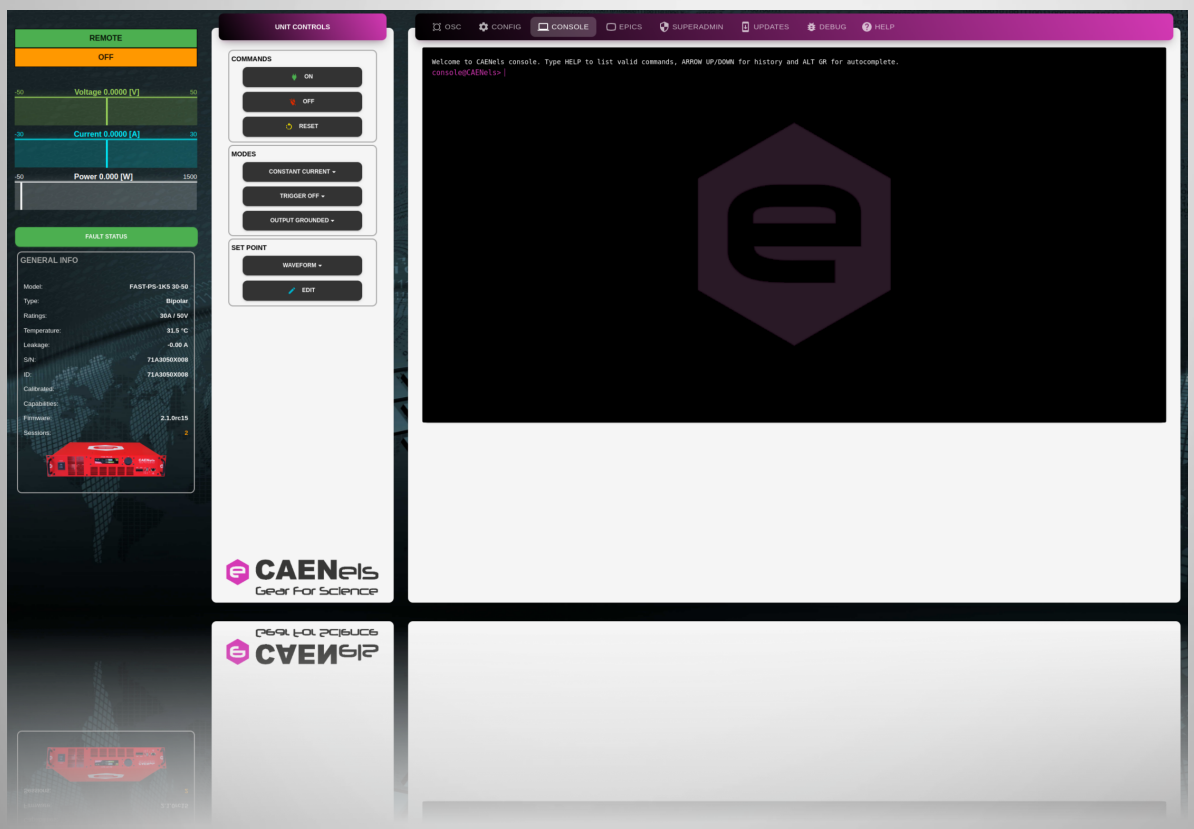
HPPS-JLAB

High-Power Power Supply Series - JLab Custom



Commands Reference Manual

CAENels
Gear For Science

All Rights Reserved
© CAEN ELS s.r.l
1.0 - November 17th, 2023
Aligned with FW: 2.1.5

CAEN ELS s.r.l
in AREA Science Park
S.S. 14 km 163,5 - 34149 Basovizza (TS)
Italy
Mail: info@caenels.com
Web: www.caenels.com

Contents

Document Revisions **4**

1 Remote interface **5**
1.1 Ethernet interface . . . 5
1.2 Command Syntax . . . 6
1.2.1 Read commands . . . 6
1.2.2 Write commands . . . 6
1.3 Command Examples . . . 8
1.4 Communication Examples . . . 9
1.4.1 Python Example . . . 9
1.4.2 Putty Example . . . 10

2 Finite State Machine FSM **11**
2.1 DC FSM . . . 12
2.2 OUT FSM . . . 14

3 Summary of Remote Commands **16**
3.1 Basic Commands . . . 17
3.2 Password Related Commands . . . 18
3.3 Read/Write Internal Memory Related Commands . . . 19
3.4 Status/Faults Register Related Commands . . . 20
3.5 Common Commands . . . 21
3.6 Limits Related Commands . . . 23
3.7 Mode Related Commands . . . 24
3.8 Interlocks Commands . . . 25
3.9 HPPS-JLab Specific Commands . . . 26
3.10 Advanced Features Commands . . . 27

4 Detailed Commands Description **28**
4.1 Basic Commands . . . 29
4.1.1 VER . . . 30
4.1.2 MRID . . . 31

4.2 Password Related Commands . . . 32
4.2.1 PASSWORD . . . 33
4.2.2 SETPSW . . . 34
4.2.3 RESETPSW . . . 35
4.3 Read/Write Internal Memory Related Commands . . . 36
4.3.1 MRG . . . 37
4.3.2 MWG . . . 38
4.3.3 MSAVE . . . 39
4.4 Faults/Status Register Related Commands . . . 40
4.4.1 MFTR . . . 41
4.4.2 MSTR . . . 42
4.4.3 MRESET . . . 43
4.5 Common Commands . . . 44
4.5.1 OUT . . . 45
4.5.2 MRI . . . 46
4.5.3 MRV . . . 47
4.5.4 MWV . . . 48
4.5.5 MWVR . . . 49
4.5.6 MSRV . . . 50
4.5.7 MWI . . . 51
4.5.8 MWIR . . . 52
4.5.9 MSRI . . . 53
4.5.10 MRW . . . 54
4.5.11 MRT . . . 55
4.5.12 UPFREQ . . . 56
4.5.13 MRP . . . 57
4.6 Limits Commands . . . 58
4.6.1 MLIMITS . . . 59
4.6.2 MPLIMITS . . . 60
4.7 Mode Related Commands . . . 61
4.7.1 LOOP . . . 62
4.7.2 UPMODE . . . 63
4.8 Interlocks Commands . . . 64
4.8.1 INTPM . . . 65
4.8.2 INTEM . . . 67
4.8.3 INTNAME . . . 69
4.8.4 INTIT . . . 70

4.9 HPPS-JLab Specific Commands 71
4.9.1 DC 72
4.9.2 MRTIGBT 73
4.9.3 MIG 74
4.9.4 MIAX 75
4.10 Advanced Features Commands 76
4.10.1 WAVE 77
4.10.2 TRIG 82

5 Internal Memory **83**

6 Faults Register **87**

7 Status Register **90**

8 Temperatures **92**

9 NAK List **93**

Document Revisions

Document Revision	Date	Requirements
1.0	November 17th, 2023	HPPS-JLab First Release

1 Remote interface

The control interface is based on ASCII commands over the TCP/IP or UDP protocol on the **port 10001**.

1.1 Ethernet interface

The device is shipped with the following factory values:

Parameter	Factory value
IP	192.168.0.10
Subnet mask	255.255.255.0
Gateway	192.168.0.1
DHCP	Disabled

The Ethernet configuration can be changed upon the customer needs.

1.2 Command Syntax

Commands are in ASCII format and are **NOT case sensitive** and therefore the command string can be sent either using uppercase or lowercase characters.
Each command is composed by one or more fields, the fields are separated by colons (":").
Each command has to be terminated with the termination sequence. Commands support two termination sequences:

- **CRLF:** "carrige return, line feed" sequence ("\r\n")
- **CR:** "carrige return" termination char ("\r")

In this documentation the commands are rapresented only in uppercase and the termination sequence is always rapresented with CRLF.

1.2.1 Read commands

The read commands are composed in the following way:

- command name (which could be followed either by one or more optional sub-commands separated by colons);
- colon (":") char;
- question mark ("?");
- termination sequence.

The reply to a read command is formed by:

- hastag ("#") ;
- command echo;
- colon (":") char;
- read value;
- termination sequence.

For example:

```
MRI:?\r\n
#MRI:1.0658\r\n
```

Some commands accept also read commands without the last colon (":") and question mark ("?").

1.2.2 Write commands

The write commands are composed in the following way:

- command name (which could be followed either by one or more optional sub-commands separated by colons);
- colon (":") char;
- write value;
- termination sequence.

The reply of the power supply to a write command can be:

- "#AK" (AcKnowledged): when the command has been **accepted**;
- "#NAK" (Not AcKnowledged): when the command is **NOT accepted**, followed by the corresponding error code and the error description.

For example:

```
LOOP:V\r\n
#AK\r\n

MWI:2\r\n
#NAK:13 Module is off\r\n
```

Note: By default the **NAK** (Not AcKnowledged) reply returns also the error description. The error description can be disabled acting on the "Error Code Description" field, which is editable using the *MWG* command (ID parameter "56", see attachment *Internal Memory*). By disabling the error description, the reply is formed by the error code only, e.g.:

```
#MWI:2\r\n
#NAK:13\r\n
```

1.3 Command Examples

- Read command:

```
VER:?\r\n
#VER:NGSP 100-50:2.0.1\r\n
```

- Write command, accepted

```
LOOP:I\r\n
#AK\r\n
```

- Write command, NOT accepted

```
MWI:20\r\n
#NAK:13\r\n
```

Note 1: in the later sections the termination sequence will not be displayed for better readability, but it must be always present otherwise the commands are not parsed.

Note 2: before sending the next command, it is necessary to wait the response of the unit to the previous one.

1.4 Communication Examples

1.4.1 Python Example

By means of Python scripts it is very simple to establish a communication with the power unit. In the following example, it is possible to see how to firstly open a socket and later sent a request of the firmware version to the power unit:

```
#!/usr/bin/env python3

import socket

IP = "192.168.0.10"

try:
    # Create sockets
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)     # TCP
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.connect((IP, 10001))

    # Set psw
    print("Get Version")
    s.sendall("VER:?\r\n".encode())       # Note: it works also with '\r'
       terminator
    data = s.recv(2048).decode()
    print(repr(data))

    # Close socket
    s.close()

except :
    print("Error: Error in communication with the module")
```

1.4.2 Putty Example

Putty it is a free telnet client that is available for both Windows and Linux OS. By means of Putty, it is possible to send commands to the power unit, by configuring the IP address of the unit, set the communication port to 10001 and finally set the connection type to “Raw”.

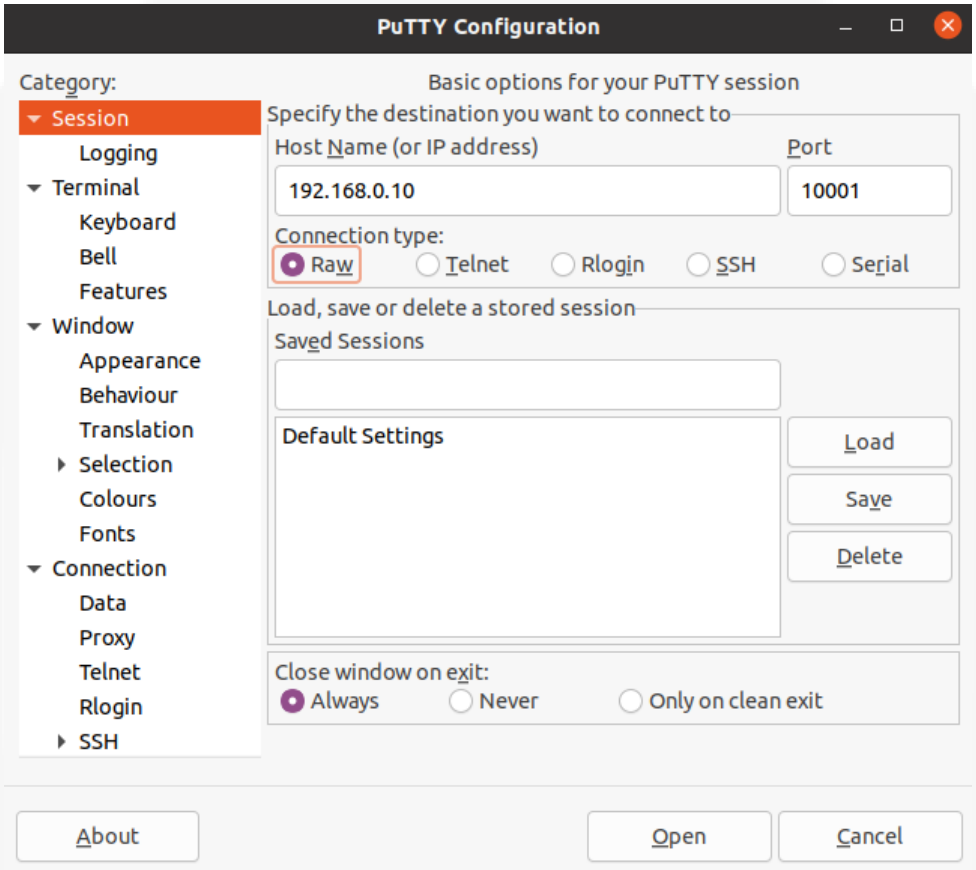


Figure 1.1: Putty Configuration

Once the configuration has been completed, it is possible to send the ASCII command directly to the unit:

Figure 1.2: Putty Example

2 Finite State Machine FSM

The behaviour of the power unit output is determinated by a *Finite State Machine*.

Following event can involve a transition:

- external command
- internal transition
- fault (*Hard Fault* or *Soft Fault*)

OUT FSM controls the power output.

The HPPS model uses an additional *DC FSM* to control the capacitor bank charging. For this family *OUT FSM* can evolved only when the capacitor bank charging.

For this family *OUT FSM* can evolved only when the capacitor bank charging.

2.1 DC FSM

The *DC FSM* is the following:

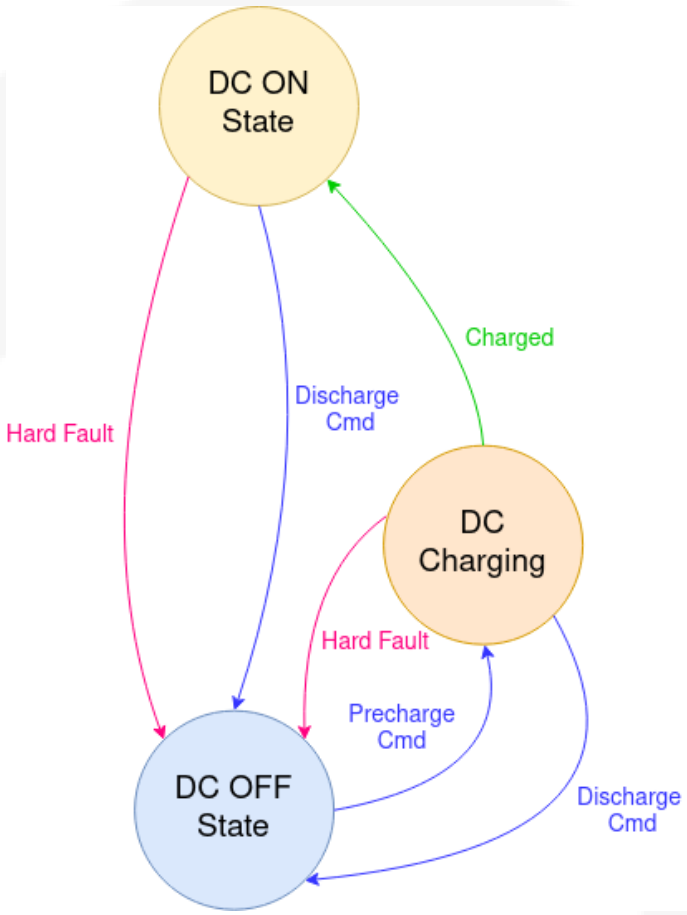


Figure 2.1: DC Link FSM

The four states are:

- **DC OFF State**: in this state the capacitor bank is discharged. This is the default state at machine startup.
- **DC Charging State**: in this state the precharge contactor is closed and the capacitor bank is in charging state. Once the capacitor banks reaches the nominal voltage value, the machine evolves to **DC ON State**.
- **DC ON State**: the capacitor bank is charged and the machine is ready to operate.

The transition between the states are the following:

- **Command-related transitions** (in blue): that are defined by sofware commands (e.g. **DC:ON** DC-link charge command or **DC:OFF** DC-link discharge command).
- **Fault-related transitions** (in red): that are driven by a **Fault condition**. When a fault occurs, the machine is in a **Fault State**.
  There are two types of *Fault condition*: *Soft Fault* and *Hard Fault*. *Soft Fault* are not critical and so the *DC FSM* ignores it. Otherwise a *Hard Fault* forces the *DC FSM* to evolve in **DC OFF State**.
- **Internal transitions** (in green): that is related to the charging state of the DC-link. This transition occurs when the capacitor bank voltage reaches the nominal value and the *DC FSM* automatically switches to the **DC ON State**.

2.2 OUT FSM

Once the capacitor bank is charged (*DC FSM* is in **DC ON State**), it is possible to evolve the *OUT FSM*. When the capacitor bank is not ready (the *DC FSM* is either in **DC Charging State** or **DC OFF State**), the *OUT FSM* remains in **OUT OFF State**.

The *OUT FSM* is the following:

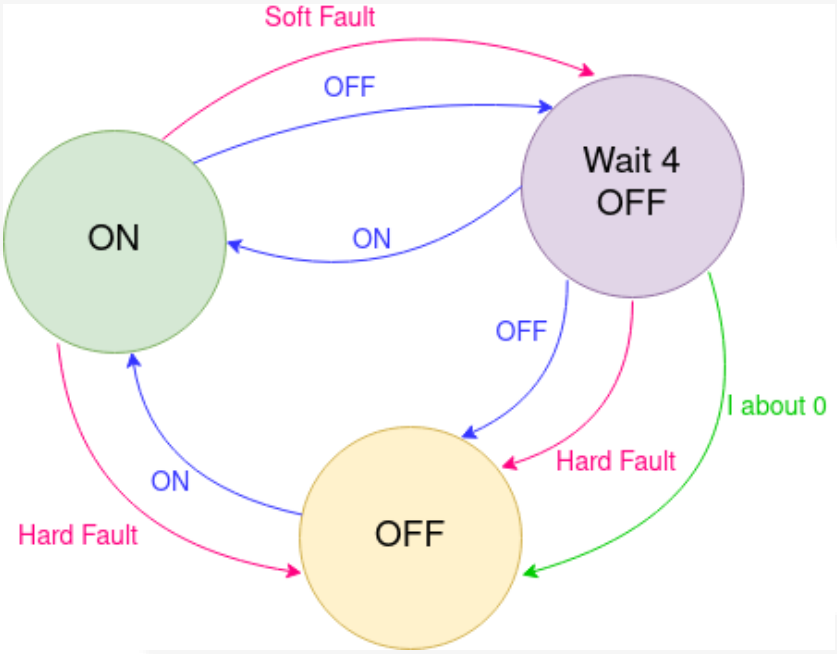


Figure 2.2: Output FSM

The three states are:

- **OUT OFF State**: in this state the output drivers and PID regualator are disabled. This is the default state at machine startup.
- **OUT Wait for Off State**: in this state the unit performs a ramp-down to 0V and waits until the output current reaches 0A. When the current is about 0A, the unit evolves to **OUT OFF State**.
- **OUT ON State**: in this state the output drivers and PID regulator are enabled and the unit can accepts a new setpoint. The default setpoint that is set, when the output is enabled is 0A in *CC Mode* or 0V in *CV Mode* (see *Mode Related Commands* chapter).

The transition between the states are the following:

- **Command-related transitions** (in blue): that are defined by sofware commands (for example by **OUT:ON** or **OUT:OFF** commands)
- **Fault-related transitions** (in red): that are defined by a **Fault condition**. When a fault occurs, the machine is in a **Fault State**.
  There are two types of *Fault condition*: *Soft Fault* and *Hard Fault*. *Soft Fault* is the one that does not impact on the machine regulation and so the *OUT FSM* goes to **OUT Wait for Off State** in which the output is set to 0V and when the current is about 0A, the module goes to **OUT OFF State**. Otherwise with a *Hard Fault*, the machine is not able to regulate the output and so the FSM evolves directly to **OUT OFF State**.
- **Internal transitions** (in green): there is only one internal transition that is related to the output current. This transition is used only when the *OUT FSM* is in **OUT Wait for Off State** and the output current is close to 0A, at this point the *OUT FSM* can evolve to **OUT OFF State**.

3 Summary of Remote Commands

The list of the commands used by power units and the corresponding syntax is listed in the following sections.

3.1 Basic Commands

Command	R/W	Options	Short Description
VER	R	VER:?	Gets module model and the installed firmware version
MRID	R	MRID:?	Gets module identification

3.2 Password Related Commands

Command	R/W	Options	Short Description
PASSWORD	R	PASSWORD:?	Queries the actual user privileges
	W	PASSWORD:******	Sets the password
	W	PASSWORD:LOCK	Restores *USER* privileges
SETPSW	W	SETPSW:<*password*>	Changes default *ADMIN* password
RESETPSW	W	RESETPSW:<*reset_pin*>	Resets *ADMIN* password

3.3 Read/Write Internal Memory Related Commands

Command	R/W	Options	Short Description
MRG	R	MRG:<*id*>:?	Reads the given parameter field
MWG	W	MWG:<*id*>:<*value*>	Writes to the given parameter field
MSAVE	W	MSAVE	Saves the parameter in the non-volatile memory

3.4 Status/Faults Register Related Commands

Command	R/W	Options	Short Description
MFTR	R	MFTR:?	Gets module *Faults Register*
MSTR	R	MSTR:?	Gets module *Status Register*
MRESET	W	MRESET	Reset *Faults Register*

3.5 Common Commands

Command	R/W	Options	Short Description
OUT	R	OUT?	Queries *OUT State*
	W	OUT:ON	Enables module output
	W	OUT:OFF	Disable module output
MRI	R	MRI:?	Reads output current
MRV	R	MRV:?	Reads output voltage
MWV	R	MWV:?	Queries last voltage setpoint
	W	MWV:<*float*>	Sets output voltage setpoint
MWVR	R	MWVR:?	Queries last voltage setpoint
	W	MWVR:<*float*>	Goes at voltage setpoint through a ramp
MSRV	R	MSRV:?,	Queries voltage ramp slew rate
	W	MSRV:<*float*>	Sets voltage ramp slew rate
	R	MSRV:MAX:?	Gets maximum voltage ramp slew rate
MWI	R	MWI:?	Queries last current setpoint
	W	MWI:<*float*>	Sets output current setpoint
MWIR	R	MWIR:?	Queries last current setpoint
	W	MWIR:<*float*>	Goes at current setpoint through a ramp
MSRI	R	MSRI:?	Queries current ramp slew rate
	W	MSRI:<*float*>	Sets current ramp slew rate
	R	MSRI:MAX:?	Gets maximum current ramp slew rate
MRW	R	MRW:?	Gets the active output power
MRT	R	MRT:?	Reads temperatures
	R	MRT:NUM:?	Gets number of temperatures
	R	MRT:ALL:?	Gets all temperatures
	R	MRT:<*id*>:?	Reads given temperature
UPFREQ	R	UPFREQ:?	Gets *Update Frequency*

Command	R/W	Options	Short Description
MRP	R	MRP:?	Gets DC-link voltage

3.6 Limits Related Commands

Command	R/W	Options	Short Description
MLIMITS	R	MLIMITS:?	Gets hardware V and I limits
	R	MLIMITS:HW:?	Gets hardware V and I limits
	R	MLIMITS:SW:?	Gets software V and I limits
MPLIMITS	R	MPLIMITS:?	Gets power limits

3.7 Mode Related Commands

Command	R/W	Options	Short Description
LOOP	R	LOOP:?	Queries *Loop Mode*
	W	LOOP:<*mode*>	Sets *Loop Mode*
UPMODE	R	UPMODE:?	Queries module *Update Mode*
	W	UPMODE:<*mode*>	Sets the module *Update Mode*
	R	UPMODE:LIST:?	Gets *Update Mode* list

3.8 Interlocks Commands

Command	R/W	Options	Short Description
INTPM	R	INTPM:?	Queries *Interlock Polarity Mask*
	W	INTPM:<*mask*>	Sets *Interlock Polarity Mask*
	R	INTPM:<*id*>:?	Queries *Interlock Polarity*
	W	INTPM:<*id*>:<*polarity*>	Sets *Interlock Polarity*
INTEM	R	INTEM:?	Queries *Interlock Event Mask*
	W	INTEM:<*mask*>	Sets *Interlock Event Mask*
	R	INTEM:<*id*>:?	Queries *Interlock Event*
	W	INTEM:<*id*>:<*status*>	Sets *Interlock Event*
INTNAME	R	INTNAME:<*id*>:?	Queries *Interlock Name*
	W	INTNAME:<*id*>:<*name*>	Sets *Interlock Name*
INTIT	R	INTIT:<*id*>:?	Queries *Interlock Intervention Time*
	W	INTIT:<*id*>:<*time*>	Sets *Interlock Intervention Time*

3.9 HPPS-JLab Specific Commands

Command	R/W	Options	Short Description
DC	R	DC:?	Queries *DC State*
	W	DC:ON	Enables module DC
	W	DC:OFF	Disable module DC
MRTIGBT	R	MRTIGBT:?	Reds maximum IGBT temperature
	R	MRTIGBT:ALL:?	Reads IGBT temperatures of all modules
	R	MRTIGBT:<*id*>:?	Reads IGBT temperatures
MIG	R	MIG:?	Reads leakage ground current
MIAX	R	MIAX:?	Reads output current

3.10 Advanced Features Commands

Command	R/W	Options	Short Description
WAVE	W	WAVE:START	Starts waveform generation
	W	WAVE:STOP	Stops waveform generation
	R	WAVE:N_PERIODS:?	Queries waveform number of periods
	W	WAVE:N_PERIODS:<*value*>	Sets waveform number of periods
	R	WAVE:PRESCALER:?	Queries waveform prescaler value
	W	WAVE:PRESCALER:<*value*>	Sets waveform prescaler value
	R	WAVE:POINTS:?	Queries waveform points
	W	WAVE:POINTS:<*p_1*>:<*p_2*>:...:<*p_x*>	Sets waveform points
	R	WAVE:TRIGGER:?	Queries *Wave Trigger Mode*
	W	WAVE:TRIGGER:START	Sets *Wave Trigger Mode* to *START*
	W	WAVE:TRIGGER:POINTS	Sets *Wave Trigger Mode* to *POINTS*
	W	WAVE:TRIGGER:GATE	Sets *Wave Trigger Mode* to *GATE*
	W	WAVE:TRIGGER:GATE_RESET	Sets *Wave Trigger Mode* to *GATE_RESET*
TRIG	R	TRIG:?	Queries *Trigger Mode*
	W	TRIG:<*mode*>	Sets *Trigger Mode*

4 Detailed Commands Description

In this chapter are listed the detailed description of the power supply commands.

4.1 Basic Commands

These commands are used to identify the power module.

4.1.1 VER

The **VER** command returns the power supply model and the actual installed firmware version.

R/W	Command	Response
R	VER:?	#VER:<*name*>:<*version*>

Parameters:

Name	Description
name	Device name
version	Firmware version

Examples:

```
VER:?
#VER:NGPS 100-50:2.1.01
```

4.1.2 MRID

The **MRID** command returns the identification name string of the module (see attachment *Internal Memory*) to easily identify the power supply among the other ones connected in the same network.
The default ID is the serial number.
The ID can be set using the command *MWG:30* (see *MWG* command)

R/W	Command	Response
R	MRID:?	#MRID:<*name*>

Parameters:

Name	Description
name	Device identification name

```
MRID:?
#MRID:PS_01
```

4.2 Password Related Commands

These commands are used to change the power supply privileges.

The power supply has two levels of privileges:

- **USER**
- **ADMIN**

With *USER* privileges it is possible to write only basic parameters (such as slew-rate etc.), otherwise with *ADMIN* privileges, it is possible to configure the power supply parameters (such as the PID parameters, interlocks etc.).

For the whole list of parameters and of previleges needed to modify them see attachment *Internal Memory*.

4.2.1 PASSWORD

The **PASSWORD** command is used to change the user privileges.

R/W	Command	Response
R	PASSWORD:?	#PASSWORD:<*privileges*>
W	PASSWORD:<*password*>	#AK

Parameters:

privileges	**Description**
USER	Lowest privileges level: it can use the unit (setting the mode of operation, put the module in on, set the current or voltage etc.), but it can change only the basic configuration parameters
ADMIN	Highest privileges level: it can modify all available parameters and use all commands

password	**Description**
LOCK	*USER* associated password
USER	Equivalent to *LOCK* (no longer supported)
PS-ADMIN	*ADMIN* associated password (see *SETPSW* command to change it)

Examples:

```
PASSWORD:LOCK
#AK

PASSWORD:?
#PASSWORD:ADMIN
```

4.2.2 SETPSW

The **SETPSW** command is used to change the *ADMIN* password (default is "*PS-ADMIN*").
To change the password it is necessary to have already the *ADMIN* privileges.

R/W	Command	Response
W	SETPSW:<*password*>	#AK

Parameters:

Name	Type	Description
password	string[1]	New password associated to ADMIN

1: string cannot contain special character (e.g. "?") or be empty

IMPORTANT:
Pay attention: if the custom password is lost it will not be possible to access with *ADMIN* privileges to the power unit.
To restore the default password see *RESETPSW* command.

Examples:

```
SETPSW:<password>
```

4.2.3 RESETPSW

The **RESETPSW** command is used to restore the default *ADMIN* password word (default is *PS-ADMIN*). To restore the password it is necessary to insert the *reset_pin*; to get it is necessary to contact the CAENels support.

R/W	Command	Response
W	RESETPSW:<*reset_pin*>	#AK

Parameters:

Name	Type	Description
reset_pin	hex[1]	Code provided by CAENels

1: It is a code that doesn't start with '0x'

Examples:

```
RESETPSW:0123456789ABCDEF0123456789ABCDEF
```

4.3 Read/Write Internal Memory Related Commands

These commands are used to read and write *Internal Memory* parameters.

See attachment *Internal Memory* for:

- whole list of parameters
- type of parameter
- privileges needed to modify a field
- description

The *Internal Memory* is model/family dependent.

4.3.1 MRG

The **MRG** command returns the value of the given parameter identifier.
See attachment *Internal Memory*.

R/W	Command	Response
R	MRG:<*id*>:?	MRG:<*id*>:<*value*>

Parameters:

Name	Type	Range	Description
id	int	see attachment *Internal Memory*	*Internal Memory* field ID
value	see attachment *Internal Memory*	-	Value of *Internal Memory* field

Examples:

```
MRG:0:?
#MRG:0:FAST-PS
```

4.3.2 MWG

The **MWG** command is used to write a desired value in the given parameter identifier.
See attachment *Internal Memory*.

R/W	Command	Response
W	MWG:<*id*>:<*value*>	#AK

Parameters:

Name	Type	Range	Description
id	int	see attachment *Internal Memory*	*Internal Memory* field ID
value	see attachment *Internal Memory*	-	Value of *Internal Memory* field

Examples:

```
MWG:30:DEVICE_01
#AK
```

4.3.3 MSAVE

The **MSAVE** command is used to store parameter fields in the non-volatile memory, otherwise they will be lost at the unit power-off.

R/W	Command	Response
W	MSAVE	#AK

```
MSAVE
#AK
```

4.4 Faults/Status Register Related Commands

In this section are listed faults/status commands.

When the module reaches a *Fault condition*, the relative bit is stored in the *Faults Register*, the module goes in the **Fault State** and the power supply goes in a safety condition (please refer to *Finite State Machine* chapter in order to verify the FSM transitions).
For example if the *OUT FSM* is in *ON State* and a *Hard Fault* is reached, the module goes directly to *OFF State*.

To recover the module from the *Fault State* it is necessary to solve the cause of the fault and to send a status reset command (see *MRESET* command).

Faults are classified in:

- *Hard Fault*: those faults compromised the functionality of the module (for example DCCT fault).
- *Soft Fault*: those faults do not compromize the funtionality of the module (for example a temperature fault).

The evolution of the *OUT FSM* depens over the fault.

In *Fault State* the *DC FSM* evolves concordly with the fault type.
In case of *Hard Fault* (for example a Emergency Button pushed), *DC FSM* evolves to *DC OFF State* where the DC-link is disabled (capacitor bank is discharged). *DC FSM* ignores *Soft Fault*.

See attachment *Faults Register* for:

- fault associated bit
- fault name
- fault description

The *Status Register* provides informations about the power supply status (such as if the output on, fault condition, etc.).

See attachment *Status Register* for:

- status associated bit
- status value
- status description

4.4.1 MFTR

The **MFTR** command returns the *Faults Register*.
See attachment *Faults Register*.

R/W	Command	Response
R	MFTR:?	MFTR:<*faults*>

Parameters:

Name	Type	Size	Description
faults	hex	64bits	*Faults Register* (LSB is associated to fault 1)

Examples:

```
MFTR:?
MFTR:0x0
```

4.4.2 MSTR

The **MSTR** command returns internal *Status Register*.
See attachment *Status Register*.

R/W	Command	Response
R	MSTR:?	MSTR:<*status*>

Parameters:

Name	Type	Size	Description
status	hex	64bits	*Status Register* (LSB is associated to status 1)

Examples:

```
MSTR:?
MSTR:0x1
```

4.4.3 MRESET

The MRESET performs a reset of the module *Faults Register*.
System exits from **Fault State**.

R/W	Command	Response
W	MRESET	#AK

Examples:

```
MRESET
#AK
```

4.5 Common Commands

The commands used to turn on and turn off the power supply, set setpoint, read current and voltage output and some common readback (as temperatures).

4.5.1 OUT

The **OUT** command is used to evolve the *OUT FSM* and query the actual state.

R/W	Command	Response
R	OUT:?	#OUT:<*status*>
W	OUT:<*status*>	#AK

Parameters:

status	**Mode**	**Description**
ON	R	*Power Module* is enabled, it is ready to regulate[1] and to have connected a load
	W	*OUT FSM* evolves from *OUT OFF State* (or *OUT Wait for Off State*) to *OUT ON State*
OFF[1]	R	The output driver is disabled
	W	*OUT FSM* evolves from *OUT ON State* state to *OUT Wait for Off State*[2]
WAIT4OFF	R	The power supply is performing a ramp down to zero (the slew-rate of the ramp is factory defined)

1: In *CV Mode* voltage output is set to 0V, in *CC Mode* current output is set to 0A (see *LOOP* command)

2: If *OUT FSM* is in *OUT Wait for Off State*, *OUT:OFF* command forces *OUT FSM* to *OUT OFF State*

Examples:

```
OUT:?
#OUT:OFF

OUT:ON
#AK
```

4.5.2 MRI

The **MRI** command returns the readback value of the power supply output current.

R/W	Command	Response
R	MRI:?	#MRI:<*value*>

Parameters:

Name	Type	Unit	Description
value	float	A	Output current

Examples:

```
MRI:?
#MRI:10.1234567
```

4.5.3 MRV

The **MRV** command returns the readback value of the power supply output voltage.

R/W	Command	Response
R	MRV:?	#MRV:<*value*>

Parameters:

Name	Type	Unit	Description
value	float	V	Output voltage

Examples:

```
MRV:?
#MRV:10.1234567
```

4.5.4 MWV

The **MWV** command is used to set the output voltage during the *CV Mode* (see *LOOP* command), or query the last voltage setpoint.
A *Trigger Event* indicates when the setpoint become effective (see *TRIG* command).

R/W	Command	Response
R	MWV:?	#MWV:<*value*>
W	MWV:<*value*>	#AK

Parameters:

Name	Type	Unit	Description
value	float	V	Desired output voltage value

The output voltage can be set only during *OUT ON State* and if *NORMAL Update Mode* is selected (see *UPMODE* command) and *CV Mode* is used (see *LOOP* command).

Examples:

```
MWV:?
MWV:5.0000000

MWV:35
#AK
```

4.5.5 MWVR

The **MWVR** command is used to achieve the set output voltage through a ramp, whose slew rate could be changed (see *MSRV* command), or query the last voltage setpoint.
A *Trigger Event* indicates when the setpoint become effective (see *TRIG* command).

R/W	Command	Response
R	MWVR:?[1]	#MWVR:<*value*>
W	MWVR:<*value*>	#AK

1: equivalent to *MWV* command

Parameters:

Name	Type	Unit	Description
value	float	V	Desired output voltage value

The output voltage can be set only during *OUT ON State* and if *NORMAL Update Mode* is selected (see *UPMODE* command) and *CV Mode* is used (see *LOOP* command).

Examples:

```
MWIR:?
#MWI:15.0000000

MWVR:10
#AK
```

4.5.6 MSRV

The **MSRV** command is used to dynamically set or queries the voltage ramp slew rate. It is possible also get the maximum voltage slew rate.

R/W	Command	Response
R	MSRV:?	#MSRV:<*value*>
W	MSRV:<*value*>	#AK
R	MSRV:MAX:?	#MSRV:MAX:<*max_value*>

Parameters:

Name	Type	Range	Unit	Description
value	float	(0, *max_value*]	V/s	Voltage slew rate
max_value	float	-	V/s	Maximum voltage slew rate

Examples:

```
MSRV:?
#MSRV:30.0000000

MSRV:15
#AK

MSRV:MAX:?
#MSRV:MAX:50.0000000
```

4.5.7 MWI

The **MWI** command is used to set the output current during the *CC Mode* (see *LOOP* command), or query the last current setpoint.
A *Trigger Event* indicates when the setpoint become effective (see *TRIG* command).

R/W	Command	Response
R	MWI:?	#MWI:<*value*>
W	MWI:<*value*>	#AK

Parameters:

Name	Type	Unit	Description
value	float	A	Desired output current value

The output current can be set only during *OUT ON State* and if *NORMAL Update Mode* is selected (see *UPMODE* command) and *CC Mode* is used (see *LOOP* command).

Examples:

```
MWI:?
MWI:10.0000000

MWI:15
#AK
```

4.5.8 MWIR

The **MWIR** command is used to achieve the set output current through a ramp, whose slew rate could be changed (see *MSRI* command), or query the last current setpoint.
A *Trigger Event* indicates when the setpoint become effective (see *TRIG* command).

R/W	Command	Response
R	MWIR:?[1]	#MWIR:<*value*>
W	MWIR:<*value*>	#AK

1: equivalent to *MWI* command

Parameters:

Name	Type	Unit	Description
value	float	A	Desired output current value

The output current can be set only during *OUT ON State* and if *NORMAL Update Mode* is selected (see *UPMODE* command) and *CC Mode* is used (see *LOOP* command).

Examples:

```
MWIR:<setpoint>
#AK

MWIR:?
#MWI:<setpoint>
```

4.5.9 MSRI

The **MSRI** command is used to dynamically set or get the current ramp slew rate.
It is possible also query the maximum current slew rate.

R/W	Command	Response
R	MSRI:?	#MSRI:<*value*>
W	MSRI:<*value*>	#AK
R	MSRI:MAX:?	#MSRI:MAX:<*max_value*>

Parameters:

Name	Type	Range	Unit	Description
value	float	(0, *max_value*]	A/s	Current slew rate
max_value	float	-	A/s	Maximum current slew rate

Examples:

```
MSRI:10
#AK

MSRI:?
#MSRI:15

MSRI:MAX:?
#MSRI:MAX:30
```

4.5.10 MRW

The **MRW** command returns the value of the active output power, estimated as the product between the output voltage and output current readbacks.

R/W	Command	Response
R	MRW:?	#MRW:<*value*>

Parameters:

Name	Type	Unit	Description
value	float	W	Active power

Examples:

```
MRW:?
#MRW:23.5489764
```

4.5.11 MRT

The **MRT** query the power supply temperatures.
It is possible also get the number of available temperatures.
See attachment *Temperatures* for more informations.

R/W	Command	Response
R	MRT:?	#MRT:<*max_temp*>
R	MRT:NUM:?	#MRT:NUM:<*temp_number*>
R	MRT:ALL:?	#MRT:ALL:<*temp_1*>:<*temp_x*>:...
R	MRT:<*x*>:?	#MRT:<*x*>:<*temp_x*>

Parameters:

Name	Type	Range	**Description**
temp_number	int	[1,3][1]	Number of available temperatures
x	int	[1,3][1]	Temperature number

1: Available temperatures are family/model dependent

max_temp / **temp_x**	**Type**	**Unit**
(Max) Temperature	float	°C

Examples:

```
MRT:?
#MTR:33.5

MRT:NUM:?
#MRT:NUM:3

MRT:ALL:?
#MRT:ALL:38.4:35.6:38.4:65.1

MRT:1:?
#MRT:1:32.6
```

4.5.12 UPFREQ

The **UPFREQ** command gets the power supply *Update Frequency*.
Update frequency is model/family dependent.

R/W	Command	Response
R	UPFREQ:?	#UPFREQ:<*frequency*>

Parameters:

Name	Unit	Description
frequency	Hz	*Update Frequency* used by PID and waveform generator (see *WAVE* command)

Examples:

```
UPFREQ:?
#UPFREQ:100000
```

4.5.13 MRP

The **MRP** command return DC-link voltage.

R/W	Command	Response
R	MRP:?	#MRP:<*value*>

Parameters:

Name	Type	Range	Unit	Description
value	float	>=0	V	DC-link voltage

Examples:

```
MRP:?
#MRP:40.0000000
```

4.6 Limits Commands

The power supply has two type of internal limits for current and voltage:

- Hardware limits *HW Limits*
- Software limits *SW Limits*

HW Limits are factory defined and rapresents power supply limits.

SW Limits are user defined. These limits prevent setting the setpoint out of safe range.
See attachment *Internal Memory* to change these values.

From *HW Limits* derives *Power Limits*

4.6.1 MLIMITS

The **MLIMITS** command queries the current and voltage limits.

R/W	Command	Response
R	MLIMITS:?	#MLIMITS:<*hw_min_V*>:<*hw_max_V*>:<*hw_min_I*>:<*hw_max_I*>
R	MLIMITS:HW:?	#MLIMITS:<*hw_min_V*>:<*hw_max_V*>:<*hw_min_I*>:<*hw_max_I*>
R	MLIMITS:SW:?	#MLIMITS:<*sw_min_V*>:<*sw_max_V*>:<*sw_min_I*>:<*sw_max_I*>

Parameters:

Name	Type	Unit	Description
hw_min_V	float	V	Hardware minimum voltage
hw_max_V	float	V	Hardware maximum voltage
hw_min_I	float	A	Hardware minimum current
hw_max_I	float	A	Hardware maximum current
sw_min_V	float	V	Software minimum voltage
sw_max_V	float	V	Software maximum voltage
sw_min_I	float	I	Software minimum current
sw_max_I	float	I	Software maximum current

Examples:

```
MLIMITS:?
#MLIMITS:-10:10:-30:30

MLIMITS:HW:?
#MLIMITS:-10:10:-30:30

MLIMITS:SW:?
#MLIMITS:0:10:0:20
```

4.6.2 MPLIMITS

The **MPLIMITS** command returns the value of maximum allowed output active power.

R/W	Command	Response
R	MPLIMITS:?	#MPLIMITS:<*min_power*>:<*max_power*>

Parameters:

Name	Type	Unit	Description
min_power	float	W	Minimum allowed active power
max_power	float	W	Maximum allowed active power

Examples:

```
MPLIMITS:?
#MPLIMITS:0:200
```

4.7 Mode Related Commands

In this chapter are listed the commands to handle the control loop mode *Loop Mode* and the setpoint mode *Update Mode*.

The *Loop Mode* command defines the loop control type:

- *Costant-Current Mode (CC)*
- *Costant-Voltage Mode (CV)*

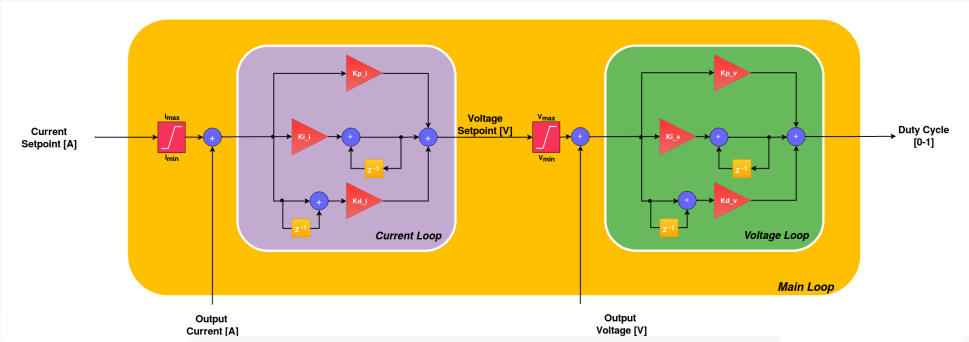


Figure 4.1: CC Mode

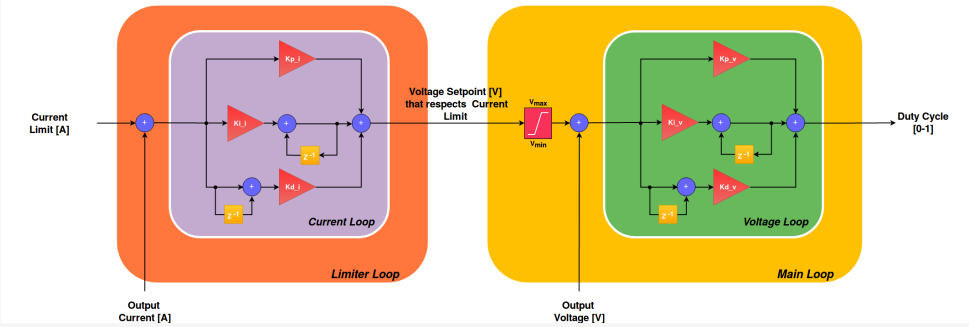


Figure 4.2: CV Mode

The *Update Mode* defines the setpoint source. The module can accept setpoint from different sources for example form the Ethernet command, Analog input (when available), etc.

4.7.1 LOOP

The **LOOP** command is used to select or get the regulation mode of the power supply.

R/W	Command	Response
R	LOOP:?	#LOOP:<*mode*>
W	LOOP:<*mode*>	#AK

Parameters:

mode	**Description**
I	*CC Mode*
V	*CV Mode*

Loop Mode cannot be set whe the module is in *OUT ON State*.

Examples:

```
LOOP:?
#LOOP:I

LOOP:V
#AK
```

4.7.2 UPMODE

The **UPMODE** command allows the user to select or read the *Update Mode*. It can be used also to list the available options.

R/W	Command	Response
R	UPMODE:?	#UPMODE:<*mode*>
W	UPMODE:<*mode*>	#AK
R	UPMODE:LIST:?	#UPMODE:<*mode_0*>:<*mode_1*>:...: <*mode_x*>

Parameters:

mode / ***mode_x***	**Description**
NORMAL	Setpoint is set using standard Ethernet communication
WAVE	Setpoint is taken from the internal function generator module
ANALOG[1]	Setpoint is set using analog input [-10V,+10V]
SFP[2]	Setpoint is taken from a dedicated custom command sent to SFP interface

1: Family and model dependent

2: Family dependent

Examples:

```
UPMODE:WAVE
#AK

UPMODE:?
#UPMODE:NORMAL

UPMODE:LIST:?
#UPMODE:LIST:NORMAL:WAVE:AIN:SFP
```

4.8 Interlocks Commands

These commands can be used to configure the user configurable external interlocks.
In case of an external interlock event, the output is disabled (see the *Finite State Machine FSM* chapter) and the fault is signaled in the *Fault Register* (see *Faults/Status Register Related Commands*.

The interlocks commands allow to:

- enable/disable interlocks
- set interlocks polarity
- change the default interlocks name
- set an intervetion time

4.8.1 INTPM

The **INTPM** command reads or sets the interlocks polarity.
It is possible set the polarity for a single or all interlocks simultaneously.

R/W	Command	Response
R	INTPM:?	#INTPM:<*mask*>
W	INTPM:<*mask*>	#AK
R	INTPM:<*id*>:?	#INTPM:<*id*>:<*polarity*>
W	INTPM:<*id*>:<*polarity*>	#AK

Parameters:

Name	Type	Range	Description
mask	hex	[0x0,0x3FF][1]	Interlocks polarity mask (LSB is associated to interlock 0)
id	int	[0,9][1]	Interlock ID
polarity	bit	[0x0;0x1]	Interlock polarity

1: The number of editable interlocks is family dependent

polarity	Description
0x0	Direct polarity
0x1	Inverted polarity

Examples:

```
INTPM:?
#INTPM:0x3

INTPM:0x2
#AK

INTPM:1:?
#INTPM:1:0x1
```

```
INTPM:0:0x0
#AK
```

4.8.2 INTEM

The **INTEM** command enables or disables the interlocks.
It is possible enable or disable a single or all interlocks simultaneously.

R/W	Command	Response
R	INTEM:?	#INTEM:<*mask*>
W	INTPM:<*mask*>	#AK
R	INTPM:<*id*>:?	#INTEM:<*id*>:<*status*>
W	INTPM:<*id*>:<*status*>	#AK

Parameters:

Name	Type	Range	Description
mask	hex	[0x0,0x3FF][1]	Interlocks enable mask (LSB is associated to interlock 0)
id	int	[0,9][1]	Interlock ID
status	bit	[0x0;0x1]	Interlock status

1: The number of editable interlocks is family dependent

status	Description
0x0	Disabled
0x1	Enabled

Examples:

```
INTPM:?
#INTPM:0x1

INTPM:0x3
#AK

INTPM:0:?
#INTPM:0:0x0
```

```
INTPM:1:0x1
#AK
```

4.8.3 INTNAME

The **INTNAME** command gets or sets the interlocks name.

R/W	Command	Response
R	INTNAME:<*id*>:?	#INTNAME:<*id*>:<name*>
W	INTNAME:<*id*>:<*name*>	#AK

Parameters:

Name	Type	Range	Description
id	int	[0,9][1]	Interlock ID
name	string[2]	-	Interlock name

1: The number of editable interlocks is family dependent

2: string cannot contain special character (e.g. "?") or cannot be empty.

Examples:

```
INTNAME:0:?
#INTNAME:0:Interlock 0

INTNAME:1:Int1
#AK
```

4.8.4 INTIT

The **INTIT** command gets or sets the interlocks intervention time.

R/W	Command	Response
R	INTIT:<*id*>:?	#INTIT:<*id*>:<*time*>
W	INTIT:<*id*>:<*time*>	#AK

Parameters:

Name	Type	Range	Unit	Description
id	int	[0:9][1]	-	Interlock ID
time	int	[0:10000]	ms	Interlock intervention time

1: The number of editable interlocks is family dependent

Examples:

```
INTIT:0:?
#INTIT:0:10000

INTIT:1:0
#AK
```

4.9 HPPS-JLab Specific Commands

The following commands are specific for the HPPS-JLab family.

4.9.1 DC

The **DC** command is used to evolve the *DC FSM* or to query the *DC State*.

R/W	Command	Response
R	DC:?	#DC:<*status*>
W	DC:<*state*>	#AK

Parameters:

status	**Mode**	**Description**
ON	R	Capacitor bank reaches the nominal value
	W	*DC FSM* evolves from *DC OFF State* to *DC Charging State*
OFF	R	The capacitor bank is not ready
	W[1]	*DC FSM* evolves from *DC ON State* state to *DC OFF State*

1: In *DC Charging State* it is not possible use *DC:OFF*

The *DC:ON* command cannot be used if DC-link voltage isn't about 0V.

Examples:

```
DC:?
#DC:OFF

DC:ON
#AK
```

4.9.2 MRTIGBT

The **MRTIGBT** command return the maximum IGBTs temperature and the temperature of all 3 IGBTs for every *Power Modules*.
It is possible query the absolute maximum temperature, the temperatures for all modules simultaneously or for a single *Power Module*.

R/W	Command	Response
R	MRTIGBT:?	#MRTIGBT:<*max_all*>
R	MRTIGBT:ALL:?	#MRTIGBT:ALL:<*max_1*>:<*t_11*>:<*t_12*>:<*t_13*>:<*max_2*>:...:<*t_xy*>:...
R	MRTIGBT:<*x*>:?	#MRTIGBT:<*x*>:<*max_x*>

Parameters:

value	Range	Description
x	[1,4]	Number of *Power Module*
y	[1,3]	Number of IGBT

max_all / max_x / t_xy	Type	Unit
(Max) IGBTs temperature	float	°C

Examples:

```
MRTIGBT:?
#MRTIGBT:42.3

MRTIGBT:ALL:?
#MRTIGBT:ALL:53.4:46.8:47.9:53.4:38.9:36.4:38.9:35.5:...

MRTIGBT:2:?
#MRTIGBT:2:53.6
```

4.9.3 MIG

The **MIG** command return the leakage ground current.

R/W	Command	Response
R	MIG:?	#MIG:<*ground_current*>

Parameters:

Name	Type	Unit	Description
ground_current	float	A	Leakage ground current

Examples:

```
MIG:?
#MIG:0.1
```

4.9.4 MIAX

The **MIAX** command return the output current.

R/W	Command	Response
R	MIAX:?	#MIAX:<*current*>

Parameters:

Name	Type	Unit	Description
current	float	A	Output current

Examples:

```
MIAX:?
#MIAX:1001.7
```

4.10 Advanced Features Commands

This section explain some advance power supply functionality.

Power supply allows to generate custom waveform both in *CC Mode* or *CV Mode*.

The *WAVE* command is used to define the waveform, start and stop the generation and select the mode.

The *TRIG* command is used to set the *Trigger Event*.

4.10.1 WAVE

The **WAVE** command allows to perform a custom waveform setpoint input.
To use this functionality it is necessary to set the *WAVE Update Mode* (see *UPMODE* command).

R/W	Command	Response
W	WAVE:START	#AK
W	WAVE:STOP	#AK
R	WAVE:N_PERIODS:?	#WAVE:N_PERIODS:<*n_periods*>
W	WAVE:N_PERIODS:<*n_periods*>	#AK
R	WAVE:PRESCALER:?	#WAVE:PRESCALER:<*prescaler*>
W	WAVE:PRESCALER:<*prescaler*>	#AK
W	WAVE:POINTS:<*p_1*>:<*p_2*>:...:<*p_x*>	#AK
R	WAVE:POINTS:?	#WAVE:POINTS:<*p_1*>:<*p_2*>:...:<*p_x*>
W	WAVE:TRIGGER:<*mode*>	#AK
R	WAVE:TRIGGER:?	#WAVE:TRIGGER:<*mode*>

Name	Type	Default	Range
n_periods	int	0	>=0

The *n_periods* parameter specify the number of periods that has to be executed.
With *n_periods* equal to 0, the waveform will will be executed infinite times (the execution can be stopped anytime with the *WAVE:STOP* command).

Name	Type	Default	Range
prescaler	int	1	[1,100]

The *prescaler* parameter specify the waveform prescaler value. This is used to reduce the default *Update Frequency* and so to increase the length of the waveform.
For example: if the power unit has an *Update Frequency* of 100kHz and the *prescaler* is set to default value 1, the maximum waveform legth will be 5s. With a *prescaler* of 4, the waveform points will be

updated at 25kHz and in this specific case the maximum waveform length will be increase up to 20s).
The *Update Frequency* is family/model dependent (see *UPFREQ* command).

Name	Type	Unit	Range
p_x	float	V or A	[5,500'000]

The *p_x* values rapresent the waveform points that have to be executed.
The waveform points will be updated with the module *Update Frequency* (see *UPFREQ* command) divided by the value of the *prescaler*.
The minimum number of points is 5 and the maximum is 500'000.
The points has to be represented in ASCII mode.

mode	**Description**
START	Starts the waveform execution on *Trigger Event*
POINTS	Reproduces the next point on *Edge Trigger Event*
GATE	Waveform execution is enabled using a *Level Trigger Event*
GATE_RESET	Is the same of *GATE*, but waveform is rearmed when disabled

mode can be set once the *Trigger Event* is enabled (see *TRIG* command):

- *START*: The waveform execution start on *Trigger Event* (see *TRIG* command). This mode support also a software start using *WAVE:START*. To stop the execution it is necessary use *WAVE:START* command.
- *POINTS*: On *Edge Trigger Event* the next waveform point is reproduced. To arm the waveform execution it is necessary a *WAVE:START* command.
  This mode is meant to use the waveform like a buffer of setpoints and at every *Edge Trigger Event* (see *TRIG* command), the next point is executed. For this reason the *prescaler* setting is bypassed.
- *GATE*: The *Level Trigger State* is used as an enable or disable signal. To arm the waveform execution it is necessary to send the *WAVE:START* command.
- *GATE_RESET*: Is the same of *GATE*, but waveform is rearmed when disabled.

In following examples it is used:

- sine waveform of amplitude 1V
- frequency 1Hz
- number of periods 3

- prescaler 1.

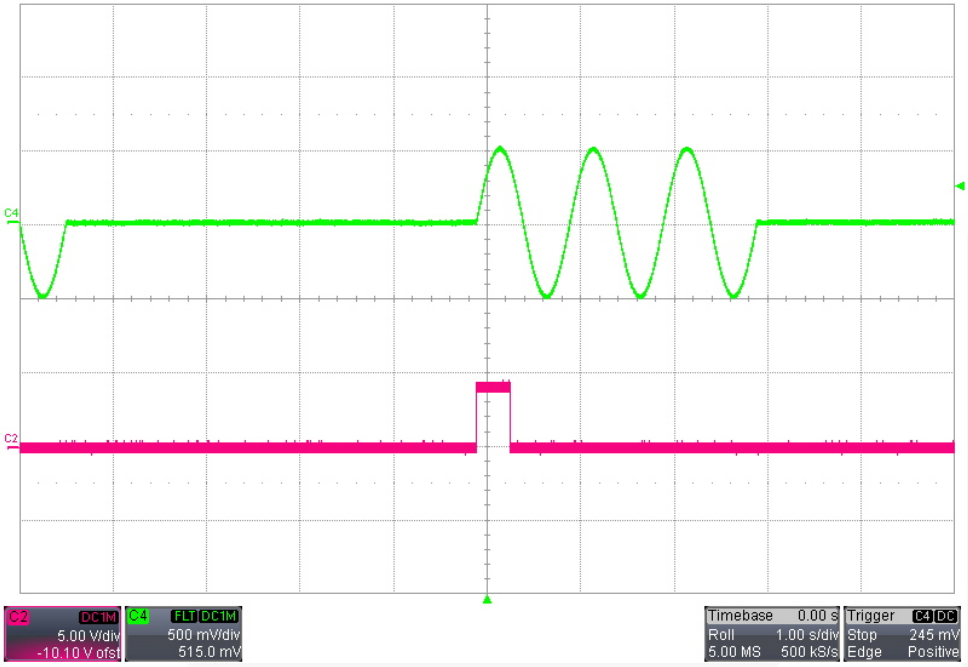


Figure 4.3: WAVEFORM:TRIGGER:START with TRIG:POS option

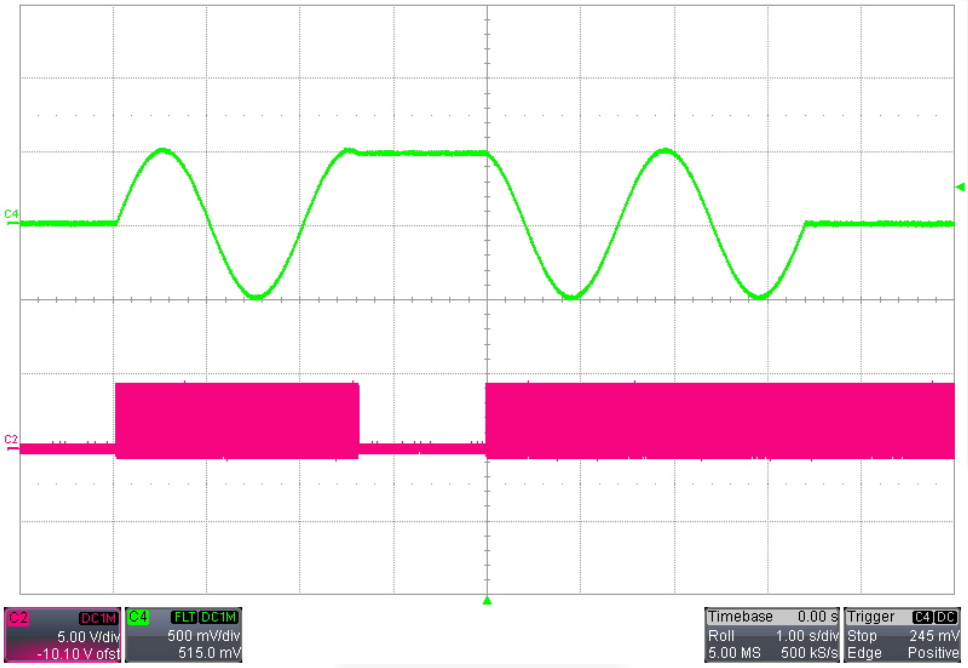


Figure 4.4: WAVEFORM:TRIGGER:POINTS with TRIG:POS option and trigger frequency of 50kHz

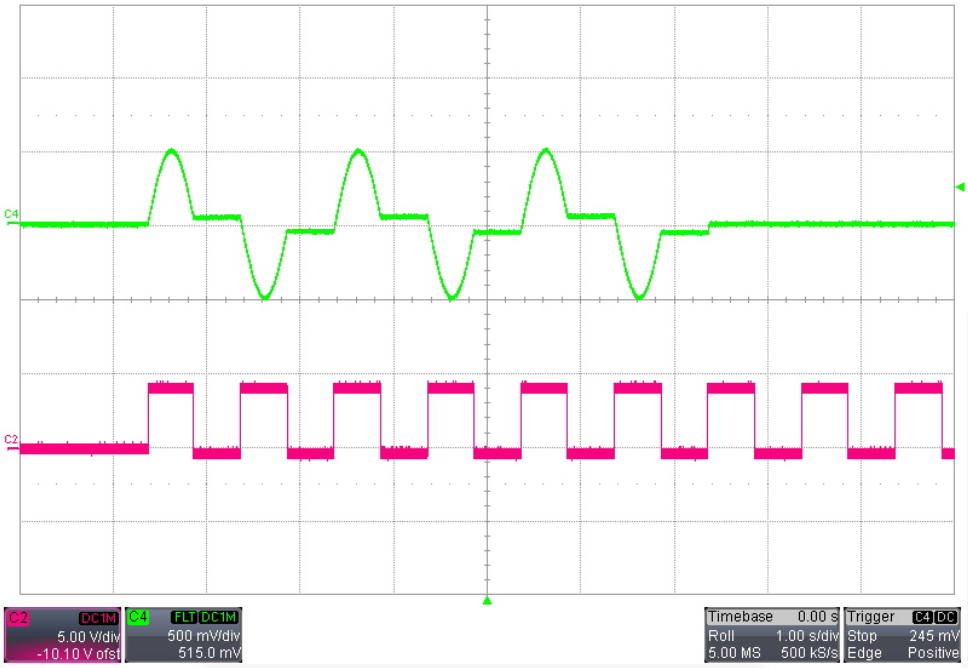


Figure 4.5: WAVEFORM:TRIGGER:GATE with TRIG:HIGH_LEVEL option and trigger frequency of 1 Hz

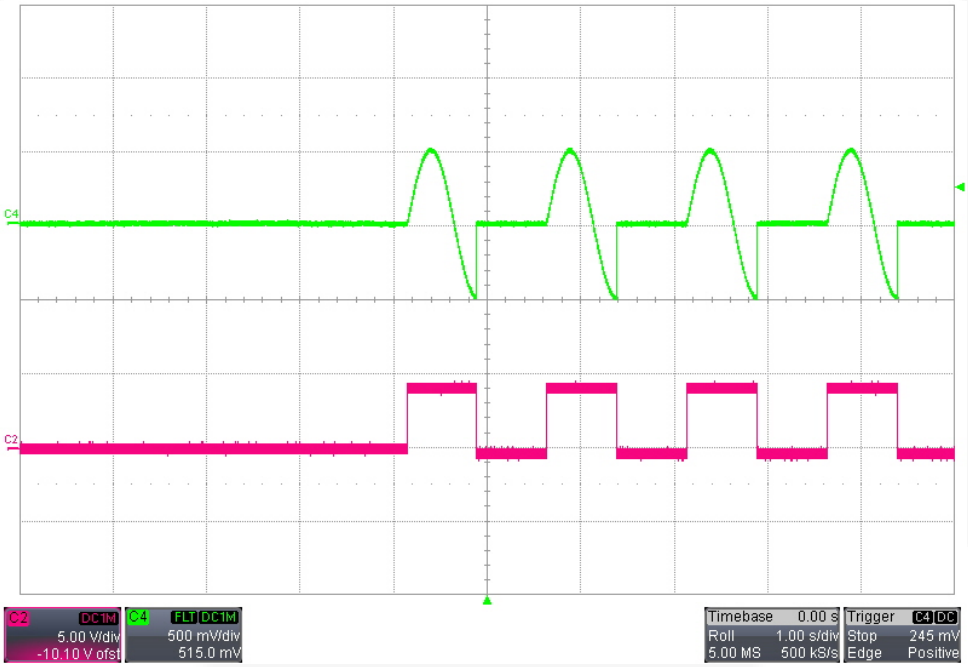


Figure 4.6: WAVEFORM:TRIGGER:GATE_RESET with TRIG:HIGH_LEVEL option and trigger frequency of 0.667 Hz (period 1,5s)

Examples:

```
WAVE:N_PERIODS:?
#WAVE:N_PERIODS:0

WAVE:N_PERIODS:3
#AK

WAVE:PRESCALER:?
#WAVE:PRESCALER:1

WAVE:PRESCALER:10
#AK

WAVE:POINTS:0:10:15:20:10:5:10:...
#AK

WAVE:POINTS:?
#WAVE:POINTS:0:10:15:10:0:5:...

WAVE:TRIGGER:?
#WAVE:TRIGGER:POINTS

WAVE:TRIGGER:START
#AK

WAVE:START
#AK

WAVE:STOP
#AK
```

4.10.2 TRIG

The **TRIG** command is used to enable and set the external trigger.

R/W	Command	Response
R	TRIG:?	#TRIG:<*mode*>
W	TRIG:<*mode*>	#AK

mode	**Description**
OFF	Disable the *Trigger Event* (default value)
POS	*Edge Trigger Event* on positive edge
NEG	*Edge Trigger Event* on negative edge
BOTH	*Edge Trigger Event* on positive and negative edge
HIGH_LEVEL	*Level Trigger Event* when signal is high
LOW_LEVEL	*Level Trigger Event* when signal is low

```
TRIG:?
#TRIG:OFF

TRIG:POS
#AK
```

5 Internal Memory

ID	Field	Type	Privileges	Description
0	Firwmare ID	string	R-only	Firmware ID
1	Model	string	R-only	Model
2	Serial Number	string	R-only	Serial Number
3	MAC Address	string	R-only	MAC Address
9	Calibration Date	string	R-only	Calibration date
10	Current Calib. Param. a	float	R-only	Current cubic calib. param. a [4]
11	Current Calib. Param. b	float	R-only	Current cubic calib. param. b [4]
12	Current Calib. Param. c	float	R-only	Current cubic calib. param. c [4]
13	Current Calib. Param. d	float	R-only	Current cubic calib. param. d [4]
14	Voltage Calib. Param. a	float	R-only	Voltage cubic calib. param. a [4]
15	Voltage Calib. Param. b	float	R-only	Voltage cubic calib. param. b [4]
16	Voltage Calib. Param. c	float	R-only	Voltage cubic calib. param. c [4]
17	Voltage Calib. Param. d	float	R-only	Voltage cubic calib. param. d [4]
18	DC-link Calib. Param. a	float	R-only	DC-link linear calib. param. a [5]
19	DC-link Calib. Param. b	float	R-only	DC-link linear calib. param. b [5]
20	DC-link AC Calib. Param. a	float	R-only	DC-link AC linear calib. param. a [5]
21	DC-link AC Calib. Param. b	float	R-only	DC-link AC linear calib. param. b [5]
26	Aux. 1 Calib. Param. a	float	R-only	Aux1 linear calib. param. a [5]
27	Aux. 1 Calib. Param. b	float	R-only	Aux1 linear calib. param. b [5]
28	Aux. 2 Calib. Param. a	float	R-only	Aux2 linear calib. param. a [5]
29	Aux. 2 Calib. Param. b	float	R-only	Aux2 linear calib. param. b [5]
30	Module ID	string	Admin	Default is the serial number [2]

ID	Field	Type	Privileges	Description
31	Current Slew Rate	float	User	Slew rate used in *CC Mode* [A/s]
32	Voltage Slew Rate	float	User	Slew rate used in *CV Mode* [V/s]
35	TFT timeout	int	Admin	LCD timeout [minutes]
36	Feed Forward	0 | 1[1]	Admin	Feed Forward
40	PID CC Kp_v	float	Admin	Kp Voltage in *CC Mode*
41	PID CC Ki_v	float	Admin	Ki Voltage in *CC Mode*
42	PID CC Kd_v	float	Admin	Kd Voltage in *CC Mode*
43	PID CC Kp_i	float	Admin	Kp Current in *CC Mode*
44	PID CC Ki_i	float	Admin	Ki Current in *CC Mode*
45	PID CC Kd_i	float	Admin	Kd Current in *CC Mode*
46	PID CC Max Voltage	float	Admin	Max Voltage in *CC Mode* [V]
47	PID CC Min Voltage	float	Admin	Min Voltage in *CC Mode* [V]
48	PID CC Max Current	float	Admin	Max Current in *CC Mode* [A]
49	PID CC Min Current	float	Admin	Min Current in *CC Mode* [A]
50	PID CC Mode	float	Admin	CC Mode: 0 single; 1 double
55	Allow Remote OFF	0 | 1[1]	Admin	REMOTE OFF in LOCAL Mode
56	Error Code Description	0 | 1[1]	Admin	Error description in NAK
57	Enable EPICS	0 | 1[1]	Admin	Enable EPICS server
60	PID CV Kp_i	float	Admin	Kp Current in *CV Mode*
61	PID CV Ki_i	float	Admin	Ki Current in *CV Mode*
62	PID CV Kd_i	float	Admin	Kd Current in *CV Mode*
63	PID CV Kp_v	float	Admin	Kp Voltage in *CV Mode*
64	PID CV Ki_v	float	Admin	Ki Voltage in *CV Mode*
65	PID CV Kd_v	float	Admin	Kd Voltage in *CV Mode*
66	PID CV Max Current	float	Admin	Max Current in *CV Mode* [A]
67	PID CV Min Current	float	Admin	Min Current in *CV Mode* [A]
68	PID CV Max Voltage	float	Admin	Max Voltage in *CV Mode* [V]

ID	Field	Type	Privileges	Description
69	PID CV Min Voltage	float	Admin	Min Voltage in *CV Mode* [V]
80	Output Over-Current Limit	float[2]	Admin	Output current threshold [A]
81	Output Over-Voltage Limit	float[2]	Admin	Output voltage threshold [V]
82	Max Temperature	float	Admin	Max Cabinet temperature threshold [°Celsius]
83	Undervoltage Threshold	float	Admin	Undervoltage threshold [V]
86	Regulation Fault Current Limit	float	Admin	Current regulation fault [A]
87	Regulation Fault Voltage Limit	float	Admin	Voltage regulation fault [V]
88	Regulation Fault Intervention Time	float	Admin	Regulation time fault [s]
90	Interlock Enable Mask	hex[3]	Admin	Interlocks enable
91	Interlock Activation Level Mask	hex[3]	Admin	Interlocks polarity
92	Interlock #0 Intervention Time	int	Admin	Interlock 0 intervetion time [ms]
93	Interlock #0 Name	string	Admin	Interlock 0 name
94	Interlock #1 Intervention Time	int	Admin	Interlock 1 intervetion time [ms]
95	Interlock #1 Name	string	Admin	Interlock 1 name
96	Interlock #2 Intervention Time	int	Admin	Interlock 2 intervetion time [ms]
97	Interlock #2 Name	string	Admin	Interlock 2 name
98	Interlock #3 Intervention Time	int	Admin	Interlock 3 intervetion time [ms]
99	Interlock #3 Name	string	Admin	Interlock 3 name
130	Capabilities	string	R-only	Available capabilities
140	Max IGBT Temperature	float	Admin	IGBT temperature threshold [°Celsius]

ID	Field	Type	Privileges	Description
141	Charging Timeout	int	R-only	Charging timeout [ms]
149	Fan Activation Temperature	float	Admin	Fan Activation Temperature [°Celsius]
180	Auxiliary Output Over-Current Limit	float	Admin	Auxiliary output current threshold [A]
181	Ground Current Limit	float	Admin	Ground current threshold [A]
190	Interlock #4 Intervention Time	int	Admin	Interlock 4 intervetion time [ms]
191	Interlock #4 Name	string	Admin	Interlock 4 name
192	Interlock #5 Intervention Time	int	Admin	Interlock 5 intervetion time [ms]
193	Interlock #5 Name	string	Admin	Interlock 5 name
194	Interlock #6 Intervention Time	int	Admin	Interlock 6 intervetion time [ms]
195	Interlock #6 Name	string	Admin	Interlock 6 name
196	Interlock #7 Intervention Time	int	Admin	Interlock 7 intervetion time [ms]
197	Interlock #7 Name	string	Admin	Interlock 7 name
198	Interlock #8 Intervention Time	int	Admin	Interlock 8 intervetion time [ms]
199	Interlock #8 Name	string	Admin	Interlock 8 name
200	Interlock #9 Intervention Time	int	Admin	Interlock 9 intervetion time [ms]
201	Interlock #9 Name	string	Admin	Interlock 9 name

1. 0 : disabled; 1 : enabled
2. 0 to disable
3. Format: 0x<mask>
4. Formula: $y=dx\hat{3}+cx\hat{2}+bx+a$
5. Formula: $y=bx+a$

6 Faults Register

Bit	Name	Description
1	Overtemperature	Max temperature > Max Cabinet temperature threshold see *Temperatures* chapter see *Internal Memory* chapter ID 82
2	DC Link Undervoltage	DC-Link voltage < Undervoltage threshold see *Internal Memory* chapter ID 83
4	Overpower	Power out of *Power Limits*
7	Regulation Fault	Output not reaches the setpoint range in time see *Internal Memory* chapter ID 86,87 and 88
9	DCCT Error	DCCT is not connected
14	Output Overcurrent	Output current > Output Over-Current Limit see *Internal Memory* chapter ID 80
15	Output Overvoltage	Output voltage > Output Over-Voltage Limit see *Internal Memory* chapter ID 81
17	Interlock #0	Interlock 0 event
18	Interlock #1	Interlock 1 event
19	Interlock #2	Interlock 2 event
20	Interlock #3	Interlock 3 event
21	Interlock #4	Interlock 4 event
22	Interlock #5	Interlock 5 event
23	Interlock #6	Interlock 6 event
24	Interlock #7	Interlock 7 event
25	Interlock #8	Interlock 8 event

Bit	Name	Description
26	Interlock #9	Interlock 9 event
29	Module 1 Communication Fault	*Power Module* 1 communication error
30	Module 2 Communication Fault	*Power Module* 2 communication error
31	Module 3 Communication Fault	*Power Module* 3 communication error
32	Module 4 Communication Fault	*Power Module* 4 communication error
33	Overtemp. Module 1	Overtemperature of the *Power Module* 1
34	Overtemp. Module 2	Overtemperature of the *Power Module* 2
35	Overtemp. Module 3	Overtemperature of the *Power Module* 3
36	Overtemp. Module 4	Overtemperature of the *Power Module* 4
37	Driver Fault Module 1	IGBT *Power Module* 1 saturation
38	Driver Fault Module 2	IGBT *Power Module* 2 saturation
39	Driver Fault Module 3	IGBT *Power Module* 3 saturation
40	Driver Fault Module 4	IGBT *Power Module* 4 saturation
41	Emergency Button	Emergency push button is engaged
42	3 Phase Fault	Three-phase loss
43	Door State Fault	Door opened
44	Water Flow Fault	Water flow loss
45	Transformer Overtemp.	Transformer overtemperature
46	Rectifier Overtemp.	Rectifier overtemperature
47	Main Contactor Fault	Wrong main contactor status
48	DC Inductors Overtemp.	DC inductor overtemperature
49	IGBT Heatsink Overtemp.	Power Module heatsink overtemperature
50	Damp Resistor Overtemp.	Output filter resistor overtemperature
51	Precharge Res. Overtemp.	Precharge resistors overtemperature
52	Out Inductors Overtemp.	Output inductors Overtemperature

Bit	Name	Description
53	Voltage Measure Breaker Fault	Voltage measure breaker open
54	Ground Breaker Open	Ground breaker tripped
55	3 Phase Current Unbalanced	Three-phase current unbalanced
56	Key released	Key released
57	Charging Timeout	Main contactor timeout
58	Ground Overcurrent	Ground overcurrent
59	Safety Fault	System is in safety stop
60	Water Leakage	Water leakage
61	Output Overcurrent Aux. DCCT	Output overcurrent auxiliary DCCT
62	Personnel Safety System Fault	Personnel Safety System

7 Status Register

Bit	Value	Description
1	0	Module is not in *OUT ON State*
	1	Module is in *OUT ON State*
2	0	Module is not in *Fault State*
	1	Module is in *Fault State*
3	0	Module is not in OUT *Wait for Off State*
	1	Module is in OUT *Wait for Off State*
5	0	*Loop Mode* is *CC Mode*
	1	*Loop Mode* is *CV Mode*
9-10	00	*Update Mode* is *NORMAL*
	10	*Update Mode* is *WAVE*
13	0	*Trigger Event* is disabled
	1	*Trigger Event* is enabled
17	0	Ramp is disabled
	1	Ramp is enabled
18	0	A custom wave is not in execution
	1	A custom wave is in execution
23	0	*Local Mode* is enabled
	1	*Local Mode* is disabled

Bit	Value	Description
33	0	Module is not in *DC ON State*
	1	Module is in *DC ON State*
34	0	Module is not in *DC Charging State*
	1	Module is in in *DC Charging State*

8 Temperatures

ID	Name
1	Temperature Cabinet 1
2	Temperature Cabinet 2
3	Shunt Temperature
Max	Max(Temp_1, Temp_2)

9 NAK List

Code	Description
01	Unknown Command
02	Unknown Parameter
03	Invalid Parameter
04	Not Enough Arguments
05	Privilege Level Requirement not met
06	Save Error on device
07	Invalid Password
08	Module is in Fault state
09	Module is in ON state
10	Parameter is out of hardware limits
11	Parameter is out of defined limits
12	Parameter is not a number
13	Module is in OFF state
14	Slew rate out of limits
15	Device is set in local mode, cannot modify values from remote interface while in this state
16	Module is not currently generating a waveform
17	Module is currently generating a waveform
18	Device is in remote mode, cannot modify values from local interface while in this state
19	Loop mode already set to desired value
20	Loop mode is not the same that uses the variable required to change
21	Module is not in normal update mode

Code	Description
22	Float mode already set to desired value
23	Unknown sub-command for SFP command
24	Unknown feature or feature not available
25	Parallel fault
26	Waveform error
27	Cannot open the required file
28	Cannot change set point because the module is inverting polarity
29	Cannot write waveform data
30	Polarity switch not allowed
31	Cannot set options for socket used by oscilloscope
32	Cannot change settings because in parallel slave mode
33	MASTER and SLAVES have different firmware versions
34	MASTER and SLAVES are different models
35	MASTER and SLAVES have different ratings
36	The required feature is not available
37	UDP buffer overflow
38	Module is in WAIT FOR OFF state
39	The debug field is read only
40	Cannot parse input name for debug field
41	Cannot parse input value for debug field
42	Cannot parse type for debug field
43	DHCP is enabled
44	The specified command is disabled
45	Output current fault
46	Dissipative Unit Enabled
47	DC-link not ready
48	EPICS is disabled

Code	Description
49	Module is not in wave update mode
50	DC-link is not in OFF condition
51	SFP is not available
52	NTP is active
80	Post Mortem Monitor is not ready
99	Unknown Error