



# EEZ BB3 User manual

*Modular T&M solution*

Ver. 1.05 – 04/2021  
[www.envox.hr](http://www.envox.hr) – [github.com/eez-open](https://github.com/eez-open)



## Table of Contents

1.1.Legal notices.....	7
1.2.Contact Us.....	7
1.3.Acknowledgments.....	7
1.4.Revision history.....	7
1.5.Who Should Use This Manual.....	8
1.6.Safety Requirements.....	8
1.6.1.Risk of electric shock.....	8
1.6.2.Risk of electric shock due to exceeding low voltage protection.....	8
1.7.Care and cleaning.....	8
1.8.Environmental Considerations.....	9
1.9.Product End-of-Life Handling.....	9
2.Key features .....	10
2.1.General.....	10
2.2.User interface and programming.....	10
2.3.Connectivity.....	11
2.4.Protections and controls.....	11
3.Operation Environment .....	11
3.1.Storage environment .....	11
4.Unpacking and Checking the EEZ BB3 .....	11
4.1.Delivery List .....	12
4.1.1.Assembled unit.....	12
4.1.2.Kit version.....	12
4.2.Placing the EEZ BB3.....	12
4.3.Starting the EEZ BB3 .....	13
4.4Replacing the fuses.....	13
4.5.Battery replacement.....	13
4.6.Micro SD card (non-volatile memory).....	14
5.The EEZ BB3 overview.....	15
5.1.Front panel overview.....	15
5.2.Module front panels overview.....	16
5.3.Rear panel overview.....	17
5.4.Display pages overview.....	17
5.5.Module display views.....	19
5.5.1.Two modules display view.....	20
6.Getting started.....	23
6.1.Powering on.....	23
6.2.Standby mode and switching off .....	23
7.Basic operations.....	25
7.1.Turn channel output on and off.....	25
7.2.Set channel output values.....	25
7.3.Data input methods.....	26
7.4.Accessing channel protections and settings.....	27
7.5.Accessing system settings.....	28
8.System settings.....	29
8.1.System protection settings.....	29
8.2.Digital I/O settings.....	30
8.3.General trigger settings.....	31
8.4.Ramp & delay.....	32
8.5.System temp and fan.....	33
8.6.USB settings.....	34
8.7.Ethernet.....	36

8.7.1.MQTT.....	37
8.7.2.Static LAN settings (DHCP disabled).....	38
8.8.Date & time.....	38
8.9.Encoder & User sw.....	39
8.10.Display settings.....	40
8.11.Sound settings.....	42
8.12.System information.....	43
9.System functions.....	45
9.1.Event viewer.....	45
9.1.1.Events filtering.....	45
9.2.Tracking.....	46
9.3.File manager.....	47
9.4.Power outputs coupling.....	49
9.5.User profiles.....	51
9.6.Display lock / unlock.....	53
9.7.Power / reset control.....	53
9.8.User SW.....	54
10.Channel protections.....	55
10.1.Current limit and OCP.....	56
10.2.Voltage limit and OVP.....	56
10.3.Hardware OVP.....	57
10.4.Power limit and OPP.....	57
10.5.Temperature (OTP).....	58
11.Special channel functions.....	59
11.1.Channel trigger settings.....	59
11.2.List programming.....	61
11.2.1.Inserting a new step.....	63
11.2.2.Deleting of list items.....	63
11.3.Display view.....	64
11.4.Info.....	65
11.5.Advanced options.....	65
11.5.1.Current range.....	66
11.5.2.Remote voltage sensing.....	66
11.5.3.Remote voltage programming.....	67
11.5.4.Down-programmer (DP).....	67
12.Power modules calibration.....	69
12.1.Start calibration wizard.....	69
12.2.Voltage calibration setup.....	70
12.2.1.Voltage calibration steps.....	70
12.3.Current calibration setup.....	71
12.3.1.Current calibration steps.....	72
12.3.2.Deleting existing and adding new calibration points.....	73
12.3.3.Save calibration parameters.....	74
13.Firmware upgrade.....	75
13.1.MCU module firmware upgrade.....	75
13.1.1.Linux.....	75
13.1.2.Windows.....	77
13.1.3.Mac OSX.....	80
13.2.Peripheral module firmware upgrade.....	81
13.2.1.Using EEZ Studio to upgrade peripheral module firmware.....	81
13.2.2.Upgrade peripheral module firmware using GUI on EEZ BB3.....	83
14.Data logging.....	87
14.1.Data logging.....	87

14.2.Displaying recorded data.....	88
14.3.Logged data transfer to a computer.....	90
14.4.Initiate data log transfer on the EEZ BB3.....	92
14.5.Establishing serial (via USB) communication with a Linux computer.....	92
<b>15.MQTT.....</b>	<b>95</b>
15.1.Set channel parameter using subscribe topic.....	95
15.2.Drawing a topic graph.....	97
15.3.Publish topics.....	98
15.3.1.System.....	98
15.3.2.Dcpsupply.....	98
15.4.Subscribe topics.....	98
15.4.1.System.....	98
15.4.2.Dcpsupply.....	98
<b>16.Scripting with MicroPython.....</b>	<b>99</b>
16.1.Distinctive features of EEZ BB3 MicroPython implementations.....	99
16.2.Sample MicroPython script .....	99
16.2.1.Creating an EEZ Studio project.....	99
16.2.2.MicroPython script.....	108
16.2.3.MicroPython script deployment and execution.....	109
16.3.EEZ Python module.....	110
16.4.MicroPython script examples.....	111
<b>DECLARATION OF CONFORMITY.....</b>	<b>113</b>
<b>RoHS Compliance Statement.....</b>	<b>115</b>

## **1.1. Legal notices**

The information in this document is subject to change without notice.

Envox d.o.o. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Envox d.o.o. shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document is released under *open license FDL v1.3* from [GNU.org](#). Therefore you are entitled to freely copy and redistribute it, with or without modifying it, either commercially or non-commercially. For additional details please consult the content of the [license](#).

## **1.2. Contact Us**

If you have any problem or requirement when using EEZ products or this manual, please contact Envox:

Discord server: <https://discord.gg/dhYMnCB>

E-mail: [support@envox.hr](mailto:support@envox.hr)

Website: [www.envox.hr](http://www.envox.hr)

## **1.3. Acknowledgments**

Icons made by *pixel-perfect* and *freepik* from [flaticon.com](#)

Document created in OpenOffice / LibreOffice

Proofreading by [@xife](#)

## **1.4. Revision history**

Date	Version	Changes
2020-05-10	1.01	New calibration procedure Firmware upgrade corrections
2020-08-24	1.02	Two modules display view Display test USB OTG, mouse and mass storage Peripheral module firmware upgrade Logged data transfer to a computer Additional notice in calibration section
2020-11-11	1.03	Custom logo Encoder & User SW settings page update System information page update Output INHibit page added MCU module firmware upgrade instructions for Mac OSX Scripting with MicroPython
2021-02-11	1.04	Firmware v1.6 new features Fixed YT scroll and scan view pages Redesigned Power / reset control page MCU module version selection (System information) Labels & Colors (Display settings) Bargraphs scaling (Display view) Redesigned Data logging settings page
2021-03-23	1.05	CE Declaration of conformity, RoHS Compliance Statement

## 1.5. Who Should Use This Manual

This manual is designed for users who are familiar with basic electrical theory, especially as it applies to the operation of power supplies. This includes understanding of Constant Voltage and Constant Current operating modes and the control of input and output power, as well as the observance of safe techniques while making supply or pin connections and any changes in settings.

## 1.6. Safety Requirements

The EEZ BB3 is only designated for use in laboratory, residential, business, commercial or small business settings.

The EEZ BB3 should only be used for its designated purpose. You must observe all safety and usage instructions documented in this manual and the operating conditions and performance limitations as detailed in the data sheet.

This product documentation helps you to use a EEZ BB3 safely and efficiently.  
Keep the product documentation in a safe place and pass it on to the subsequent users.

To avoid any injuries or damage to the EEZ BB3 or any device connected to it you must review the following safety advice carefully before operating the device. Only use the EEZ BB3 as outlined in this manual.

- Use an AC line power cord designed for the EEZ BB3 and authorized by your local regulations. Ensure it is not damaged in any way before connecting it.
- **The EEZ BB3 is grounded through the Protective Earth (PE) lead of the AC line power cord. To avoid electric shock, it is essential to connect the earth terminal of the AC line power cord to the Protective Earth terminal before any inputs or outputs.**
- To avoid fire or shock hazard, observe all ratings and markers on the EEZ BB3 and check your manual for more information about ratings before connecting.
- Do not connect the AC line power cord or operate the EEZ BB3 with its covers or panels removed.
- Make sure that fuses specified for installed modules and AC line voltage are installed.
- If you suspect that the EEZ BB3 is damaged or not operating correctly (e.g. the power up self-test failed or any function does not work as expected) disconnect the device and contact Envox support.
- Ensure adequate clearance to the rear panel cooling fan air intake. Inadequate ventilation may result in high temperature and premature or multiple triggers of the OTP (over-temperature protection).
- In order to avoid short circuit or electric shock, please do not operate the EEZ BB3 in a humid environment.
- In order to avoid damaging the device or personal injury, do not expose the device to flammable gasses.
- Keep EEZ BB3 surfaces clean and dry to avoid the influence of dust and/or moisture in the operating environment.
- Operate in an electrostatic discharge protective area to avoid damage induced by static discharge.

### 1.6.1. Risk of electric shock

The casing and all chassis parts are connected to a Protective Earth (PE) conductor. Disconnection of the earthed protective connection inside or outside the EEZ BB3 is prohibited.

### 1.6.2. Risk of electric shock due to exceeding low voltage protection

Output coupling of two power module in series provides voltage as high as 80 V. In this case, any contact with live components is life threatening. Only qualified and trained personnel should operate the EEZ BB3 and any connected loads.

## 1.7. Care and cleaning

Do not store or leave the EEZ BB3 where it will be exposed to direct sunlight for extended periods as this may result in damage to the TFT display.

It is recommended to clean the EEZ BB3 regularly according to its operating conditions. To clean the exterior surface, perform the following steps:

- Disconnect the EEZ BB3 from AC line power
- Use a lint-free cloth (with a mild detergent or water) to clean the loose dust on the outside of the EEZ BB3. Take extra care when cleaning the TFT display to avoid scratching it.

*WARNING: To avoid injury resulting from short circuit, make sure the EEZ BB3 is completely dry before connecting it to AC line power.*

## 1.8. Environmental Considerations

The following symbol indicates that this product complies with the applicable European Union requirements according to Directives 2012/19/EU on waste electrical and electronic equipment (WEEE) and batteries. This affixed product label indicates that you must not discard this electrical or electronic product in domestic household waste.

With reference to the equipment types in the WEEE directive Annex 1, this instrument is classified as a “Monitoring and Control Instrument” product.



## 1.9. Product End-of-Life Handling

The EEZ BB3 may contain substances that could be harmful to the environment or human health. In order to avoid release of such substances into the environment and harm to human health, we encourage you to recycle this product in an appropriate system that will ensure that most of the materials are reused or recycled appropriately. Please contact your local authorities for disposal or recycling information.

## 2. Key features

The EEZ BB3 is a modular DC power supply with multiple isolated channels, combined power output up to 465 W, maximum voltage output ranging from 20 to 80 V and maximum current output of up to 10 A. The graphical user interface features a large color TFT touch-screen display combined with an incremental encoder providing a clear, responsive and easy to operate interface. Outstanding key features of the EEZ BB3 are:

### 2.1. General

- Modular and open-source design based on [EEZ DIB v1.0](#) specification enabling multiple configurations and future-proof upgrading
- Three module slots for installation of different power modules in any combination
- Minimized wire harness and easily accessible modules simplify assembly, upgrade and maintenance
- High performance 32-bit ARM Cortex M7 master MCU
- Power output coupling *without* external wiring thanks to built in power relays that provide coupling in series, parallel or split rail for two modules and common ground for all three modules
- Down-programmer with high current capability (DCP405 module only)
- Full range auto-switch AC input (115 / 230 Vac)
- Low noise Ø80 mm cooling fan with speed algorithm based on multiple temperature sensors and output current readings
- Compact size: 290 (W) x 123 (H) x 240 (D) mm
- Free and open source [software](#) and open source [hardware](#) design

### 2.2. User interface and programming

- Large color 4.3" (470 x 272) TFT touch-screen display as human machine interface (HMI) for fast and simple navigation through all available options and each channel's detailed information
- Multiple color themes
- Encoder with switch plus user-defined switch complement the TFT touch-screen for even faster and easier navigation
- Ten user-defined profiles with power-up recall control
- Calibration wizard
- High programming and readback resolution for applications with the highest demands
- Independent real-time voltage, current and power measurement of all installed channels
- Four display modes: numerical, horizontal bar, vertical bar and YT-view
- Two YT-view modes (cursor or data scroll) and with high refresh rate (down to 10 mS)
- Output Enable (OE) synchronization and sequencing with programmable delays
- Arbitrary waveform generation function with user friendly GUI for both voltage and current
- Multi-channel tracking function
- Simple to use data logger and viewer
- File manager
- USB DFU firmware upgrade for master MCU and slave MCUs on peripheral boards
- Trigger system with various inputs (manual, software, digital input, etc.)
- Comprehensive SCPI support (300+ commands)
- [MicroPython](#) scripting
- [MQTT](#) support for IoT networking
- Cross-platform (Linux, OS X, Windows) [EEZ Studio](#) comes with intuitive HMI editor and provides centralized control, programming, data acquisition and analysis of multiple EEZ BB3 and 3<sup>rd</sup> party SCPI-enabled instruments. Free to [download](#)

### 2.3. Connectivity

- Remote control via Ethernet or USB interface
- Removable Micro SD card
- 10/100 Mbit/s Ethernet with Auto-negotiation and [Auto-MDIX](#)
- USB FS with [OTG](#) (virtual COM port, mass storage, mouse support)
- RTC with battery backup and NTP synchronization (via Ethernet)
- Digital control lines (2 x input + 2 x output, buffered and protected)

### 2.4. Protections and controls

- AC soft-start (inrush current limiter) and standby control
- AC input protection (MOV, VAR, SAR)
- Multiple channel- and system-wide protections against over-voltage, over-current (with electronic fuse function), over-power and over-temperature with adjustable trip delay
- Remote sense reverse polarity protection (DCP405 module only)
- Fast over-voltage protection with triac crowbar (DCP405 module only)
- User interface lock function with password protections

For a detailed specification, refer to the data sheet.

## 3. Operation Environment

- Location: Indoor, no direct sunlight, dust free, almost non-conductive pollution (note below)
- Relative Humidity: < 80 %
- Altitude: < 2000 m
- Temperature: 0°C to 40 °C

(Pollution Degree) EN 61010-1:2001 specifies the pollution degrees and their requirements as follows. The EEZ BB3 falls under degree 2. Pollution refers to “addition of foreign matter, solid, liquid, or gaseous (ionized gases), that may produce a reduction of dielectric strength or surface resistivity”.

- Pollution degree 1: No pollution or only dry, non-conductive pollution occurs. The pollution has no influence.
- Pollution degree 2: Normally only non-conductive pollution occurs. Occasionally, however, a temporary conductivity caused by condensation must be expected.
- Pollution degree 3: Conductive pollution occurs, or dry, non-conductive pollution occurs which becomes conductive due to condensation which is expected. In such conditions, equipment is normally protected against exposure to direct sunlight, precipitation, and full wind pressure, but neither temperature nor humidity is controlled.

### 3.1. Storage environment

- Location: Indoor
- Relative Humidity: < 70 %
- Temperature: -10 °C to 70 °C

## 4. Unpacking and Checking the EEZ BB3

To remove the EEZ BB3 from its packaging and check the equipment for completeness, proceed as follows:

- Check the package for damage
- Carefully unpack the EEZ BB3 and the accessories
- Check the package contents for completeness using the *Delivery list* and package contents
- Check the equipment for any visible shipping-related damage or other mechanical problems, e.g. loose parts inside

If there is damage or anything missing, contact the supplier and the carrier who delivered the EEZ BB3. Do not operate the EEZ BB3 in this case.

*Retain the original packing material. If the EEZ BB3 needs to be transported or shipped later, you can use the material to protect the exposed and fragile elements on the front panel.*

The EEZ BB3 must be stored in dry, closed, indoor premises. If the EEZ BB3 was transported under extreme temperatures, it is recommended that you allow a minimum of two hours to reach the appropriate temperature before operating the EEZ BB3.

## **4.1. Delivery List**

### **4.1.1. Assembled unit**

The EEZ BB3 comes with the following components:

- EEZ BB3 preloaded with selected peripheral modules
- AC mains cable for the selected region (EU, US or AUS)
- *Read me first* leaflet with links to online documentation
- Set of DC power cables, clip-on probes and crocodile clips (optional)

### **4.1.2. Kit version**

*The EEZ BB3 kit is not an end user product. As such it was not put into any conformance testing and it may not comply with some or any technical or legal requirement that are applicable to finished products including, without limitation, directives regarding electromagnetic compatibility, FCC, CE, or UL.*

Assembling and using the EEZ BB3 requires an understanding of electronic circuits. Additionally, basic computer knowledge is recommended for performing firmware upgrades.

The EEZ BB3 comes in different kit versions. Typical content of the kit version:

- Enclosure kit
- Wire harness, nuts&bolts package
- Set of basic modules (i.e. AUX-PS, MCU and BP3C)
- Set of selected peripheral modules
- *Read me first* leaflet with links to online documentation
- Set of DC power cables, clip-on probes and crocodile clips (optional)

Kit assembly instructions are available on the following link: <https://bit.ly/2VtWsZu>

## **4.2. Placing the EEZ BB3**

If the EEZ BB3 is operated on a bench top, the surface must be flat. You can place the EEZ BB3 horizontally, or in a slope position by unfolding the front feet.



*Fig. 1: Adjustable front feet*

The feet can fold in if they are not folded out completely or if the EEZ BB3 is shifted. Collapsing feet can cause injury or damage the EEZ BB3.

- Fold the feet completely in or out to ensure stability of the EEZ BB3.
- Never shift the EEZ BB3 when the feet are folded out.
- When the feet are folded out, do not work under the EEZ BB3 and avoid placing anything

underneath.

The front feet in folded out position can break if they are overloaded. The overall load on the folded out feet must not exceed 50 N. If the front feet are folded in, it is possible to stack the EEZ BB3 with more instruments securely.

### 4.3. Starting the EEZ BB3

The EEZ BB3 is equipped with an AC power supply connector, (IEC C14 type) which can be used with different AC power voltages in wide range without need to adjust manually input voltage or frequency.

*IMPORTANT: The EEZ BB3 must only be connected to an outlet that has a functional ground contact*

- Check the AC line voltage.
- Check the fuse type and if necessary replace the fuse type suitable for the line voltage.
- Connect the EEZ BB3 to the AC mains using the supplied power cable.

### 4.4. Replacing the fuses

The EEZ BB3 has main power fuses, which are located on the rear panel of the unit. Handling the fuses while power is on can lead to electric shock. Therefore, before opening the fuse holder (see Fig. 2), make sure that the EEZ BB3 is disconnected from AC power.



Fig. 2: Opening the fuse holder

To replace the fuses the EEZ BB3 has an externally accessible fuse holder combined with the IEC socket.

The nominal current of the fuse depends on the line voltage and number of installed power modules. Select the fuses according to the voltage of the AC line, as specified in Table 1. Fuse size is 20 x 5 mm of *time-lag* type and has to be rated for the selected AC line voltage.

No. of power modules	1	2	3
115 V	2 x 3.15 A	2 x 6 A	2 x 8 A
230 V	2 x 2 A	2 x 4 A	2 x 6 A

Table 1: Fuses selection

### 4.5. Battery replacement

The CR2032 coin cell battery is used as a power backup for the RTC. It uses a non-rechargeable lithium battery that will need to be replaced when its voltage drops to approximately 90 % of its rated value. The current battery value can be viewed on the [System information](#) page.

To access the EEZ BB3 interior, First, turn off the EEZ BB3 and unplug the power cord from the back, then remove the top cover, which is secured by four screws (two on each side).

Fig. 3 shows where the battery is located. You can remove the existing battery by pinching it. When inserting a new one, keep in mind the polarity (+ side must face inwards).

Do not use metal pliers instead of your fingers to insert or remove the battery as it may cause a short

circuit. If you cannot reach the battery with your fingers, use a plastic tweezers instead.

*Do not expose the battery to high temperature or fire. Keep it out of the reach of children. Improper change of a battery may cause damage, fire or explosion.*

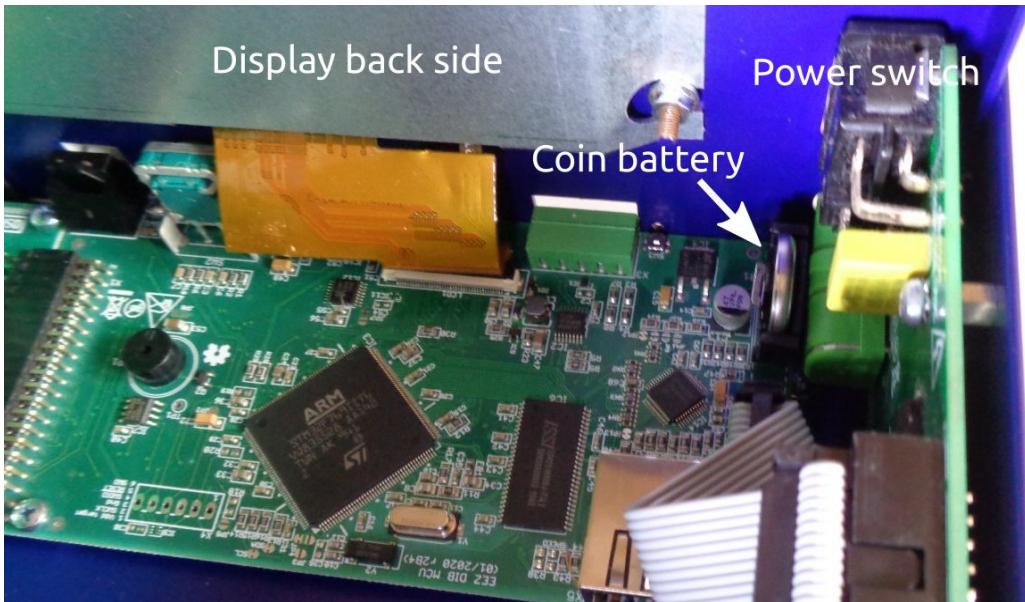


Fig. 3: Coin battery location (MCU r2B4)

#### 4.6. Micro SD card (non-volatile memory)

The EEZ BB3 uses a micro SD card as a non-volatile memory. Due to its small size, extra care will be required when inserting and pulling out to prevent damage to the card or socket.

Attention should also be paid that the card is inserted correctly into the socket otherwise it could fall inside the enclosure. Therefore, it is recommended that the EEZ BB3 be completely switched off (not in standby mode!) when the card is inserted.

Fig. 4 shows how the card should be positioned when inserted: its connectors should be facing up (i.e. must be visible).



Fig. 4: Micro SD card insertion

## 5. The EEZ BB3 overview

### 5.1. Front panel overview



Fig. 5: Left side of the EEZ BB3 chassis front panel

- ① 4.3" (480 x 272) color TFT touch-screen display, 16-bit (65536) colors with brightness and luminosity control: used for local system and module specific parameter display and settings.
- ② AC mains power switch: turns the EEZ BB3 on and off.
- ③ Standby indicator: on when EEZ BB3 is in standby mode (power modules AC mains power is switched off).
- ④ Protective Earth (PE) 4 mm socket.
- ⑤ Master MCU *BOOT0* miniature tactile switch access hole (*MCU r2B4 only*): when pressed on start up the EEZ BB3 enters a special bootloader mode for system firmware uploading.
- ⑥ Digital I/O terminals (two input and two output, protected)
- ⑦ Micro SD card with card detection switch.
- ⑧ 5-pin USB Mini AB socket (with OTG support)
- ⑨ Incremental encoder with switch
- ⑩ User-defined tactile switch can be assigned to perform selected action e.g. entering inhibit mode, taking screenshot, manual triggering, etc. (see [User SW](#)). In the version with MCU module r3B3 button is also used to enter a special bootloader mode.

## 5.2. Module front panels overview

The EEZ BB3 can accommodate up to three modules. An example with three power modules are shown on Fig. 9.



Fig. 6: Right side of the EEZ BB3 chassis front panel

- ① Operation mode indicators: CC – Constant Current mode, CV – Constant Voltage mode, Rprog – Remote programming is active (DCP405 only), Sense – Remote sensing is selected (DCP405 only)
- ② Advanced options input terminals (DCP405 only): Remote programming, Remote sensing and PE. If PE is used for shielding remote sensing cable, connect it on one side only to avoid creation of [ground loop](#).
- ③ Positive output (Vout+) power connector, 4 mm, “[floating](#)” (i.e. galvanically isolated from the MCU module ground, therefore not connected to the PE potential).
- ④ OE (Output Enable) indicator, bi-color: Green – output is active and uncoupled, Red – output is active and coupled.
- ⑤ Negative output (Vout-) power connector, 4 mm, “[floating](#)”
- ⑥ Positive output (Vout+) power connector, 4 mm (DCM220 only)
- ⑦ Channel indicator, bi-color (DCM220 only): Green – output is active in CV mode, Red – output is active in CC mode.
- ⑧ Negative output (Vout-) power connector, 4 mm (DCM220 only). Please note that Vout- of Ch1 and Ch2 are on the same potential but still “[floating](#)” (i.e. not connected to PE potential).

### 5.3. Rear panel overview



Fig. 7: The EEZ BB3 rear panel

- ① Low noise Ø80 mm cooling fan with speed algorithm based on multiple temperature sensors and output current readings.
- ② Kensington Security Slot™ (K-slot) for anti-theft cable.
- ③ RJ-45 Ethernet socket (10/100 Mbit/s)
- ④ AC power inlet with two 20 x 5 mm fuses

### 5.4. Display pages overview

The color display is used as a primary means of user interaction. Its content is dynamically rendered in order to enable quick and structured access to a multitude of system and module specific parameters. The touch-screen completely eliminates the need for specialized function keys, dedicated numeric keypad, etc. It takes no more than a couple of clicks to access any function, prioritizing access to those functions that are more important or used more frequently.

Fig. 8 shows home page of the EEZ BB3 loaded with three different power module (see Fig. 9) using the default view (*numeric*) for presenting output states, and measured and set values. This module configuration (i.e. DCP405 and DCM220) will be used as an example throughout this manual.

Items ① to ③ are module specific while the rest (displayed on the status bar) are used for accessing various system options.

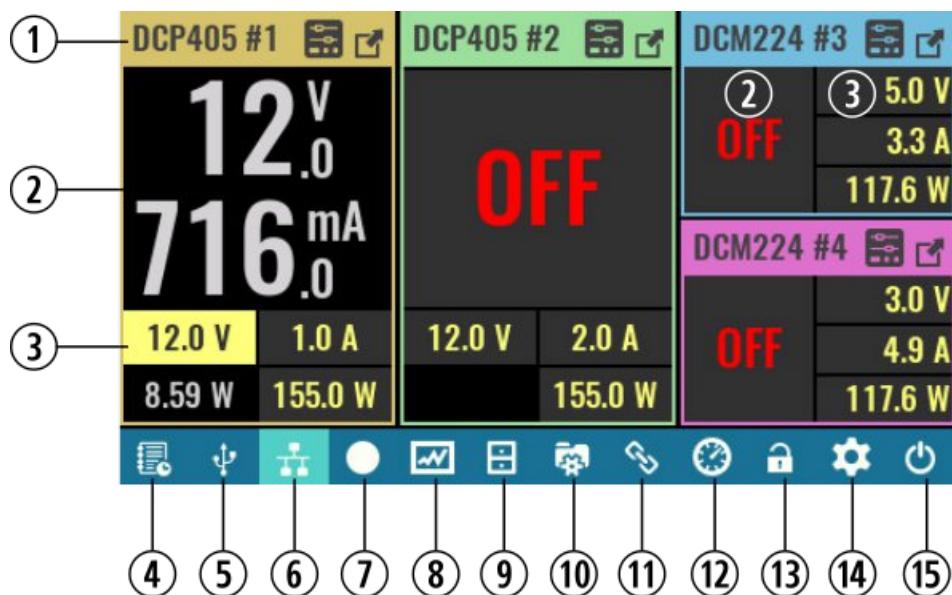


Fig. 8: Home page overview

- ① **Channel header:** for better distinction between modules different colors are used for each header. Module name and assigned channel number are displayed together with shortcuts to channel settings (a shortcut to set or clear protection modes – If any have been tripped – and a channel “maximize” button).
- ② **Channel main display section** contains information about measured output voltage and current. Units of measurement are also displayed depending on the measured value (i.e. mV or V for voltage or  $\mu$ A, mA or A for current).
- Note – this section looks different for 2-channel modules such as DCM220 due to more limited display area.*
- ③ **Channel supplemental display section** is used for set output voltage, current and power. Additionally is used for displaying measured output power together with related units of measurement (mW or W).
- Note – this section looks different for 2-channel modules such as DCM220 due to more limited display area.*
- ④ **Event view:** View the event log. The icon color will change color if unread warning or error messages have been added to the log.
- ⑤ **USB activity** indicator and shortcut to serial communication settings. The icon is not visible if USB/serial communication is disabled and its background becomes highlighted if USB/serial communication is established.
- ⑥ **LAN activity** indicator and shortcut to the Ethernet communication settings. The icon is not visible if Ethernet communication is disabled and its background becomes highlighted if LAN communication is established.
- ⑦ **Recording / Stop:** Data logger control. Use to start data logging or to stop logging before a defined end.
- ⑧ **Data log view:** if logging is in progress it displays real time data. When logging is completed or stopped, it will display data from the last log.
- ⑨ **File manager:** Allows access to data on a SD card such as user profiles, program lists, logged data or screenshots. Logged data and screenshots can be uploaded to EEZ Studio with one click.
- ⑩ **User profiles:** 10 different profiles (with auto-recall function) are available for saving and recalling module parameters.
- ⑪ **Tracking and coupling settings:** tracking allows simultaneous programming of two or more module's output channels. Coupling can be used to safely connect (under firmware control) two or more module's outputs in series, parallel, as “split rail” (i.e. series with common output) or with Vout- tied together. No additional external connection wiring is required, and in case of

coupling in series the output voltage range is doubled (e.g. to 80 V). Similarly, when parallel coupling is performed the output current range is doubled (e.g. to 10 A).

- (12) **Display view selection:** toggle between five different presentations of output module parameters (default is numeric view – as shown on Fig. 8).
- (13) **Display lock/unlock:** when locked all local user interaction via touch-screen display will be disabled until the same option is selected again. Unlocking is not possible without entering valid system password (the system password is *not* defined by default).
- (14) **System settings:** use it to access system-wide protection settings, display, sound, trigger, digital I/O, date & time, Encoder and user switch, AUX temperature & fan control, Ethernet, Serial/USB and channels calibration
- (15) **Power / Reset control:** allows access to standby, “soft” reset (see also [\\*RST](#)), “hard” reset (equivalent to power up reset) and option to turn off display

## 5.5. Module display views

Five different views are available to satisfy various use cases as well as user preference for numeric or graphic representation of measured output values.

Regardless of the selected display view and color theme each channel will be represented in consistent order with its assigned color. The same convention is also used when a channel is displayed over the whole screen (e.g. in the case of *Protection mode settings* or the *channel settings* page). In those cases where a channel is maximized the channel color is visible on the status bar instead of the header area.



Fig. 9: Display view icon

Use icon shown on Fig. 6 to toggle between all available display views.

### Numeric

Default display view with the largest font used to show measured voltage and current.

Output values of 2-channel modules like DCM220 are displayed with a regular font size due to lack of space.

**SCPI**  
DISPLAY:VIEW 1

DCP405 #1	DCP405 #2	DCM224 #3	DCM224 #4
8 V 1.19	12 V 1.0	2.84 V 1.97 A 5.59 W	3.0 V 3.3 A 117.6 W
1 A .0	1 A .4295	1.64 V 940.0 mA	12.0 V 1.0 A
12.0 V 8.19 W	12.0 V 17.15 W	1.54 W	117.6 W
155.0 W	155.0 W		

### Vertical bar

This view is useful when the output values are changing rapidly.

Currently measured values are displayed with regular font size, hence user has to be closer to the EEZ BB3 to read the displayed values.

**SCPI**  
DISPLAY:VIEW 2

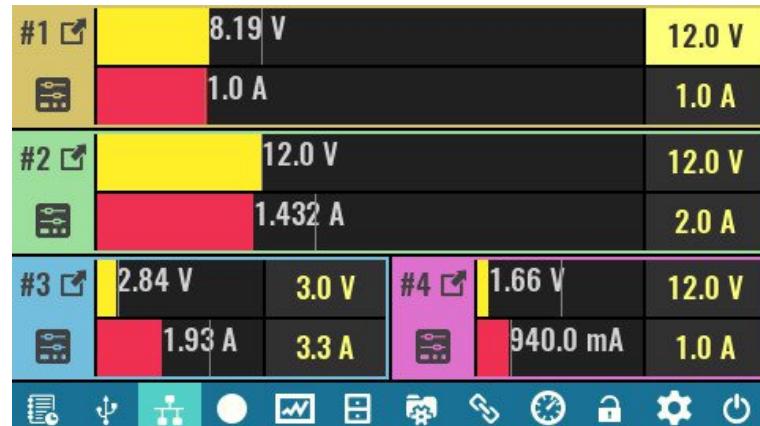
DCP405 #1	DCP405 #2	DCM224 #3	DCM224 #4
8.19 V 12.0 V	1.0 A 1.4315 A	2.84 V 1.94 A 5.51 W	3.0 V 3.3 A 117.6 W
1.0 A	2.0 A	12.0 V 940.0 mA	12.0 V 1.0 A
155.0 W	155.0 W	1.58 W	117.6 W
17.18 W			

## Horizontal bar

Similarly to previous view but output values are displayed as horizontal bars.

This view allows graphical presentation of the 2-channels modules, too.

**SCPI**  
DISPLAY:VIEW 3

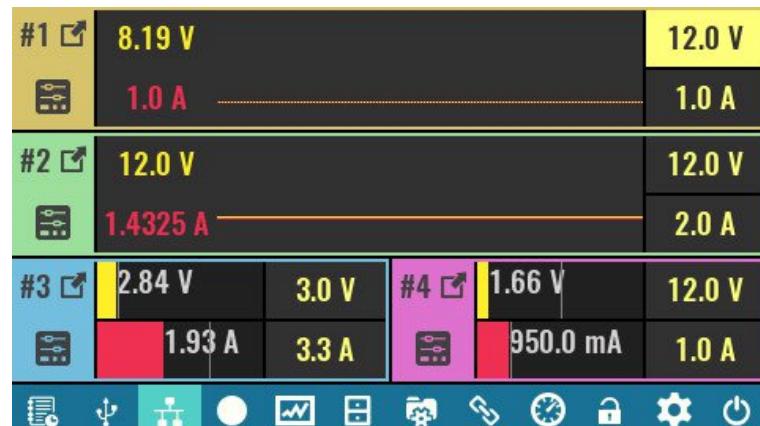


## YT view (scan)

Similar to the previous view but a moving cursor is not used, and all data displayed moves from right to left with the speed defined by *YT view sampling rate* (default is 100 ms).

This type of presentation is also known as *roll mode*.

**SCPI**  
DISPLAY:VIEW 4

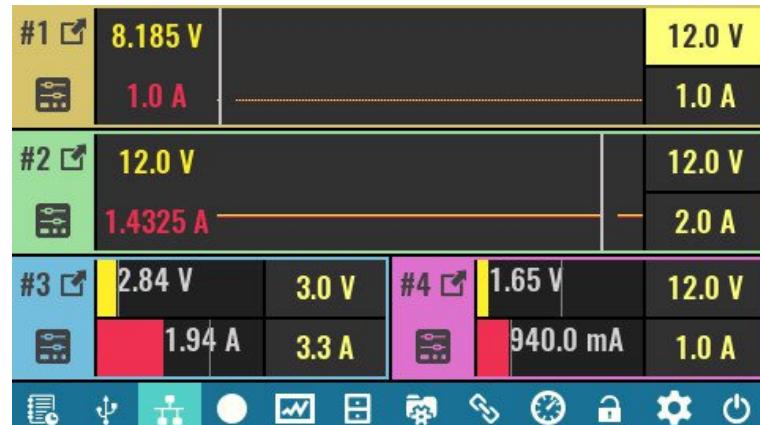


## YT view (scroll)

Measured values are displayed on the Y-axis of a 2D graph. The cursor position moves with the speed defined by *YT view sampling rate* (default is 100 ms).

The cursor moves cyclically – rewriting older data with new data.

**SCPI**  
DISPLAY:VIEW 5

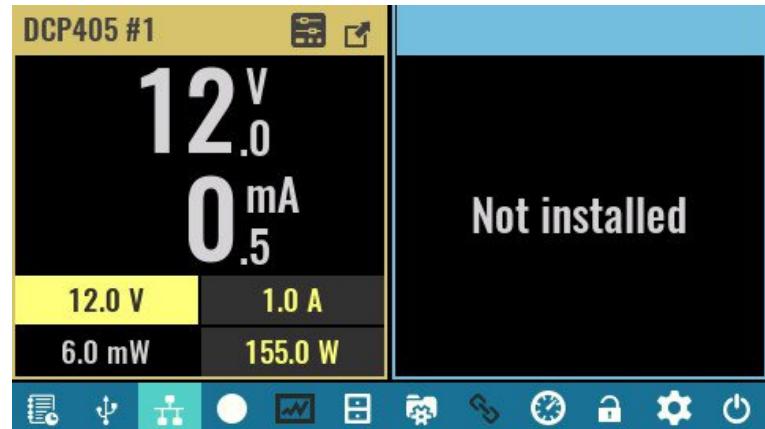


### 5.5.1. Two modules display view

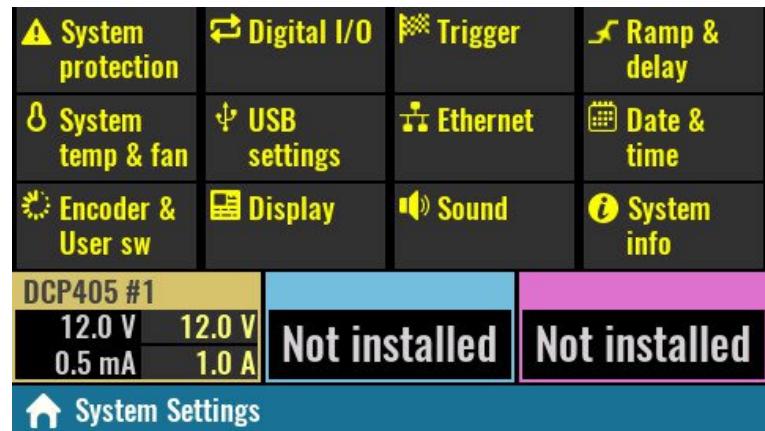
When all three slots are not populated with modules the two modules display view is used. otherwise one third of the screen would be unused (in that part the message *Not installed* would be displayed continuously).



However, if only one module is installed and maximized view is not used, *Not installed* message will be displayed on half of the screen.



*Not installed* message will also appear in place of the empty slot on the System settings page.



## 6. Getting started

### 6.1. Powering on

Before switching on check that [Safety Requirements](#) are observed. Connect the AC line power cable to the IEC power socket on the rear panel and switch on the main power switch on the front panel. The welcome page will appear (Fig. 10) and a self-test procedure will be performed to detect the installed modules and diagnose any error conditions which may inhibit normal operation.

The EEZ logo on the welcome page can be replaced with a custom logo in jpg or .bmp format which can be max. 480 x 136 pixels in size. The image file should be stored in the root (/) folder on the SD card (logo.jpg or logo.bmp).

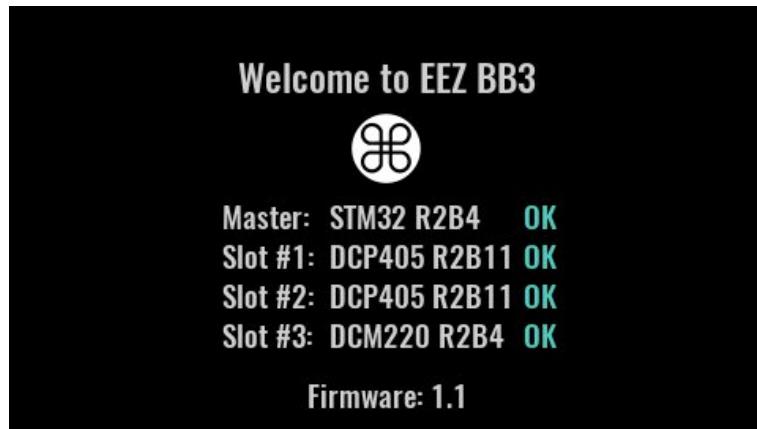


Fig. 10: Welcome page

On the first power up all channel outputs are turned off to prevent any connected load from being damaged unintentionally.

The channel output state at subsequent power-ups will depend on what is defined in the selected user profile. A special option in [System settings](#) can also be set which overrides the state set by the particular selected user profile.

The default display view is *numeric* and the default color theme is *dark*. Both can be changed. The home page of the 4-channel configuration example is shown in Fig. 11:

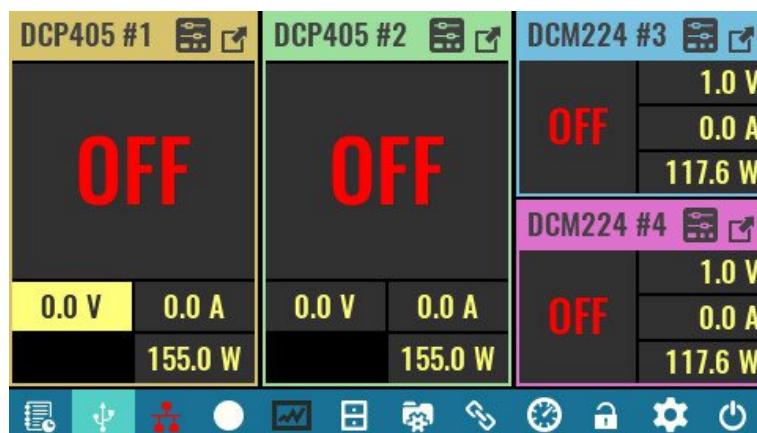


Fig. 11: Home page on first start

### 6.2. Standby mode and switching off

The EEZ BB3 offers two options to switch off / power down: entering standby mode or using AC power switch on the front panel. In both cases all installed modules are disconnected from AC power. In standby mode the MCU module remains powered (as indicated by the *Standby* indicator on the front panel). Fig. 12 shows the user interface prior to entering Standby mode and the info message that is fading out. To resume normal operation from standby mode, tap any part of the display and wait a moment for the power-up procedure to complete.



Fig. 12: Entering standby mode

**SCPI**

SYSTem:POWer OFF

The main power switch on the front panel can be used in any situation to immediately cut off AC power. However, if it is not an emergency situation, it is recommended to initiate a graceful shutdown that ensures that the latest information are stored in non-volatile memory. Use the option as shown in Fig. 13 and wait until the message appears that your EEZ BB3 can be switched off using the front panel switch.

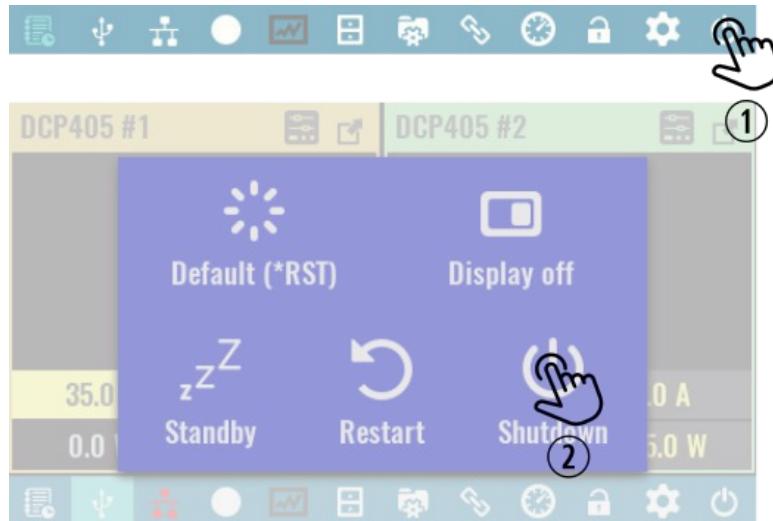


Fig. 13: Initiate safe shutdown

## 7. Basic operations

### 7.1. Turn channel output on and off

#### Turn channel on (OE on)

Turning the channel output on is performed by tapping anywhere inside the main channel display area.

The changed output state is indicated by display of the measured output values instead of "OFF".

#### SCPI

OUTPut ON

OUTPut ON, <channel>



#### Turn channel off (OE off)

Turning the channel output off is performed by tapping anywhere inside the main channel display area. The content of that area will be replaced with "OFF".

#### SCPI

OUTPut OFF

OUTPut OFF, <channel>



### 7.2. Set channel output values

Channel output values can be set regardless of the currently selected display view and channel output state as long as a trigger mode is not active (i.e. channel's trigger is set to *Fixed*).

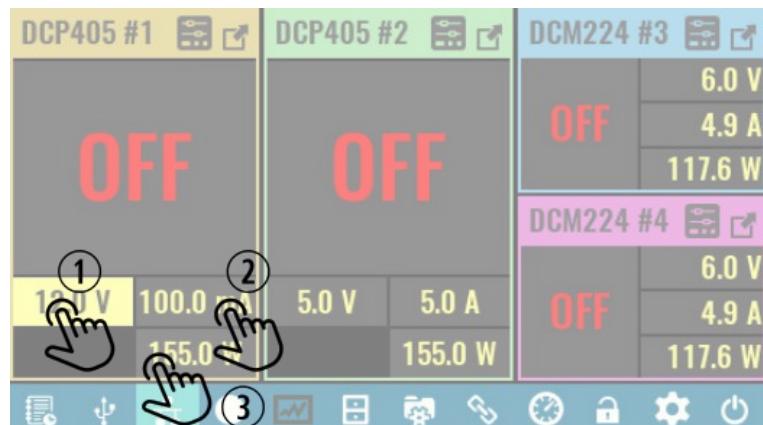
By default clicking the encoder switch navigates between editable parameters; when a parameter is highlighted (as shown in the figure below) its value can then be set with the encoder knob.

#### ① Set output voltage

Tap inside the *voltage* area of the channel secondary display area. If it is already selected (as shown), a keypad will appear as described in [Data input methods](#). The voltage set here is the max. output voltage when channel is in CV mode.

#### SCPI

VOLTage <voltage>



## ② Set output current

First tap the *current* area of the secondary channel display. A second tap is required if this option was not selected previously (as in the above example).

The current set here is the max. output current when channel is in constant current (CC) mode.

**SCPI**

CURRent <current>

## ③ Set output power limit

Follow the above described procedure.

*Note that when a power output limit is specified, it is the product of the voltage and current parameters used to calculate the output power limit rather than the actual measured output values. For example, if power limit is set to 100 W (DCP405) then set current cannot exceed 2.5 A for a max. voltage of 40 V. Similarly the output voltage cannot be set above 20 V if max. current is selected as 5 A even if the actual measured output current with a connected load is well below the set limit.*

**SCPI**

POWER:LIMit <power>

## 7.3. Data input methods

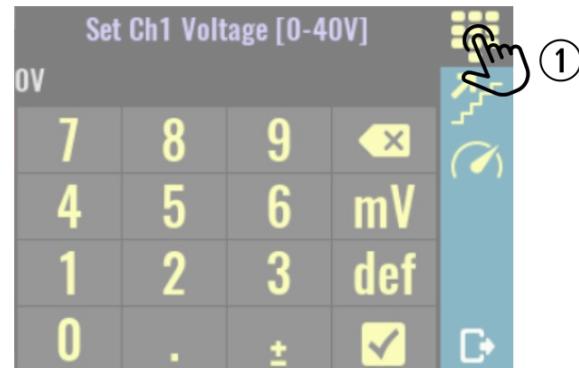
In addition to the physical rotary encoder on the front panel there are three touch-screen options to enter data. Two of those (*Step* and *Slider*) can be used in *non-confirmation* (default) or *confirmation* mode, or, in combination with the encoder knob. All data input methods are accessed from the same popup menu with the three options shown as tabs on the right side. Tapping the icon will display that particular data input tab.

### Keypad

- ① This is the default data entry option.  
The keypad is used to enter a new value for the parameter that is displayed in the header (Ch1 voltage in this example).

You can also select unit of measurement if more than one is available (e.g. mV and V for voltage).

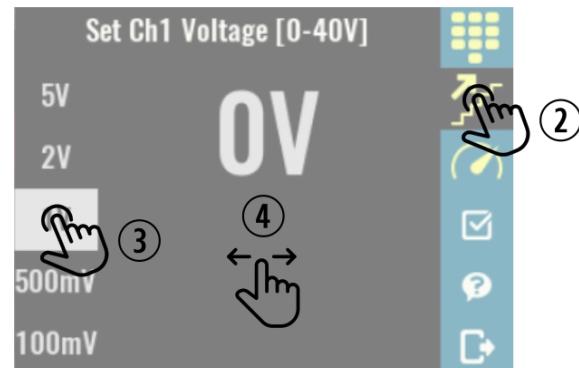
This data entry method always requires confirmation.



### Step

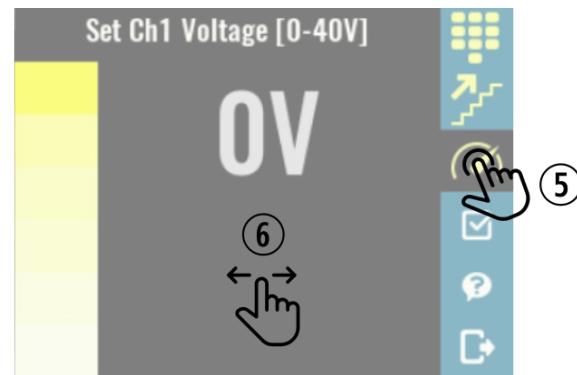
- ② Tap on *step* method icon.
- ③ Select step value or increment.
- ④ Sliding across the main part of the window will increase (right) or decrease (left) the value of the selected parameter using the increments selected.

The encoder knob can also be used to change the value. (1 V is the selected increment in this example).



**Slider**

- ⑤ Tap on *slider* method icon.
- ⑥ Sliding across the main part of the window will increase (right) or decrease (left) the value of the selected parameter.

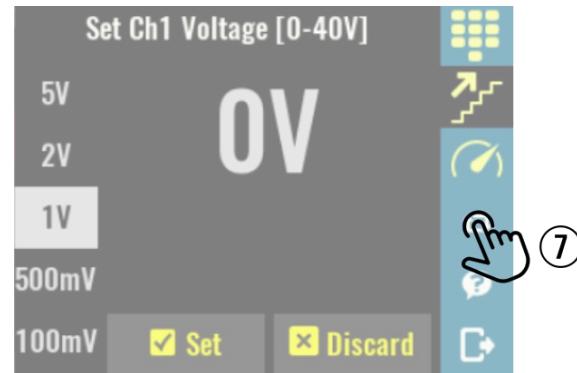
**Confirmation mode**

- ⑦ Confirmation mode blinks the new value of the setting until it is confirmed by pressing the *Set* button, at which point the new value is applied. The parameter value can be reset to the previous value using the *Discard* button.

The set and discard buttons are shown on the bottom of the screen.

To enable or disable confirmation mode tap the checkbox icon on the right.

Note: confirmation mode can be used in *Step* and *Slider* input modes only.



## 7.4. Accessing channel protections and settings

The EEZ BB3 firmware includes comprehensive control of each power module and the feature list goes well beyond the basic operation of setting and displaying output parameters. Other channel specific functions are shown in the *Protections and settings* page.

The channel *Protections and settings* page can be reached whenever the channel is in regular or maximized view.

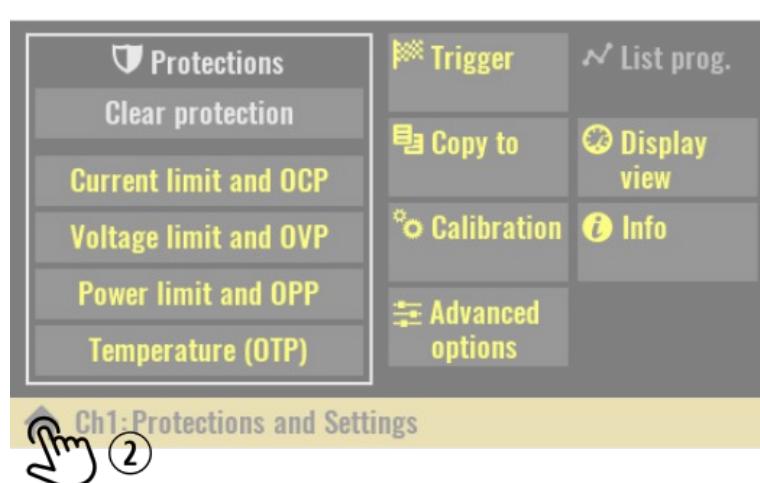
Note that if any of channel's protection modes have tripped, tapping the *Protection* icon will take you to the *Clear / Disable protection* page – regardless of whether the channel is in regular or maximized view.

**① Access protections and settings in regular view**

Select the channel whose settings you want to change and tap on its *Protection* icon.

A new page with channel's protection section and channel settings will be displayed.

Availability of certain options depends on the channel mode (e.g. when channel is in tracking or coupling mode *Calibration* is disabled) and capability (e.g. DCM220 module does not have any *Advanced options*).

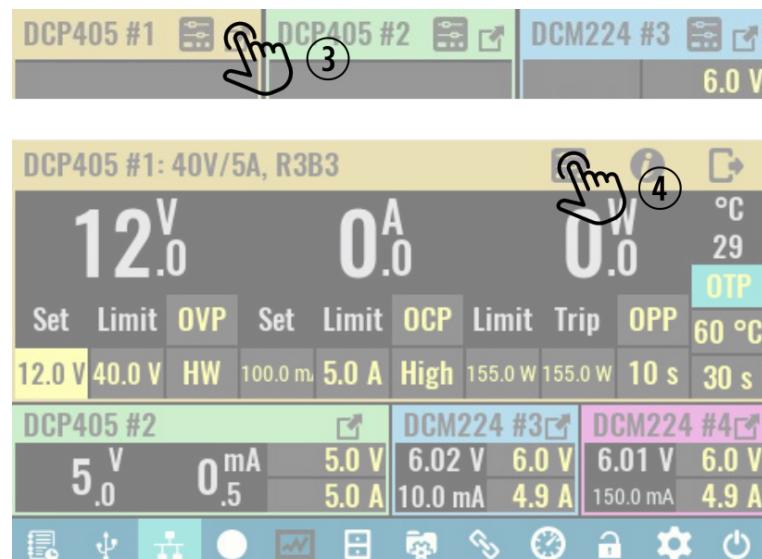


- ② Exit *Protections and settings* page.

- ③ Access protections and settings from maximized view

If channel is in regular view, tap on the channel's *Maximize* icon.

- ④ Tap on its *Protection* icon.



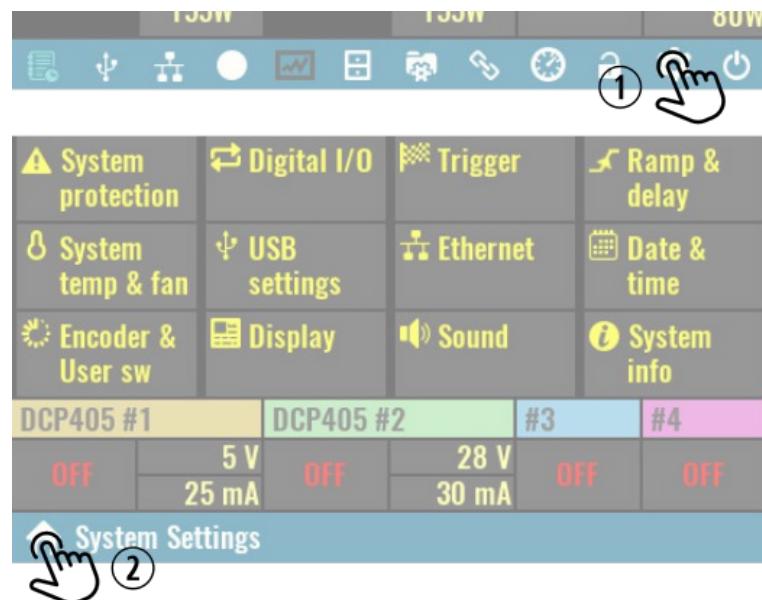
## 7.5. Accessing system settings

- ① Tap on *System settings* icon on the status bar.

The number of available options depends of firmware version.

When System settings page is displayed it is still possible to check channel's output state and values (e.g. in this example all outputs are turned off).

- ② Exit *System settings* page.



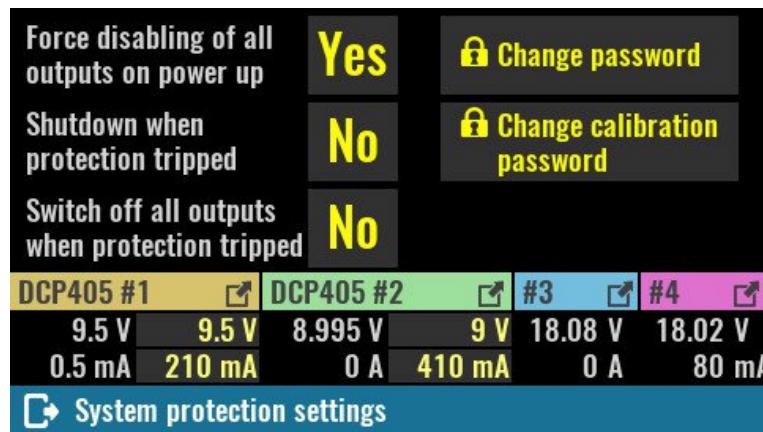
## 8. System settings

- [System protection settings](#)
- [Digital I/O settings](#)
- [General trigger settings](#)
- [Ramp and delay](#)
- [System temperature and fan](#)
- [USB settings](#)
- [Ethernet interface](#)
- [Date and time](#)
- [Encoder and User switch](#)
- [Display settings](#)
- [Sound settings](#)
- [System Information](#)

### 8.1. System protection settings

Parameters in this section define channel output states on power up and options when a protection mode is tripped to prevent or reduce the possibility of damaging the connected loads.

Additionally, the system password that is used to [unlock the display](#) and calibration password can be defined or changed here.



#### Force disabling of all outputs on power up

Active by default this option ensures that all channel outputs will be switched off on power up to prevent connected loads from being damaged unintentionally.

#### SCPI

```
SYSTem:POWer:OUTPut:DISable ON
```

#### Shutdown when protection tripped

When selected, any protection mode that is tripped on any channel will automatically put the EEZ BB3 in standby mode.

#### SCPI

```
SYSTem:POWER:PROTection:TRIP ON
```

#### Switch off all outputs when protection tripped

Similar to the previous option except instead of entering standby mode all channel outputs will be turned off.

#### SCPI

```
OUTPut:PROTection:COUPLE ON
```

#### Change password

The system password is used to unlock the display. By default system password is not defined ("") and it may contain up to 16 characters. Minimum length is 4 characters. The new password is automatically stored in non-volatile memory.

When selected an on-screen keyboard will appear. Characters entered will be replaced with \* shortly after entry. The



new password has to be entered twice.

**SCPI**

```
SYSTEM:PASSWORD:NEW {<old>} , {<new>}
```

**Change calibration password**

The calibration password is used to secure access to calibration data. Unlike a system password this password is defined by default (eezbb3) and has to be entered first before new password can be entered.

The calibration password may contain up to 16 characters. Minimum length is 4 characters.

The new password is automatically stored in non-volatile memory.

**SCPI**

```
CALibration:PASSWORD:NEW {<old>} , {<new>}
```

**8.2. Digital I/O settings**

Two protected/buffered digital input pins plus two digital output pins are available on the EEZ BB3 [front panel](#).

Their function and polarity can be set on this page which also displays their current state.

**Function**

Input pins may be assigned with one of the following functions:

- Input* – The pin is in digital input mode.
- Inhibit* – When pin is configured as an inhibit input, a true signal at the pin will disable all output channels.  
If the channel outputs were turned on, all output values will flash in red when entering inhibit mode, as shown in the figure to the right.
- Trigger input* – When configured as a trigger input, the pin can be selected as the source for trigger signals.

Name:	Din1	Din2	Dout1	Dout2
Pin #:	1	2	3	4
Function:	None	None	None	None
Polarity:	Neg	Neg	Neg	Neg
State:	Unassigned	Unassigned	Unassigned	Unassigned

DCP405 #1	22.5V	DCP405B #2	12V	#3	#4
OFF	5A	OFF	2A	OFF	OFF

↳ Digital I/O pin settings

DCP405 #1	0 V	DCP405 #2	0 V	DCM224 #3	0.0 V	5.0 V
0.0	0.0	0.0	0.0	0.0 A	1.0 A	
0 A	0.0	0 A	0.0	0.0 W	117.6 W	
5.0 V	1.0 A	6.0 V	1.0 A	0.0 V	8.0 V	
0.0 W	117.6 W	0.0 W	117.6 W	0.0 A	1.0 A	
				0.0 W	117.6 W	

Output pins could be assigned with one of the following functions:

- Output* – The pin is in digital output mode.
- Fault* – pin functions as an isolated fault output. The fault signal is true when any output is in a protected state (from OCP, OVP, OTP, OPP) or Fan fault is detected.
- Channel ON couple* – pin synchronizes channel output state.
- Trigger output* – This allows a BUS trigger to be sent to any digital port pin that has been configured as a trigger output. A trigger out pulse (100 ms) is generated when the state is on and a bus trigger is received. A BUS trigger is generated using the [\\*TRG](#) command.
- PWM* – (Dout2, pin 4 only) when selected, a square wave will be generated with set frequency from 0.03 Hz to 5 MHz and duty from 0 to 100 %.

**SCPI**

```
SYSTem:DIGItal:PIN<n>:FUNCTION {<function>}
```

```
SYSTem:DIGItal:OUTPut:PWM:FREQuency {<pin>} , {<frequency>}
```

```
SYSTem:DIGItal:OUTPut:PWM:DUTY {<pin>} , {<duty>}
```

**Polarity**

- *Pos* – a logical true signal is a voltage high at the pin. For trigger inputs and outputs, positive means a rising edge.
- *Neg* – a logical true signal is a voltage low at the pin. For trigger inputs and outputs, negative means a falling edge.

**SCPI**

SYStem:DIGItal:PIN&lt;n&gt;:POLarity {&lt;polarity&gt;}

**State**

Current state of all digital inputs and outputs are displayed in this section. If the pin function is not yet defined, the displayed state will be *Unassigned*.

**SCPI**

SYStem:DIGItal:INPut:DATA? {&lt;pin&gt;}

SYStem:DIGItal:OUTPut:DATA? {&lt;pin&gt;}

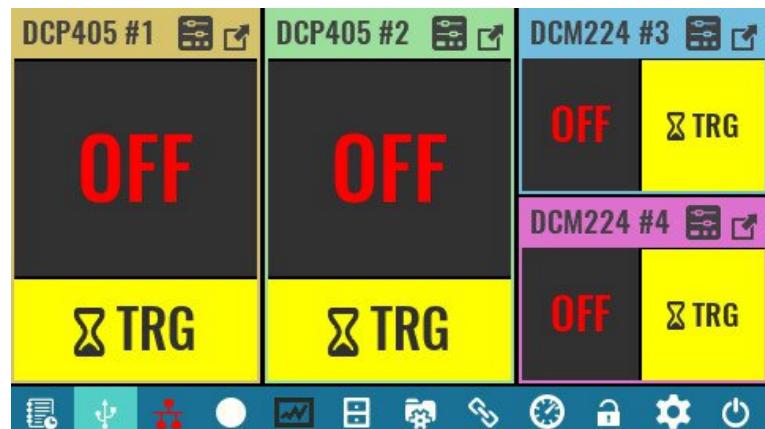
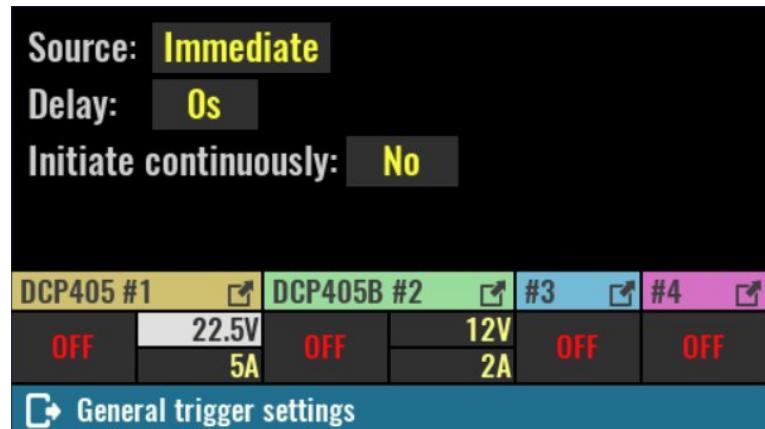
**8.3. General trigger settings**

The BB3's triggering system allows a change in output state, voltage and current or start internal data logging when receiving a trigger from selected trigger source.

One common triggering system is used to control all available channels, but will only affect those channels that are not in *Fixed* trigger mode (see [Channel trigger settings](#) for channel trigger mode settings).

**Source**

- *Bus* – enables LAN and serial (via USB) triggering using the [\\*TRG](#) command. When this mode is selected, *TRG* text will blink when you turn on channels that have triggers defined (e.g. *List* or *Step*).
- *Immediate* – the EEZ BB3 executes a complete trigger operation immediately after executing the [INITiate](#) command without delay.
- *Manual* – allows you to manually trigger with the User SW (if assigned this function, see [User SW](#)) or by selecting the flashing TRG text that will appear on the screen as it shows the example in the picture to the right (Ch2 to Ch4 have triggers defined).
- *Pin<n>* – selects a digital port pin configured as a trigger input. <n> specifies the pin number.

**SCPI**

TRIGger[:SEQUence]:SOURce {&lt;source&gt;}

**Delay**

Sets the time delay between the detection of an event on the specified trigger source and the start of any corresponding trigger action on the channel output.

**SCPI**

TRIGger[:SEQUence]:DELay {&lt;delay&gt;}

### Initiate continuously

This option defines whether the trigger system is continuously initiated or not. When set to OFF, the system remains in the IDLE state until it is set to ON or an *INITiate:IMMediate* command is received. Once it is set to ON, the trigger system will be initiated and exit the IDLE state. On completion of each trigger cycle, the trigger system immediately commences another trigger cycle without entering the IDLE state.

When this option is set to OFF, the current trigger cycle will be completed before entering the IDLE state. The return to IDLE also occurs as the result of an *ABORT* or *\*RST* command.

### SCPI

```
INITiate:CONTinuous {<bool>}
```

### 8.4. Ramp & delay

This page will show all output channels that have support for ramp and delay.

Ch.	Voltage ramp	Current ramp	Out. delay
<input type="checkbox"/> #1			
<input type="checkbox"/> #2			
<input type="checkbox"/> #3			
<input type="checkbox"/> #4			
	 Ramp and delay		

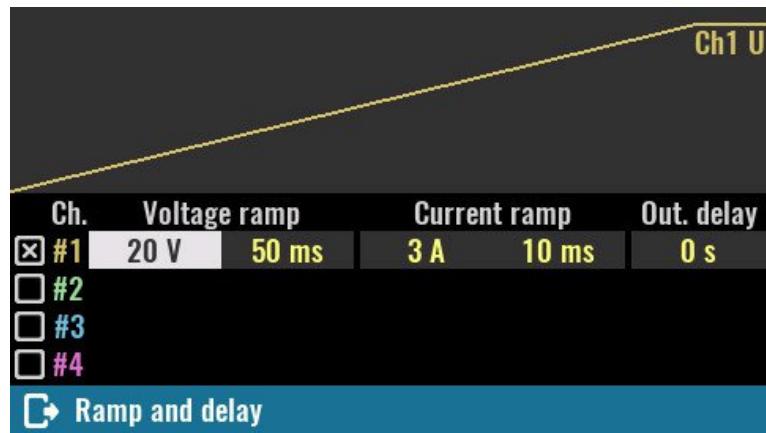
Select the channel you want to define ramp and delay settings for.

Ch.	Voltage ramp	Current ramp	Out. delay
 #1			
<input type="checkbox"/> #2			
<input type="checkbox"/> #3			
<input type="checkbox"/> #4			
	 Ramp and delay		

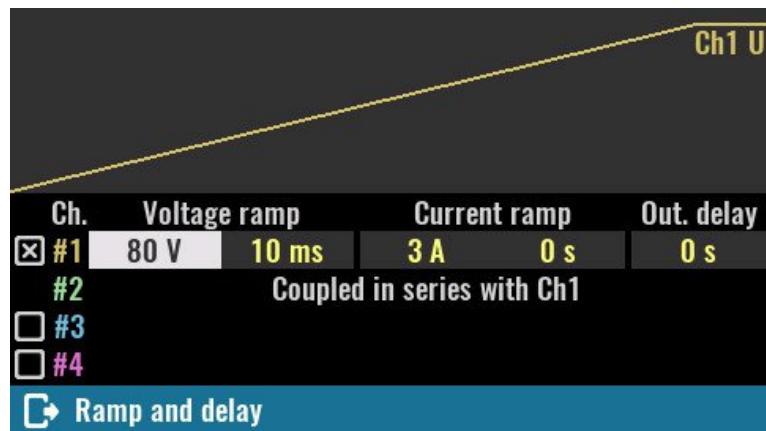
Tap on the parameter that you want to set.

Ch1 U					
Ch.	Voltage ramp	Current ramp	Out. delay		
<input checked="" type="checkbox"/> #1	 10 ms	0 A	10 ms	0 s	
<input type="checkbox"/> #2					
<input type="checkbox"/> #3					
<input type="checkbox"/> #4					
	 Ramp and delay				

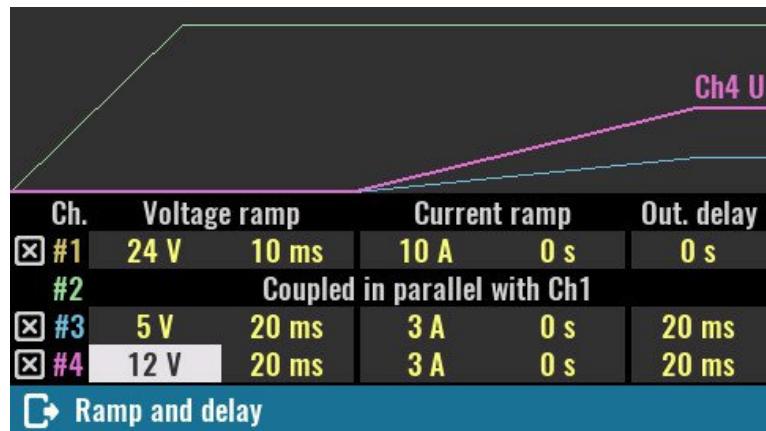
The graph above the list shows the output profile of the selected channel.



The channel list will reflect coupling in series. For example, the voltage ramp is able to be set at twice the value (up to 80 V).



This screenshot shows an example of a complex ramp and delay setup where the first two output modules are coupled in parallel and the last two modules turn on after a delay of 20 ms.



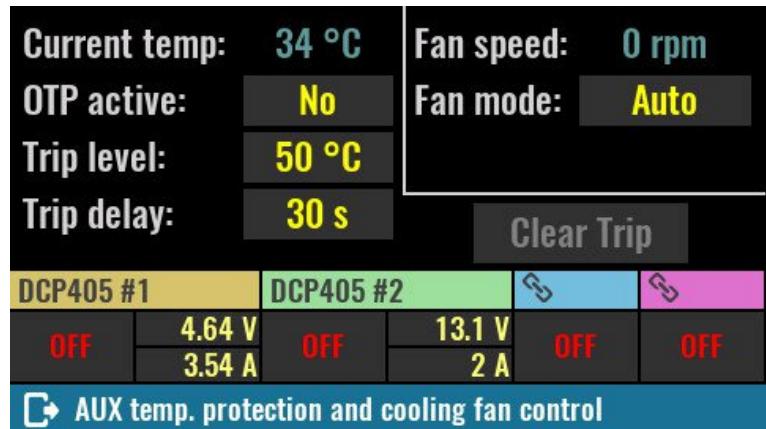
## 8.5. System temp and fan

### Current temp.

The "system" temperature is measured on the AUX-PS module and can be used for over-temperature protection (OTP).

### SCPI

MEASure:TEMPerature? AUX



### OTP active

Enable or disable over-temperature protection associated with the system (AUX) temperature sensor.

**SCPI**

SYSTem:TEMPerature:PROTection:STATE

**Trip level**

Set the over-temperature protection (OTP) value in degrees Celsius (°C). Default value for this sensor is 45 °C.

**SCPI**

SYSTem:TEMPerature:PROTection {&lt;temperature&gt;}

**Trip delay**

Specifies how long the temperature should be equal to or higher than the set *trip level* for the protection to activate.

**SCPI**

SYSTem:TEMPerature:PROTection {&lt;temperature&gt;}

**Clear trip**

When protection is activated (tripped), this condition will be latched and further usage of the EEZ BB3 will be disabled until protection is cleared.

**SCPI**

SYSTem:TEMPerature:PROTection[:HIGH]:CLEar

**Fan speed**

Current fan speed indicator stated in rpm (revolutions per seconds). The diagnosed fan fault will be also displayed here.

**Fan mode**

The fan is controlled by default with firmware algorithm (*Auto* mode) that takes into account measured temperature on all temperature sensors and measured output currents (i.e. the higher the output current, the sooner the fan speed will increase).

*Manual* mode can be selected for e.g. fan testing purpose when firmware speed control algorithm is completely bypassed.

**Set speed to**

This option becomes available when manual mode is selected. Default value is 100 % (full speed).

**SCPI**

SYSTem:FAN:STATus?

SYSTem:FAN:SPEEd?

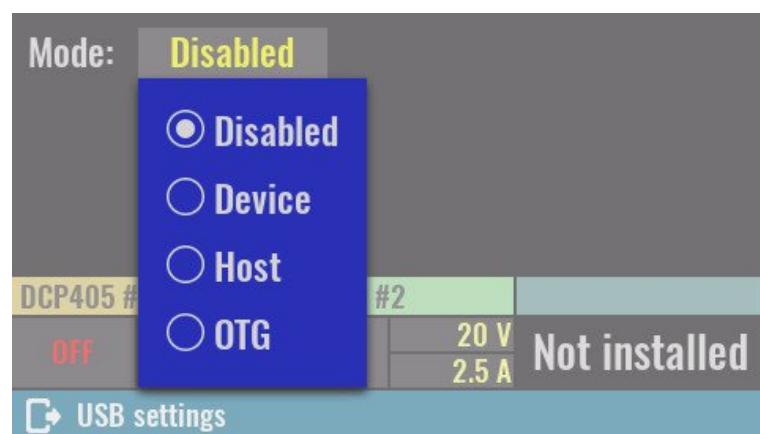
**8.6. USB settings****Mode**

Sets the operating mode or turns off the USB interface.

**SCPI**

SYSTem:COMMUnicatE:ENABLE {&lt;bool&gt;}, USB

SYSTem:COMMUnicatE:USB:MODE



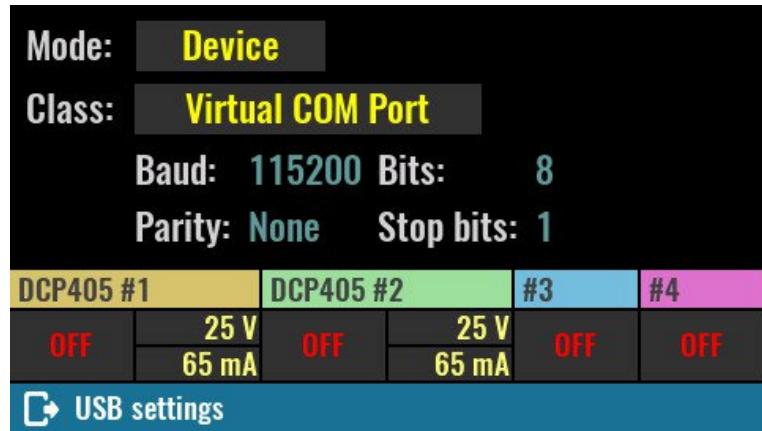
**Virtual COM Port**

Sets the parameters of the serial interface over USB connection (i.e. Virtual COM port).

When selected serial communication parameters will be displayed (*Baud*, *Bits*, *Parity* and *Stop bits*).

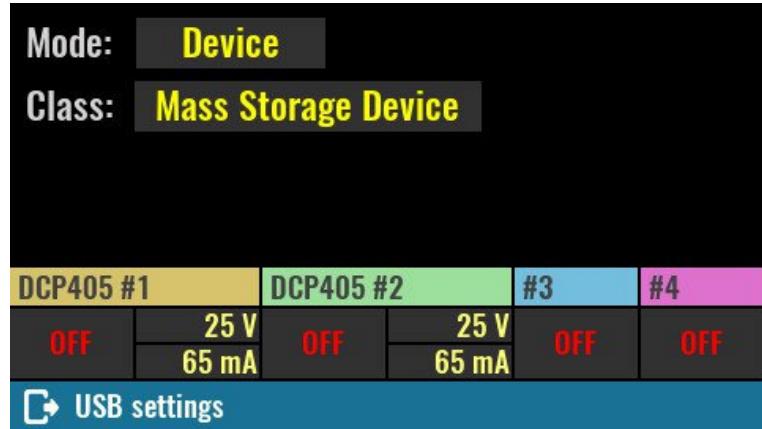
**SCPI**

SYStem:COMMunicate:USB:CLAss  
VCOM

**Mass storage device**

When this class is selected the operating system of the connected computer will see the installed SD card as a mass storage device. It will be possible to work with the SD card as well as other disks (browse, read, write, delete, create new folders, rename, etc.).

The selection of this class will be valid until the next restart when it will return to *Virtual COM port* again.

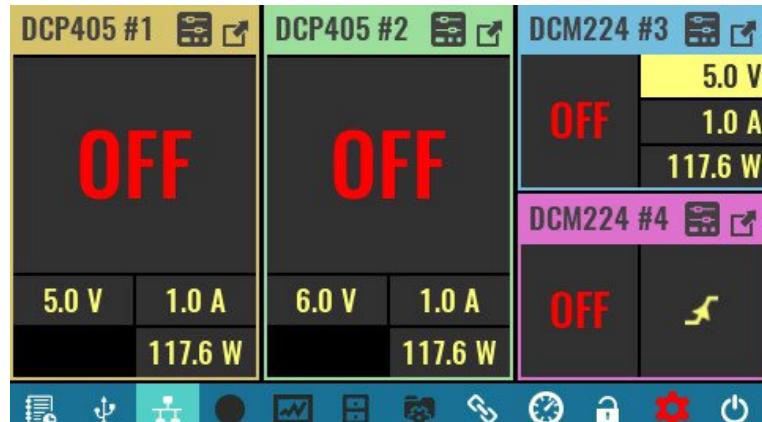


**IMPORTANT:** When the SD card is in this mode, EEZ BB3 cannot save events to log, access the file manager or user profiles, etc.

For this reason, there will those icons will be disabled in the status bar. The system settings icon will turn red because the status of the SD card will change to Not present.

**SCPI**

SYStem:COMMunicate:USB:CLAss  
MSTO

**Host**

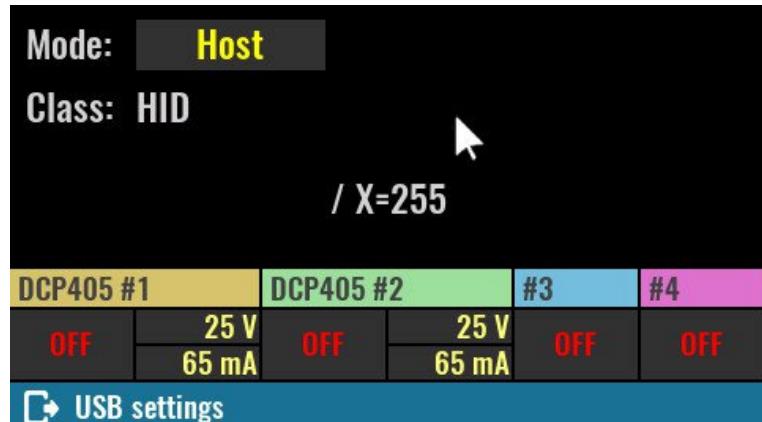
When this mode is selected the EEZ BB3 provides VBUS power for connected USB device (e.g. mouse, keypad, foot pedal, etc.).

The USB Class will be automatically set to HID.

USB mouse (wired or wireless) is supported and if connected, the cursor will appear and information on the cursor coordinates will be displayed.

The USB keyboard is partially supported so that the following functions are enabled:

- [Tab] + [Space]/[Enter] as navigation and selection

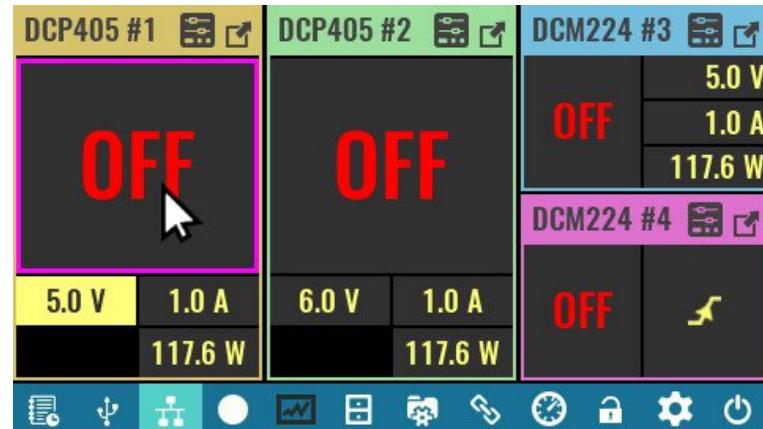


- [Print Scrn] for taking screenshot
- [Home] returns to home page
- [Esc] can be used as exit/cancel option
- Numeric keys [1] to [6] as Ch1 to Ch6 output state toggle (on/off)

**SCPI**

SYSTem:COMMunicate:USB:MODE HOST

When a USB mouse is used for navigation, the clickable item on which the cursor is positioned above will be highlighted with a magenta box.

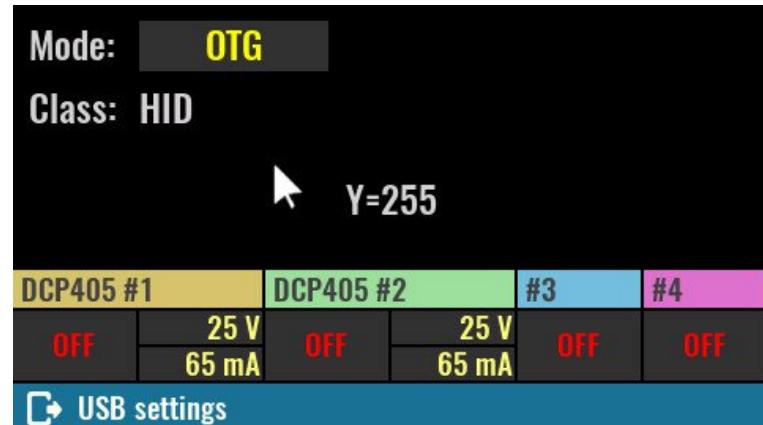
**OTG**

“On-the-go” mode allows detection of USB ID pin (set by USB cable or adapter) and adopt its USB mode and class in accordance with detected device.

For example if mouse is connected with OTG adapter cable, the *USB Mode* will be set automatically to Host and *Class* to HID and mouse cursor will appear.

**SCPI**

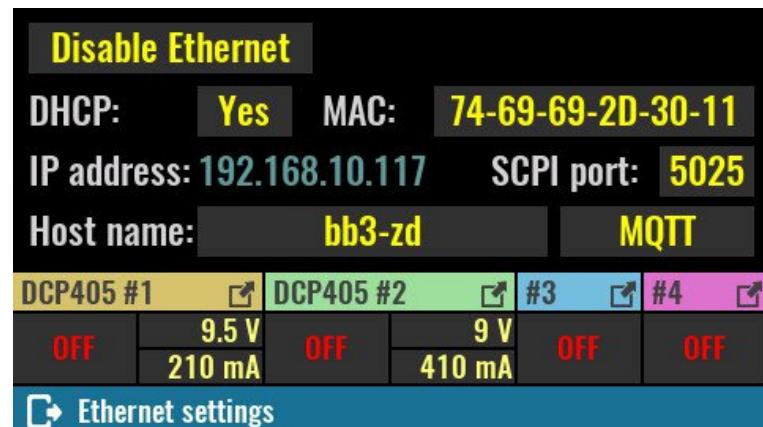
SYSTem:COMMunicate:USB:MODE OTG

**8.7. Ethernet****Enable/disable**

Defines activity of the serial interface. When LAN cable is disconnected or network communication is not yet established, *Connecting ...* will be displayed until communication is established again.

**SCPI**

SYSTem:COMMunicate:ENABLE {&lt;bool&gt;}, ETHernet

**DHCP**

Enable or disable the DHCP mode. In DHCP mode, the DHCP server in the current network assigns network parameters (*IP address*, *DNS* address, *Gateway* address and the *Subnet mask*). Assigned IP address will be displayed, and cannot be changed.

When DHCP is disabled all network parameters have to be entered manually using the *Edit static address* as described below.

**SCPI**

SYSTem:COMMunicate:ETHernet:DHCP {&lt;bool&gt;}

**MAC**

Defines Ethernet MAC address, a unique identifier assigned to Ethernet interface for communications at the data link layer of a network segment. Any combination of six hexadecimal values is allowed separated by “-”. The Ethernet connection will work as long as two different machines in the LAN don't have the same MAC address.

**SCPI**

```
SYSTem:COMMUnicatE:ETHernet:MAC {<mac_address>}
```

**SCPI port**

Defines Ethernet communication port for SCPI connections. Default port is 5025.

**SCPI**

```
SYSTem:COMMUnicatE:ETHernet:PORT {<number>}
```

**Host name**

Defines local area network (LAN) connection unique host name. Host name could contains up to 63 alphanumeric characters. “-” is allowed only if is not used as the first character. Space is not allowed and default name is *EEZ-BB3*.

**SCPI**

```
SYSTem:COMMUnicatE:ETHernet:PORT {<number>}
```

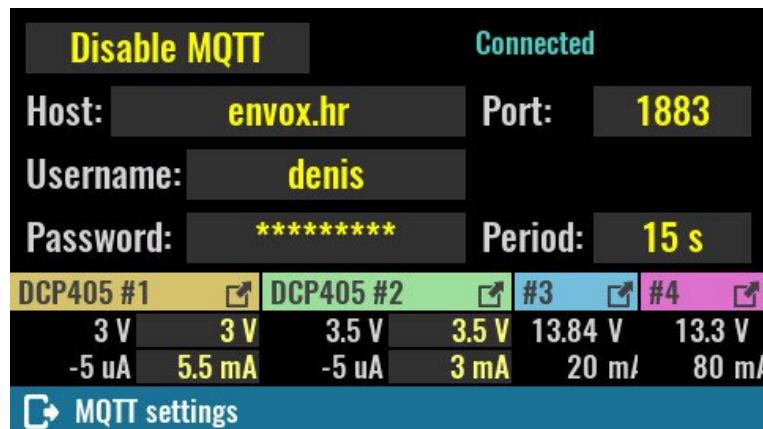
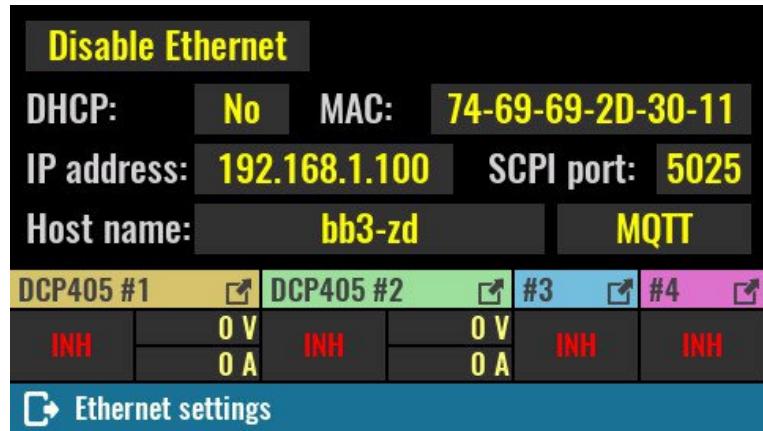
**8.7.1. MQTT**

When MQTT is enabled, the following parameters should be defined to establish communication:

- *Host* – The name of MQTT server (aka broker)
- *Port* – TCP/IP port for MQTT communication (standard port is 1883)
- *Username* – login name of a valid account on the MQTT server
- *Password* – login password of a valid account on the MQTT server
- *Period* – MQTT messages publishing frequency

**SCPI**

```
SYSTem:COMMUnicatE:MQTT:SETTings {<address>}, {<port>}, {<user>}, {<password>}, {<period>}
```



## 8.7.2. Static LAN settings (DHCP disabled)

The parameters that follows should be defined when *DHCP* is not selected. All values should be specified in IPv4 address format (four values from 0 to 255 separated by dots).

### IP address

Static local area network (LAN) address of the BB3.

#### SCPI

```
SYSTem:COMMUnicatE:ETHernet:A  
DDReSS {<ip_address>}
```

### DNS

IP address of the DNS (Domain Name Service) server that translates domain names into IP addresses.

#### SCPI

```
SYSTem:COMMUnicatE:ETHernet:D  
NS {<ip_address>}
```

### Gateway

IP network gateway address for accessing the EEZ BB3 from outside the current sub-network.

#### SCPI

```
SYSTem:COMMUnicatE:ETHernet:GATEway {<ip_address>}
```

### Subnet mask

IP network subnet mask. The subnet mask is used to determine if a client IP address is on the same local subnet.

#### SCPI

```
SYSTem:COMMUnicatE:ETHernet:SMASK {<mask>}
```

## 8.8. Date & time

### Date

Sets the date of the system clock (RTC). Specify the year, month, and day.

#### SCPI

```
SYSTem:DATE {<year>},  
{<month>}, {<day>}
```

### Time

Sets the time of the system clock (RTC). Specify the hours, minutes, and seconds.

#### SCPI

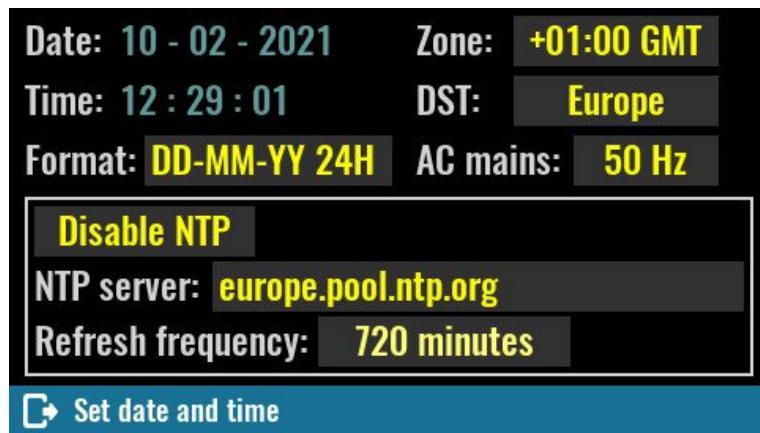
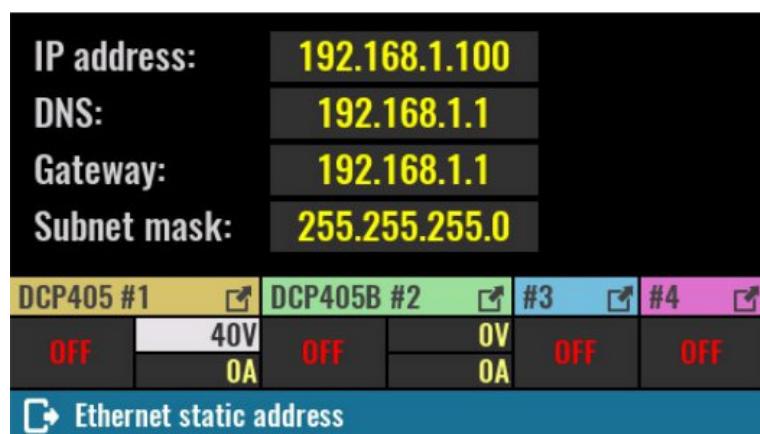
```
SYSTem:TIME {<hours>}, {<minutes>}, {<seconds>}
```

### Zone

Defines time zone as offset from [GMT](#).

#### SCPI

```
SYSTem:TIME:ZONE {<zone>}
```



**DST**

Determines Daylight saving time ([DST](#)) rules used in your region. The following rules may apply: *Europe, USA or Australia*.

**SCPI**

```
SYSTem:TIME:DST {<rules>}
```

**AC mains**

Set the frequency of your AC mains here. This parameter determines the PLC (power line cycle) used for A/D conversion in modules such as MIO168 (default value is 50 Hz).

**SCPI**

```
SYSTem:LFREquency <frequency>
```

**Format**

Sets one of the possible four combination of date, month, year and 12/24 hour that is used for e.g. displaying date and time of the file when accessing it from the [File manager](#).

**SCPI**

```
SYSTem:DATE:FORMAT {<format>}
```

**Enable / Disable NTP**

Internal clock synchronization with [NTP](#) server can be enabled or disabled. When enabled, a *NTP server* entry field will be displayed.

The NTP communication requires active Ethernet connection.

List of most appropriate and available NTP servers is available at [ntppool.org](http://ntppool.org)

**SCPI**

```
SYSTem:COMMUnicatE:ENABLE {<bool>} , NTP
```

**NTP server**

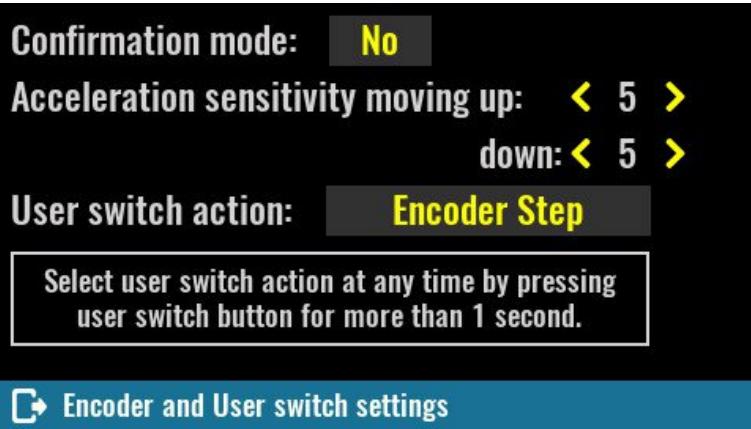
NTP service server network address. The EEZ BB3 will try to establish connection with the selected NTP server on every power up (hard reset), when [\\*RST](#) is issued or once per day.

**SCPI**

```
SYSTem:COMMUnicatE:NTP {<server>}
```

**8.9. Encoder & User sw****Confirmation mode**

The encoder comes with a tactile switch and if this mode is selected then it will be used to confirm the newly changed value. By definition, the mode is not enabled which means that pressing the switch will instead move the "cursor" to the next editable field.

**Acceleration sensitivity moving up/down**

The rotation speed of the encoder is measured and defines in what steps the selected value will change.

The reaction (sensitivity) to increasing the rotation speed in one direction or the other (increasing values or decreasing values) can be set by these parameters.

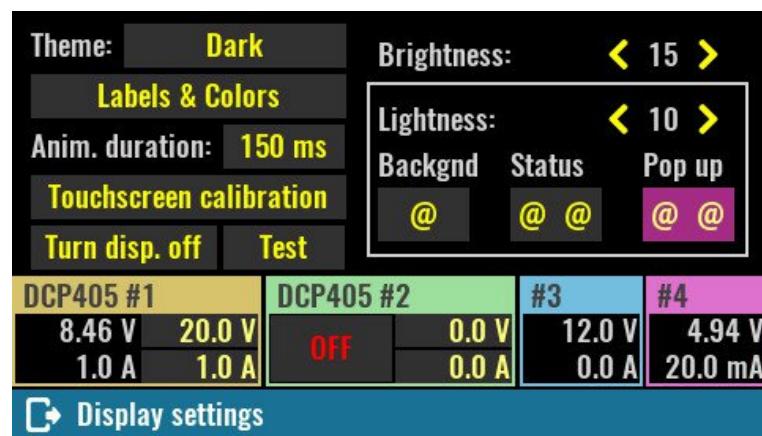
**User switch action**

Opens a menu to select what function to assign to the front panel User SW. See [User SW](#).

## 8.10. Display settings

Display color theme selection, brightness, luminosity, animation duration and status of display can be selected on this page.

An option for entering screen calibration (that is run by default on first start) is also available.



### Theme

The EEZ BB3 comes with three color themes that apply to every page. The default theme is *Dark*.

### Labels & Colors

Displays a new page where you can customize the labels and colors of the currently installed modules.

### Change label

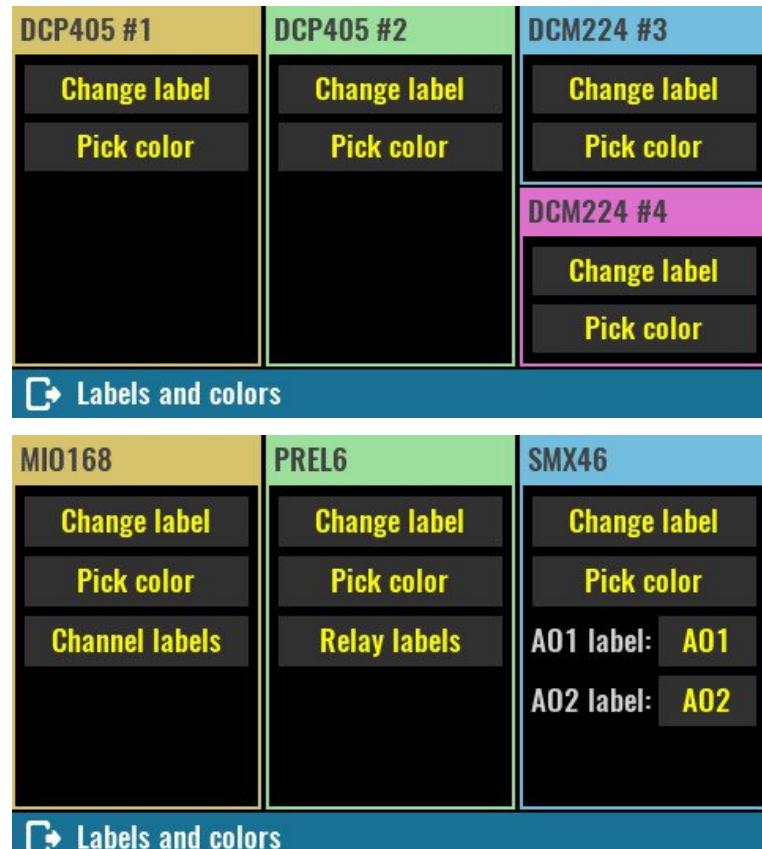
It is possible to change the label for each module. The default label is the name of the module and slot in which it is installed. (e.g. DCP405 #1 if the DCP405 power module is installed in slot 1).

Selecting this option will display the keyboard for entering a new label up to 10 characters long.

If the peripheral module has additional resources (e.g. relays, analog and digital inputs and outputs) that can be labeled, an additional option will appear to access such resources as shown in the figure on the right when for example MIO168, PREL6 and SMX46 modules are installed.

### SCPI

```
SYStem:SLOT:LABel {slot},  
{label}
```



### Pick color

The header and frame module or resources (as in the case of a DCM module that has two) displayed on the screen will get a default color that can be replaced with one from the 24 color palette.

Once the color has been changed revisiting this page will show a new option; *Default* will be available in case you want to return to the original color.

### SCPI

```
SYStem:SLOT:COLor {slot},  
{color}
```



**IMPORTANT:** Changes to Labels & Colors depend on what user profile selected. If it is Profile 0 and you want the changes to be visible during the next power up then make sure that the Autorecall on Power-on option is set (see [User profiles](#)).

### Anim. duration

Define how long the animation effect lasts when transitioning between pages. Default is 250 ms. Set this parameter to 0 ms if you want to disable the effect.

### Brightness

Controls the intensity of the display backlight. The range of the parameter is 1 to 20, where 20 is full intensity and 1 is backlight off. Default value is 10.

#### SCPI

```
DISPLAY:BRIGHTNESS {<value>}
```

### Lightness

This parameter defines the [lightness](#) of displayed colors if they are represented in so called HSL color space (hue, saturation, lightness). By increasing this value all colors will appear "lighter" or washed away. Similarly, when decreased all colors will appear "darker" or more dull. Lightness is not affected when set to 10. However, due to limitation of the LCD display (color representation is reduced to 16-bit) a default value is set to 5 to effectively compensate color difference.

The effect of changing this parameter can be observed by the color of three screen "widgets" (*Background, Status, Pop up*).

### Turn display off

By turning display off user interaction is no longer possible. To reactivate it, you have to touch and hold anywhere on the display for a short time.

#### SCPI

```
DISPLAY OFF
```

### Test

Display test is used to detect defective pixels (aka "dead" pixels).

After the initial message, with each subsequent touch, the entire screen will be painted in primary colors (red, green, blue) following by white. It will be easy to notice if any of the pixels are defective.

Switch colors by screen tap or encoder.

To exit push encoder or user switch.

**Close**

### Touchscreen calibration

Screen calibration is required to give the firmware the necessary information about its geometry, which will increase the accuracy of data entry.

*Initiate this procedure as necessary if you notice that the accuracy of data entry is no longer satisfactory.*

*New screen calibration can be initiated also by touching the screen anywhere and hold for more than 15 seconds.*

Use a stylus or something similar to touch as close to the center of the highlighted crosshair as possible. Keep holding as still as possible until the highlight is removed. Repeat for all crosshairs in sequence.

**⚠ Do not use your finger as a stylus or the result will be imprecise.**



**Start**

#### SCPI

```
CALibration:SCReen:INIT
```

For a successful calibration, you have to use stylus or similar rounded tip object and touch the three dots on the screen as accurately as possible.



After successfully defining the three points, a new page will appear where you can choose between saving calibration data, repeating or canceling the calibration.

On the same screen is possible to test precision by touching the surface beyond the options mentioned.

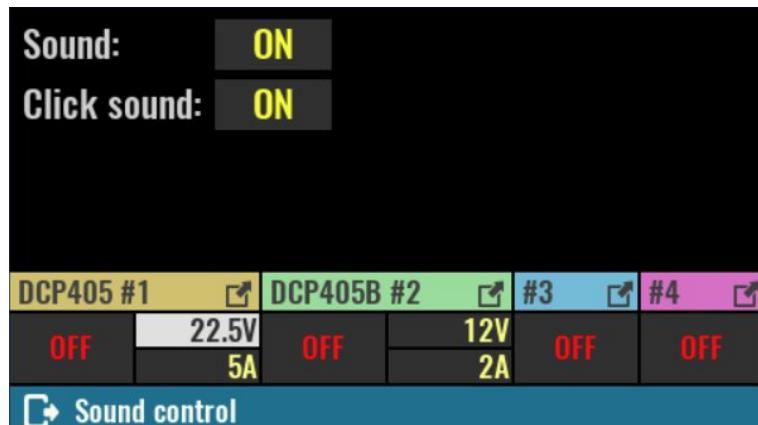


## 8.11. Sound settings

### Sound

When the beeper is enabled, the EEZ BB3 generates audible sound in any of the following situations:

- the power is turned on or off
- when error occurs during local (via display) or remote operation
- self-test is failed or
- any of the protection functions are tripped.



### SCPI

SYStem:BEEPer:STATE {<bool>}

### Click sound

Select this option to enable or disable generation of an audible “click” sound as confirmation of selected item on the screen.

### SCPI

SYStem:BEEPer:KEY:STATE {<bool>}

## 8.12. System information

The page summarized various system information and contains a list of all recognized modules.

### Total On time

Total active time of the EEZ BB3 (i.e. MCU module). Resolution is 1 minute and this information is stored every 10 minutes in non-volatile memory. Therefore it's possible that up to 10 minutes is lost after restart caused by a power outage or ungraceful power shutdown.

<b>Total On time:</b>	<b>25d 17h 48m</b>	<b>Firmware:</b>	<b>1.6</b>
<b>Last On time:</b>	<b>1m</b>	<b>CPU module:</b>	<b>STM32 R3B3</b>
<b>Temp. AUX:</b>	<b>30 °C</b>	<b>Slot #1:</b>	<b>DCP405 R3B3</b>
<b>Fan speed:</b>	<b>0 rpm</b>	<b>Slot #2:</b>	<b>DCP405 R3B2</b>
<b>Battery:</b>	<b>3.05 V</b>	<b>Slot #3:</b>	<b>DCM224 R3B1</b>
<b>SD card:</b>	<b>Present</b>		
<b>Host name:</b>	<b>EEZ-BB3-1</b>		
<b>Serial No.:</b>	<b>003D001E3338510738323535</b>		

System information

### SCPI

```
SYSTem:CPU:INFormation:ONTime:TOTal?
```

### Last On time

This query returns time passed after last activation of the EEZ BB3. Resolution is 1 minute and this information is stored every 10 minutes in non-volatile memory. Therefore it's possible that up to 10 minutes is lost after restart caused by a power outage or system reset.

### SCPI

```
SYSTem:CPU:INFormation:ONTime:LAST?
```

### Temp. AUX

The "system" temperature is measured on the AUX-PS module and can be used for over-temperature protection (OTP). See [System temp & fan](#).

### SCPI

```
MEASure:TEMPerature? AUX
```

### Fan speed

Current fan speed indicator stated in rpm (revolutions per seconds). Any diagnosed fan faults will be also displayed here. See also [System temp & fan](#).

### SCPI

```
DIAGnostic:FAN?
```

### Battery

Current voltage of the RTC (Real-time-clock) lithium 3 V coin battery (CR2032, Ø20 x 3.2 mm).

### SCPI

```
SYSTem:MEASure?
```

### SD card

Information about presence of mass-storage media (i.e. micro SD card).

### SCPI

```
MMemory:INFormation?
```

### Host name

Local area network (LAN) connection host name.

### SCPI

```
SYSTem:COMMUnicatE:ETHernet:PORT {<number>}
```

### Serial No

The EEZ BB3 (MCU module) serial number.

**SCPI**

\* IDN?

**Firmware**

The EEZ BB3 (MCU module) firmware version number.

**CPU module**

The model name and version of the MCU module.

*IMPORTANT: Here you can choose for which version of the MCU module the system firmware will be activated. In case of choosing the wrong version of the module BB3 will not be functional which can be manifested with various problems in communication with peripheral modules. If you are not sure whether the correct version is selected, you can find the version label on the MCU module itself, e.g. r2B4 or r3B3.*

**SCPI**

SYSTem:CPU:MODel?

**Slot #<n>:**

The model name and version of all detected modules.

**SCPI**

SYSTem:SLOT:MODel? {<slot>}

## 9. System functions

Functions that are not related to a specific module or channel are called system functions and most are accessible from the status bar of the home page. In this section the following functions will be discussed:

- [Event viewer](#)
- [Tracking](#)
- [File manager](#)
- [User profiles](#)
- [Power outputs coupling](#)
- [Display lock / unlock](#)
- [Power / Reset control](#)
- [User SW](#)

### 9.1. Event viewer

During operation, EEZ BB3 will record the time of events of certain user activities, as well as events that require user attention (i.e. warnings and errors).

Events are saved in an event log that can be accessed by tapping the event view icon. The number of events stored in the event log is limited only by available space on the SD Card.

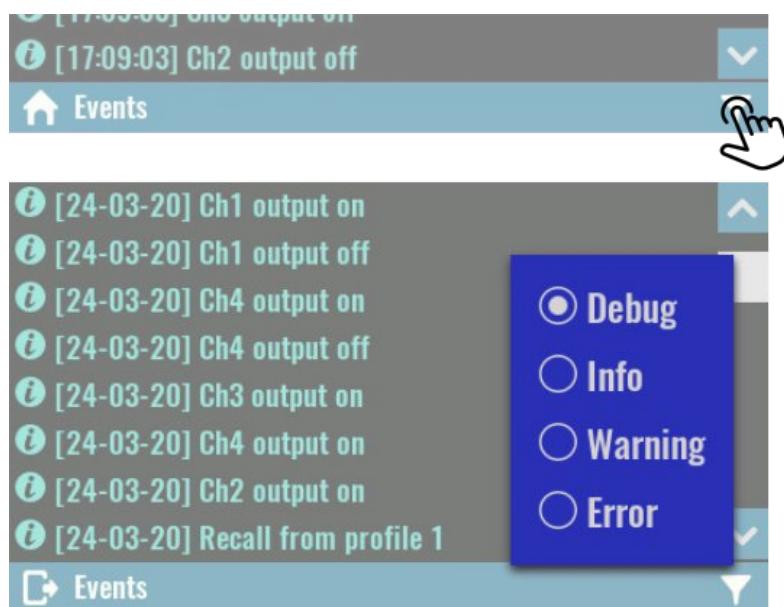
The appearance of a new event that requires user attention and occurred after the last review of the event log will result in a change in the color of the icon.



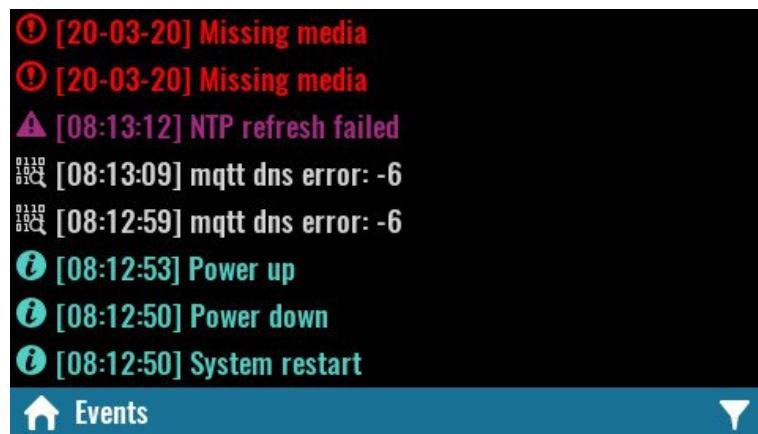
If SD card is not present or dysfunctional only the latest 50 events will be accessible via Event viewer. That events will not be written to non-volatile memory, and will be lost after the restart.

#### 9.1.1. Events filtering

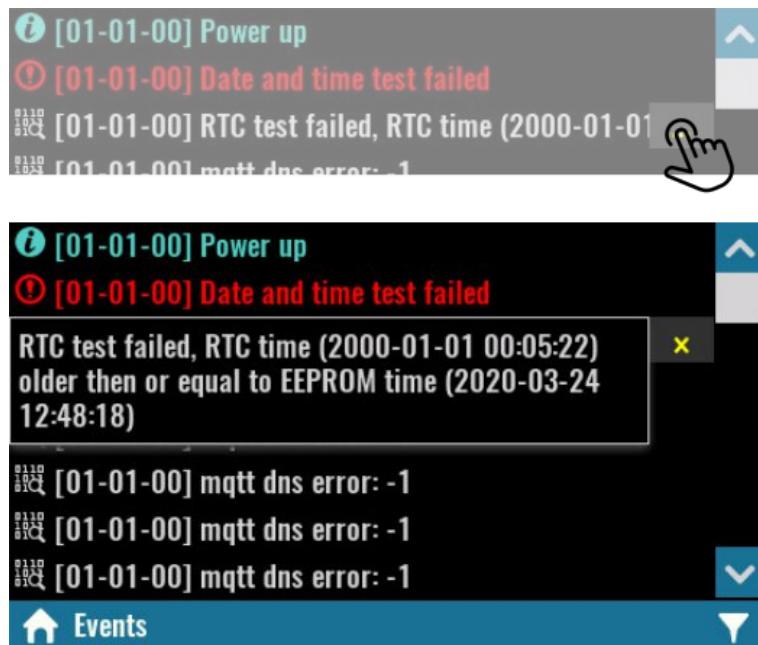
Four types of logged events exists and their appearance in the event viewer depends of the selected filter. The *Debug* filter represents the most comprehensive view that contains all type of events will *Error* filter allows appearance of error messages.



Events types identification is simplified by using different icons and colors.



If the content of the message cannot be displayed in one line due to its size, an expansion option will appear.



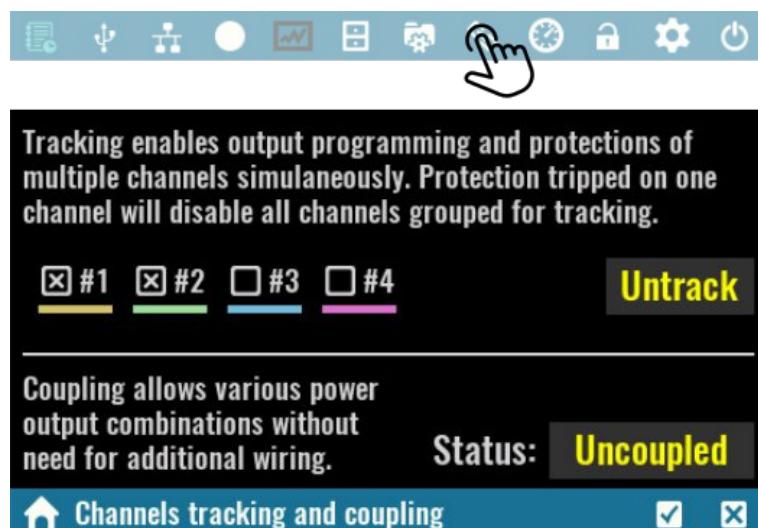
## 9.2. Tracking

Tracking is a convenient way to manage two or more channels simultaneously, i.e. there is no “master-slave” configuration, when changes can only be made via the master channel.

Only one tracking group can be set up. Channels that are in the tracking group will be able to define the following parameters at the same time:

- Output set current and limit
- Output set voltage and limit
- Output power limit
- OCP, OVP, OTP and OPP parameters (activity, delay, status)
- Tripped protection clear
- Trigger type, source and delay
- Output list definitions (dwell, current, voltage)

When channels are in a tracking group, some of their options will be disabled (e.g. calibration). The *Untrack* option can be used for clear selections and will become visible only if two or more channels are selected.



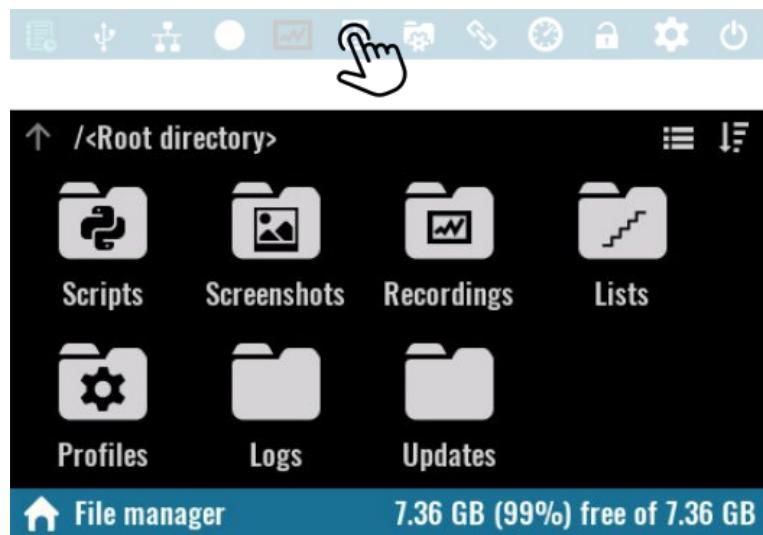
SCPI

OUTPut:TRACK[:STATE] (@&lt;chanlist&gt;)

### 9.3. File manager

File manager lets you access and work with files on a SD card. The working data is organized into multiple folders (directories) as follows:

- *Scripts* – contains MicroPythons scripts (.py files) and related resources (.res files )
- *Screenshots* – contains all screenshots taken (see [User SW](#))
- *Recordings* – data logging function use this folder to store recorded data



- *Lists* – contains program lists that can be executed on the selected channel
- *Profiles* – location for user profiles
- *Logs* – recorded events that are accessible via [Event viewer](#).
- *Updates* – contains firmware binary images that can will can be selected from module's firmware download section.

Folders and files in File manager can be displayed as list.

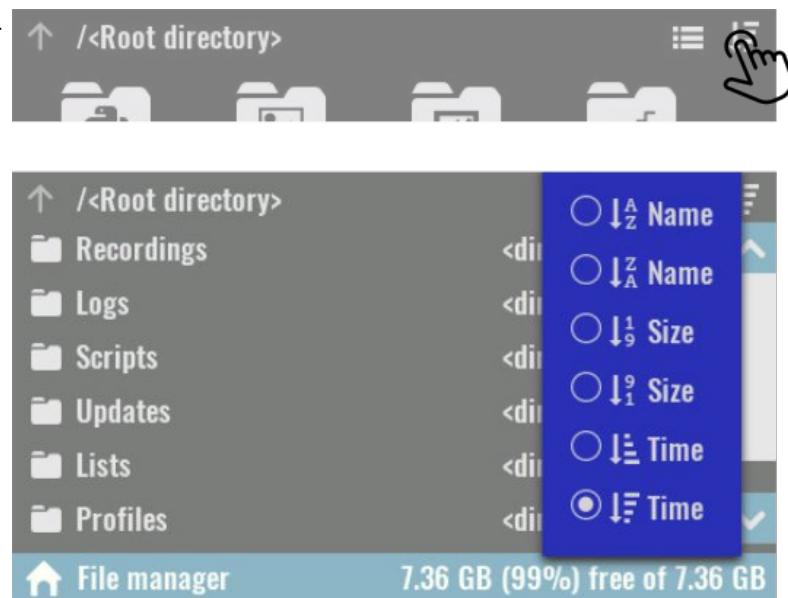


↑ /<Root directory>			
Logs	<dir>	17:36:36	▲
Scripts	<dir>	17:16:30	
Updates	<dir>	02-01-00	
Lists	<dir>	02-01-00	
Profiles	<dir>	02-01-00	
Recordings	<dir>	02-01-00	▼
File manager	7.36 GB (99%) free of 7.36 GB		

Files and folders displayed can be sorted by name, size and time in descending order and ascending order. Selecting the *Sort* icon will display a menu through which you can select the sorting method.

**SCPI**

```
MMEMemory:CATalog?
MMEMemory:CATalog:LENgth?
MMEMemory:INFOrmation?
MMEMemory:TIME? {<filename>}
```



Clicking on a file name will open a menu whose options will be enabled or disabled depending on the file type and whether a computer connection is established.

**Open**

Displays an image in .jpg format, or opens a viewer for logged data in .dlog format

**Upload**

Initiates file transfer to the computer. If the file is large, the progress percentage will be displayed.

/Recordings			
[W]	25_03_20-12_32_18.dlog	47.11 KB	12:33:18
[W]	24_03_20-23_1	11 KB	24-03-20
[W]	zener20v.dlog	Bytes	24-03-20
[W]	24_03_20-14_0	11 KB	24-03-20
[W]	24_03_20-14_0	11 KB	24-03-20
[W]	zener62v.dlog	37 KB	24-03-20

File manager      13 items

**SCPI**

```
MMEMemory:UPLoad? {<filename>}
```

**Rename**

Rename a file or folder.

**SCPI**

```
MMEMemory:MOVE {<source>}, {<destination>}
```

**Delete**

Deleting a file or folder.

**SCPI**

```
MMEMemory:DElete {<filename>}
```

## 9.4. Power outputs coupling

Coupling provides a safe and convenient way to combine power outputs to increase capacity and offer output configurations.

Four power relays are used to combine the power outputs, avoiding the possibility of poor connections or accidental loss of connection, which could have detrimental effects on the connected load.

Tracking enables output programming and protections of multiple channels simultaneously. Protection tripped on one channel will disable all channels grouped for tracking.

#1    #2    #3    #4

Coupling allows various power output combinations without need for additional wiring.

Status: **Uncoupled**

 Channels tracking and coupling



Furthermore, in the case of coupling into series or parallel, when doubling the output values, this will be taken into account, thus avoiding errors in the interpretation of the set and measured values. By default all outputs are independent, i.e. separated from each other and “floating” or isolated in reference to MCU ground or PE (Protection Earth) potential.

*Please note that regardless of the set coupling mode all channel outputs will remain isolated from the MCU ground and PE.*

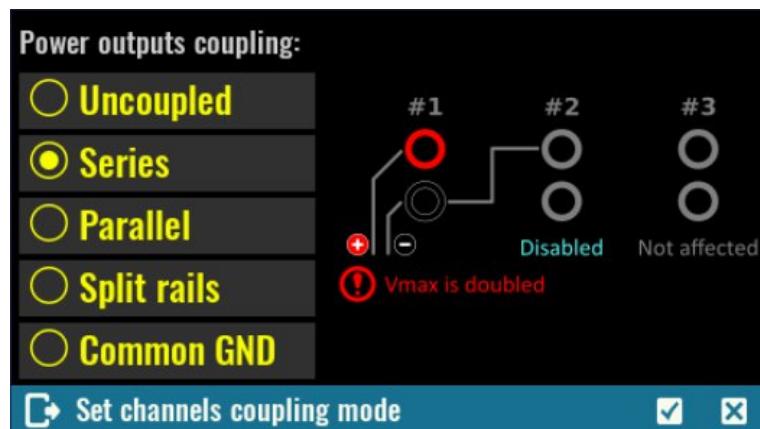
**SCPI**

INSTRument:COUPLE:TRACKing NONE

**Series**

Coupling in series is only possible between DCP405 modules on the first two slots. When coupled, the output connectors of the module in slot 2 will be disabled, and the output voltage on the connectors of the module in slot 1 will be able to be set to twice the value (i.e. 80 V for DCP405).

The maximum output current will remain unchanged (5 A).



Access to the module in slot 2 will be disabled on display. All set and measured output voltage and power values shown on the display for the module in slot 1 will be doubled.

*Please note that according to many standards, voltage of 50 V and above are considered to be hazardous regardless of additional factors such as air humidity and temperature, skin moisture, etc. Take all necessary precautions when the set output voltage exceeds this value.*

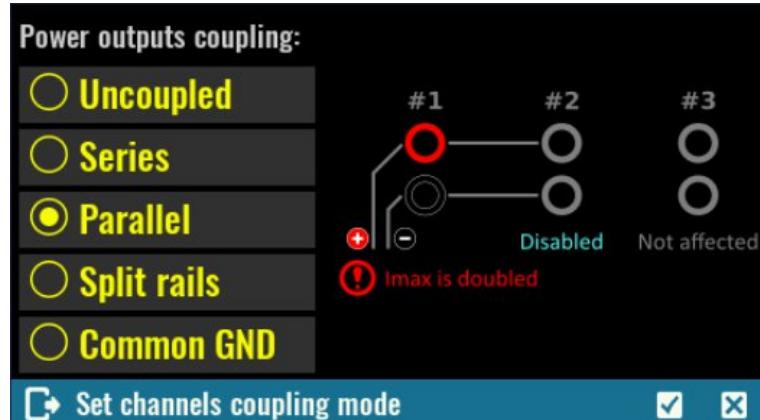
**SCPI**

INSTRUMENT:COUPLE:TRACKing SERies

**Parallel**

Coupling in parallel is only possible between DCP405 modules on the first two slots. When coupled, the output connectors of the module in slot 2 will be disabled, and the output current on the connectors of the module in slot 1 will be doubled (i.e. 10 A for DCP405).

The maximum output voltage will remain unchanged (40 V).



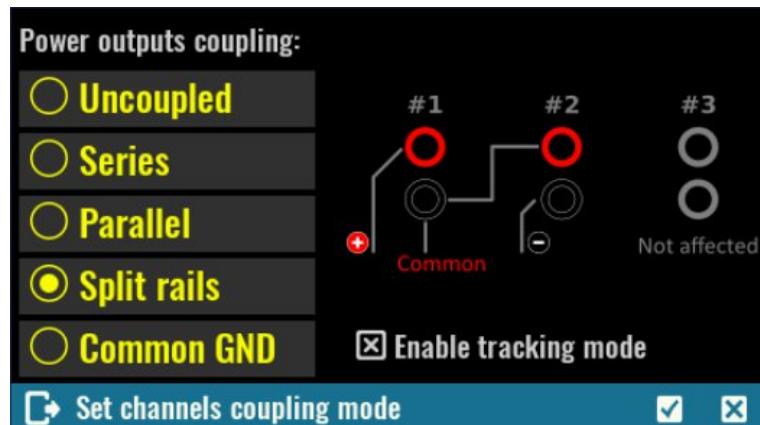
Access to the module in slot 2 will also be disabled on display. The set and measured output current and power values shown on the display for the module in slot 1 will be doubled.

**SCPI**

INSTRUMENT:COUPLE:TRACKing PARallel

**Split rails**

This type of coupling is similar to serial coupling except the output of both channels is still active and the coupling between the first and second channels is used as a common ground. In this way, a symmetrical output is obtained where  $V_{out+}$  of the first channel represents a positive rail and  $V_{out-}$  of the second negative rail.



Channels can still be individually controlled, but since this configuration often requires the same output values (e.g. for powering operational amplifiers) a default is offered that places these two channels into a tracking group.

**SCPI**

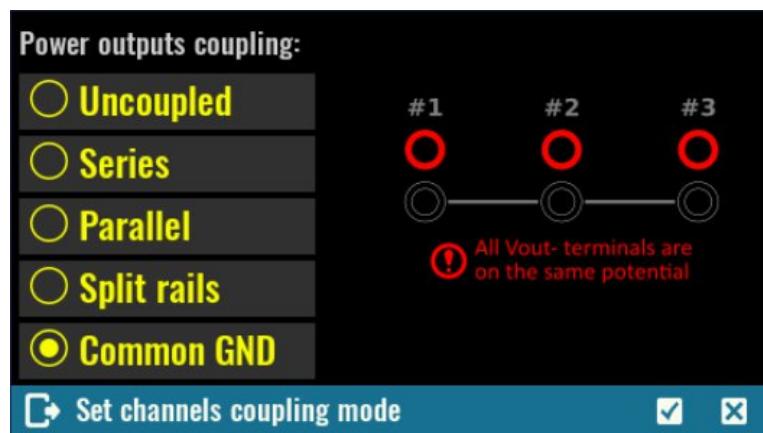
INSTRUMENT:COUPLE:TRACKing SRAil

### Common GND

This is the simplest way of coupling the output and results in the connection of Vout- all channels on all slots to a common potential.

As mentioned in the beginning, the Vout- coupled output potential will still not be at the MCU ground or PE potential.

*This coupling is not required for Vout-connections between two channels on the DCM220 because they are already internally connected.*



### SCPI

INSTRUMENT:COUPLe:TRACKing CGND

### 9.5. User profiles

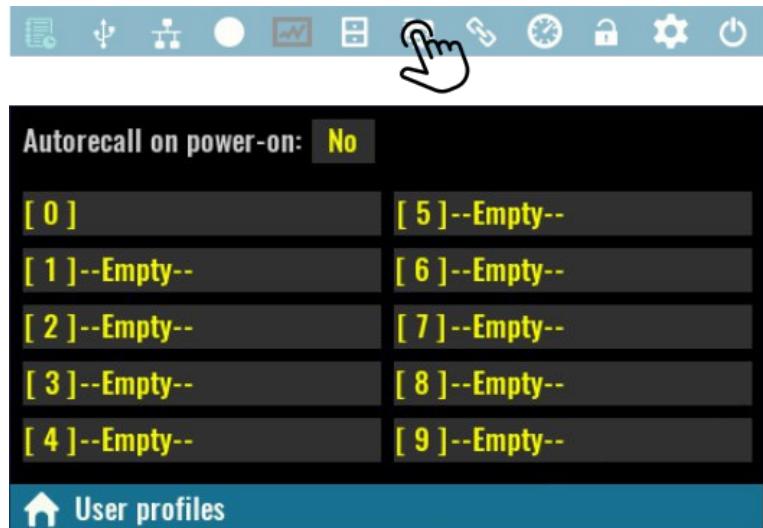
User profiles are used to store and recall system parameters and parameters of installed modules.

There are 10 profiles available that are displayed by number and name.

The profile 0 has a special status, i.e. it contains the current parameter states. Its name cannot be defined, but when a recall is made from a profile, it will receive its name.

### SCPI

MEMORY:STATE:CATalog?



### Autorecall on power-on

Defines whether or not to load parameter states from the selected user profile at power up.

When selected, a *Power-on profile* number is also displayed, that is set by default to user profile 0.

If this option is off, the power up parameters will be set to initial values (i.e. "factory settings").

*Parameters stored in an empty user profile cannot be selected as an auto-recall profile.*



### SCPI

MEMORY:STATE:RECall:AUTO ON

**Save**

Saving the parameter state can be done in an empty user profile or one that has already been used.

The following system states and parameters will be stored in non-volatile memory at the position of the selected user profile:

- Calibration status
- Output enable state
- Output track state
- Channel coupling state
- Remote sense state
- Output set voltage, voltage limit, and voltage step
- OVP status, and OVP delay
- Output set current, current limit, and current step
- OCP status and OCP delay
- Output power limit, OPP level, OPP status and OPP delay
- OTP level, OTP status and OTP delay
- Power on state

**SCPI**

\*SAV {<profile>}

When saving a user profile, it is necessary to define its name (i.e. *Remark* as shown on next picture). In case of saving to an empty user profile, the name will be offered as a combination of *Saved at* and the current date and time. If saved to previously used profiles, an existing name will be offered.

**SCPI**

MEMORY:STATE:NAME  
{<profile>}, {<name>}

**Use as Power-up profile**

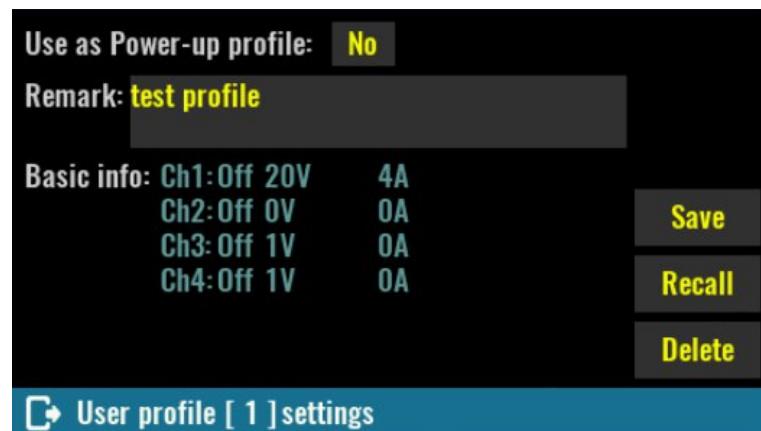
Specifies whether the initial states will be loaded or not from the profile if the *Autorecall on power-on* option is selected.

**SCPI**

MEMORY:STATE:RECall:SElect  
{<profile>}

**Recall**

Use this option to load immediately the parameter status from user profile.

**SCPI**

\*RCL {<profile>}

**Delete**

Empty user profile and reset its name (remark) to *--Empty--*.

**SCPI**

MEMORY:STATE:DElete {<profile>}

## 9.6. Display lock / unlock

Tap the *lock* icon to lock display. When locked the icon changes color and lock state is saved in non-volatile memory. Therefore, the front panel remains locked even after power is cycled.

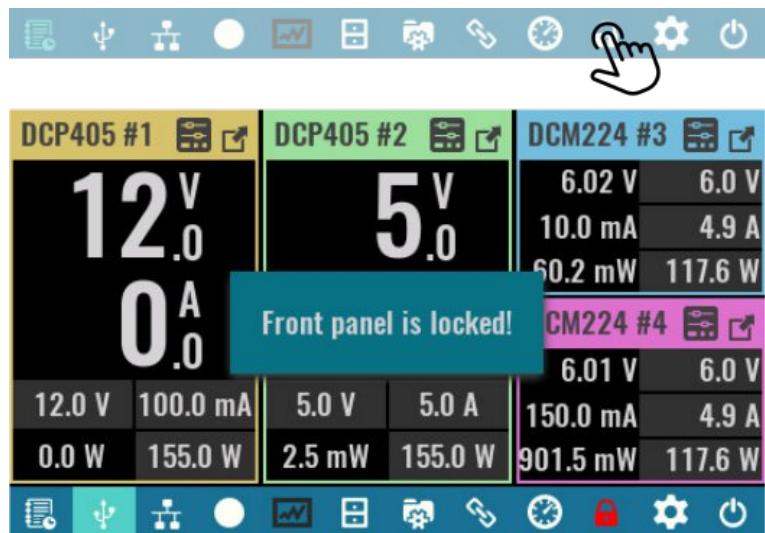
### SCPI

SYSTem:KLOCK

Unlocking is performed by tap on the lock icon, and may require system password for unlock if it is defined.

### SCPI

SYSTem:LOCal



## 9.7. Power / reset control

### Default (\*RST)

Initiate so-called soft reset procedure. All outputs are set to OFF, and voltage and current are programmed to 0.

### SCPI

\*RST

### Standby

Results in disconnecting of AC power for all installed peripheral modules.



Standby mode keeps the MCU module powered that is indicated with the *Standby* indicator on the front panel. The power up can be initiated by tap and hold action anywhere on the screen.

### SCPI

SYSTem:POWer OFF

### Restart

This action is equivalent to recycling input power. On power up all modules will be initialized, self test will be performed and initial values will be set depending on the selected user profile and *Autorecall on power-on* status as defined in [User profiles](#).

### SCPI

SYSTem:RESet

### Shutdown

Initiates a graceful shutdown that ensures that the latest information are stored in non-volatile memory. Wait until a message appears on the screen that EEZ BB3 can be safely turned off.

### Display off

While the display is off, future user interaction will be disabled until it is turned on again. Turn the display back on by tap and hold action anywhere on the screen.

### SCPI

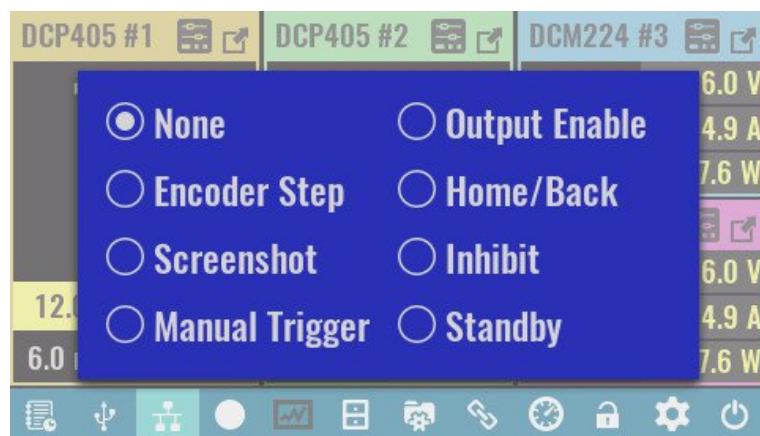
DISPLAY OFF

## 9.8. User SW

### None

If selected, no function is assigned to the user switch. A short push of the user switch will open the menu showing available options. Press the button once again if no changes is needed.

Once an option is assigned to the user switch it will be necessary to press and hold the button for about 1 second for the menu to reappear.



### Encoder Step

Cycle through the sensitivity of the encoder to change the selected output value (voltage, current or power). Initially, the encoder is set to *Auto* mode when the change will depend on the speed at which it will rotate. Its turn sensitivity in one direction or the other can be changed in the [Encoder settings](#).

### Screenshot

Take screenshots of current screen content and save to SD card in the *screenshots* folder. The image will be saved in jpeg format and its name formatted as *yyyy\_mm\_dd-hh\_mm\_ss.jpg*

### Manual Trigger

Allows the user switch to be used to initiate the trigger system (see [General trigger settings](#)).

### Output Enable

When two or more channels are in tracking mode, this option allows their outputs to be switched on and off simultaneously

### Home/Back

When this option is selected, user switch can be used as an option to return to the previous or main page. It can also be used to exit maximized channel view.

### Inhibit

This option is selected by default, when user switch can be used to enter inhibit mode when all active outputs will be temporarily disabled that is indicated with INH text. To exit inhibit mode, you will need to press the button once again.



### Standby

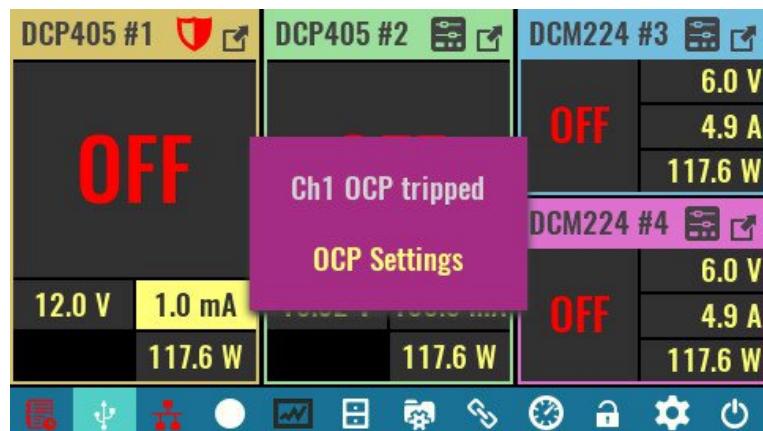
If EEZ BB3 is turned on and this option is selected, the first time a button is pressed, it goes into standby mode. The next time the button is pressed, it exits standby mode.

## 10. Channel protections

Power modules are designed with multiple protection functions to avoid various unexpected conditions, for example output values exceeding the set limits which (if unchecked) could cause permanent damage to a connected device or the power module itself.

Clearing an activated protection mode, disabling protection modes or setting protection mode parameters may be accessed via *Protections* section on the [Channel protection and settings](#) page.

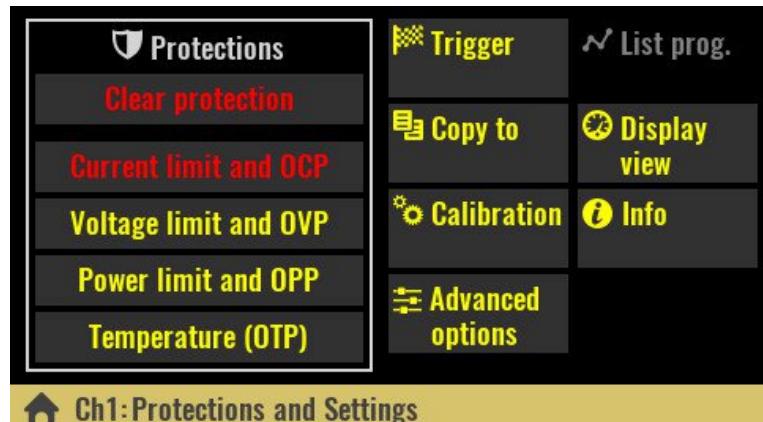
If any of enabled protections tripped, as for example over-current protection, popup message will appear and channel settings icon will be changed to shield icon colored in red.



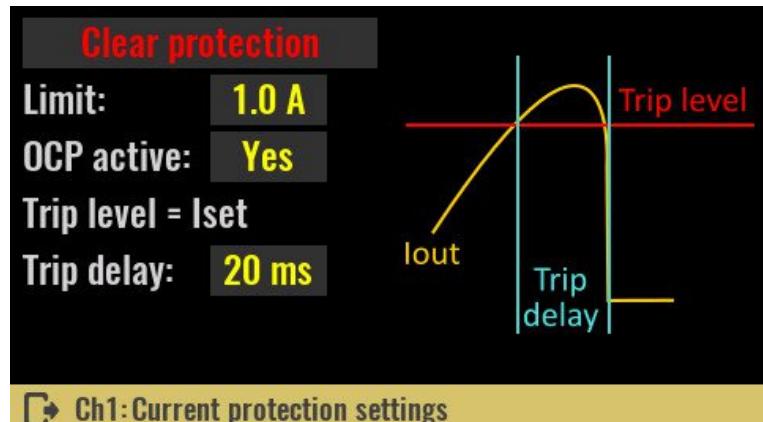
### Clear protection

Use this option on *Protections and Settings* page to clear all tripped protections so that channel outputs can be re-enabled as needed.

The status of active protections is not affected by this option.



The same option could be used from tripped protection settings page to clear related protection trip.



### SCPI

```
OUTPut:PROTection:CLEar
CURRENT:PROTection:STATE OFF
POWER:PROTection:STATE OFF
VOLTage:PROTection:STATE OFF
SYSTem:TEMPerature:PROTection:STATE OFF, <sensor>
```

## 10.1. Current limit and OCP

### Limit

Defines the max. value of the output current that the user will be able to set. The initial value is max. current the module can source (e.g. 5 A for the DCP405 module).

*Changing the limit will also affect graphical views (vertical bar, horizontal bar and YT views), if the display settings of the channel is set to Limit (see [Display view](#) section). For example, if the limit is set to 1 A, this value will become 100% of the scale instead of only 20% when the limit is at 5 A.*

### SCPI

```
CURRent:LIMit {<current>}
```

### OCP active

The OCP (Over-Current Protection) is software protection that will activate when drawn current reaches the set value of the allowed output current. A protection mode trip will be recorded in the event log and the channel output will be turned off.

Use this option to turn OCP on or off. If OCP is off, if the output current reaches the set value ( $I_{set}$ ) the channel will switch from CV to CC mode without turning off the channel output.

### SCPI

```
CURRent:PROTection:STATE {<bool>}
```

### Trip delay

If the OCP is enabled and the output current reaches the set value ( $I_{set}$ ), an OCP trip will occur at the earliest after the set delay time expires.

### SCPI

```
CURRent:PROTection:DELay {<time>}
```

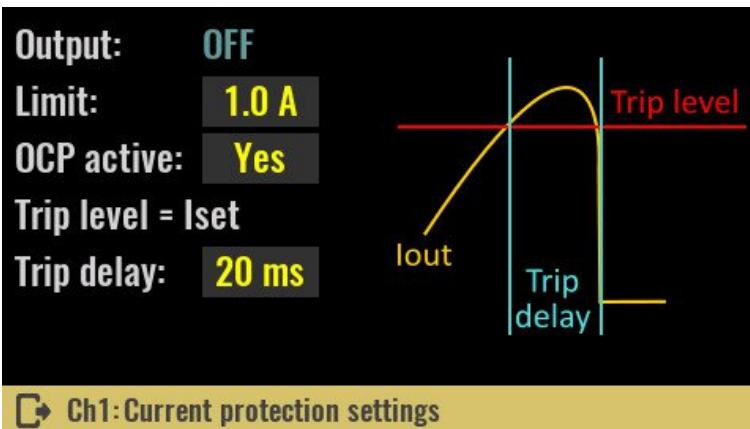
## 10.2. Voltage limit and OVP

### Limit

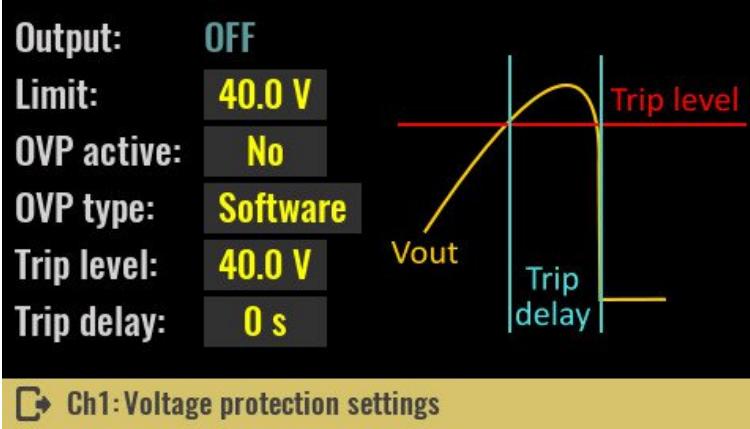
Defines the max. value of the output voltage that the user will be able to set. The initial value is max. voltage the module can source (e.g. 40 V for the DCP405 module).

### SCPI

```
VOLTage:LIMit {<voltage>}
```



Ch1: Current protection settings



Ch1: Voltage protection settings

### OVP active

Turns OVP on or off. If OVP is off (which is default), when the output voltage reaches the set value ( $U_{set}$ ) the channel will enter the CV mode (which is normal behavior if the connected load does not draw more current than the maximum set). Therefore, when activating OVP without load or with connected load that draws less than the set current, the channel will enter CV mode and the OVP will immediately trip. To avoid this the channel should be in CC mode prior to the output being turned on.

### SCPI

```
VOLTage:PROTection:STATE {<bool>}
```

**OVP type**

Selection between software and hardware OVP is only possible with DCP405 modules.

**SCPI**

```
VOLTage:PROTection:TYPE SW
```

**Trip level**

The trip level value may be greater than or equal to the set voltage ( $V_{set}$ ) and determines the value of the output voltage at which the OVP will trip. In case the set value of the output voltage increases, the trip level will automatically increase.

For example, if the trip level was set at 10 V while the output voltage was also at 10 V and its value increased to 12 V, the trip level value would also increase to 12 V. But, if the voltage drops to say 5 V, the trip level will remain unchanged.

*Max. trip value is 0.05 V greater than max. output voltage (i.e. 40.05 V for DCP405 module).*

**SCPI**

```
VOLTage:PROTection {<voltage>}
```

**Trip delay**

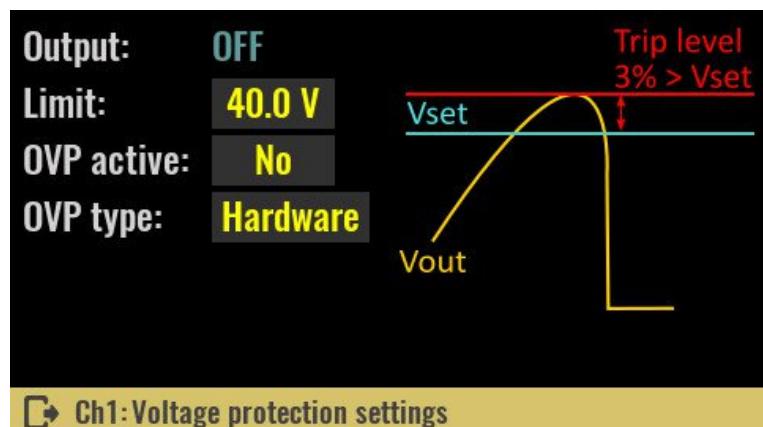
If the OVP is enabled and the output voltage reaches the set *trip level*, an OVP trip will occur at the earliest after the set delay time expires.

**SCPI**

```
VOLTage:PROTection:DElay {<time>}
```

**10.3. Hardware OVP**

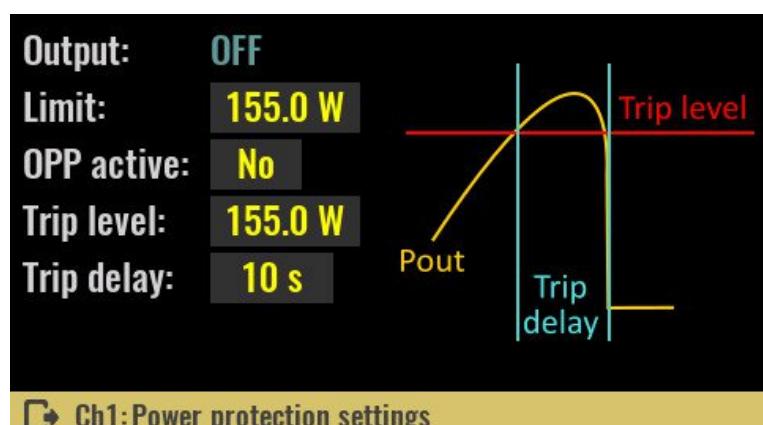
The DCP405 module also contains hardware OVP that works on a slightly different principle. Unlike software OVP where it is possible to adjust the trip level, with this protection the trip level is automatically set at approximately 3% above the set voltage output. The response rate (ie trip delay) also cannot be changed, but the OVP circuit will respond as quickly as possible.

**SCPI**

```
VOLTage:PROTection:TYPE HW
```

**10.4. Power limit and OPP****Limit**

Defines the max. value of the output power that the user will be able to set. The initial value is max. power the module can source (e.g. 155 W for the DCP405 module).

**OPP active**

The OPP (Over-Power Protection) is software protection that will activate when output power reaches the programmed value. Protection trip will be recorded in the event log and the channel output will be turned off.

**SCPI**

```
POWer:PROTection:STATE {<bool>}
```

### Trip level

The trip level value may be less than or equal to the set *limit* and determines the value of the output power at which the OPP will trip. When the power *limit* is set to a lower value, the trip level will also decrease.

#### SCPI

```
POWer:PROTection {<power>}
```

### Trip delay

If the OPP is enabled and the output power reaches the set *trip level*, an OPP trip will occur at the earliest after the set delay time expires.

#### SCPI

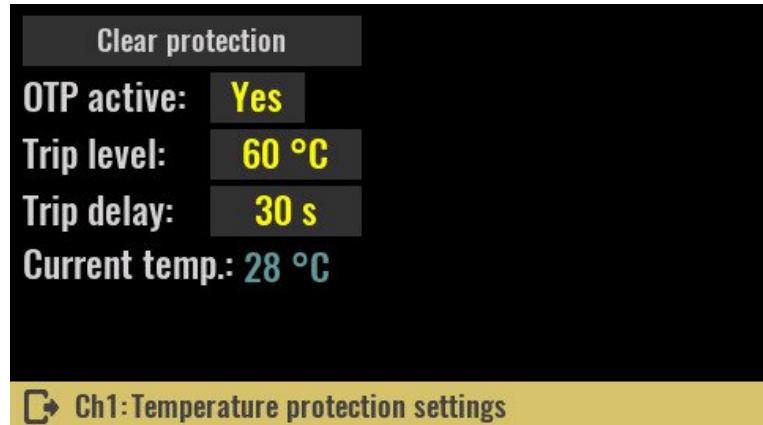
```
POWer:PROTection:DELay {<time>}
```

## 10.5. Temperature (OTP)

### OTP active

The OTP (Over-Temperature Protection) is software protection that will activate when the temperature measured on the channel's temperature sensor is equal to or greater than the set *trip level* value for a duration of *trip delay*.

Protection trip will be recorded in the event log and the channel output will be turned off.



Dual channel modules like the DCM220 have two temperature sensors, one for each channel.

#### SCPI

```
SYSTem:TEMPerature:PROTection:STATE {<bool>} [, <sensor>]
```

### Trip level

The trip level value may be less than or equal to the set *limit* and determines the value of the measured temperature at which the OTP will trip.

#### SCPI

```
SYSTem:TEMPerature:PROTection {<temperature>}, <sensor>
```

### Trip delay

If the OTP is enabled and the temperature reaches the set *trip level*, an OTP trip will occur at the earliest after the set delay time expires.

#### SCPI

```
SYSTem:TEMPerature:PROTection:DELay {<delay>}, <sensor>
```

## 11. Special channel functions

The channel [Protections and settings](#) page also provides access to various additional features as follows:

- [Trigger](#)
- [List programming](#)
- [Display view](#)
- Calibration (see [Power modules calibration](#))
- [Info](#)
- [Advanced options \(DCP405 only\)](#)

### 11.1. Channel trigger settings

#### General trigger settings

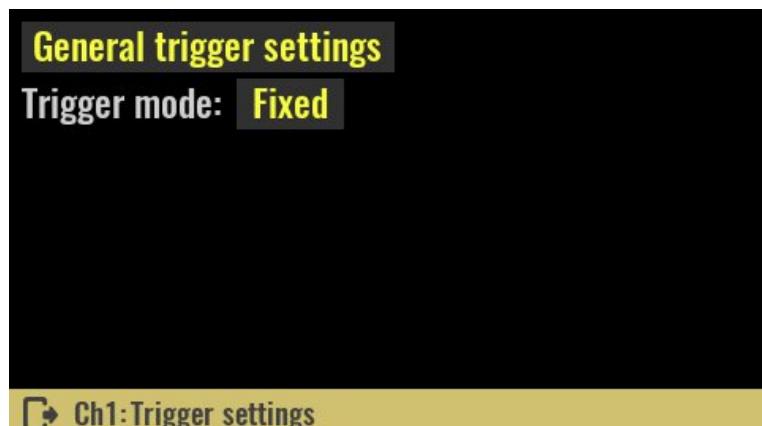
A shortcut to [General trigger settings](#) page where you can adjust general settings related to the trigger system, like trigger source, delay or whether the trigger system is continuously initiated or not.

##### Fixed

Channel output state and values will remain at the immediate value, i.e. the channel is not affected by the triggering system.

##### SCPI

```
CURRent:MODE FIX  
VOLTage:MODE FIX
```

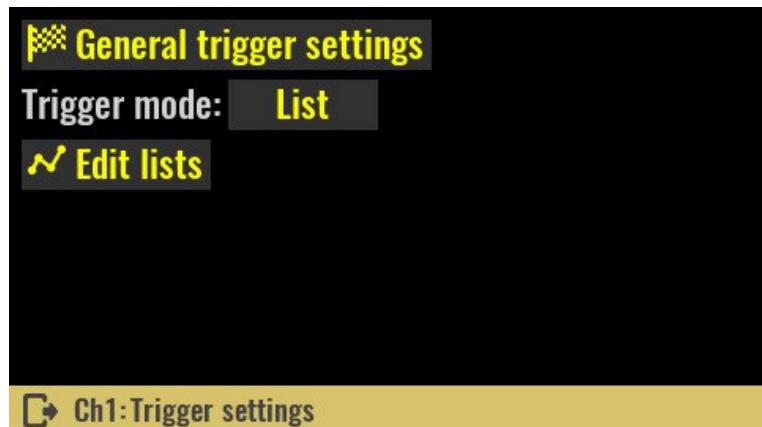


##### List

Channel output state and values will change according to the defined sequence of output voltages and currents of specified duration (see [List programming](#)).

##### SCPI

```
CURRent:MODE LIST  
VOLTage:MODE LIST
```



When trigger list is active, channel's output voltage and current set options will be replaced with the *list* icon that leads directly to the channel trigger settings page (i.e. the page shown above).



**Step**

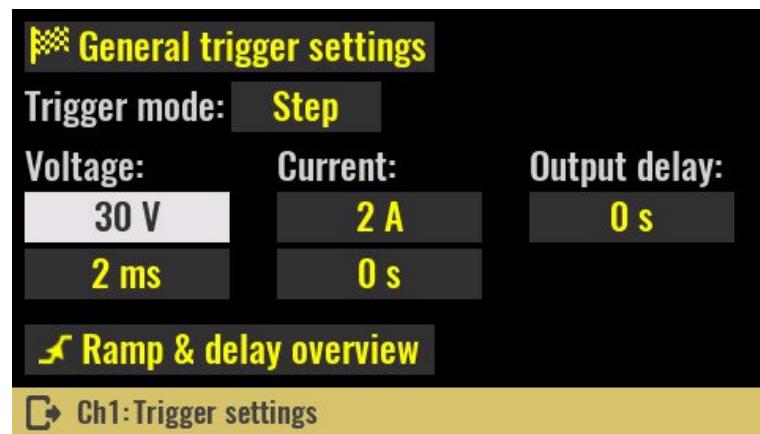
Channel output state and values goes to the step level when a trigger occurs.

**SCPI**

```
CURRENT:MODE STEP
VOLTAGE:MODE STEP
```

**Voltage:**

Programmed new value to which the output voltage will be set when the trigger system is initiated.

**SCPI**

```
VOLTage:TRIGgered {<voltage>}
```

The time value – set in the cell below the voltage value - defines how long the output voltage will take to reach the set value. If 0 is entered, the output voltage will be set in the shortest possible time (constrained by the current limit specified - i.e. if the current limit is set to a low value, it will take longer to reach the output voltage due to charging of the capacitor on the power output and input capacitor of the connected load). For a controlled output voltage ramp, values from 2 ms to 10 s can be used.

**SCPI**

```
VOLTage:RAMP:DURation {<duration>}
```

**Current:**

Programmed new value to which the output current will be set when the trigger system is initiated.

**SCPI**

```
CURRent:TRIGgered {<current>}
```

Value in the cell below defines how much time will take that output current limit reach set value. When set to 0, the output current limit will try to set in the shortest possible time. For a controlled output current limit ramp, values from 2 ms to 10 s can be used.

**SCPI**

```
CURRent:RAMP:DURation {<duration>}
```

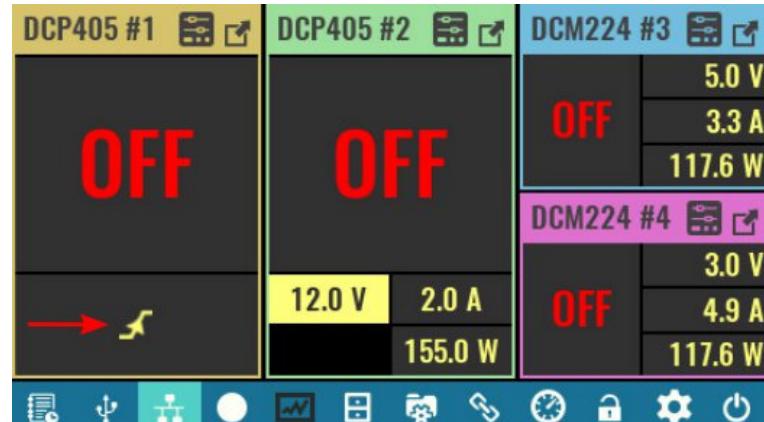
**Output delay:**

Output turn on delay in seconds after the trigger system is initiated. Delays from 2 ms to 10 ms can be specified.

**Ramp & delay overview**

A shortcut to [Ramp & delay](#) page where output voltage and current ramp and delay parameters can be viewed and edited on the single place.

When trigger step is active, channel's output voltage and current set options will be replaced with the **step** icon that leads directly to the channel trigger settings page (i.e. page shown above).



## 11.2. List programming

### Overview

The list programming editor makes it easy to set multiple output voltages and currents that will be set in sequence.

A program list can have up to 255 steps with equal or varying durations or time intervals (*dwells*).

Entering a new or editing an existing value will require a tap on the cell when the numeric keypad will open. A preview of the entered sequences will be displayed at the top of the page.

#	Dwell	Voltage	Current	
1	—	—	—	
2	—	—	—	
3	—	—	—	
4	—	—	—	
5	—	—	—	

List count:  
1  
On list stop:  
Output OFF

Navigation icons: back, forward, search, list, list next, checkmark, close.

If the values of voltage and current defined in the list exceed the currently active limits, the list will be interrupted at the first step with too high value. For example, if the output voltage is limited to 10 V, and the list has the following steps: 5, 15, 20, 30 V, the list execution will be interrupted when the 15 V step is reached.

The number of steps in the program list is 255. The list editor displays 5 steps at a time, and when at least one cell in each of first five steps is filled, a navigation arrow will be enabled in the status bar to move to the next five steps.

### List import and export

An already defined list can be imported or exported. The storage location is the Lists folder on the SD card.

Selecting the file manager icon will open a menu to export or import the list.

#	Dwell	Voltage	Current	
1	—	—	—	
2	—	—	—	
3	—	—	—	
4	—	—	—	
5	—	—	—	

List count:  
1  
On list stop:  
Output OFF

Import Export

Navigation icons: back, forward, search, list, list next, checkmark, close.



If you import a list, the contents of the Lists folder will be displayed.

/Lists			
/	dcm220_1.list	55 Bytes	07:36:20
/	new2.list	69 Bytes	23-03-20
/	test.list	57 Bytes	20-03-20
/	sinewave.list	1.26 KB	20-03-20

Import list

Navigation icons: back, forward, search, list, list next, checkmark, close.

While exporting list can overwrite existing list or saved in a new file by choosing *New file* option.



A programming list can have "symmetric" and "asymmetric" columns. A symmetric column is used when the minimum two different values need to be defined. For example, if a four-step list has a first step of 100 ms and all others of 500 ms, then it will be necessary to enter 100, 500, 500, 500 ms. However, if the output voltage and current in all steps will have the same duration, e.g. 100 ms, then it will only be necessary to enter 100 ms in the first cell and the rest will be able to remain blank.

#	Dwell	Voltage	Current
1	100 ms	5 V	3 A
2	500 ms	10 V	-
3	-	20 V	-
4	-	30 V	-
5	-	-	-

List count:  
1  
On list stop:  
**Output OFF**

The picture above shows an example of an incorrectly defined list that has more than one dwell value, but less than the number of defined steps. So, either 500 ms should be deleted in the second row (marked with arrow) or dwell values for the remaining cells should be entered.

### Dwell

Dwell is the time that the output will remain at a specific step. Dwell times can be programmed from 0 through 65535 seconds.

*If the dwell is set to 0 ms, the firmware will try to spend the shortest possible time in this step, but there is no minimum duration that can be guaranteed (it depends on the current load of the microcontroller).*

### SCPI

```
LIST:DWELL {<time>}
```

### Voltage

This column contains the voltage values for each step.

### SCPI

```
LIST:VOLTage {<voltage>}
```

### Current

This column contains the current values for each step.

### SCPI

```
LIST:CURREnt {<current>}
```

### List count

Sets the number of times that the list is executed before it is completed. The list count range is 1 through 65535 or it could be set to *infinity* ( $\infty$  sign on the numeric keypad) to execute a list continuously.

### SCPI

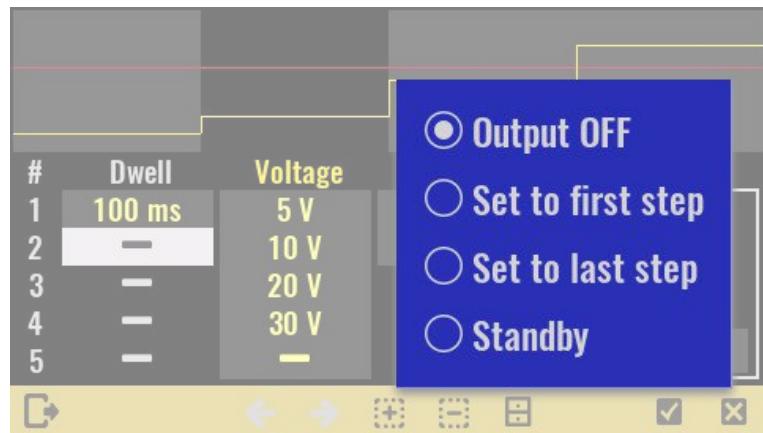
```
LIST:COUNT {<count>}
```

**On list stop**

Define channels condition when programming list execution is not prematurely stopped (e.g. with ABORT command or by user action).

**SCPI**

```
TRIGger:EXIT:CONDITION
{<condition>}
```

**11.2.1. Inserting a new step**

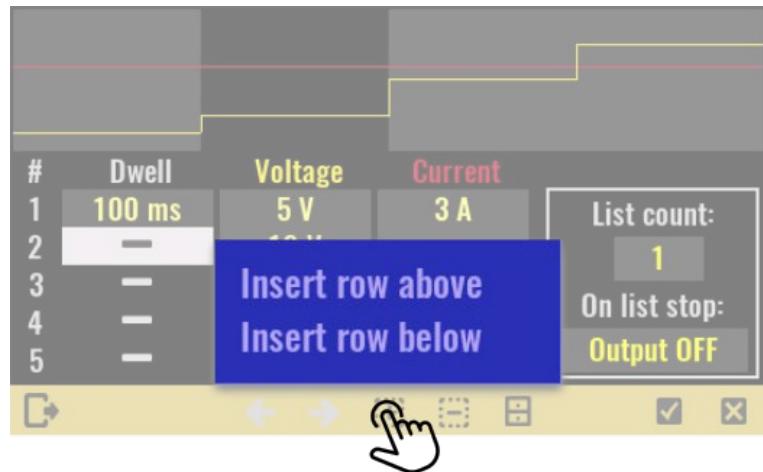
Inserting a new line is possible by selecting the + icon on the status bar when the menu with the options described below will open.

**Insert row above**

Lets you add a new step above the current cursor position. All values from the step the cursor was on will be copied to a new step.

**Insert row below**

Lets you add a new step below the current cursor position. All values from the step the cursor was on will be copied to a new step.

**11.2.2. Deleting of list items**

Deleting part or all of the list is possible by selecting the - icon on the status bar when the menu with the options described below will open.

**Delete row**

Deleting a row at the cursor position. The row must have at least one value entered to be deleted.

**Clear column from cursor down**

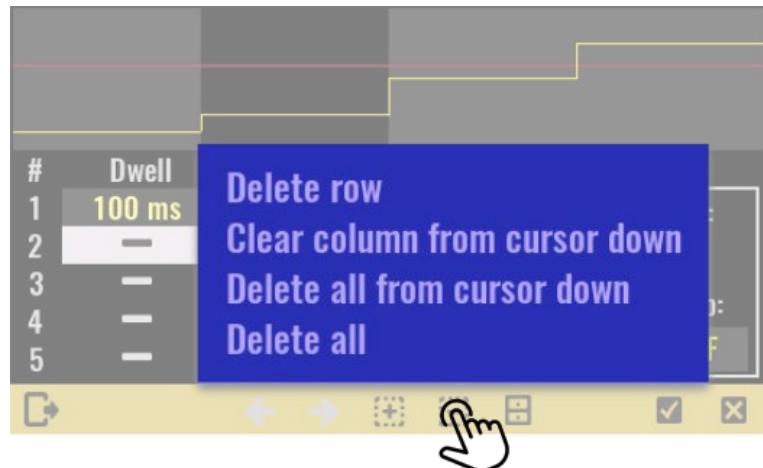
Deleting only the contents of the column from the cursor position down.

**Delete all from cursor down**

Deletes the contents of all rows from the cursor position down.

**Delete all**

Delete all contents of the list.



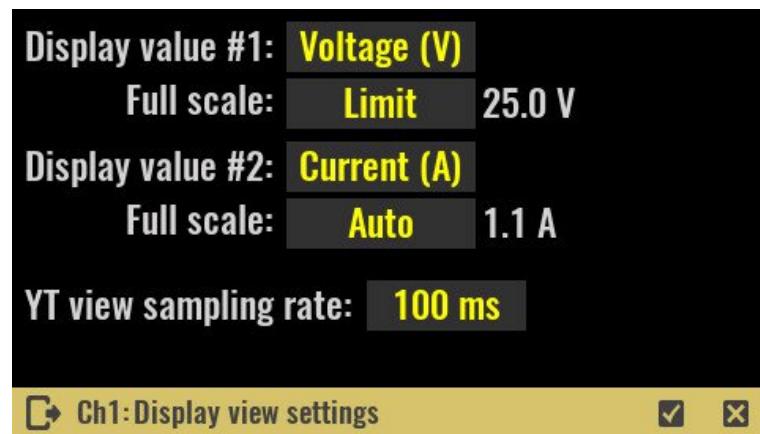
### 11.3. Display view

#### YT view value #1

YT view provides a graphical view of two measured output channels at the same time. The selected values are displayed in different colors. By definition, voltage is shown as the first value and current as the second value. Through this menu, it is possible to set a new value that will be displayed in the color of the first value.

#### SCPI

```
INSTRUMENT:DISPLAY:TRACe1 {<value>}
```



#### Full scale

This parameter defines the full scale value when displaying output values on bar graphs and YT view. Possible values are:

- *Maximum* – the max value that can be obtained is set (for example for the voltage on the DCP405 module it is 40 V)
- *Limit* – full scale is a user-set limit, for example 25 V out of a possible 40 V in the case of a DCP405 module
- *Auto* – it will dynamically monitor the user set value and increase it by 10%. For example for set 1 A, this will be 1.1 A
- *Custom* – setting the full scale to a user-defined value that must be positive and not greater than the maximum possible (e.g. 5 A to display the current on the DCP405 module).

#### SCPI

```
INSTRUMENT:DISPLAY:SCALEe1 {<scale_range>}  
INSTRUMENT:DISPLAY:SCALEe2 {<scale_range>}
```

#### YT view value #2

use this menu to set a new output value that will be displayed in a color of the second value.

#### SCPI

```
INSTRUMENT:DISPLAY:TRACe2 {<value>}
```

#### Swap values

Swap the positions of the selected display values.

#### SCPI

```
INSTRUMENT:DISPLAY:TRACe:SWAP
```

#### YT view sampling rate

This option swaps the position of the selected output values.

#### SCPI

```
INSTRUMENT:DISPLAY:YT:RATE {<duration>}
```

## 11.4. Info

### Model

Displays module's model name and version. In the case of a two-channel module, the same model name and version will be displayed for both channels.

#### SCPI

`SYSTem:CHANnel:MODEl?`

### Brand

Module manufacturer name

### Serial No

On-board MCU serial number (e.g. not applicable to DCP405)

### Total On time

Channel's total active time, i.e. the time it was powered on. Resolution is 1 minute and this information is stored every 10 minutes in module's non-volatile memory. Therefore it's possible that up to 10 minutes is lost after restart caused with power outage or system reset.

#### SCPI

`SYSTem:CHANnel:INFORmation:ONTime:TOTal?`

### Last On time

Displays the elapsed time since the channel was last powered on. Resolution is 1 minute and this information is stored every 10 minutes in module's non-volatile memory. Therefore it's possible that up to 10 minutes is lost after restart caused with power outage or system reset.

#### SCPI

`SYSTem:CHANnel:INFORmation:ONTime:LAST?`

### Temperature

Returns the temperature read on the channel's temperature sensor. The two-channel module has two separate temperature sensors, so the measured values will not necessarily be the same.

*All temperature sensors are periodically tested. When sensor does not pass the test, programmed output current is automatically limited to 2 A. If load draws more than 2 A, output current will be set to zero.*

#### SCPI

`MEASure:TEMPerature? {<channel>}`

## 11.5. Advanced options

The DCP405 module, in addition to its Hardware OVP, comes with the following advanced features:

- Dual current ranges
- Remote voltage sensing (RSense)
- Remote voltage programming (RProg) and
- Down-programmer (DP)

<b>Model:</b>	<b>DCM220 #3: 20V/4A, R2B6</b>
<b>Brand:</b>	<b>Envox</b>
<b>Serial No.:</b>	<b>002F0036434E510A20373437</b>
<b>Total On time:</b>	<b>31d 6h 22m</b>
<b>Last On time:</b>	<b>1m</b>
<b>Temperature:</b>	<b>21 °C</b>

Ch3: Information

<b>Current range</b>	<b>Best</b>
Remote voltage sensing (CV mode only)	<b>OFF</b>
Remote voltage programming using external signal source (+2.5 V for full scale)	<b>OFF</b>
Down-programmer (max. sinking power: 5 W)	<b>ON</b>

Ch1: Advanced options

### 11.5.1. Current range

#### Best (default)

The DCP405 module has two current ranges. This option allows the current limit to be set in the best possible resolution. This will automatically activate a lower range for a set current limit of less than or equal to 50 mA, and a higher range for a current limit of more than 50 mA.



#### SCPI

```
CURRent:RANGE DEF
```

Ch1: Advanced options

#### High (5A)

Sets the current range to 5 A. Therefore set output current limit could be anything between 0 and 5 A.

#### SCPI

```
CURRent:RANGE 5
```

#### Low (50mA)

Sets the current range to 50 mA. Therefore set output current limit could be anything between 0 and 50 mA. If the previous current limit was above 50 mA, the new limit will be set to 50 mA.

#### SCPI

```
CURRent:RANGE 0.05
```

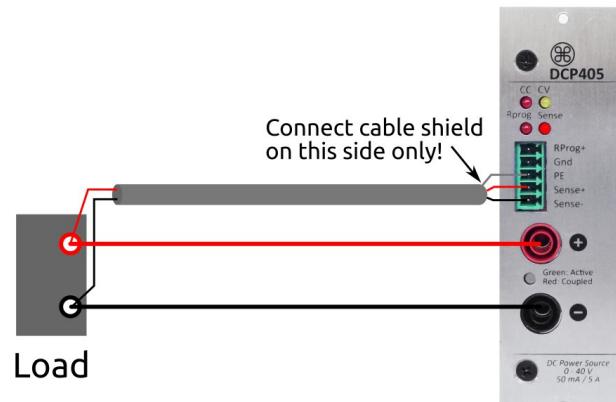
*Important: when power module is operating in CC mode, and load is changed dynamically that current vary between low and high current range, switching between ranges will generates voltage overshoots due to finite switching speed. Set current range manually in that case to avoid voltage overshoots.*

### 11.5.2. Remote voltage sensing

The main purpose of the remote (external) sensing function is to ensure that the programmed voltage is accurately delivered to the load in such a way as to cancel the voltage drop caused by the long and thin cables used to connect the load.

Remote sensing needs to connect the power leads from the output terminals to the load and the sense leads from the sense terminals to the load.

Remote sensing cables should be twisted-pair and preferably shielded. When shielded, only one end needs to be connected to an PE potential.



Any noise picked up on the sense leads also appears at the power output and may adversely affect the voltage load regulation. Twist the sense leads to minimize external noise pickup and run them parallel and close to the load leads. In noisy environments it may be necessary to shield the sense leads. When shielded, only one end needs to be connected to a PE potential (at the module end only). Do not use the shield as one of the sense conductors.

The switching between internal and remote (external) voltage sensing is carried out under the control of a firmware that controls a small signal relay. For this reason no special wiring is required to select a sensing point. The choice of external sensing is indicated by the RSense indicator on the front panel of the module.

Please note that if the wiring to a load is long, the phase shift caused by the inductance and capacitance of the wiring could become significant and could generate instability. In that case add a small capacitor on the load end of the cable to prevent oscillation.

An integral part of the remote sensing function is the reverse polarity protection, which will turn off the output immediately in case of incorrect connection of sense inputs (i.e. Sense+ to Power-, Sense- to Power+ or both).

Keep in mind that reverse polarity protection is effective only for output voltage set above 1.5 V.

When the remote sensing is active, the OVP senses the voltage at the sensing points (load) and not the output terminals.

## SCPI

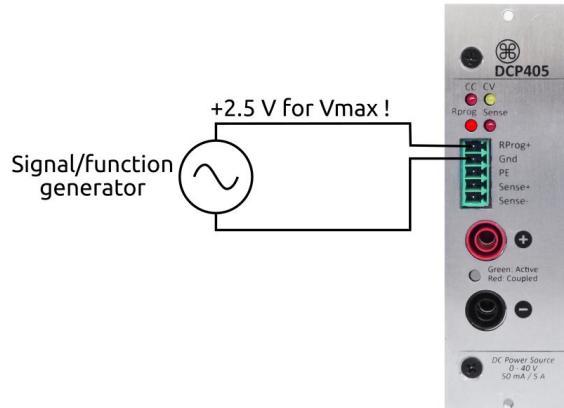
VOLTage :SENSe EXT

### 11.5.3. Remote voltage programming

Remote voltage programming allows the output voltage to be set from an external source like signal/function generator. The input sensitivity is 16 V/V so the control voltage should not exceed 2.5 V.

When this mode is set, the firmware will automatically activate the OVP and set it to max. value (40.5 V).

*Remote voltage programming will not be possible if the channel is coupled in series or in parallel with another module or tracking mode is active.*



## SCPI

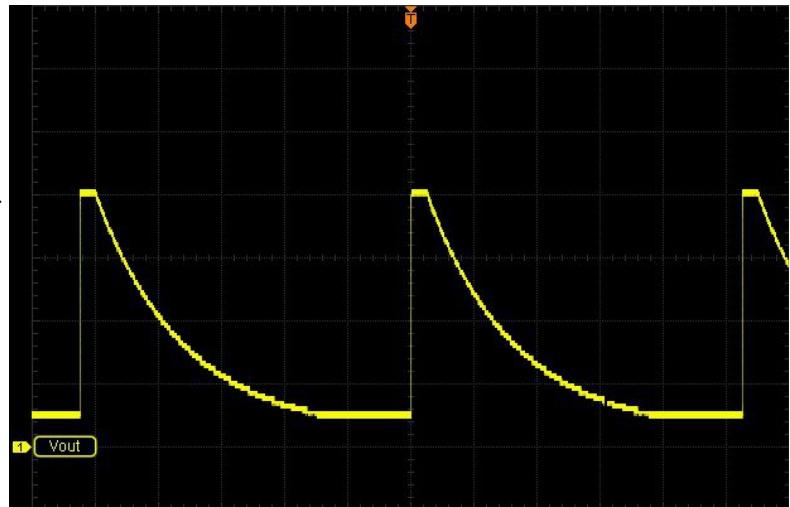
VOLTage :PROGram EXT

### 11.5.4. Down-programmer (DP)

The down-programmer (DP) can be thought of as an internal load across the power module's output terminals that helps bring the output voltage down quickly.

The primary function of the DP is to discharge the output capacitor but in some cases this feature may be used as a load to the connected device. The ability to rapidly transition from a higher to a lower constant voltage level also greatly improves the power module's output response time.

When DP is active its continuous current sinking is limited by firmware to 5 W.



Output voltage without load and with DP disabled

However, it can sink over 2.5 A in short time that is sufficient for rapidly down-programming the output capacitor together with majority of loads connected to the output.

*The DP state is relevant only while the channel output is turned on.*

**SCPI**

OUTPut:DPRog {<state>}



Output voltage without load and with DP enabled

## 12. Power modules calibration

This section gives an overview of the calibration features of the power modules. Recommended calibration interval for power modules is 1 year. This will ensure that power modules remains within specification for the next calibration interval.

For optimum calibration results the following condition are recommended:

- The calibration ambient temperature is stable and between 20 °C and 30 °C.
- Ambient relative humidity is less than 80 %
- Allow a one hour warm-up with no load connected
- Use short and thick cables to connect test setups

### 12.1. Start calibration wizard

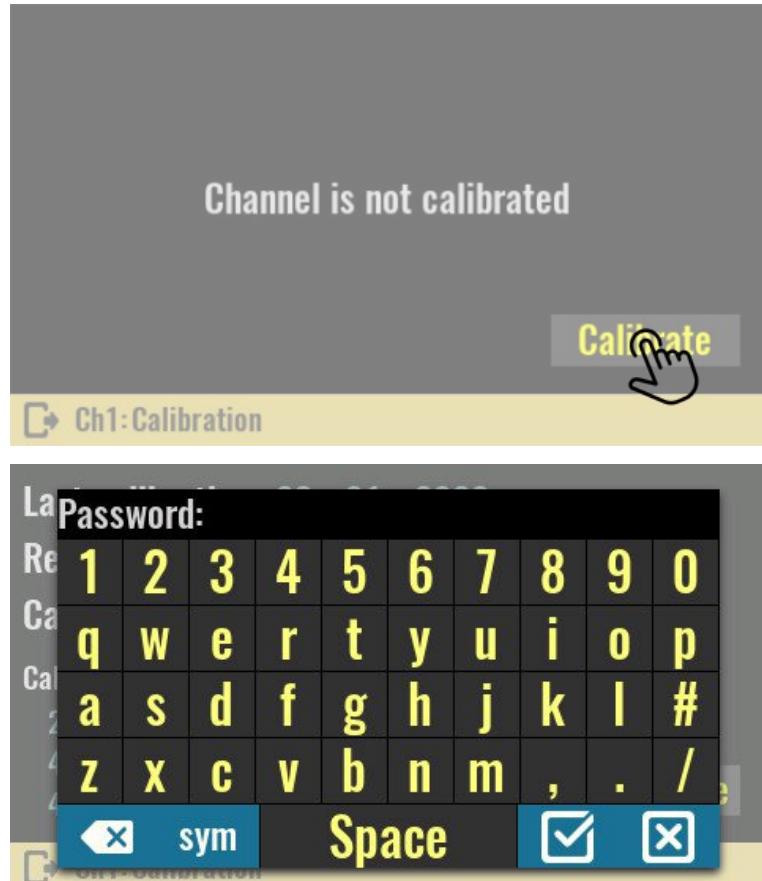
The procedure for first calibration and subsequent re-calibration is identical. If the module has not yet been calibrated or the calibration data has been deleted (using the SCPI command) a “*Channel is not calibrated*” message will be displayed.

**SCPI**  
CALibration:CLEar  
{<password>}

Calibration is a multistep process that results in slight adjustments to output voltage and current so that programmed and measured values are as accurate as possible.

Calibration involves entering the difference between set (programmed) and measured values for at least two points within the allowable output range.

The default values displayed at the start of the process can change as can the number of measuring points (up to 20).



Only one channel can be calibrated at a time. Within the same calibration session both output voltage and current can be calibrated for the currently selected channel. The calibration procedure can be canceled at any point (and entered data will be discarded).

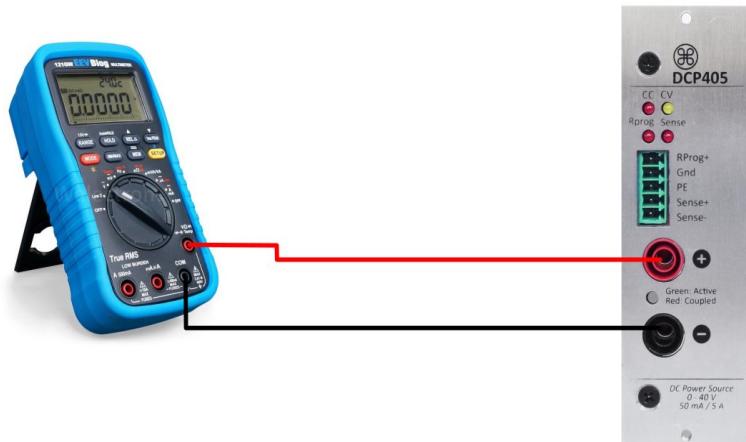
The calibration process begins by selecting the *Calibrate* option for the currently selected channel. The initial calibration password is *eezbb3* and should not be confused with a [system password](#) that is not set by default.

**SCPI**  
CALibration ON, {<password>}

## 12.2. Voltage calibration setup

For voltage calibration, disconnect all loads from the power module and connect a digital multimeter (DMM) across the output terminals.

Make sure that the power module is in the CV mode.



### 12.2.1. Voltage calibration steps

This is the first step of the calibration procedure. The *Voltage* is selected for calibration, and the first voltage point is *Set* to 150 mV.

*IMPORTANT: it is possible that module for the set 150 mV gives a voltage very close to zero or is even slightly negative. If this is the case, set the new step to e.g. 200 or even 250 mV (follow the procedure described below) that the output voltage is at least 50 mV.*

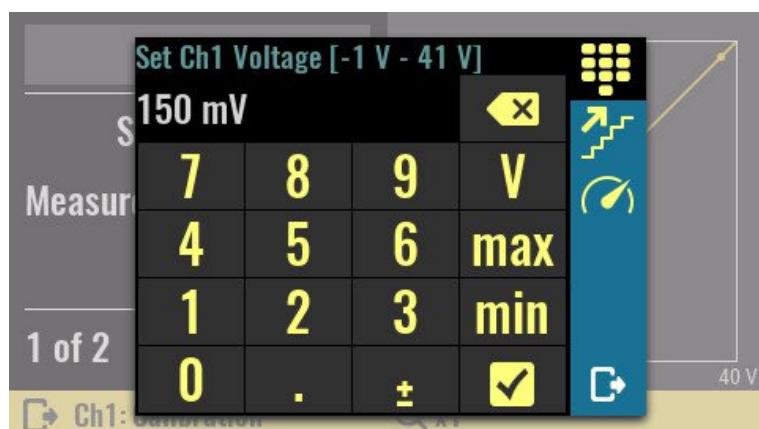
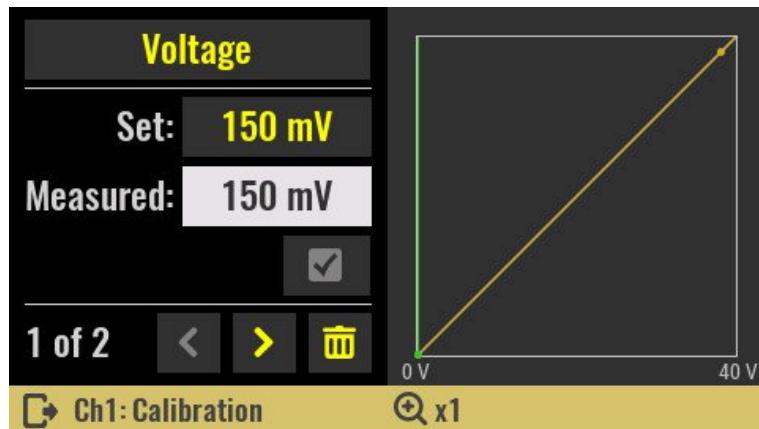
#### Set:

Displays output value that will be programmed for currently selected calibration point.

*This value is shown for information only. Instructions for deleting or changing the total number of calibration points is described below.*

#### Measured:

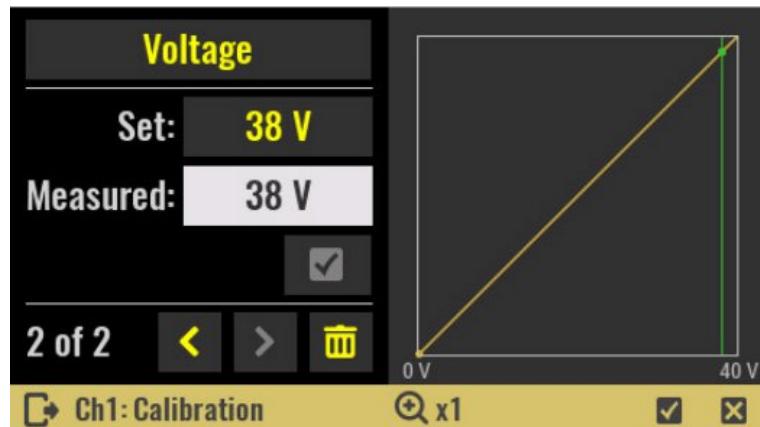
The numeric keypad will be displayed by selecting this option. The value displayed on the external DMM should be entered with an arbitrary number of decimal places. Save the calibration data by selecting the *Confirm* icon.



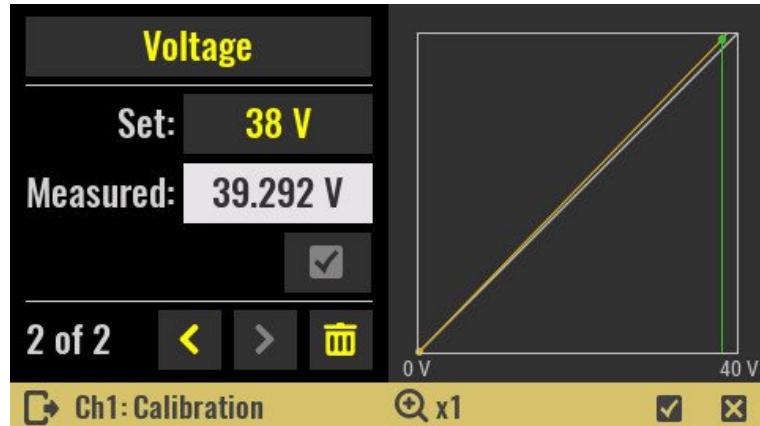
#### SCPI

CALibration:VOLTage:LEVel

Move to the next calibration point using the right navigation icon. Measure the voltage and enter it into the *Measured* field.

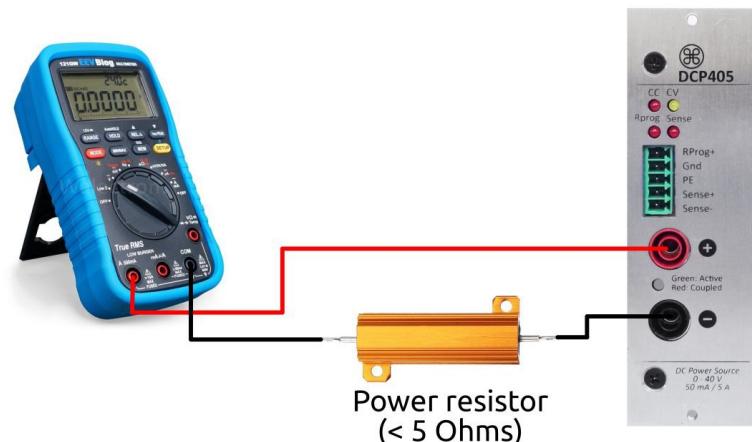


After each entry of the measured value for the set calibration points, it will be possible to see the change in the graph to the right, i.e. the deviation of the calibration curve (yellow) from the ideal curve (white).



### 12.3. Current calibration setup

For current calibration, connect an appropriate power resistor (less than  $5 \Omega$ ) in series with DMM set to current measurement.

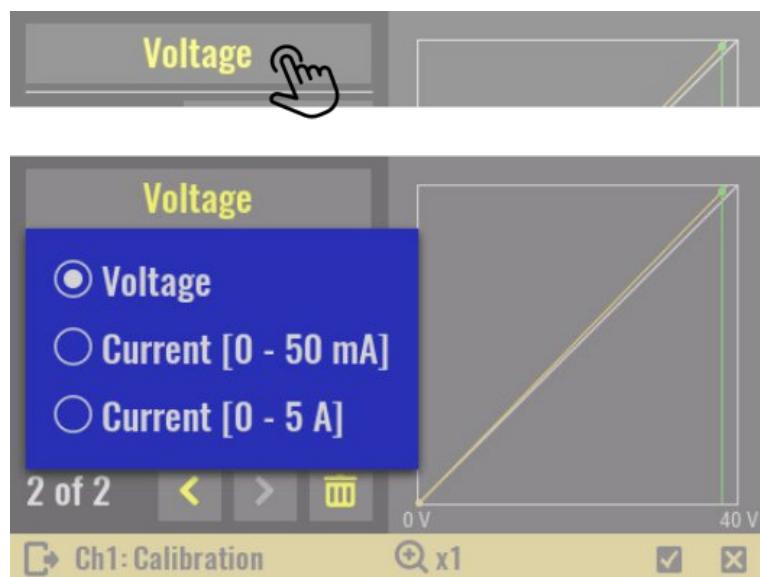


### 12.3.1. Current calibration steps

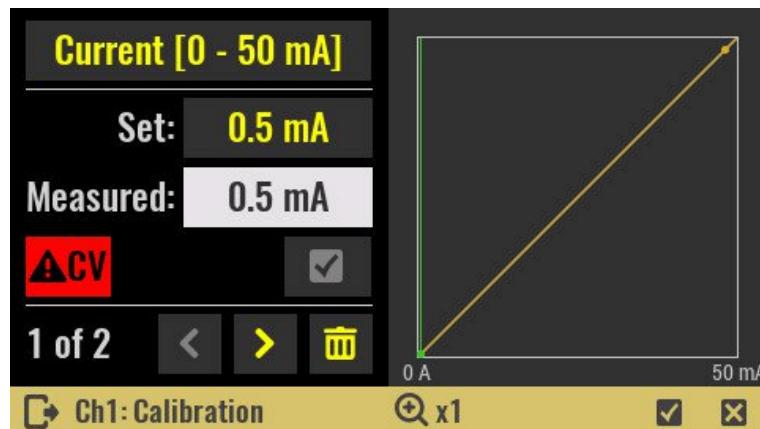
Power modules such as the DCP405 have two current ranges. Select the low (0 – 50 mA) range from the list of parameters.

SCPI

CALibration:CURRent:RANGE LOW



If the power resistor is of the appropriate value and properly connected, the power module will enter CC mode. Otherwise, a warning sign will be displayed that the output is still in CV mode.



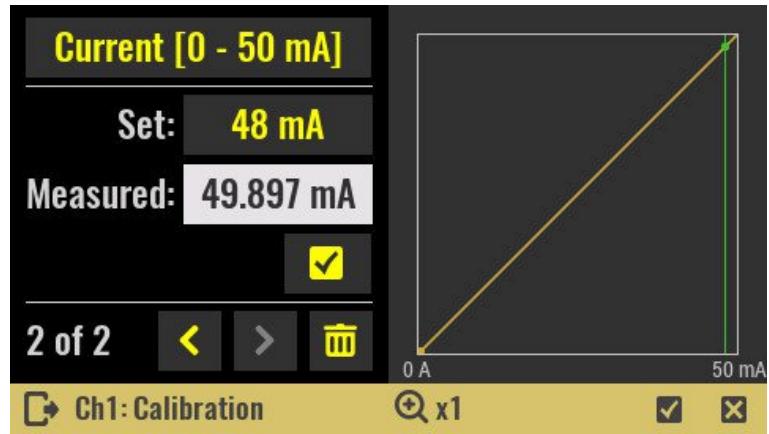
Enter the measured value for the first point.

SCPI

CALibration:CURrent:LEVel



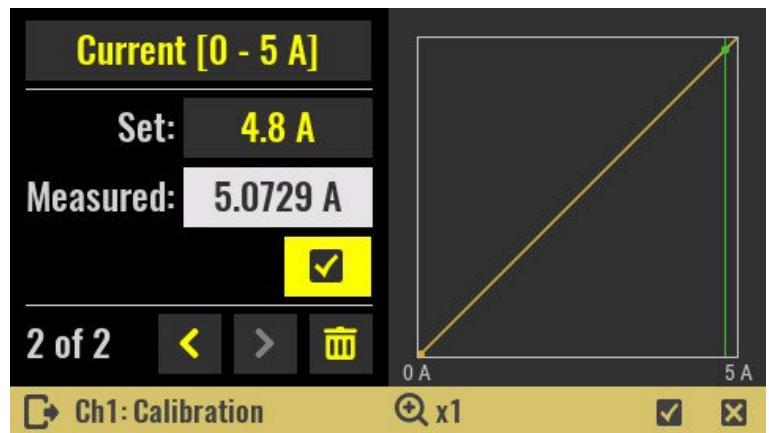
Move to the next calibration point using the right navigation icon and enter measured value for the second point.



After calibrating the low range current, repeat the same procedure for the high range and enter the measured value for the first point.



Once again move to the next calibration point using the right navigation icon and enter measured value for the second point.



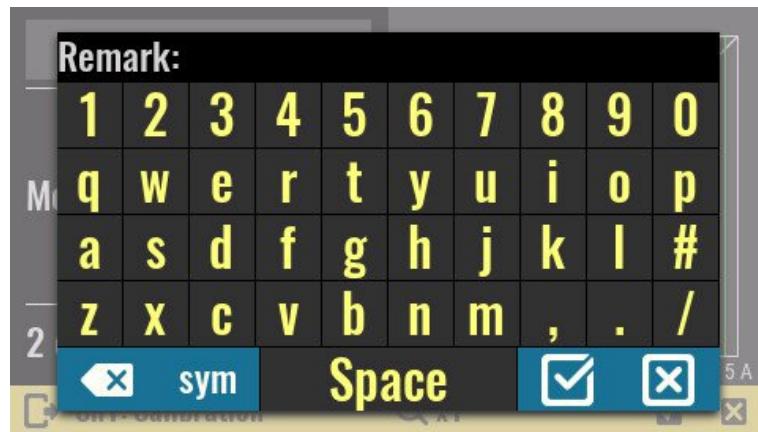
### 12.3.2. Deleting existing and adding new calibration points

Any calibration point can be deleted by selecting *Trash* icon or new calibration point can be added by simply entering a new value in calibration point *Set* field. The maximum number of calibration points for any selected parameter (i.e. voltage, low range current or high range current) is 20.

### 12.3.3. Save calibration parameters

The final step is to save the calibration parameters. Upon confirmation, the calibration parameters are written to non-volatile memory within the power module; calibration parameters are retained even if the power module is moved to another slot.

The last parameter that can be set during calibration is remark. The remark has two parts:



- a mandatory one that is system defined (a datetime stamp in format *yyyymmdd*) and
- an optional one that can be up to 32 characters long that contains a description of the calibration (eg only voltage is calibrated, etc.)

#### SCPI

CALibration:SAVE

CALibration:REMark {<user remark>}

After the calibration is successfully complete the calibration information will be displayed.

#### Calib. enabled

This option defines whether calibration parameters are used to set and read-out voltage and current output values. It can only be enabled when at least one of the calibration ranges (i.e. voltage, current high or low range) is successfully stored.



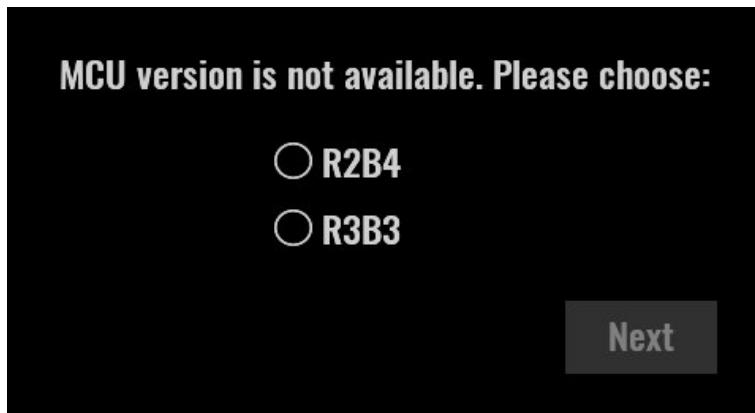
#### SCPI

CALibration:STATE ON

## 13. Firmware upgrade

The EEZ BB3 (i.e. MCU module) firmware is not protected and can be upgraded when a new version is released. In addition, some of the peripheral modules (e.g. DCM220) have their own MCU, which is also advisable to upgrade with newer versions. The upgrade procedures for these firmware are different as described in this chapter.

*Important notice: If you are upgrading the firmware from a version older than 1.6 to 1.6 or later according to the instructions below, the following message will appear on the first restart only:*



If you got BB3 as a kit through a Crowd Supply crowdfunding [campaign](#) and the color of the MCU module PCB is green, choose r2B4 (the newer version of r3B3 has a PCB in blue). If you accidentally select the wrong revision of the MCU module, you can correct it later on the [System information](#) page.

Total On time:	25d 17h 48m	Firmware:	1.6
Last On time:	1m	CPU module:	<b>STM32 R3B3</b>
Temp. AUX:	30 °C	Slot #1:	DCP405 R3B3
Fan speed:	0 rpm	Slot #2:	DCP405 R3B2
Battery:	3.05 V	Slot #3:	DCM224 R3B1
SD card:	Present		
Host name:	EEZ-BB3-1		
Serial No.:			
<a href="#">System information</a>			

### 13.1. MCU module firmware upgrade

This section describes the procedure for downloading firmware to EEZ BB3 (i.e. MCU module) from Linux and Windows operating systems using the USB DFU (*Device Firmware Upgrade*). Regardless of the operating system, the following steps will be required:

- ① Disconnect any connected loads from all outputs.
- ② Download the latest firmware available at <https://github.com/eez-open/modular-psu-firmware/releases>

#### 13.1.1. Linux

The procedure described below is made on Ubuntu 18.04 LTS and may be somewhat different with other Linux distributions. DFU is an official USB device class specification. It is natively supported by Linux. Therefore, no additional driver will need to be installed.

- ① Connect EEZ BB3 to PC using USB cable. The connector on the EEZ BB3 side should be of USB Mini type.

- ② Turn power off using the power switch on the front panel.

**MCU r2B4:**

Keep the BOOT0 switch pressed while turning the power on. The BOOT0 switch is located in the lower left corner of the front panel. If the BOOT0 switch has been pressed long enough, EEZ BB3 enters DFU mode.

**MCU r3B3:**

Keep the USER switch pressed while turning the power on. EEZ BB3 enters DFU mode.

The existing firmware will be deactivated and the Welcome page will not be displayed. The cooling fan will stop spinning after few seconds.

- ③ You can check if Linux correctly identified EEZ BB3, which must be in DFU mode. Open the *terminal* application and enter:

```
lsusb
```

A list of all connected and recognized USB devices will be displayed. If EEZ BB3 is properly connected and enters DFU mode a red marked line should appear.

```
Bus 004 Device 002: ID 174c:55aa ASMedia Technology Inc. ASM1051E SATA
6Gb/s bridge, ASM1053E SATA 6Gb/s bridge, ASM1153 SATA 3Gb/s bridge
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 108: ID 0483:df11 STMicroelectronics STM Device in DFU
Mode
Bus 001 Device 103: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 001 Device 003: ID 04d9:1400 Holtek Semiconductor, Inc. PS/2
keyboard + mouse controller
Bus 001 Device 006: ID 8087:0025 Intel Corp.
Bus 001 Device 004: ID 046d:081b Logitech, Inc. Webcam C310
Bus 001 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

- ④ Install *dfu-util* (you need sudo access rights):

```
sudo apt install dfu-util
```

- ⑤ Run *dfu-util* in folder where firmware image is downloaded:

```
dfu-util -a 0 -D bb3.dfu
```

- ⑥ Wait until the firmware image download is complete. A typical *dfu-util* output is shown below.

```
dfu-util 0.9
```

```
Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/
```

```
Match vendor ID from file: 0483
Match product ID from file: 0000
Opening DFU capable USB device...
ID 0483:df11
Run-time device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuERROR, status = 10
dfuERROR, clearing status
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 2048
```

```
DfuSe interface name: "Internal Flash"
file contains 1 DFU images
parsing DFU image 1
image for alternate setting 0, (2 elements, total size = 901464)
parsing element 1, address = 0x08000000, size = 504
Download      [=====] 100%          504 bytes
Download done.
parsing element 2, address = 0x08000200, size = 900944
Download      [=====] 100%          900944 bytes
Download done.
done parsing DfuSe file
```

- ⑦ Disconnect USB cable and recycle power using the front panel power switch.

Alternatively it is possible to download .hex firmware image:

- ① First, the .hex firmware image has to be converted into .bin format:

```
objcopy --input-target=ihex --output-target=binary bb3.hex bb3.bin
```

- ② dfu-util -a 0 -s 0x08000000:leave -D bb3.bin

Wait until the firmware image download is complete. A typical *dfu-util* output is shown below.

```
dfu-util 0.9
```

```
Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/
```

```
dfu-util: Invalid DFU suffix signature
dfu-util: A valid DFU suffix will be required in a future dfu-util
release!!!
Opening DFU capable USB device...
ID 0483:df11
Run-time device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuERROR, status = 10
dfuERROR, clearing status
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 2048
DfuSe interface name: "Internal Flash"
Downloading to address = 0x08000000, size = 919352
Download      [=====] 100%          919352 bytes
Download done.
File downloaded successfully
dfu-util: Error during download get_status
```

### 13.1.2. Windows

- ① Visit the following page  
<https://www.st.com/en/development-tools/stsw-stm32080.html> and select *en.stsw-stm32080.zip* download. You may be required to register and login on the ST web site first.

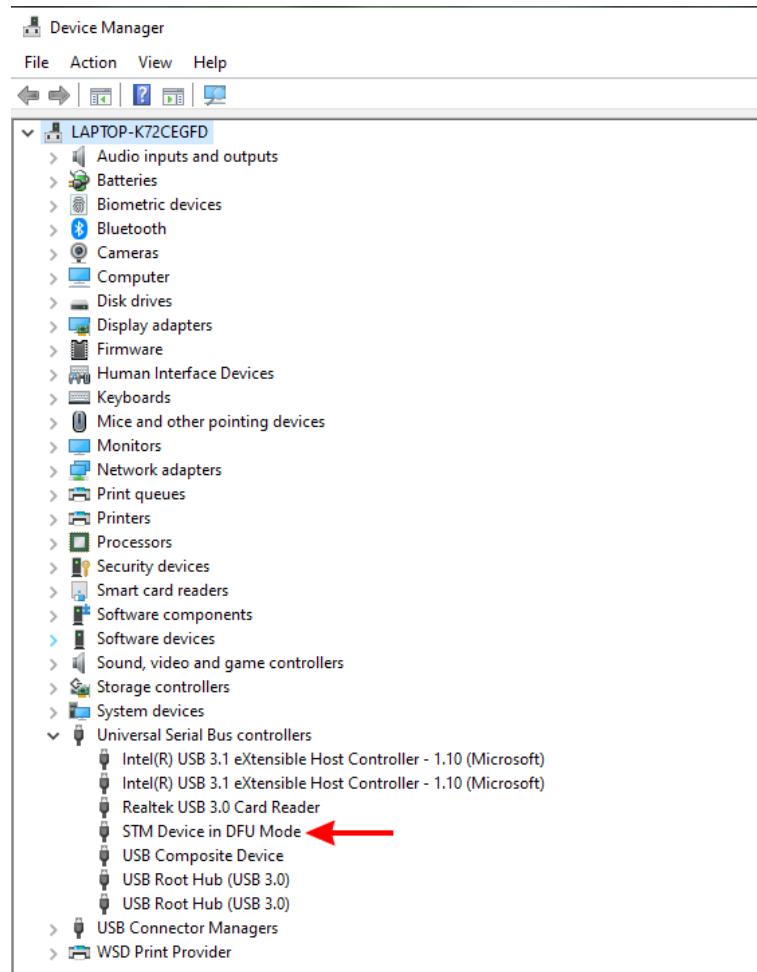
Get Software		
Part Number	Supplier	Download
STSW-STM32080	ST	<a href="#">Download</a>
General Description		DfuSe USB device firmware upgrade STMicroelectronics extension: contains the demo GUI, debugging GUI, all sources files and the protocol layer (UM0412)
Software Version	3.0.6	

- ② Unzip *en.stsw-stm32080.zip* and start installation of *DfuSe\_Demo\_V3.0.6\_Set-up.exe*

en.stsw-stm32080.zip			
	Name	Type	Compre
	DfuSe_Demo_V3.0.6_Setup.exe	Application	
	readme.txt	Text Document	
	SLA0044.txt	Text Document	
	version.txt	Text Document	

- ③ Connect EEZ BB3 to your PC using a USB cable. The connector on the EEZ BB3 side should be of USB Mini type.
- ④ Turn power off using the power switch on the front panel and keep the BOOT0 switch pressed when power is turned on. The BOOT0 switch is located in the lower left corner of the front panel. If the BOOT0 switch has been pressed long enough, EEZ BB3 enters DFU mode and the existing firmware will be deactivated and the Welcome page will not be displayed. The cooling fan will stop spinning after few seconds.

- ⑤ Check if Windows recognizes the EEZ BB3 as a device in DFU mode. It should be listed in *Device Manager* under *Universal Serial Bus controllers* section as *STM Device in DFU mode*



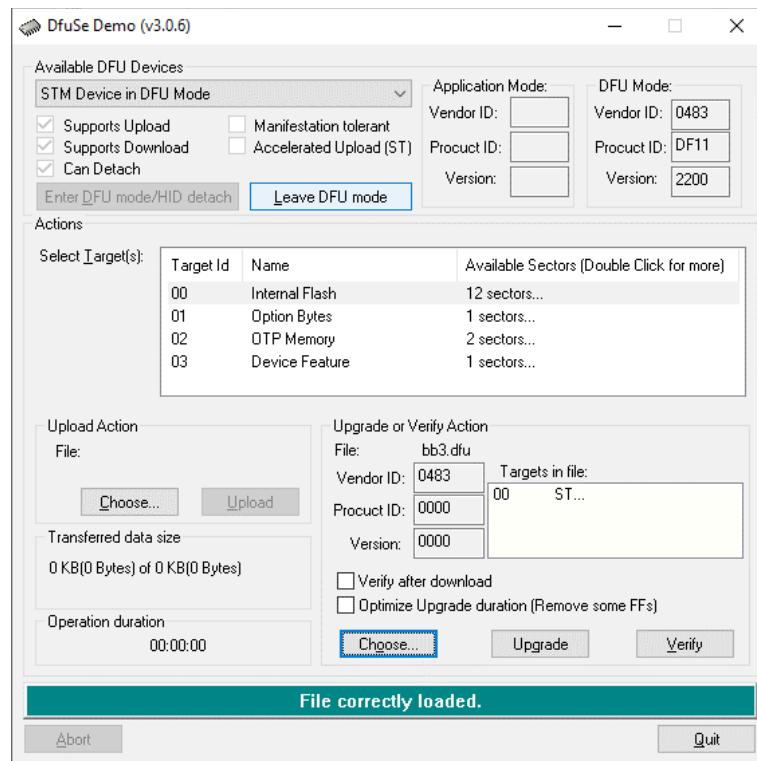
If the EEZ BB3 USB is not recognized you have to install its driver manually from the following location (64-bit):

*C:\Program Files (x86)\STMicroelectronics\Software\DFuSe v3.0.6\Bin\Driver\Win7\x64*

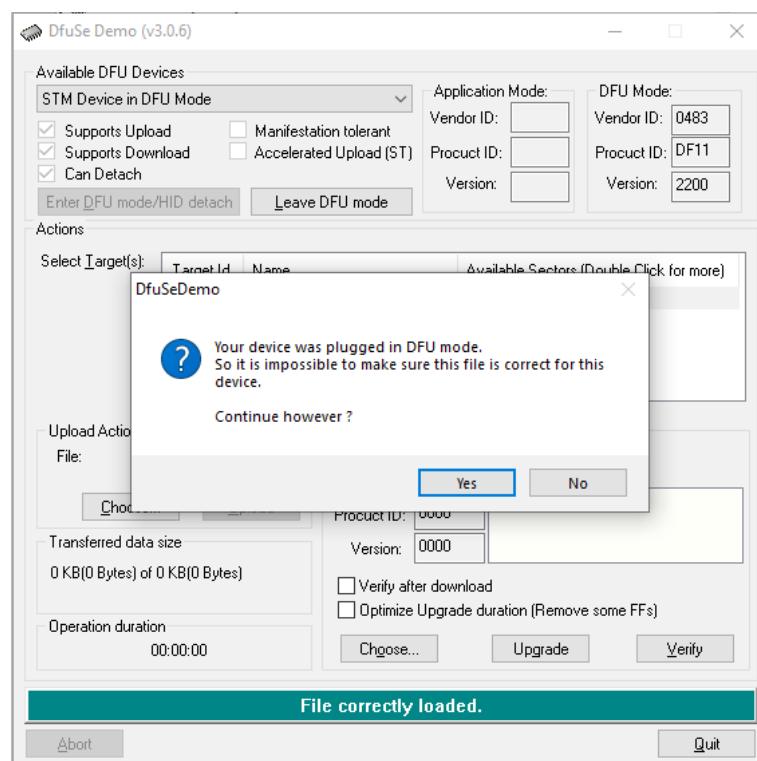
or for 32-bit at

*C:\Program Files (x86)\STMicroelectronics\Software\DFuSe v3.0.6\Bin\Driver\Win7\x86*

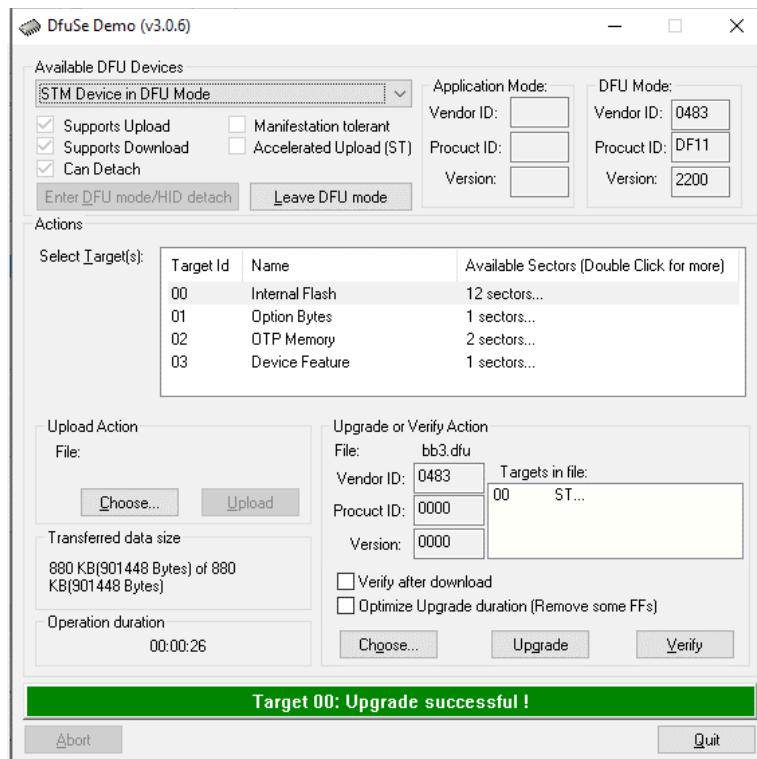
- ⑥ Start the *DfuSe Demo*, select the *Choose* button under *Upgrade or Verify Action* section to load *bb3.dfu* firmware image. A message should appear if firmware image is loaded successfully.



- ⑦ Select the *Upgrade* button under the same *Upgrade or Verify Action* section and select Yes when message box with question appears.



- ⑧ Wait until the firmware upgrade is complete.



- ⑨ Disconnect USB cable and recycle power using the front panel power switch.

### 13.1.3. Mac OSX

- ① Press *Command+Space* and type

Terminal

- ② Run in Terminal app:

```
ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
< /dev/null 2> /dev/null
```

If the screen prompts you to enter a password, please enter your Mac's user password to continue. When you type the password, it won't be displayed on screen, but the system would accept it. So just type your password and press *Enter/return* key. Then wait for the command to finish.

- ③ Run:

```
brew install dfu-util
```

- ④ Run *dfu-util* in folder where firmware image is downloaded:

```
dfu-util -a 0 -D bb3.dfu
```

- ⑤ Wait until the firmware image download is complete. A typical *dfu-util* output is shown below.

```
dfu-util 0.9
```

```
Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/
```

```
Match vendor ID from file: 1209
Match product ID from file: 2018
Opening DFU capable USB device...
ID 0483:df11
Run-time device DFU version 011a
```

```

Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuERROR, status = 10
dfuERROR, clearing status
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 2048
DfuSe interface name: "Internal Flash"
file contains 1 DFU images
parsing DFU image 1
image for alternate setting 0, (2 elements, total size = 1201048)
parsing element 1, address = 0x08000000, size = 504
Download [=====] 100% 504 bytes
Download done.
parsing element 2, address = 0x08000200, size = 1200528
Download [=====] 100% 1200528 bytes
Download done.
done parsing DfuSe file

```

## 13.2. Peripheral module firmware upgrade

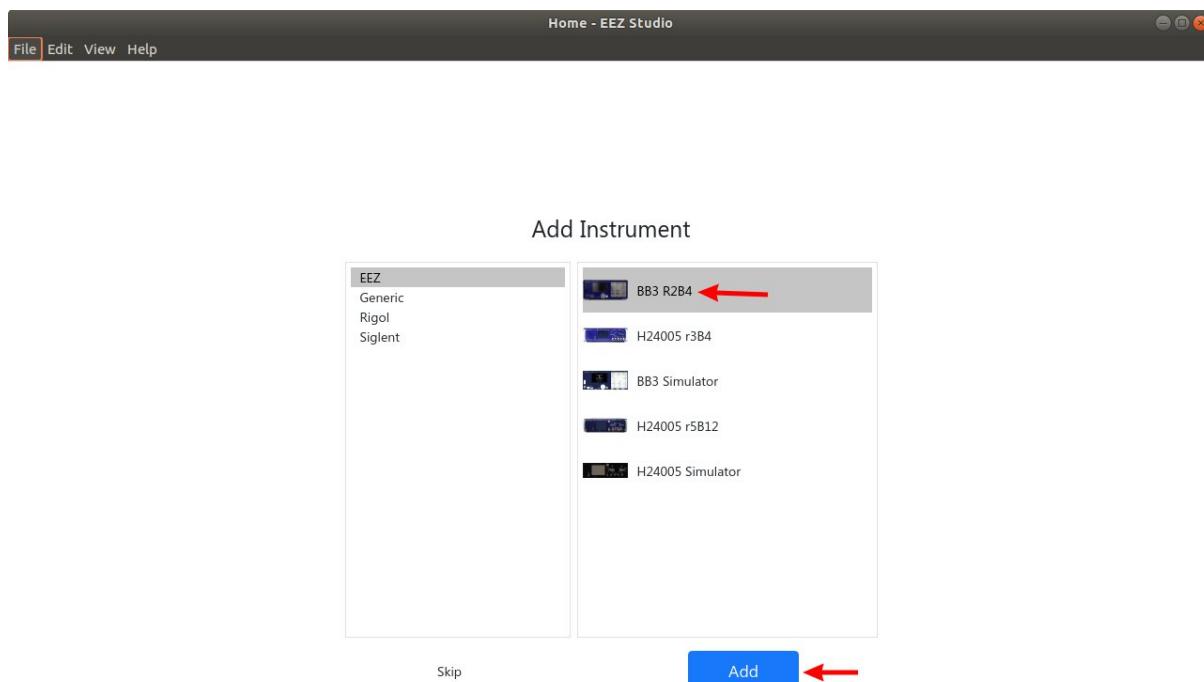
Upgrading the peripheral module firmware can be done in two ways by following the step by step procedures described below. As with the MCU module firmware upgrade, disconnect any connected loads from all outputs because after the firmware upgrade is complete, all peripheral modules will be reset.

### 13.2.1. Using EEZ Studio to upgrade peripheral module firmware

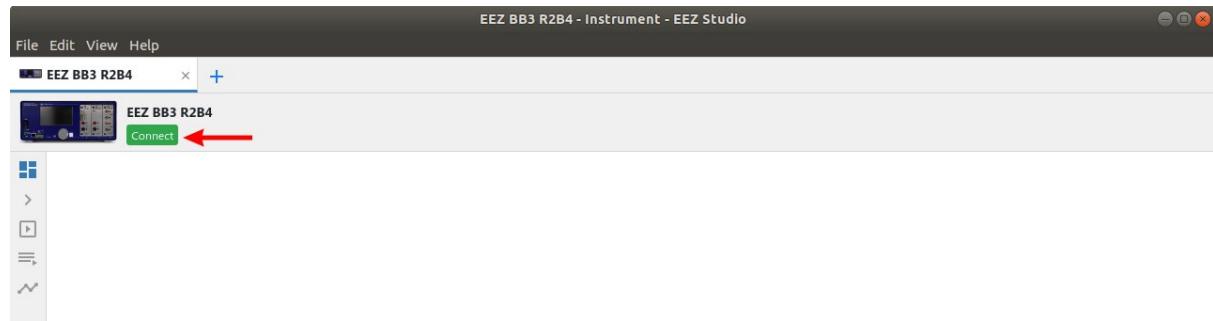
- ① The easiest and fastest way to upgrade the firmware on a peripheral module is through EEZ Studio. When EEZ Studio is installed and communication with EEZ BB3 is established, it is a one-click action. If EEZ Studio is not installed, download the latest EEZ Studio available at <https://github.com/eez-open/studio/releases> and install it on your computer.

*Important: A EEZ BB3 MCU module firmware version 1.1 or later is required to use this method and the computer on which EEZ Studio is running should have an active internet connection.*

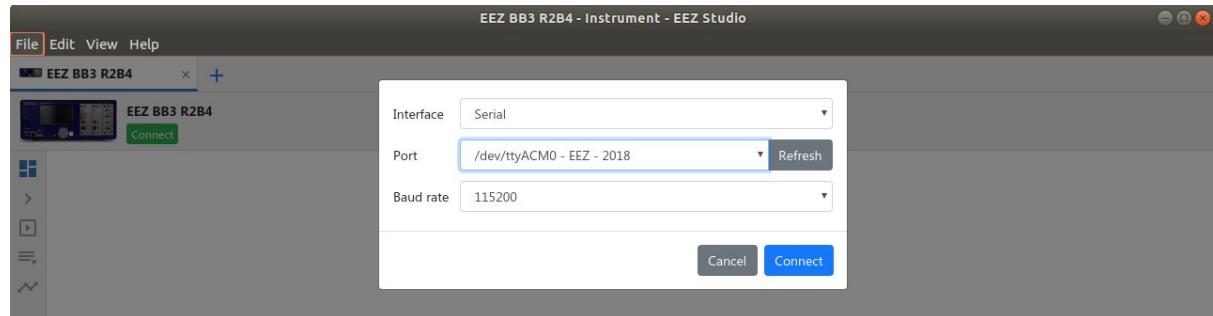
- ② Start EEZ Studio and add EEZ BB3 from the list of preinstalled instruments on the Home page.



- ③ Select Connect option to open connection menu.

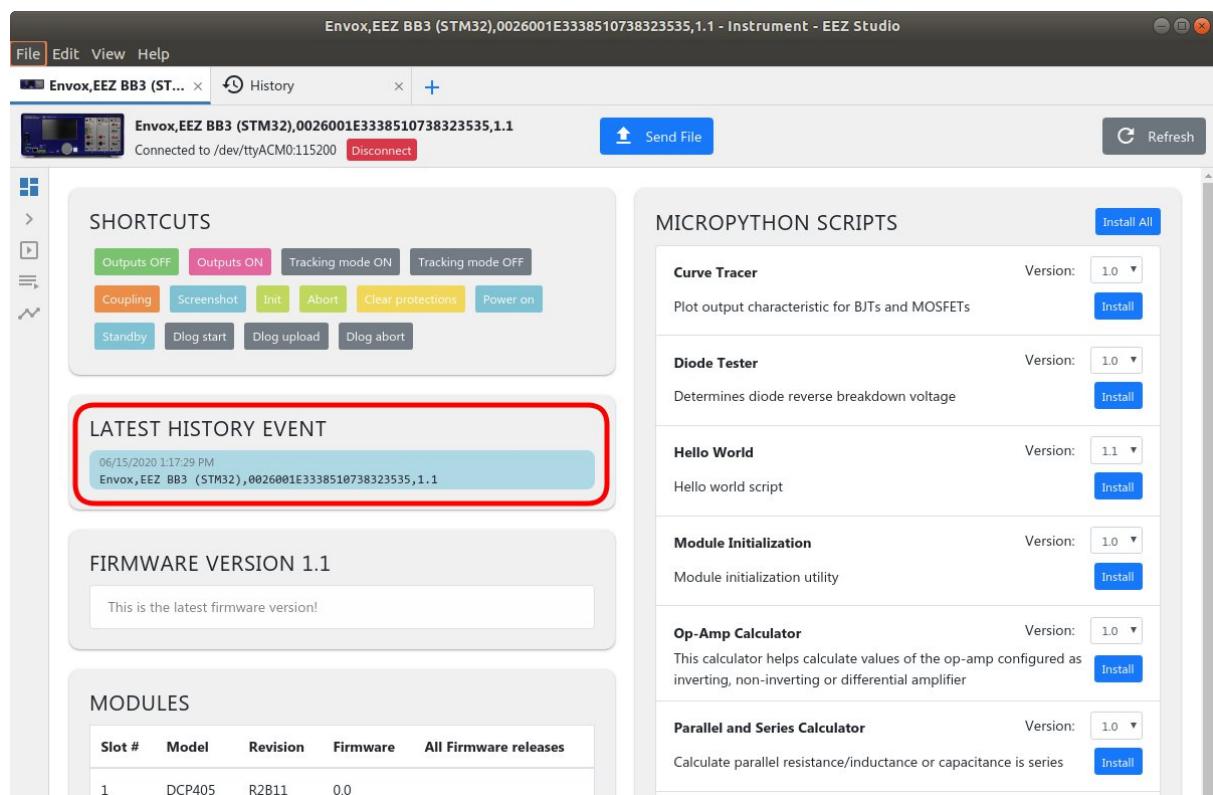


- ④ Select connection type, for USB choose *Serial* from *Interface* combo box. Click on *Refresh* button if *EEZ - 2018* is not visible in the list of serial terminals.

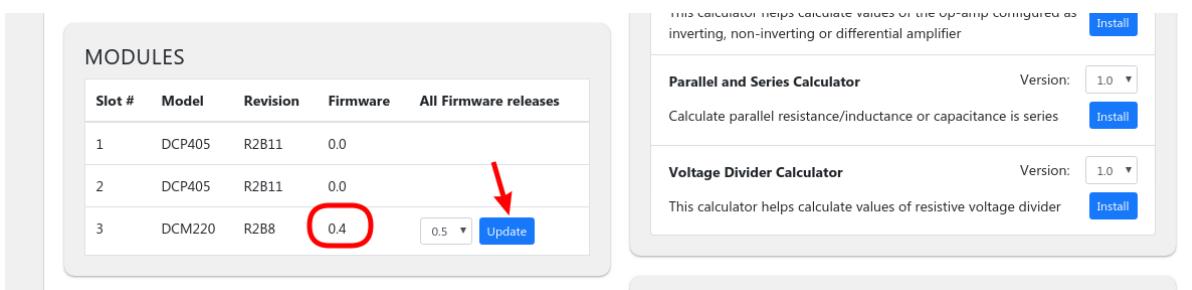


When *Ethernet* is selected as communication interface, the *Server address* has to be defined. Enter here EEZ BB3's IP address (See [Ethernet](#) settings).

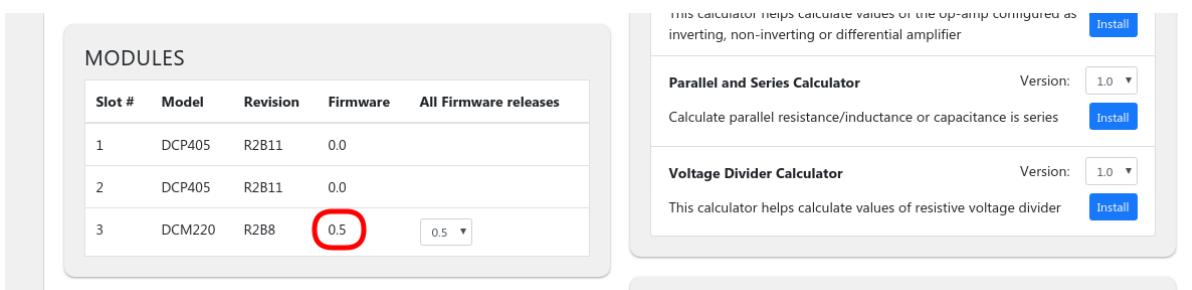
Successfully established connection is indicated by receiving identification string as response to *\*IDN?* SCPI query that EEZ Studio will send automatically to the EEZ BB3 over the selected interface. The identification string can be seen in the *Latest history event* section.



- ⑤ Scroll down to the Modules section where all recognized installed modules are visible. For mod-

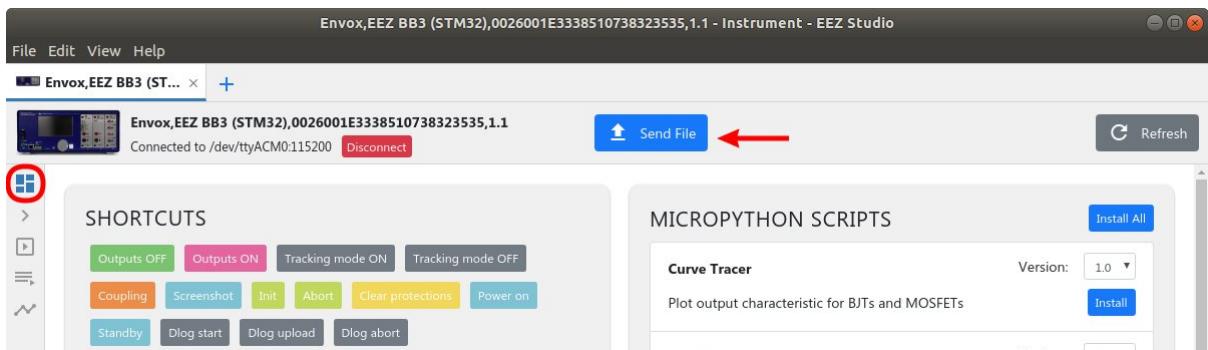


- ⑥ When the download is successful, the displayed firmware version will also be updated.

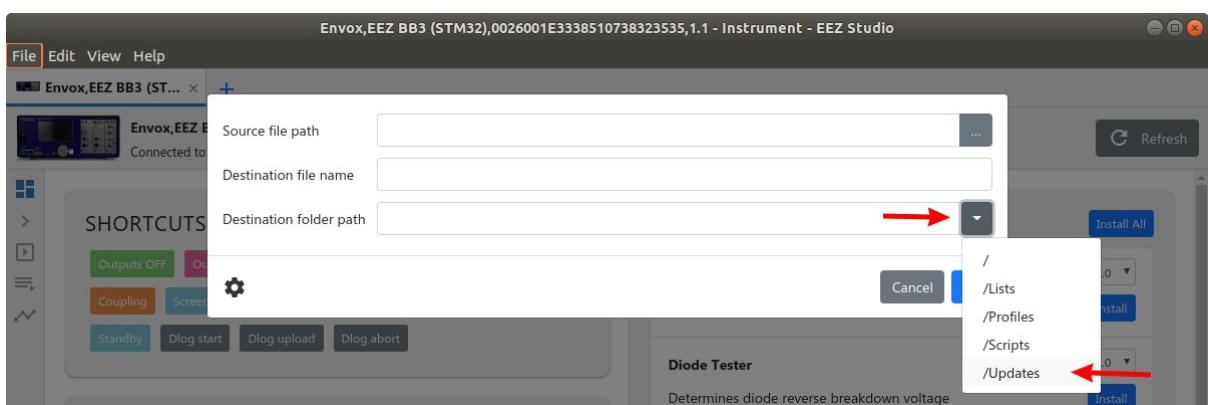


### 13.2.2. Upgrade peripheral module firmware using GUI on EEZ BB3

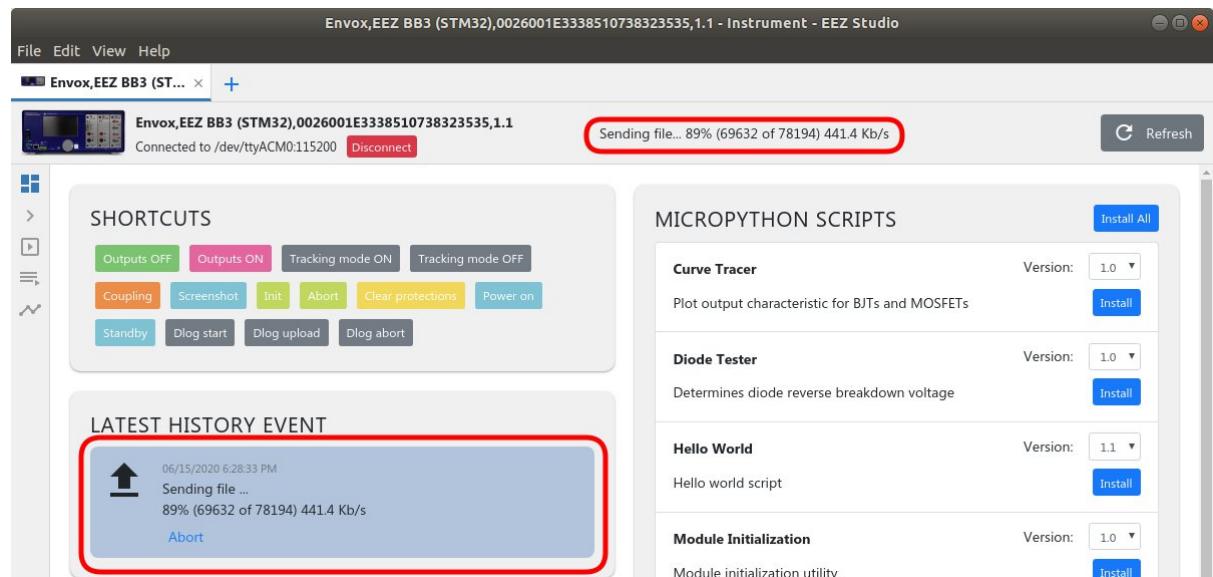
- ① Visit <https://www.envox.hr/eez/eez-bench-box-3/firmware.html> for a list of all existing modules. Download the .hex file for the module whose firmware you want to upgrade.
- ② The selected .hex file will need to be transferred to the */Updates* folder on the EEZ BB3 SD card. Since there is currently no option to mount the SD card as a USB mass storage you will need to remove the SD card and insert it into the SD card reader of your PC and skip to step ⑤. Alternatively, you can do this using the EEZ Studio and *Send file* option.



- ③ Make sure the *Destination folder path* is set to */Updates* otherwise the peripheral module will not be able to find the firmware image file.



- ④ Wait until the file transfer is complete.

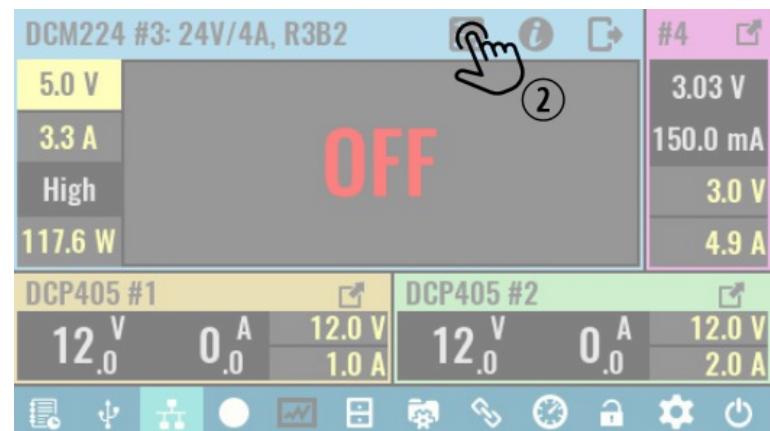


- ⑤ The procedure can now be continued on EEZ BB3 side. In the case of a dual channel DCM220 or DCM224 power module, it does not matter which channel will be used to get to the channel settings page. In the example shown, a third channel was used.

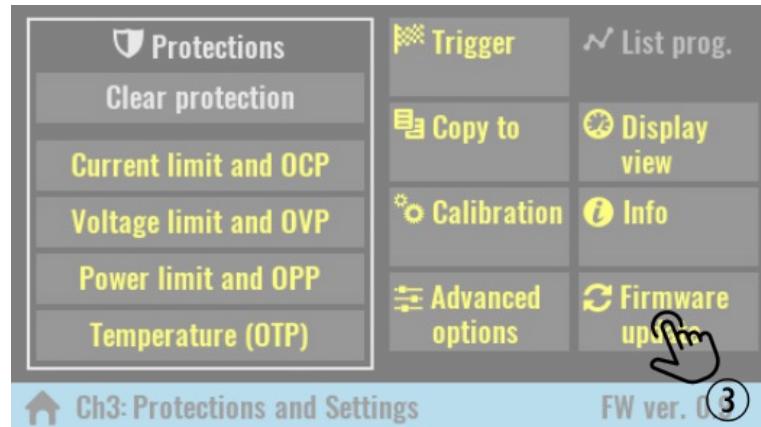
First you need to tap on *Maximize* icon.



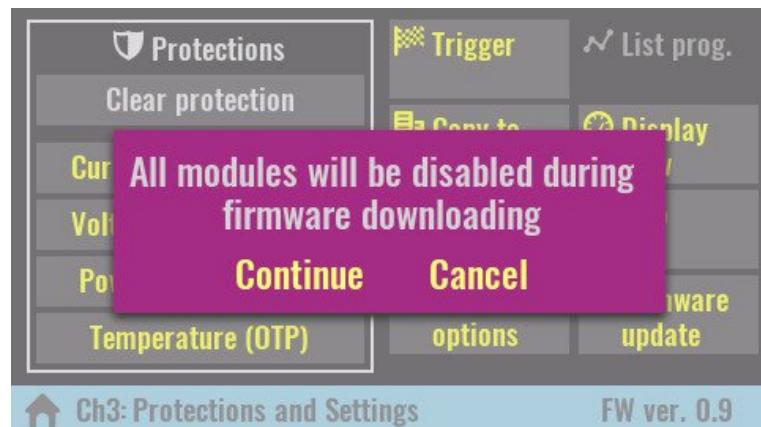
- ⑥ Now select the *Settings* icon.



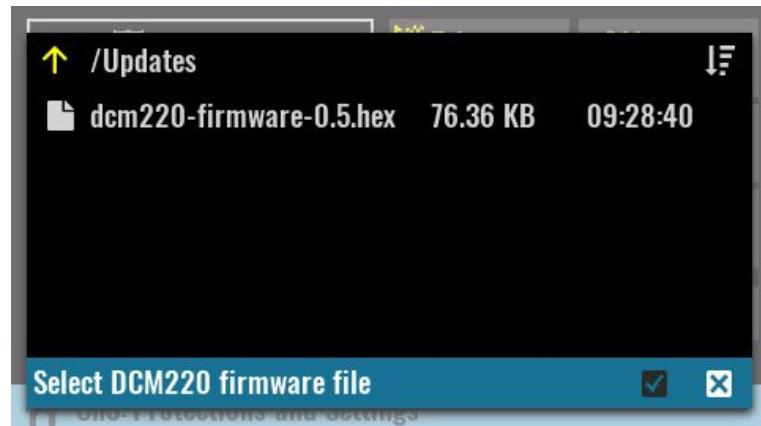
- ⑦ On the channel settings page you can see the latest version of the firmware (*FW ver.*), and next to it is *Update* option.



- ⑧ Selecting the *Update* option will first display a warning message.



- ⑨ Finally, you will need to select the file that has been transferred and wait for the firmware download to complete.



## 14. Data logging

EEZ BB3 provides simple data logging of up to 4 output values of installed power modules. Logged output values can be voltage, current or power. Logged data is stored on the SD card and can be viewed while the logging is in progress or upon completion or termination of logging.

Finally, it will be possible to transfer logged data to a computer using the [EEZ Studio](#) application or using similar applications using SCPI commands.

### 14.1. Data logging

To start data logging, tap on the *Dlog* icon, when a menu will open with the following options:

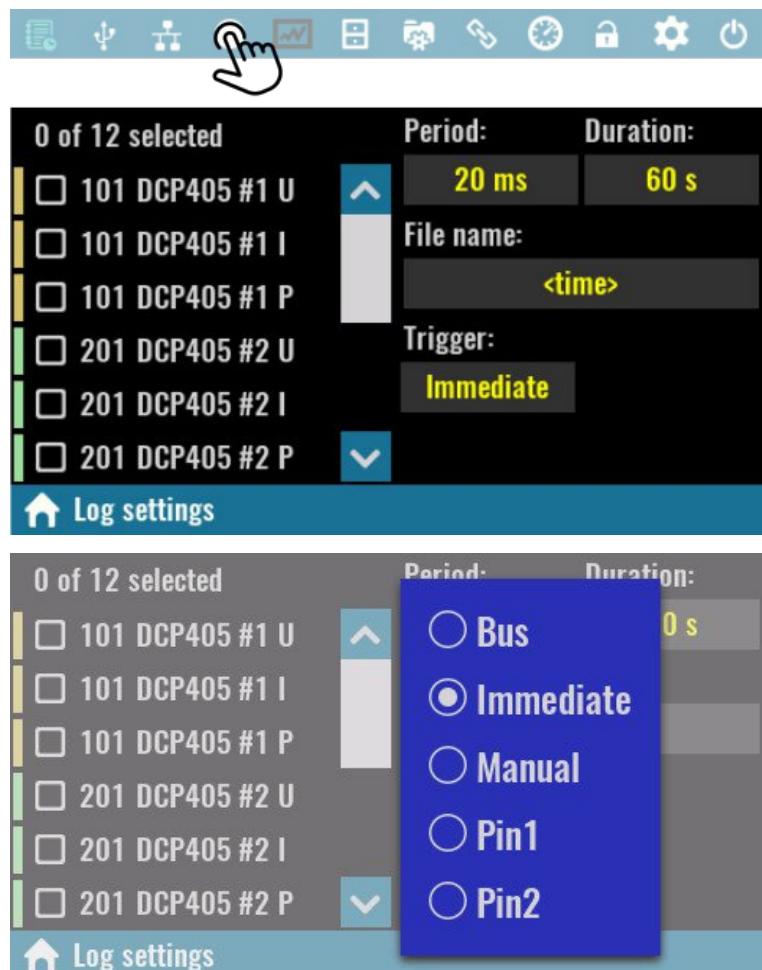
Scrollable list of all available measurement targets, e.g. voltage, current and power for each power channels.

#### Period

Specifies the sampling rate of data logging, i.e. the time between two measurements to be recorded. For example, if the period is 20 ms, this means that measurements will be taken 50 times per second.

#### Duration

Max. duration of data logging. If you want continuous data logging, enter infinity ( $\infty$ ) from keypad. Logging can be interrupted at any time with a new tap on the *Dlog* icon when its color has changed to red. Once interrupted data logging cannot be resumed.



#### File name

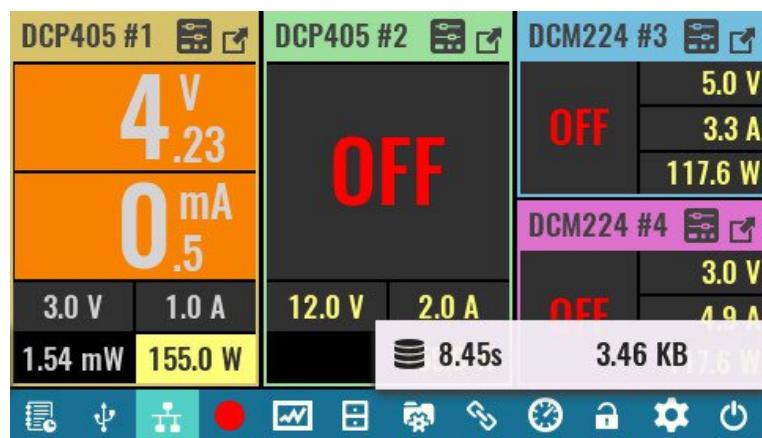
The name of the file where the logged data will be stored. The file with the name entered will be created on the SD card in the *Recordings* folder. If the file name is not entered, the current time and date will be used in the following format: *yyyy\_mm\_dd-hh\_mm\_ss.dlog*.

#### Trigger

Data logging can be initiated by various trigger events. See [General trigger settings](#) for more information.

The *Start recording* option triggers data logging when the color of the *Dlog* icon will change to red. The output values being recorded will also be clearly marked.

Data logging can be interrupted at any time, and cannot last longer than the *Duration* set. If duration infinity ( $\infty$ ) is selected, logging will continue while EEZ BB3 is active and will be limited by free space available on the SD card.



During data logging, a floating menu will appear that will be able to move anywhere on the screen, displaying the elapsed time and data size.

## 14.2. Displaying recorded data

Log viewer lets you view recorded data while data logging is in progress. In this case, the recorded data will be displayed at the rate at which it was sampled (i.e. if the period is set at half a second, then new data will also appear every half a second).

The status bar will show information about the elapsed time and data size, as well as a *D/og* icon that can terminate logging early.

To return to the main page, use the *Home* icon.

When logging is completed or terminated early, the *log viewer* will open automatically. It will now be possible to scroll through the recorded data using the following controls displayed in the status line at the bottom:

### Time offset

Position on timeline from start of measurement. Moving along the timeline is possible by directly entering a new value with a numeric keypad, using an encoder knob, or using a scroll bar on the right side of the status bar.



### Time div

Logging data display area has 12 x 6 cells graticule that helps to determine time duration and measured amplitude more easily. This parameter defines the duration of one time division. The minimum duration is 800 ms and is determined by screen resolution and displaying principle (i.e. minimum one pixel per sample).

### Total duration

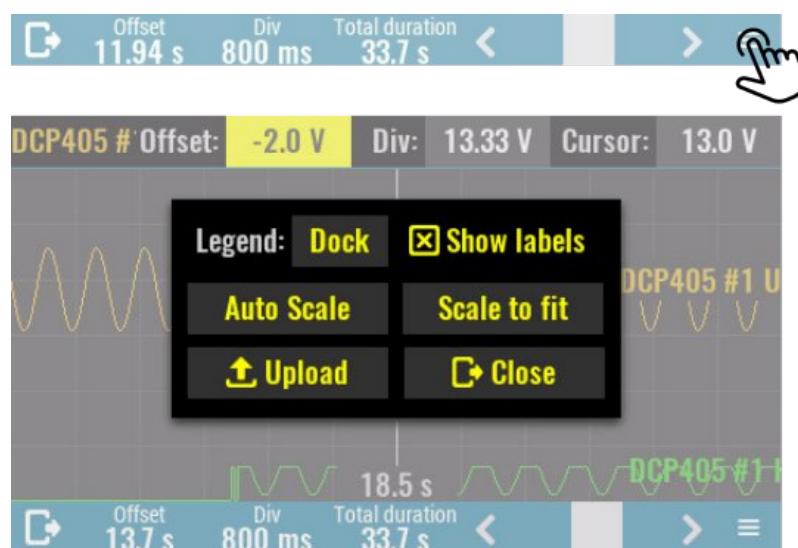
Displays the total duration of logged data in seconds.

The *Log viewer* has in its right down corner an icon that provides access to the new menu with the following options:

### Legend

Select one of the following logged data legend:

- *Hidden* – no Legend menu is shown
- *Float* – displays a floating (“unpinned”) Legend menu with one row per recordings
- *Dock* – the single row legend menu is displayed at the top



**Auto Scale**

Reset *Offset* and *Div* values for the all traces to initial values.

**Upload**

Transfer logged data to the EEZ Studio. This option is available only if active connection exists (via USB or Ethernet).

**Show labels**

Displays units on the right end of the trace.

**Scale to fit**

Maximize the view of all data traces.

**Close**

Close this menu.

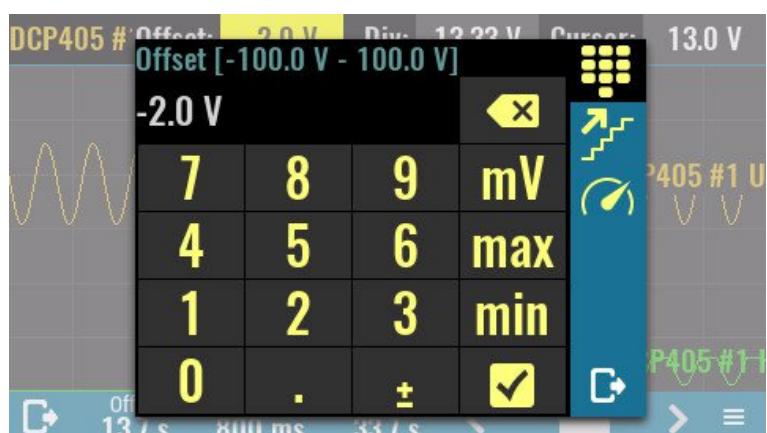
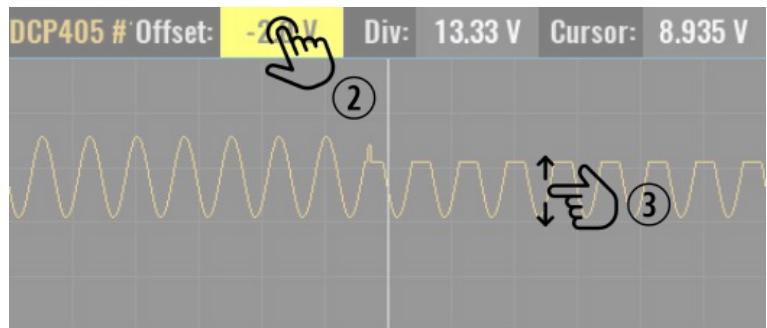
When opening the *log viewer* will sort all recorded traces one above the other for better visibility. However, this order can be changed as desired by changing the *Offset* parameter of each displayed trace. Trace y-axis offset can be changed in the following way:

- ① Tap on the desired trace *Offset* cell on the *Legend* menu.



- ② Tap once again if you'd like to enter exact value of the new offset. The numeric keypad will appear and any value within displayed range can be entered (e.g. -100 to 100 V as shown below).

*Offset* value define trace position in reference to the middle of the viewing area that represents 0 V.



- ③ The trace offset can be also changed by simply tap on trace and move it new position. While moving displayed *Offset* value will change accordingly.

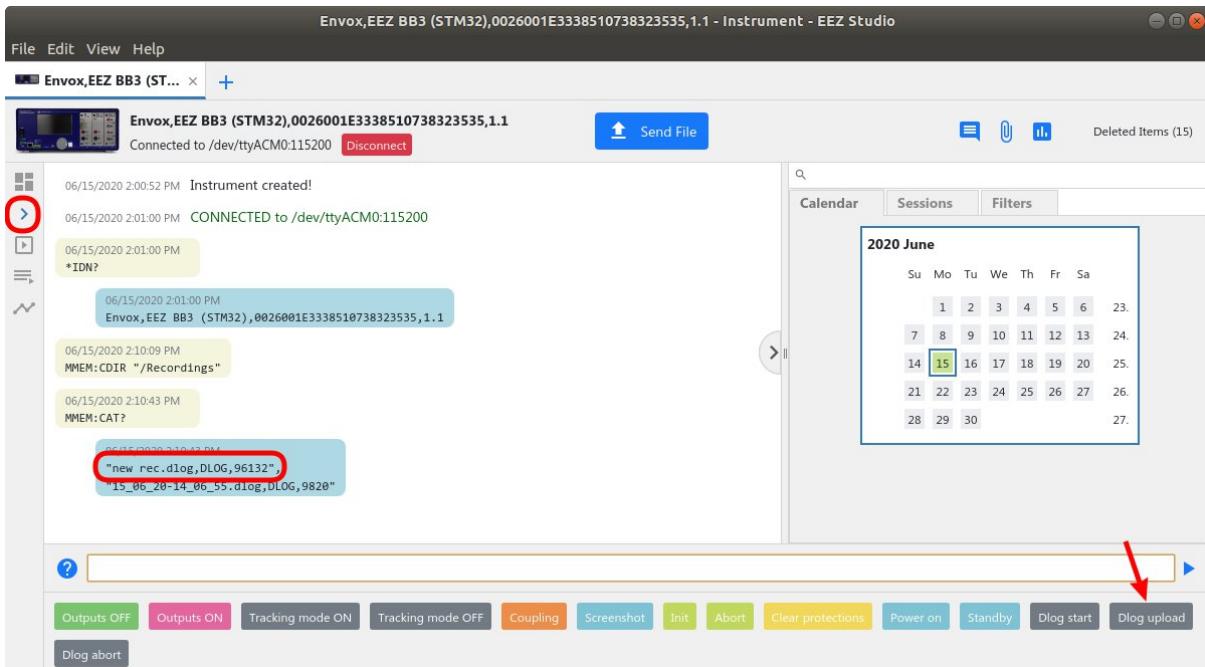
Trace amplitude can be changed in the similar way by changing its *Div* value in the *Legend* menu.

Note that when a docked legend is selected, a trace can be selected by tapping on its label



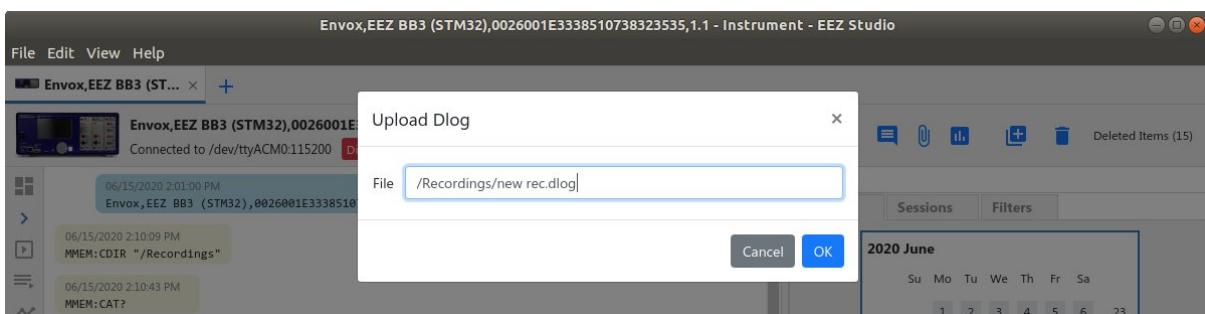
### 14.3. Logged data transfer to a computer

- ① Start EEZ Studio or, if not already installed, follow the installation steps as described in [Using EEZ Studio to upgrade peripheral module firmware](#) section.
- ② Data log files are stored into *Recordings* folder on the SD card. To access that folder switch to the terminal tab by selecting the icon of the same name on the left and enter **MMEM:CDIR "/Recordings"**. The **MMEM:CAT?** query returns all data log filenames.

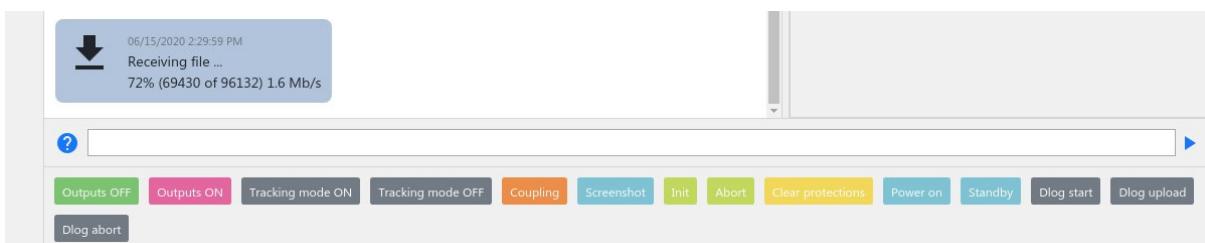


Select *Dlog upload* javascript from the list of the shortcuts.

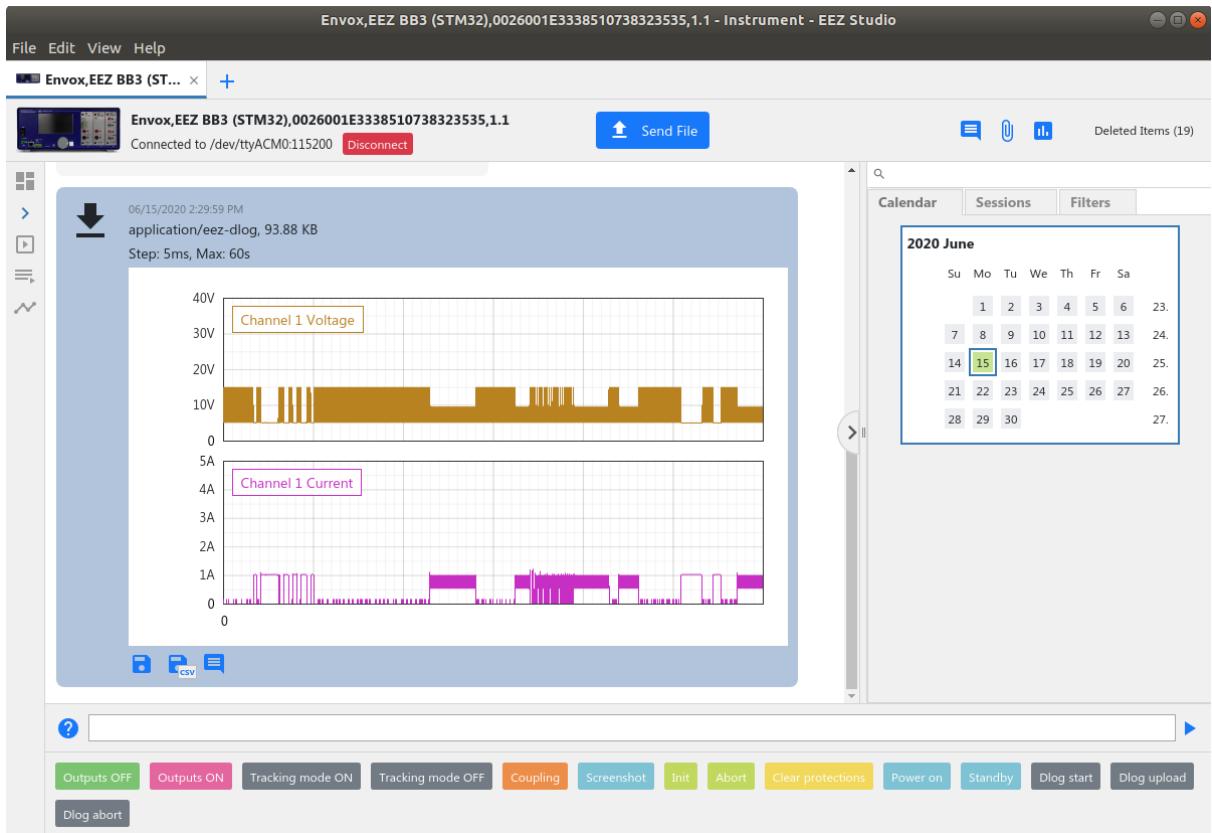
- ③ Enter the data log file name:



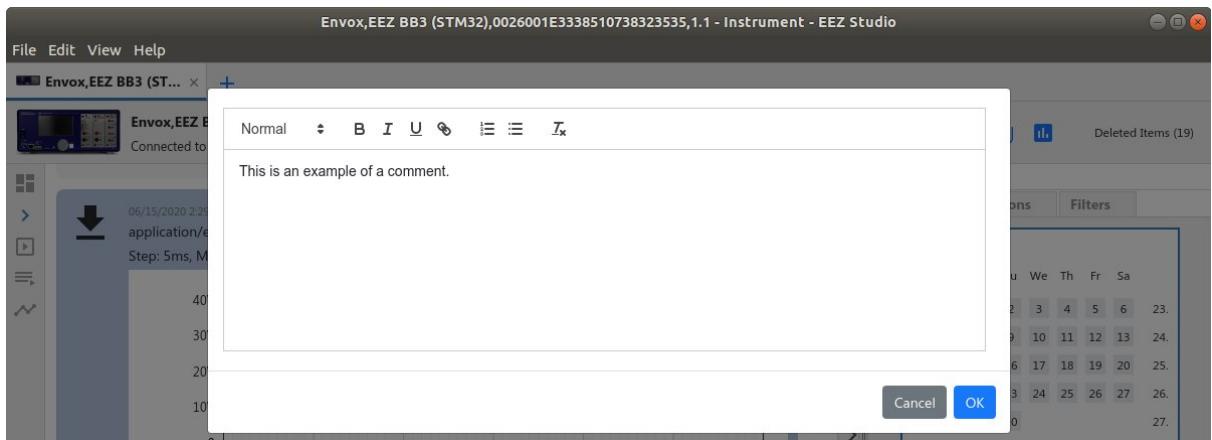
- ④ An upload progress bar will be displayed.



- ⑤ Data log will appear as new item in the SCPI session view. You can open it for further analysis



An arbitrary comment can be added by selecting the *Comment* icon when new entry form will appear:



Once added, comment can be modified or deleted by hover mouse on comment area when two new icons will appear:

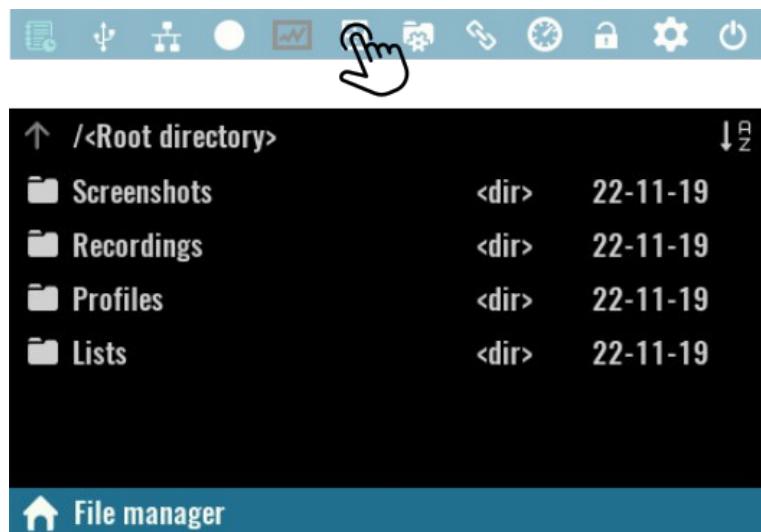


#### 14.4. Initiate data log transfer on the EEZ BB3

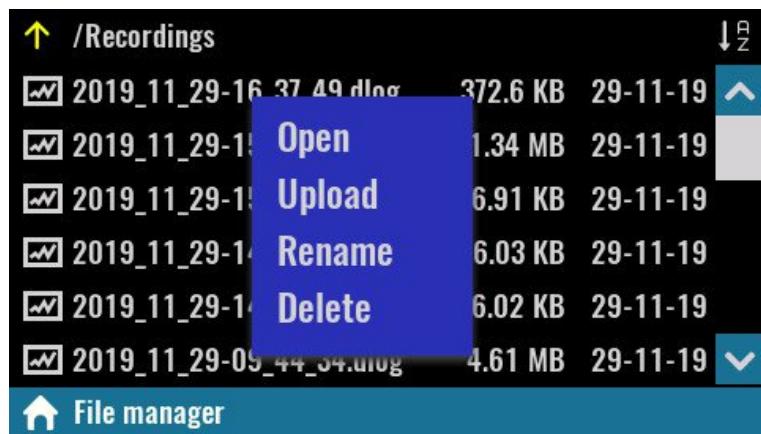
Transferring the data log to the EEZ Studio can be initiated on the EEZ BB3, too. Before initiating the transfer, make sure that the connection with the EEZ Studio is established as described in steps ① to ④

in previous section.

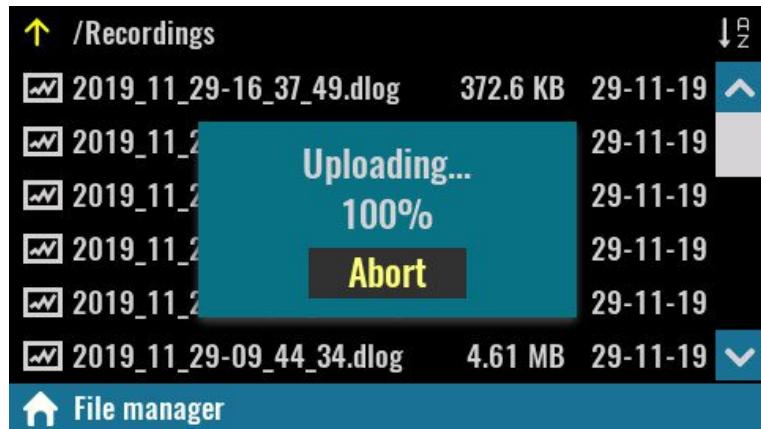
- ① Open the *File Manager* and move to *Recordings* folder.



- ② Click on desired data log file and select *Upload* option from the menu.



- ③ Wait until the data transfer is complete.



#### 14.5. Establishing serial (via USB) communication with a Linux computer

EEZ Studio users on a Linux computer may receive a series of error messages every time they successfully connect to EEZ BB3 via the serial interface. The reason for this is the existence of a *Modem manager* application that will send a series of modem commands to the connected device (EEZ BB3 in this case). The EEZ BB3 has ability to interpret only SCPI commands and therefore on each modem command it will response with an error message.

If the *Modem Manager* application is not needed (i.e. there is not other serial devices connected on the computer) it can be simply deinstalled with the following command from the *Terminal* application:

```
sudo apt-get remove modemmanager
```

However, if you want to keep the *Modem Manager*, it can be configured to avoid sending unnecessary

modem commands to the connected EEZ BB3 in the following way:

- ① Start *Terminal* application and enter the following command to determine which filter policy *Modem Manager* is using on your system by viewing its status:

```
sudo systemctl status ModemManager
```

- ModemManager.service - Modem Manager
   
Loaded: loaded (/lib/systemd/system/ModemManager.service...)
   
Active: active (running) since ...
   
...
   
CGroup: /system.slice/ModemManager.service
 └─644 /usr/sbin/ModemManager --filter-policy=strict

- ② *Modem Manager* filter policy has to be changed from *strict* to *default*. Use your favorite text editor like *gedit* to make that modification:

```
sudo gedit /lib/systemd/system/ModemManager.service
```

Additionally you can add under *[Service]* section the following line:

```
Environment="MM_FILTER_RULE_TTY_ACN_INTERFACE=0"
```

- ③ After changing the service file, reload the *systemctl* configuration and restart *Modem Manager*:

```
sudo systemctl daemon-reload
sudo systemctl restart ModemManager
```

Optionally you can check once again the *Modem Manager* filter policy that should be like this:

- ModemManager.service - Modem Manager
   
Loaded: loaded (/lib/systemd/system/ModemManager.service...)
   
Active: active (running) since ...
   
...
   
CGroup: /system.slice/ModemManager.service
 └─1010 /usr/sbin/ModemManager --filter-policy=default

## 15. MQTT

MQTT (*Message Queuing Telemetry Transport*) is a lightweight publish/subscribe messaging protocol designed for M2M (machine to machine) communication. MQTT is fast becoming one of the leading protocols for IoT (internet of things) deployments.



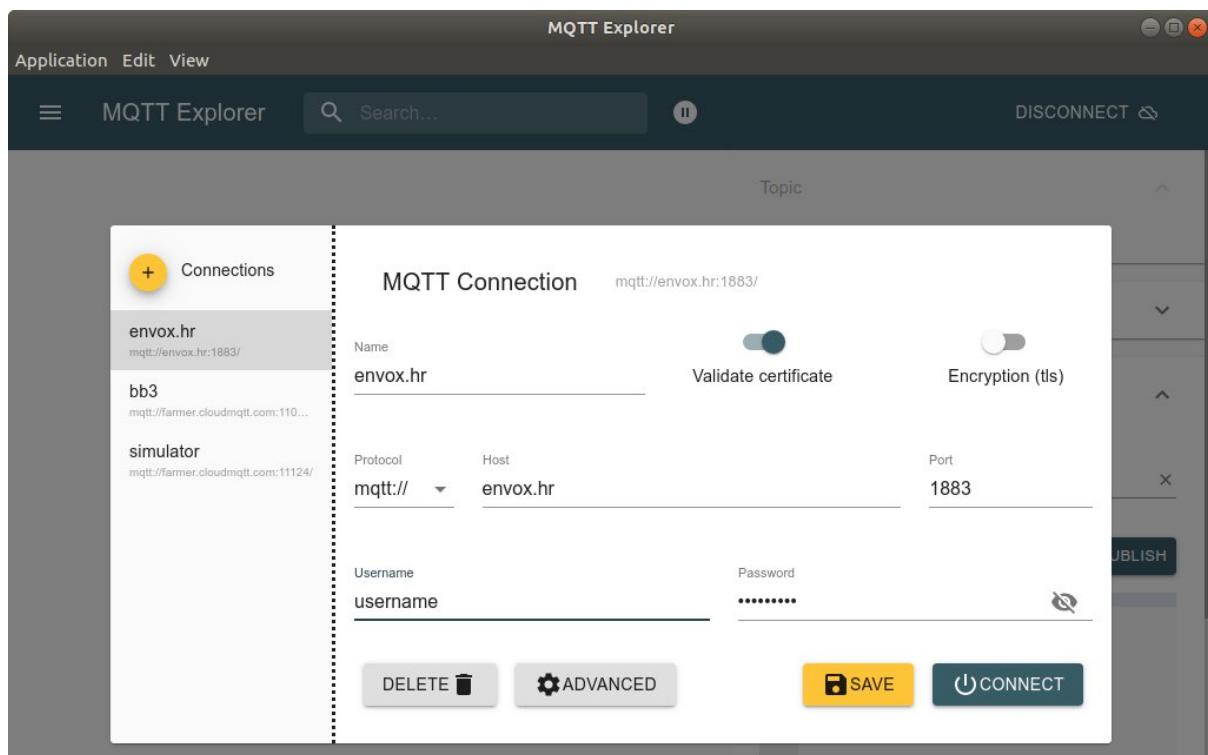
Thanks to MQTT support, EEZ BB3 can be easily and securely remotely monitored and controlled. Furthermore, EEZ BB3 can be integrated into existing IoT infrastructure.

For efficient and fast deployment, it is recommended to familiarize yourself with the basic terms of the MQTT protocol. An example of an interaction with EEZ BB3 via the MQTT description protocol is described below using the MQTT Explorer application which is available for free at <https://mqtt-explorer.com/>

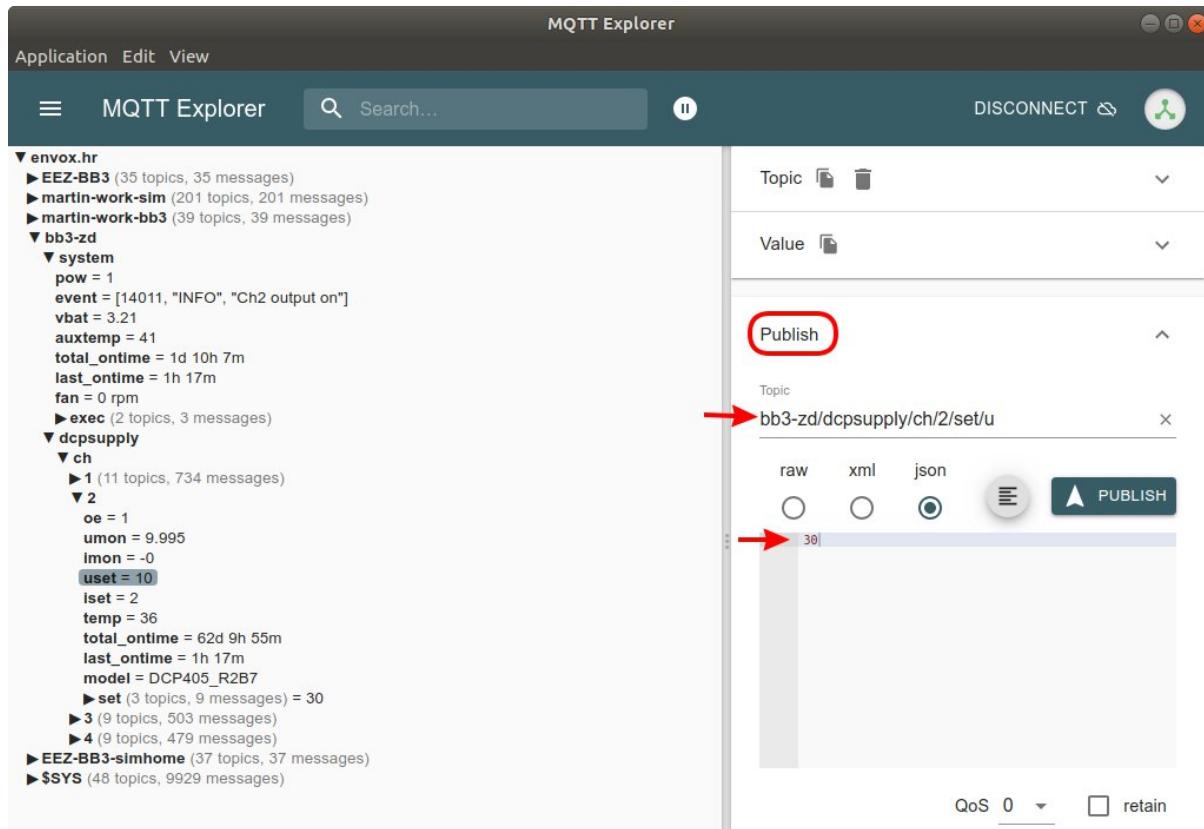
*Please note that topics are case-sensitive, i.e. <hostname>/system/exec/restart is not equal to <hostname>/system/exec/RESTART.*

### 15.1. Set channel parameter using subscribe topic

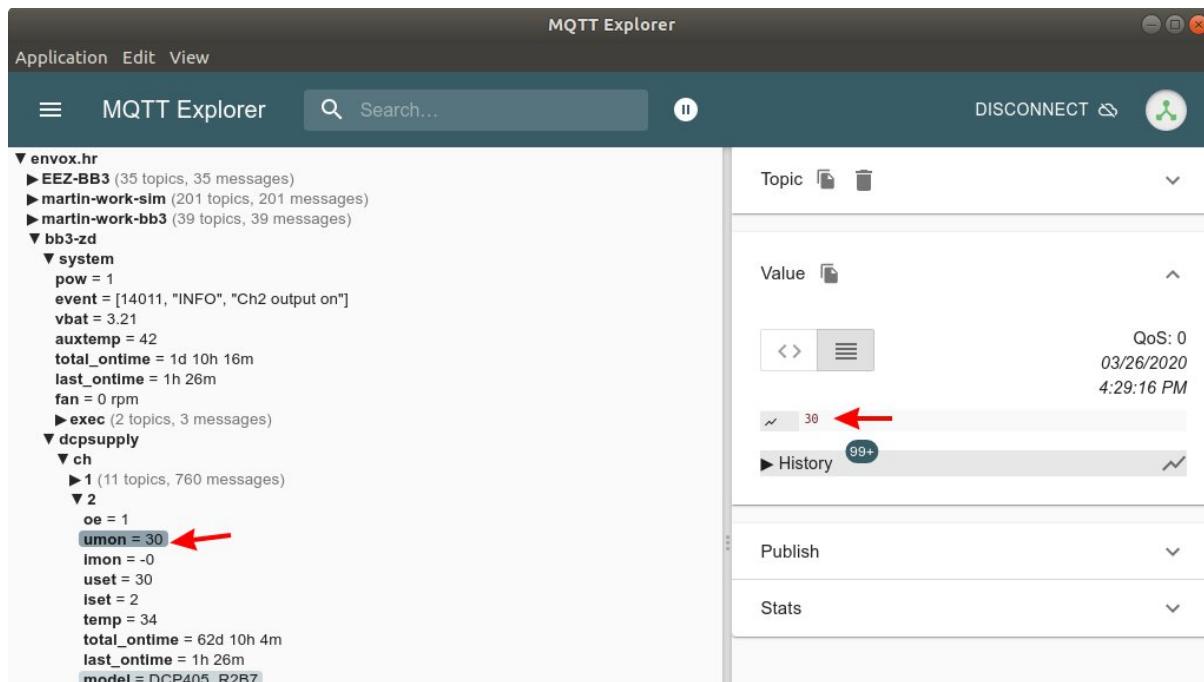
- ① Start *MQTT Explorer* and enter parameters for connection to your MQTT server (aka broker): *Name, Protocol, Host, Port, Username and Password*.



- ② For example, if we want to set the output voltage on a channel, it will be necessary to type in the Publish section as topic <hostname>/dcpsupply/ch/<ch>/set/u. In this example, the hostname is bb3-zd and to set the output voltage on channel 2 it will be necessary to type as topic bb3-zd/dcpsupply/ch/2/set/u, set its value to 30 (type is json) and click on the Publish button.

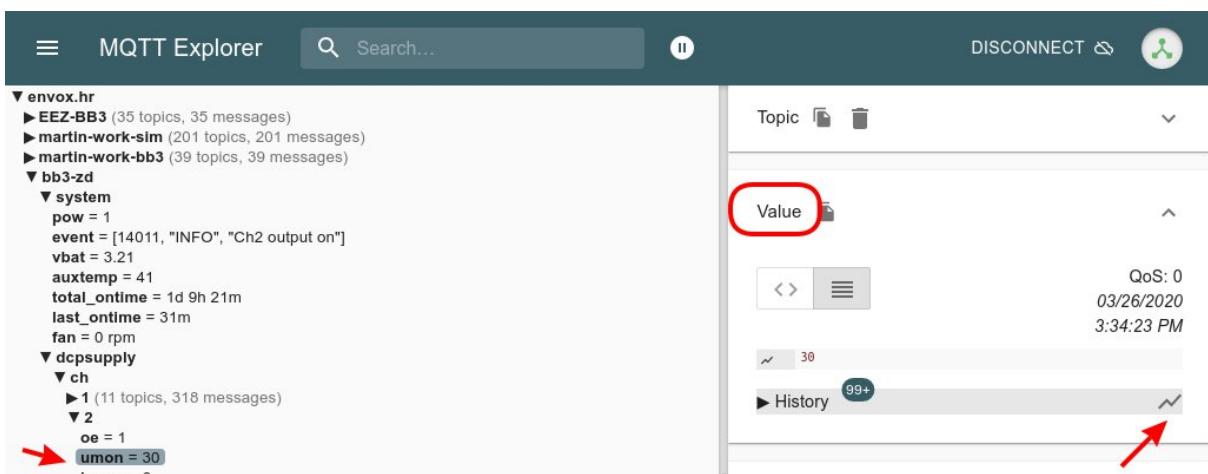


- ③ The newly set output voltage value will be visible after the topic *bb3-zd/dcpsupply/ch/2/umon* is refreshed.

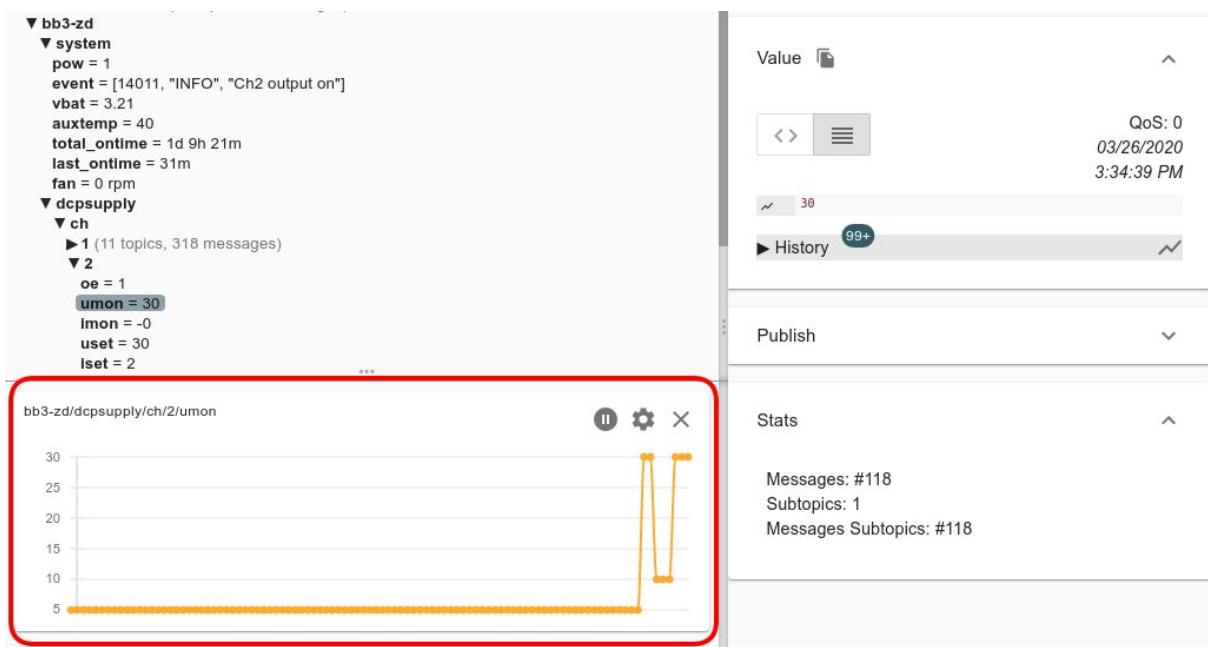


## 15.2. Drawing a topic graph

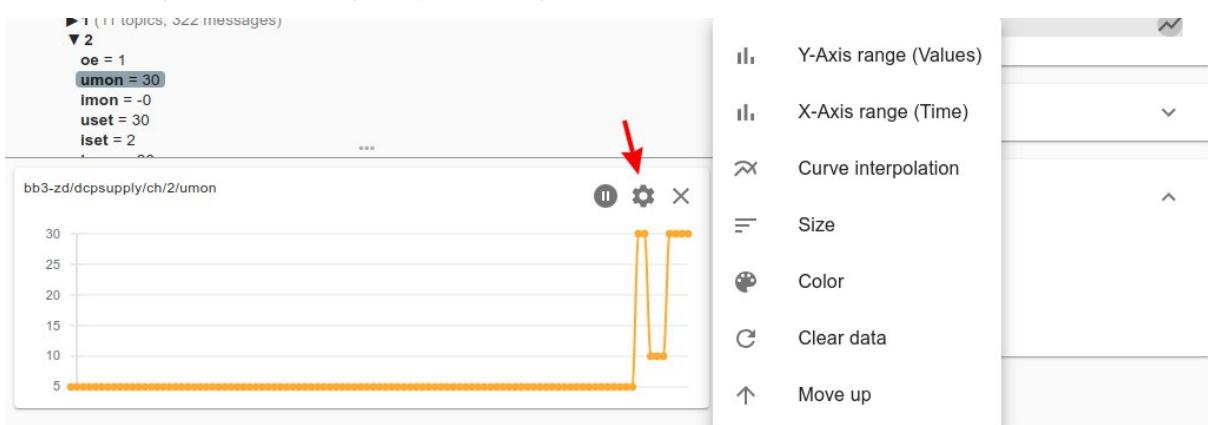
- ① Select a topic of interest, and click the graph icon in the *History* section



- ② The default settings graph will appear below the topic list.



- ③ Graph settings can be changed by selecting the *settings* icon.



### 15.3. Publish topics

#### 15.3.1. System

<b>Topic</b>	<b>Description</b>
<hostname>/system/pow	Returns 0 when EEZ BB3 enters standby mode, or 1 when in active mode
<hostname>/system/event	Latest message from the event log formatted as [<event_id>, <event_type>, <event_message>]. For example: [14011, "Info", "Ch2 output on"]
<hostname>/system/pow/battery	RTC battery voltage
<hostname>/system/pow/auxtemp	AUX sensor temperature
<hostname>/system/pow/fan	Cooling fan status and speed
<hostname>/system/pow/total_ontime	Total active time of the EEZ BB3 (MCU module). Resolution is 1 minute
<hostname>/system/pow/last_ontime	Duration since the EEZ BB3 was turned on. Resolution is 1 minute

<b>Topic</b>	<b>Description</b>

## 16. Scripting with MicroPython

One of the prominent features of EEZ BB3 is support for [MicroPython](#) scripting. Thanks to MicroPython scripting it will be possible to add new functionality without intervening in the EEZ BB3 firmware and compromising its performance. Scripting opens up the possibility of creating a whole range of new applications from tasks to automate the programming of individual peripheral modules, to useful utilities such as various parameter calculators that do not require working with modules at all, but can be useful and at hand.



MicroPython is a lean and efficient implementation of the [Python 3](#) programming language that includes a small subset of the Python standard library and is optimized to run on MCUs such as STM32 used in EEZ BB3. MicroPython aims to be as compatible with normal Python as possible to allow you to transfer code with ease from the desktop to a MCU or embedded system. For more information, visit the official [micropython.org](https://micropython.org) website.

This chapter will describe the procedure for creating a front-end GUI, deployment, and running a MicroPython script.

### 16.1. Distinctive features of EEZ BB3 MicroPython implementations

In order to avoid misunderstandings and wrong expectations, the specifics of the MicroPython implementation should be explained first.

The implementation of MicroPython often means that MicroPython will take control of all hardware resources and processes. This is not the case with EEZ BB3 and it was done on purpose. Execution of the MicroPython script takes place in a separate thread controlled by [FreeRTOS](#) and has a lower priority than the main thread that is in charge of all vital system functions. This ensures that a faulty script cannot compromise basic functionality and can be aborted from the main thread.

When the execution of the MicroPython script is complete, its thread goes to sleep until the next call.

Although the lack of direct access to hardware resources from the MicroPython script may seem like a serious limitation, access is still possible but indirectly by using a large number of implemented SCPI commands and queries that cover all important aspects of working with hardware resources of the EEZ BB3 chassis and installed peripheral modules.

For this reason, to work effectively with MicroPython scripts, it is recommended to familiarize yourself with the SCPI command set, which is described in the [EEZ BB3 SCPI reference guide](#).

The EEZ BB3 MicroPython implementation has another distinctive feature, and that is the ability to manage user interaction via a GUI in run time using custom made pages.

The creation of new GUI pages is made possible thanks to the EEZ Studio application, which is also used to create the entire EEZ BB3 GUI for the color TFT touch-screen display.

A step-by-step procedure on how to create a simple EEZ Studio project that the MicroPython script will be able to use in run time to interact with the user will be described below.

### 16.2. Sample MicroPython script

As an example of using MicroPython scripting, it will be shown how to create a user form that will appear on the display and through which it will be possible to change the voltage on the first channel of the available power module. The required procedure is divided into three sections: creating an EEZ Studio project, writing a MicroPython script and MicroPython script deployment and execution.

#### 16.2.1. Creating an EEZ Studio project

As a first step, make sure you have the latest version of EEZ Studio, which you can download at <https://github.com/eez-open/studio/releases> and install it on your computer.

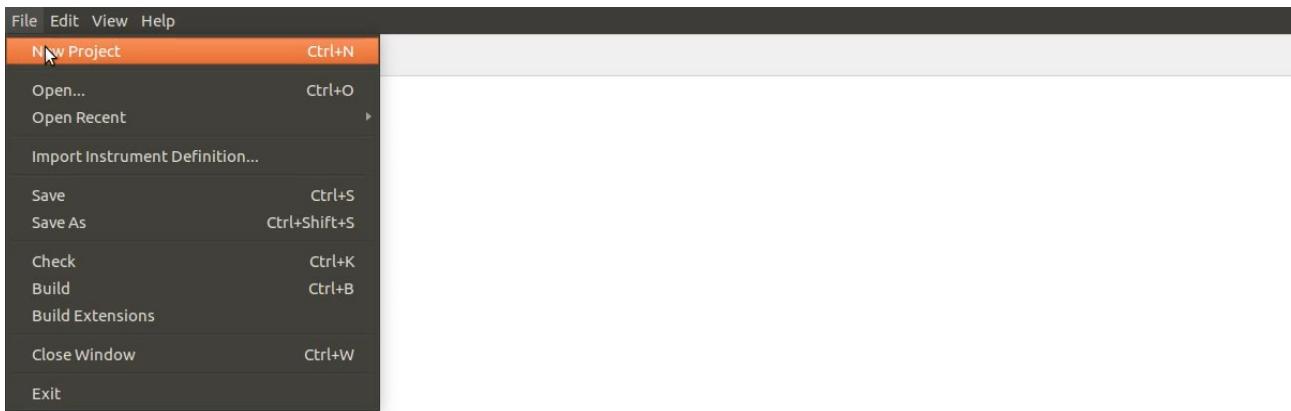


Fig. 14: Open new project

Start EEZ Studio and select the *New Project* option from the *File* menu (Fig. Error: Reference source not found). A new window will open displaying the Setting general page as shown in Fig. 15.

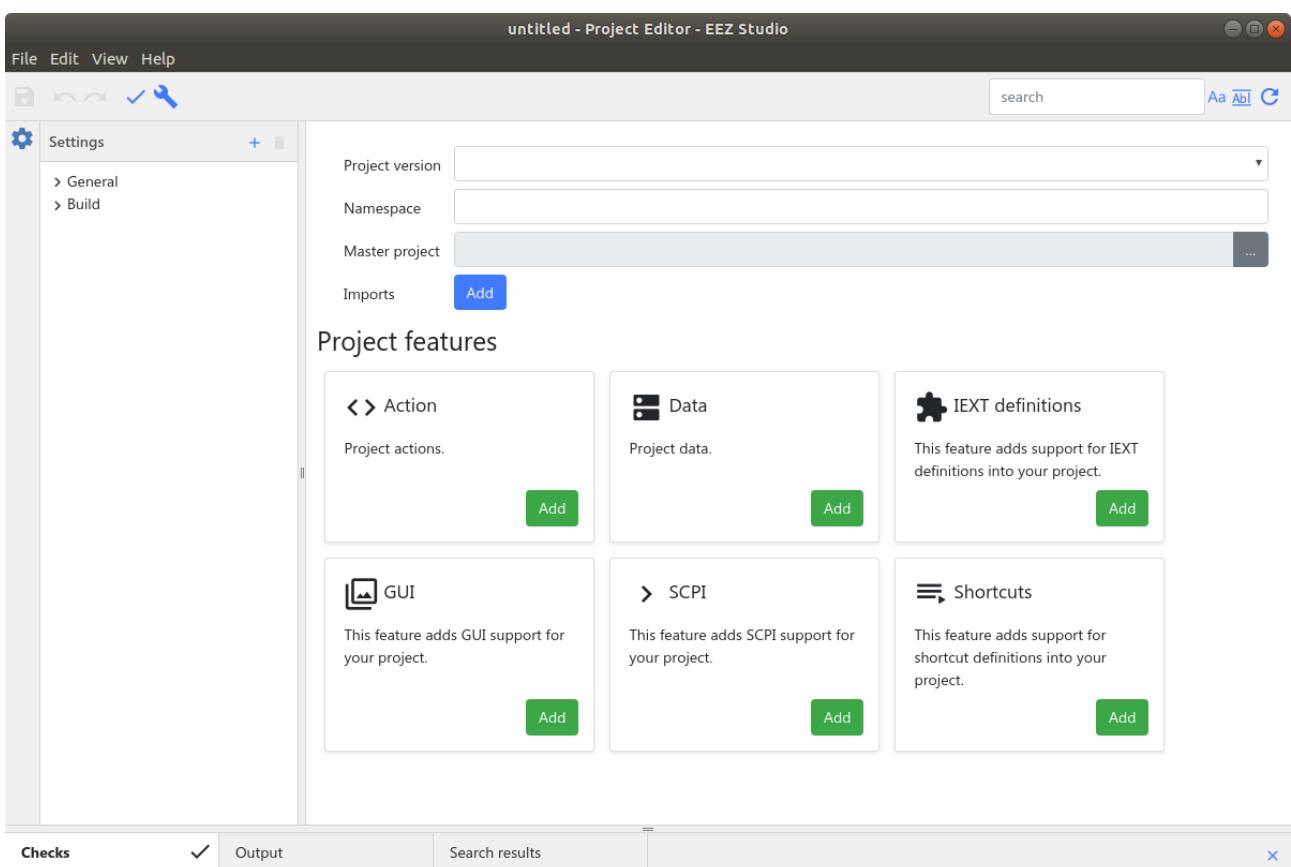


Fig. 15: Newly added project Settings general page

General settings contain the following parameters that will need to be defined:

- *Project version* – project version number
- *Namespace* – not currently used
- *Master project* – path to EEZ BB3 master firmware project
- *Project features* – determines which features will be used in the project.

Before you define the general parameters of the project, first save it to the desired destination. The extension of the EEZ Studio project is *.eez-project* and Linux users need to enter the name and extension when saving, in our case it will be *Hello World.eez-project*. When a project is saved its name will appear instead of *untitled* in the window header.

Now we can define the version of the project, which should be V2. Next, define the path to the master firmware of the project, i.e. to *modular-psu-firmware.eez-project* which can be found at the following link: <https://github.com/eez-open/modular-psu-firmware/blob/master/modular-psu-firmware.eez-project>.

The easiest way is to copy the master project to the folder where the saved and newly created project is located. Access to the master project is required in order to gain access to styles, color themes and fonts so that the page we will create is in line with other content on the screen.

It remains to define the *Project features* that will be used to interact via the TFT touch-screen display. Since such interaction is based on the event-driven principle it will be necessary to define the following: GUI layout, allowed actions and data to be exchanged during the action (event). Therefore we need to choose (using *Add* button) the following three features: *GUI*, *Action* and *Data*.

If we have well defined all the above parameters, the General settings page will look like in Fig. 16.

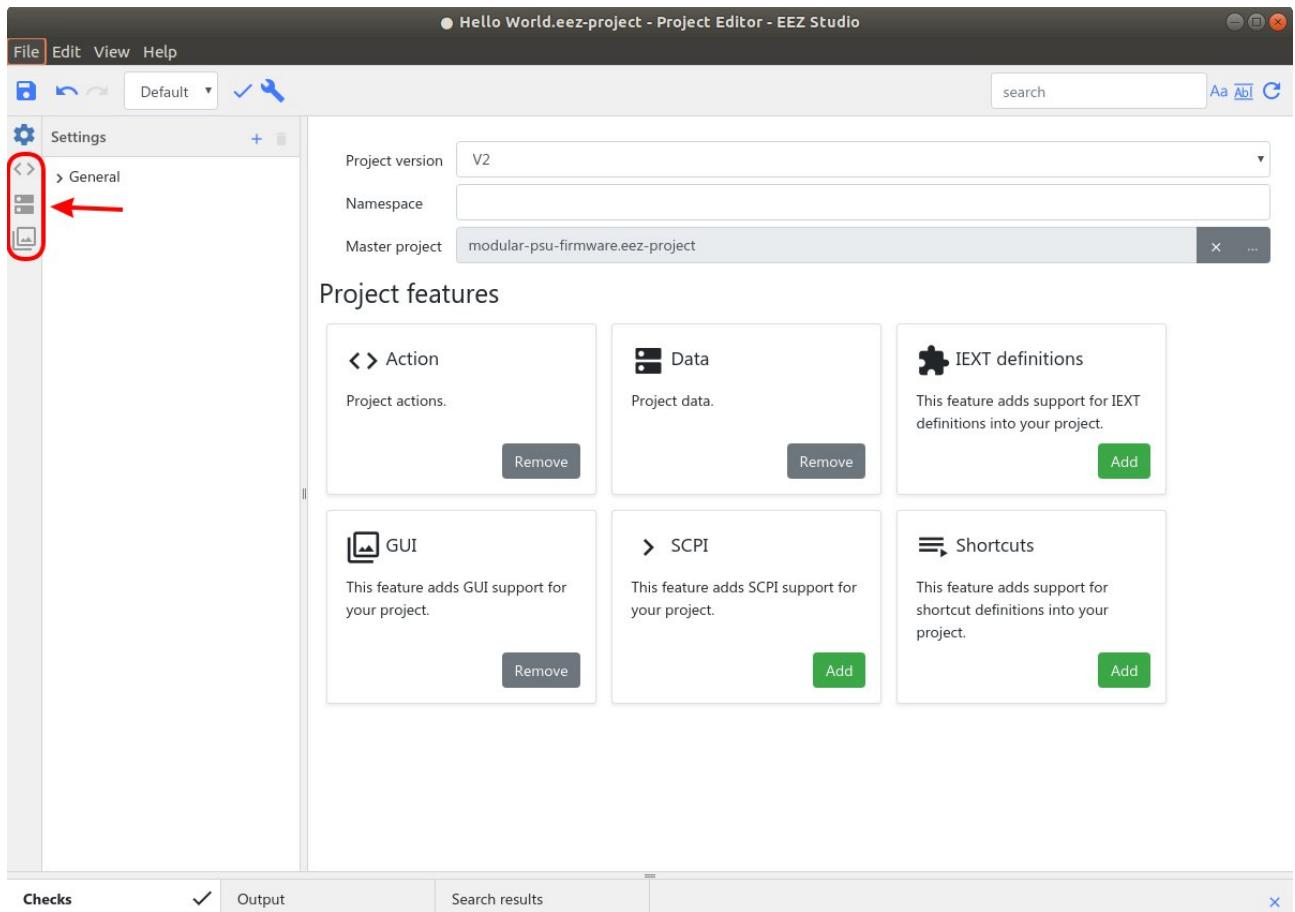


Fig. 16: General settings page with defined project parameters

Note that each of the selected *Project features* has added a new tab on the left side through which it will be possible to access the parameters of the selected feature.

To get the functional page that will be called from the MicroPython script it will be necessary to define the parameters via these three tabs that have appeared.

We will first define the names of the actions. For example, *input\_voltage* to enter the desired output voltage, *set\_voltage* to set the output voltage and *close* to close the page and stop execution of the MicroPython script. Use the + option to add all above mentioned actions by name, and leave default values of the other action parameters (i.e. *Description* and *Used in*). Once the action list is added it should look like in Fig. 17.

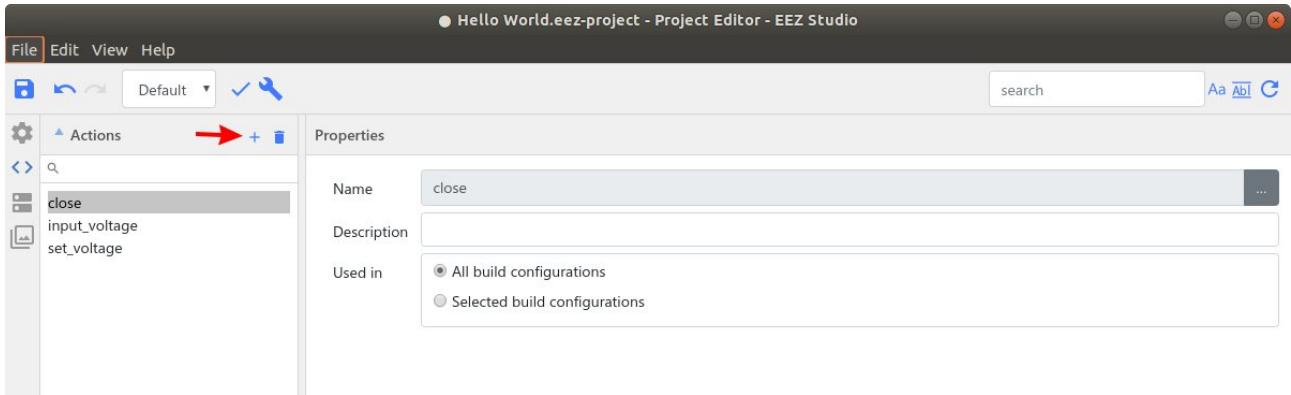


Fig. 17: List of defined actions

In the next tab we will define the data to be used for the actions. A minimum of three parameters will need to be defined for each data: name, type, and default value. These will be `can_set_voltage` as `Boolean`, default 0 (false) and voltage as `Float`, default 3.0 V. The default value will appear in the GUI item when it is assigned a specific data. Properties of both data are shown in Fig. 18.

Property	Value
Name	can_set_voltage
Type	Boolean
Default value	0
Used in	All build configurations

Property	Value
Name	voltage
Type	Float
Default value	3.0 V
Used in	All build configurations

Fig. 18: Defined project data

The GUI section is the most complex and before describing the creation of the page, it is necessary to explain what it contains. The section without any defined page is shown in Fig. 19.

This section is divided into two subsections: *Pages (Layouts)* and *Bitmaps* which have corresponding tab icons. In our example, we will not use bitmaps, so that subsection will not be described.

*This section can also contain other subsections such as Fonts and Styles, however in our example this is not applicable because we chose to inherit the fonts and styles from the master project.*

*Pages (Layouts)* consists of the following sections:

- *Pages (Layouts)* – a list of names of defined pages
- *Page structure* – the tree structure of all widgets used. It can be used to quickly move widgets within a structure
- *Page preview* – a central space that has no title, and will get a tab with the name of the page for each selected page. It displays selected page widgets and can be used to select and move one or more selected widgets
- *Properties* – displays all parameters of the selected widget
- *Widget palette* – a menu with all currently implemented widgets. In our example, we will use the two most commonly used widgets: *Container* and *Text*.
- *Themes* – color theme list. In our example they are inherited from the master project

The procedure for defining a page is as follows: add a new page, insert the necessary widgets, and assign actions and data to the widgets as needed. We will add a new page using the + option in the *Pages (Layouts)* section.

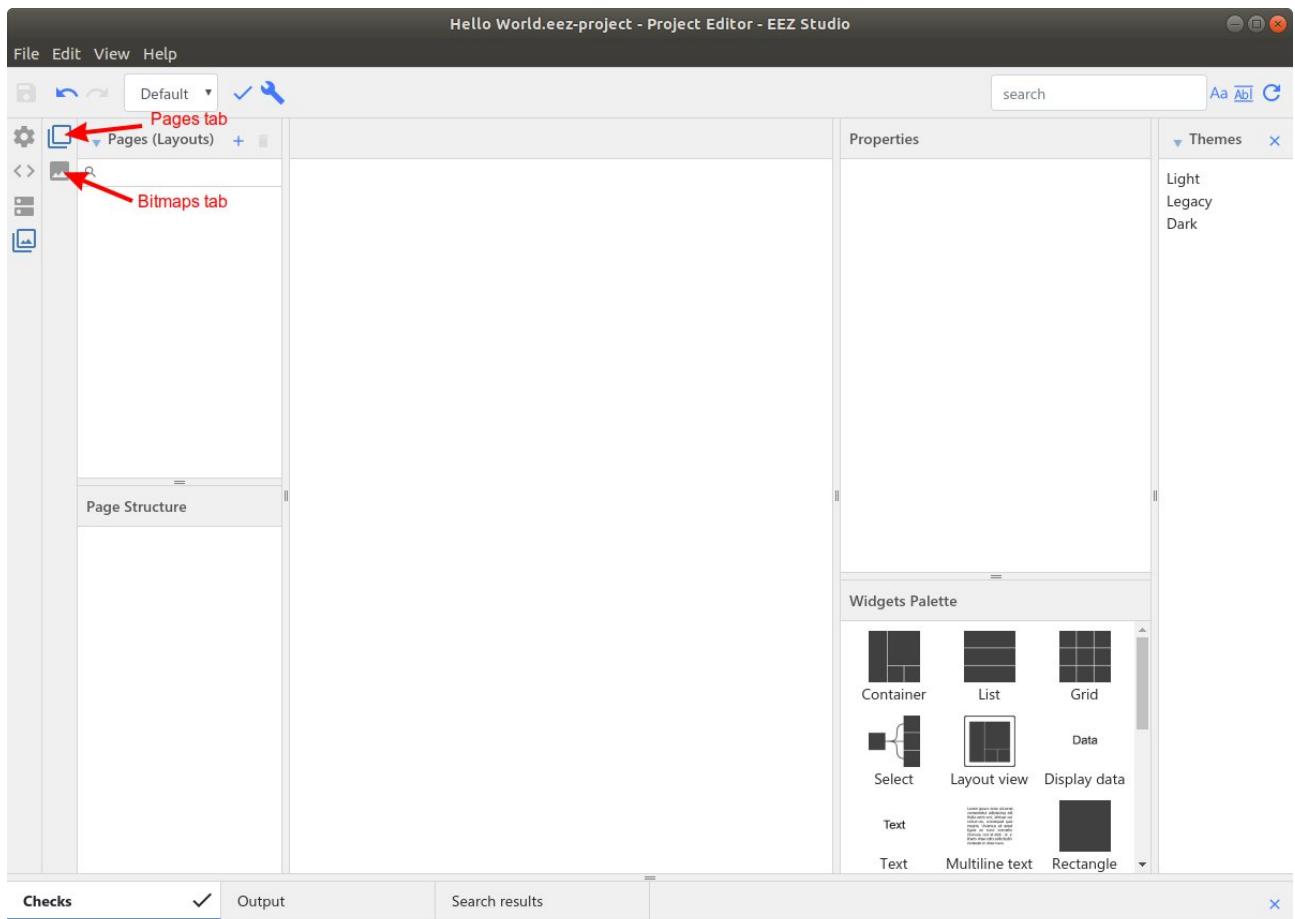


Fig. 19: Empty GUI feature section

The new items *Page* will appear and its *Properties* will be displayed. This will look like Fig. 20 (Checks and Themes sections are minimized for a simplified view). In *Properties* we can also see that the name is *Page* and we can change it to something else, say *Main* for which we will use [...] button at the right of the *Name* field.

We can now start inserting the widgets one by one. To do this, select the widget from the *Widgets palette* and drag & drop it to the central section (it may or may not be inside the page area since the final coordinates will be defined later).

As the first widget we will take one *Container* which, as its name suggests, is used to contain several

other widgets (including other containers) which simplify the organization of widgets and their manipulation.

The container may or may not have a defined name, and for its size and position we will use *Position* and *size* properties, so we will set:

- *Width – 204*
- *Height – 90*
- *Left – 138*, if we want a container of these dimensions to appear in the middle, but we can also enter here equation like  $(480 - 204) / 2$  since these fields accept basic mathematical operations (480 because it is the width of the page, 204 because it is the chosen width of the container and we divide by 2 to center it).
- *Top – 75*

*Note that the Pin to edge, Fix size and Preview options are intended for automatic positioning and have not yet been implemented.*

We will add a few widgets to this container. Adding a widget to an existing container can be done by dragging it in the desired container into the *Page structure* tree on the left. When positioning inside a tree structure, make sure that the purple marker is indented under the container to which you want the widget to belong (see Fig. 22).

*If no container is created, you can select one or more of existing widgets and insert it into the new container that will be created on that occasion. To do this, use the Put in container option from the right mouse button menu.*

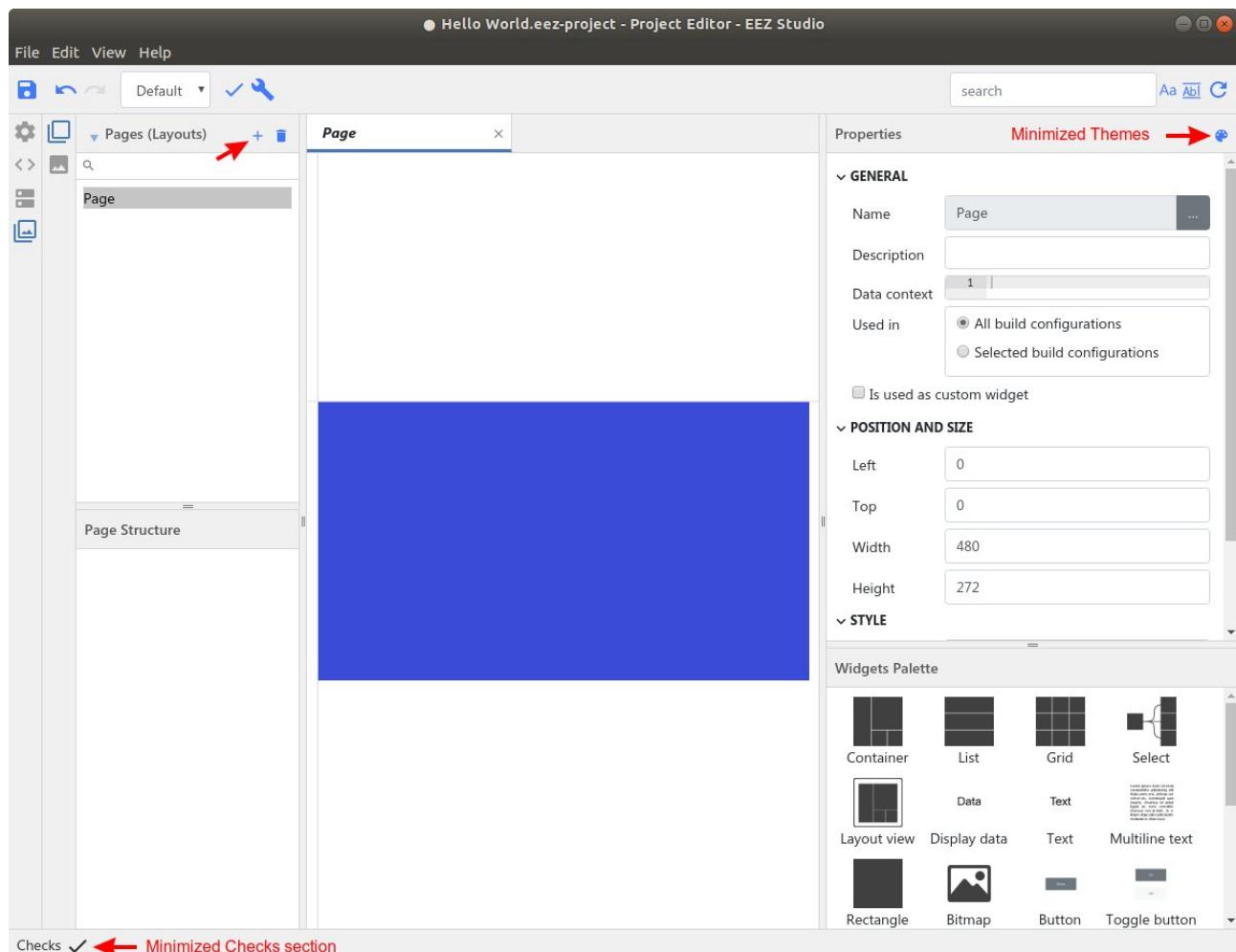


Fig. 20: Newly created GUI page

We will add the following widgets to the same container:

- *Text – Left: 0, Top: 0, Width: 84, Height: 40* (Absolute position will become 138, 75)
- *Text – Left: 84, Top: 0, Width: 120, Height: 40* (Absolute position will become 222, 75)

- **Button – Left: 84, Top: 50, Width: 120, Height: 40** (*Absolute position will become 222, 125*)

In addition, we will create another container that will define the status line in which the exit option will appear, for this we will need:

- **Container – Name: Status line, Left: 0, Top: 240, Width: 480, Height: 32** (*Absolute position will become 0, 240*)
- **Text – Left: 0, Top: 0, Width: 41, Height: 32** (*Absolute position will become 0, 240*)
- **Text – Left: 41, Top: 0, Width: 439, Height: 32** (*Absolute position will become 41, 240*)

In the next step, we can define additional widget properties: their *Style* (i.e. fonts and colors) and *Specific* properties. Since we have chosen to inherit styles from the master project, only the styles defined there as exportable (i.e. have a defined *Id* in properties) will be available when [...] button is selected to the right of the *Normal style* input field. A new window will then open with a list of all available styles as shown in Fig. 23.

Using this option for widgets from the first container we will set the following:

- **Text – Normal style: default\_M\_left, Text: Voltage:**
- **Text – Normal style: edit\_value\_active\_S\_center, Text: (change to blank)**. The *Text* property can be left filled with default value (i.e. Text), but in that case the default value of the data that we will associate with that widget (3.0 V) will not be displayed on the page.
- **Button – Normal style: button\_M, Disabled style: button\_M\_disabled, Text: Set**

For the widgets from the 2nd container we will set the following:

- **Text – Normal style: status\_icon\_enabled, Text: E** (please note that this style uses an icon font so the letter E is used because the Exit icon corresponds to that position)
- **Text – Normal style: status\_title, Text: Hello World**

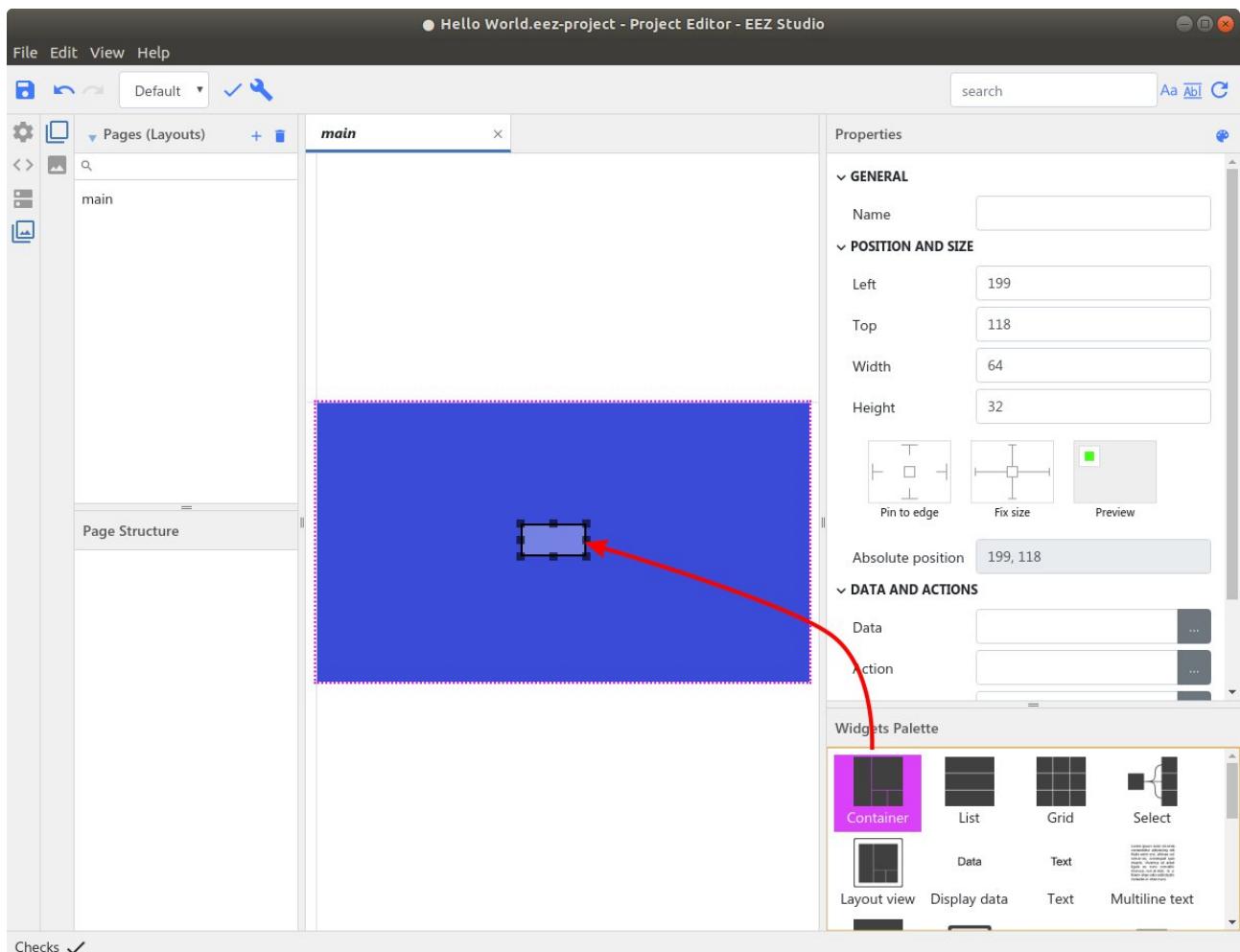


Fig. 21: Adding container widget

If you have done well so far you should have a fully defined page as shown in Fig. 24. It remains to set the action and data on certain widgets which we will do in the next step.

First container:

- *Voltage*: text widget is informative, and no action is expected on it. Therefore, its *Data* and *Action* fields should be left blank.
- The second text widget will be used to display the existing voltage and set the new voltage value at the CH1 output. For this reason, we will choose **voltage** for *Data*, and **input\_voltage** for the *Action*, which we defined at the very beginning. To select *Data* and *Action*, we will use the corresponding [...] options to the right of the *Data and actions* properties input fields.
- We will use the *Set* button widget to confirm the entry and execute the part of the MicroPython script that will set the entered value. We leave the *Data* blank. As *Action* we choose **set\_voltage**. This widget has another additional action property and that is *Enabled*. If there is a criterion during the execution of the MicroPython script that the button is enabled, we will be able to touch it, for which we will select the **can\_set\_voltage** from list of actions.

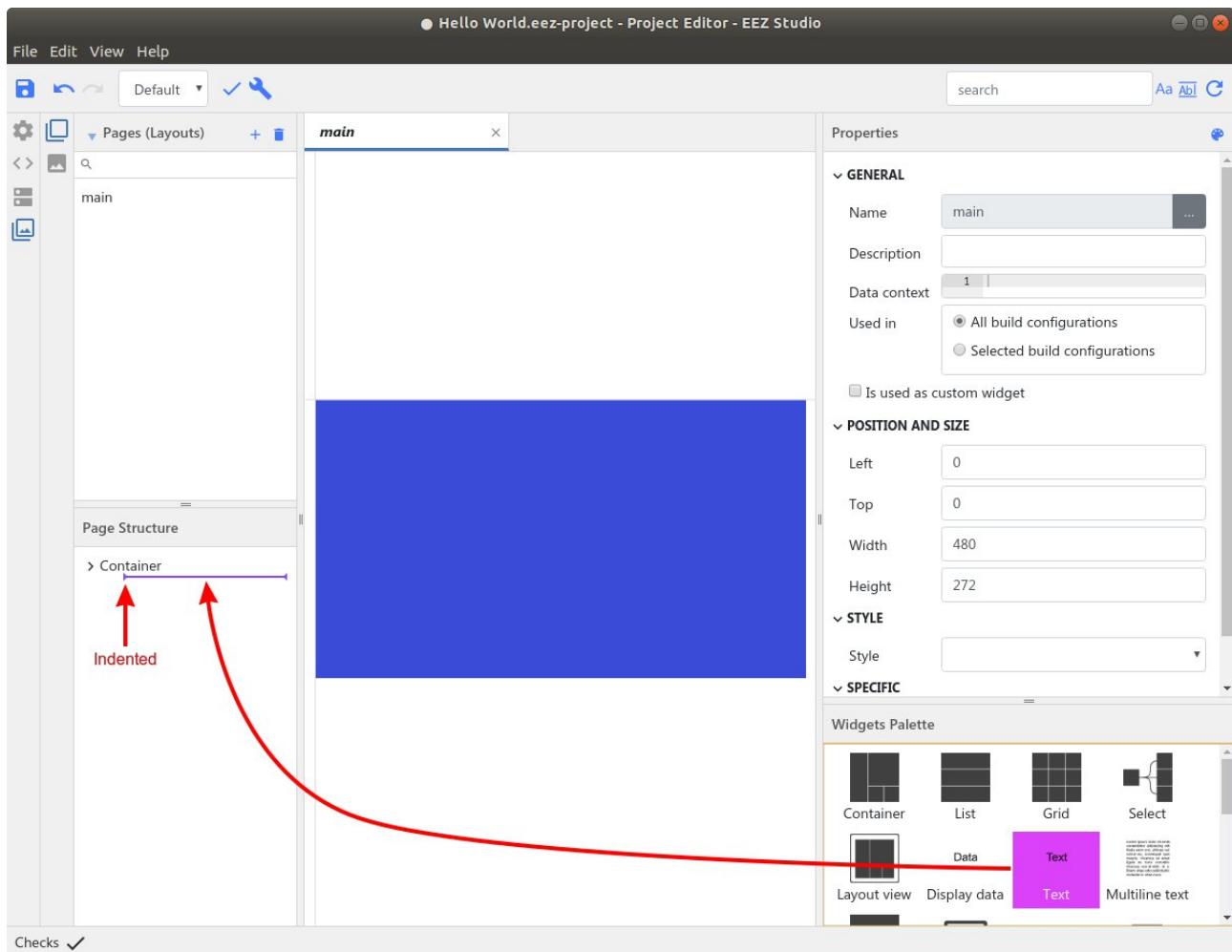


Fig. 22: Inserting new widget into container

Second container:

- The *E* text widget will be used to return from this page to the main page, which also means stopping further execution of the MicroPython script so the output voltage will not be set to the new value. For this reason, we will select here **close** as *Action*. *Data* field should be left blank.
- *Hello world* text widget is informative, and no action is expected on it. Therefore, its *Data* and *Action* fields should be left blank.

This completes the creation of the page to be used to interact with the MicroPython script. All we have to do is write a MicroPython script and deployment to EEZ BB3 which will be described below.

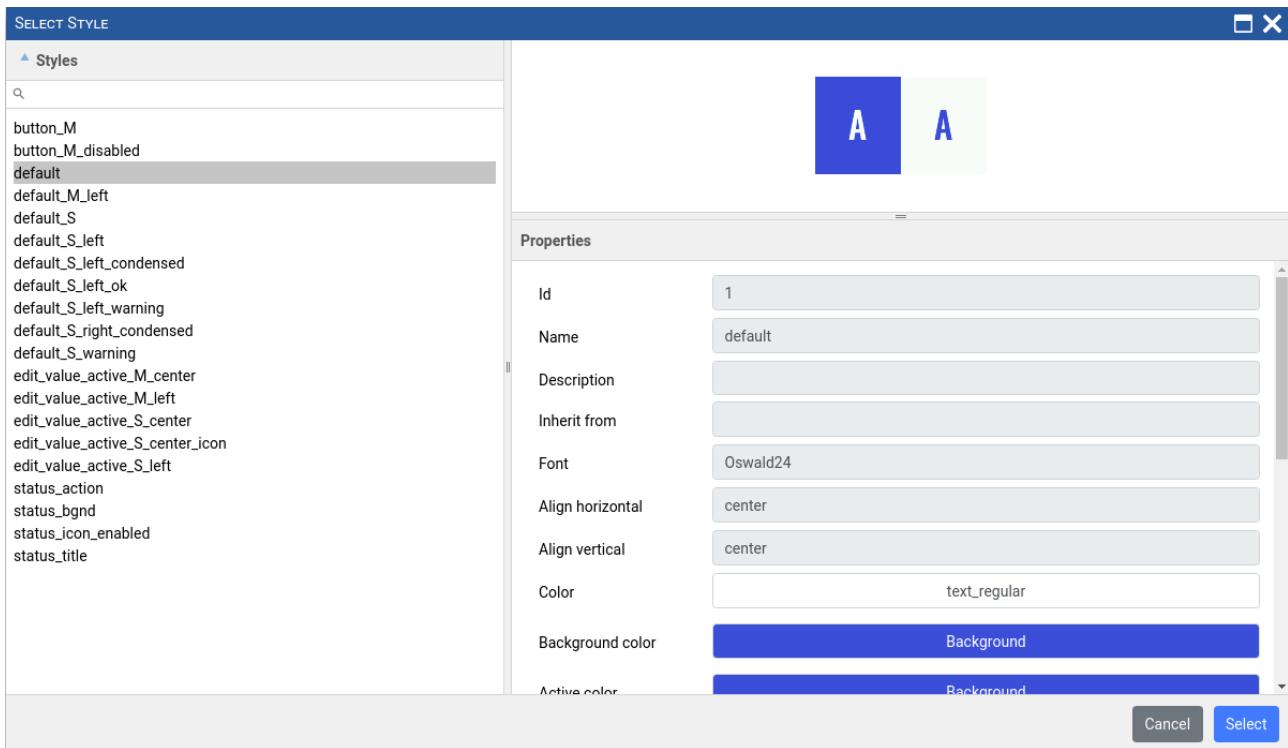


Fig. 23: Inherited Styles selection

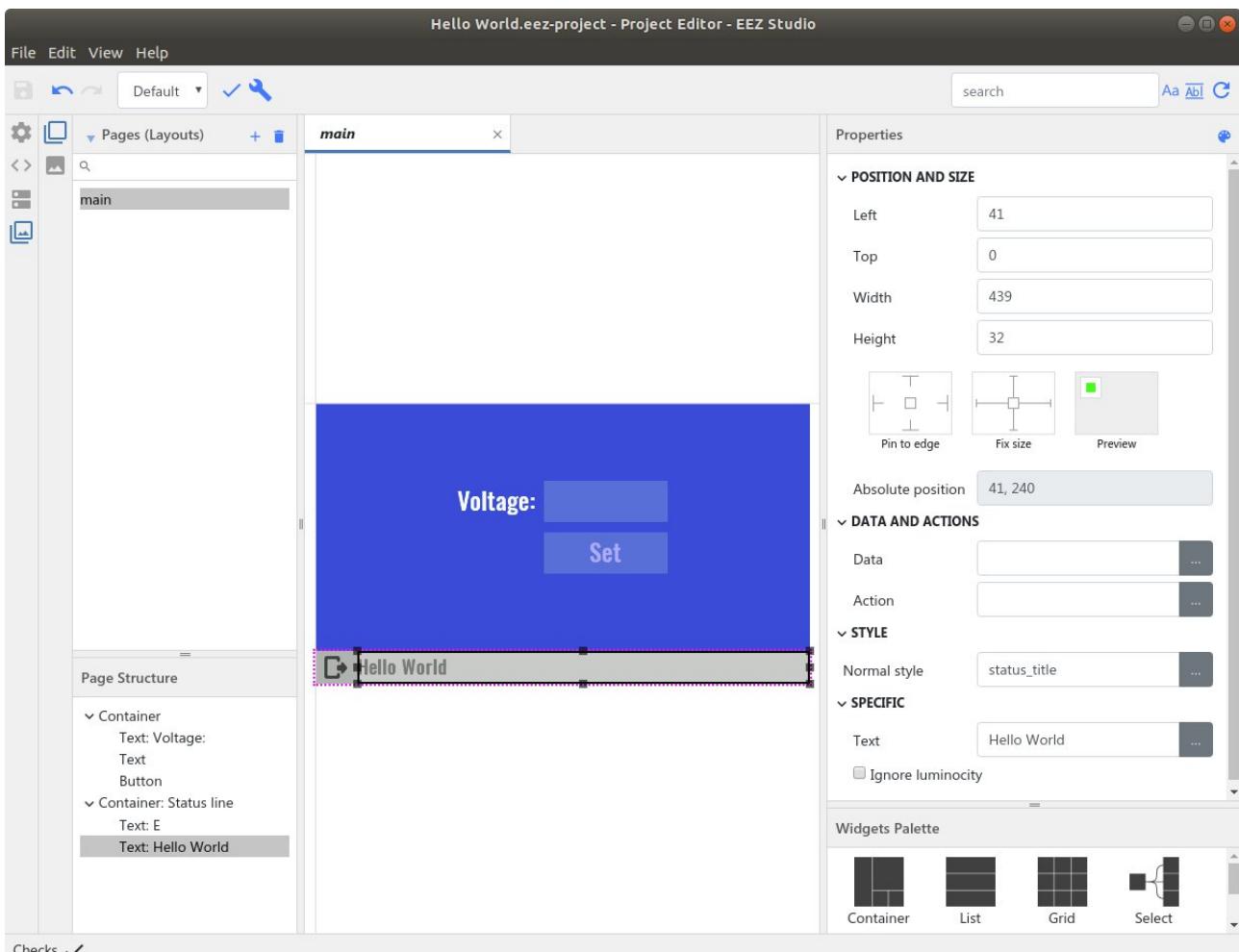


Fig. 24: Page view with all widgets and defined styles

## 16.2.2. MicroPython script

EEZ Studio does not currently offer the creation and editing of text files. Therefore, you can use your favorite text editor to write the MicroPython script listed below. The name of the MicroPython file should be *Hello world.py* to match the name of the EEZ Studio project.

Code	Explanation
# Hello, World!	The text that will appear as a description of the MicroPython script when displayed on EEZ BB3 (see <a href="#">File manager</a> section).
from eez import scpi	Include <i>scpi</i> function from <i>eez</i> module
def input_voltage():     global voltage, max_voltage     value = scpi('DISP:INPUT? "",NUMBER,VOLT,0.0,' + str(max_voltage) + '.0,' + str(voltage))	This is the implementation of the <i>input_voltage</i> action defined in the EEZ studio project. When one click on the <i>voltage</i> widget this function will be launched.
if value != None:     voltage = float(value)     scpi('DISP:DIALog:DATA "voltage",FLOAT,VOLT,' + str(voltage))     scpi('DISP:DIALog:DATA "can_set_voltage",INT,1')	The SCPI command <a href="#">DIAG:INPUT?</a> opens the input dialog on the screen where voltage can be entered.
def set_voltage():     scpi("INST ch1")     scpi("VOLT " + str(voltage))	If <i>voltage</i> is entered then two data items are set: <i>voltage</i> to the entered value and <i>can_set_voltage</i> to 1 which means that voltage is entered and can be set on the channel, which results in the <i>Set</i> button widget being enabled.
def main():     global voltage, max_voltage     scpi("INST ch1")     voltage = scpi("VOLT?")     max_voltage = scpi("VOLT? MAX")	Select CH1 as set its output voltage.
scpi("DISP:DIAL:OPEN \"Scripts/Hello World.res\"")	Main program loop. SCPI command <a href="#">INST</a> is used to select CH1 on power module and two SCPI queries that return currenty set voltage ( <a href="#">VOLT?</a> ) and max allowed voltage ( <a href="#">VOLT? MAX</a> ).
try:     scpi('DISP:DIAL:DATA "voltage",FLOAT,VOLT,' + str(voltage))	Invoke <i>Hello World.res</i> file using the <a href="#">DISP:DIAL:OPEN</a> that contains page created in EEZ Studio as discussed above.
while True:         action = scpi("DISP:DIALog:ACTION?")         if action == "input_voltage":             input_voltage()         elif action == "set_voltage":             set_voltage()             break         elif action == "close" or action == 0:             break	This is a dispatcher that asks via the SCPI command <a href="#">DISP:DIALOG:ACTION?</a> which action the GUI requires to be performed.
	The dispatcher is executed in a loop: when it receives from the GUI which action to perform, it executes it and asks again what the next action is.
	If the detected action is <i>close</i> or 0 (this means that the firmware has requested closing the dialog) then the dispatcher loop is

```

finally:
    scpi("DISP:DIAL:CLOS")

main()

```

broken and goes to *finally* section described below.

Execute SCPI command **DISP:DIAL:CLOS** to close last dialog window opened with the DISP:DIAL:OPEN command. This section will be also executed in case of an error, i.e. exception.

### 16.2.3. MicroPython script deployment and execution

Once we are done with the EEZ Studio project and the MicroPython script we are ready for deployment. In order for the page created in EEZ Studio to be used successfully on EEZ BB3, it must not contain any errors and should be "compiled". The *Check* and *Build* options are used for this, and Fig. 25. shows the result of the *Build* action in the *Output* tab. It will create in the same folder where the EEZ Studio project resource file *Hello world.res*.

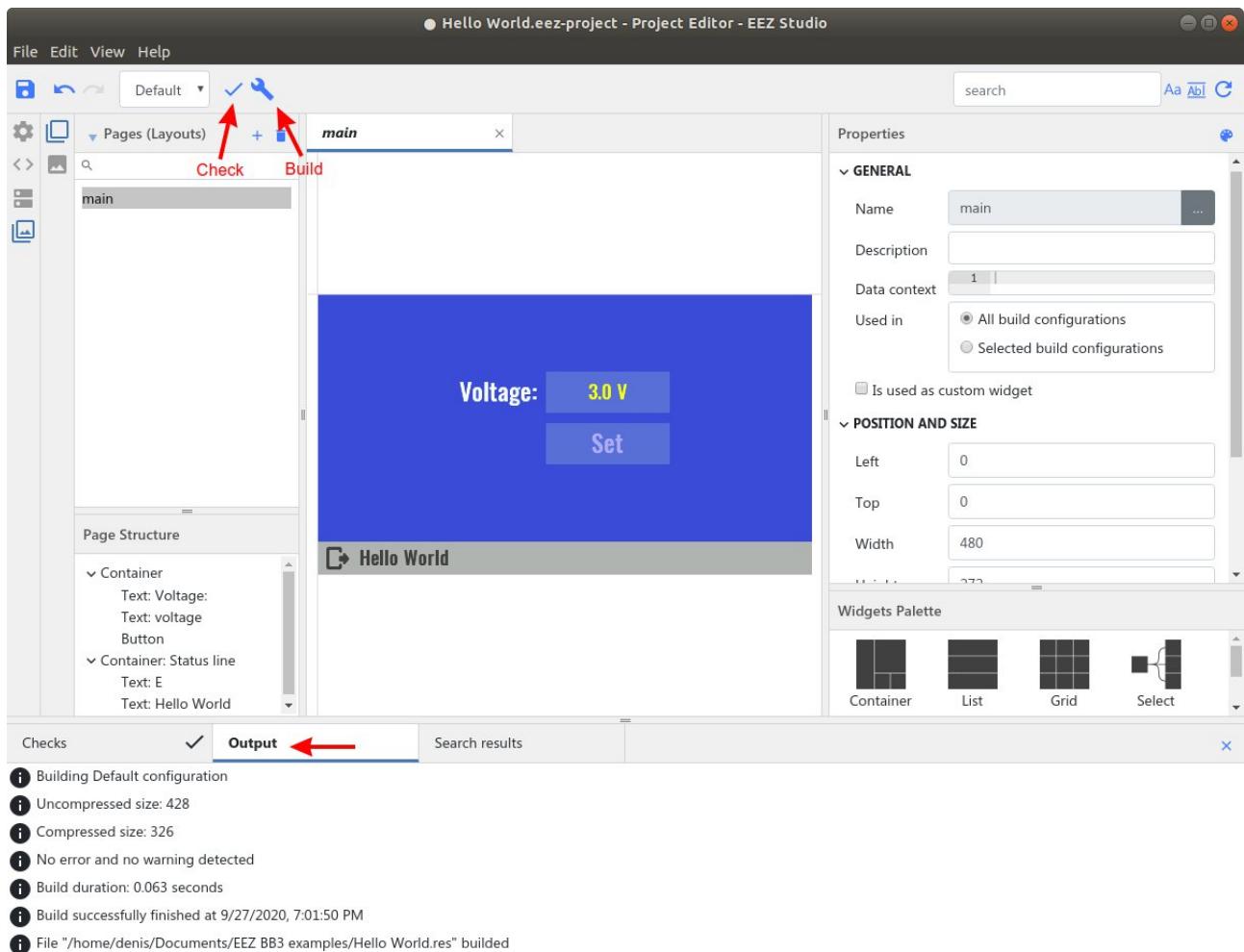


Fig. 25: EEZ Studio project Check and Build options

*Hello world.py* and *Hello world.res* files will now need to be downloaded to EEZ BB3 in the *Scripts* folder. We can also use EEZ Studio for this, but its ESW part used to communicate with SCPI instruments such as EEZ BB3. The procedure is the same as in the case of the [Upgrade peripheral module firmware using GUI on BB3](#) section in Chapter 13. The download parameters of the MicroPython script are shown in Fig. 27. The same will need to be done for the resource file (*Hello world.res* ).

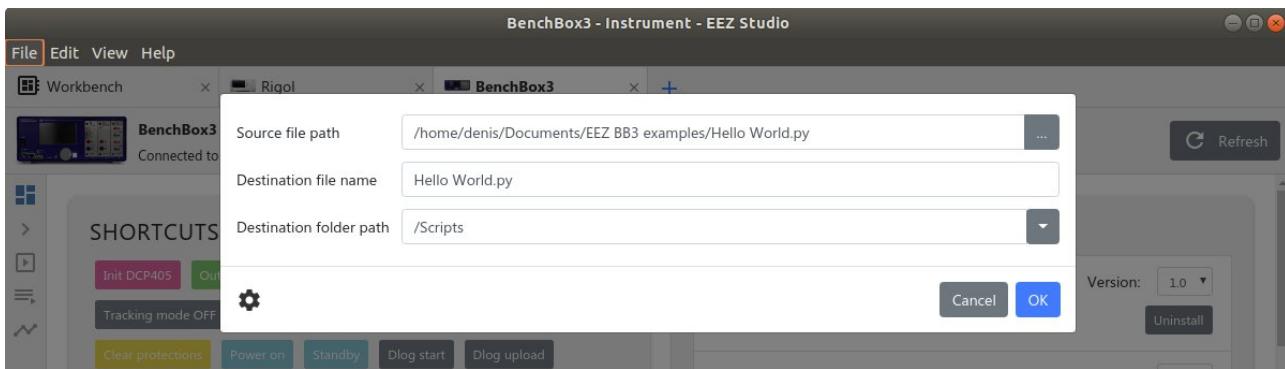


Fig. 26: File transfer using EEZ Studio Send File option

During execution, if the MicroPython script has an error, a message will appear as in Fig. 14. The show debug trace log option leads to the event log viewer where more details about the error can be found (Fig. 21).

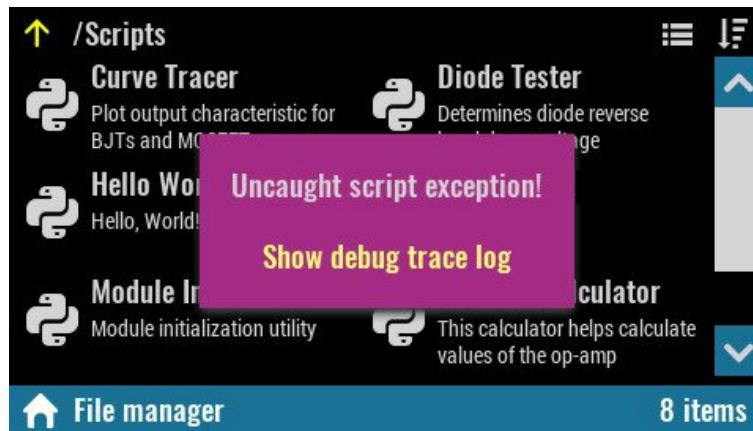


Fig. 27: MicroPython script error message

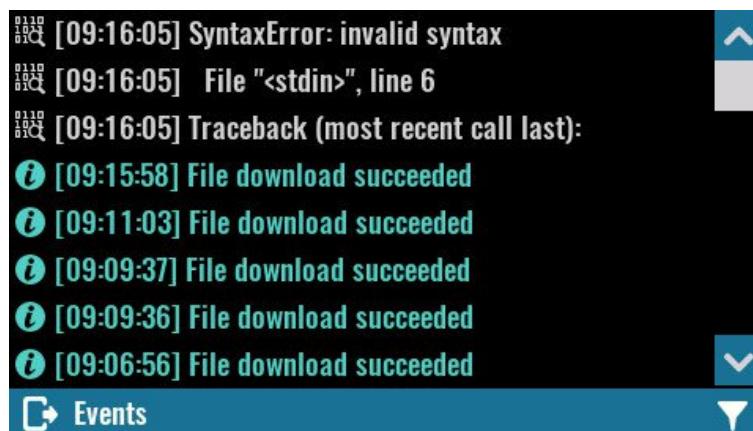


Fig. 28: MicroPython script error details in event log

The current implementation does not offer a script debugger or the ability to modify the script on the EEZ BB3 side. During development, all MicroPython script modifications will need to be made to the computer side and the changes will need to be downloaded as previously shown.

### 16.3. EEZ Python module

In our example the MicroPython script used eez module. The table below contains the names and descriptions of all the functions that the module contains. The current version of the module can be found at:

[https://github.com/eez-open/modular-psu-firmware/blob/master/src/third\\_party/micropython/ports/bb3/mod/eez/README.md](https://github.com/eez-open/modular-psu-firmware/blob/master/src/third_party/micropython/ports/bb3/mod/eez/README.md)

For MicroPython references visit: <http://docs.micropython.org/en/latest/reference/index.html>

A list of currently implemented / enabled functions in EEZ BB3 firmware can be found at this link:

[https://github.com/eez-open/modular-psu-firmware/blob/master/src/third\\_party/micropython/ports/bb3/mpconfigport.h](https://github.com/eez-open/modular-psu-firmware/blob/master/src/third_party/micropython/ports/bb3/mpconfigport.h)

Function name	Description
<code>eez.scpi(commandOrQuery)</code>	Execute any SCPI command or query. If command is executed, then None is returned. If query is executed, then it returns query result as integer or string.
<code>eez.getU(channelIndex)</code>	Returns measured voltage as float for the given channel index.
	This is the same as <a href="#">MEASure[:SCALAR]:VOLTage[:DC]?</a> query. Use this function instead of SCPI query when performance requirement is critical.
<code>eez.setU(channelIndex, currentLevel)</code>	This function sets the immediate voltage level for the given channel index.
	It is equal to <a href="#">[SOURce[:n]]:VOLTage[:LEVel] [:IMMEDIATE][:AMPLitude]</a> command. Use this function instead of SCPI command when performance requirement is critical.
<code>eez.getI(channelIndex)</code>	Returns measured current as float for the given channel index.
	This is the same as <a href="#">MEASure[:SCALAR]:CURRent[:DC]?</a> query. Use this function instead of SCPI query when performance requirement is critical.
<code>eez.setI(channelIndex, currentLevel)</code>	This function sets the immediate current level for the given channel index.
	This is the same as <a href="#">[SOURce[:n]]:CURREnt[:LEVel] [:IMMEDIATE][:AMPLitude]</a> command. Use this function instead of SCPI command when performance requirement is critical.
<code>eez.getOutputMode(channelIndex)</code>	For the given channel index, returns: <ul style="list-style-type: none"> <li>• "CV" when channel is in Constant Voltage mode</li> <li>• "CC" when channel is in Constant Current mode</li> <li>• "UR" when channel is neither in Constant Voltage or Constant Current mode</li> </ul>
	This is the same as <a href="#">OUTPut:MODE?</a> query. Use this function instead of SCPI query when performance requirement is critical.
<code>eez.dlogTraceData(value, ...)</code>	For current DLOG trace file, this function adds one point in time for each defined Y-axis. It expects one or more value arguments depending of how much Y-axis values are defined for currently started DLOG trace.
	This is the same as <a href="#">SENSe:DLOG:TRACE[:DATA]</a> command. Use this function instead of SCPI command when performance requirement is critical.

## 16.4. MicroPython script examples

EEZ Studio communicates with SCPI instruments using the so-called IEXT (*Instrument EXTension*). IEXT for EEZ BB3 includes, among other things, several MicroPython scripts that can be easily transferred to EEZ BB3, as well as later updated if newer versions of EEZ BB3 IEXT include changes to Mi-

croPython scripts. EEZ Studio projects and MicroPython scripts can be found at <https://github.com/eez-open/modular-psu-firmware/tree/master/scripts>

<b>Script name</b>	<b>Description</b>
<i>Curve Tracer</i>	Plot output characteristic for BJTs and MOSFETs
<i>Diode Tester</i>	Determines diode reverse breakdown voltage
<i>Hello World</i>	The script used in the example previously described in this chapter
<i>Module Initialization</i>	EEZ peripheral module initialization utility
<i>Op-Amp Calculator</i>	This calculator helps calculate values of the op-amp configured as inverting, non-inverting or differential amplifier
<i>Parallel and Series Calculator</i>	Calculates parallel resistance/inductance or capacitance in series
<i>Voltage Divider Calculator</i>	This calculator helps calculate values of resistive voltage divider



# DECLARATION OF CONFORMITY (IZJAVA O SUKLADNOSTI)



We, (Mi)

ENVOX d.o.o.  
Zagreb, Croatia

declare under our sole responsibility that the product  
(izjavljujemo pod punom odgovornošću da je proizvod)

Type, Model: (Tip, Model):	EEZ BB3 STM32F7
Brand Name: (Komercijalni naziv):	EEZ Bench Box 3
Product: (Proizvod):	Modular test and measurement chassis Modularna šasija za testne i mjerne module
Product Options: (Varijante proizvoda):	This declaration covers all options of the above product(s) with DCP405 and DCM224 power modules (Ova deklaracija uključuje sve varijante gore navedenog proizvoda s DCP450 i DCM224 perifernim modulima.)

is in conformity with the essential requirements of Directives and regulations

(sukladan bitnim zahtjevima Direktiva i propisa):

Electromagnetic compatibility Directive ( 2014/30/EU ), LVD Directive (2014/35/EU), ROHS Directive (2011/65/EU), Ecodesign Directive (2009/125/EC), Regulation (EC) 278/2009

Pravilnik o elektromagnetskoj kompatibilnosti (NN br.28/2016), Pravilnik o električnoj opremi namijenjenoj za uporabu unutar određenih naponskih granica (NN br.43/2016), Pravilnik o ograničavanju uporabe određenih opasnih tvari u električnoj i elektroničkoj opremi (NN br.131/2013, 16/2014, 90/2014, 142/14), Pravilnik o utvrđivanju zahtjeva za ekološki dizajn proizvoda povezanih s energijom (NN br. 80/2013), Uredba (EK) 278/2009

by applying the following standards (primjenom sljedećih normi):

- EN 61326-1:2013 (HRN EN 61326-1:2013) Emission limits-class A according to CISPR11, Immunity values according to table 2 except BURST test that comply with Criteria C
- EN IEC 61000-3-2:2019 (EN IEC 61000-3-2:2019)
- EN 61000-3-3:2013+A1:2019 (HRN EN 61000-3-3:2013)
- EN 61010-1:2010 (HRN EN 61010-1:2010)
- EN 62311:2008

## Supplementary Information:

This product is intended for use in a basic electromagnetic environment.

This is a sensitive measurement apparatus by design and may have some performance loss when exposed to ambient continuous electromagnetic phenomenon.

The CE marking is affixed from the year (CE oznaka stavljena na proizvod): 2021

Zagreb, Croatia  
23.03.2021.

ooo  
envox d.o.o. za  
proizvodnju  
i testiranje  
Fallerova Šetalište  
120f, Zagreb

Denis Kotlar



envox

### **RoHS Compliance Statement**

Envox d.o.o. certifies that products supplied are in compliance with, and conform to the European Union's Restriction of Use of Hazardous Substances in Electrical and Electronic Equipment (RoHS) Directive 2002/95/EC and 2011/65/EU (RoHS2). Said products have no intentional addition of:

- Lead (Pb)
- Cadmium (Cd)
- Mercury (Hg)
- Hexavalent Chromium (Cr)
- Polybrominated Biphenyls (PBB)
- Polybrominated Diphenyl Ethers (PBDE)

Any trace impurities of these substances are below the threshold limits as specified by the RoHS directive; specifically Cr+6, Hg, Pb, PBB, PBDE do not exceed 1000 ppm (0.1%) and Cd does not exceed 100 ppm (0.01%).

This declaration is based upon the information provided by the material suppliers and assembly manufacturers used by Envox to manufacture these products. As such, Envox makes no independent representations or warranties, expressed or implied, and assumes no liability in connection with the use of this information.

This certification is valid unless superseded by a revised certification at a later date.

March, 22 2021

---

For more info visit: [www.envox.hr/eez](http://www.envox.hr/eez)  
File repository: <https://github.com/eez-open>

Version: 1.05  
Date: 2021-04-01