echo. This can be used as a handshake information as well as an indication, that the command was successfully sent to the device. More information about this topic can be found in section 3.1.1: Programming.

DATA FORMATS

- UI4: An unsigned integer in decimal presentation, with value in the range 0..4294967295. This is the format used for status and event registers.
- Float: A floating point value with the format +1234.567E-6 (sign, mantissa, exponent). This is the format used for voltage and current values.
- String: A character string like "ON" or "OFF". These strings have to be given without quotes ("").

COMMAND KEYWORDS

Command keywords must exactly match the defined set for the device. The first field in a command is always the keyword.

Leading and trailing whitespaces are allowed and do not change the behavior of a command. The device evaluates commands case-insensitive, so both uppercase, lowercase or mixed notations are possible.

Uppercase bold letters given in this manual must always be written, lowercase letters are optional. For example, the command **:CONF**igure:**RAMP**:**VOLT**age can be written in any of the following forms with the same meaning:

```
:CONF:RAMP:VOLT
:conf:ramp:volt
:CONFIGure:RAMP:voltAGE
:CONFIGURE:RAMP:VOLT
:configure:ramp:voltage
```

PARAMETERS

Order commands need parameters to specify the action to take. The parameter is always separated from the command with one or more space characters. For example `:VOLT_100<CR><LF>` will set the voltage to 100 Volt.

The following types of parameters exist:

- Number: <Number> means either an integer or a float number can be given here
- String: A string without quotes ("")
- Range: {0...63} means the values 0 up to 63 (both included) are possible parameters here
- Enumeration: {125000|250000} means either 125000 or 250000 is a possible parameter here

COMMAND HIERARCHY

SCPI commands form a hierarchy or a tree, starting at the SCPI root, which is indicated by a colon ":".

A command line is always terminated by the character sequence carriage return and line feed <CR><LF>. The first command in a line always starts in the SCPI root, which means the first leading ":" can be omitted. So the following commands are both correct:

```
:VOLT_ON<CR><LF>
VOLT_ON<CR><LF>
```

Multiple SCPI commands can be given in one line, separated by a semicolon ";". All following commands start in the same hierarchy as the previous command, if no leading ":" is given. This can greatly reduce the amount of data transmitted to the HV device. So the following two commands are identical:

```
:MEAS:VOLT?;:MEAS:CURR?<CR><LF>
:MEAS:VOLT?;CURR?<CR><LF>
```

For request commands, the answer follows the same principle: each part is separated with a semicolon ";". So the complete answer for one of the commands above could look like this:

```
123.456V;1.2345E-3A<CR><LF>
```

CHANNELS

Most devices have multiple high voltage channels, numbered from 0 to ChannelNumber - 1. To address a specific channel within a command, the suffix (@<Channel>) is used. Depending on the command being an order or a query, a different syntax applies.

- *Order commands*

The channel suffix is separated from the parameter with a comma: `:VOLT 100, (@1) <CR><LF>`

- *Query commands*

The channel suffix is separated from the query with a space: `:MEAS:VOLT?_ (@3) <CR><LF>`

Especially to simplify query commands (but also usable for order commands), multiple channels can be used together with a single command. Therefore, the channel range (@0-3) and the channel enumeration (@0,1,3,5) notations exist. Both notations can be combined: (@0-2,5-7) which will operate on the channels 0, 1, 2, 5, 6, 7.

The different answer parts for such a request are separated by comma ",".

Example:

```
:MEAS:CURR?_ (@3-5) <CR><LF>
```

Answer:

```
1.1234E-3A, 1.2345E-3A, 1.3456E-3A<CR><LF>
```

3 Interfaces

The SCPI command set can be used with the following interfaces:

- USB (Virtual Serial Port)
- RS-232
- Ethernet (TCP/IP)
- GPIB (IEEE-488.2)

The following table shows which interfaces are available the different device types. The firmware name gives the possibility to distinguish between different devices from the control computer side. The firmware name can be queried with the command `:READ:FIRMWARE:NAME?`

Device	Firmware name	USB	RS-232	Ethernet	GPIB
NHS	N06C2	Yes	-	-	-
NHR	N04C2	Yes	-	-	-
SHR	S04C2	Yes	-	Yes	-
MICC	MICC	Yes	-	-	-
	MICCETH	Yes	-	Yes	-
EHQ 1 channel	E01C0	Yes	Yes	-	-
HPS compact 350 W	H201C0	Yes	Optional	Optional	-
HPS 19" 300/800 W	H101C0	Yes	Optional	Optional	Optional
HPS 19" \geq 1.5 kW	H101C1	Yes	Optional	Optional	Optional
FPS	FLM501	Yes	Optional	Optional	-

Table 4: Overview of available interfaces

A concrete description of the different interfaces follows in the next sections.

Many devices have more interfaces, which are not covered by this document. For example, the CAN interface with EDCP / DCP command list is described in the manual [CAN EDCP Programmers-Guide.pdf](#)¹.

There are ready made applications to use the SCPI command set without programming, such as [iseq Terminal](#)¹, [iseq SCPI Control](#)¹, and [iseq Control](#)¹. These allow to control the devices with its HV channels more convenient and without programming an own application.

Also, the [iseq Hardware Abstraction Layer](#)¹ allows connecting to the devices by SCPI and provides a generic programming API.

¹ See 12 References

3.1. USB and RS-232 interface

It is recommended to turn off the device with mains switch before connecting/disconnecting the interface cable.

If the device is equipped with both RS-232 and USB interface, only one of them must be connected at the same time.

3.1.1 Programming

The following description applies to both RS-232 and USB interface.

The (virtual) serial interface is set to 9600 Bit/second, 8 Bit/character, no parity, 1 Stop-Bit.

The data transfer is character oriented, and the synchronization in the direction "Computer to high voltage device" (Input direction) is established by echoes. That means every character received by the device on the RXD line is sent back over the TXD line to the computer. The transfer direction "HV-PS to computer" (Output direction) is free running, i.e. without echo.

The command transfer uses ASCII characters. Every command line is terminated by <CR><LF> (0x0D 0x0A or 13 10).

A new command may be sent immediately after the last answer was completely received (including <CR><LF>). For commands that don't return an answer, the simplest thing is to add *OPC? in [EDCP instruction set](#) (see chapter 12: References) :

Command sent by Computer	Echo and answer sent by HV device
:VOLT 500; :VOLT ON; *OPC?<CR><LF>	:VOLT 500; :VOLT ON; *OPC?<CR><LF>
	1<CR><LF>

Table 5: Programming the serial interface

3.1.2 Serial Echo Handling

An important difference between the USB- and RS-232 interface, in contrast to e.g. Ethernet, is that the device sends back every character it receives from the control computer. This is called "echo" and allows verifying that the command sent to the device was correctly received.

For the older devices EHQ 1-channel and HPS 300/800 W it is also recommended to use the echo as handshake. That means, every character (including <CR> and <LF>) should be sent on its own, waiting for the correct echo before sending the next character. As example, a *IDN? request would look like this:

```

Computer: * I D N ? <CR> <LF>
Device: * I D N ? <CR> <LF> iseg Spezialelektronik GmbH [...]<CR><LF>
  
```

Since this character-wise sending procedure slows down the communication, it should only be used for the devices that really need it (EHQ 1-channel and HPS 300/800 W).

For all other devices (NHR/S, SHR, MICC, HPS compact, HPS ≥ 1.5 kW, FPS), the serial stream will look like this:

```

Computer: *IDN?<CR><LF>
Device: *IDN?<CR><LF> iseg Spezialelektronik GmbH [...]<CR><LF>
  
```


3.1.3 RS-232 interface

The RS-232 interface is implemented with a female D-SUB-9 connector. Depending on the device, the connector is located at the front or the back panel.

The electric transfer is performed via the RXD and TXD lines, which are related to floating GND of the interface. The D-SUB 9 pin assignment is given in the following table.

The cable connection to the computer is direct (1:1, no null-modem-cable). If no 9-pin cable is available, the internal connection bridges for pin 4, 6, and 8 must be set up externally.

Signal	HV device		Computer
RS-232	D-SUB 9	Internal	D-SUB 9
RXD	2		2
TXD	3		3
GND	5		5
DTR	4	↖ (bridged)	4
DSR	6	↗ (bridged)	6
CTS	8	↘ (bridged)	8

Table 6: D-SUB 9 pin assignment

3.1.4 USB interface

Depending on the device, the USB interface is realized with different USB chipsets and connectors. The following table shows the hardware implementation for the different devices.

All devices have in common, that they operate as a virtual serial port in the PC, and can be used with every program that supports serial ports. Most often a driver installation is necessary before the virtual serial port can be used, though. For detailed driver installation procedure see the following chapters.

Device	USB connector	USB hardware
NHR, NHS, MICC	Female USB-B at the front panel	FTDI FT232R, see chapter 12: References
EHQ 1-channel	Female Mini-USB at the front panel	
HPS 300/800 W	Female USB-B at the back panel	
HPS compact 350 W	Female USB B at the front panel	
HPS ≥ 1.5 kW	Female USB-B at the back panel	
SHR	Female USB-B at the front panel	USB CDC-ACM, see chapter 12: References

Table 7: USB hardware implementations

3.1.5 Windows USB driver installation

FTDI USB DRIVER FOR NHS, NHR, MICC, EHQ 1 CHANNEL, HPS, AND FPS DEVICES

The FTDI VCP driver (Virtual COM Port) can be downloaded from our homepage:

<http://www.iseq-hv.com> → SUPPORT → DOWNLOADS → SOFTWARE → Tools → “FTDI-USB-Serial-Driver-Windows-2.x.x.exe”.

The installation starts with double click on the downloaded EXE file.

See chapter 12 References

CDC-ACM USB DRIVER FOR SHR DEVICES

This driver is already included in all Windows versions since Windows 7. The driver information file linux-cdc-acm.inf is needed after connecting a SHR in order to load the Windows driver.

The file linux-cdc-acm.inf can be downloaded from our homepage:

<http://www.iseq-hv.com> → SUPPORT → DOWNLOADS → SOFTWARE → Tools → “linux-cdc-acm.inf”.

See chapter 12 References

USB CONNECTION TEST

To determine if the installation was successful, check if the serial USB interface is shown in the Device Manager.

Start the Device Manager with:

Windows XP: Start → Settings → Control Panel → System → Device Manager

or Windows 7: Start → Control Panel → Device Manager

or Windows 10: Press the key combination Windows + R. Type the command: devmgmt.msc.

All devices get an USB Serial Port assigned in section Ports (COM & LPT), in this case COM3:

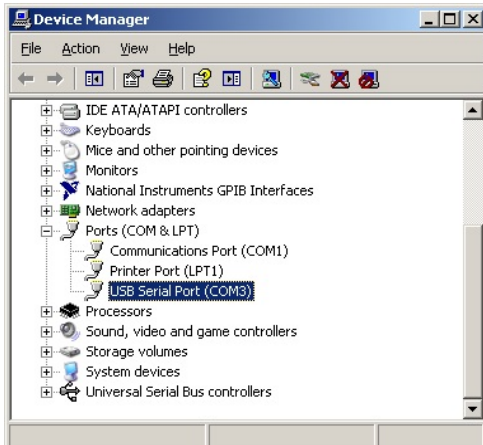


Figure 1: Windows Device Manager

The connection to the device can be tested with [iseq Terminal](#), see chapter 12: References.

3.1.6 Linux USB driver installation

FTDI USB DRIVER FOR NHS, NHR, MICC, EHQ 1 CHANNEL, HPS, AND FPS DEVICES

The driver is already included in Linux Kernel series 2.6 and higher and should be loaded automatically when connecting the device. The driver provides a virtual serial port like `/dev/ttyUSB0` that can be accessed with a Terminal program (e.g. [CuteCom](#)).

Make sure you have the proper permissions to access the serial port `/dev/ttyUSB0`, e.g. by adding your user to the group `dialout`.

The following dmesg output shows how the device is recognized and the driver loaded:

```
[234.496011] usb 1-2: new full speed USB device using uhci_hcd and address 2
[234.694884] usb 1-2: configuration #1 chosen from 1 choice
[234.704371] usb 1-2: New USB device found, idVendor=0403, idProduct=6001
[234.704376] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[234.704380] usb 1-2: Product: FT232R USB UART
[234.704382] usb 1-2: Manufacturer: FTDI
[234.704385] usb 1-2: SerialNumber: A60075cx
[234.807627] usbcore: registered new interface driver usbserial
[234.807649] usbserial: USB Serial support registered for generic
[234.807679] usbcore: registered new interface driver usbserial_generic
[234.807683] usbserial: USB Serial Driver core
[234.816739] usbserial: USB Serial support registered for FTDI USB Serial Device
[234.816774] ftdi_sio 1-2:1.0: FTDI USB Serial Device converter detected
[234.816805] ftdi_sio: Detected FT232RL
[234.816855] usb 1-2: FTDI USB Serial Device converter now attached to ttyUSB0
[234.816872] usbcore: registered new interface driver ftdi_sio
[234.816876] ftdi_sio: v1.4.3:USB FTDI Serial Converters Driver
```

CDC-ACM USB DRIVER FOR SHR DEVICES

The driver is already included in Linux Kernel series 2.6 and higher and should be loaded automatically when connecting the device. The driver provides a virtual serial port like `/dev/ttyACM0` that can be accessed with a Terminal program (e.g. [CuteCom](#)).

Make sure you have the proper permissions to access the serial port `/dev/ttyACM0`, e.g. by adding your user to the group `dialout`.

The following dmesg output shows how the device is recognized and the driver loaded:

```
[548.860461] usb 5-4: new high-speed USB device number 3 using xhci_hcd
[549.009618] usb 5-4: New USB device found, idVendor=0525, idProduct=a4a7
[549.009620] usb 5-4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[549.009622] usb 5-4: Product: Gadget Serial v2.4
[549.009623] usb 5-4: Manufacturer: Linux 4.1.35-v2.7b1 with 2184000.usb
[549.049630] cdc_acm 5-4:2.0: ttyACM0: USB ACM device
[549.049850] usbcore: registered new interface driver cdc_acm
[549.049851] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN
```

TESTING WITH CUTECOM

The following [screenshot shows](#) the connection to a NHR with the graphical Terminal program [CuteCom](#). Download is at <http://cutecom.sourceforge.net>.

To communicate with an iseg USB or serial device, the following settings are needed:

Device /dev/ttyUSB0 (or other interface, according to dmesg output)
Baud rate: 9600
Data bits:8
Stop bits:1
Parity: None
Handshake: None
Line end: CR,LF

Now the serial interface can be opened by „Open device“ to test the communication:

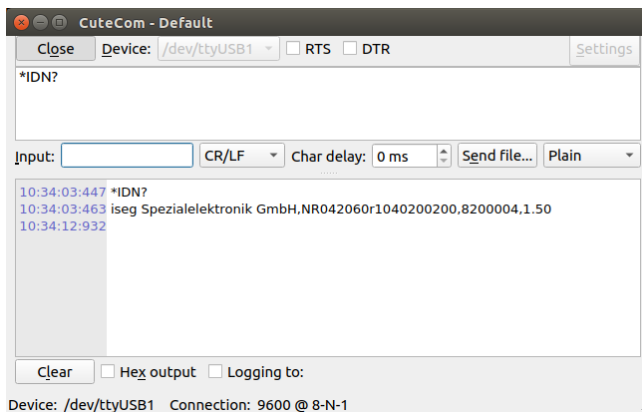


Figure 2: Linux Terminal CuteCom

3.2. Ethernet

The Ethernet interface can be used to send SCPI commands through raw TCP sockets. On device side, the TCP port 10001 is used.

3.2.1 Hardware Implementations

Depending on the device, different Ethernet implementations are utilized. Though the hardware is different, all devices follow the same programming model.

Device	Ethernet implementation
SHR	Standard Ethernet hardware and software implementation based on Linux network stack
MICC	Lantronix XPort Ethernet hardware with serial protocol based software stack in the microcontroller
HPS compact	
PS ≥ 1.5 kW	
FPS	
HPS 300/800 W	Vendor specific Ethernet hardware and software stack in the microcontroller

Table 8: Ethernet hardware implementations

3.2.2 Connecting over Ethernet

The device can be connected directly to a switch with a patch cable. If it shall be connected to a PC directly (without switch), a crossover cable has to be used.

The connection to the device can be tested with the ping command (Start → Programs → Accessories → Command):

```
C:\>ping 192.168.16.221 -h 4
Ping will done for 192.168.16.221 with 32 bytes data:
Answer from 192.168.16.221: bytes=32 time=4ms TTL=128
Answer from 192.168.16.221: bytes=32 time=4ms TTL=128
Answer from 192.168.16.221: bytes=32 time=4ms TTL=128
Answer from 192.168.16.221: bytes=32 time=4ms TTL=128
Ping statistic for 192.168.16.221:
Package: sent = 4, received = 4, lost = 0
Time in millisecond:
minimum = 1ms, maximum = 4ms, average = 1ms
```

The factory IP settings are given in the following table:

The factory IP settings		
IP address	HPS 300/800 W	192.168.16.13
	All other devices	192.168.16.221
Netmask	255.255.255.0	
Default gateway	192.168.16.1	
Command TCP port	10001 (fixed)	

Table 9: Factory IP settings

During communication, the HV device acts as a server, the control PC acts as a client. The following table shows the principle TCP socket communication sequence between PC and HV device. The communication can be monitored with a network analyzer like [Wireshark](#), see chapter 12: References.

Step	TCP function call	TCP packet Computer → HV device	TCP packet HV device → Computer
1	connect()	SYN	
2			SYN, ACK
3		ACK	
4	send()	PSH “*IDN?\r\n”	
5	recv()		PSH “iseq Spezialelektronik GmbH[...]\r\n”
6	closesocket()	FIN, ACK	
7			FIN, ACK
8		ACK	

Table 10: TCP communication between computer and HV device

The first three packages establish a TCP connection between Computer and HV device (three way handshake). In the fourth step, the request from PC is pushed to the HV unit. The command is [ASCII coded](#) (see chapter 11: ASCII character table) in the data field of the TCP packet. The answer is also [ASCII coded](#) (see chapter 11: ASCII character table) pushed back to the PC in step 5. Here it is possible that longer answers are transmitted in multiple TCP packets. Step 6 confirms the reception of the packet and sends a FIN to terminate the connection. Step 7 and 8 are the connection termination confirmation from HV unit and Computer.

3.2.3 Ethernet programming example

A simple programming example (without error handling) for communication with the HV device over Ethernet is provided. This program was originally compiled and tested with [Microsoft Visual C++ 6.0](#) on Windows XP, but still works with recent compilers on newer Windows versions too. Make sure to adopt the device address, which is hard coded to 192.168.16.221 to your environment.

```
#include <stdio.h>
#include <winsock.h>

int main()
{
    char cmd[256] = "*IDN?\r\n";
    char ans[256] = "";

    // init sockets (Berkeley style, UNIX compatible)
    WSADATA wsadata;
    WSASStartup(2, &wsadata);

    // create TCP socket
    SOCKET sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

    // bind socket to dynamic local port
    SOCKADDR_IN sockaddr_in;
    memset(&sockaddr_in, 0, sizeof(sockaddr_in));

    sockaddr_in.sin_family = AF_INET; // UDP, TCP
    sockaddr_in.sin_port = htons(10001); // Remote port
    sockaddr_in.sin_addr.S_un.S_un_b.s_b1 = 192; // IP address
    sockaddr_in.sin_addr.S_un.S_un_b.s_b2 = 168; // (adjust to your
    sockaddr_in.sin_addr.S_un.S_un_b.s_b3 = 16; // environment)
    sockaddr_in.sin_addr.S_un.S_un_b.s_b4 = 221;

    // connect to device (three way handshake)
    connect(sock, (SOCKADDR *)&sockaddr_in, sizeof(SOCKADDR_IN));

    // send command to device
    send(sock, cmd, (int)strlen(cmd), 0);

    // read answer from device until <CR><LF> is found
    for (;;) {
        char buf[256] = "";
        int length = recv(sock, buf, sizeof(buf), 0);
        if (length <= 0)
            break;

        buf[length] = 0;
        strncat(ans, buf, sizeof(ans) - strlen(ans) - 1);

        char *crlf = strstr(ans, "\r\n");
        if (crlf != NULL) {
            *crlf = 0;
            break;
        }
    }

    // close socket (three way handshake) and clean up
    closesocket(sock);
    WSACleanup();

    // print answer
    printf("%s\n", ans);
    return 0;
}
```

4 Common instruction set

Command	Description
*IDN?	Query the module identification Example: *IDN? → iseg Spezialelektronik GmbH,NR042060r4050000200,8200002,1.12
*CLS	Clear the Module Event Status and all Channel Event Status registers Example: *CLS
*RST	Reset the device to save values: <ul style="list-style-type: none"> • turn high voltage off with ramp for all channel • set voltage set V_{set} to zero for all channels • set current set I_{set} to the current nominal for all channels Example: *RST
*INSTR?	Query the currently selected instruction set. All devices support the EDCP command set. Some devices (HPS, EHQ) support further command sets, refer to the devices manual for them. Example: *INSTR? → EDCP
*INSTR,EDCP	Switch the device to the EDCP command set. Only for devices that support other command sets beside EDCP. For HPS and EHQ with other command sets, refer to the devices manual. This setting is permanent. Example: *INSTR,EDCP
*LLO	Local Lockout: Front panel buttons and rotary encoders are disabled. The device can only be controlled remotely. Example: *LLO
*GTL	Goto Local: Front panel buttons and rotary encoders are enabled. Example: *GTL
*OPC?	Query the operation complete status. The query returns "1" when all commands before this query have been processed. Example: :VOLT:ON,(@0);*OPC?

Table 11: Common instruction set

5 NHR, SHR, NHS and MICC

This manual uses the notation NHR or NHS if there is a special option for NHR or NHS and uses the notation NHR/S for common things. Commands that apply to MICC only are marked accordingly too.

5.1. SCPI Instruction Set

For the list of common commands, see section 4: Common instruction set.

Command, for channels	Description
:VOLTage	
<Voltage>[V],(@<Channel>)	Set the channel voltage set V_{set} in Volt MICC: If the channel is configured with EPU, the voltage sign defines the polarity of the output voltage. Example: :VOLT_1234,(@0)
_ON ,(@<Channel>)	Switch on High Voltage with the configured ramp speed Example: :VOLT_ON,(@1)
_OFF ,(@<Channel>)	Switch off High Voltage with the configured ramp speed Example: :VOLT_OFF,(@1)
_EMCY_OFF ,(@<Channel>)	Shut down the channel High Voltage (without ramp). The channel stays in Emergency Off until the command EMCY_CLR is given. Example: :VOLT_EMCY_OFF,(@2)
_EMCY_CLR ,(@<Channel>)	Clear the channel from state emergency off. The channel goes to state off. Example: :VOLT_EMCY_CLR,(@2)
:BOU nds_[V],(@<Channel>)	Set the channel voltage bounds V_{bounds} in Volt Example: :VOLT:BOUNDS_10,(@3)
:CURRent	
<Current>[A],(@<Channel>)	Set the channel current set I_{set} in Ampere Example: :CURR_1E-3,(@1)
:BOU nds_<Current>[A],(@<Channel>)	Set the channel current bounds I_{bounds} in Ampere Example: :CURR:BOUNDS_100E-6,(@3)
:EVent	
_CLEAR ,(@<Channel>)	Clear the Channel Event Status register (8.3). Example: :EVENT_CLEAR,(@4)
:MASK_<Word> ,(@<Channel>)	Set the Channel Event Mask register (8.4). Example: :EVENT:MASK_65535,(@4)

Command, for channels	Description
:CONFigure	Set/get channel configuration
:TRIP ^(1,4)	Configures the Delayed Trip function.
:TIME_<Time> [ms],(@<Channel>)	Set the trip timeout with one millisecond resolution (1..4095 ms) Example: :CONF:TRIP:TIME_250,(@0)
:TIME?_ (@<Channel>)	Query the programmed trip timeout in milliseconds Example: :CONF:TRIP:TIME?_(@0)
:ACTion_<Action> ,(@<Channel>)	Set or query the action that should happen when a current trip for the channel occurs.
:ACTion?_ (@<Channel>)	Action: 0 – no action, status flag Trip will be set after timeout 1 – turn off the channel with ramp 2 – shut down the channel without ramp 3 – shut down the whole module without ramp 4 – disable the Delayed Trip function Example: :CONF:TRIP:ACTION_2,(@0)
:INHibit ^(1,4)	Configures the External Inhibit function
:ACTion_<Action>	Set or query the action that should happen when an External Inhibit for the channel occurs.
:ACTion?_ (@<Channel>)	Action: 0 – no action, status flag External Inhibit will be set 1 – turn off the channel with ramp 2 – shut down the channel without ramp 3 – shut down the whole module without ramp 4 – disable the External Inhibit function Example: :CONF:INHP:ACTION_3,(@0)
:OUTPut ⁽¹⁾	Configures the output mode and polarity.
:MODE_ {1 2 3},(@<Channel>)	Set the channel output mode. Only values that are contained in output mode list are allowed. Example: :CONF:OUTPUT:MODE_3,(@0)
:MODE?_ (@<Channel>)	Query the configured channel output mode. Example: :CONF:OUTPUT:MODE?_(@0) → 3
:LIST?_ (@<Channel>)	Query the available channel output modes as list. Example: :CONF:OUTPUT:MODE:LIST?_(@0) → 1,2,3
:POLarity_ {p n},(@<Channel>)	Set output polarity (positive = “p”, negative = “n”) Example: :CONF:OUTPUT:POL_n,(@0)
:POLarity?_ (@<Channel>)	Query the current output polarity. Example: :CONF:OUTPUT:POL?_(@0) → n
:LIST?_ (@<Channel>)	Query the available channel output polarities. Example: :CONF:OUTPUT:POL:LIST?_(@0) → p, n

Command, for channels	Description
:READ	Query channel data
:VOLTage?_(@<Channel>)	Query the voltage set V_{set} in Volt Example: :READ:VOLT?_(@1) → 1.23400E3V
:LIMit?_(@<Channel>) [Ⓐ]	Query the voltage limit V_{lim} in Volt Example: :READ:VOLT:LIM?_(@1) → 3.00000E3V
:NOMinal?_(@<Channel>)	Query the channel voltage nominal V_{nom} in Volt Example: :READ:VOLT:NOM?_(@1) → 6.00000E3V
:MODE?_(@<Channel>) [Ⓐ]	Query the configured channel voltage mode with polarity sign in Volt Example: :READ:VOLT:MODE?_(@1) → 6.0E3V
:LIST?_(@<Channel>) [Ⓐ]	Query the available channel voltage modes as list which corresponds to the request :CONF:OUTP:MODE:LIST? Example: :READ:VOLT:MODE:LIST?_(@1) → 2.0E3V,4.0E3V,6.0E3V
:BOUnds?_(@<Channel>)	Query the channel voltage bounds V_{bounds} in Volt Example: :READ:VOLT:BOUNDS?_(@1) → 0.00000E3V
:ON?_(@<Channel>)	Query the channel control bit Set On Example: :READ:VOLT:ON?_(@1) → 1
:EMCY?_(@<Channel>)	Query the channel control bit Set Emergency Off Example: :READ:VOLT:EMCY?_(@1) → 0
:CURRent?_(@<Channel>)	Query the current set I_{set} in Ampere Example: :READ:CURR?_(@1) → 50.000E-6A
:LIMit?_(@<Channel>) [Ⓐ]	Query the current limit I_{lim} in Ampere Example: :READ:CURR:LIM?_(@1) → 5.00000E-3A
:NOMinal?_(@<Channel>)	Query the channel current nominal in Ampere, answer is absolute value Example: :READ:CURR:NOM?_(@1) → 6.00000E-3A
:MODE?_(@<Channel>) [Ⓐ]	Query the configured channel current mode in Ampere Example: :READ:CURR:MODE?_(@1) → 2.00000E-3A
:LIST?_(@<Channel>) [Ⓐ]	Query the available channel current modes as list which corresponds to the request :CONF:OUTP:MODE:LIST? Example: :READ:CURR:MODE:LIST?_(@1) → 6.0E-3A,4.0E-3A,2.0E-3A
:BOUnds?_(@<Channel>)	Query the channel current bounds I_{bounds} in Ampere Example: :READ:CURR:BOUNDS?_(@1) → 0.00000E-3A
:RAMP	
:VOLTage?_(@<Channel>) [Ⓐ]	Query the channel voltage ramp speed in Volt/second Example: :READ:RAMP:VOLT?_(@1) → 0.25000E3V/s
:MIN?_(@<Channel>) [Ⓐ]	Query channel voltage ramp speed minimum in Volt/second Example: :READ:RAMP:VOLT:MIN?_(@1) → 0.00005E3V/s
:MAX?_(@<Channel>) [Ⓐ]	Query channel voltage ramp speed maximum in Volt/second Example: :READ:RAMP:VOLT:MAX?_(@1) → 1.20000E3V/s
:CURRent?_(@<Channel>) [Ⓐ]	Query channel current ramp speed in Ampere/second Example: :READ:RAMP:CURR?_(@1) → 2.0000E-3A/s
:MIN?_(@<Channel>) [Ⓐ]	Query channel current ramp speed minimum in Ampere/second Example: :READ:RAMP:CURR:MIN?_(@1) → 1.0000E-3A/s
:MAX?_(@<Channel>) [Ⓐ]	Query channel current ramp speed maximum in Ampere/second Example: :READ:RAMP:CURR:MAX?_(@1) → 6.0000E-3A/s

Command, for channels	Description
:READ	Query channel data
:CHANnel :CONtrol?_(@<Channel>) :STATus?_(@<Channel>) :EVENt :STATus?_(@<Channel>) :MASK?_(@<Channel>)	Query the Channel Control register (8.5) Example: :READ:CHAN:CONTROL?_(@1) → 8 Query the Channel Status register (8.2). Example: :READ:CHAN:STATUS?_(@1) → 132 Query the Channel Event Status register (8.3). Example: :READ:CHAN:EVENT:STATUS?_(@1) → 128 Query the Channel Event Mask register (8.4). Example: :READ:CHAN:EVENT:MASK?_(@1) → 0
:MEASure	
:VOLTage?_(@<Channel>) :CURRent?_(@<Channel>)	Query the measured channel voltage in Volt. Example: :MEAS:VOLT?_(@1) → 1.23456E3V Query the measured channel current in Ampere. Example: :MEAS:CURR?_(@1) → 1.23456E-3A
:CONFigure	
:RAMP :VOLTage_<RampSpeed>[%/s] :UP_<RampSpeed>[V/s]_(@<Channel>)¹ :UP?_(@<Channel>)¹ :DOWN_<RampSpeed>[V/s]_(@<Channel>)¹ :DOWN?_(@<Channel>)¹ :CURRent_<RampSpeed>[%/s] :UP_<RampSpeed>[A/s]_(@<Channel>)¹ :UP?_(@<Channel>)¹ :DOWN_<RampSpeed>[A/s]_(@<Channel>)¹ :DOWN?_(@<Channel>)¹	Set the module or channel ramp speeds. Set the module voltage ramp speed in percent/second. Example: :CONF:RAMP:VOLT_20 Set the channel voltage ramp up speed in Volt/second. Example: :CONF:RAMP:VOLT:UP_250,(@2) Query the channel voltage ramp up speed in Volt/second. Example: :CONF:RAMP:VOLT:UP?_(@2) → 0.25000E3V/s Set the channel voltage ramp down speed in Volt/second. Example: :CONF:RAMP:VOLT:DOWN_125,(@2) Query the channel voltage ramp down speed in Volt/second. Example: :CONF:RAMP:VOLT:DOWN?_(@2) → 0.12500E3V/s Set the module current ramp speed in percent/second. Example: :CONF:RAMP:CURR_50 Set the channel current ramp up speed in Ampere/second. Example: :CONF:RAMP:CURR:UP_2E-3,(@2) Query the channel current ramp up speed in Ampere/second. Example: :CONF:RAMP:CURR:UP?_(@2) → 2.00000E-3A/s Set the channel current ramp down speed in Ampere/second. Example: :CONF:RAMP:CURR:DOWN_1.5E-3,(@2) Query the channel current ramp down speed in Ampere/second. Example: :CONF:RAMP:CURR:DOWN?_(@2) → 1.50000E-3A/s

Command, for module	Description
:CONFigure	
:AVERage _{1 16 64 256 512 1024}	Set the number of digital filter averaging steps. Factory default is 64. Example: :CONF:AVER_16
:AVERage?	Query the digital filter averaging steps. Example: :CONF:AVER? → 16
:KILL _{1 0}	Set function kill enable (1) or kill disable (0). Factory default is Kill Disable. Example: :CONF:KILL_0
:KILL?	Query the current value for the kill enable function Example: :CONF:KILL? → 0
:ADJust _{1 0}	Set the fine adjustment function on (1) or off (0) Factory default is Fine adjustment on. Example: :CONF:ADJ_1
:ADJust?	Query the fine adjustment state Example: :CONF:ADJ? → 1
:EVeNt	
_CLEAR	Reset the Module Event Status register (8.7). Example: :CONF:EVENT_CLEAR
:MASK _{<Word>}	Set the Module Event Mask register (8.8). Example: :CONF:EVENT:MASK_0
:MASK?	Query the Module Event Mask register (8.8). Example: :CONF:EVENT:MASK? → 0
:CHANMASK _{<Word>}	Set the Module Event Channel Mask register (8.11). Example: :CONF:EVENT:CHANMASK_255
:CHANMASK?	Query the Module Event Channel Mask register (8.11). Example: :CONF:EVENT:CHANMASK? → 255
:CAN	
:ADDRess _{0...63}	Set the modules CAN bus address (0...63). Can only be set in configuration mode. Example: :CONF:CAN:ADDR_23
:ADDRess?	Query the modules CAN bus address Example: :CONF:CAN:ADDR? → 23
:BITRate _{125000 250000}	Set the CAN bus bit rate to 125 kBit/s or 250 kBit/s. Can only be set in configuration mode. Example: :CONF:CAN:BITRATE_250000
:BITRate?	Query the modules CAN bus bit rate Example: :CONF:CAN:BITRATE? → 250000
:SERIAL	
:ECHO?	Query if there is serial echo is enabled (1) or disabled (0). Example: :CONF:SERIAL:ECHO? → 1
:ECHO _{1 0}	1: The device echos all characters received on the serial interface (factory default) 0: The device does not echo received characters on the serial interface. Be careful when switching off the echo as there is no other possibility to synchronize the HV device with the computer (no hardware/software handshake). This mode is only available for compatibility reasons and without support.

Command, for module	Description
:READ	Query module data
:VOLTage	
:LIMit?	Query the module voltage limit in percent Example: :READ:VOLT:LIM? → 102.0%
:CURRent	
:LIMit?	Query the module current limit in percent Example: :READ:CURR:LIM? → 80.0%
:RAMP	
:VOLTage?	Query the module voltage ramp speed in percent/second Example: :READ:RAMP:VOLT? → 20.0%/s
:CURRent?	Query the module voltage ramp speed in percent/second Example: :READ:RAMP:CURR? → 100.0%/s
:MODule	
:CONTrol?	Query the Module Control register (8.9). Example: :READ:MODULE:CONTROL? → 6144
:STATus?	Query the Module Status register (8.6). Example: :READ:MODULE:STATUS? → 30464
:EVent	
:STATus?	Query the Module Event Status register (8.7). Example: :READ:MODULE:EVENT:STATUS? → 1024
:MASK?	Query the Module Event Mask register (8.8). Example: :READ:MODULE:EVENT:MASK? → 1024
:CHANSTATus?	Query the Module Event Channel Status register (8.10). Example: :READ:MODULE:EVENT:CHANSTAT? → 7
:CHANMASK?	Query the Module Event Channel Mask register (8.11). Example: :READ:MODULE:EVENT:CHANMASK? → 0
:SUPply	Query the module supplies.
:P24V?	Query the module supply voltage +24 Volt. Example: :READ:MODULE:SUPPLY:P24V? → 24.0V
:N24V?	Query the module supply voltage -24 Volt. Example: :READ:MODULE:SUPPLY:N24V? → -23.9V
:P5V?	Query the module supply voltage +5 Volt. Example: :READ:MODULE:SUPPLY:P5V? → 5.0V
:P3V?	Query the module internal supply voltage +3.3 Volt. Example: :READ:MODULE:SUPPLY:P3V? → 3.3V
:P12V?	Query the module internal supply voltage +12 Volt. Example: :READ:MODULE:SUPPLY:P12V? → 11.9V
:N12V?	Query the module internal supply voltage -12 Volt. Example: :READ:MODULE:SUPPLY:N12V? → -12.1V
:TEMPerature?	Query the module temperature in degree Celsius. Example: :READ:MODULE:TEMPERATURE? → 31.9C
:CHANnelnumber?	Query the number of channels in the module. Example: :READ:MODULE:CHANNELNUMBER? → 4

Command, for module	Description
:READ	Query module data
:FIRMware NAME?	Query the modules firmware name Example: :READ:FIRMWARE:NAME? → N06C2
RELEASE?	Query the firmware release version (this is also part of the answer of the *IDN? command) Example: :READ:FIRMWARE:RELEASE? → 1.23
:SYSstem	
:USER	
:CONFig_<SerialNumber> ¹	Set the device to configuration mode to change the CAN bitrate or address. Only possible if all channels are off. As parameter, the device serial number must be given. For MICC, this also switches the device in a mode where user calibration is possible. Example: :SYSTEM:USER:CONFIG_1234567
:CONFig_0	Set the device back to normal mode. Example: :SYSTEM:USER:CONFIG_0
:CONFig?	Returns 1 in configuration mode, otherwise 0. Example: :SYSTEM:USER:CONFIG? → 1
_SAVE ²	SHR: Saves the changed output mode or polarity to icsConfig.xml. Example: :SYSTEM:USER:CONFIG_SAVE
Notes: ¹ Instruction for NHR or SHR only ² Instruction for SHR only ³ Instruction for MICC only ⁴ Instruction for NHS	

Table 12: SCPI commands for NHR, SHR, NHS and MICC

MICC USER CALIBRATION CONFIRMATION

After each power on, it's the users task to check the user calibration and to confirm the correct calibration with the :CONFIGURE:HVMICC_HV_OK command. Otherwise, the channel can not be turned on with :VOLT_ON.

Command, for module	Description
:CONFfigure	MICC user calibration confirmation commands
:HVMICC?	Query the current state of the user calibration confirmation. Answer HV_OK or HV_NOT_OK Example: :CONF:HVMICC? → HV_OK
:HVMICC	
_HV_NOT_OK	The HV configuration of the MICC system is not confirmed. It is <i>not</i> possible to switch on any HV channel. The system can be set to configuration mode to change the nominal values of the installed HV modules. Example: :CONF:HVMICC_HV_NOT_OK
_HV_OK	The MICC system is properly configured and the configuration is confirmed by this command. It is possible to switch on any of the HV channels afterwards. Example: :CONF:HVMICC_HV_OK

Table 13: SCPI commands for MICC user calibration confirmation

MICC USER CALIBRATION (MICC FIRMWARE RELEASE 4.XX)

Before the modules in a MICC system can be used, a user calibration is needed. For systems completely delivered from factory, this is already done. If the system is changed afterwards, the calibration may need to be updated.

iseq provides a tool, [iseq Micc Config](#) (see 12 References) for that purpose and strongly recommends using this tool. For reference, the used SCPI commands are noted in the table below.

Commands	Description
:SYS tem	MICC user calibration commands
:USER	
:CONFI g_<SerialNumber>	Switch the MICC in configuration mode to allow modifying the user calibration. To switch back from user calibration mode, any other number can be specified, e.g. 0. Example: :SYS:USER:CONFIG_1234567
:CONFI g?	Returns 1 if the MICC is in user calibration mode, 0 otherwise Example: :SYS:USER:CONFIG? → 1
:VOLT age	
:NOM inal_<Nominal>,@<Channel>	Configures the hardware nominal voltage $V_{nom,hw}$ for the channel in Volt. This value has to match the modules hardware nominal voltage. Note: This command is only accepted in configuration mode. Example: :SYS:USER:VOLT:NOMINAL_4000,@0
:NOM inal?_(@<Channel>)	Returns the channels hardware nominal voltage $V_{nom,hw}$ in Volt Example: :SYS:USER:VOLT:NOMINAL?_(@0) → 4000V
:LIMI t_<Limit>,@<Channel>	Configures a software limit for the channel in Volt. This limit must be in the range $0 \dots V_{nom,hw}$ with a default of $V_{nom,hw}$. In normal operation, this value appears as the nominal voltage V_{nom} . Note: This command is only accepted in configuration mode. Example: :SYS:USER:VOLT:LIMIT_3500,@0
:LIMI t?_(@<Channel>)	Returns the channels software voltage limit in Volt Example: :SYS:USER:VOLT:LIMIT?_(@0) → 3500V
:CURR ent	
:NOM inal_<Nominal>,@<Channel>	Configures the nominal current $I_{nom,hw}$ for the channel in Ampere. This value has to match the modules hardware nominal current. Note: This command is only accepted in configuration mode. Example: :SYS:USER:CURR:NOMINAL_(@0) → 3E-3A
:NOM inal?_(@<Channel>)	Returns the channels hardware nominal current $I_{nom,hw}$ in Ampere Example: :SYS:USER:CURR:NOMINAL?_(@0) → 3E-3A
:LIMI t_<Limit>,@<Channel>	Configures a software limit for the channel in Ampere. This limit must be in the range $0 \dots I_{nom,hw}$ with a default of $I_{nom,hw}$. In normal operation, this value appears as the nominal current I_{nom} . Note: This command is only accepted in configuration mode. Example: :SYS:USER:CURR:LIMIT_2.5E-3,@0
:LIMI t?_(@<Channel>)	Returns the channels nominal software limit in Ampere Example: :SYS:USER:CURR:LIMIT?_(@0) → 2.5E-3A

Commands	Description
:SYS tem	MICC user calibration commands
:USER	
:EPU _{0 1},(@<Channel>)	Configures the channel as EPU (electronic polarity switch), (1) or not (0). Channels with EPU can be polarity switched by the sign of V_{set} Note: This command is only accepted in configuration mode. Example: :SYS:USER:EPU_1,(@0)
:EPU? _(@<Channel>)	Returns 1 if the channel is configured as EPU (electronic polarity switch) or 0 otherwise Example: :SYS:USER:EPU?_(@0) → 1
:ACT ive_{0 1},(@<Channel>)	Configures if the user calibration is active (1) or inactive (0). Note: This command is only accepted in configuration mode. Example: :SYS:USER:ACTIVE_1,(@0)
:ACT ive?_(@<Channel>)	Returns 1 if the user calibration is active, otherwise 0 Example: :SYS:USER:ACTIVE?_(@0) → 1
:SER IAL_<Number>,(@<Channel>)	Configures the serial number of the module assigned to Channel. This is mainly thought for user information. Note: This command is only accepted in configuration mode. Example: :SYS:USER:SERIAL_2345678,(@0)
:SER IAL?_(@<Channel>)	Returns the serial number of the module assigned to Channel Example: :SYS:USER:SERIAL?_(@0) → 2345678
:ID ENTification?_(@<Channel>)	Returns the Chip-ID of the module assigned to Channel Example: :SYS:USER:IDENT?_(@0) → 123456ABCDEF
:CALIB	
_WRITE ,(@<Channel>)	Performs the user calibration and stores all configured and calculated values permanently in the devices flash memory. Note: This command is only accepted in configuration mode. Example: :SYS:USER:CALIB_WRITE,(@0)
_LOAD ,(@<Channel>)	Read back the stored configuration, discarding all changed values since the last WRITE operation. Note: This command is only accepted in configuration mode. Example: :SYS:USER:CALIB_LOAD,(@0)

Table 14: SCPI commands for MICC user calibration

5.2. Hardware Options

5.2.1 Option VCT

The following commands are only available for devices with option VCT (Voltage controlled by temperature).

Command	Description
:CONF igure	
:VCT :COEFF icient_k,(@<Channel>) :COEFF icient?_(@<Channel>)	Set a new VCT coefficient in Volt/Kelvin Example: :CONF:VCT:COEFF_1.234,(@0) Query the VCT coefficient in Volt/Kelvin Example: :CONF:VCT:COEFF?_(@0) → 1.234V/K
:READ	
:CHAN nel :TEMP erature?_(@<Channel>)	Reads the temperature of the channels external temperature sensor in degree Celsius Example: :READ:CHAN:TEMPERATURE?_(@0) → 25.9C

Table 15: SCPI commands for hardware option VCT

6 HPS and FPS

For the list of common commands, see section 4: Common instruction set.

6.1. SCPI Instruction Set

Command	Description
:VOLTage	
<Voltage>[V]	Set the channel voltage set V_{set} in Volt. Example: :VOLT_1234
:LIMIT_<Voltage>[V]	Set the channel voltage limit V_{lim} in Volt. Example: :VOLT:LIM_1500 Note: This limit should only be changed if needed, e.g. when setting the maximum allowed voltage for an experiment. It is stored in the processors EEPROM permanently and not thought to be changed continuously.
_ON	Switch on the High Voltage with the configured ramp speed. Example: :VOLT_ON
_OFF	Switch off the High Voltage with the configured ramp speed. Example: :VOLT_OFF
_EMCY_OFF	Shut the High Voltage emergency off (without ramp) The channel stays in Emergency Off (EMCY_OFF) until it is cleared with EMCY_CLR. Example: :VOLT_EMCY_OFF
_EMCY_CLR	Clear the "Shut channel High Voltage emergency off". The channel goes to state off. The Event Emergency Off must be cleared separately (e.g. with *CLS) Example: :VOLT_EMCY_CLR
:CURRent	
<Current>[A]	Set the channel current set I_{set} in Ampere. Example: :CURR_0.9
:LIMIT_<Current>[A]	Set the channel current limit I_{lim} in Ampere. Example: :CURR:LIM_1.5 Note: This limit should only be changed if needed, e.g. when setting the maximum allowed current for an experiment. It is stored in the processors EEPROM permanently and not thought to be changed continuously.
:EVent	
_CLEAR	Clear the Channel Event Status register (8.3). Example: :EVENT_CLEAR
:MASK_<Word>	Set the Channel Event Mask register (8.4). Example: :EVENT:MASK_12345
:MEASure	
:VOLTage?	Query the measured voltage in Volt. Example: :MEAS:VOLT? → 1.23456E3V
:CURRent?	Query the measured current in Ampere. Example: :MEAS:CURR? → 2.34567E-3A

Command	Description
:CONFigure	
:RAMP	
:VOLTage_<RampSpeed>[V/s]	Set the module voltage ramp speed in Volt/second Example: :CONF:RAMP:VOLT_500
:VOLTage_MAX	Disable software ramp for HPS 300/800 W with option CLD. The output voltage changes with maximum speed then. Example: :CONF:RAMP_MAX
:VOLTage?	Query the module voltage ramp speed in Volt/second Example: :CONF:RAMP:VOLT? → 1.23456E3V/s
:CURRent_<RampSpeed>[A/s]	Set the module current ramp speed in Ampere/second Example: :CONF:RAMP:CURR_0.7
:CURRent?	Query the module current ramp speed in Ampere/second Example: :CONF:RAMP:CURR? → 1.43256E-3A/s
:EVeNt	
_CLEAR	Reset the Module Event Status register (8.7). Example: :CONF:EVENT_CLEAR
:MASK_<Word>	Set the Module Event Mask register (8.8). Example: :CONF:EVENT:MASK_12345
:MASK?	Query the Module Event Mask register (8.8). Example: :CONF:EVENT:MASK? → 0
:KILL_{ 0 1 }	Set the function Kill enable (1) or Kill disable (0). Example: :CONF:KILL_1
:KILL?	Query the current value for the Kill enable function Example: :CONF:KILL? → 0
:AVERage_{ 1 16 64 256 }	Set the number of digital filter averaging steps. Example: :CONF:AVER_16
:AVERage?	Query the digital filter averaging steps. Example: :CONF:AVER? → 16
:ETHernet ⁴⁾	Configure the Ethernet settings for HPS 300/800 W. Note: Devices with XPort are configured in the XPort web interface.
:ADDRess_<xxx.xxx.xxx.xxx> ⁴⁾	Set the devices IP address. Example: :CONF:ETH:ADDR_192.168.16.13
:ADDRess? ⁴⁾	Query the devices IP address. Example: :CONF:ETH:ADDR? → 192.168.16.13
:NETmask_<xxx.xxx.xxx.xxx> ⁴⁾	Set the devices IP netmask. Example: :CONF:ETH:NET_255.255.255.0
:NETmask? ⁴⁾	Query the devices IP netmask. Example: :CONF:ETH:NET? → 255.255.255.0
:GATEway_<xxx.xxx.xxx.xxx> ⁴⁾	Set the devices IP gateway. Example: :CONF:ETH:GATE_192.168.16.1
:GATEway? ⁴⁾	Query the devices IP gateway. Example: :CONF:ETH:GATE? → 192.168.16.1
:MAC? ⁴⁾	Query the devices MAC address. Example: :CONF:ETH:MAC? → 01:23:45:67:89:ab

Command	Description
:CONFigure	
:GPIB	Configures the GPIB settings for devices with GPIB interface.
:ADDRESS?	Query the configured GPIB address. Example: :CONF:GPIB:ADDRESS? → 15
:ADDRESS_{ 1...30 }¹⁾	Set a new GPIB address in the range 1 to 30. Example: :CONF:GPIB:ADDR_16
:SERIAL	
:BAUDrate?	Query the devices serial baud rate. Example: :CONF:SERIAL:BAUD? → 9600
:ECHO?	Query the serial echo state on (1) or off (0). Example: :CONF:SERIAL:ECHO? → 1
:ECHO_{ 0 1 }¹⁾	Enables (1) or disables (0) the serial echo. Note: disabling the echo is not recommended and without support. Example: :CONF:SERIAL:ECHO_1
:CAN	
:ADDRESS?	Query the devices CAN bus address. Example: :CONF:CAN:ADDR? → 1
:ADDRESS_{ 0...63 }¹⁾	Set the devices CAN bus address. Example: :CONF:CAN:ADDR_5
:BITrate?¹⁾	Query the devices CAN bus address. Example: :CONF:CAN:BIT? → 250000
:BITrate_{ 125000 250000 }¹⁾	Set the devices CAN bus bitrate. Example: :CONF:CAN:BIT_250000
:INTERface	
_LOCAL¹⁾	Turns off the high voltage and switches from digital interface operation to local operation. For devices without front panel controls, the device switches to analog interface (AIO). Example: :CONF:INTERFACE_LOCAL
:MODE?¹⁾	Returns 0 (default) if all interface commands turn the device into remote mode, or 1 if only set commands turn it to remote mode. Mode 1 is called Listening mode, as it allows to control the device by front panel or AIO and still monitor the measured values by SCPI. Example: :CONF:INTERFACE:MODE? → 1
:MODE_{ 0 1 }¹⁾	If set to mode 0, all SCPI commands turn the device into remote operation, in mode 1 (Listening) it only turns to remote operation after receiving a set command. Example: :CONF:INTERFACE:MODE_1
:LIST?	Query the available interfaces in the device. Example: :CONF:INTERFACE:LIST? → CAN,USB,AIO

Command	Description
:CONFigure	
:ARC	ARC Management configuration. HPS compact provides only a limited set of configuration, while HPS 2 allows full ARC management customization.
:CONtrol? ²¹	Query the ARC Management operation mode. Example: :CONF:ARC:CONTROL? → 0
:CONtrol_{0 1} ²¹	Disable ARC counting (0) or enable ARC counting (1). Example: :CONF:ARC:CONTROL_1
:NUMber? ²¹	Query the number of allowed ARCs. Example: :CONF:ARC:NUMBER? → 8
:NUMber_{1...n} ²¹	Set the number of allowed ARCs. The maximum number of allowed ARCs n is device specific. Example: :CONF:ARC:NUMBER_1
:TIME? ¹¹⁾	Query the configured ARC time, in which the allowed number of ARCs may occur. Example: :CONF:ARC:TIME? → 500E-3s
:TIME_<Time>[s] ²¹	Set the ARC-Time in which the allowed number of ARCs may occur. The range of allowed times is device specific. Example: :CONF:ARC:TIME_0.1
:WAIT? ²¹	Query the configured wait time before turning the high voltage on again after exceeding the configured number of ARCs. Example: :CONF:ARC:WAIT? → 0.07s
:WAIT_<Time>[s] ²¹	Set the wait time before turning the high voltage on again after exceeding the configured number of ARCs. The allowed range of wait time is device specific. Example: :CONF:ARC:WAIT_0.05
:RAMP? ¹¹⁾	Query the configured ARC ramp speed that is used to turn on the high voltage after exceeding the configured number of ARCs Example: :CONF:ARC:RAMP? → 2.00000E6V/s
:RAMP_<RampSpeed>[V/s]	Set the ARC ramp speed that is used to turn on the high voltage after exceeding the configured number of ARCs. The allowed range of ARC ramp speed is device specific. Example: :CONF:ARC:RAMP_2E6
:SYStem	
:USER ²¹⁾	
:CONFig_<SerialNumber>	Enable configuration mode (to set the CAN address and bitrate). SerialNumber is the devices serial number included in *IDN? Note: Enabling configuration mode is only possible if high voltage generation is turned off. Example: :SYS:USER:CONFIG_1234567
:CONFig_0	Exit configuration mode Example: :SYS:USER:CONFIG_0
:CONFig?	Returns 1 if configuration mode is enabled, otherwise 0. Example: :SYS:USER:CONFIG? → 1

Command	Description
:READ	
:VOLTage? :LIMit? :NOMinal? :CURRent? :LIMit? :NOMinal?	<p>Query the channel voltage set V_{set} value in Volt. Example: :READ:VOLT? → 1.23000E3V</p> <p>Query the channel voltage limit V_{lim} in Volt. Example: :READ:VOLT:LIM? → 1.50000E3V</p> <p>Query the channel nominal voltage V_{nom} in Volt. Example: :READ:VOLT:NOM? → 2.00000E3V</p> <p>Query the channel current set value I_{set} in Ampere. Example: :READ:CURR? → 0.50000A</p> <p>Query the channel current limit I_{lim} in Ampere. Example: :READ:CURR:LIM? → 0.80000A</p> <p>Query the channel nominal current I_{nom} in Ampere. Example: :READ:CURR:NOM? → 1.50000A</p>
:READ	
:RAMP :VOLTage? :CURRent? :MODule :STATus? :EVent :STATus? :MASK? :SUPply? :TEMPerature? :CHANnel :STATus? :EVent :STATus? :MASK?	<p>Query the voltage ramp speed in Volt/second Example: :READ:RAMP:VOLT? → 2.00000E3V/s</p> <p>Query the current ramp speed in Ampere/second Example: :READ:RAMP:CURR? → 0.50000A/s</p> <p>Query the Module Status register (8.6). Example: :READ:MOD:STAT? → 43210</p> <p>Query the Module Event Status register (8.7). Example: :READ:MOD:EVENT:STAT? → 54321</p> <p>Query the Module Event Mask register (8.8). Example: :READ:MOD:EVENT:MASK? → 65432</p> <p>Query the module supply state (1 = good, 0 = not good). Example: :READ:MOD:SUPPLY? → 1</p> <p>Query the measured module temperature in °C. Example: :READ:MOD:TEMP? → 24.6C</p> <p>Query the Channel Status register (8.2). Example: :READ:CHAN:STAT? → 12345</p> <p>Query the Channel Event Status register (8.3). Example: :READ:CHAN:EVENT:STAT? → 23456</p> <p>Query the Channel Event Mask register (8.4). Example: :READ:CHAN:EVENT:MASK? → 34567</p>

Command	Description
:READ	
:FIRMware :NAME? :RELease?	Query the device firmware name. Example: :READ:FIRMWARE:NAME? → H201C0 Query the device firmware version. Example: :READ:FIRMWARE:RELEASE? → 2.35
Notes: 1) – only for HPS ≥ 1.5 kW and HPS compact (350 W) 2) – only for HPS ≥ 1.5 kW 3) – all “USER” command only for HPS compact 350 W 4) – only for HPS 300/800 W	

Table 16: SCPI commands for HPS and FPS

6.2. ARC Management configuration for HPS ≥ 1.5 kW

The following example shows how to configure the ARC management to a full ARC recovery of 200 ms. For more information about the ARC management, refer to the devices hardware manual.

The ARC management settings are permanently stored in the processors internal flash and are therefore restored after power-on. Therefore these commands only need to be sent once.

```
:CONF:ARC:CONT_1      Activate the ARC management
:CONF:ARC:NUM_10     Allow up to 10 ARCs...
:CONF:ARC:TIME_1     ... during one second
:CONF:ARC:WAIT_100E-3 Wait time between two ARCs is 100 ms
:CONF:ARC:RAMP_1E5   Voltage ramp after ARC: 100 kV/s (10000 % of Vnom = 10 kV)
```

6.3. Controlling HFS devices

6.3.1 Introduction

High voltage devices of the device class HFS combine a power HV device of series HPS ≥ 1.5 kW with one or two integrated filament supplies. The main device is internally divided into different channels. The channel assignment is described in Table 17: Channel assignment in HFS devices.

Channel	Name	Function
0	HV0 (HV Generator)	$V_{set,HV}$, $V_{meas,HV}$, $I_{set,Total}$, $I_{meas,Total}$
1	Reserved	
2	Reserved	
3	Reserved	
4	Reserved	High side current measurement
5	Reserved	High side current measurement
6	FIL0 (Filament 0, Heating)	$V_{set,F0}$, $V_{meas,F0}$, $I_{set,F0}$, $I_{meas,F0}$
7	EMI0 (Filament 0, Emission)	$V_{set,E0}$, $V_{meas,E0}$, $I_{set,E0}$, $I_{meas,E0}$
8	FIL1 (Filament 1, Heating)	$V_{set,F1}$, $V_{meas,F1}$, $I_{set,F1}$, $I_{meas,F1}$
9	EMI1 (Filament 1, Emission)	$V_{set,E1}$, $V_{meas,E1}$, $I_{set,E1}$, $I_{meas,E1}$

Table 17: Channel assignment in HFS devices

To access the channels 6...9, the SCPI commands provide an extended syntax. They get a channel suffix containing the channel number appended, as shown below:

- Write commands: „:VOLT_500“ becomes „:VOLT_500,@0“ (Suffix appended with comma)
- Read commands: „:MEAS:VOLT?“ becomes „:MEAS:VOLT?,@0“ (Suffix appended with space)

To access channel 7, the suffix would be (@7) instead.

All channel instructions described in section 6 can be applied to the different channels. Furthermore, commands for parametering the FPS devices exists. These are described in the next chapter.

6.3.2 FPS Heating and Emission Regulation parameters

The following commands apply to both integrated FPS devices. It is currently not possible to configure them individually. Therefore, no channel suffix needs to be given here.

The instruction set for the heating channels (6 and 8) are extended with the instruction "CURRent:MINimum" and "CURRent:MAXimum". The "Filament Ok" signal outputs a high level, if the filament measures at least a current of 90 % of the minimum filament current "CURR:MIN". Furthermore, the Emission regulation will not exceed the programmed minimum and maximum filament currents.

```
:CONFigure:FILament:CURRent:MINimum?
:CONFigure:FILament :CURRent:MINimum <value>
:CONFigure:FILament :CURRent:MAXimum?
:CONFigure:FILament:CURRent:MAXimum <value>
```

Example to set the minimum current to 8 Ampere:

```
:CONF:FILEA:CURR:MIN 8
```

The PID parameter of the Emission controller can be parameterized. The value is given without unit.

```
:CONFigure:FILament:EMISSION:Proportional?
:CONFigure:FILament:EMISSION:Proportional <value>
:CONFigure:FILament:EMISSION:Integral?
:CONFigure:FILament:EMISSION:Integral <value>
:CONFigure:FILament:EMISSION:Differential?
:CONFigure:FILament:EMISSION:Differential <value>
```

Example to set the P parameter to 1.234:

```
:CONF:FILEA:EMI:P_1.234
```

The default values for the parameters of the emission control unit are:

P = 2.0, I = 0.1, D = 2.0

6.3.3 Direct Heating and Emission Regulation

For direct heating control, V_{set} and I_{set} values for channel 6 resp. 8 have to be given and then the channel has to be turned on. It is possible to give an I_{set} value smaller than the minimum filament current, but not higher than filament maximum current.

For emission control, additionally to the steps before, the I_{set} for channel 7 resp. 9 has to be set to the expected emission current. Afterwards, the emission channel has to be turned on. It will now take control over the heating channel and change the $I_{set,F}$ to control the emission current. The regulation will operate in the programmed minimum and maximum filament currents, but not exceed them. Therefore, these limits have to be chosen carefully. The minimum value must be small enough that barely emission occurs, the maximum value however must be small enough that no damage can happen.

7 Output formats for voltage and current

INFORMATION



INFORMATION

The number of displayed output digits is shown in the following table.
 The actual measurement resolution depends on the device hardware (standard/high precision), the ADC sample rate and the digital filter setting.

V_{nominal}		Output format for voltage values	
$1 \text{ V} \leq$	$V_{\text{nom}} < 10 \text{ V}$		1.23456V
$10 \text{ V} \leq$	$V_{\text{nom}} < 100 \text{ V}$		12.3456V
$100 \text{ V} \leq$	$V_{\text{nom}} < 1 \text{ kV}$		123.456V
$1 \text{ kV} \leq$	$V_{\text{nom}} < 10 \text{ kV}$		1.23456E3V
$10 \text{ kV} \leq$	$V_{\text{nom}} < 100 \text{ kV}$		12.3456E3V

Table 18: Output formats for voltage values

I_{nominal}		Output format for current values	
$10 \mu\text{A} \leq$	$I_{\text{nom}} < 100 \mu\text{A}$		12.3456E-6A
$100 \mu\text{A} \leq$	$I_{\text{nom}} < 1 \text{ mA}$		123.456E-6A
$1 \text{ mA} \leq$	$I_{\text{nom}} < 10 \text{ mA}$		1.23456E-3A
$10 \text{ mA} \leq$	$I_{\text{nom}} < 100 \text{ mA}$		12.3456E-3A
$100 \text{ mA} \leq$	$I_{\text{nom}} < 1 \text{ A}$		123.456E-3A
$1 \text{ A} \leq$	$I_{\text{nom}} < 10 \text{ A}$		1.23456EA
$10 \text{ A} \leq$	$I_{\text{nom}} < 100 \text{ A}$		12.3456EA

Table 19: Output formats for current values

For high precision NHR and SHR (N04C2 or N06C2), there are some differences:

- The voltage format has one digit more precision
- The current format is the same, but a second current measurement range with the corresponding exponent exists. Depending on the range, the exponent is E-3 or E-6.

7.1. Examples

7.1.1 Simple examples

Set channel 2 voltage to 1000.501V	:VOLT_1000.501,(@2)
Set channel 2 current to 1.58mA	:CURR_0.00158,(@2)
Set voltage ramp speed to 10 % of V_{nom} per second	:CONF:RAMP:VOLT_10
Set channel 2 on with ramp	:VOLT_ON,(@2)
Query channel 2 status word. The answer 152 indicates the following bits set: Is Constant Voltage, Is On and Is Ramping	:READ:CHAN:STAT?_(@2)
Set channel 2 off with ramp	:VOLT_OFF,(@2)

7.1.2 Advanced examples

Set voltage ramp speed for all channels to 20 percent of V_{nom} per second:	:CONF:RAMP:VOLT_20%/s
Query measured voltage and current from channel 1:	:MEAS:VOLT?_(@1);CURR?_(@1) 2.00002V;1.99973E-3A
Set voltage set value to 1000 Volt for channel 0, and 2 to 4:	:VOLT_1000V,(@0,2-4)
Query voltage set value of channel 0, and 2 to 4:	:READ:VOLT?_(@0,2-4) 1.00000E3V,1.00000E3V,1.00000E3V,1.00000E3V
Set HV on for channel 0, and 2 to 4:	:VOLT_ON,(@0,2-4)

7.1.3 Commands for NHR and SHR

Query the available output modes:	:CONF:OUTP:MODE:LIST?_(@0) 1,2,3
Query the available output polarities:	:CONF:OUTP:POL:LIST?_(@0) p,n
Query the voltage and current values corresponding to the output modes above:	:READ:VOLT:MODE:LIST?_(@0) 2.0E3V,4.0E3V,6.0E3V :READ:CURR:MODE:LIST?_(@0) 4.0E-3A,3.0E-3A,2.0E-3A
Query the currently configured output mode and polarity for channel 0:	:CONF:OUTP:MODE?_(@0);POL?_(@0) 2,n
Query the voltage and current values for the currently configured output mode:	:READ:VOLT:MODE?_(@0);:READ:CURR:MODE?_(@0) -4.0E3V,3.0E-3A

8 Description of control, status, event, and mask registers

8.1. Status and control register decoding

Status and control register requests are answered as an unsigned integer number. In this number, distinct bits are set or cleared.

For example, the request `:READ:CHAN:STAT? (@2)` could return 152, which is the sum of 128 (Bit07), 16 (Bit04), and 8 (Bit03).

These bits can now be decoded from the Channel Status table. This gives the status: Is Constant Voltage, Is On and Is Ramping.

8.2. Channel Status

Access: **read-only**

Data-Type: **UI4**

Command: **:READ:CHANnel:STATus?**

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22 ⁽¹⁾	Bit21 ⁽¹⁾	Bit20 ⁽¹⁾	Bit19 ⁽¹⁾	Bit18 ⁽¹⁾	Bit17 ⁽¹⁾	Bit16 ⁽¹⁾
Reserved	Is Voltage Bound Lower	Is Voltage Bound Upper	Is Voltage Ramp Down	Is Voltage Ramp Up	Is Current Ramp Down	Is Current Ramp Up	Is Current Ramp
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Is Voltage Limit	Is Current Limit	Is Current Trip	Is External Inhibit	Is Voltage Bounds	Is Current Bounds	Is Arc Error	Is Low Current Range
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Is Constant Voltage	Is Constant Current	Is Emergency Off	Is Voltage Ramp	Is On	Is Input Error	Is Arc	Is Positive
Notes: ⁽¹⁾ NHR and SHR only							

Table 20: Channel Status Register

The Channel Status register describes the actual status. Depending on the current status of the channel the bits will be set or reset.

The individual bits are described in detail on the following page.

Example: The Channel Status 153 read from the device is decoded as follows:

153 = 0x0099 = 0000 0000 1001 1001 (bin) = Is Constant Voltage + Is Voltage Ramp + Is On + Is Positive

Status Bit	Description when bit is set to 1
Is Voltage Bound Lower	The measured voltage is lower than the programmed lower bound
Is Voltage Bound Upper	The measured voltage is higher than the programmed upper bound
Is Voltage Ramp Down	Voltage is ramping down to a lower absolute value
Is Voltage Ramp Up	Voltage is ramping up to a higher absolute value
Is Current Ramp Down	Current is ramping down
Is Current Ramp Up	Current is ramping up
Is Current Ramp	A current ramp is running
Is Voltage Limit	The measure voltage exceeds the voltage limit V_{lim}
Is Current Limit	The measured current exceeds the current limit I_{lim}
Is Current Trip	Either voltage limit or current limit has been exceeded in Kill Enable or current measure exceeded current set in Kill Enable or Delayed Trip
Is External Inhibit	An External Inhibit is active and prevents turning on the channel
Is Voltage Bounds	The measured voltage is out of the programmed bounds V_{bounds}
Is Current Bounds	The measured current is out of the programmed bounds I_{bounds}
Is Arc Error	The high voltage has been shut down because the number of allowed electrical ARCs has been exceeded
Is Low Current Range	The current measurement value originates from the Low Current Current Range (for devices with two current ranges only)
Is Constant Voltage	Voltage control is active (evaluation is guaranteed when no ramp is running)
Is Constant Current	Current control is active (evaluation is guaranteed when no ramp is running)
Is Emergency Off	The channel is in state emergency off
Is Voltage Ramp	A voltage ramp is running
Is On	The channel is turned on
Is Input Error	An input error occurred (e.g. a parameter is out of range)
Is Arc	An electrical ARC has occurred recently
Is Positive	The channel has positive output polarity (0 is negative polarity)

Table 21: Channel Status Register bit description

8.3. Channel Event Status

Access: **read-write**
Data-Type: **UI4**
Command: **:READ:CHANnel:Event:STATus?**
:Event_CLEAR

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22 ⁽¹⁾	Bit21 ⁽¹⁾	Bit20 ⁽¹⁾	Bit19 ⁽¹⁾	Bit18 ⁽¹⁾	Bit17 ⁽¹⁾	Bit16 ⁽¹⁾
Reserved	Event Voltage Bound Lower	Event Voltage Bound Upper	Event Voltage Ramp Down	Event Voltage Ramp Up	Event Current Ramp Down	Event Current Ramp Up	Event End of Current Ramp
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Event Voltage Limit	Event Current Limit	Event Current Trip	Event External Inhibit	Event Voltage Bounds	Event Current Bounds	Event Arc Error	Reserved
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Event Constant Voltage	Event Constant Current	Event Emergency Off	Event End Of Voltage Ramp	Event On To Off	Event Input Error	Event Arc	Reserved
Notes: ⁽¹⁾ NHR and SHR only							

Table 22: Channel Event Status Register

The Channel Event Status contains the captured status. An event bit is set whenever the corresponding Channel Status bit is 1 or is changing to 1.

Different to the status bit an event bit isn't automatically reset. A reset has to be done by the user by writing an 1 to this event bit. All channel events can be cleared by the `:EV_CLEAR,(@Channel)` command. With the command `*CLS` all module and channel event registers are cleared at once.

Some event bits also perform a blocking function, which means if this Bit and it's corresponding event mask bit is set, then:

- The voltage set value can not be increased while the high voltage is on
- After the high voltage is turned off, it cannot be turned on again

After either the event bit or the corresponding mask is cleared, the blocking is disabled again.

The individual bits are described in detail on the following page.

Example: The Channel Event Status 144 read from the device is decoded as follows:

144 = 0x0090 = 0000 0000 1001 0000 (bin) = Event Constant Voltage + Event End Of Voltage Ramp

Event Bit	Blocking	Description when Bit is set to 1
Event Voltage Bound Lower		The measured voltage has been or is lower than the programmed lower bound
Event Voltage Bound Upper		The measured voltage has been or is higher than the programmed upper bound
Event Voltage Ramp Down	-	Voltage ramp down to a lower absolute value has been started
Event Voltage Ramp Up	-	Voltage ramp up to a higher absolute value has been started
Event Current Ramp Down	-	Current ramp down has been started
Event Current Ramp Up	-	Current ramp up has been started
Event End of Current Ramp	-	A running current ramp has reached its destination
Event Voltage Limit	Yes	Voltage limit V_{lim} has been or is exceeded
Event Current Limit	Yes	Current limit I_{lim} has been or is exceeded
Event Current Trip	Yes	Trip is set when V_{lim} or I_{lim} or I_{set} has been exceeded in Kill Enable or I_{set} has been exceeded in Delayed Trip
Event External Inhibit	Yes	External Inhibit has been or is active
Event Voltage Bounds		Measured voltage has been or is out of bounds
Event Current Bounds		Measured current has been or is out of bounds
Event Arc Error	Yes	The channel has been shut down because the number of allowed electrical ARCs has been exceeded.
Event Constant Voltage	-	Constant Voltage has been or is active
Event Constant Current	-	Constant Current has been or is active
Event Emergency Off	Yes	Channel was shut down with emergency off
Event End Of Voltage Ramp	-	A running voltage ramp has reached its destination
Event On To Off		The channel was switched off due to a blocking event (Voltage/Current Limit, Current Trip, External Inhibit or Emergency Off)
Event Arc	-	An electrical ARC has occurred
Event Input Error	-	An Input Error has occurred

Table 23: Channel Event Status Register bit description

8.4. Channel Event Mask

Access: **read-write**
Data-Type: **UI4**
Command: **:CONF:Event:MASK?**
:READ:CHANnel:Event:MASK?
:CONF:Event:MASK_<Mask>

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22 ⁽¹⁾	Bit21 ⁽¹⁾	Bit20 ⁽¹⁾	Bit19 ⁽¹⁾	Bit18 ⁽¹⁾	Bit17 ⁽¹⁾	Bit16 ⁽¹⁾
Reserved	Mask Event Voltage Bound Lower	Mask Event Voltage Bound Upper	Mask Event Voltage Ramp Down	Mask Event VoltageRamp Up	Mask Event Current Ramp Down	Mask Event Current Ramp Up	Mask Event End of Current Ramp
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Mask Event Voltage Limit	Mask Event Current Limit	Mask Event Trip	Mask Event External Inhibit	Mask Event Voltage Bounds	Mask Event Current Bounds	Mask Event Arc Error	Reserved
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Mask Event Constant Voltage	Mask Event Constant Current	Mask Event Emergency Off	Mask Event End Of Ramp	Mask Event On To Off	Mask Event Input Error	Mask Event Arc	Reserved
Notes: ¹⁾ SHR only							

Table 24: Channel Event Mask Register

The event mask provides a mean to propagate a specific channel event to the global Module Event Channel Status register (8.10). They also block turning on or increasing the high voltage for some event bits. See section 8.3: Channel Event Status for more information.

The individual bits are described in detail on the following page.

Example: The Channel Event Mask 144 read from the device is decoded as follows:

144 = 0x0090 = 0000 0000 1001 0000 (bin) = Mask Event Constant Voltage + Mask Event End Of Voltage Ramp

Mask Bit	Description when Bit is set to 1
Mask Event Voltage Bounds Lower	Mask the Event Voltage Bounds Lower
Mask Event Voltage Bounds Upper	Mask the Event Voltage Bounds Upper
Mask Event Voltage Ramp Down	Mask the Event Voltage Ramp Down
Mask Event Voltage Ramp Up	Mask the Event Voltage Ramp Up
Mask Event Current Limit Down	Mask the Event Current Ramp Down
Mask Event Current Ramp Up	Mask the Event Current Ramp Up
Mask Event End of Current Ramp	Mask the Event End Of Current Ramp
Mask Event Voltage Limit	Mask the Event Voltage Limit
Mask Event Current Limit	Mask the Event Current Limit
Mask Event Trip	Mask the Event Trip
Mask Event External Inhibit	Mask the Event External Inhibit
Mask Event Voltage Bounds	Mask the Event Voltage Bounds
Mask Event Current Bounds	Mask the Event Current Bounds
Mask Event Arc Error	Mask the Event Arc Error
Mask Event Constant Voltage	Mask the Event Constant Voltage
Mask Event Constant Current	Mask the Event Constant Current
Mask Event Emergency Off	Mask the Event Emergency Off
Mask Event End Of Ramp	Mask the Event End Of Voltage Ramp
Mask Event On To Off	Mask the Event On To Off
Mask Event Arc	Mask the Event Arc
Mask Event Input Error	Mask the Event Input Error

Table 25: Channel Event Mask Register bit description

8.5. Channel Control

Access: **read-write**
Data-Type: **UI4**
Command: **:READ:CHANnel:CONtrol?**
:VOLTage_{ ON | OFF | EMCY OFF | EMCY CLR }

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Reserved	Reserved	Set Emer- gency Off	Reserved	Set On	Reserved	Reserved	Reserved

Table 26: Channel Control Register

The Channel Control register contains the control information of the channel.

There is no direct write access to this register by SCPI, instead the commands :VOLT_ON/OFF and :VOLT:EMCY_OFF/CLR are used to control the channel.

The individual bits are described in detail in the following table.

Example: The Channel Control 8 read from the device is decoded as follows:

8 = 0x0008 = 0000 0000 0000 1000 (bin) = Set On

Control Bit	Description
Set On	1 if the channel is set on by :VOLT_ON or 0 if the channel is off.
Set Emergency Off	1 if the channel is set to emergency off by :VOLT:EMCY_OFF or 0 otherwise

Table 27: Channel Control Register bit description

8.6. Module Status

Access: **read-only**
Data-Type: **UI4**
Command: **:READ:MODule:STATus?**

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22	Bit21 ⁽¹⁾	Bit20	Bit19	Bit18	Bit17	Bit16 ⁽¹⁾
Reserved	Reserved	Is Voltage Ramp Speed Limited	Reserved	Reserved	Reserved	Reserved	Is Fast Ramp Down
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Is Kill Enable	Is Temperature Good	Is Supply Good	Is Module Good	Is Event Active	Is Safety Loop Good	Is No Ramp	Is No Sum Error
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Reserved	Is Input Error	Reserved	Is Service	Is High Voltage On	Reserved	Reserved	Is Fine Adjustment
Notes: ⁽¹⁾ SHR only							

Table 28: Module Status Register

The Module Status register describes the actual status. Depending on the current status of the module the bits will be set or reset.

The individual bits are described in detail on the following page.

Example: The Module Status 29440 read from the device is decoded as follows:

29440 = 0x7300 = 0111 0011 0000 0000 (bin)

= Is Temperature Good + Is Supply Good + Is Module Good + Is No Ramp + Is No Sum Error

Status Bit	Description when Bit is 1
Is Voltage Ramp Speed Limited	Voltage ramp speed limitation caused by Kill Enable or Delayed Trip is active
Is Fast Ramp Down	All channels ramp down fast to zero due to power fail
Is Kill Enable	Module is in state Kill Enable
Is Temperature Good	Module temperature is in good conditions. All channels will be turned off with ramp if the temperature becomes bad.
Is Supply Good	The external power supplies are all good (24 V \pm 10 % and 5 V \pm 5 %)
Is Module Good	Module in good state good, that means the condition Is No Sum Error AND NOT (Event Temperature Not Good OR Event Supply Not Good OR Event Safety Loop Not Good) is satisfied
Is Event Active	At least one Module Event Channel Status bit together with the corresponding Module Event Channel Mask bit is set
Is Safety Loop Good	Safety Loop is closed. If the safety loop is opened, all channels are shut down without ramp.
Is No Ramp	All channels are stable, no ramp is running.
Is No Sum Error	All channels are in good condition, that means the condition (Is Voltage Limit OR Is Current Limit OR Is Current Trip OR Is External Inhibit OR Is Voltage Bounds OR Is Current Bounds OR Is Arc Error) is zero for all channels
Is Input Error	An Input error occurred while accessing a module command or a channel outside the allowed range.
Is Service	A hardware failure of the HV module has been detected. The HV will be switched off without a possibility to switch on again. Consult the manufacturer.
Is Fine Adjustment	Fine adjustment is enabled for all channels
Is High Voltage On	At least one channel is actively generating High Voltage or has a measured voltage greater 60 Volt.

Table 29: Module Status Register bit description

8.7. Module Event Status

Access: **read-write**
Data-Type: **UI4**
Command: **:READ:MODule:Event:STATus?**
:CONFigure:EVent_CLEAR

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Reserved	Event Temperature Not Good	Event Supply Not Good	Reserved	Reserved	Event Safety Loop Not Good	Reserved	Reserved
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Reserved	Event Input Error	Reserved	Event Service	Reserved	Reserved	Reserved	Reserved

Table 30: Module Event Status Register

The Module Event Status contains the captured status. An event bit is set if the corresponding Module Status bit is 1 or is changing to 1.

Different to the status bit an event bit isn't automatically reset. A reset has to be done by the user by writing an 1 to this event bit. All channel events can be cleared by the **:CONF:EV_CLEAR** command. With the command ***CLS** all module and channel event registers are cleared at once.

Some event bits also perform a blocking function, which means if this Bit and it's corresponding event mask bit is set, then:

- The voltage set value can not be increased while the high voltage is on
- After the high voltage is turned off, it cannot be turned on again

After either the event bit or the corresponding mask is cleared, the blocking is disabled again.

The individual bits are described in detail in the following table.

Example: The Module Event Status 1024 read from the device is decoded as follows:

1024 = 0x0400 = 0000 0100 0000 0000 (bin) = Event Safety Loop Not Good

Event Bit	Blocking	Description when Bit is set to 1
Event Temperature Not Good	Yes	Temperature has been or is above it's upper limit
Event Supply Not Good	Yes	At least one of the external supplies is out of it's good range
Event Safety Loop Not Good	Yes	Safety loop has been or is open
Event Input Error	-	An Input error in connection with a module access has occurred
Event Service	Yes	A hardware failure of the HV module has been detected. The HV will be switched off without a possibility to switch on again. Consult the manufacturer.

Table 31: Module Event Status Register bit description

8.8. Module Event Mask

Access: **read-write**
Data-Type: **UI4**
Command: **:CONF:Event:MASK_<Mask>**
:CONF:Event:MASK?
:READ:MODule:EVent:MASK?

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Reserved	Mask Event Temperature Not Good	Mask Event Supply Not Good	Reserved	Reserved	Mask Event Safety Loop Not Good	Reserved	Reserved
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Reserved	Mask Event Input Error	Reserved	Mask Event Service	Reserved	Reserved	Reserved	Reserved

Table 32: Module Event Mask Register

The event mask provides a mean to propagate a specific channel event to the global event register. They also block turning on or increasing the high voltage for some event bits. See section 8.7: Module Event Status for more information.

The individual bits are described in detail in the following table.

Example: The Module Event Mask 1024 read from the device is decoded as follows:

1024 = 0x0400 = 0000 0100 0000 0000 (bin) = Mask Event Safety Loop Not Good

Mask Bit	Description
Mask Event Temperature Not Good	Mask the Event Temperature Not Good
Mask Event Supply Not Good	Mask the Event Supply Not Good
Mask Event Safety Loop Not Good	Mask the Event Safety Loop Not Good
Mask Event Input Error	Mask the Event Input Error
Mask Event Service	Mask the Event Service

Table 33: Module Event Mask Register bit description

8.9. Module Control

Access: **read-write**
Data-Type: **UI4**
Command: **:READ:MODule:CONTrol?**
:CONF:ADJust_{ 0 | 1 }
:CONF:KILL_{ 0 | 1 }
***CLS**

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16 ⁽¹⁾
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Disable Voltage Ramp Speed Limit
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Reserved	Set Kill Enable	Reserved	Set Fine Adjustment	Set Big Endian	Reserved	Reserved	Reserved
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Reserved	Do Clear	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Notes: ¹⁾ SHR Only							

Table 34: Module Control Register

The Module Control register contains the control information of the module.

There is no direct write access to this register by SCPI. Instead the commands **:CONF:ADJust**, **:CONF:KILL** and ***CLS** can be used.

The individual bits are described in detail in the following table.

Example: The Module Control 4096 read from the device is decoded as follows:

4096 = 0x1000 = 0001 0000 0000 0000 (bin) = Set Fine Adjustment

Bit	Description when Bit is set to 1
Disable Voltage Ramp Speed Limit	If set to one disables ramp speed limitation caused by Kill Enable or Delayed Trip
Set Kill Enable	Kill Enable is turned on
Set Fine Adjustment	Fine adjustment is turned on
Set Big Endian	Order of bytes in word: 0 = Little Endian (INTEL); 1 = Big Endian (MOTOROLA) This bit is always set to 1, i.e. all data is in Big Endian format.
Do Clear	Clear all event status registers of the module and the channels. This bit always reads as 0, the clear all events the command *CLS can be used.

Table 35: Module Control Register bit description

8.10. Module Event Channel Status

Access: **read-write**
Data-Type: **UI4**
Command: **:READ:MODule:EVent:CHANSTATus?**

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

Table 36: Module Event Channel Status Register

The n-th bit of the register is set, if an event is active in the n-th channel and the associated bit in the Channel Event Mask register (section 8.4) of the n-th channel is set too.

$$CH_n = \text{Channel Event Status}[n] \& \text{Channel Event Mask}[n] \quad (0 \leq n \leq 31)$$

A bit is reset by writing a 1 to this bit.

8.11. Module Event Channel Mask

Access: **read-write**
Data-Type: **UI4**
Command: **:READ:MODule:EVent:CHANMASK?**

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Mask CH31	Mask CH30	Mask CH29	Mask CH28	Mask CH27	Mask CH26	Mask CH25	Mask CH24
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
Mask CH23	Mask CH22	Mask CH21	Mask CH20	Mask CH19	Mask CH18	Mask CH17	Mask CH16
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08
Mask CH15	Mask CH14	Mask CH13	Mask CH12	Mask CH11	Mask CH10	Mask CH9	Mask CH8
Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
Mask CH7	Mask CH6	Mask CH5	Mask CH4	Mask CH3	Mask CH2	Mask CH1	Mask CH0

Table 37: Module Event Channel Mask Register

This register decides whether a pending event in the Module Event Channel Status leads to the sum event flag of the module.

If the n-th bit of the mask is set and the n-th channel has an active event in the Module Event Channel Status register, the bit Is Event Active in the Module Status register (section 8.6 Module Status) is set.

9 iseg Terminal

iseq Terminal is a Windows or Linux program to control iseg devices with RS-232, USB, GPIB (IEEE-488) or Ethernet interface using their ASCII command sets.

The latest version is available from the iseg website:

<https://iseq-hv.com/download/> → Software → ISEG Terminal → Current → "iseqTerminal-x.x.x-Windows-x86.zip"

For serial communication, choose "1 RS-232/USB" in the Interface List and select the COM-Port you connected your iseg device to. With a click on Init, the serial port is opened and you can send commands to your device by clicking the Query button (or simple pushing the Enter key).

The answer is shown in the output window.

For detailed command set descriptions, please see the documentation for the specific device.

The following command is used for device identification query: `*IDN?`

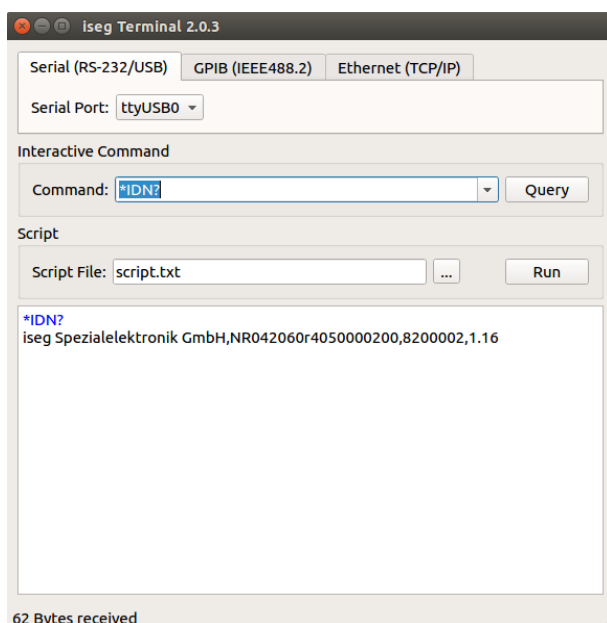


Figure 3: iseg Terminal

10 iseg SCPI Control

iseg SCPI Control is a Windows or Linux program to control iseg devices with SCPI command set by the USB or Ethernet interface. All important data points will be displayed via a table and an input of new set value can be made via short cuts. The corresponding short cuts are displayed with the text entries of the menu bar.

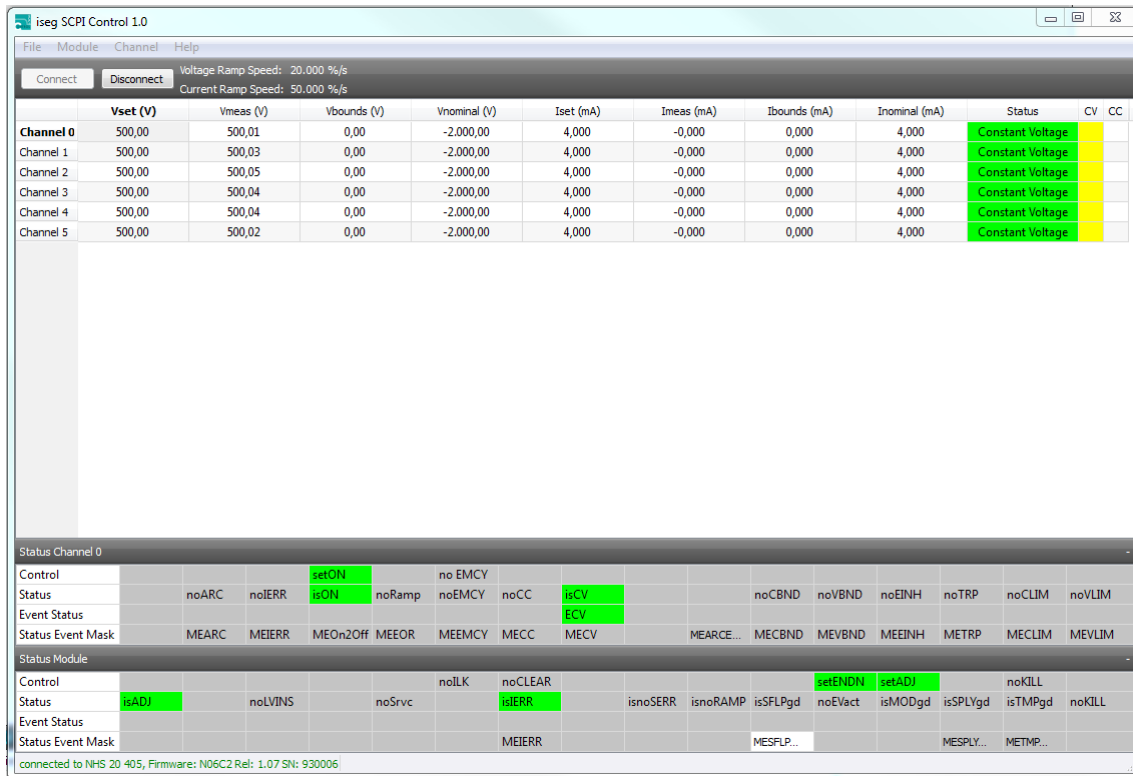


Figure 4: iseg SCPI Control

The control of iseg USB devices requires a driver installation (see 3.1.5 Windows USB driver installation.)

The connection will be established by means of the button "Connect" and a continuous request of the data points is started.

<https://iseg-hv.com/download> → SOFTWARE → isegSCPIcontrol → current → isegScpiControlSetup-win32-x.x.xx.x.exe

11 ASCII character table

Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	<NUL>	0x20	32	␣	0x40	64	@	0x60	96	`
0x01	1	<SOH>	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	<STX>	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	<ETX>	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	<EOT>	0x24	36	\$	0x44	68	D	0x64	100	d
0x05	5	<ENQ>	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	<ACK>	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	<BEL>	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	<BS>	0x28	40	(0x48	72	H	0x68	104	h
0x09	9	<HT>	0x29	41)	0x49	73	I	0x69	105	i
0x0A	10	<LF>	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	<VT>	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	<FF>	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	<CR>	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	<SO>	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	<SI>	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	<DLE>	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	<DC1>	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	<DC2>	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	<DC3>	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	<DC4>	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	<NAK>	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	<SYN>	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	<ETB>	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	<CAN>	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25		0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	<SUB>	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	<ESC>	0x3B	59	;	0x5B	91	[0x7B	123	{
0x1C	28	<FS>	0x3C	60	<	0x5C	92	\	0x7C	124	
0x1D	29	<GS>	0x3D	61	=	0x5D	93]	0x7D	125	}
0x1E	30	<RS>	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	<US>	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

12 References

For more information please use the following download links:

This document
https://iseg-hv.com/download/?dir=SOFTWARE/isegSCPI/SCPI_Common_Programmers-Guide.pdf
CAN EDCP Programmers-Guide
https://iseg-hv.com/download/SOFTWARE/isegEDCP/CAN_EDCP_Programmers-Guide.pdf
iseg Hardware Abstraction Layer
http://download.iseg-hv.com/SYSTEMS/MMS/isegHardwareAbstractionLayer.pdf
isegScpiControl
https://iseg-hv.com/download/?dir=SOFTWARE/isegSCPIcontrol/current/
FTDI-USB-Serial-Driver
https://iseg-hv.com/download/?dir=SOFTWARE/Tools/
CDC-ACM usb driver for SHR
https://iseg-hv.com/download/?dir=SOFTWARE/Tools/
https://developer.ridgerun.com/wiki/index.php/How_to_use_USB_CDC_ACM_and_MS_composite_Linux_gadget_driver
The Wireshark network analyzer
http://www.wireshark.org
CuteCom
http://cutecom.sourceforge.net
Microsoft Visual C++ 6.0
https://www.microsoft.com
iseg Micc Config
https://iseg-hv.com/download/?dir=SOFTWARE/isegMICCconfig/Win/
isegTerminal
https://iseg-hv.com/download/?dir=SOFTWARE/isegTERMINAL/current
iseg SCPI Control
https://iseg-hv.com/download/?dir=SOFTWARE/isegSCPIcontrol/current
iseg Control
https://iseg-hv.com/download/?dir=SOFTWARE/isegControl/current
Option VCT
https://iseg-hv.com The exact description can be found in the hardware manuals of the devices. For example SHR .

13 Manufacturer contact

iseg Spezialelektronik GmbH

Bautzner Landstr. 23

01454 Radeberg / OT Rossendorf

GERMANY

FON: +49 351 26996-0 | FAX: +49 351 26996-21

www.iseg-hv.com | info@iseg-hv.de | sales@iseg-hv.de