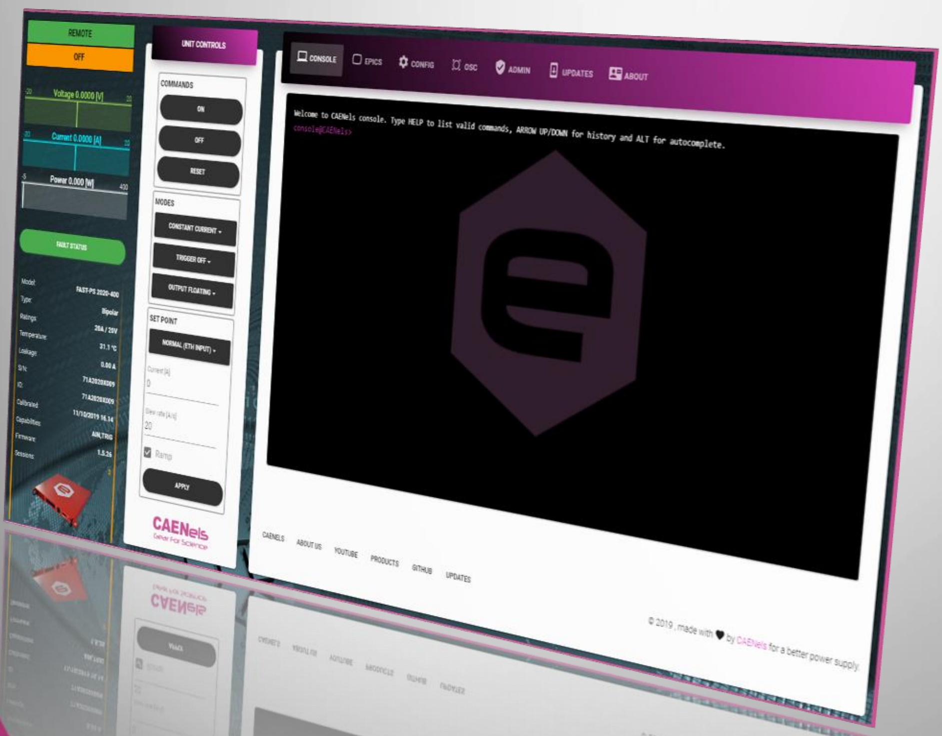
Regul8OR Series

Remote Control Manual



MAGNET POWER SUPPLY SYSTEMS

Rev. 1.0 – June 2021

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

Remote Control Manual – Models

This manual covers the following models:

- **REGUL8OR**

Table Of Contents

Document Revisions

Revision	Date	Comment
1.0	June 3^{rd} 2021	First Release

1. Overview

In this manual, the user can find all the information related to the dedicated high-level control and programming language.

Chapter 2 discusses in detail the web server control, while chapter 4 the dedicated programming language. When the control unit is controlled via the web server or via programming language, possible operations are the same.

The main difference is that the programming language is a ready-to-use (ASCII-based) proprietary language which allows the user to perform specific routines for its specific application, this guaranteeing high freedom and flexibility.

2. Dedicated Software

The described utilities allow a user-friendly and fast access to the functionalities and configuration of the Regul8OR module unit.

Two different software packages are available for operation with CAEN ELS power supplies: "*CAENels Device Manager*" and "*Visual*" Software. The first utility can be downloaded free-of-charge from the CAENels website www.caenels.com. An overview of both utilities is given in the next sections.

2.1 CAENels Device Manager

The "*CAENels Device manager*" software can be used to detect:

- CAEN ELS devices described within this document and connected to the local network;
- Their network configuration;

The "*CAENels Device manager*" is available for Windows and Linux platform. The system requirements are:

1. Windows minimum system requirements:
   - Windows® XP or newer
   - Intel® or equivalent processor
   - 70 MB available HD space
   - Ethernet network card

2. Linux minimum system requirements:
   - Linux kernel 2.2.x or newer
   - Intel® or equivalent processor
   - 70 MB available HD space
   - Ethernet network card

2.1.1 Searching for connected devices

The following steps have to be performed in order to carry out a search of all the CAEN ELS units connected to the local network:

- Install the *"CAENels Device manager"* software;
- Launch the software;
- Perform a scan to discover the connected e.g. FAST-PS device(s) by clicking the *"Scan"* button as indicated in **Figure 1**. If there are multiple available connections it is possible to select the network/networks to be scanned in the *"Selected network interfaces"* window available under the "*Options*" menu. All the informations about the selected devices are shown in the right side of the main window.

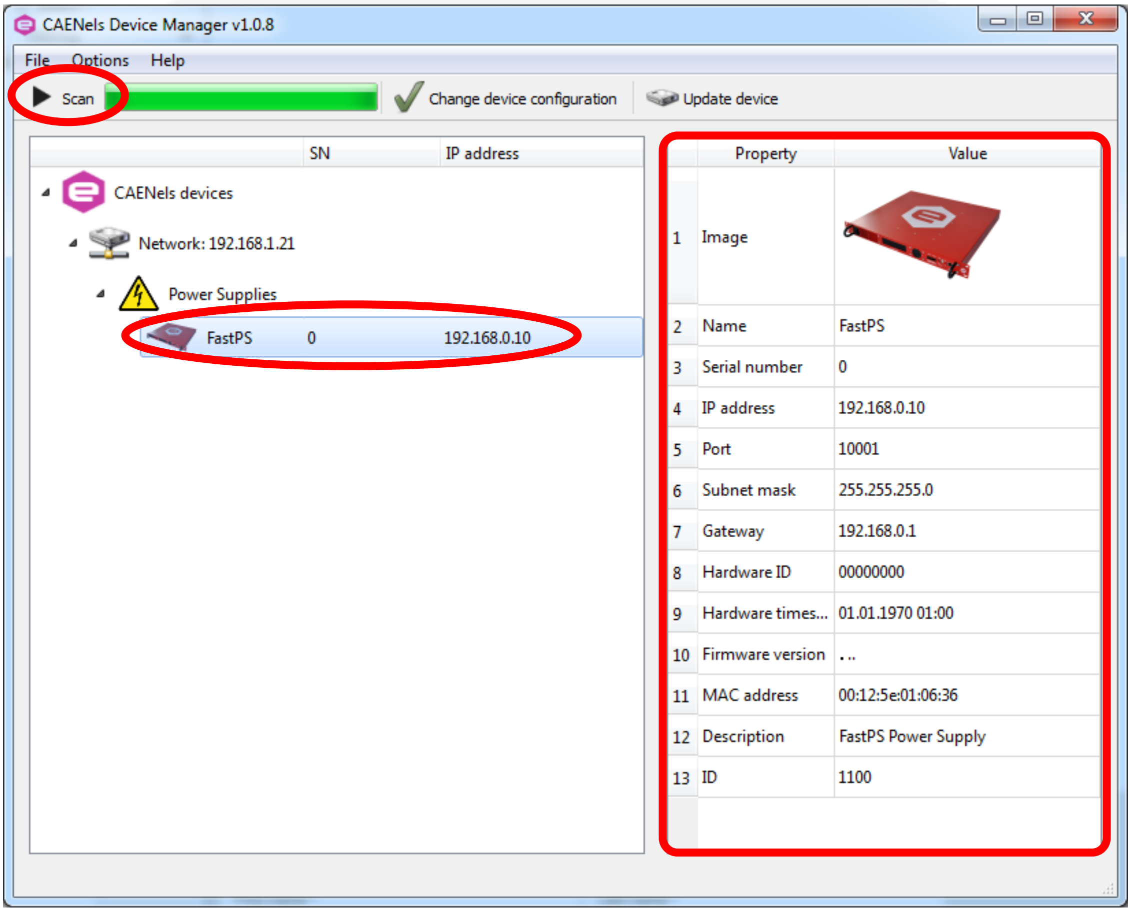


Figure 1: Device Manager - Main interface

Make sure that the firewall is not preventing communication between your computer and the unit(s). The "*CAENels Device manager*" uses **UDP port 30719** to find the device, so ensure that the UDP traffic is allowed in both directions on this port.

2.1.2 Device Configuration

The software allows also to change the Network configuration of the found device(s) in the local network.

In order to change the network configuration of the unit it is necessary to select the desired device and click on the "*Change device configuration*" button in the main window as shown in **Figure 2**. The configurable Network options are:

- Device IP address;
- TCP/IP communication port;
- Subnet mask;
- Gateway.

To apply the changes on the device configuration it is necessary to edit the corresponding fields and then to click on the "*Save*" button. A screenshot of a sample device configuration is shown in the following picture:

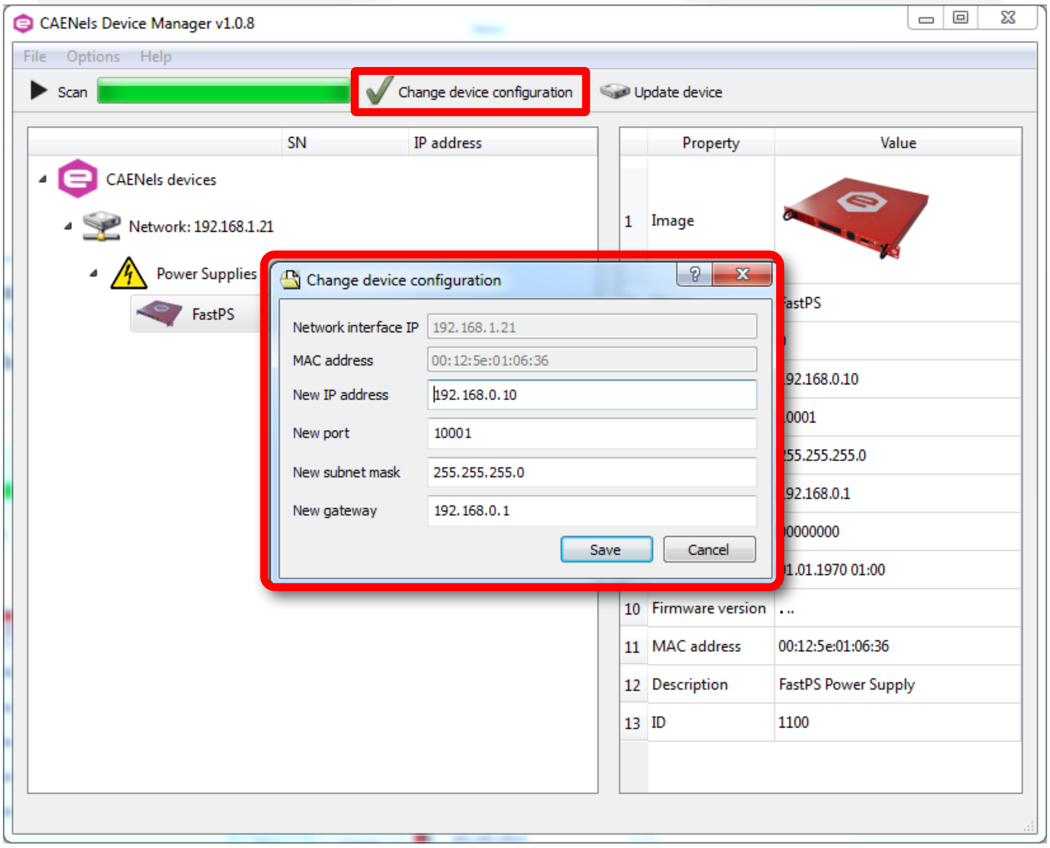


Figure 2: Device Manager - Change device configuration

3. Web Server

A running web server makes it easy to remote control the main features of the CAEN ELS power converters and control units using a Graphic User Interface (GUI).

In order to establish connection with the unit, be sure that the network configuration of the PC in use allows the connection to the module (same subnet of the power module). Then, it is necessary to type the power supply IP address in the web browser in order to connect to it:

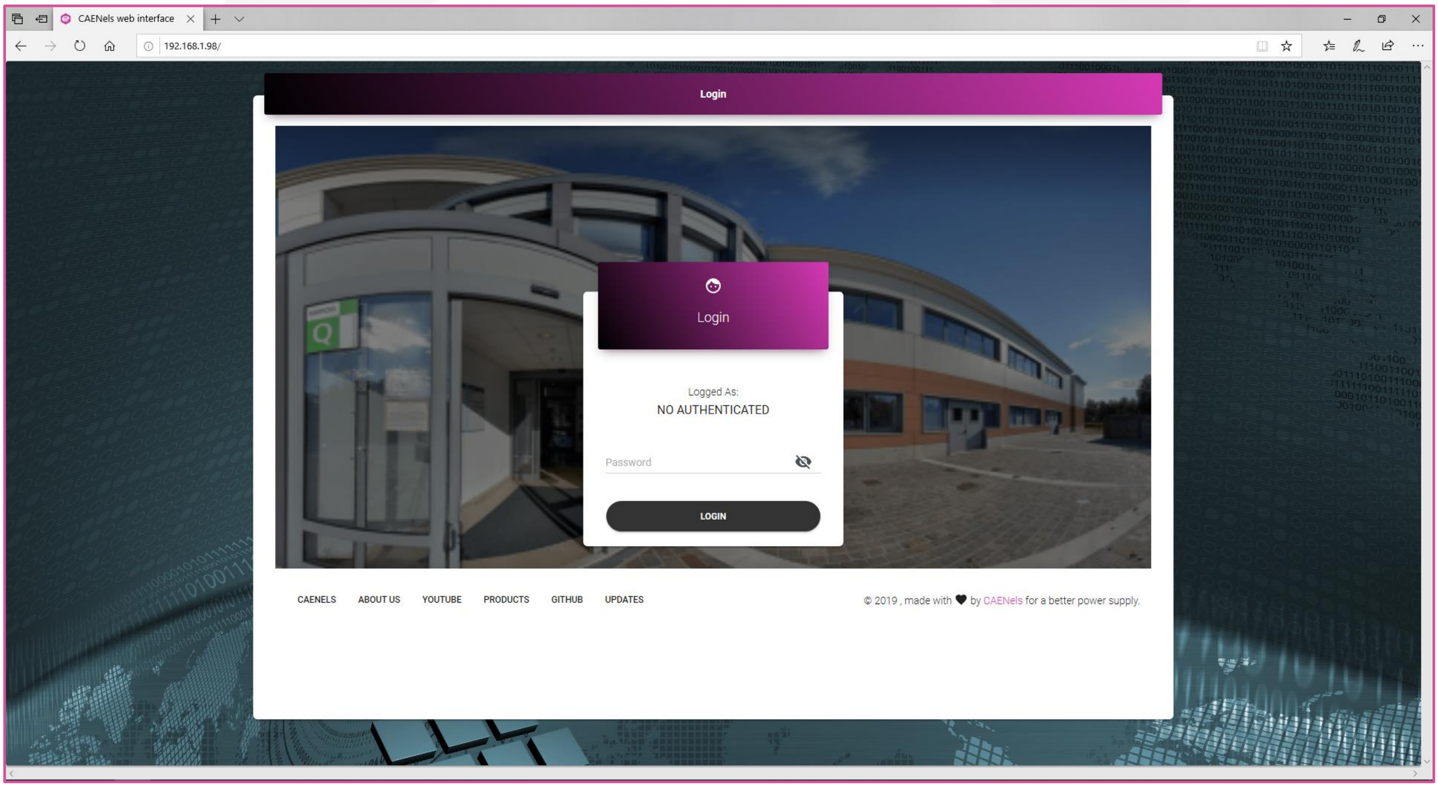


Figure 3: Web Server Login

When in the Login page as shown in **Figure 3**, set Password:

- “ps-admin”: for administrator privileges (unlock password protected memory cells);

In either case, some cells are locked as these are factory protected.

3.1 Main Windows

The web server main window is presented below:

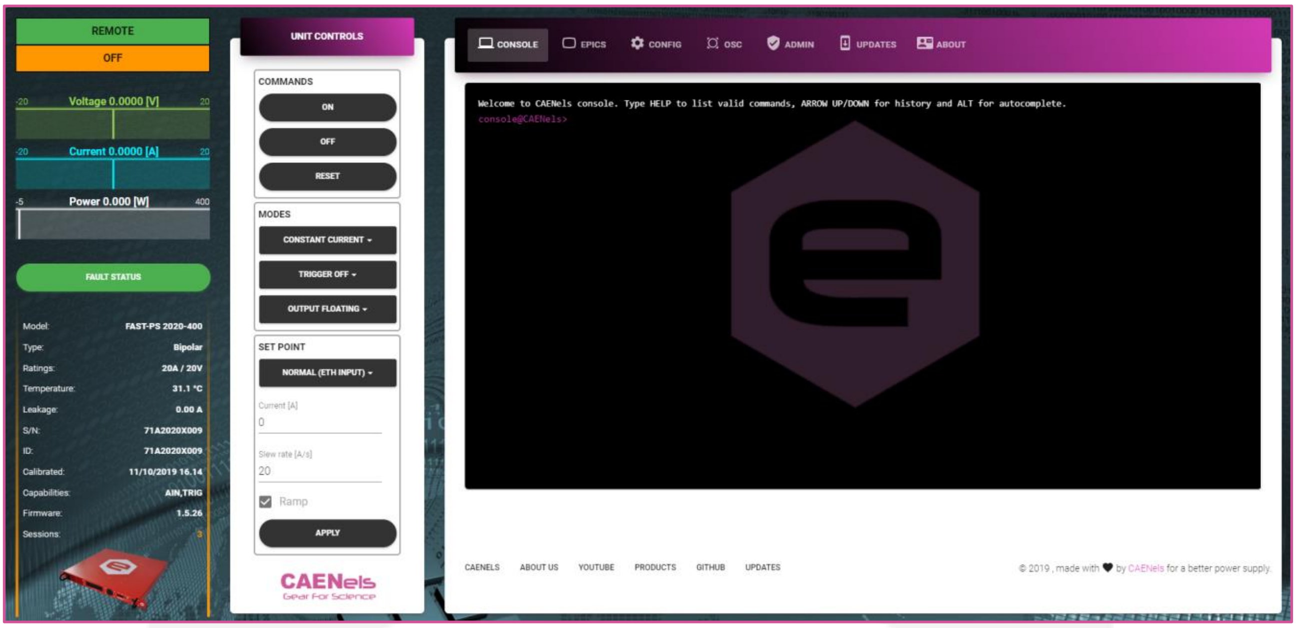


Figure 4: Web Server – Main Window

The “CONSOLE” is used to send direct commands to the control unit, (refer to chapter 4 for the supported commands). In addition, by typing “clear”, it is possible to reset the console, and by clicking Alt autocomplete of the commands is available.

On the left side, device informations are reported:

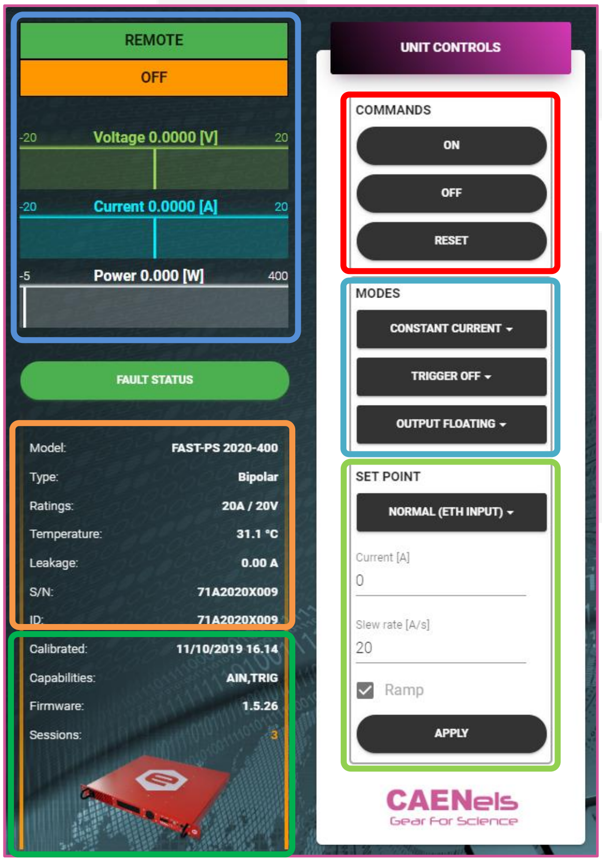


Figure 5: Web Server – Main Window Information

Unit Controls: allow to Switch ON/OFF the unit and to Reset its status register (in case of a fault).

Regulation mode: allows to select the mode of operation. The regulation mode can be changed only when the unit is switched OFF.

Voltage/Current Set section: this section allows to set the Voltage (for the “*Constant Voltage*” mode) or Current (for the “*Constant Current*” mode) setpoint. To apply a setpoint it is necessary to click on the *Ramp* or *Set* button. The *Ramp* button performs a ramp to the selected setpoint, otherwise the *Set* button applies directly the selected setpoint.

General information: indicates some information regarding the connected unit, like the model and its serial number.

Unit status: indicates some information regarding output status, temperature, current leakage, unit control (local or remote) and fault status. By clicking on the fault status indicator, it is possible to visualize detailed fault status windows.

Output monitor: indicates the actual output voltage, current and power.

3.1.1 Faults reset

Please refer to section 4.7 of this manual for the complete list of possible faults;

If a fault happens, the “FAULT STATUS” green button in **Figure 4** and **Figure 5** will turn to red.

To understand the nature of the fault, please click on the red “FAULT STATUS” button, the web interface will show the fault.

In order to reset the fault and thus, to turn the control unit on again:

1. Remove the fault cause
2. Click on the RESET button
3. Turn ON the control module and so the power supply.

3.1.2 Unit Controls window

The “UNIT CONTROLS” window in **Figure 5** allows to turn the module ON/OFF and to reset eventual faults (when the “FAULT STATUS” widget in **Figure 5** is red). Refer to section 4.7 for further information on faults list.

3.1.3 Regulation Mode

The “MODES” window in **Figure 5** allows to select between two different control modes:

- Constant Current (CC): the module set-point is in [A]
- Constant Voltage (CV): the module set-point is in [V]

In addition, the user can select different triggering options:

- Off: no trigger is used
- Pos: positive edge is used
- Neg: negative edge is used
- Both: both edges (pos and neg) are used

Please refer to the User’s Manual of the device in use in order to have information on the HW operation of the trigger.

For further information on the chain from the trigger signal input to the power source output, refer to section 3.1.4.

3.1.4 Set Point Control Modes

The Regul8or unit may be controlled in different ways:

- Normal: single set point
- Waveform: pre-stored and custom waveforms generation
- Analog: the power source acts as amplifier of an external analog input

In the following sections these three modes are discussed.

3.1.4.1 *Normal*

In this control mode the user can set a single set point. In order to do so, just type the desired value in [A] or [V] (depending on CV or CC, see section 3.1.3) and click "APPLY":

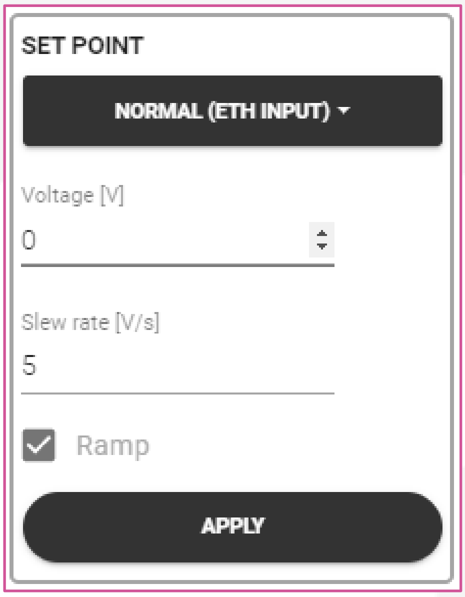


Figure 6: SET POINT – NORMAL

If "Ramp" is selected, the module will apply a specific A/s or V/s ramping to the setpoint (it may be changed by changing the Slew Rate value in **Figure 6**). The A/s or V/s default speed is stored in the internal memory (see section 3.3.3) and it is named "Default Current Slew Rate [A/s]" for CC and "Default Voltage Slew Rate [V/s]" for CV.

Please note that the maximum current slew rate is equal to the current full-scale times ten, and the voltage one is equal to the voltage full scale times ten.

3.1.4.2 *Waveform*

An embedded waveform generator may be used for set-points generation. The arbitrary waveform generator uses a DMA (Direct Memory Access) module to reproduce the waveform's setpoints which are stored in a physical memory, this zeroing the latency time TCP or even UDP protocols would bring.

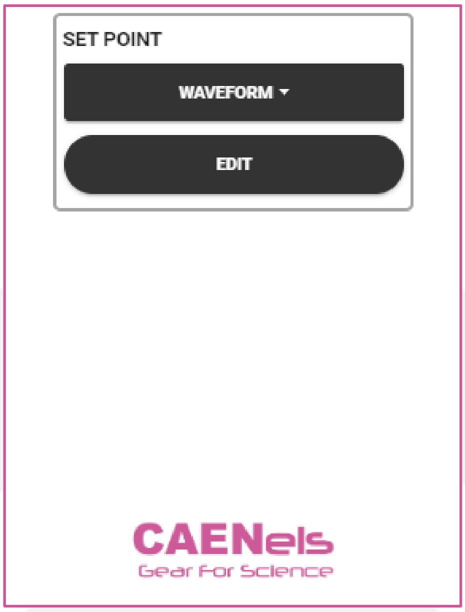


Figure 7: SET POINT – WAVEFORM

By clicking "EDIT" (**Figure 7**) the user can select between 4 different waveforms:

- Sine Wave
- Square Wave
- Triangular Wave
- Custom Wave

In the following pages these 4 waveforms are discussed.

Sine Wave

Once sine wave is selected, the below page will be accessed on the web server:

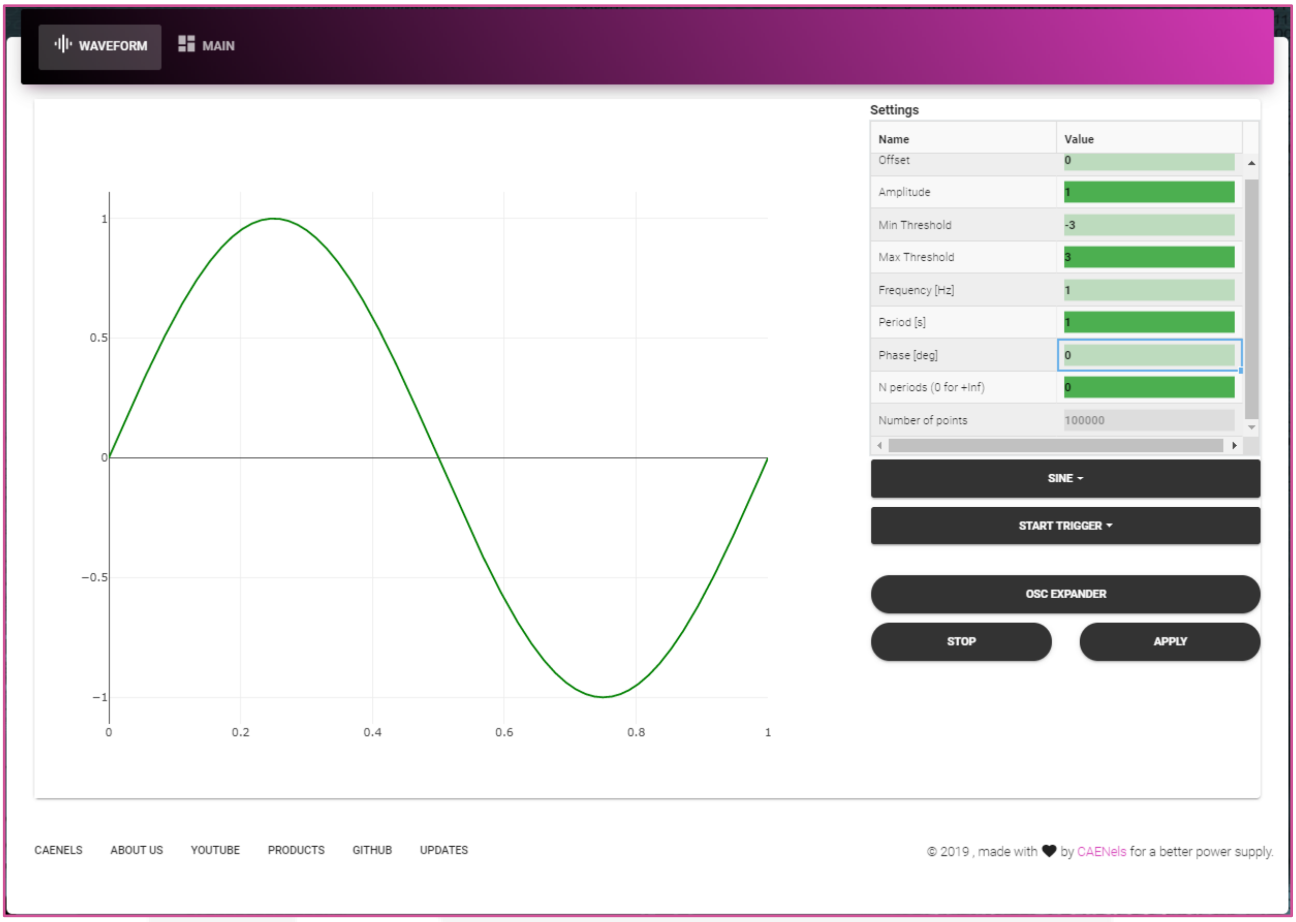


Figure 8: Sine Wave

Several parameters are present on the right side of the window, which the user is allowed to edit. Once the wanted waveform has been described, click "APPLY" in order to reproduce it and "STOP" to stop it.

In addition, two trigger options are present, if needed:

- START TRIGGER: the entire waveform will be reproduced once the trigger signal will be detected. Trigger must be set in order to do so (see section 3.1.3).
- POINTS TRIGGER: the waveform will be reproduced point by point per each trigger signal detected. Trigger must be set in order to do so (see section 3.1.3).

"OSC EXPANDER" may be used in order to see in parallel the waveform window, the oscilloscope one (see section 3.4), and to also have access to the PID settings (see section 3.3.2), in order to modify in real time the dynamic of the module and to observe the response changing.

Square Wave

Once square wave is selected, the below page will be accessed on the web server:

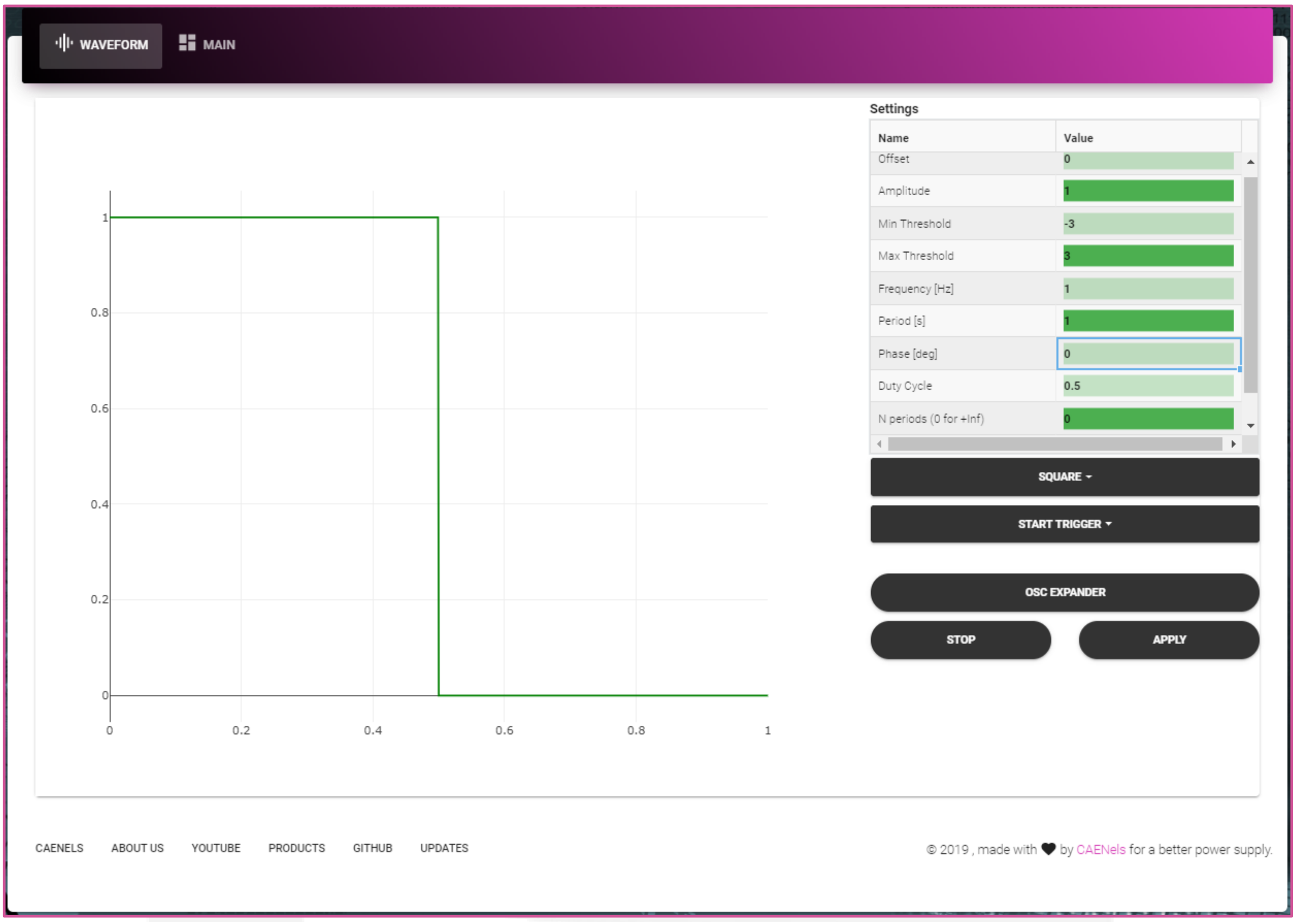


Figure 9: Square Wave

Several parameters are present on the right side of the window, which the user is allowed to edit. Once the wanted waveform has been described, click "APPLY" in order to reproduce it and "STOP" to stop it.

In addition, two trigger options are present, if needed:

- START TRIGGER: the entire waveform will be reproduced once the trigger signal will be detected. Trigger must be set in order to do so (see section 3.1.3).
- POINTS TRIGGER: the waveform will be reproduced point by point per each trigger signal detected. Trigger must be set in order to do so (see section 3.1.3).

"OSC EXPANDER" may be used in order to see in parallel the waveform window, the oscilloscope one (see section 3.4), and to also have access to the PID settings (see section 3.3.2), in order to modify in real time the dynamic of the module and to observe the response changing.

Triangular Wave

Once triangular wave is selected, the below page will be accessed on the web server:

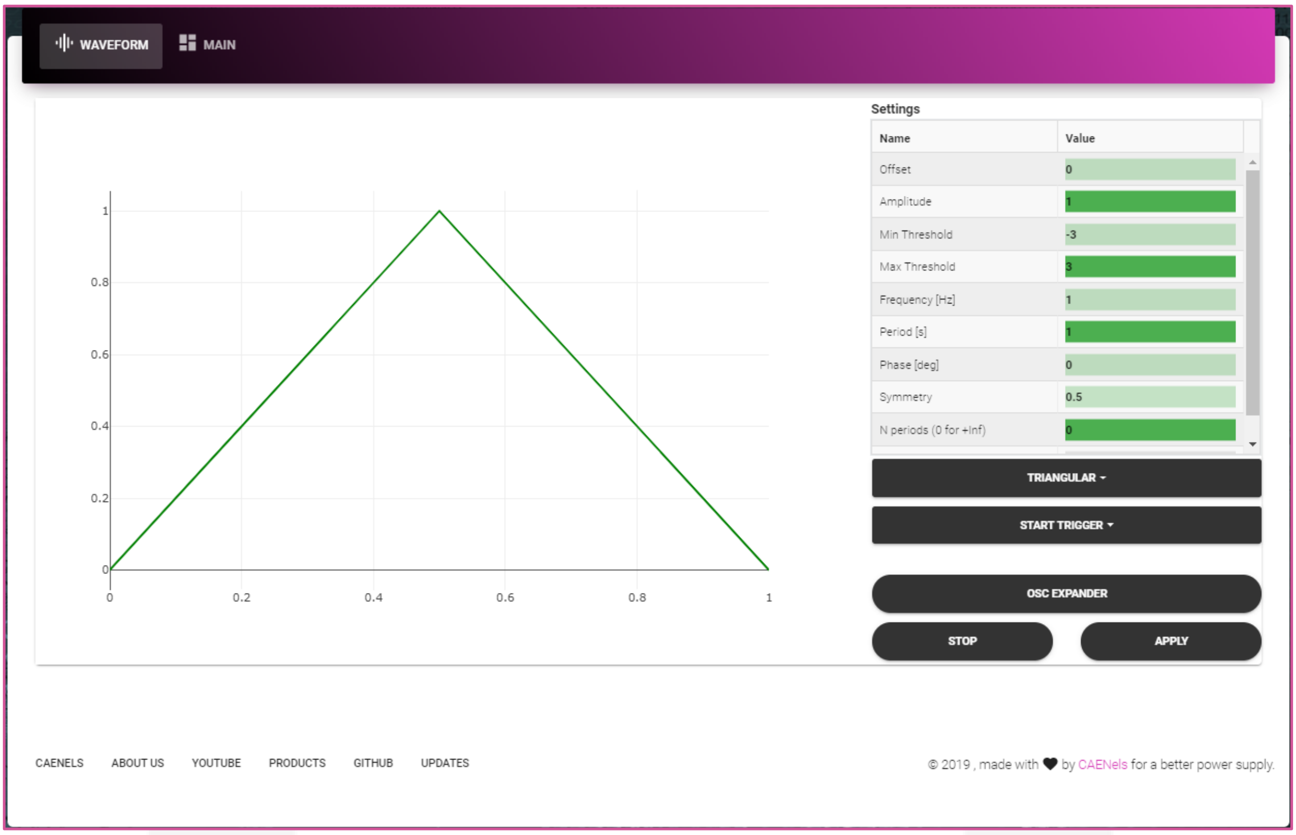


Figure 10: Triangular Wave

Several parameters are present on the right side of the window, which the user is allowed to edit. Once the wanted waveform has been described, click "APPLY" in order to reproduce it and "STOP" to stop it.

In addition, two trigger options are present, if needed:

- START TRIGGER: the entire waveform will be reproduced once the trigger signal will be detected. Trigger must be set in order to do so (see section 3.1.3).
- POINTS TRIGGER: the waveform will be reproduced point by point per each trigger signal detected. Trigger must be set in order to do so (see section 3.1.3).

"OSC EXPANDER" may be used in order to see in parallel the waveform window, the oscilloscope one (see section 3.4), and to also have access to the PID settings (see section 3.3.2), in order to modify in real time the dynamic of the module and to observe the response changing.

Custom Wave

Once triangular wave is selected, the below page will be accessed on the web server:

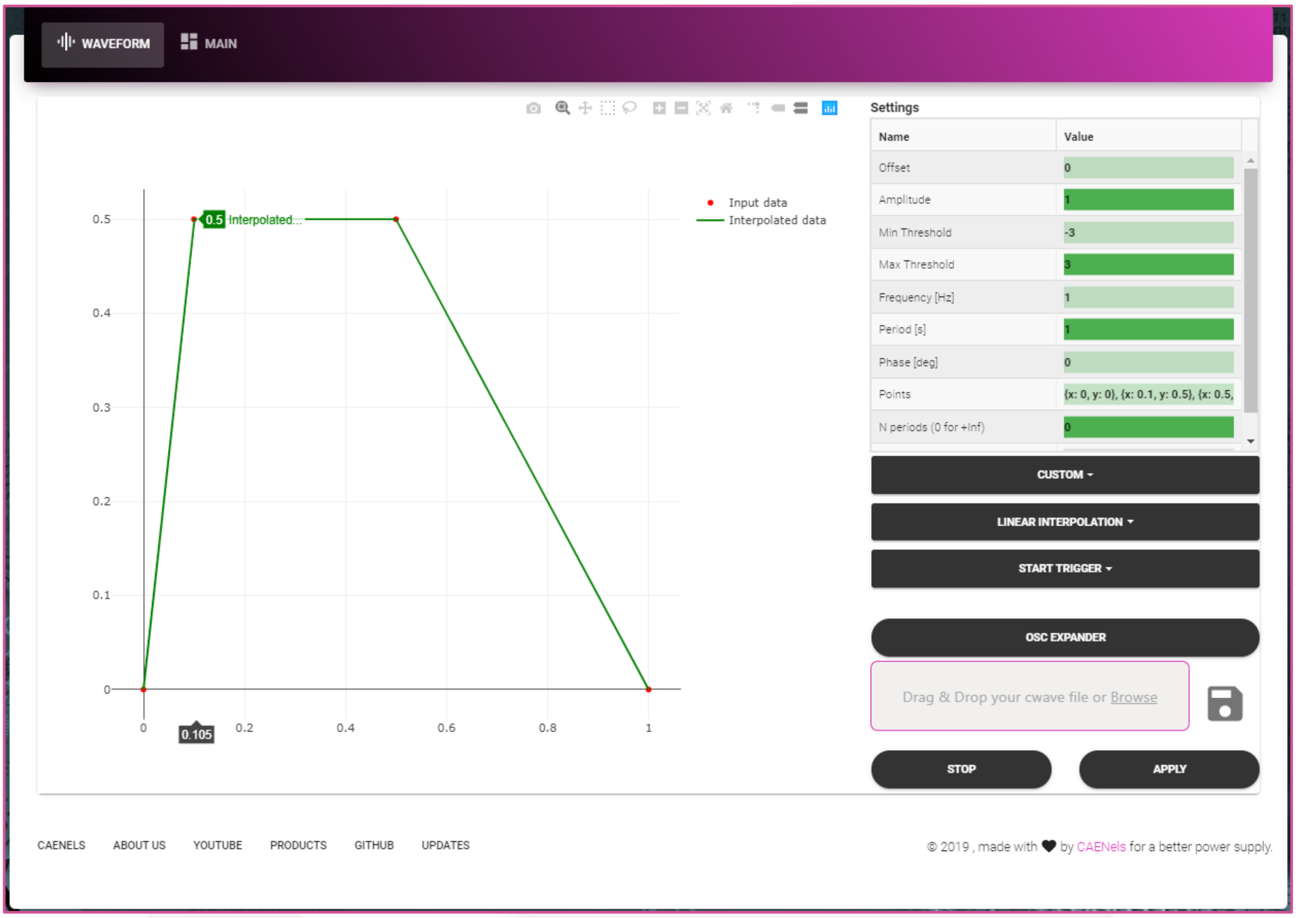


Figure 11: Custom Wave

The custom waveform can be uploaded to the power unit by preparing a text file (e.g. .txt files) with following formatting:

{x: x0, y: y0}, {x: x1, y: y1}, … ,{x: xn, y: yn}

Where xi and yi are respectively time and value of each point, and n is a number lower than 500’000. The file can then be dragged and dropped in the purple framed rectangle in **Figure 11**.

Different interpolations of the points are then available:

- Linear
- Step
- Polynomial
- Spline

. Once the wanted waveform has been described, click “APPLY” in order to reproduce it and “STOP” to stop it.

In addition, two trigger options are present, if needed:

- START TRIGGER: the entire waveform will be reproduced once the trigger signal will be detected. Trigger must be set in order to do so (see section 3.1.3).
- POINTS TRIGGER: the waveform will be reproduced point by point per each trigger signal detected. Trigger must be set in order to do so (see section 3.1.3).

"OSC EXPANDER" may be used in order to see in parallel the waveform window, the oscilloscope one (see section 3.4), and to also have access to the PID settings (see section 3.3.2), in order to modify in real time the dynamic of the module and to observe the response changing.

3.1.4.3 *Analog (Voltage Input)*

Analog control mode once selected is shown in **Figure 12**:

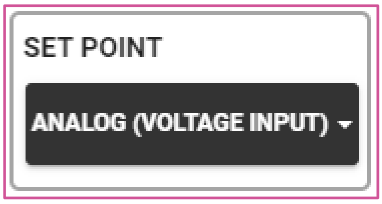


Figure 12: Analog Control

When in analog control mode, the control module input accepts signals ranging from -10V to +10V (for bipolar units) or from 0V to +10V (for unipolar units) and it generates an output which is proportional to the input signal and the input voltage and current full-scale value.

An example of the relation between the analog input signal and the output (can be either current or voltage, depending on the Regulation mode) is shown in **Figure 13**:

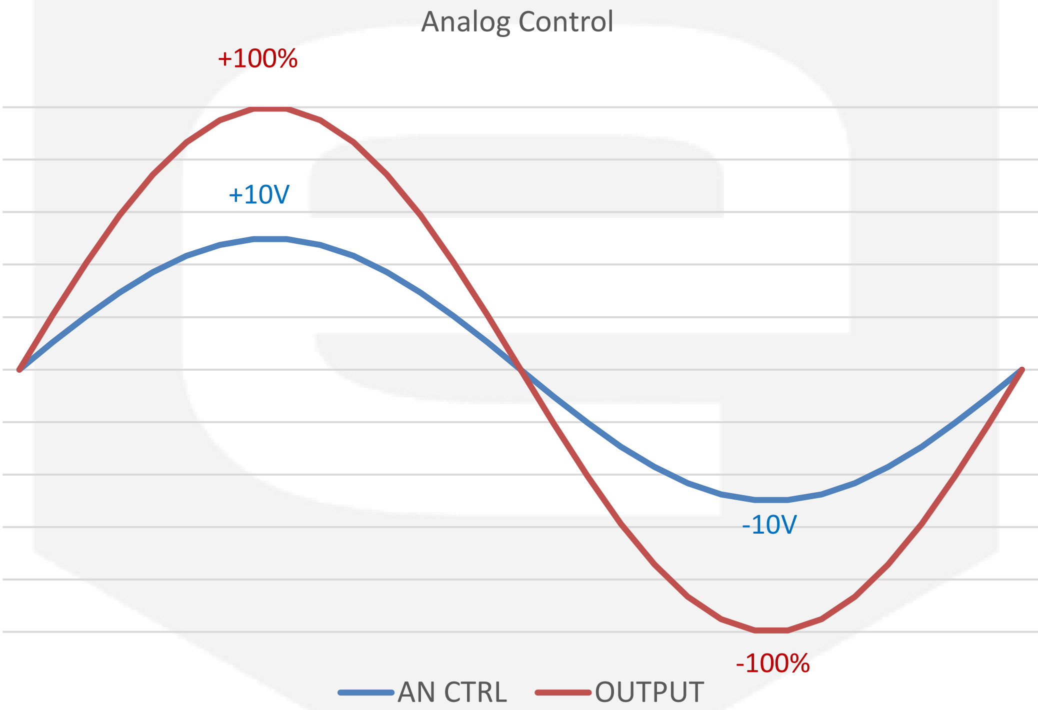


Figure 13: Analog Control – Output Relation

In order to avoid drifts, offset and external noise pick-up it is always suggested to use the digital interface – e.g. Ethernet – to control the device in order to get the best performance.

Please note that the bandwidth of the analog control input is internally limited to 1 kHz.

3.2 EPICS IOC

An on-board EPICS IOC server running on the control device is present and the main window is presented below. Protocols shows the current EPICS server settings:

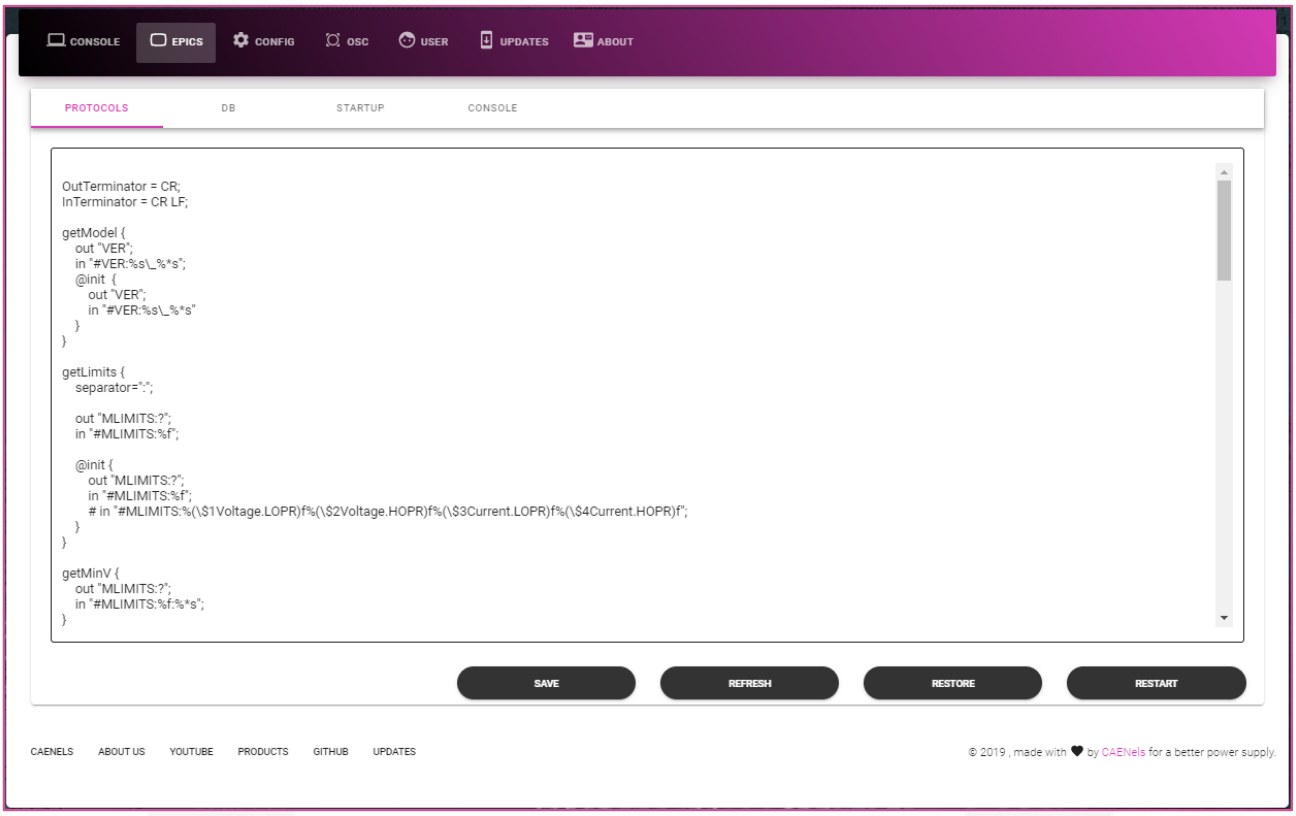


Figure 14: EPICS - Main Window

EPICS PV (Process Variables) can be changed in "DB" window:

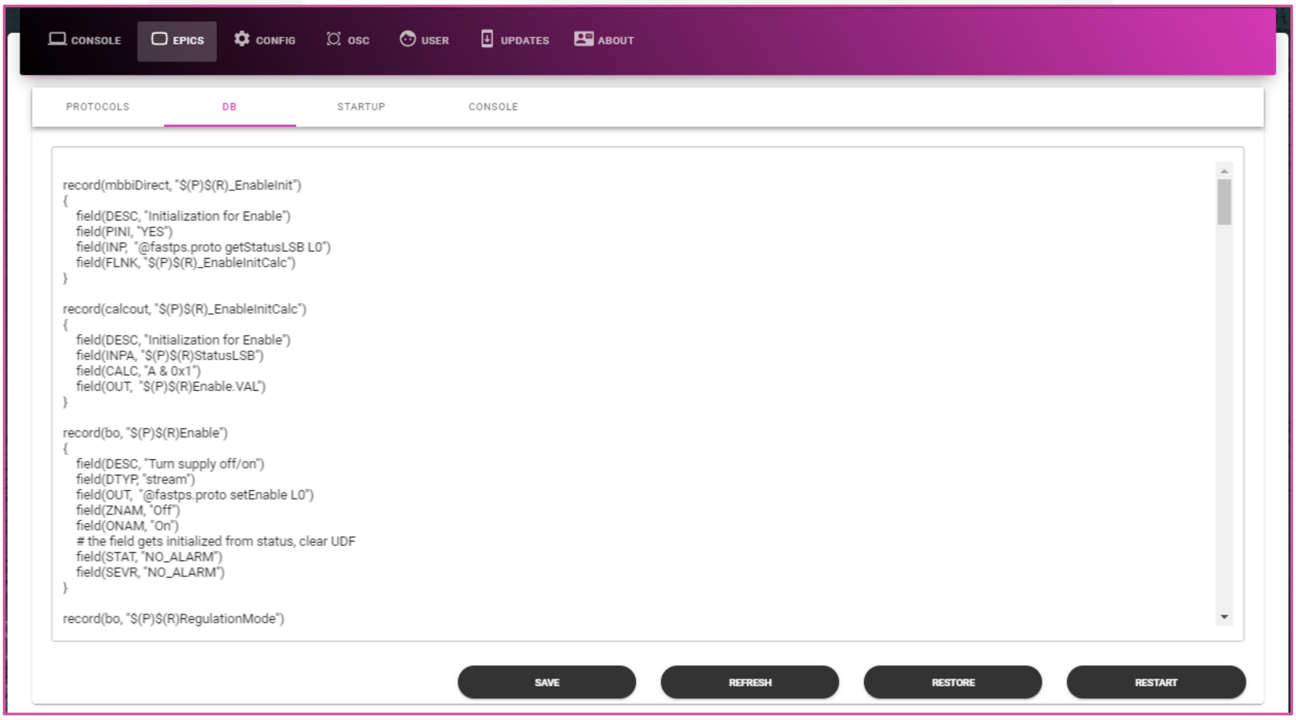


Figure 15: EPICS - DB

After the editing has been performed, click "SAVE" to store the current configuration.

EPICS records can be changed in "STARTUP" window:

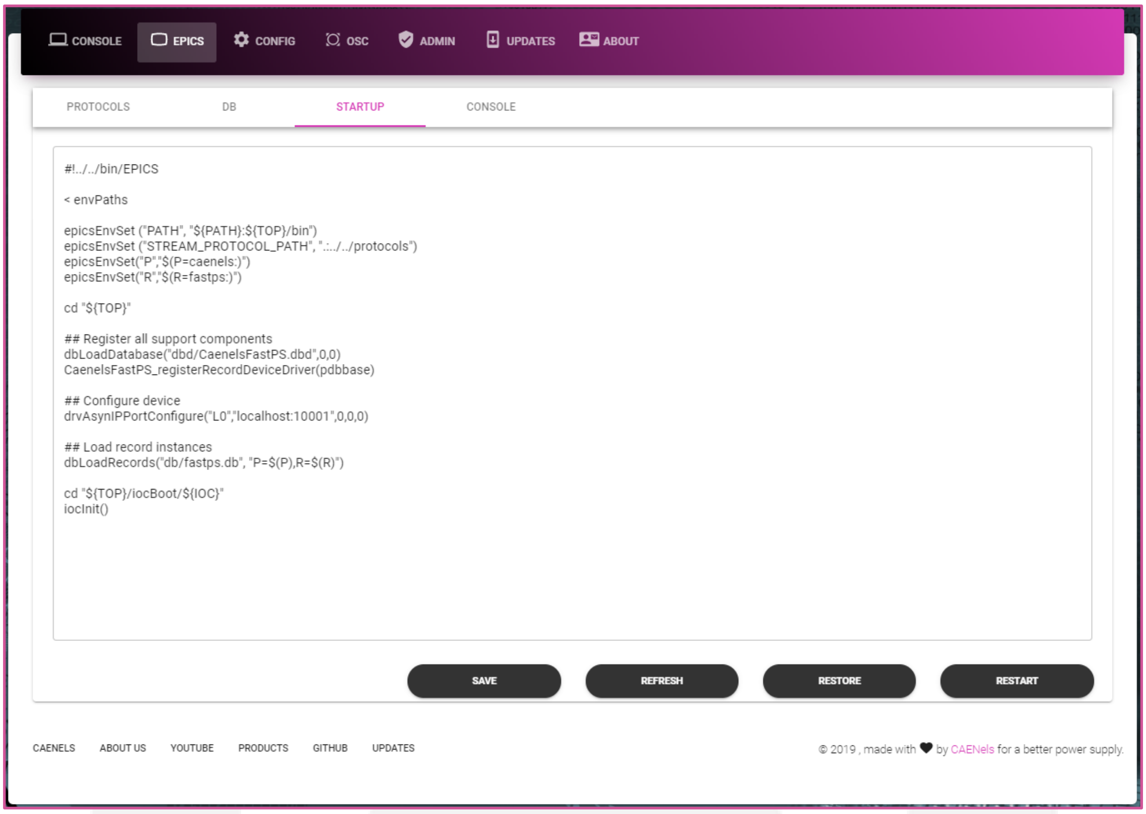


Figure 16: EPICS – STARTUP

In "CONSOLE" window EPICS-supported commands may be typed in order to get commands' replies (e.g. by typing "dbl" the power source will reply with the current EPICS-IOC configuration of records and variables).

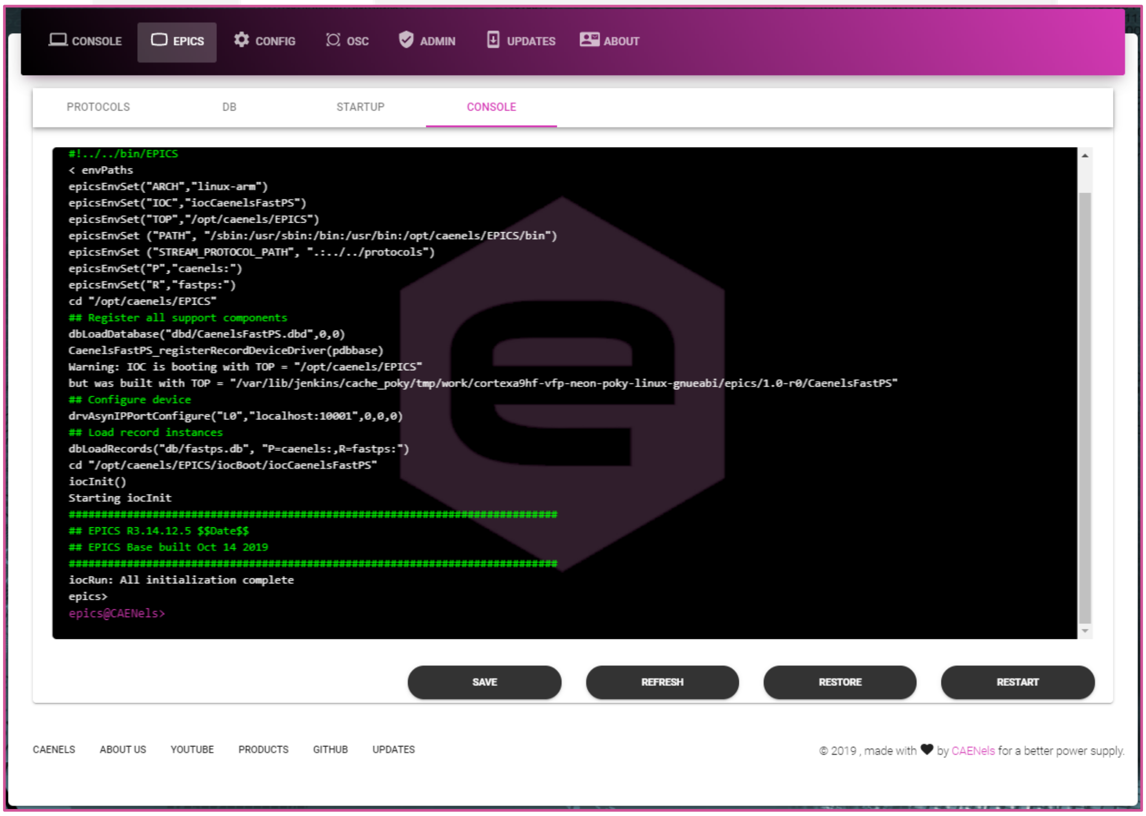


Figure 17: EPICS – CONSOLE

3.3 Unit Configuration

To display the configuration Window, click on the "CONFIG" button on the main Toolbar. From this window, it is possible to configure the unit. Several fields are password protected. Please remember to set "Ps-admin" as password at the login in order to have administrator privileges.

The "CONFIG" window is presented below:

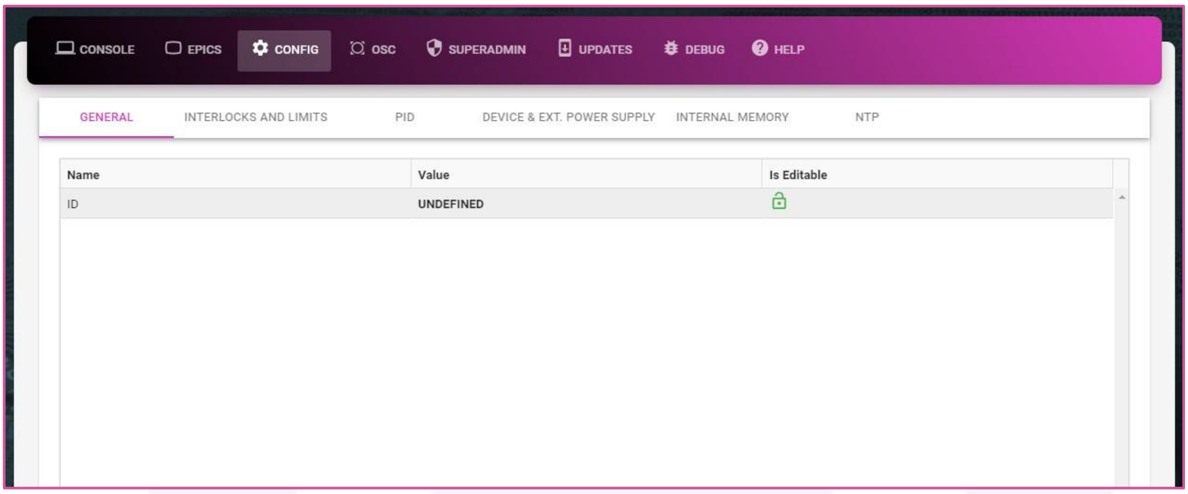


Figure 18: Web Server – General Section

- **General tab:** shows the serial number of the module.
- **Interlock and Limit tab:** from this tab the user can:
  - display and edit the names, direction, status (enabled or disabled) and intervention time of the available external interlocks,
  - visualize and edit the module limits (which generate a fault condition) as: the maximum allowable temperature, min DC-link voltage, Leakage current limit etc.
- **PID tab:** from this tab it is possible to edit the PID regulators parameters – i.e. proportional, derivative and integral terms.
- **Internal Memory tab:** in this tab the user can display, edit and save the content of the unit memory.
- **Ext Device tab**: in this tab the user can set:
  - Current, voltage and power of the power supply connected to the control device.
  - Input current range.
  - Input voltage range.
  - Output voltage DAC range.

- Status signals trigger signal.

3.3.1 Interlocks and limits

The "INTERLOCK AND LIMITS" window is divided into 3 sub-windows:

- External Interlock Setup
- Limits
- Regulation Fault

In the following 3 paragraphs each of these 3 sub-windows is explained.

3.3.1.1 *External Interlock Setup*

To set interlocks levels, at first click on the "External Interlock Setup" window:

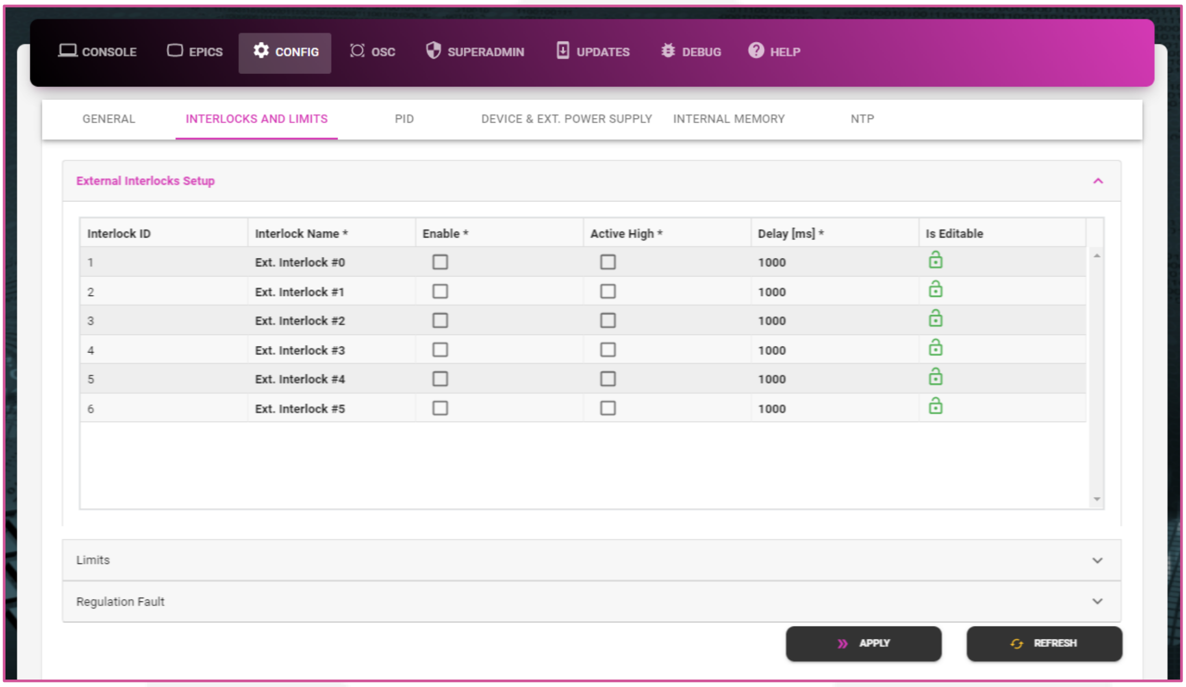


Figure 19: Web Server – External Interlock Setup

Administrator privileges are needed for interlocks editing, please refer to login instructions at page 12 of this Manual.

It is now possible to enable interlocks (checking the "Enable" box), to set the high or low level and the intervention time (delay).

For interlocks enabled at high level, fault will appear when the interlock pin is shorted with the common pin, for the ones enabled at low level, fault will appear when the interlock pin and the common pin are in open loop.

More information regarding the number of interlocks available on the power supply and the interlock pinout are available in the specific power source Manual.

When the interlock configuration has been changed, click "APPLY" in order to save it.

3.3.1.2 *Limits*

Under "Limits" section, different important parameters are listed, and it is recommended to not change such values:

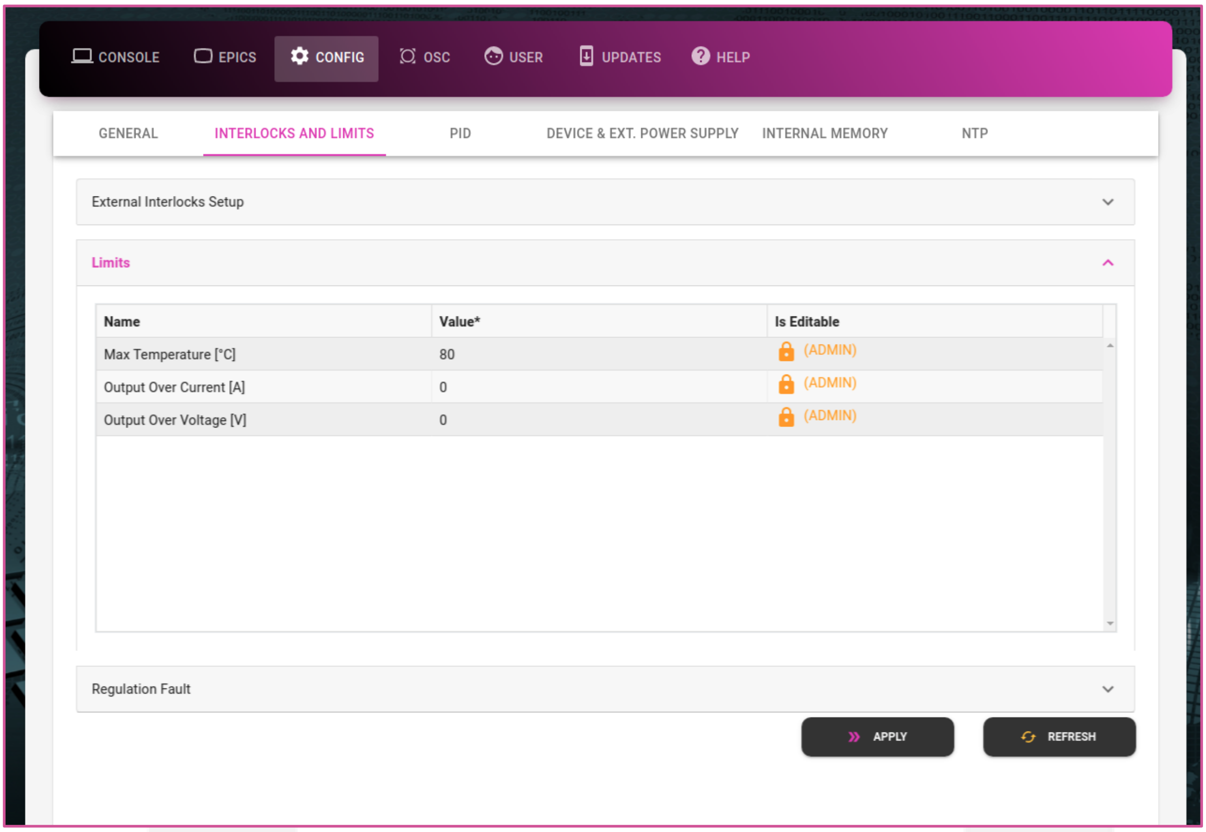


Figure 20: Web Server – Limits Section

The meaning of these limits is:

- Max Temperature: max temperature allowed for the ADCs
- Output Over Current:
- Output Over Voltage:

If the machine reads a value which is not within these limits, a fault will be generated.

3.3.1.3 *Regulation Fault*

Whenever the unit is not able to reach a set point, either a voltage one or a current one, a "Regulation Fault" will appear.

E.g., if the user sets a 15A set-point on a 2 Ohm load and the maximum voltage output is 20V, a "Regulation Fault" will appear as soon as the power source senses that it is impossible to reach the set-point.

Under "Regulation Fault" section the following limits are listed:

- Current Regulation Fault Limit: maximum difference between the current set-point and the current which can be reached.
- Voltage Regulation Fault Limit: maximum difference between the voltage set point and the voltage which can be reached
- Regulation Fault Intervention Time: maximum time allowed to reach the set-point

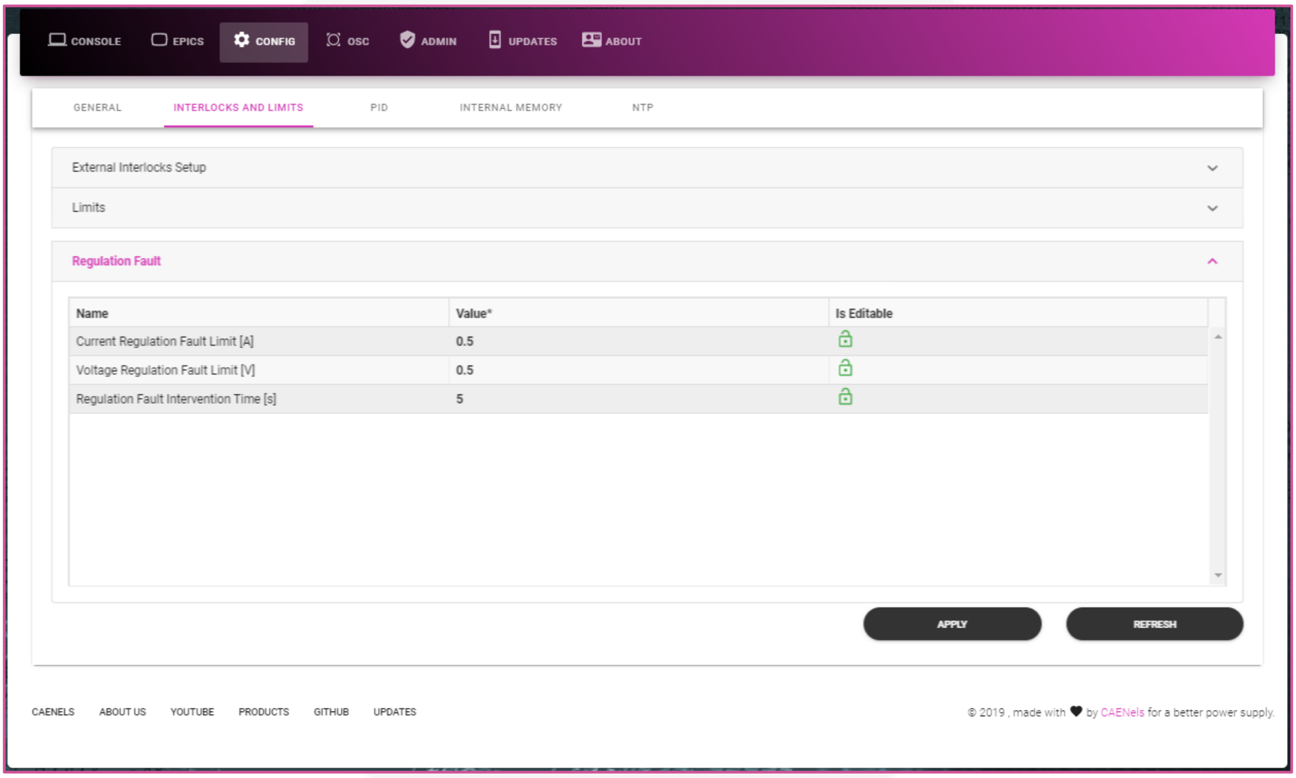


Figure 21: Web Server – Regulation Fault Section

In this case also it is recommended to not change the factory values. However, this may be useful in specific applications (e.g., high inductive loads with high time constants).

3.3.2 PID

CAEN ELS power supplies run a digital control loop instead of the more commons analog ones based on hardware components.

The PID loops run on FPGA logic, this allowing very fast dynamics, while the possibility of changing the P, I, D parameters give infinite possibilities of use.

PID parameters are found on the “CONFIG” Window under “PID” section:

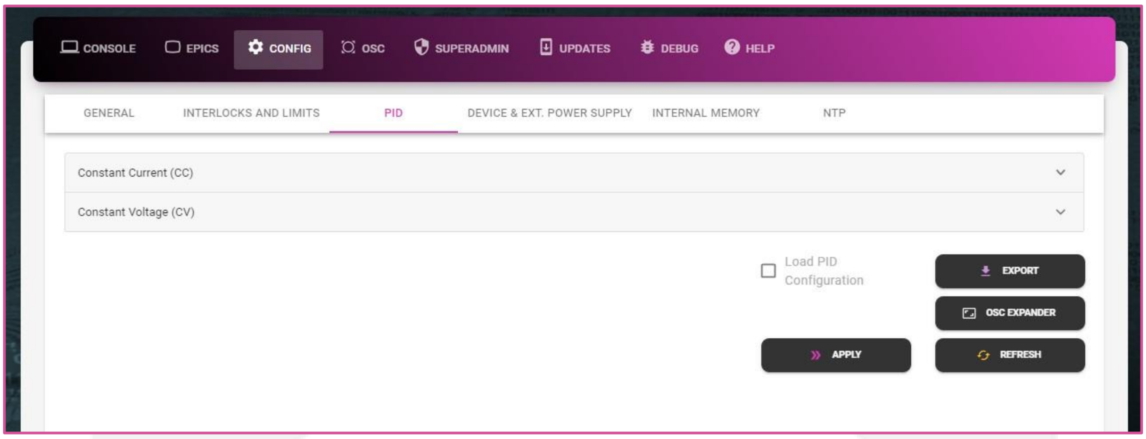


Figure 22: Web Server – PID section

PID parameters must be set separately for current control (Constant Current, CC) mode or for voltage control (Constant Voltage, CV) mode, the PID architecture is shown in **Figure 23**.

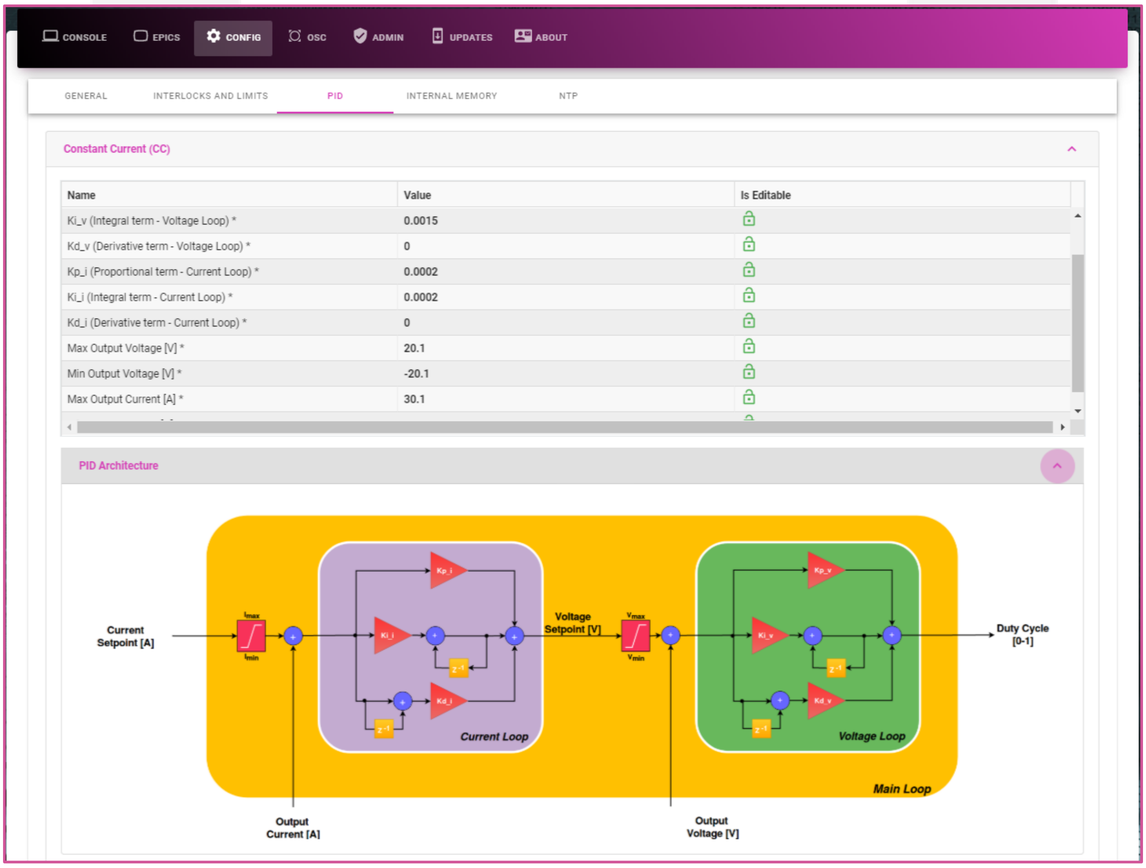


Figure 23: Web Server – PID section (CC) and PID architecture

3.3.2.1 *Save and use pre-set PID configurations*

In order to save and load pre-set configurations, a folder on the PC in use must be created.

Indeed, at the next run of the Visual Software, configurations will be searched in the last folder used to store the PID configurations.

PID configurations are stored as .json files (readable by mean of any text editor).

To export a configuration, under "Configuration Window" open "PID"; in the bottom, click "Export":

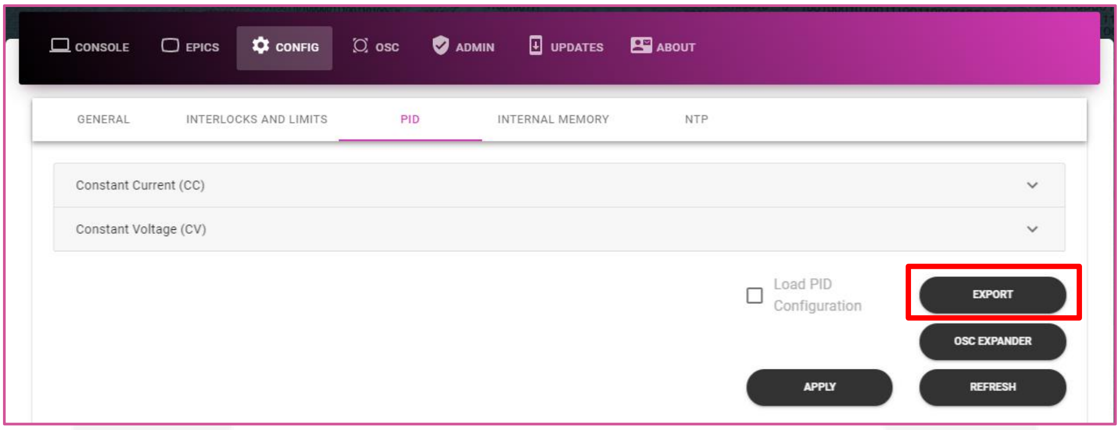


Figure 24: Web Server – export PID settings

The configuration can now be saved and easily accessed in the future. <u>It is recommended to save the file using a name which refers to the load in use and/or to the dynamic achieved.</u>

To use a stored configuration, drag and drop the .json file in the window under "Use preload PID's parameters" (the box needs to be set):

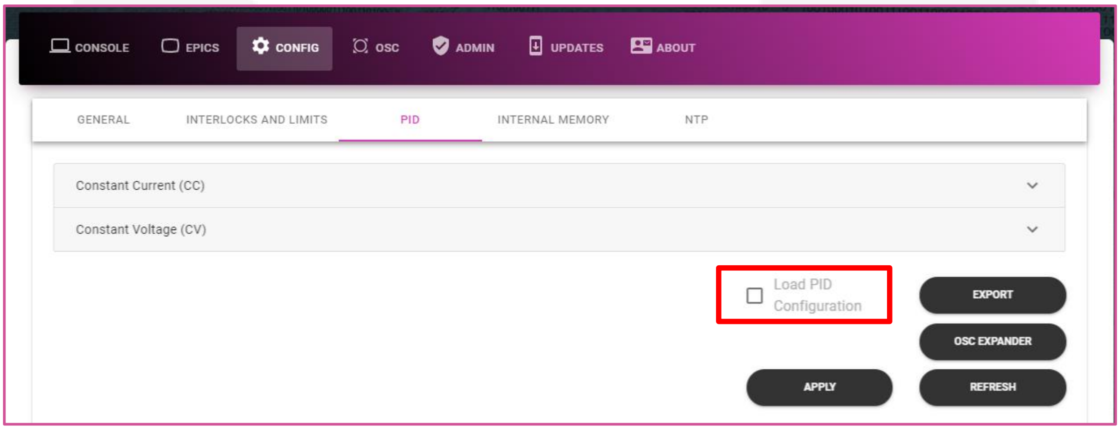


Figure 25: Web Server – import PID settings

<u>**Once the configuration is selected, it is automatically stored and saved into the module by clicking "APPLY".**</u>

3.3.3 Internal memory

As the name suggests, in this window the internal memory cells are listed:

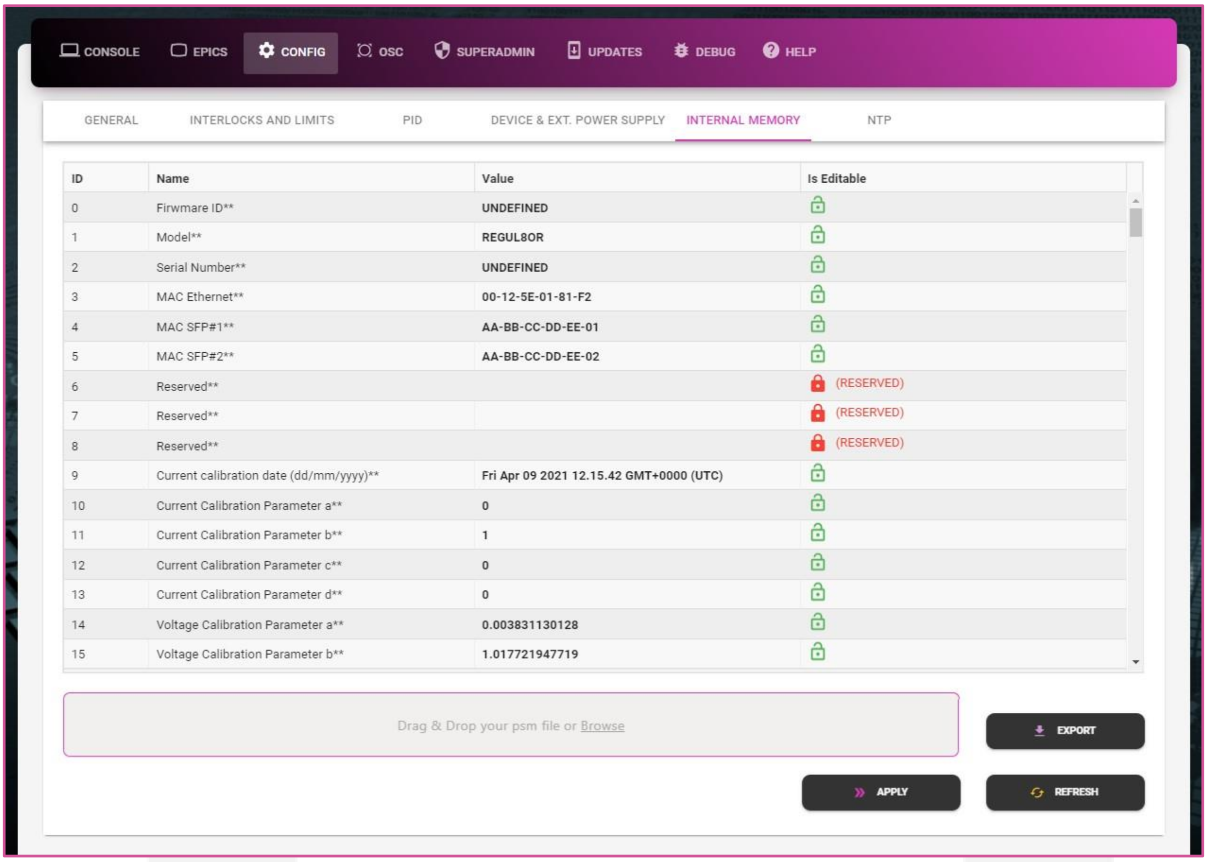


Figure 26: Web Server – internal memory

The internal memory is mapped differently depending on the power source family; in the following pages the mapping for the Regul8OR is reported.

Once in the “INTERNAL MEMORY” window, the user is allowed to change the memory cells; in order to do so, it is required to overwrite the values and to click “APPLY”.

By clicking “EXPORT”, a .psm file listing the memory mapping can be exported, and later re-imported by dragging and dropping it in the purple framed rectangle in **Figure 26**: Web Server – internal memory **Figure 26**

Parameter #	Access Privileges	Parameter Name
#0	Read Only	Firmware ID
#1	Read Only	PS Model
#2	Read Only	Serial Number
#3	Read Only	MAC Ethernet
#4	Read Only	MAC SFP #1
#5	Read Only	MAC SFP #2
#6 - #8	/	Reserved
#9	Read Only	Current calibration date
#10	Read Only	Current Calibration Parameter a
#11	Read Only	Current Calibration Parameter b
#12	Read Only	Current Calibration Parameter c
#13	Read Only	Current Calibration Parameter d
#14	Read Only	Voltage Calibration Parameter a
#15	Read Only	Voltage Calibration Parameter b
#16	Read Only	Voltage Calibration Parameter c
#17	Read Only	Voltage Calibration Parameter d
#18	Read Only	Voltage calibration date
#19	Read Only	Current ADC gain
#20	Read Only	Voltage ADC gain
#21	/	Reserved
#22	/	Reserved
#23	/	Reserved
#24	Read Only	Analog Input Calibration Parameter a
#25	Read Only	Analog Input Calibration Parameter b
#26-#29	/	Reserved
#30	Admin	Module Identification
#31	Admin	Default Current Slew Rate [A/s]
#32	Admin	Default Voltage Slew Rate V [V/s]
#33-#4	/	Reserved
#35	Admin	Turning off time of Display [minutes] – 0 to disable
#36 - #39	/	Reserved
#40	Admin	PID V: Kp_v
#41	Admin	PID V: Ki_v
#42	Admin	PID V: Kd_v
#43	Admin	PID I: Kp_i
#44	Admin	PID I: Ki_i
#45	Admin	PID I: Kd_i
#46	Admin	PID V: Voltage Upper Limit
#47	Admin	PID V: Voltage Lower Limit
#48	Admin	PID I: Current Upper Limit
#49	Admin	PID I: Current Lower Limit

Parameter #	*Access Privileges*	*Parameter Name*
#50 - #54	/	Reserved
#55	Admin	Allow Remote OFF command
#56	Admin	Error Code Description (Disable (0)/Enable(1))
#57 - #79	/	Reserved
#80	Admin	Output Over-Current Limit [A]
#81	Admin	Output Over-Voltage Limit [A]
#82	Admin	Max Temperature ADCs
#83 - #85	/	Reserved
#86	Admin	Current Regulation Fault Limit [A]
#87	Admin	Voltage Regulation Fault Limit [A]
#88	Admin	Regulation Fault Intervention [s]
#89	/	Reserved
#90	Admin	Interlock Enable Mask
#91	Admin	Interlock Activation State
#92	Admin	Interlock #1 intervention time [ms]
#93	Admin	Interlock #1 name
#94	Admin	Interlock #2 intervention time [ms]
#95	Admin	Interlock #2 name
#96	Admin	Interlock #3 intervention time [ms]
#97	Admin	Interlock #3 name
#98	Admin	Interlock #4 intervention time [ms]
#99	Admin	Interlock #4 name
#100	Admin	Interlock #5 intervention time [ms]
#101	Admin	Interlock #5 name
#102	Admin	Interlock #6 intervention time [ms]
#103	Admin	Interlock #6 name
#104	/	Reserved
#105	Admin	Input Voltage Range
#106	/	Reserved
#105	Admin	Input Current Range
#109-#114	/	Reserved
#115	Admin	Signal Status Polarity Mask
#116	Admin	Signal Status Event Qualifier
#117-#175	/	Reserved
#176	Admin	DAC Output Range
#177	Admin	DAC Calibration offset
#178	Admin	DAC Calibration Gain
#179	Admin	DAC Output full-scale
#180	Admin	DAC Gain
#181-#199	/	Reserved

Parameter #	Access Privileges	Parameter Name
#200	Admin	Is the power supply monopolar?
#201	Admin	PS max output voltage [V]
#202	Admin	PS Max output current {A]
#203	Admin	PS Max output power [W]
#204	Admin	PS Max absorbed power [W]
#205	Admin	DAC Gain
#206	Admin	DAC Gain
#205-#299	/	Reserved
#300	Admin	Input ADC max current
#301	Admin	Input ADC max voltage

Table 1: Parameters table for REGUL8OR

3.3.4 NTP – Network Time Protocol

The Network Time Protocol (NTP) is an internet protocol intended for the synchronization (under Coordinated Universal Time - UTC) of several clients in a client-server architecture.

In such architecture the power source acts as client, and an NTP server must be available in order to achieve synchronization.

The NTP server can run in local on the PC in use (in this case several guides online can be found, depending on the OS in use) or the NTP server can be reached by Internet (e.g. refer to pool.ntp.org).

Once the IP of the NTP server is known, insert it in the NTP window in the CAEN ELS web server:

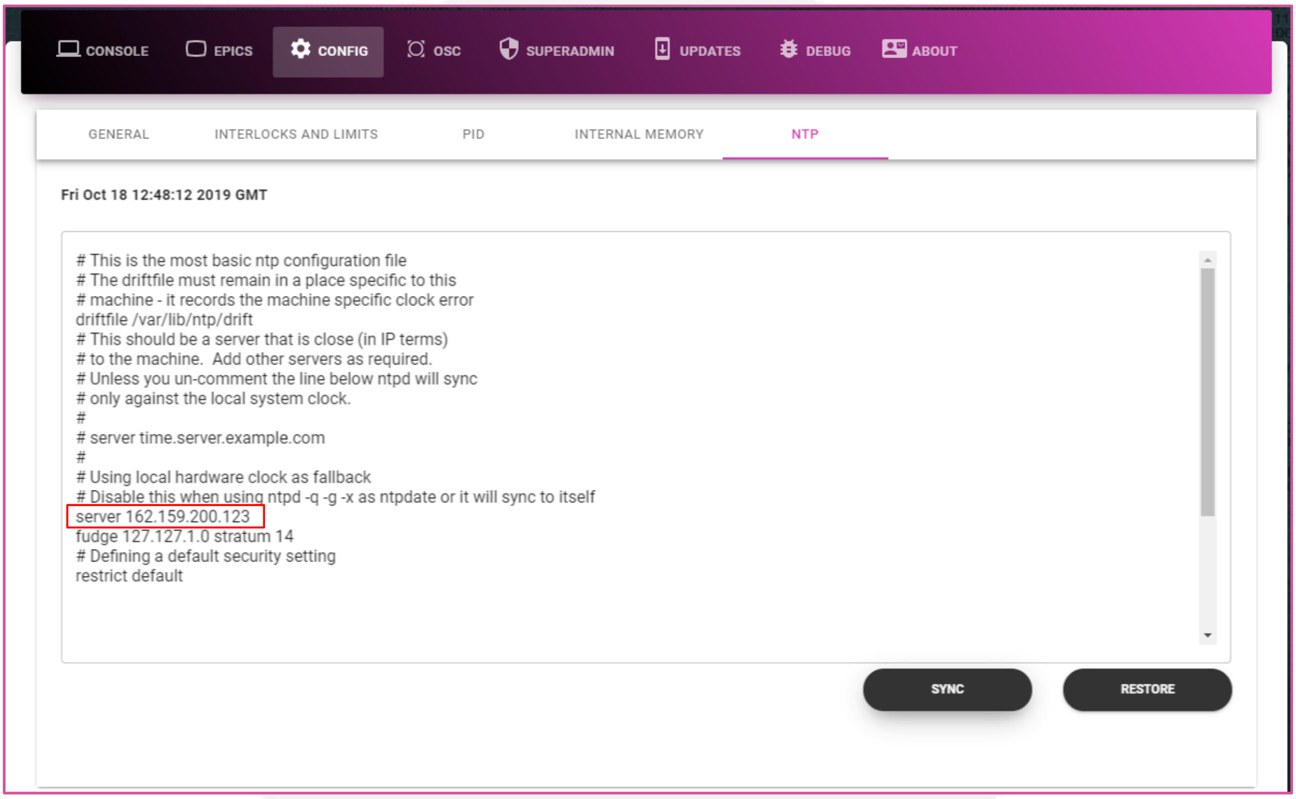


Figure 27: Web Server – NTP server

When the configuration has been set, click "SYNC" in order to synchronize the power module to the NTP server.

3.3.5 Device & External Power Supply

The “EXTERNAL DEVICE” window is divided into 3 sub-windows:

- External Device
- Input current and voltage
- Output DAC
- Signal Status

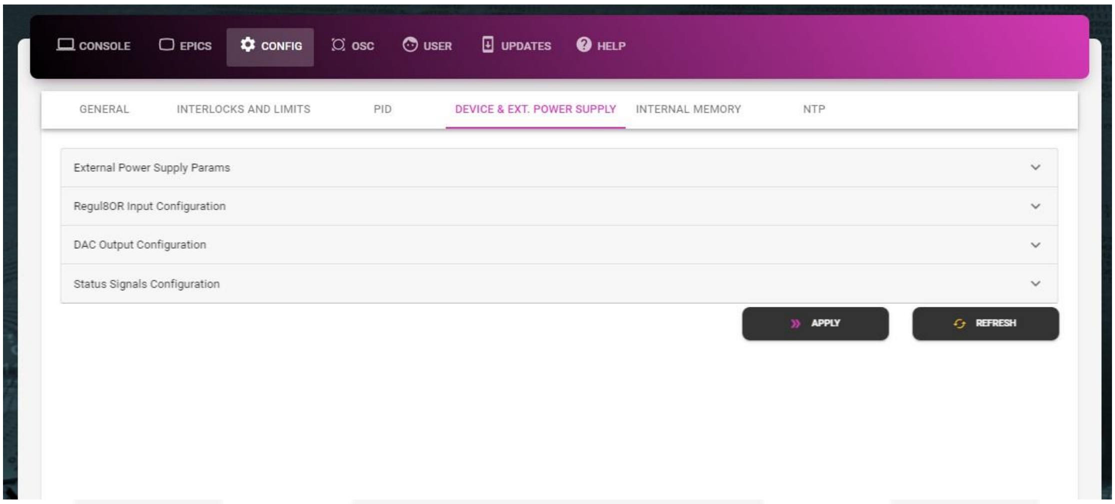


Figure 28: Web Server - Device & Ext. Power Supply

3.3.5.1 *External Power Supply Parameters*

In the External Device sub-window, it is possible to set the current, voltage and power of the external power supply connected to the Regul8OR. These operations can be performed by writing the value into the corresponding text fields.

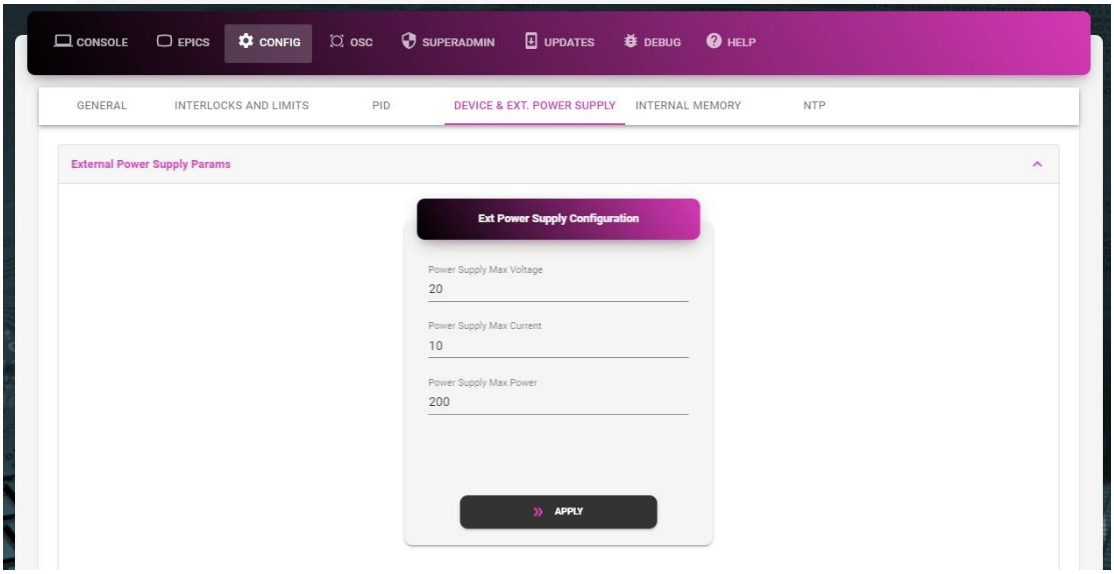


Figure 29: Web Server - External Power Supply Parameters

3.3.5.2 *Input Current and Voltage*

Current and voltage ranges may be set in the Input Current and Voltage sub-window by selecting the right calibration file. The Regul8OR unit has the voltage input ranges properly calibrated and already available, while the current input path depends from DCCT(s) purchased with the unit.

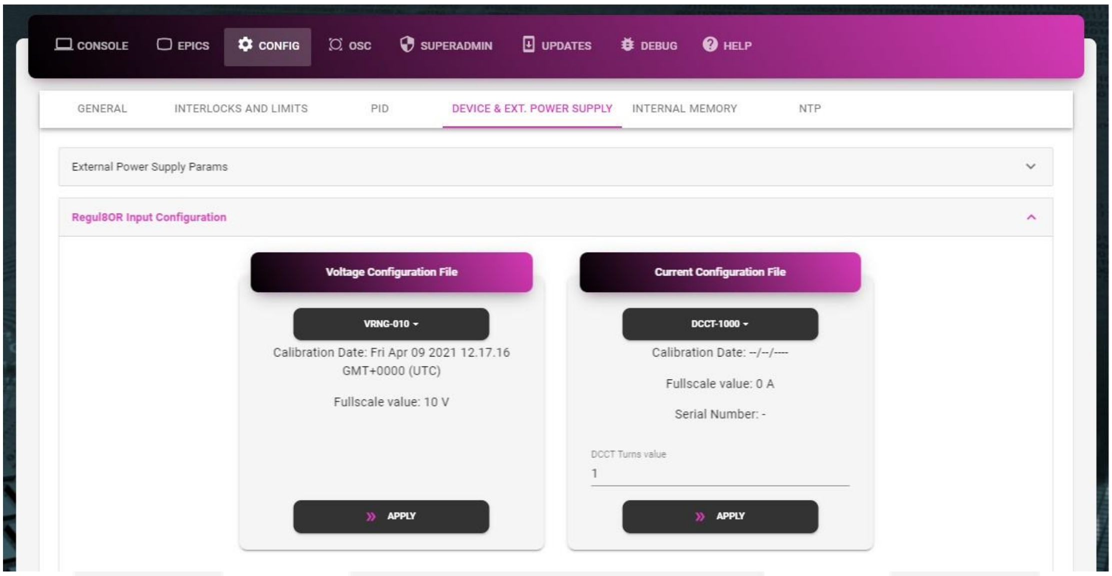


Figure 30: Web Server - Regul8OR Input Configuration

3.3.5.3 *Output DAC*

In the Output DAC sub-window, the user may set the voltage output DAC range by selecting the right calibration file. In order to control a wide range of power supplies it is possible to choose two ranges: 5 V or 10 V.

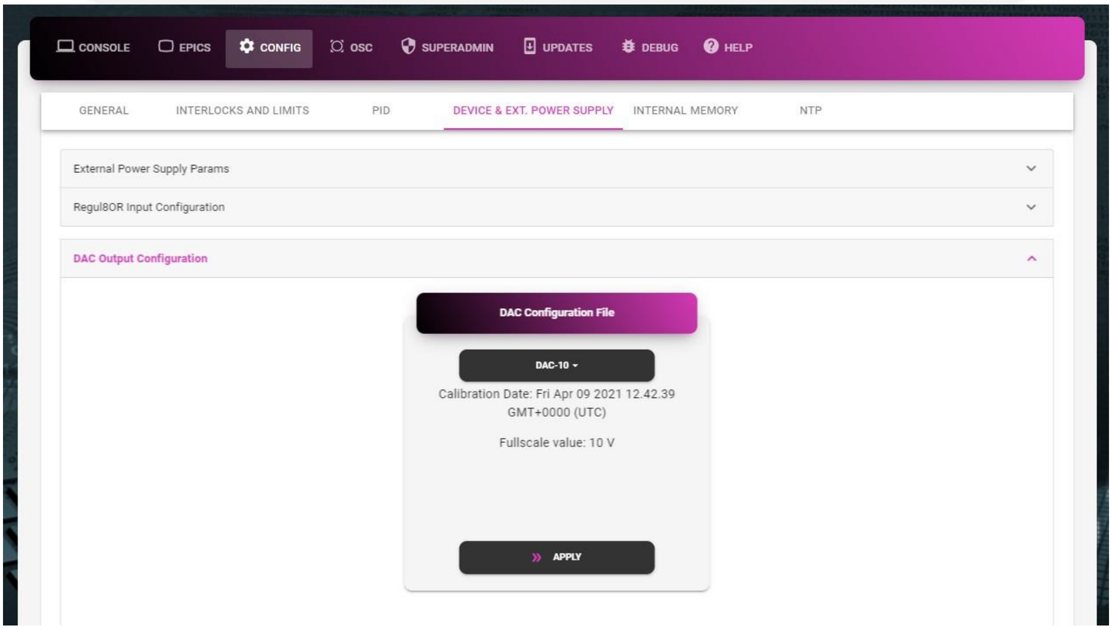


Figure 31: Web Server - DAC Output Configuration

3.3.5.4 *Signal Status*

In the Signal Status sub-window, it is possible to choose the trigger signal for each status signals (6). The status signals are divided as follow:

- 2 magnetic relays (NC-C-NO)
- 2 dry contact
- 2 TTL signals

The trigger signals are:

- Interlocks
- Faults
- Module ON

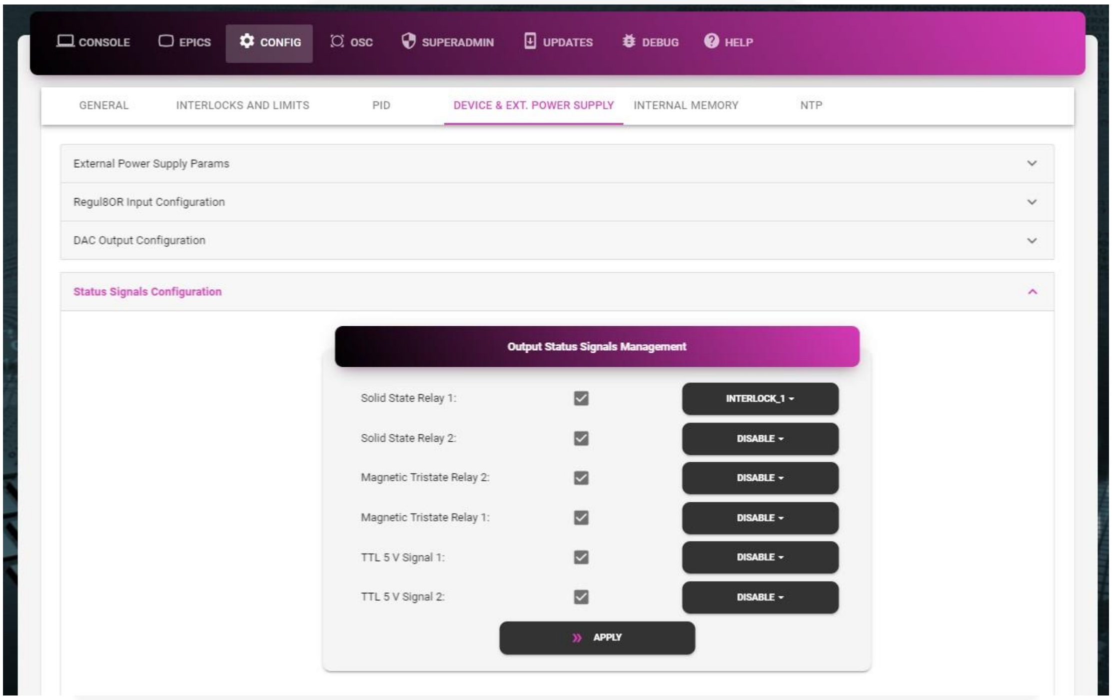


Figure 32: Web Server - Status Signal Configuration

3.4 Oscilloscope

The web server also allows to access the on-board oscilloscope, plotting in real time and at 100 kHz, via DMA (Direct Memory Access), relevant information:

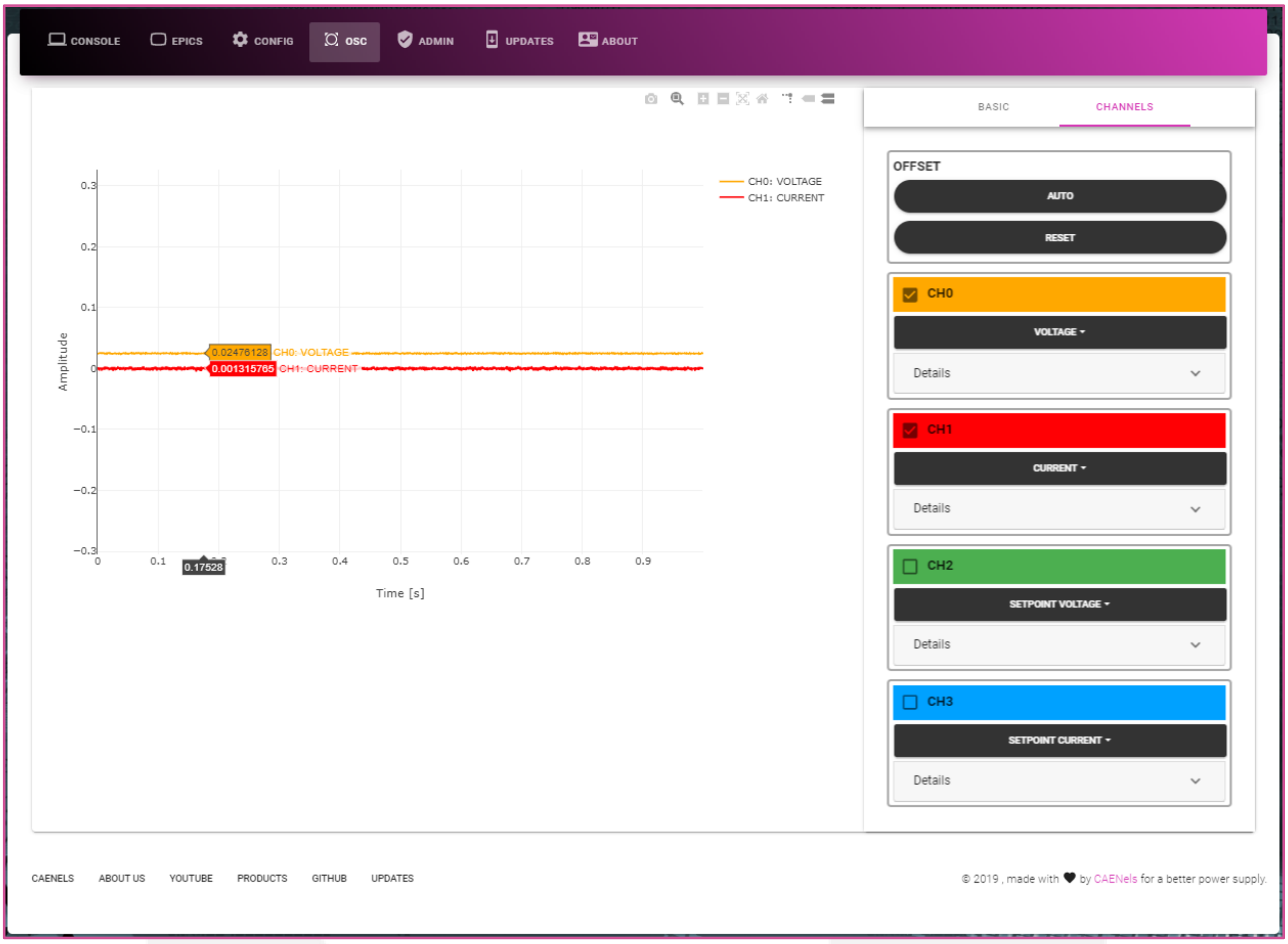


Figure 33: Web Server – Embedded Oscilloscope

Above the oscilloscope plotting several icons are present:

Figure 34: Web Server – Embedded Oscilloscope – zooming features

These icons provide different features, from left to right:

- Save plot as .png file
- Zoom (creating a zoom region with mouse cursor)
- Zoom in
- Zoom out
- Autoscale
- Reset axes
- Toggle spike lines (it correlates X and Y points for easier oscilloscope reading)
- Show closest data (shows time,value per each channel when passing on the channels plotting with mouse cursor)
- Compare data (shows valueCH1,…,valueCHn,time when passing on the channels plotting with mouse cursor)

On the upright corner in **Figure 33** two windows are present: “BASIC” and “CHANNELS”, here below discussed.

3.4.1 Channels

The embedded Oscilloscope allows to plot up to 4 values, to be decided within:

- Current set-point
- Voltage set-point
- Current actual values
- Voltage actual values

In addition, two offset functions are available by clicking on the related widgets:

- AUTO: Sets all the channels to zero
- RESET: Re set the real values for each channel

3.4.2 Basic

Here the user can set different values and parameters related to the oscilloscope operation and triggering options:

Figure 35: Web Server – Embedded Oscilloscope – BASIC window

Four different operation modes are available from the “MODE” window:

- AUTO: continuous real time plotting
- NORM: used in trigger mode, oscilloscope waits for trigger signal. It plots each trigger.
- SINGLE: used in trigger mode, oscilloscope waits for trigger signal. It plots only the first trigger.
- STOP: stops plotting

"HORIZONTAL" window allows to change the time-width of the oscilloscope plotting.

"TRIGGER"* window is used to edit all the values related to trigger modes (NORM and SINGLE):

- SOURCE (CH1, CH2, CH3 or CH4)
- EDGE (POS, NEG, BOTH)
- Level
- Position

* The trigger must be hardware installed (it is optional on FAST PS and FAST PS M families). If installed, it will be present as "TRIG" in General Information - Capabilities (**Figure 5**).

3.5 Firmware Update & Support

"UPDATES" and "HELP" windows provide additional support and information to end-users such as firmware update and connection to CAEN ELS website as well as contacts for maintenance of power sources when required.

In order to update the unit's firmware, follow the simple instructions reported in **Figure 36**. Once the .updt file has been dropped, the power source will automatically start the upgrade procedure.

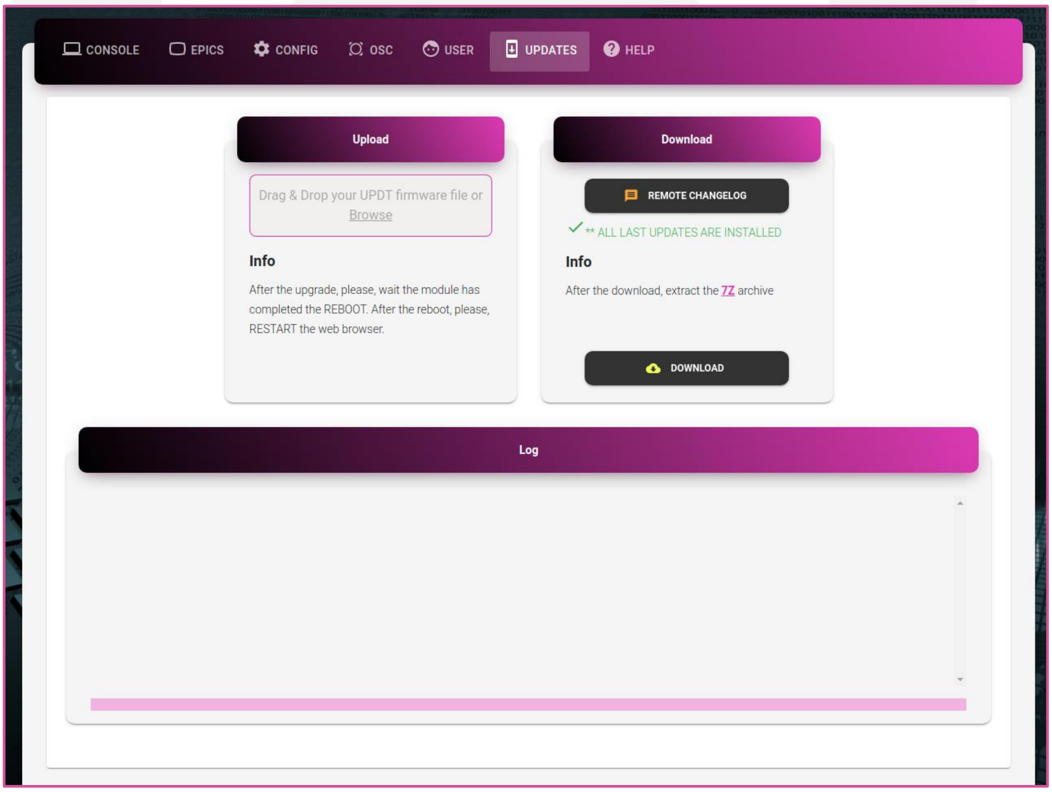


Figure 36: Web Server – "UPDATES"

4. Software Commands

This chapter describes the base TCP/IP software commands used for the control and configuration of the power module.

4.1 Ethernet Interface

The device is shipped with default IP address, subnet mask, and gateway and TCP-IP communication port:

Parameter	*Factory Value*
IP address	192.168.0.10
Subnet mask	255.255.255.0
Gateway	192.168.0.1
TCP/IP port	10001

Table 2: Default Ethernet Settings

4.2 Command Syntax

The command syntax used by the protocol is described in the following sections.

Commands must be sent in ASCII format and are composed by a *"command field"* and one, two or none *"parameter field"*, separated by a colon (*"**:**"* or *"0x3A"* in hexadecimal notation). The number of *"parameter fields"* depends on the specific command. Commands are **NOT case sensitive** and therefore the command string can be sent either using uppercase or lowercase characters (conversion to uppercase characters is performed internally). Each command must be terminated with the termination sequence. The protocol supports two termination sequences:

- *"carriage return"* termination char "**\r**" (*"0x0D"* in hexadecimal notation or commonly CR) or
- "*carriage return\line feed*" sequence "***\r\n***" ("*0x0D 0x0A"* in hexadecimal notation or commonly CRLF).

Command Example:

MWI:20.5580\r or ***MWI:20.5580\r\n***

- ***"MWI"*** is the command field;
- ***":"*** is the parameter's separation character;
- ***"20.5580"*** is the first parameter field;
- ***"\r"*** or ***"\r\n"*** are the termination sequences of the command.

In the following command description the ***"\r"*** termination char is used, but it can be always replaced with the termination sequence ***"\r\n"***.

Commands are processed one at a time; therefore, **user must wait for a response from the unit before sending the next command**.

All the responses from the FAST-PS module are in upper case and are terminated with the same *"carriage return\line feed"* sequence ("***\r\n***"), "*0x0D 0x0A"* in hexadecimal notation or commonly CRLF.

MWI:10.5875*\r\n* →

← **#ACK*\r\n***

or:

MWI:10.5875*\r* →

← **#ACK*\r\n***

4.3 Command Replies

The reply from the module depends on the given command. In general, the command can be grouped in two categories: Write commands and Read commands.

For **write commands** there are two specific replies that indicate that the command has been correctly elaborated or not. Those replies are hereafter presented:

- **A**c**K**nowledge ("***#AK***") indicates that the command is valid and it was correctly elaborated by the device:

#AK\r\n

- ***"#AK"*** is the **A**c**K**nowledged response to a valid command;
- ***"\r\n"*** is the termination sequence of the reply.

- **N**ot **A**c**K**nowledge ("***#NAK***") indicates that the command is either not valid or that it was not accepted by the device; the *"NAK"* reply is followed by an *"error code"* field, which can be used to determine the cause of the error (see the List of the Error Codes appendix, section 4.4, for a detailed list of all possible error codes):

#NAK:01\r\n

- ***"#NAK"*** is the **N**ot **A**c**K**nowledged response to an invalid command;
- ***":"*** is the parameter's separation character;
- ***"01"*** is the error code,
- ***"\r\n"*** is the termination sequence of the reply.

For **read commands**, the replies are generally formed by an echo string, followed by the corresponding read value. The echo string is preceded by the hash character ("**#**") and the echo is separated from the "**:**" separation character.

Some examples are hereafter shown:

MRI\r →

← **#MRI:12.8875\r\n**

or:

MWI:?\r →

← **#MWI:10.9850\r\n**

or:

MRG:90\r →

← **#MRG:90:0x2\r\n**

- the read commands are highlighted in **blue**;
- the echo string is highlighted in **green**;
- the read value is in **purple**;
- the termination char is highlighted in **red**.

For more detailed information about the single command please refer to the specific command section.

4.4 Error Table

The list of error codes returned with the ***#NAK*** reply and their description are hereafter shown:

Error Code #	*Description*
01	Unknown command
02	Unknown Parameter
03	Index out of range
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 bounds
11	Parameter is out of software 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 not set in local 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

Error Code #	*Description*
22	Float mode already set to desired value
23	Unknown sub-command for SFP command
24	Unknown feature or feature not available for actual module (AIN,TRIG)
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 FW version
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 (check internal memory)
99	Unknown error

Table 3: NAK Error code table

4.5 Command Table

The list of commands used within the communication protocol and the corresponding syntax is hereafter presented as well as a description of each command purpose and any special requirements related to the specific command. The base commands are summarized in Table 4.

Command	Read/ Write	Parameter #1	Parameter #2	Detailed description	Reply value
VER	R	/	/	Return the module model and installed firmware versions	ASCII indicating the module model and firmware version
MON	W	/	/	Turn on the module	“AK” or “NAK”
MOFF	W	/	/	Turn the module OFF	“AK”
LOOP	W R	“I” or “V” “?”	/ /	Set the power module loop mode Query for the power supply loop mode	“AK” or “NAK” Loop mode (“I” or “V”)
MSTR (*MST)	R	/	/	Read module internal status register	Internal status register (Hex representation)
MFTR (*MFT)	R	/	/	Read module internal fault register	Internal fault register (Hex representation)
MRESET	W	/	/	Reset the module status register	“AK” or “NAK”
MRI	R	/	/	Read output current value	ASCII indicating the output read current
FStatusMRV	R	/	/	Read output voltage value	ASCII indicating the output read voltage
MRW	R	/	/	Read output power	ASCII indicating the output read power
MRIA	R	/	/	Read instantaneous output current value	ASCII indicating the output read current
MRVA	R	/	/	Read instantaneous output voltage value	ASCII indicating the output read voltage

Command	Read/ Write	Parameter #1	Parameter #2	Detailed description	Reply value
MRWA	R	/	/	Read instantaneous output power	ASCII indicating the output read power
MRIO	R	/	/	Read output current offset	ASCII indicating the output read current
MRVO	R	/	/	Read output voltage offset	ASCII indicating the output read voltage
MRT	R	/	/	Read MOSFET Heatsink Temperature [°C]	ASCII indicating the temperature value
MGC	R	/	/	Read Earth Leakage current	ASCII indicating the Earth leakage current
MWV	W R	V Setpoint "?"	/ /	Set the new voltage setpoint (ASCII) Query for the last applied setpoint	"AK" or "NAK" ASCII indicating the voltage setpoint
MWVR	W R	V Setpoint "?"	/ /	Go to the given setpoint with a ramp (ASCII) Query for the last accepted final ramp setpoint	"AK" or "NAK" ASCII indicating the voltage setpoint
MWI	W R	I Setpoint "?"	/ /	Set the new current setpoint (ASCII) Query for the last applied current setpoint	"AK" or "NAK" ASCII indicating the current setpoint
MWIR	W R	I Setpoint "?"	/ /	Go to the given setpoint with a ramp (ASCII) Query for the last accepted final ramp setpoint	"AK" or "NAK" ASCII indicating the current setpoint
MSRI	W R	I Ramp Slew rate "?"	/ /	Set the I ramp slew rate [A/s] (ASCII) Query for the I ramp slew-rate	"AK" or "NAK" ASCII indicating the I ramp slew-rate
MSRV	W R	I Ramp Slew rate "?"	/ /	Set the I ramp slew rate [V/s] (ASCII) Query for the I ramp slew-rate	"AK" or "NAK" ASCII indicating the I ramp slew-rate

Command	*Read/ Write*	*Parameter #1*	*Parameter #2*	*Detailed description*	*Reply value*
MPLST	**R**	**/**	**/**	**Read Current, Voltage and Status simultaneously (mean values)**	**ASCII indicating the outputs readings and internal status register (Hex representation)**
MPLSTA	**R**	**/**	**/**	**Read Current, Voltage and Status simultaneously (instantaneous value)**	**ASCII indicating the outputs readings and internal status register (Hex representation)**
MRW	**R**	**/**	**/**	**Read estimated active output power value [W]**	**ASCII indicating the active output power value**
MRID	**R**	**/**	**/**	**Read module identification**	**Module identification (ASCII)**
HELP	**R**	**/**	**/**	**Print this command List**	**Command List**
MREBOOT	**W**	**/**	**/**	**Performs a reboot of the system**	**“AK” or “NAK”**
PASSWORD	**W** **R**	**Password word** **“?”**	**/**	**Set the password word (ASCII)** **Query for the actual user privileges**	**“AK” or “NAK”** **User privileges (ASCII representation)**
MRG	**R**	**Parameter field #**		**Read the given parameter field**	**Field content (ASCII)**
CHANGELOG	**R**	**“?”**	**/**	**Gets changelog information**	**ASCII text**
MFTRNAME	**R**	**/**	**/**	**Gets module fault register names**	**ASCII text fault information**
MLIMITS	**R**	**“?”**	**/**	**Gets voltage and current module limits**	**ASCII current and voltage limits**
MPLIMITS	**R**	**“?”**	**/**	**Gets power module limits**	**ASCII power limits**

Command	Read/ Write	Parameter #1	Parameter #2	Detailed description	Reply value
POL	R	"?"	/	Gets polarity inverter status	ASCII polarity
TRIG	W	"?" BOTH LEVEL NEG POS ON/OFF HIGH_LEVEL LOW_LEVEL	/ / HIGH/LOW (?) / / /	Set and monitor trigger options	"AK" or "NAK"
WAVE	W	KEEP_START N_PERIODS POINTS START STOP TRIGGER	/ Integer Floating array / / START/POINT	Set waveform generator	"AK" or "NAK"
UPMODE	W R	"?" AIN NORMAL WAVEFORM	/ / / /	Set the new update mode (ASCII) Query for the current update mode	"AK" or "NAK" ASCII indicating the current setpoint
MLCMDS	R	/	/	List of all available commands	ASCII indicating a list of all the available commands
INTPM	W R	Parameter field # Hex mask " ?"	0/1 / /	Sets/Gets interlock/s polarity mask	"AK" or "NAK"
INTEM	W R	Parameter field # Hex mask " ?"	0/1 / /	Sets/Gets Interlocks enable mask	"AK" or "NAK"
INTNAME	W R	Parameter field #	Name "?"	Sets/Gets Interlock name	"AK" or "NAK"

Command	*Read/ Write*	*Parameter #1*	*Parameter #2*	*Detailed description*	*Reply value*
INTIT	W R	Parameter field #	Time "?"	Sets/Gets Interlock Intervention Time	"AK" or "NAK"
PARADCI	W R	File name (ASCII) ?	/	Sets/Gets ADC I parameters	ASCII indicating all the configuration files available "AK" or "NAK"
PARADCV	W R	File name (ASCII) ?	/	Sets/Gets ADC V parameters	ASCII indicating all the configuration files available "AK" or "NAK"
PARDAC	W R	File name (ASCII) ?	/	Sets/Gets DAC parameters	ASCII indicating all the configuration files available "AK" or "NAK"
SSEQ	W R	Parameter field # "?"	Event #	Sets/Gets status event qualifier	ASCII indicating event qualifier
SSPM	W R	Parameter field # "?"	0/1	Sets/Gets status polarity mask	ASCII indicating polarity mask
MWG	W	Parameter field #	Cell content (ASCII)	Write to the given parameter field	"AK" or "NAK"
MSAVE	W	/	/	Save the used parameter in the non-volatile memory	"AK" or "NAK"
NETWORK	W R	DHCP IP NETMASK GATEWAY	1/0 IP NETMASK GATEWAY	Set the network configuration and/or enable/disable the DHCP	"AK" or "NAK" Network configuration

Table 4: Commands overview table

4.6 Basic Commands

In the following section are described the basic commands that allows to control the power unit and the control unit and to monitor their status.

4.6.1 MON Command

The ***MON*** (Module ON) command is intended to turn ON the module output driver, thus enabling the output current terminals and allowing the power supply to regulate and feed current or voltage to the connected load.

After the reception of a "MON" command, the control module automatically sets output current to 0A or 0V (depending if the module is set in constant current or constant voltage mode).

Replies from the control module to a ***MON*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "*xx*" indicating the error code. The complete list of the error codes is shown in the

Error Table. Sending a ***MON*** command when the module output is already enabled generates a non-acknowledgment response.

Examples:

MON command example:

MON\r →

← **#AK\r\n**

MON command example when the module is already enabled (09 code):

MON\r →

← **#NAK:09\r\n**

4.6.2 MOFF Command

The ***MOFF*** (Module OFF) command is intended to turn OFF the module output driver, thus disabling the output terminals.

The ***MOFF*** command automatically sets output current to 0A or 0V with a ramp before disabling the output drivers. This is done in order to ovoid output overshoots (especially in constant current regulation mode). The slew-rate of the ramp is factory defined.

Replies from the unit to a ***MON*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "*xx*" indicating the error code.

Examples:

MOFF command example:

MOFF\r →

← **#AK\r\n**

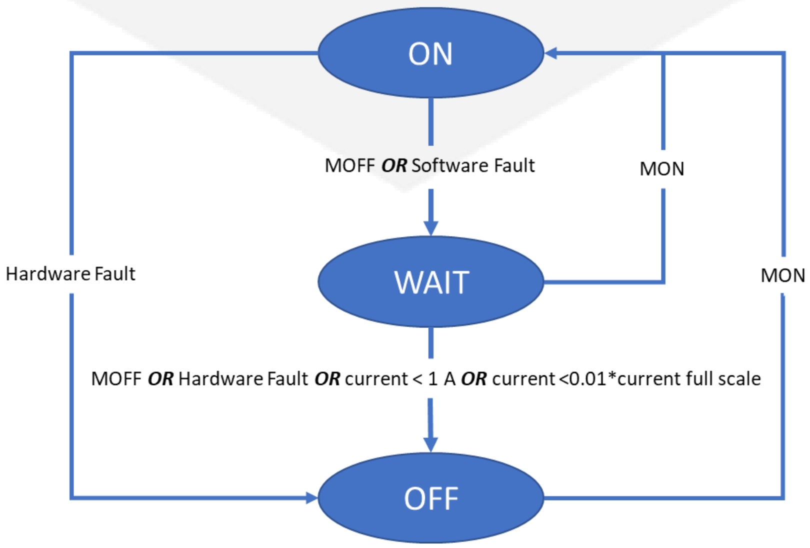
MOFF command example when the module is in local mode:

MOFF\r →

← **#NAK:15\r\n**

When the module receives an MOFF command, it will be driven to 0A setpoint carefully, in order to avoid any possible issue on highly-inductive loads.

Thus, the module operation is regulated as in the state machine shown below:



4.6.3 VER Command

The ***VER*** command returns the information regarding the model and the current installed firmware version.

The response to the ***VER*** command is in the following format:

#VER:model:fw_version\r\n

where "***#VER***" is the echo string, "***model***" is REGUL8OR ASCII and "***fw_version***" is the current firmware version. The echo, model and firmware information are separated by "**:**"character and the string is terminated with the standard "***\r\n***" character sequence.

Example:

VER command example:

VER\r →

← **#VER:FAST-PS 2020-400:0.9.01\r\n**

4.7 Faults Monitoring and Reset

This section describes how to behave in the case of a fault (as described in the "User Manual"), and so how to reset the power supply (section 4.7.5), and how to monitor the fault presence (section 4.7.1).

4.7.1 MSTR Command

The ***MSTR*** command returns the value of the power supply internal status. The response to the ***MSTR*** command is in the following format:

#MSTR:status_reg\r\n

where "***#MSTR***" is the echo string and "***status_reg***" is the hexadecimal representation of the internal status register. The internal status register has 32 bits and so its representation is composed by 8 hexadecimal values.

The ***MSTR*** command, being a reading command, returns a response in any module condition.

Example:

MSTR command example:

MSTR\r →

← **#MSTR: 08000002\r\n**

In this example the value 08000002 (hex) has the binary representation:

0000 1000 0000 0000 0000 0000 0000 0010

Which means that the 2^{th} bit and the 27^{th} bit are set. One can refer to the status register (next pages) to see to what these bits are referred to.

4.7.2 REGUL8OR Status Register

The following table shows the REGUL8OR internal status register structure:

Bit #	*Bit name*	*Description*
#31 - #24	/	*reserved*
23	Float Mode	*reserved*
#22	Local Mode	**This bit is set when the module is controlled via the LCD display**
#21	Polarity Inverter Enabled	**This bit is set when polarity inverter is controlled with the Regul8OR**
#20	FeedForward	*reserved*
#19 - #18	/	*reserved*
#17	Waveform	**Waveform is in execution**
#16	Ramping	**Module is ramping current or voltage**
#15 - #13	/	*reserved*
#12	Trigger Mode	***Set the module in trigger mode***
#11 - #10	/	*reserved*
#9 - #8	Update mode [2 bits]	**Normal [00], Analog input [01], Waveform [10], SFP [11]**
#7 - #5	/	*reserved*
#4	Loop Mode [2 bits]	**C.C. [0] or C.V. [1] output regulation mode**
#3	**Inverting Polarity**	**Indicates if there is an inversion of the polarity**
#2	**Wait For OFF status**	**The bit is set when the module is in wait for off status (i.e. in the transition between ON and OFF status)**
#1	Fault condition	**This bit is set if the module has experienced a fault condition**
#0	ON/OFF	**This bit is set when the module is enabled and correctly regulating the output**

Table 5: REGUL8OR Status Register structure

4.7.3 MFTR Command

The ***MFTR*** command returns the value of the device internal fault register. The response to the ***MFTR*** command is in the following format:

#MFTR:fault_reg\r\n

where "***#MFTR***" is the echo string and "***fault_reg***" is the hexadecimal representation of the internal fault register. The internal fault register has 32 bits and so its representation is composed by 8 hexadecimal values.

The ***MFTR*** command, being a reading command, returns a response in any module condition.

Example:

MFTR command example:

MFTR\r →

← **#MFTR: 00000102\r\n**

In this example the value 00000102 (hex) has the binary representation:

0000 0000 0000 0000 0000 0001 0000 0010

Which means that the 2th bit and the 9th bit are set. One can refer to the status register (next pages) to see to what these bits are referred to.

4.7.4 Regul8OR Fault Register

The following table shows the REGUL8OR internal fault register structure:

Bit #	Bit name	Description
#31 - #27	/	/
#26	/	/
#25	/	/
#24	/	/
#23 - #22	/	Reserved
#21	Interlock #6	Interlock 6 fault
#20	Interlock #5	Interlock 5 fault
#19	Interlock #4	Interlock 4 fault
#18	Interlock #3	Interlock 3 fault
#17	Interlock #2	Interlock 2 fault
#16	Interlock #1	Interlock 1 fault
#15	Pwm Fault	This bit means there is a problem with the PWM signals (internal or external)
#14	/	/
#13	/	/
#12	/	Reserved
#11 - #9	/	Reserved
#8	Dcct Error	This bit means there is a problem with the DCCT connected to the Regul8OR
#7	/	Reserved
#6	Regulation Fault	Control device can not regulate voltage or current correctly
#5	/	Reserved
#4	/	Reserved
#3	/	?
#2	/	?
#1	/	Reserved
#0	Temperature Fault	This bit is set when there is a problem with the ADCs temperatures

Table 6: Regul8OR fault register

4.7.5 MRESET Command

The ***MRESET*** command has to be used in order to perform a complete reset of the module status register. This is needed, for example, to enable the channel output again after a fault condition has been fixed.

Before sending the ***MRESET*** command to the power supply it is important to remove the fault presence, or the power supply will instantly get a new fault issue. After an ***MRESET*** command with reply in the form "***#AK\r\n***", the power supply can be turned on again.

Replies from the unit to a ***MRESET*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code). The complete list of the error codes is shown in the

Error Table**,** section 4.4**)**.

Examples:

MRESET command example:

MRESET\r →

← **#AK\r\n**

MRESET command example when the module is in local mode:

MRESET\r →

← **#NAK:15\r\n**

4.8 Current and Voltage Setting/Reading

In this section, specific commands to set or to read the value of the current and/or voltage are discussed.

4.8.1 UPMODE Command

First of all, in order to update the setpoint through remote control, the operation mode needs to be in ***NORMAL*** mode. ***UPMODE*** command can be used in order to select the update mode of the set-point. There are two possible mode of operation:

- **NORMAL** – in this mode of operation the control unit works in the standard update mode. The direct set-point or ramp commands are received using the standard Ethernet communication and they are applied immediately when the command is received. The analog input signal is ignored.
- **ANALOG** – in this mode of operation the control unit receives the set-point from its analog input. The analog input goes from -10 V up to +10 V. The setting command from the local or remote interface are not accepted.
  Note: this update mode is available only on units that have the analog input. This option has to be communicated at the order of the unit.

To set the update mode of operation the following commands has to be used:

UPMODE:mode\r\n

where "***mode***" is a string indicating the mode of operation:

- "**NORMAL**" for normal update mode,
- "**AIN**" for analog input update mode.

Replies from the unit to a ***UPMODE*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:****xx****\r\n***", when the command cannot be executed ("*xx*" is the error code). The analog update mode of operation can be set only when the module is turned OFF.

To read the current used loop mode of operation the query command: "***UPMODE:?***" has to be used. The response to the "***UPMODE:?***" query command is in the following format:

#UPMODE:mode \r\n

where "***#UPMODE***" is the echo string, "***mode***" is a single character indicating the loop mode ("***normal***" for normal update mode and "***ain***" for analog input update mode).

The update mode is also visible in the status register (bit #8 0 = Normal and 1 = AIN). For further information, please see the MSTR command.

Examples:

UPMODE example to set the update mode to analog:

UPMODE:AIN\r →

← **#AK\r\n**

UPMODE query example when the module is in normal mode:

UPMODE:?\r →

← **#UPDMODE: NORMAL\r\n**

4.8.2 MRI Command

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

The readback current value is represented with 6-digit precision. Replies from the control device to this command are in the following format:

#MRI:current_value\r\n

where "***#MRI***" is the echo string, "***current_value***" is the output current value readback in Ampere [A].

Example:

MRI command example:

MRI\r →

← **#MRI:22.123456\r\n**

4.8.3 MRV Command

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

The voltage readback value is represented with 6-digit precision. Replies from the control device to this command are in the following format:

#MRV:voltage_value\r\n

where "***#MRV***" is the echo string, "***voltage_value***" is the output voltage value readback in Volts [V].

Example:

MRV command example:

MRV\r →

← **#MRV:10.123456\r\n**

4.8.4 LOOP Command

The ***LOOP*** command can be used in order to select the mode of loop control of the REGUL8OR unit. There are two possible modes of operation:

- Constant Current (c.c.),
- Constant Voltage (c.v.).

To set the mode of operation the following commands has to be used:

LOOP:mode\r\n

where "***mode***" is a single char indicating the mode of operation:

- "**I**" for Constant Current (c.c.) mode and
- "**V**" for Constant Voltage (c.c.) mode.

Replies from the unit to a ***LOOP*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code). The two modes of operation can be changed only when the module is turned OFF.

To read the current used loop mode of operation the query command: "***LOOP:?***" has to be used. The response to the "***LOOP:?***" query command is in the following format:

#LOOP:mode \r\n

where "***#LOOP***" is the echo string, "***mode***" is a single character indicating the loop mode ("***I***" for constant current mode and "***V***" for constant voltage mode).

Examples:

LOOP set example to set the constant current mode:

LOOP:I\r →

← **#AK\r\n**

LOOP set example when the module is ON:

LOOP:V\r →

← **#NAK:09\r\n**

LOOP query example when the module is in constant voltage (c.v.) mode:

LOOP:?\r →

← **#LOOP:V\r\n**

4.8.5 MWI Command

The ***MWI*** command can be used to set the output current value when the module is in the constant current mode (see ***LOOP*** Command). This command is usually needed when running feedback-related applications and for small changes in the output current.

The use of this command is alternative to the MWIR Command (ramping current command), which is advised for regular use.

This command has the following format:

MWI:current_setpoint\r\n

where "***current_setpoint***" is the desired current set-point expressed in Ampere [A].

Replies from the module to a ***MWI*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code).

To read last applied current setpoint the query command: "***MWI:?***" has to be used. The response to this query command is in the following format:

#MWI:current_setpoint\r\n

where "***#MWI***" is the echo string, "***current_setpoint***" is the last applied current setpoint expressed in Ampere [A].

Examples:

MWI set example, with current setpoint +1.52 A:

MWI:1.52\r →

← **#AK\r\n**

MWI set example when the module is OFF:

MWI:1.52\r →

← **#NAK:13\r\n**

MWI query example:

MWI:?\r →

← **#MWI:1.52\r\n**

4.8.6 MWV Command

The ***MWV*** command can be used to set the output voltage value when the constant voltage mode is used (see ***LOOP*** Command). The use of this command is alternative to the ***MWVR*** Command (ramping voltage command).

This command has the following format:

MWV:voltage_setpoint\r\n

where "***voltage_setpoint***" is the desired voltage set-point expressed in Volts [V].

Replies from the unit to a ***MWV*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code).

To read last applied voltage setpoint the query command: "***MWV:?***" has to be used. The response to this query command is in the following format:

#MWV:voltage_setpoint \r\n

where "***#MWV***" is the echo string, "***voltage_setpoint***" is the last applied voltage setpoint expressed in Volts [V].

Examples:

MWV set example, with voltage setpoint +10.525 V:

MWV:10.525\r →

← **#AK\r\n**

MWV set example when the module is OFF:

MWV:10.525\r →

← **#NAK:13\r\n**

MWV query example:

MWV:?\r →

← **#MWV:10.525\r\n**

4.8.7 MWIR Command

The ***MWIR*** command can be used to perform a ramp to the given current setpoint. This command can be used when the constant current mode is selected (see ***LOOP*** Command).

The use of this command is alternative to the ***MWI*** Command**.** The difference between the ***MWI*** command and the ***MWIR*** command is that the first one generates a direct change in output current characterized by the PID regulator parameters (the command is ideally suited for small output current changes and feedback purposes) while the second one makes the power supply go from the previous to the actual current value performing a ramp, defined by a slew-rate in [A/s].

The default value of the slew-rate is stored in the parameter table and it can be read and modified using the ***MSRI*** command (section 4.8.8).

To dynamically change the current slew-rate value it is possible using the ***MSRI*** Command. This command has the following format:

MWIR:final_ramp_setpoint\r\n

where "***final_ramp_setpoint***" is the final current value expressed in Ampere [A] to which the power unit will ramp with the defined slew-rate.

Replies from the power unit to a ***MWIR*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code).

To read the selected final ramp setpoint, the query command: "***MWIR:?***" has to be used. The response to this query command is in the following format:

#MWIR:final_ramp_setpoint\r\n

where "***#MWIR***" is the echo string and "***final_ramp_setpoint***" is the final ramp setpoint expressed in Ampere [A].

Examples:

MWIR set example, with final ramp setpoint +10.5 A:

MWIR:10.5\r →

← **#AK\r\n**

MWIR set example when the module is OFF:

MWIR:10.5\r →

← **#NAK:13\r\n**

MWIR query example:

MWIR:?\r →

← **#MWIR:10.5\r\n**

4.8.8 MSRI Command

The ***MSRI*** command can be used to dynamically change the value of the current ramp slew-rate. The default slew-rate, used at start-up of the unit, is the value stored in the parameters table.

This command has the following format:

MSRI:slew_rate\r\n

where "***slew_rate***" is slew-rate for the current ramp expressed in Ampere per second [A/s].

Replies from the power converter to a ***MSRI*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code).

To read the current used slew-rate for the current ramp, the query command: "***MSRI:?***" has to be used. The response to this query command is in the following format:

#MSRI:slew_rate\r\n

where "***#MSRI***" is the echo string and "***slew_rate***" is the slew-rate value used for the current ramp expressed in Ampere per second [A/s].

Examples:

MSRI example, to set the current slew-rate to 10 A/s:

MSRI:10\r →

← **#AK\r\n**

MSRI set example when the unit is in local mode:

MSRI:10\r →

← **#NAK:15\r\n**

MSRI query example:

MSRI:?\r →

← **#MSRI:10\r\n**

4.8.9 MWVR Command

The ***MWVR*** command can be used to perform a ramp to the given voltage setpoint. This command can be used, when the constant voltage mode is selected (see ***LOOP*** Command).

The use of this command is alternative to the ***MWV*** Command**.** The difference between the ***MWV*** command and the ***MWVR*** command is that the first one generates a direct change in output voltage characterized by the PID regulator parameters while the second one makes the power supply go from the previous to the actual current value performing a ramp, defined by a slew-rate in [V/s].

To dynamically change the slew-rate value it is possible to use the ***MSRV*** Command (section 4.8.10).

This command has the following format:

MWVR:final_ramp_setpoint\r\n

where "***final_ramp_setpoint***" is the final voltage value expressed in Volts [V] to which the power unit will ramp with the defined slew-rate.

Replies from the FAST-PS to a ***MWVR*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code).

To read the selected final ramp setpoint, the query command: "***MWVR:?***" has to be used. The response to this query command is in the following format:

#MWVR:final_ramp_setpoint\r\n

where "***#MWVR***" is the echo string and "***final_ramp_setpoint***" is the final ramp setpoint expressed in Volts [V].

Examples:

MWVR set example, with final ramp setpoint +15.2 A:

MWVR:15.2\r →

← **#AK\r\n**

MWVR set example when the module is OFF:

MWVR:15.2\r →

← **#NAK:13\r\n**

MWVR query example:

MWVR:?\r →

← **#MWIR:15.2\r\n**

4.8.10 MSRV Command

The ***MSRV*** command can be used to dynamically change the value of the voltage ramp slew-rate. The default slew-rate, used at start-up of the unit, is the value stored in the parameters table.

This command has the following format:

MSRV:slew_rate\r\n

where "***slew_rate***" is slew-rate for the voltage ramp expressed in Volts per second [V/s].

Replies from the unit to a ***MSRV*** set are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code).

To read the current used slew-rate for the voltage ramp, the query command: "***MSRV:?***" has to be used. The response to this query command is in the following format:

#MSRV:slew_rate\r\n

where "***#MSRV***" is the echo string and "***slew_rate***" is the slew-rate value used for the voltage ramp expressed in Volts per second [V/s].

Examples:

MSRV example, to set the current slew-rate to 10 V/s:

MSRV:10\r →

← **#AK\r\n**

MSRV set example when the unit is in local mode:

MSRV:10\r →

← **#NAK:15\r\n**

MSRV query example:

MSRV:?\r →

← **#MSRV:10\r\n**

4.8.11 MLIMITS command

The ***MLIMITS*** command can be used to get informations about the external power supply output current and voltage values.

The response to the ***MLIMITS*** is in the following format:

#MLIMITS:min_voltage:max_voltage:min_current:max_current\r\n

where "#MLIMITS" is the echo string and the others are:

- ***min_voltage*: minimum output voltage of the external power supply**
- ***max_voltage*: maximum output voltage of the external power supply**
- ***min_current*: minimum output current of the external power supply**
- ***max_current*: maximum output current of the external power supply**

The ***MLIMITS*** command, being a reading command, returns a response in any module condition.

Example:

MLIMITS command example:

MLIMITS\r →

← **#MLIMITS:0:20:0:100\r\n**

4.8.12 MPLIMITS command

The ***MPLIMITS*** command can be used to get information about the external power supply output power.

The response to the ***MPLIMITS*** is in the following format:

#MPLIMITS:min_ output_power:max_output_power \r\n

where "***#MPLIMITS***" is the echo string and the others are:

- ***Min_output_power*: minimum output power of the external power supply**
- ***Max_output_power*: maximum output power of the external power supply**

The ***MPLIMITS*** command, being a reading command, returns a response in any module condition.

Example:

MPLIMITS command example:

MPLIMITS\r →

← **#MPLIMITS:0:2000\r\n**

4.9 Generic Monitoring Parameters

This section describes some generic parameters that can be monitored through dedicated instructions (ADC input plug-in board temperature, estimated active power applied to the connected load and so on).

4.9.1 MRT Command

The ***MRT*** command returns the value of the temperature directly measured on the ADC input plug-in boards.

The response to the ***MRT*** command is in the following format:

#MRT:temperature\r\n

where "***#MRT***" is the echo string and "***temperature***" is the temperature value expressed in Celsius [°C]. The ***MRT*** command, being a reading command, returns a response in any module condition.

Example:

MRT command example:

MRT\r →

← **#MRT:37.4\r\n**

4.9.2 MRW Command

The ***MRW*** command returns the actual value of the estimated active power applied to the connected load.

The response to the ***MRW*** command is in the following format:

#MRW:active_power\r\n

where "***#MRW***" is the echo string and "***active_power***" is the output active power readback expressed in Watts [W], estimated as the product of the output voltage and output current readbacks. The ***MRW*** command, being a reading command, returns a response in any module condition.

Example:

MRW command example:

MRW\r →

← **#MRW:100.4542\r\n**

4.9.3 MRID Command

The ***MRID*** command returns the module identification name string. This description is useful in case that there are numerous units installed and it is possible to give a description for each unit (for example the name of the load on which the unit is connected). This information is also displayed on the local display.

The response to the ***MRID*** command is in the following format:

#MRID:fast_ps_identification\r\n

where "***#MRID***" is the echo string and "***fast_ps_identification***" is the module identification string. The identification string is stored in the parameters table and so it is possible to change it using the MWG command.

Example:

MRID example with the module identification "SkewMag1.3":

MRID\r →

← **#MRID:SKEWMAG1.3\r\n**

4.9.4 POL Command

The ***POL*** command can be used to get information about the polarity inverter. The response to the ***POL*** is in the following format:

 # POL*:polarity\r\n*

where ***"#POL"*** is the echo string and the others are:

- ***min_ output_power*: minimum output power of the external power supply**
- ***min_ output_power*: maximum output power of the external power supply**

The ***POL*** command, being a reading command, returns a response in any module condition.

Example:

POL command example:

POL\r →

← **#POL:DIRECT\r\n**

4.10 Configuration Commands

In the following section are described the software commands that allow to read, set and store the working parameters of the power supply unit.

MRG Command and

MWG Command allow reading or modifying the working parameters.

The write-access to several parameters is password protected and certain parameters are read only and so it is not possible to modify them. To change the password privileges, use the ***PASSWORD*** Command.

In order to save the parameter on the on-board non-volatile memory, the ***MSAVE*** Command has to be used.

The complete list of the configuration parameters, their field index and the access privileges are hereafter shown:

Parameter #	*Access Privileges*	*Parameter Name*
#0	Read Only	Firmware ID
#1	Read Only	PS Model
#2	Read Only	Serial Number
#3	Read Only	MAC Ethernet
#4	Read Only	MAC SFP #1
#5	Read Only	MAC SFP #2
#6 - #8	/	Reserved
#9	Read Only	Current calibration date
#10	Read Only	Current Calibration Parameter a
#11	Read Only	Current Calibration Parameter b
#12	Read Only	Current Calibration Parameter c
#13	Read Only	Current Calibration Parameter d
#14	Read Only	Voltage Calibration Parameter a
#15	Read Only	Voltage Calibration Parameter b
#16	Read Only	Voltage Calibration Parameter c
#17	Read Only	Voltage Calibration Parameter d
#18	Read Only	Voltage calibration date
#19	Read Only	Current ADC gain
#20	Read Only	Voltage ADC gain
#21	/	Reserved
#22	/	Reserved
#23	/	Reserved
#24	Read Only	Analog Input Calibration Parameter a
#25	Read Only	Analog Input Calibration Parameter b
#26-#29	/	Reserved
#30	Admin	Module Identification
#31	Read Only	Default Current Slew Rate [A/s]

Parameter #	Access Privileges	Parameter Name
#32	Read Only	Default Voltage Slew Rate V [V/s]
#33-#4	/	Reserved
#35	Admin	Turning off time of Display [minutes] – 0 to disable
#36 - #39	/	Reserved
#40	Admin	PID V: Kp_v
#41	Admin	PID V: Ki_v
#42	Admin	PID V: Kd_v
#43	Admin	PID I: Kp_i
#44	Admin	PID I: Ki_i
#45	Admin	PID I: Kd_i
#46	Admin	PID V: Voltage Upper Limit
#47	Admin	PID V: Voltage Lower Limit
#48	Admin	PID I: Current Upper Limit
#49	Admin	PID I: Current Lower Limit
#50 - #54	/	Reserved
#55	Admin	Allow Remote OFF command
#56	Admin	Error Code Description (Disable (0)/Enable(1))
#57 - #79	/	Reserved
#80	Admin	Output Over-Current Limit [A]
#81	Admin	Output Over-Voltage Limit [A]
#82	Admin	Max Temperature ADCs
#83 - #85	/	Reserved
#86	Admin	Current Regulation Fault Limit [A]
#87	Admin	Voltage Regulation Fault Limit [A]
#88	Admin	Regulation Fault Intervention [s]
#89	/	Reserved
#90	Admin	Interlock Enable Mask
#91	Admin	Interlock Activation State
#92	Admin	Interlock #1 intervention time [ms]
#93	Admin	Interlock #1 name
#94	Admin	Interlock #2 intervention time [ms]
#95	Admin	Interlock #2 name
#96	Admin	Interlock #3 intervention time [ms]
#97	Admin	Interlock #3 name
#98	Admin	Interlock #4 intervention time [ms]
#99	Admin	Interlock #4 name
#100	Admin	Interlock #5 intervention time [ms]
#101	Admin	Interlock #5 name
#102	Admin	Interlock #6 intervention time [ms]
#103	Admin	Interlock #6 name

Parameter #	Access Privileges	Parameter Name
#104	/	Reserved
#105	Admin	Input Voltage Range
#106	/	Reserved
#105	Admin	Input Current Range
#109-#114	/	Reserved
#115	Admin	Signal Status Polarity Mask
#116	Admin	Signal Status Event Qualifier
#117-#175	/	Reserved
#176	Admin	DAC Output Range
#177	Admin	DAC Calibration offset
#178	Admin	DAC Calibration Gain
#179	Admin	DAC Output full-scale
#180	Admin	DAC Gain
#181-#199	/	Reserved
#200	Admin	Is the power supply monopolar?
#201	Admin	PS max output voltage [V]
#202	Admin	PS Max output current {A]
#203	Admin	PS Max output power [W]
#204	Admin	PS Max absorbed power [W]
#205	Admin	DAC Gain
#206	Admin	DAC Gain
#205-#299	/	Reserved
#300	Admin	Input ADC max current
#301	Admin	Input ADC max voltage

Table 7: Parameters table for REGUL8OR

4.10.1 MRG Command

The ***MRG*** command returns the value stored in the given parameter number. The correct form for the reading request is as follow:

MRG:parameter_index\r\n

where "***parameter_index***" is the index of the parameter to be read. The response to the ***MRG*** command is in the following format:

#MRG:parameter_index:parameter_value\r\n

where "***#MRG***" is the echo string, "***parameter_index***" is the parameter's index and "***parameter_value***" is the parameter caption. The unit replies with "***#NAK:****xx*\r***\n***", when the command cannot be executed ("*xx*" is the error code) – for example if the given parameter is out of the permitted range.

Examples:

MRG example of the A/s (parameter #31):

MRG:31\r →

← **#MRG:31:10\r\n**

MRG example of read a not valid parameter's index (parameter # -1):

MRG:-1\r →

← **#NAK:03\r\n**

4.10.2 MWG Command

The ***MWG*** command lets users write a desired value in the given parameters index.

MWG:parameter_index:parameter_value\r\n

where "***parameter_index***" is the parameter's index and "***parameter_value***" is the content to be written.

Replies from the power unit to a ***MWG*** write are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("*xx*" is the error code).

After a ***MWG*** command the values are immediately applied, but they are not stored in the internal memory. To store the modified parameters in the non-volatile internal memory it is necessary to use the **MSAVE Command**.

Examples:

MWG example of the Module ID (parameter #30)

MWG:30:MAGNET A\r →

← **#AK \r\n**

MWG write example to the read-only field #1 (PS Model):

MWG:1:MAGNET A\r →

← **#NAK:05\r\n**

4.10.3 Interlock Setting

The control unit external interlock can be enabled or disabled by writing to the corresponding Interlock Enable/Disable Mask field of the advanced configuration parameters (field #90), using the MWG command. The value to be written is in ASCII format, representing the corresponding bit mask, as shown in the following table:

Bit Mask	Interlock number	ASCII string
000001	Interlock #1	0x1
000010	Interlock #2	0x2
000100	Interlock #3	0x4
001000	Interlock #4	0x8
010000	Interlock #5	**0x10**
100000	Interlock #6	**0x20**

Table 8: Interlock Mask Parameter

Example 1: if only Interlock #2 needs to be enabled, the following command has to be se sent to the control unit (after having un-locked the password protection): “MWG:90:0x2\r”.

Example 2: if Interlock #2 and Interlock #4 needs to be enabled (0x2 + 0x8 = 0xA), the following command has to be se sent to the control unit (after having un-locked the password protection): “MWG:90:0xA\r”.

Interlock Activation Level Mask

Each external interlock can be chosen to trip at high or low logic level. For example, the high level for the dry contact interlocks means that the interlock trips when the interlock input signal is shorted, otherwise the low level that the interlock trips when the input is open.

To configure the interlock state mask, it is necessary to write on the advanced configuration parameters (field #91). The value to be written is an ASCII format representing the corresponding bit mask, as shown in the **Table 8**.

This setting has no effect if the interlock is not enabled.

***Example 1*:** if interlock #1 needs to have a high activation level (trip when the interlock input signal is shorted), the following command has to be se sent to the control unit (after having un-locked the password protection): “MWG:91:0x1\r”.

Example 2: if Interlock #1 and Interlock #4 (0x1 + 0x8 = 0x9) needs to have a high activation level (trip when the interlock input signal is shorted) the following command has to be se sent to the control unit (after having un-locked the password protection): “MWG:91:0x9\r”.

Interlock Intervention Time

The module allows to set also the interlock intervention time (how long an interlock signal needs to be at its activation level before tripping and thus generating a fault condition). The Intervention time parameters are stored in:

- field #92 for Interlock #1;
- field #94 for Interlock #2;
- field #96 for Interlock #3;
- field #98 for Interlock #4.
- field #100 for Interlock #5.
- field #102 for Interlock #6.

The value to be set is in ASCII format, representing the intervention time in milliseconds. The minimum settable value is 0 (immediate generating of fault condition) and the maximum value is 10.000 ms (corresponding to 10 seconds).

Example**:** if interlock #1 needs to have an interlock intervention time of 750 ms, the following command has to be se sent to the power unit: "MWG:92:750\r". This setting has no effect if the interlock is disabled.

Interlock Identification Name

Units also allows associating a name to the interlocks in order to read form the remote interface or to display on the local display the interlock condition name. The Intervention names are stored in:

- field #93 for Interlock #1;
- field #95 for Interlock #2;
- field #97 for Interlock #3;
- field #99 for Interlock #4;
- field #101 for Interlock #5;
- field #103 for Interlock #6;

The value to be set is in ASCII format, representing the interlock name.

Example**:** if the interlock #1 is associated to the cabinet door open, the following command can to be sent to the power unit: "MWG:93:Cabinet door\r". This setting has not effect if the interlock is disabled.

4.10.3.1 *INTEM Command*

The ***INTEM*** command can be used to read interlocks enable mask. The response to the ***INTEM*** is in the following format:

#INTEM*:*enable_mask\r\n

where ***"#INTEM"*** is the echo string and the ***enable_mask*** is the ASCII value of the interlocks enable mask.

The ***INTEM*** command, being a reading command, returns a response in any module condition.

Example:

***INTEM** command example:*

INTEM\r →

← **# *INTEM*:0x00000010*\r\n***

4.10.3.2 *INTEM:Interlock#:value command*

The ***INTEM:Interlock#:value*** command can be used to enable or disable each interlocks. For example, if interlock 2 has to be enable ***Interlock#*** has to be set to 1 and ***value*** equal to 1 (INTEM:2:1).

Interlock	Command	Value
1	INTEM:0:1/0	Enable/Disable
2	INTEM:1:1/0	Enable/Disable
3	INTEM:2:1/0	Enable/Disable
4	INTEM:3:1/0	Enable/Disable
5	INTEM:4:1/0	Enable/Disable
6	INTEM:5:1/0	Enable/Disable

Table 9: Regul8OR interlocks enable mask commands.

Replies from the control module to a ***INTEM:Interlock#:value*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx***" indicating the error code. The complete list of the error codes is shown in Error Table (section 4.4).

Example:

INTEM command example:

INTEM:1:1\r →

← #AK\r\n

4.10.3.3 *INTPM Command*

The ***INTPM*** command can be used to read or to set interlocks polarity mask. The response to the ***INTPM*** is in the following format:

 # *INTPM*:*polarity_mask*\r\n

where ***"#INTPM"*** is the echo string and the ***polarity_mask*** is the ASCII value of the interlocks polarity mask.

The ***INTPM*** command, being a reading command, returns a response in any module condition.

Example:

***INTPM** command example:*

INTPM:0x10\r →

← #AK\r\n

INTPM\r →

← # *INTPM*:0x00000010\r\n

4.10.3.4 *INTPM:Interlock#:value command*

The ***INTPM:Interlock#:value*** command can be used to set or to read the polarity of each interlocks or to read if a interlock has a direct or indirect polarity.

Interlock	Value	Command
1	1/0/? [Direct/Indirect/Read]	INTPM:0:value
2	1/0/? [Direct/Indirect/Read]	INTPM:1:value
3	1/0/? [Direct/Indirect/Read]	INTPM:2:value
4	1/0/? [Direct/Indirect/Read]	INTPM:3:value
5	1/0/? [Direct/Indirect/Read]	INTPM:4:value

Interlock	Value	Command
6	1/0/? [Direct/Indirect/Read]	INTPM:5:value

Table 11: Regul8OR interlocks polarity mask commands.

Replies from the control module to a ***INTPM:Interlock#:value*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx"*** indicating the error code. The complete list of the error codes is shown in Error Table (section 4.4).

Example:

INTPM command example:

INTPM:1:1\r →

← **#AK\r\n**

4.10.3.5 *INTNAME:Interlock#:name command*

The ***INTNAME:Interlock#:name*** command can be used to set or to read the name of each interlocks or to read the name given to a specific interlock.

For example if the interlock #1 is associated to the cabinet door open, the following command can be sent to the power unit: "***INTNAME***:0:Cabinet door\r" and to read the value "***INTNAME:0:?"*** the return value is an ASCII indicating the name given to the interlock.

Replies from the control module to a ***INTNAME:Interlock#:name*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx"*** indicating the error code. The complete list of the error codes is shown in Error Table (section 4.4).

Example:

INTNAME command example:

INTNAME:1:Cabinet door\r →

← **#AK\r\n**

INTNAME:1:?\r →

← **#INTNAME:1: Cabinet door \r\n**

4.10.3.6 *INTIT:Interlock#:value command*

The ***INTIT:Interlock#:name*** command can be used to set or to read the intervention time of each interlocks or to read the intervention time set to a defined interlock. This setting has not effect if the interlock is disabled.

Example:

INTIT command example:

INTIT:1:1000\r →

← #AK\r\n

INTIT:1:?\r →

← #INTIT:1:1000\r\n

4.10.4 Signal Status Setting

The control unit status signals can be associated to an event or disabled by writing to the corresponding Status signals Enable/Disable Mask field of the advanced configuration parameters (field #116), using the MWG command.

The value to be written is in ASCII format, and every status signal is defined by 4 bits starting from the least significant bit (that represent the first status signal).

The ASCII value written in the four bits represents the trigger event for the status signal. In the table below are shown the ASCII value of the trigger signals:

Trigger Signal	*ASCII string*
Disable	0x0
Interlock #1	0x1
Interlock #2	0x2
Interlock #3	0x3
Interlock #4	0x4
Interlock #5	0x5
Interlock #6	0x6
General Fault	0x7
Module ON	0x8

Table 10: Status Signal event trigger

Example 1: If Status Signal #1 needs to be triggered by Interlock #6, the following command has to be sent to the control unit (after having un-locked the password protection): "MWG:116:0x6\r".

Example 2: If Status Signal #1 and Status Signal #3 need to be triggered by *Interlock #6* and *Module ON*, the following command has to be sent to the control unit (after having un-locked the password protection): "MWG:116:0x806\r".

Status Signal Activation Level Mask

Each status signal can be chosen to trip at high or low logic level. For example, the high-level means that the status signal trips when the event trigger signal is up, otherwise the low level that the status signal trips when the event trigger is low. To configure the status signals polarity mask, it is necessary to write on the advanced configuration parameters (field #115). The value to be written is an ASCII format representing the corresponding bit mask, as shown in the table below.

Bit Mask	*Interlock number*	*ASCII string*
000001	Signal Status #1	0x1
000010	Signal Status #2	0x2
000100	Signal Status #3	0x4
001000	Signal Status #4	0x8
010000	Signal Status #5	**0x10**
100000	Signal Status #6	**0x20**

***Example 1*:** if Signal Status #1 needs to have a high activation level, the following command has to be se sent to the control unit (after having un-locked the password protection): “MWG:115:0x1\r”.

Example 2: if Signal Status #1 and Signal Status #4 (0x1 + 0x8 = 0x9) needs to have a high activation level the following command has to be se sent to the control unit (after having un-locked the password protection): “MWG:115:0x9\r”.

4.10.4.1 *SSEQ Command*

The ***SSEQ*** command can be used to read or to set status signals events qualifier mask. The response to the ***SSEQ*** is in the following format:

#SSEQ**:**events_qualifier_mask\r\n

where ***“#SSEQ”*** is the echo string and the ***events_qualifier_mask*** is the ASCII value of the status signals events qualifier mask.

Example:

***SSEQ** command example:*

SSEQ:0x000020\r →

← ***#AK\r\n***

SSEQ:?\r →

#SSEQ:0x000020\r\n ←

4.10.4.2 *SSEQ:Interlock#:value command*

The ***SSEQ:StatusSignal#:value*** command can be used to set or to read the trigger signal for each status signal. Table 10 is a list of all the trigger signal that can be linked to each status signal.

Replies from the control module to a ***SSEQ:StatusSignal#:value*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx***" indicating the error code. The complete list of the error codes is shown in Error Table (section 4.4).

Example:

SSEQ *command example:*

SSEQ:1:5\r →

← **#AK\r\n**

SSEQ:1:?\r →

← **#SSEQ:1:0x5\r\n**

4.10.4.3 *SSPM Command*

The ***SSPM*** command can be used to read status signals polarity mask. The response to the ***SSPM*** is in the following format:

#*SSPM*:*polarity_mask*\r\n

where ***"#SSPM"*** is the echo string and the ***polarity_mask*** is the ASCII value of the status signals polarity mask.

The ***SSPM*** command, being a reading command, returns a response in any module condition.

Example:

SSPM *command example:*

***SSPM*\r** →

← **# *SSPM*:0x00000010\r\n**

4.10.4.4 *SSPM:StatusSignal#:value command*

The ***SSPM:StatusSignal#:value*** command can be used to set or to read the polarity of each interlocks.

Interlock	Command	Value
1	1/0/? [Direct/Indirect/Read]	SSPM:0:value
2	1/0/? [Direct/Indirect/Read]	SSPM:0:value
3	1/0/? [Direct/Indirect/Read]	SSPM:0:value
4	1/0/? [Direct/Indirect/Read]	SSPM:0:value
5	1/0/? [Direct/Indirect/Read]	SSPM:0:value
6	1/0/? [Direct/Indirect/Read]	SSPM:0:value

Table 11: Regul8OR interlocks polarity mask commands.

Replies from the control module to a ***SSPM:StatusSignal#:value*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx***" indicating the error code. The complete list of the error codes is shown in Error Table (section 4.4).

Example:

SSPM command example:

SSPM:1:1\r →

← **#AK\r\n**

SSPM:1:?\r →

← **#SSPM:1:0x1 \r\n**

PASSWORD Command

The ***PASSWORD*** command can be used to unlock or lock the access to the protected parameter fields.

Several parameters are protected in order not to let inexperienced users to change some power supply parameters that might compromise the correct operation of the module. See the **Error! Reference source not found.** for further details regarding the password-protected cells (parameters with *User* access privileges are not password protected; parameters with *Admin* access privileges are password protected; parameters with *Read Only* access privileges cannot be modified).

The correct format for this command is as follows:

PASSWORD:password_word\r\n

where "***password_word***" is the password to lock or unlock the protected parameter fields, that can be:

- "***PS-ADMIN***" to receive the *Admin* access privileges and unlock the protected parameter fields;
- "***LOCK***" to return to *User* access privileges and lock the protected parameters fields.

Replies from the unit to a ***PASSWORD*** command are in the form "***#AK\r\n***" – when the command is accepted - or "***#NAK:xx\r\n***", when the command is not accepted ("*xx*" is the error code). When a wrong password word is received, the unit replies with a "***#NAK:07\r\n***" (error code 07 – invalid password) and locks the protected parameter fields.

To read the current privileges level the following query command can be used: "***PASSWORD:?***". The response to this query command is in the following format:

#PASSWORD:privileges_level\r\n

where "***#PASSWORD***" is the echo string and "***privileges_level***" is the string indicating the privileges level.

The privileges level "***ADMIN***" indicates that the user is able to modify the protected parameter fields, otherwise "***USER***" indicates that the uses is able to modify only the not protected parameter fields.

The password to unlock password-protected cells is:
PS-ADMIN

Examples:

PASSWORD example of correct password word (unlock the protected cells):

PASSWORD:PS-ADMIN\r →

← **#AK \r\n**

PASSWORD example of correct password word (lock the protected cells):

PASSWORD:LOCK\r →

← **#AK \r\n**

PASSWORD example of wrong password word:

PASSWORD:CAENELS\r →

← **#NAK:07 \r\n**

PASSWORD access level query:

PASSWORD:?\r →

← **#PASSWORD:ADMIN\r\n**

4.10.5 MSAVE Command

The ***MSAVE*** command can be used store the parameter fields in the non-volatile internal memory. If the parameter fields are not saved, they will be lost at power-off of the power supply.

Replies from the power modules to a ***MSAVE*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

MSAVE example:

MSAVE*\r* →

← **#AK *\r\n***

4.11 Advanced Features Commands

4.11.1 TRIG:? Command

The ***TRIG:?*** command is used to inquiry the control unit about the current setting of the trigger options.

Replies from the control unit to a ***TRIG:?*** are in the form "***OFF/BOTH/POS/NEG\r\n***" – depending on the current settings.

Example:

TRIG:? example:

TRIG:?\r →

← **TRIG:POS \r\n**

4.11.1 TRIG:BOTH Command

The ***TRIG:BOTH*** command is used to set the control unit in trigger mode (both NEG and POS).

Replies from the control module to a ***TRIG:BOTH*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

TRIG:BOTH example:

TRIG:BOTH\r →

← **#AK \r\n**

4.11.1 TRIG:LEVEL Command

The ***TRIG:LEVEL*** command is used to acquire the level read by the module (at the moment of setting the command).

Replies from the control module to a ***TRIG:LEVEL***are in the form "**HIGH\r\n**" – when the level is high - or "**LOW\r\n**", when the level is low.

Example:

TRIG:LEVEL example:

TRIG:LEVEL\r →

← **HIGH \r\n**

4.11.1 TRIG:NEG Command

The ***TRIG:NEG*** command is used to set the control module in trigger mode (negative edge).

Replies from the control unit to a ***TRIG:NEG*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

TRIG:NEG example:

TRIG:NEG\r →

← **#AK \r\n**

4.11.1 TRIG: OFF Command

The ***TRIG: OFF*** command is used to disable the trigger (TRIG:OFF).

Replies from the control unit to a ***TRIG: OFF*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

TRIG:OFF example:

TRIG:OFF\r →

← **#AK \r\n**

4.11.1 TRIG:POS Command

The ***TRIG:POS*** command is used to set the control module in trigger mode (positive edge).

Replies from the control unit to a ***TRIG:POS*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

TRIG:POS example:

TRIG:POS\r →

← **#AK \r\n**

4.11.2 UPMODE: WAVE Command

The ***UPMODE:WAVE*** command is used to set the control module in analog control.

Replies from the control unit to a ***UPMODE:WAVE*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

UPMODE:WAVE example:

UPMODE:WAVE\r →

← **#AK \r\n**

4.11.3 WAVE:KEEP_START Command

The ***WAVE:KEEP_START*** command is used to start the waveform generation when the module is in trigger mode (the module waits the trigger signal).

Replies from the control unit to a ***WAVE:KEEP_START*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

WAVE:KEEP_START example:

WAVE:KEEP_START\r →

← **#AK \r\n**

4.11.1 WAVE:N_PERIODS: Command

The ***WAVE:N_PERIODS:*** command is used to set the number of periods the waveform needs to be reproduced. The maximum number that can be set is 2^{32} -1 (32 bit integer). By setting "0", the waveform is reproduced with an infinite number of periods.

Replies from the control unit to a ***WAVE:N_PERIODS:*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

WAVE:N_PERIODS example:

WAVE:N_PERIODS:5\r →

← **#AK \r\n**

4.11.1 WAVE:POINTS: Command

The ***WAVE:POINTS:*** command is used to store the waveform points into the module. The minimum number of points is 100 and the maximum 500'000. The time resolution, point by point, is 10 us.

Replies from the control module to a ***WAVE:POINTS:*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

WAVE:POINTS example:

WAVE:POINTS: 1.1:0.543:…:3.42\r →

← **#AK \r\n**

4.11.1 WAVE:START Command

The ***WAVE:START*** command is used to start the waveform generation when the module is not in trigger mode.

Replies from the control module to a ***WAVE: START*** are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed ("***xx***" is the error code).

Example:

WAVE:START example:

WAVE:START\r →

← **#AK \r\n**

4.11.1 WAVE:STOP Command

The ***WAVE:STOP*** command is used to stop the waveform generation.

Replies from the control unit to a ***WAVE: STOP*** are in the form "***#AK*\r\n**" – when the command is correctly executed - or "***#NAK:xx*\r\n**", when the command cannot be executed ("***xx***" is the error code).

Example:

WAVE:STOP example:

WAVE:STOP\r →

← **#AK \r\n**

4.11.2 WAVE:TRIGGER: Command

The ***WAVE:TRIGGER:*** command is used to select the mode of operation of the waveform when in trigger mode.

The possible commands are ***WAVE:TRIGGER:START*** (when the waveform needs to be reproduced entirely) and ***WAVE:TRIGGER:POINTS*** (when the waveform needs to be reproduced point by point).

Replies from the control unit to a ***WAVE: TRIGGER*** are in the form "***#AK*\r\n**" – when the command is correctly executed - or "***#NAK:xx*\r\n**", when the command cannot be executed ("***xx***" is the error code).

Example:

WAVE:TRIGGER example:

WAVE:TRIGGER:START\r →

← **#AK \r\n**

4.12 Device Hardware Configurations

4.12.1 PARADCI Commands

The ***PARADCI*** command can be used to set a different current input range or get information about the current calibration input file stored in the device. The sub-commands to send are described in the next sub-chapters.

4.12.1.1 *PARADCI:?*

The ***PARADCI:?*** command can be used to know how many current configuration files are stored in the device and which ones. The response to the ***PARADCI:?*** is in the following format:

#PARADCI:first_conf_file:second_conf_file:…:last conf_file\r\n

where "***#PARADCI***" is the echo string and the others are:

- ***first_conf_file*: it is the first in alphabetic order current configuration file.**
- ***last conf_file*: it is the last in alphabetic order current configuration file.**

The #***PARADCI:?*** command can be used only when the module is in OFF state and there are no FAULTS.

Example:

PARADCI command example:

PARADCI:?\r →

← **#PARADCI:DCCT-100:DCCT-600 \r\n**

4.12.1.2 *PARADCI:name_file command*

The ***PARADCI:name_file*** command can be used to set a configuration file in order to change the input current range and the current input maximum and minimum limits.

Replies from the control module to a ***PARADCI:name_file*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx***" indicating the error code.

The complete list of the error codes is shown in Error Table (section 4.4).

The #***PARADCI:?*** command, being a read and write configuration command, and so can be used only when the module is in OFF state and there are no FAULTS.

Example:

PARADCI command example:

PARADCI:DCCT-100\r →

← #AK\r\n

PARADCI command example when the module is in ON state (09 code):

PARADCI:DCCT-100\r →

← #NAK:09\r\n

4.12.2 PARADCV Commands

The ***PARADCV*** command can be used to set a different voltage input range or get information about the voltage calibration input files stored in the device. The sub-commands to send are described in the next sub-chapters.

4.12.2.1 *PARADCV:?*

The ***PARADCV:?*** command can be used to know how many configuration voltage files are stored in the device and which ones. The response to the ***PARADCV:?*** is in the following format:

#PARADCV:first_conf_file:second_conf_file:…:last conf_file\r\n

where "***#PARADCV***" is the echo string and the others are:

- ***first_conf_file*: it is the first in alphabetic order voltage configuration file.**
- ***last conf_file*: it is the last in alphabetic order voltage configuration file.**

The #***PARADCV:?*** command can be used only when the module is in OFF state and there are no FAULTS.

Example:

PARADCI command example:

PARADCI:?\r →

← **#PARADCI:DCCT-100:DCCT-600 \r\n**

4.12.2.1 *PARADCV:name_file command*

The ***PARADCV:name_file*** command can be used to set a configuration file in order to change the input voltage range and the voltage input maximum and minimum limits. Replies from the control module to a ***PARADCI:name_file*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx***" indicating the error code. The complete list of the error codes is shown in Error Table (section 4.4).

The #***PARADCV:?*** command, being a read and write configuration command, and so can be used only when the module is in OFF state and there are no FAULTS.

Example:

PARADCV command example:

PARADCV:VRNG-50*\r*

#AK*\r\n*

PARADCI command example when the module is in ON state (09 code):

PARADCI:* VRNG-50*\r

#NAK:09*\r\n*

4.12.3 PARDAC Commands

The ***PARDAC*** command can be used to set a different voltage output range or get information about the voltage output DAC calibration files stored in the device. The sub-commands to send are described in the next sub-chapters.

4.12.3.1 *PARDAC:?*

The ***PARDAC:?*** command can be used to know how many output voltage DAC configuration files are stored in the device and which ones. The response to the ***PARDAC:?*** is in the following format:

#PARADCI:first_conf_file:second_conf_file \r\n

where "***#PARDAC***" is the echo string and the others are:

- ***first_conf_file*: it is the first in alphabetic order output voltage configuration file.**
- ***last conf_file*: it is the last in alphabetic order output voltage configuration file.**

The #***PARDAC:?*** command can be used only when the module is in OFF state and there are no FAULTS.

Example:

PARDAC command example:

PARDAC:?*\r*

#PARDAC:DAC-05:DAC-10 *\r\n*

4.12.3.1 *PARDAC:name_file command*

The ***PARDAC:name_file*** command can be used to set a configuration file in order to change the output voltage range. Replies from the control module to a ***PARDAC:name_file*** command are in the form "***#AK\r\n***" – when the command is correctly executed - or "***#NAK:xx\r\n***", when the command cannot be executed, with "***xx"*** indicating the error code.

The complete list of the error codes is shown in Error Table (section 4.4).

The #***PARDAC:?*** command, being a read and write configuration command, and so can be used only when the module is in OFF state and there are no FAULTS.

Example:

PARDACcommand example:

PARDAC:DAC-05\r →

← **#AK\r\n**

PARDAC command example when the module is in ON state (09 code):

***PARDAC:* DAC-05\r** →

← **#NAK:09\r\n**

4.13 Additional Features

This section is intended to collect different and peculiar features of CAEN ELS power supplies.

4.13.1 Auto Shut Down of Display

CAEN ELS power supplies, by default, have an automatic turning off time of 30 minutes, contained in Cell 35 of the power supply memory.

This time can be changed by changing the content of Cell 35 (User privilege needed).

The user is free to change the content of this cell or to disable the turning off time completely. In order to do so, cell 35 must be set to "0".

4.13.2 DHCP (Dynamic Host Configuration Protocol) Configuration

The ***NETWORK:DHCP*** command is used to enable or disable the DHCP:

NETWORK:DHCP:0 (disabled)
NETWORK:DHCP:1 (enabled)

When the DHCP is enabled the network configuration is automatically set on the module, and any change performed by the user will have no effect (e.g. sending the “NETWORK:IP” command).

When the DHCP is disabled, the user may configure the network parameters as below:

NETWORK:IP:IP
NETWORK:NETMASK:NETMASK
NETWORK:GATEWAY:GATEWAY

The DHCP is disabled by default.

The user may require to the module the current Network configuration sending the command:

NETWORK:?

4.13.3 HELP Command

The ***HELP*** command returns all the commands with the description. The response to the ***HELP*** is in the following format:

****
CAENels REGUL8OR
****
Commands:
MRI:Gets output current

The ***HELP*** command, being a reading command, returns a response in any module condition.