



EEZ Studio User manual

*Low-code embedded GUI development tool
T&M automation and management*

Ver. M14 DRAFT – 05/2023
www.envox.eu
github.com/eez-open

Table of Contents

| | |
|---|----|
| 1. Legal information..... | 5 |
| 1.1. Definitions..... | 5 |
| 1.2. Disclaimers..... | 5 |
| 1.3. Miscellaneous..... | 6 |
| 1.4. Contact information..... | 6 |
| 1.5. Revision history..... | 6 |
| 2. The EEZ Studio overview..... | 7 |
| 2.1. Introduction..... | 7 |
| 2.2. Main sections..... | 7 |
| 2.3. Known issues and issue reporting..... | 7 |
| 2.4. Donations..... | 7 |
| 3. Installation..... | 9 |
| 3.1. System requirements..... | 9 |
| 3.2. Linux..... | 9 |
| 3.3. Mac..... | 9 |
| 3.4. Windows..... | 9 |
| 3.5. Nix package manager..... | 10 |
| 3.6. Build and run from source (all operating systems)..... | 10 |
| 3.6.1. Linux only..... | 10 |
| 3.6.2. Raspbian only..... | 10 |
| 3.6.3. All platforms..... | 10 |
| 3.6.4. On Raspbian:..... | 10 |
| 3.6.5. Nix..... | 10 |
| 3.7. USB TMC..... | 10 |
| 3.7.1. Windows..... | 11 |
| 3.7.2. Linux..... | 11 |
| 3.8. FAQ..... | 11 |
| 4. Key features..... | 13 |
| 4.1. General..... | 13 |
| 4.2. EEZ Studio Project..... | 13 |
| 4.3. EEZ Studio Instrument..... | 13 |
| 5. Menu options and Settings..... | 15 |
| 5.1. Home page..... | 15 |
| 5.2. Menu options..... | 15 |
| 5.2.1. File..... | 15 |
| 5.2.2. Edit..... | 16 |
| 5.2.3. View..... | 16 |
| 5.2.4. Help..... | 17 |
| 6. Home page instrument sections..... | 19 |
| 6.1. History..... | 19 |
| 6.2. Shortcuts and Groups..... | 19 |
| 6.3. Notebooks..... | 20 |
| 6.4. Items purge and restore..... | 24 |
| 6.5. Instrument Extension (IEXT) Manager..... | 25 |
| 6.6. Add instrument..... | 26 |
| 6.7. Establishing a connection with the instrument..... | 28 |
| 7. Instrument activity bar..... | 31 |
| 7.1. Start page (EEZ BB3 only)..... | 31 |
| 7.2. Dashboard..... | 33 |

| | |
|--|----|
| 7.3. Terminal..... | 33 |
| 7.4. Scripts..... | 34 |
| 7.4.1. Edit script shortcut..... | 35 |
| 7.5. Shortcuts..... | 36 |
| 7.6. Lists..... | 37 |
| 7.6.1. Editing a list using a table..... | 38 |
| 7.6.2. Editing a list using an envelope..... | 39 |
| 7.6.3. List view options..... | 41 |
| 7.6.4. List help..... | 41 |
| 8. Home page project sections..... | 43 |
| 8.1. EEZ Studio project types..... | 43 |
| 8.2. Create new project..... | 44 |
| 8.3. Additional steps for creating EEZ BB3 projects..... | 45 |

1. Legal information

1.1. Definitions

Draft – A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. Envox does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

1.2. Disclaimers

Limited warranty and liability – Information in this document is believed to be accurate and reliable. However, Envox does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. Envox takes no responsibility for the content in this document if provided by an information source outside of Envox.

In no event shall Envox be liable for any indirect, incidental, punitive, special or consequential damages (including – without limitation – lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, Envox' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of Envox.

Right to make changes – Envox reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use – Envox products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an Envox product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Envox and its suppliers accept no liability for inclusion and/or use of Envox products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications – Applications that are described herein for any of these products are for illustrative purposes only. Envox makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using Envox products, and Envox accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the Envox product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Envox does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using Envox products in order to avoid a default of the applications and the products or of the application or use by customer's 3rd party customer(s). Envox does not accept any liability in this respect.

Suitability for use in non-automotive qualified products – Unless this data sheet expressly states that this specific Envox product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. Envox accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without Envox' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer

uses the product for automotive applications beyond Envox' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies Envox for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond Envox' standard warranty and Envox' product specifications.

Security – Customer understands that all Envox products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their life cycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by Envox products for use in customer's applications. Envox accepts no liability for any vulnerability. Customer should regularly check security updates from Envox and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by Envox.

To report a security issue, use the EEZ Studio [issue tracker](#).

1.3. Miscellaneous

Open source license and contributions – EEZ Studio uses the *GPL v3* license. To view a copy of this license, please visit <https://www.gnu.org/licenses/gpl-3.0.html>. EEZ Studio uses the [C4.1 \(Collective Code Construction Contract\)](#) process for contributions.

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](#).

Terms and conditions of commercial sale – Envox products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.envox.eu/company/terms-of-use/>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. Envox hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of Envox products by customer.

Translations – A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Export control – This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Trademarks – All referenced brands, product names, service names, and trademarks are the property of their respective owners.

1.4. Contact information

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.eu

Website: www.envox.eu

1.5. Revision history

| Date | Version | Changes |
|------------|---------|-----------------|
| xxxx-xx-xx | 1.0 | Initial release |

2. The EEZ Studio overview

2.1. Introduction

EEZ Studio was initially developed as a companion application for the in-house developed [EEZ H24005](#) programmable power supply and [EEZ BB3](#) T&M chassis to address two important tasks: a) remote programming and management and b) simplifying the development of a feature rich embedded GUI for a color touch-screen display.

The development was inspired by the idea of offering an open source alternative to some existing commercial solutions that are used for the mentioned tasks, all in order to overcome the limitations of their closed code, outdated and complex UI or sometimes awkward UX and licensing, which in our case was not in accordance with the open source of the mentioned devices that we have developed.

2.2. Main sections

EEZ Studio consists of two main sections, which are described separately in the manual:

- **Project** – creating, editing, debugging and building the code for the embedded GUI project for the selected target platform. Generated code can be directly imported into the IDE/toolchain used to build the firmware and accelerate the development process. It enables the rapid development of high quality embedded GUI and also comes with support for the open-source LVGL graphics library. The drag-and-drop editor makes it easy to utilize the many features such as widgets, animations, and styles to create a GUI reducing the coding effort. Additionally flowchart-like *EEZ flow* programming feature will further save development time and complexity.
- **Instrument** – allows access to one or more T&M instruments using several communication interfaces through which it is possible to manage and collect measurement data and screenshots using SCPI commands and queries. Collected data can be analyzed, searched, annotated and exported to other applications. Automation of test and measurement tasks using JavaScript and *EEZ flow* programming allows it to be used in different scenarios from basic development, calibration, troubleshooting and quality control using multiple devices from different manufacturers that can be in different locations connected to LANs.

In the introductory chapters of the two main sections that follow, all important features will be listed and described in detail.

2.3. Known issues and issue reporting

EEZ Studio is continuously developing and improving. A list of known issues can be found on [GitHub](#) where you are also invited to leave your suggestions for improvements and new functionality.

When reporting bugs using the GitHub tracking system, please first check if the issue you want to report has already been reported by someone else. When opening a new ticket, the following information can simplify and speed up the resolution:

- Descriptive/detail name of the issue (avoid general descriptions)
- Installed operating system version
- Installed EEZ Studio version
- Steps to reproduce the problem you are reporting

2.4. Donations

As an open source project, EEZ Studio has been largely developed thanks to donations primarily from [NLnet Foundation](#) as well as a number of smaller individual donors. If you want to contribute to further development with your donation, you can use [Liberapay](#).

3. Installation

3.1. System requirements

EEZ Studio is a 64-bit application. Therefore the minimum requirement for installation is a personal computer with a 64-bit operating system installed which has enough RAM and disk space for smooth operation.

Installation packages for supported operating systems for all versions of EEZ Studio are available for download at <https://github.com/eez-open/studio>

It is the official download page and we recommend that you get the latest version for the first installation. You will be able to check for future updates by using the option provided for that, as described below. If EEZ Studio becomes available on the websites of our partners, this information will be published on the Envox official website.

3.2. Linux

Depending on your linux distribution, choose one of the listed packages (.deb, .rpm) and start the installation using the associated installer.

In addition, there is a self-executing .AppImage version that, after downloading, needs to enable the Allow executing file as program option under file Permissions (Fig. 1) before starting it.

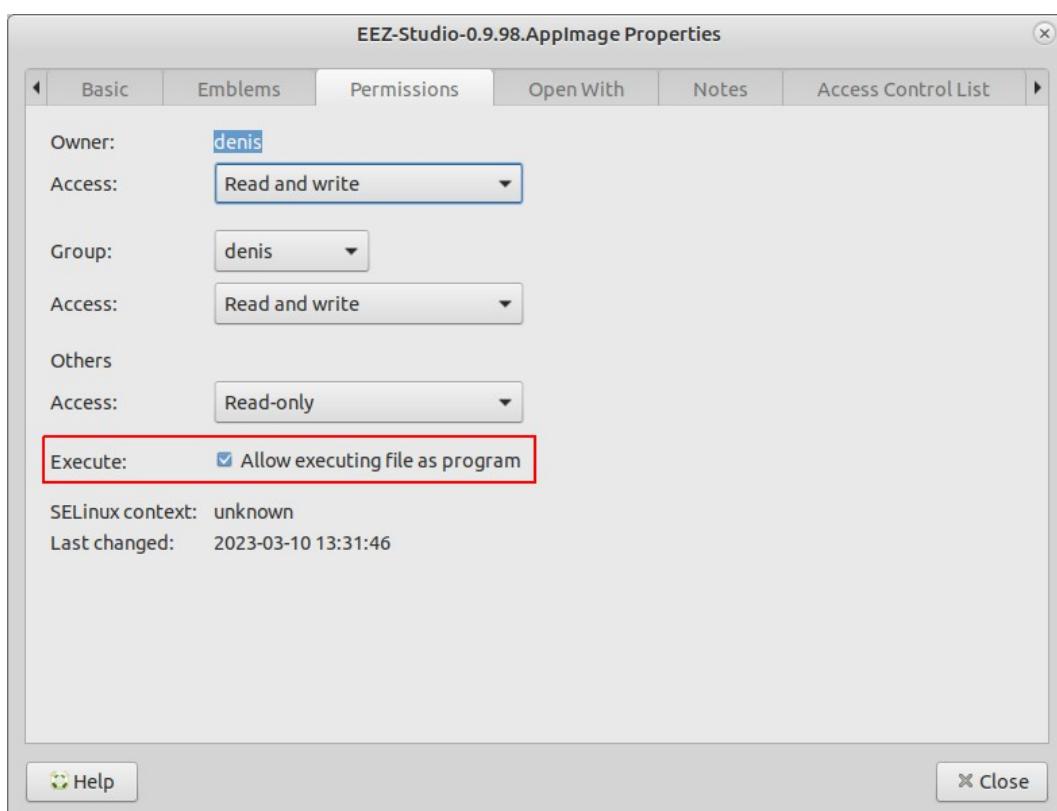


Fig. 1: .AppImage file permission

If you encounter a problem running the .AppImage version on your Linux distribution, try running it using the --no-sandbox option: ./EEZ-Studio-[version].AppImage --no-sandbox

3.3. Mac

Required OS version: macOS 10.10 (Yosemite) or newer

Download *eezstudio-mac.zip*, unpack and move *eezstudio.app* to Applications.

3.4. Windows

Required OS version: Windows 7 (64-bit) or newer

Download and start *EEZ_Studio_setup.exe*.

3.5. Nix package manager

The Nix [flake](#) provides a derivation for EEZ Studio or an overlay that provides that derivation. It can be used to install the project using [Nix package manager](#).

3.6. Build and run from source (all operating systems)

In addition to using ready-made installation packages, it is possible to build and run EEZ Studio directly from the source code located in the GitHub repository. Below is the procedure to be followed:

- Install Node.js 14.x or newer
- Install node-gyp, more information at <https://github.com/nodejs/node-gyp#installation>

3.6.1. Linux only

```
sudo apt-get install build-essential libudev-dev
```

3.6.2. Raspbian only

Install Node.js 16 and npm on Raspberry Pi: <https://linode.com/install-node-js-and-npm-on-raspberry-pi/>

```
sudo apt-get install build-essential libudev-dev libopenjp2-tools ruby-full  
sudo gem install fpm
```

3.6.3. All platforms

In the folder where you want to build the project, it will be necessary to clone the GitHub project repository, and start project building as follows:

```
git clone https://github.com/eez-open/studio  
cd studio  
npm install  
npm run build
```

Start with:

```
npm start
```

Create distribution packages (except [Raspbian](#)):

```
npm run dist
```

3.6.4. On Raspbian:

```
npm run dist-raspbian
```

3.6.5. Nix

To build:

```
nix build 'github:eez-open/studio'
```

To start:

```
nix run 'github:eez-open/studio'
```

3.7. USB TMC

The USB TMC driver must be installed if you want to access the T&M instrument using the USB-TMC interface from EEZ Studio *Instrument* section.

3.7.1. Windows

Download and start [Zadig](#). Select your device, select libusb-win32 and press “Replace Driver” button:

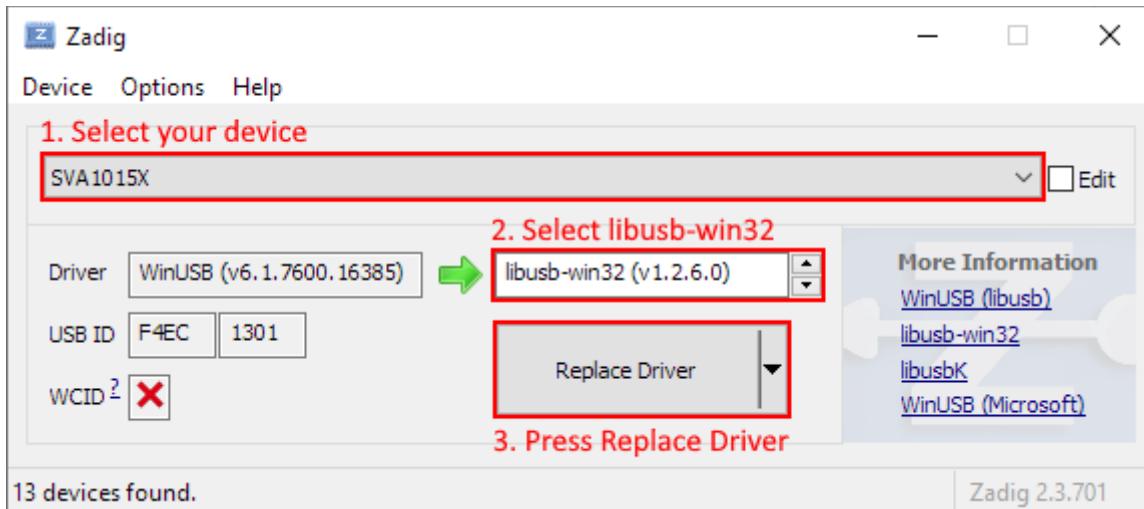


Fig. 2: Zadig driver settings

3.7.2. Linux

You will probably need to add your Linux account to the `usbtmc` group before you can access the instrument using EEZ Studio. Connect your instrument with a USB cable and turn it on. Wait until booting is complete. Now check the instrument group name by entering the following command:

```
ls -l /dev/usbtmc*
```

In case it is *root*, enter the command:

```
sudo groupadd usbtmc
```

Now, add your account (<username>) to the group:

```
sudo usermod -a -G usbtmc <username>
```

A reboot is required. After that, the *gid* of `/dev/usbtmc0` should be set to `usbtmc` and you are ready to use your instrument via USB-TMC interface.

3.8. FAQ

Q: Where is the database file by default?

A: Depending on the operating system, it can be:

- **Linux:** `~/.config/eezstudio/storage.db`
- **Mac:** `~/Library/Application\ Support/eezstudio/storage.db`
- **Windows:** `%appdata%\eezstudio\storage.db`

The default created database as well as its location can be changed later through the options in the *Settings* section of EEZ Studio.

Q: Where are the IEXTs (Instrument EXTensions) used to access T&M instruments stored?

A: Depending on the operating system, it can be:

- **Linux:** `~/.config/eezstudio/extensions`
- **Mac:** `~/Library/Application\ Support/eezstudio/extensions`
- **Windows:** `%appdata%\eezstudio\extensions`

4. Key features

4.1. General

- Modern and attractive UI/UX developed in [Electron](#)
- Light / Dark theme
- Multi-tab support for faster navigation
- Cross-platform run-time (Linux, Windows, macOS)
- Modular design based on plug-ins that can be added/removed depends of scope of the work
- Source/Version control integration ([GitHub](#) and [gitea.io](#))
- Open source project

4.2. EEZ Studio Project

- Modular visual development environment for designing TFT display screen decorations and defining user interaction (embedded GUI)
- [LVGL](#) (Light and Versatile Graphivs Library) support
- Generate C++ code for embedded GUI functionality that can be directly included in [STM32CubeIDE](#) for EEZ BB3 and other STM32 target platforms or [Arduino IDE](#) for H24005 and other Arduino compatible target platforms
- *Instrument definition file* (IDF) builder with context sensitive SCPI commands help (based on Keysight's [Offline Command Expert command set](#) XML structure) suitable for EEZ Studio *Instrument* and [Keysight Command Expert](#)
- SCPI command help generator based on bookmarked HTML generated directly from .odt file using [EEZ WebPublish](#) extension for OpenOffice/LibreOffice.
- Project templates (using giteo.io repositories) and comparison of projects
- Drag&drop editor for creating instrument's desktop dashboard (for remote control and management)
- Flowchart based low-code programming for desktop dashboard

4.3. EEZ Studio Instrument

- Dynamic environment where multiple instruments can be configured and easily accessed
- Session oriented interaction with each SCPI instrument
- Serial (via USB), Ethernet and VISA (via free [R&S®VISA](#)) T&M instrument interfaces support
- Direct import of EEZ Studio generated IDFs and Keysight's Offline Command Expert command sets
- IEXT (Instrument EXTension) catalog with growing number of supported instruments (Rigol, Siglent, Keysight, etc.)
- History of all activities with search/content filtering
- Quick navigation via calendar ("heatmap") or sessions list view
- Shortcuts (hotkeys and buttons) that can be user defined or come predefined from imported IDF. The shortcut can contain single or sequence of SCPI commands or Javascript code.
- Javascript code for task automation (e.g. logfile, or programming list upload/download, etc.) can be also assigned to the shortcut
- SCPI commands context sensitive help with search
- File upload (instrument to PC) with image preview (e.g. screenshots)
- File download (PC to instrument) automation for transferring instrument profiles
- Simple arbitrary waveform editor (envelope and table mode)
- Displaying measurement data as graphs
- FFT analysis, harmonics and simple math functions (Period, Frequency, Min, Max, Peak-to-Peak, Average)
- Export graphs as .CSV file

5. Menu options and Settings

5.1. Home page

After starting EEZ Studio, the home page is displayed, which is actually the *Home* tab that is always present (it cannot be hidden). *Main tabs* section (1) allows easy navigation between multiple open projects, instruments as well as *Extension Manager* and *Settings* sections (Fig. 3).

The main sections of EEZ Studio are *Extension Manager*, *Settings* which are accessible from the *Main option bar* (2), while the *Projects* (3) and *Instruments* (4) sections are positioned below and have their own option bars whose options when selected also appear in the *Main tabs* section.

The *Projects* section will be described in detail in chapters xx to xx, and the *Instruments* section in chapters xx to xx.

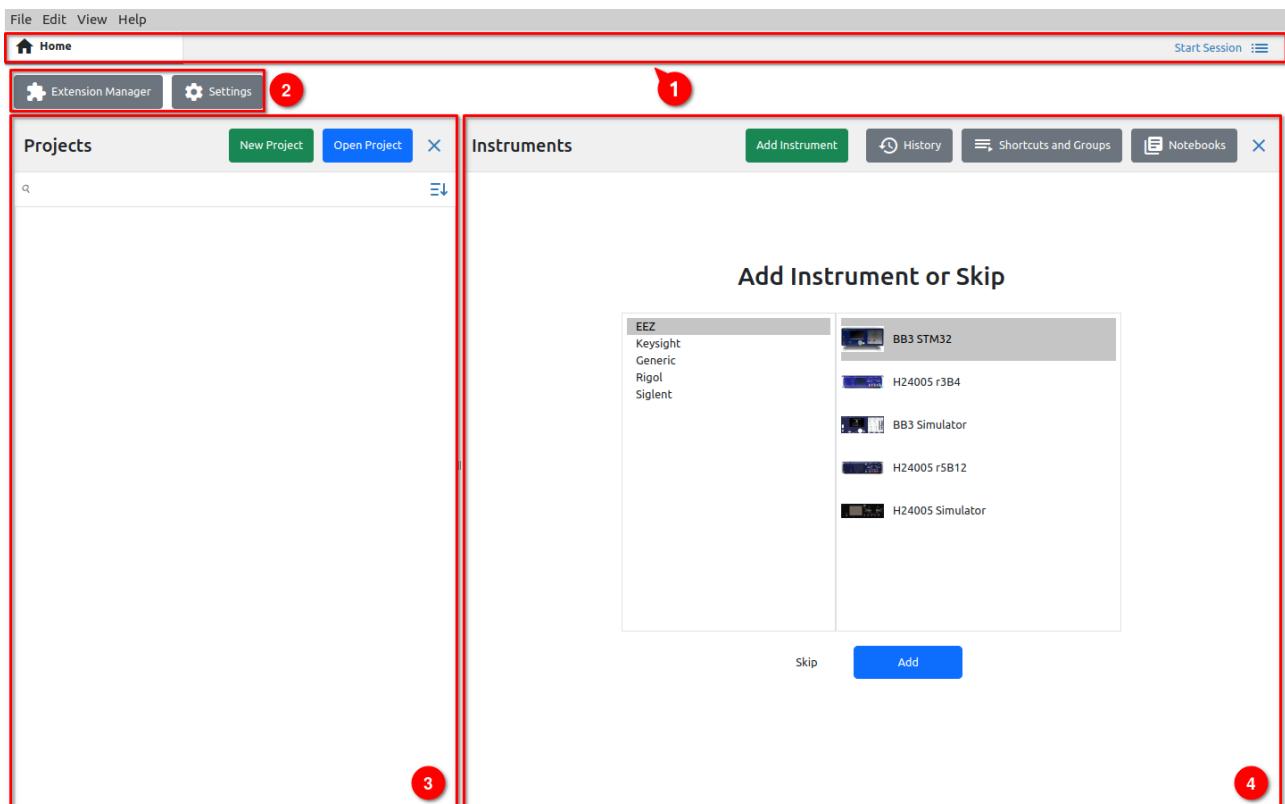


Fig. 3: Home page

5.2. Menu options

Menu options available from all main sections of EEZ Studio are listed below.

5.2.1. File

| Option | Shortcut | Description |
|---------------------------------|------------------|--|
| New project... | Ctrl + N | Creates a new project. |
| Add instrument... | Alt + Ctrl + N | Adds an instrument to the EEZ Studio workbench that can be controlled. |
| New Window | Ctrl + Shift + N | Opens a new copy of the window. |
| Open... | Ctrl + O | Opens an existing project. |
| Open Recent | - | List of recently opened projects. |
| Import Instrument Definition... | - | Import IEXT (Instrument EXTension) file. |
| Save | Ctrl + S | Saving project files. |

Exit - EEZ Studio shutdown.

5.2.2. Edit

| Option | Shortcut | Description |
|-------------------|----------|-------------------------------|
| <i>Undo</i> | Ctrl + Z | Undo previous action. |
| <i>Redo</i> | Ctrl + Y | Redo previous action. |
| <i>Cut</i> | Ctrl + X | Move content to Clipboard. |
| <i>Copy</i> | Ctrl + C | Copy content to Clipboard. |
| <i>Paste</i> | Ctrl + V | Paste content from Clipboard. |
| <i>Delete</i> | Del | Delete selected content. |
| <i>Select All</i> | Ctrl + A | Select all content. |

5.2.3. View

| Option | Shortcut | Description |
|-------------------------------|------------------|---|
| <i>Home</i> | - | Return to the <i>Home</i> tab. |
| <i>History</i> | - | Opening the Instrument's <i>History</i> tab. |
| <i>Shortcuts and Groups</i> | - | Opening the Instrument's <i>Shortcuts and Groups</i> tab. |
| <i>Notebooks</i> | - | Opening the Instrument's <i>Notebooks</i> tab. |
| <i>Extension Manager</i> | - | Opening the Instrument's <i>Extension Manager</i> tab. |
| <i>Settings</i> | - | Opening the <i>Settings</i> tab (Fig. 4). |
| <i>Toggle Full Screen</i> | F11 | View EEZ Studio in full screen (select F11 again to restore). |
| <i>Toggle Developer Tools</i> | Ctrl + Shift + I | Opening the developer tools in the right part of the window. |
| <i>Switch to Dark Theme</i> | Ctrl + Shift + T | Toggle between Light and Dark theme. |
| <i>Zoom In</i> | Ctrl + + | Zoom in (enlargement) of all screen elements. On some Linux distributions you will need to use Ctrl + Shift + + as a shortcut. |
| <i>Zoom Out</i> | Ctrl + - | Zoom out (reduction) of all screen elements. |
| <i>Reset Zoom</i> | Ctrl + 0 | Returning the zoom to the default level. |
| <i>Reload</i> | - | Reload all content. |

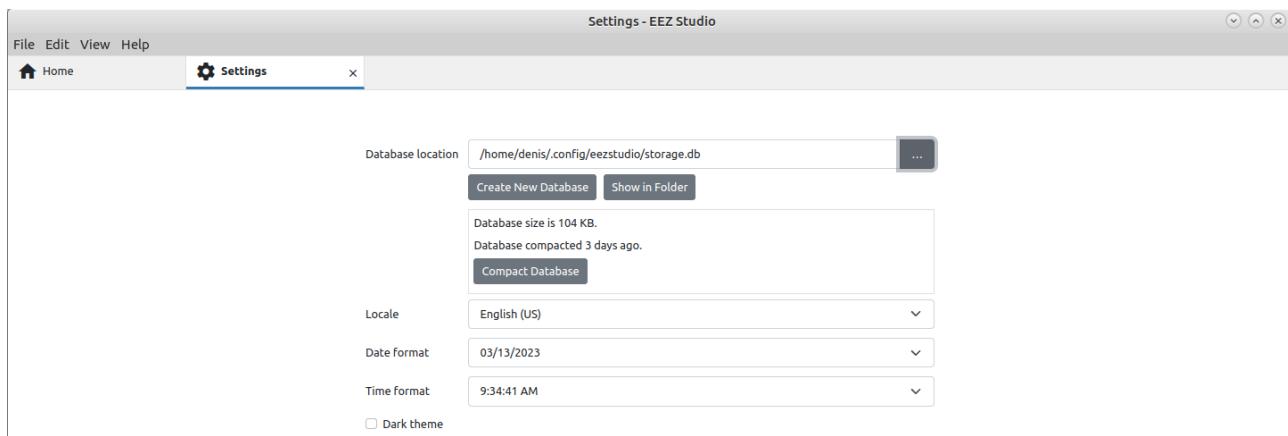


Fig. 4: Settings tab

Database location

A database is used to store the data collected in communication with the instruments. An empty base

is created at first launch and its location can be seen here. You can also change the location here to one of the existing databases (backup, imported from another EEZ Studio, etc.).

Changing the parameters of the database requires a restart of EEZ Studio. The Restart button will be displayed in the lower right corner.

Create New Database

Creating a new database with the name and location you specified.

Show in Folder

View the folder where the database is located.

Locale

Defines the date and time formats for the selected country.

Changing the Locale requires a restart of EEZ Studio. The Restart button will be displayed in the lower right corner.

Date format

Display format of all date values.

Time format

Display format of all time values.

Dark theme

Toggle between Light and Dark theme (same as shortcut Ctrl + Shift + T).

5.2.4. Help

| Option | Shortcut | Description |
|--------|----------|--|
| About | - | Opens the EEZ Studio version information (Fig. 5). |

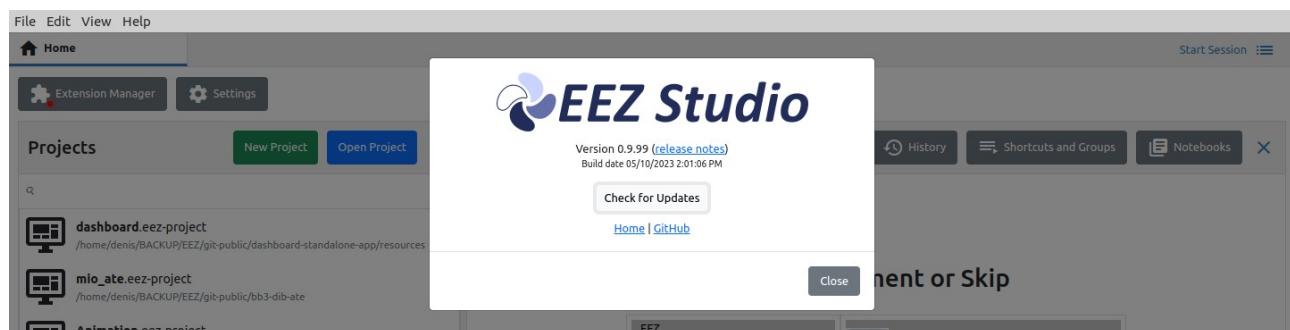


Fig. 5: About page

Check for Updates

This function requires an internet connection in order for EEZ Studio to connect to the GitHub repository and check for a newer version than the one installed.

This function does not take into account versions that have a pre-release status, but only released versions.

Home

Opens the home page of the Envox official site (requires internet browser installed).

Github

Opens Envox's GitHub home page (requires internet browser installed).

*EEZ Studio
Instrument*

6. Home page instrument sections

The top of the home page contains general options (1) for working with instruments (Fig. 6). Instrument specific *History*, *Shortcuts and Groups* and *Notebooks* options can also be accessed through the *Instruments action bar* for the currently selected instrument as described below. The Instruments section can be optionally hidden (2) when the Show Instruments option appears (Fig. 7).

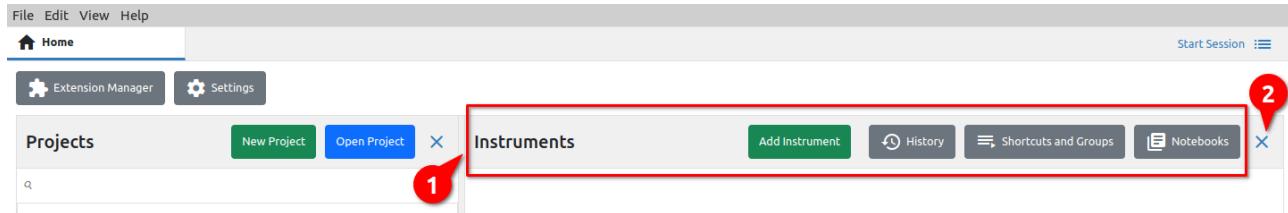


Fig. 6: Home page instrument options



Fig. 7: Home page "Show instruments" option

6.1. History

History displays communication via the *Terminal* option for all instruments in one place. In this way, it will be easier to search all activities as well as to add notes, files and graphs in the same way as in the *Terminal* of the currently selected instrument, as will be described below.

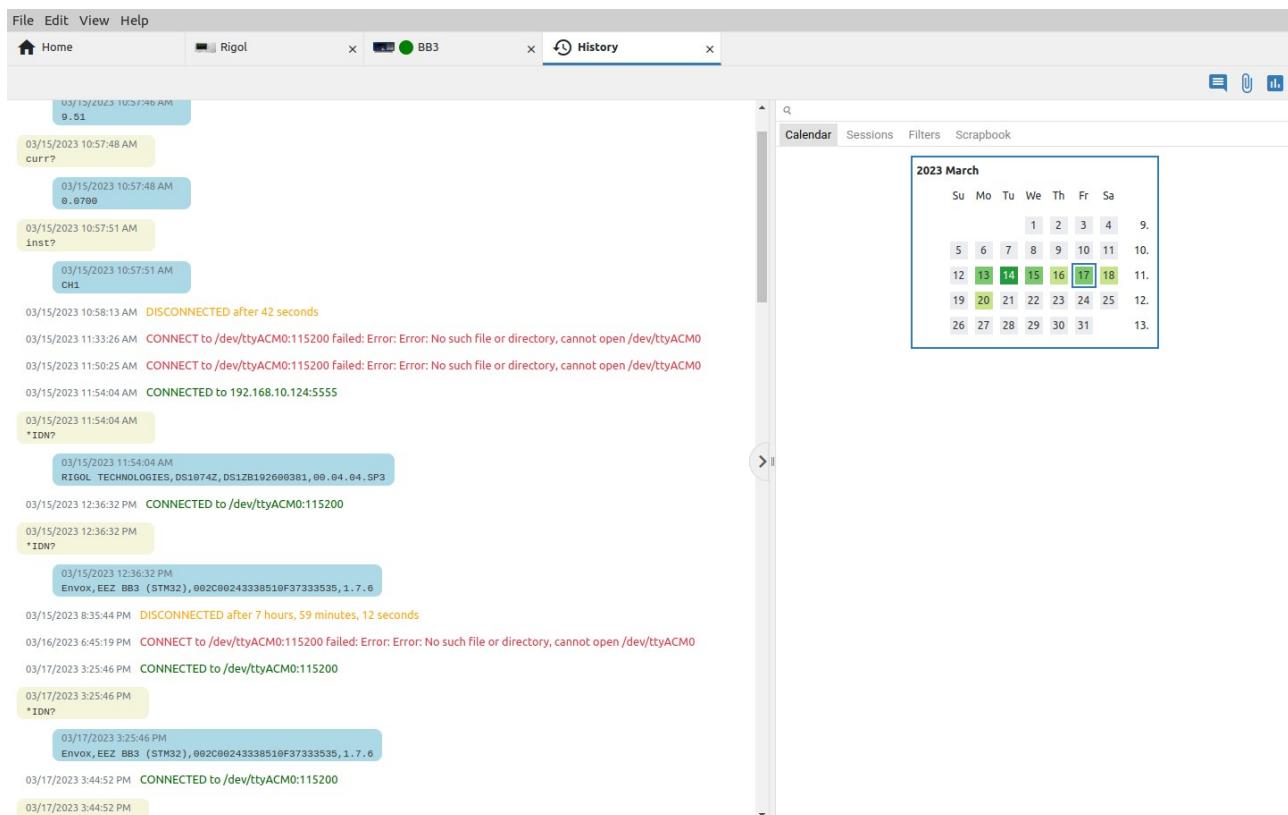


Fig. 8: Instruments History view

6.2. Shortcuts and Groups

Just like with *History*, *Shortcuts and Groups* is not a system feature, but only displays the available shortcuts and their groups in one place for easier searching, editing, deleting and adding new shortcuts and their groups.

Therefore, all operations with shortcuts on this page are possible as via the *Shortcuts* page of the currently selected instrument, which will be described below.

EEZ Studio User manual

| Name | Group / Extension | Keybinding | Action | Confirmation | Toolbar | Toolbar position |
|-------------------|---|------------|------------|--------------|---------|------------------|
| Abort | EEZ BB3 STM32 EEZ BB3 Simulator EEZ H24005 r3B4 | F9 | SCPI | ✓ | | 9 |
| Clear protections | EEZ BB3 STM32 EEZ BB3 Simulator | F10 | SCPI | ✓ | ✓ | 10 |
| Clear protections | EEZ H24005 r3B4 | F10 | SCPI | ✓ | ✓ | 10 |
| Coupling | EEZ BB3 STM32 EEZ BB3 Simulator | F5 | JavaScript | ✓ | | 5 |
| Dlog abort | EEZ BB3 STM32 EEZ BB3 Simulator EEZ H24005 r3B4 | — | SCPI | ✓ | | 15 |
| Dlog start | EEZ BB3 STM32 EEZ BB3 Simulator | — | JavaScript | ✓ | | 13 |
| Dlog start | EEZ H24005 r3B4 | — | JavaScript | ✓ | | 13 |
| Dlog upload | EEZ BB3 STM32 EEZ BB3 Simulator | — | JavaScript | ✓ | | 14 |
| Dlog upload | EEZ H24005 r3B4 | — | JavaScript | ✓ | | 14 |
| Init | EEZ BB3 STM32 EEZ BB3 Simulator EEZ H24005 r3B4 | F8 | SCPI | ✓ | | 8 |
| Outputs OFF | EEZ BB3 STM32 EEZ BB3 Simulator | F1 | SCPI | ✓ | | 1 |
| Outputs OFF | EEZ H24005 r3B4 | F1 | SCPI | ✓ | | 1 |
| Outputs ON | EEZ BB3 STM32 EEZ BB3 Simulator | F2 | SCPI | ✓ | | 2 |
| Outputs ON | EEZ H24005 r3B4 | F2 | SCPI | ✓ | | 2 |

Fig. 9: Instruments Shortcuts and Groups view

6.3. Notebooks

The *Notebooks* feature enables data collected from one or more sources (instruments) to be stored and presented in one place. Data stored in this way can be searched as if they belonged to a single source. Notebooks can also be appended, exported and imported, which facilitates the exchange of collected data.

Figure 10 shows the 'Instrument Notebooks' view in EEZ Studio. The interface includes:

- Toolbar: Includes icons for Home (1), New (2), Open (3), Save (4), and others.
- Left Panel: Shows a list of notebooks, with 'bb3 notebook test' and 'test notebook' highlighted.
- Central Area: Displays three data snippets from the notebook:
 - 03/13/2023 10:47:03 AM: Source: RIGOL TECHNOLOGIES,DS1074Z,DS1ZB192600381,00.04.04.SP3
 - 03/13/2023 10:47:03 AM: Source: RIGOL TECHNOLOGIES,DS1074Z,DS1ZB192600381,00.04.04.SP3
 - 03/13/2023 10:47:06 AM: Source: RIGOL TECHNOLOGIES,DS1074Z,DS1ZB192600381,00.04.04.SP3
- Bottom Area: Shows a screenshot of the Rigol DS1074Z oscilloscope interface.
- Right Panel: Includes a Calendar view for March 2023 (7), Filters, and Scrapbook tabs.

Fig. 10: Instrument Notebooks view

Option

1 Add / Import notebook

Description

Create a new blank notebook or import a notebook file. When creating a new notebook, you will need to enter a name. To import data into a notebook, use the Notebook option in the instrument's *Terminal*, as shown in Fig. 12: (1) go to the *Terminal* tab in the *Action bar*, (2) select one or more items and (3) export them to a notebook file, a new notebook or an already created notebook.

In the case of exporting to a file, it will be necessary to choose a destination on the local storage, and in the case of exporting to a new notebook, the name of the notebook should be entered.

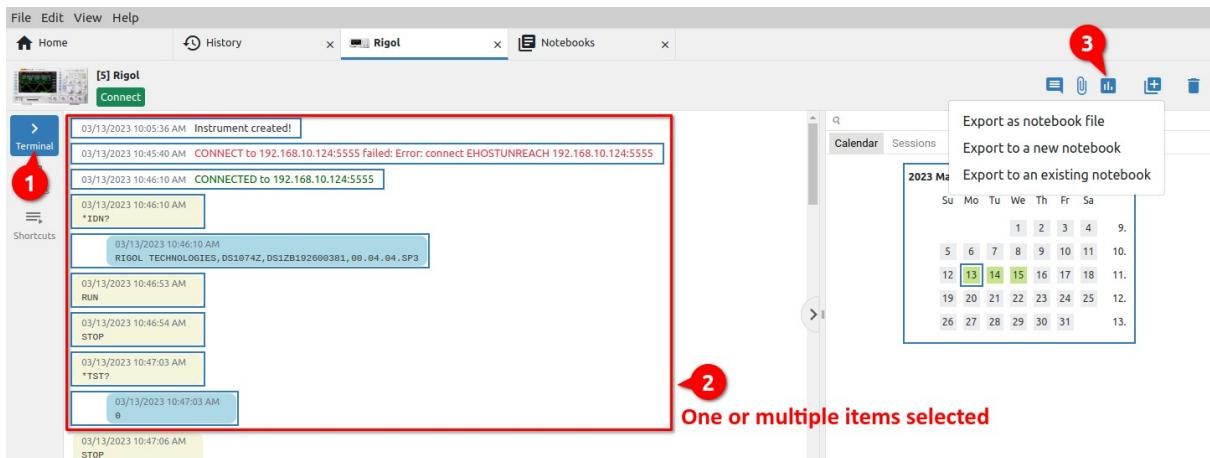


Fig. 11: Adding items to the notebook

2 Remove notebook

Remove the notebook from the list.

3 Change notebook name

Change notebook name.

4 Show deleted notebooks

Notebooks that have been removed from the list are not immediately deleted from the database. This option enables the display of all notebooks (Fig. 12) that have been removed from the list and offers the possibility to restore (return to the list) or permanently delete the notebook.

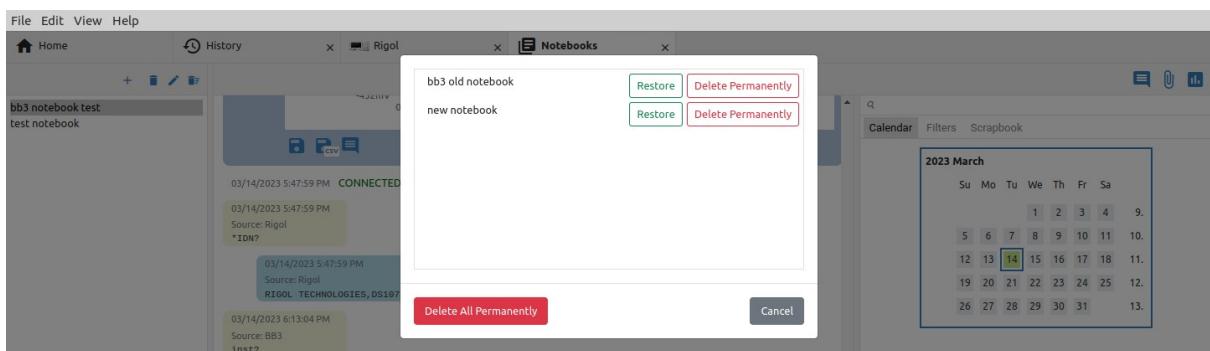


Fig. 12: Deleted notebooks

5 Add note

Adding a note to the notebook (Fig. 13). The number of notes is not limited and the last added note will appear at the bottom of the notebook (Fig. 14).

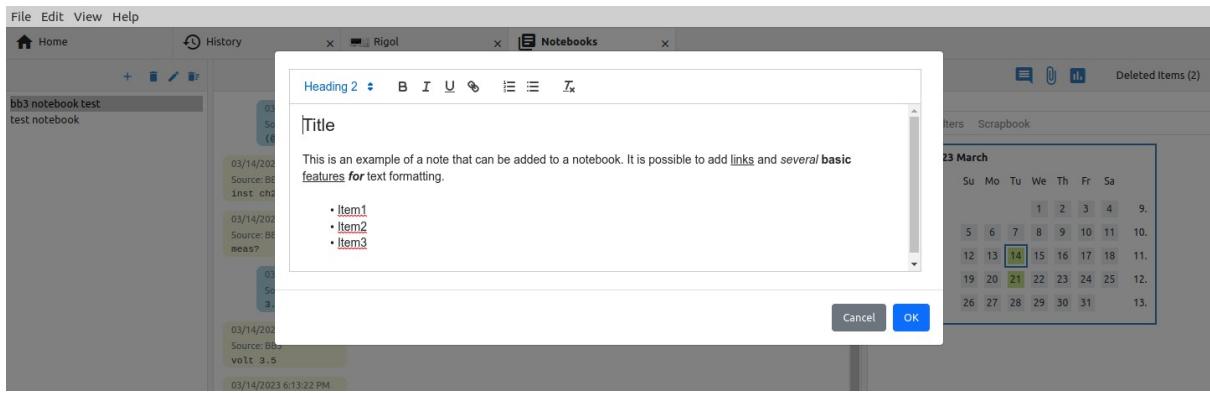


Fig. 13: Adding a new note to the notebook

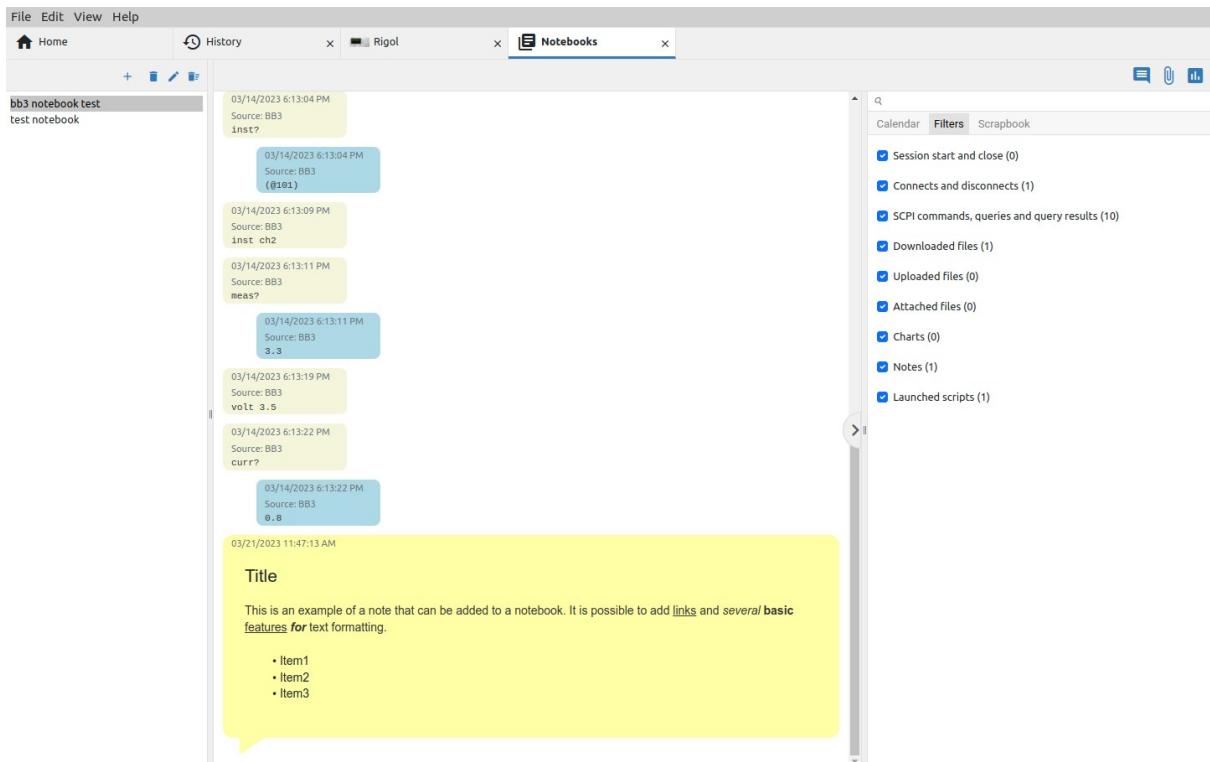


Fig. 14: Newly added note in the notebook

6 Attach file

Different files from local storage can be added to the notebook. In this way, all relevant data collected with the instruments can be combined together with images, recordings, datasheets into a whole that can be searched and further shared.

All imported files are marked with a paper clip icon in the upper left corner. It also displays the full path from where the file was imported as well as its size (Fig. 15).

Files whose format EEZ Studio can recognize (.jpeg, .png, etc.) also have a preview. Such files, in addition to the option to save to local storage and to add a note, will also have the option to copy to the clipboard.

6. Home page instrument sections

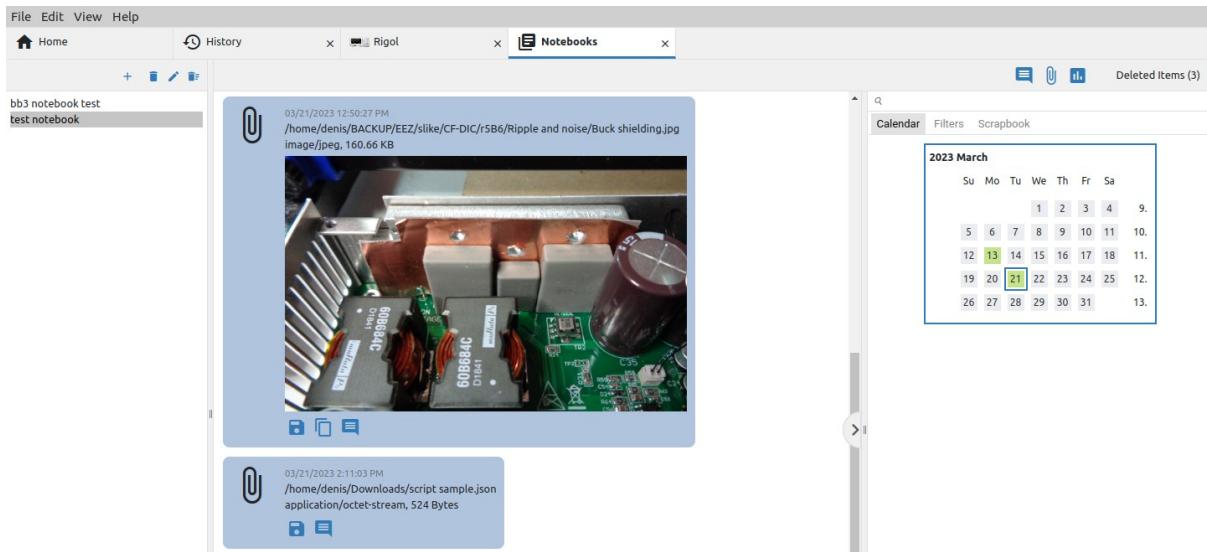


Fig. 15: Files imported into the notebook

7 Add chart

This option allows you to create a new graph from two or more existing ones and add it to the notebook. To create a new graph, you will need to select at least two of the found graphs in the currently selected notebook (1, 2) and add it to the notebook (3) as shown in Fig. 16.

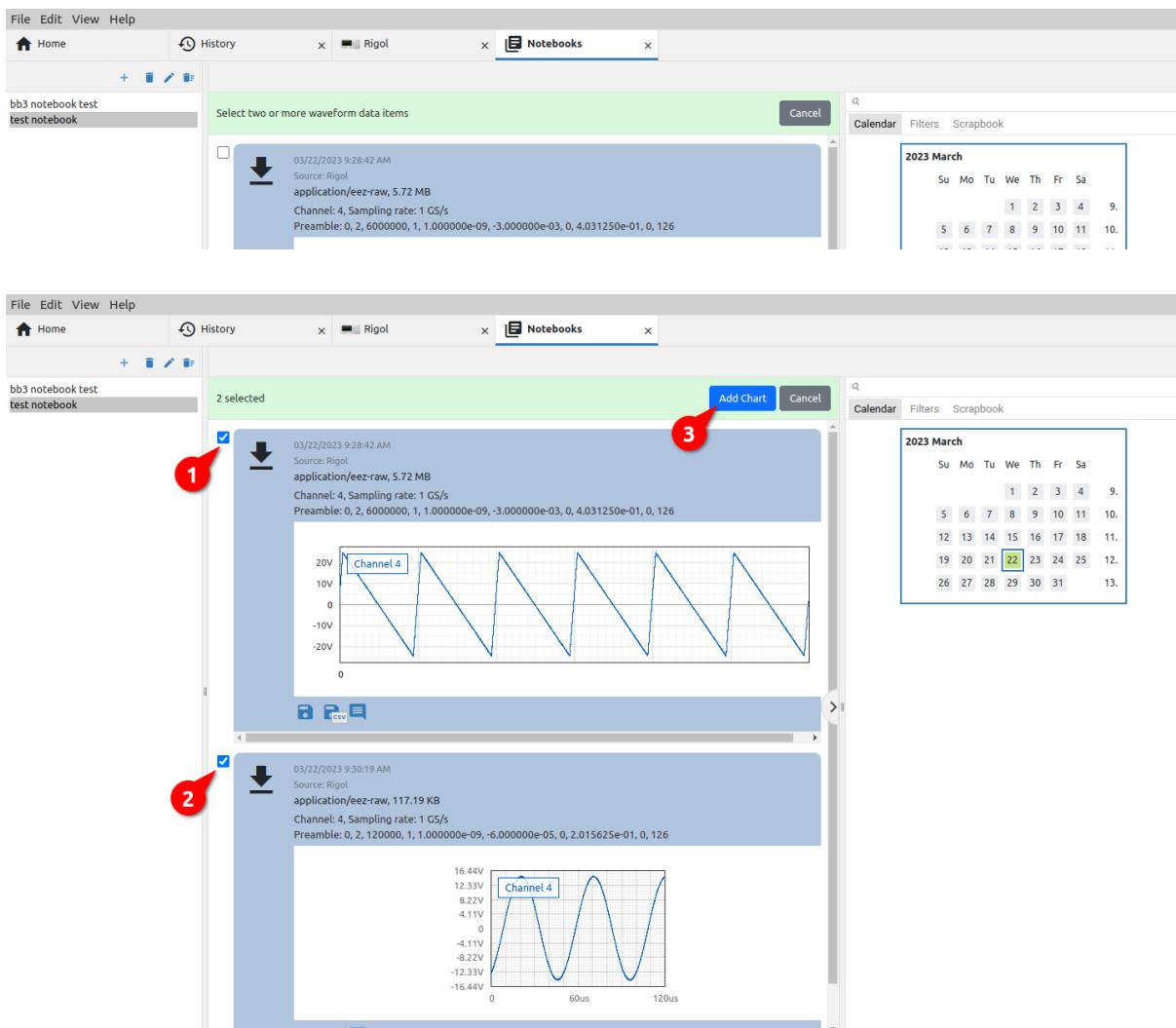


Fig. 16: A selection of graphs to add to the notebook

A successfully created graph will appear at the end of the notebook and will have a graph icon in the upper left corner (Fig. 17).

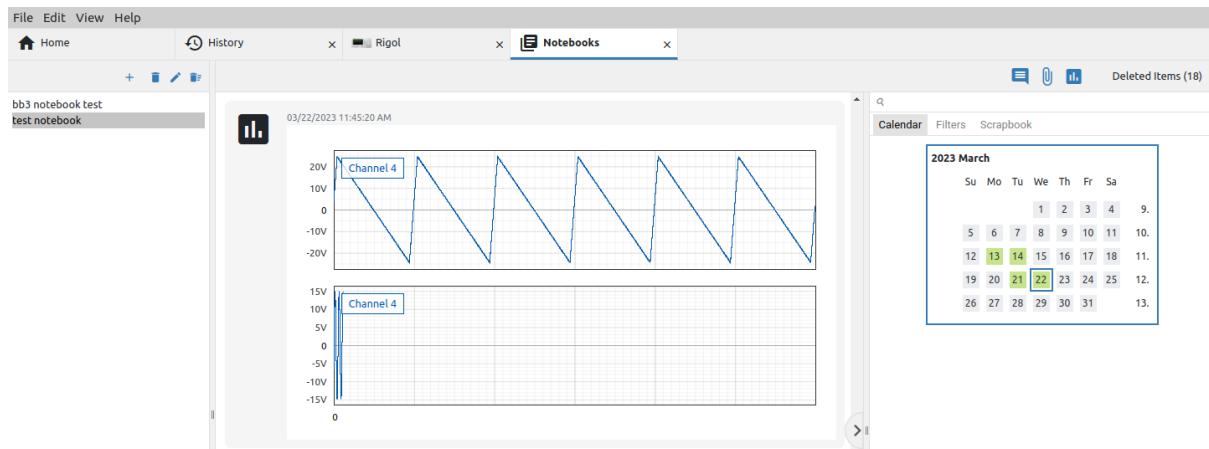


Fig. 17: Newly created graph added to notebook

6.4. Items purge and restore

Items that are removed from the list are not immediately deleted from the database, which leaves the possibility to restore them if needed. The counter of deleted items that can be restored appears in the right corner as shown in Fig. 18.

The counter can be seen in *Notebooks* but also in the *Terminal* tab of the currently selected instrument, and the same rules apply to restore or purge items in both places.

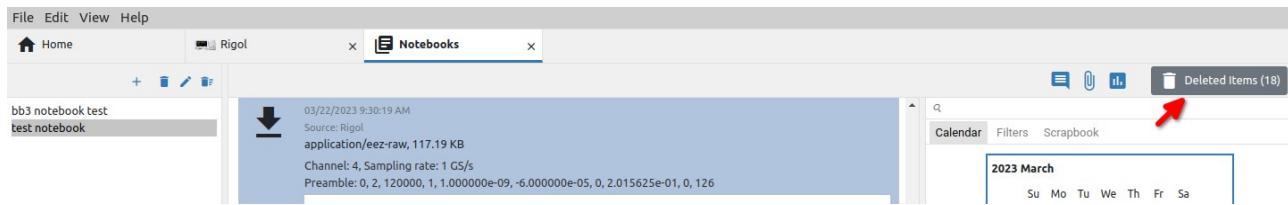


Fig. 18: Deleted items counter

When there are items to delete, they can be accessed by clicking on the counter, when the option to purge all items will first appear (Fig. 19).

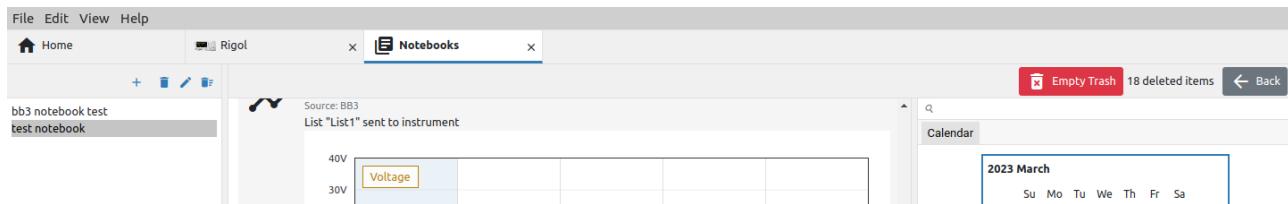


Fig. 19: Empty trash option (no selected items)

If one or more items are selected from the list of deleted items, options for restore (2) or purge (3) will appear (Fig. 20).

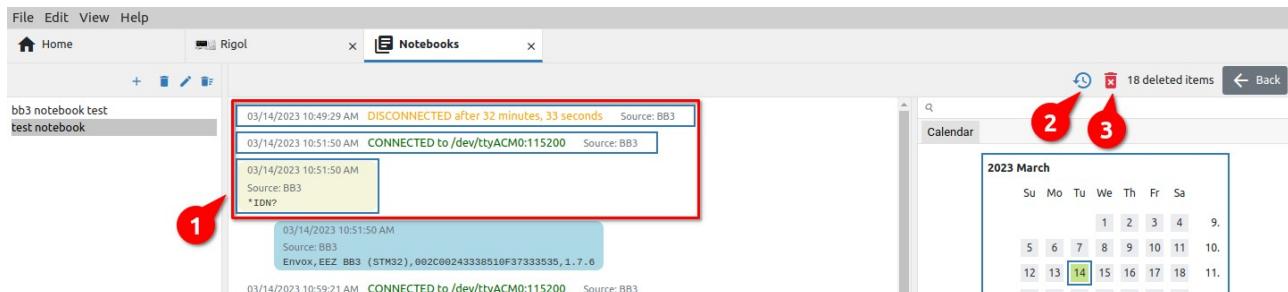


Fig. 20: Selection of deleted items for restore or purge

6.5. Instrument Extension (IEXT) Manager

The EEZ Studio use *Instrument Extensions* (IEXTs) to make communication and control of various instruments easier and more efficient. EEZ Studio comes with IEXTs for several instruments including EEZ H24005, EEZ BB3 as well as Generic SCPI which can be used for basic operations such as connection testing and sending commands and queries. (e.g. *IDN?).

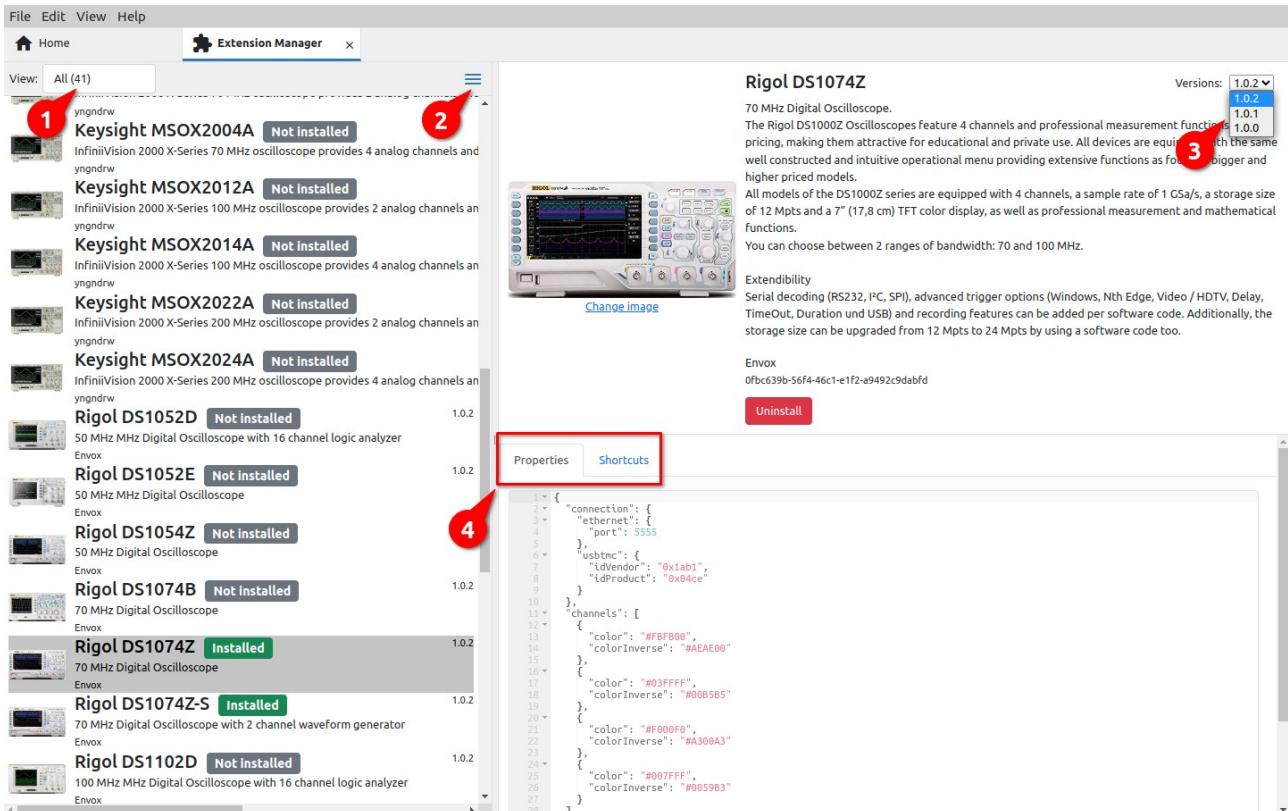


Fig. 21: Instrument extension (IEXT) Manager view

| # | Option | Description |
|---|---------------------------------|---|
| 1 | <i>View</i> | Filters for displaying IEXT in the list: it is possible to display all, only installed or only those that are not installed. The number of filtered IEXTs is displayed next to each option. |
| 2 | <i>Update / Install actions</i> | All approved IEXTs are in the catalog on GitHub, with which EEZ Studio synchronizes its catalog every time it is started. Synchronization with the IEXT catalog can also be started manually at any time using the <i>Upgrade Catalog</i> option. The <i>Install extension</i> option allows installing an IEXT that is not in the catalog (from local storage). |
| 3 | <i>Versions</i> | IEXT can have multiple versions. If there is more than one, it is possible to change the installed IEXT with one of the versions from the list. In this case, the Replace option will appear as in Fig. 22. |



Fig. 22: Changing installed IEXT version

4 Properties

IEXT for a supported instrument can have several properties that will be displayed below the IEXT description.
All displayed properties are for informational purposes and cannot be changed here.

6.6. Add instrument

By using *Add instrument* (Fig. 6), only those instruments for which there is an IEXT in the IEXT catalog can be added to the workbench.

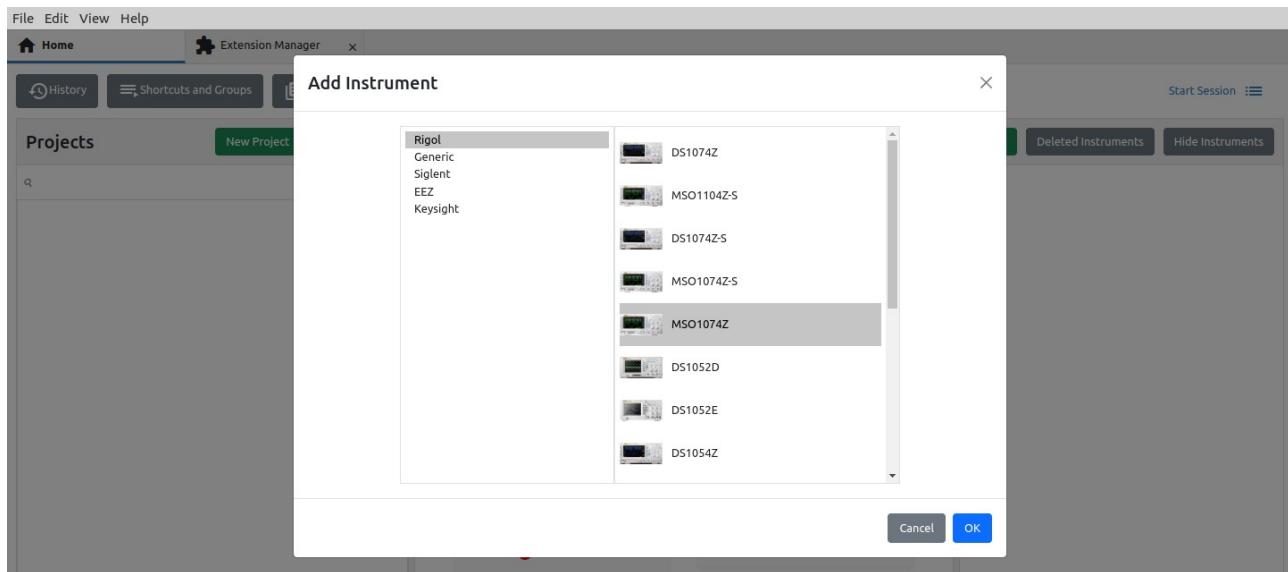


Fig. 23: Add instrument to workbench

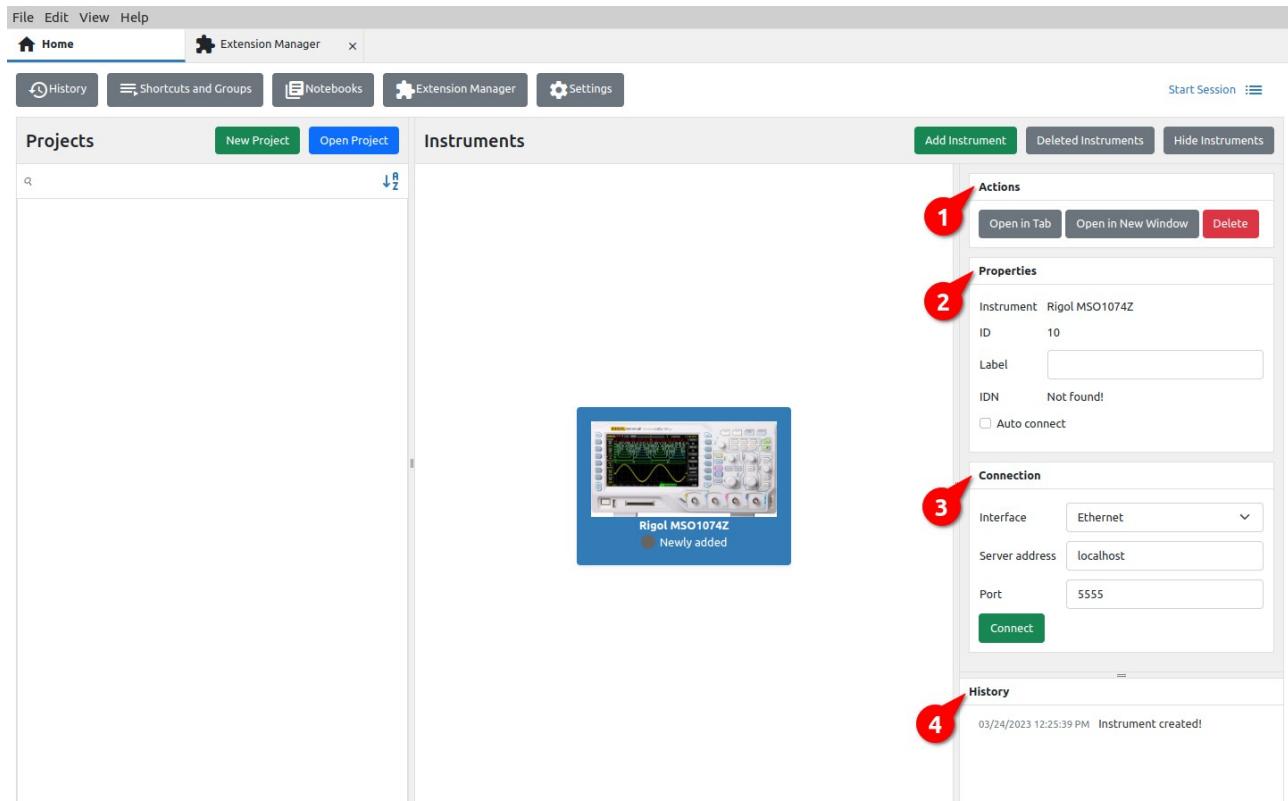


Fig. 24: New added instrument

A successfully added instrument will appear on the workbench (Fig. 24) with the label *Newly added*, and when selected, the sidebar will have the following sections:

| # | Option | Description |
|---|-------------------|--|
| 1 | <i>Actions</i> | Basic set of actions for displaying the instrument in a separate tab or new window and for removing it from the workbench. |
| 2 | <i>Properties</i> | The properties of the instrument contain information about the IEXT name, the internal ID, the instrument label that can be changed as desired, the identification string that the instrument returns in response to the SCPI query *IDN? and the option to automatically establish a connection with the instrument when starting EEZ studio. |
| 3 | <i>Connection</i> | Connection type. Connections to the instrument are defined in IEXT and there can be several of them. Depending on the type of connection (e.g. Serial, Ethernet, USBTMC, VISA), the associated connection parameters will also be displayed. |

Please note that the USBTMC and VISA interfaces are experimental and may not work properly on your computer.

For normal communication via the VISA interface, it will be necessary to install a free [R&S®VISA](#) driver. In case it is not installed or there is some problem in communication with it, an error message will appear as in Fig. 25.

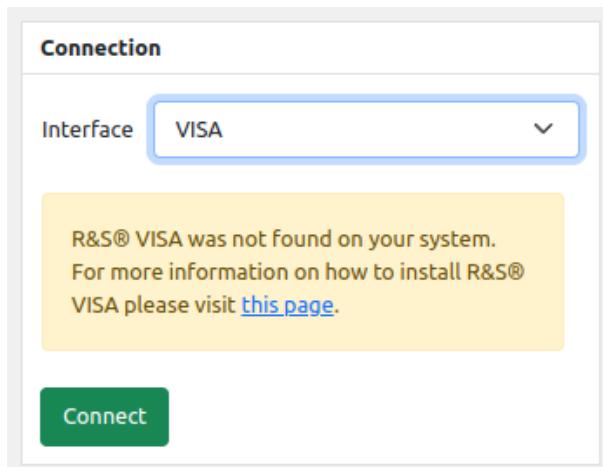


Fig. 25: VISA driver error message

- 4 *History* Preview history of interaction with the instrument using *Terminal*.

6.7. Establishing a connection with the instrument

Connection to the instrument added to the workbench will be possible as shown in Fig. 26: select the instrument from the workbench (1), select the interface in the Connection section (2) and click the Connect button (3).

If the Instrument tab (1) is open, as shown in Fig. 27 to establish a connection, it will be necessary to click on the *Connect* button (2) when a dialogue for choosing an interface will open in which the connection parameters are defined.

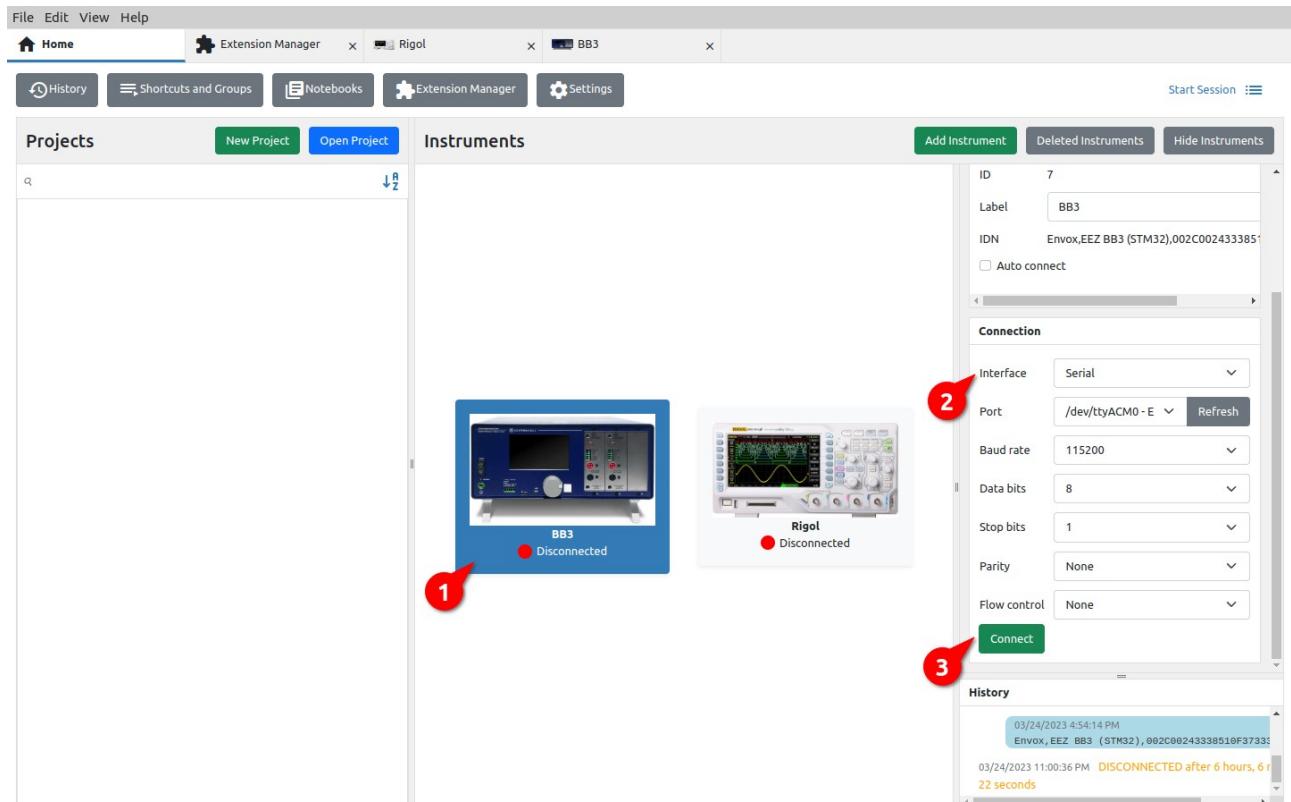


Fig. 26: Selecting an instrument on the workbench to establish a connection

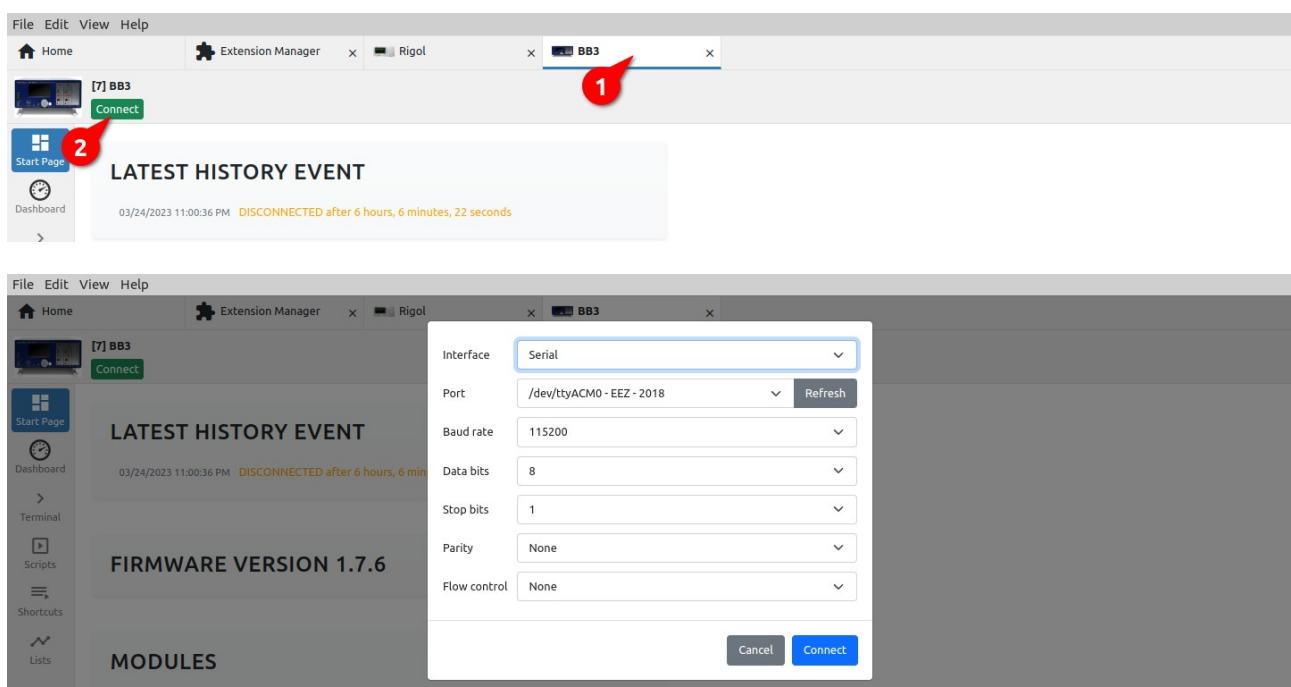


Fig. 27: Establishing a Connection from the Instrument tab

6. Home page instrument sections

Once the connection is established, it will be possible to close the connection by selecting the Disconnect button (Fig. 28).



Fig. 28: Option to close the connection

7. Instrument activity bar

When we open the instrument in its view, an *Activity bar* will be displayed along the left edge. The number of options in the activity bar is defined by IEXT and may vary for different instruments.

7.1. Start page (EEZ BB3 only)

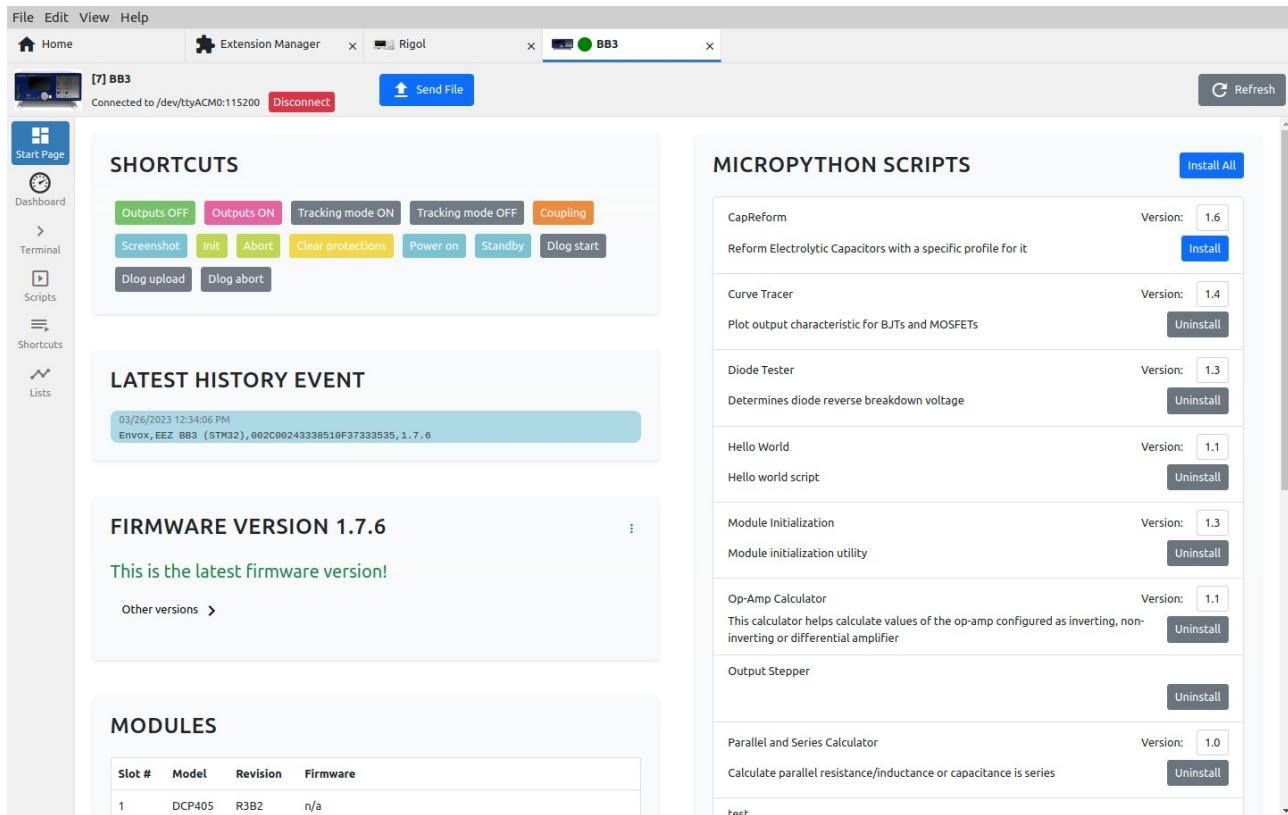


Fig. 29: EEZ BB3 start page

Section / option

Send file

Description

Opens a dialog for sending the file to EEZ BB3. To send, it is necessary to choose the source file, the desired name of the destination file. The destination folder path can be chosen from the offered list or set a new one. The parameters of the send file protocol are predefined and can be viewed and changed via the "gear" button in the lower left corner.

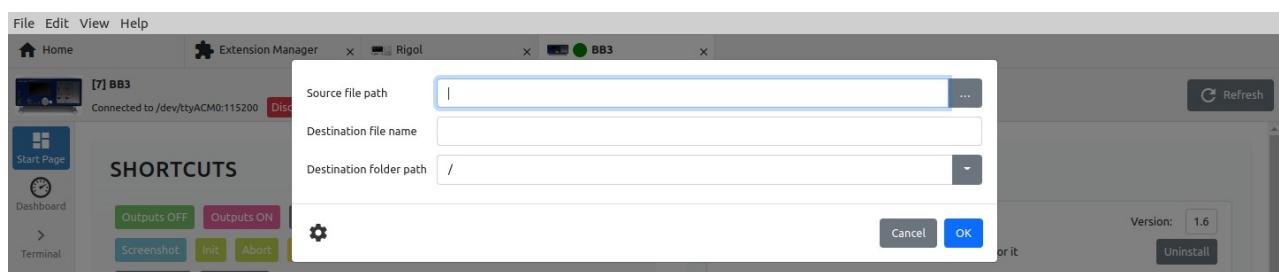


Fig. 30: Sending a file to EEZ BB3

Refresh

Refresh all data displayed on the *Start page*.

Shortcuts

List of available shortcuts from which they can be executed directly.

Latest history event

Shows the last result of interaction with the instrument via the *Terminal* tab.

Firmware version

Displays information about the installed firmware version. If a newer version than the currently installed one is published, an up-

grade option will be offered. It is also possible to manually install another version of the firmware (1) or downgrade the version from the offered list (2) as shown in Fig. 31.



Fig. 31: EEZ BB3 firmware version section

Modules

Display of installed modules. If the module has firmware, information will be displayed as to whether it is up-to-date or not and the possibility to upgrade or install another version.

Upload Pinout Pages is used to update pinout images of all modules.

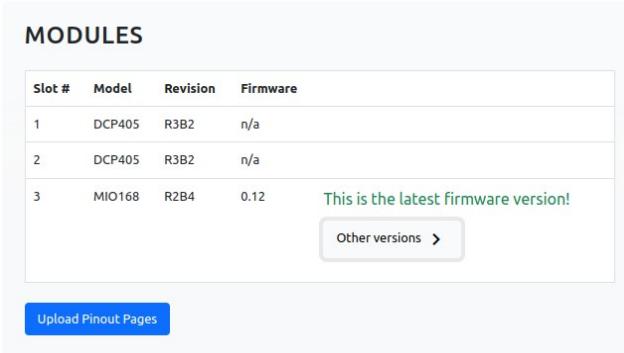


Fig. 32: EEZ BB3 modules section

Micropython scripts

List of all Micropython scripts that are on EEZ BB3. For scripts that are synchronized with the GitHub repository, their versions and options to install or uninstall will be displayed.

For scripts created by the user, versions will not be displayed, only the option to install or uninstall.

Lists

Program lists created by the user (see Section 7.4.). and which are located on EEZ BB3. Lists can be downloaded, uploaded and edited.

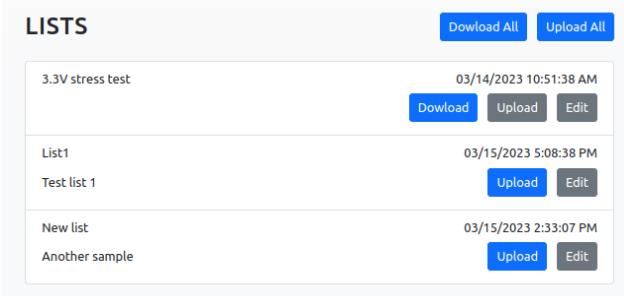


Fig. 33: EEZ BB3 program lists

7.2. Dashboard

Fig. 34 shows an example Dashboard that enables simple operations with EEZ BB3 modules.

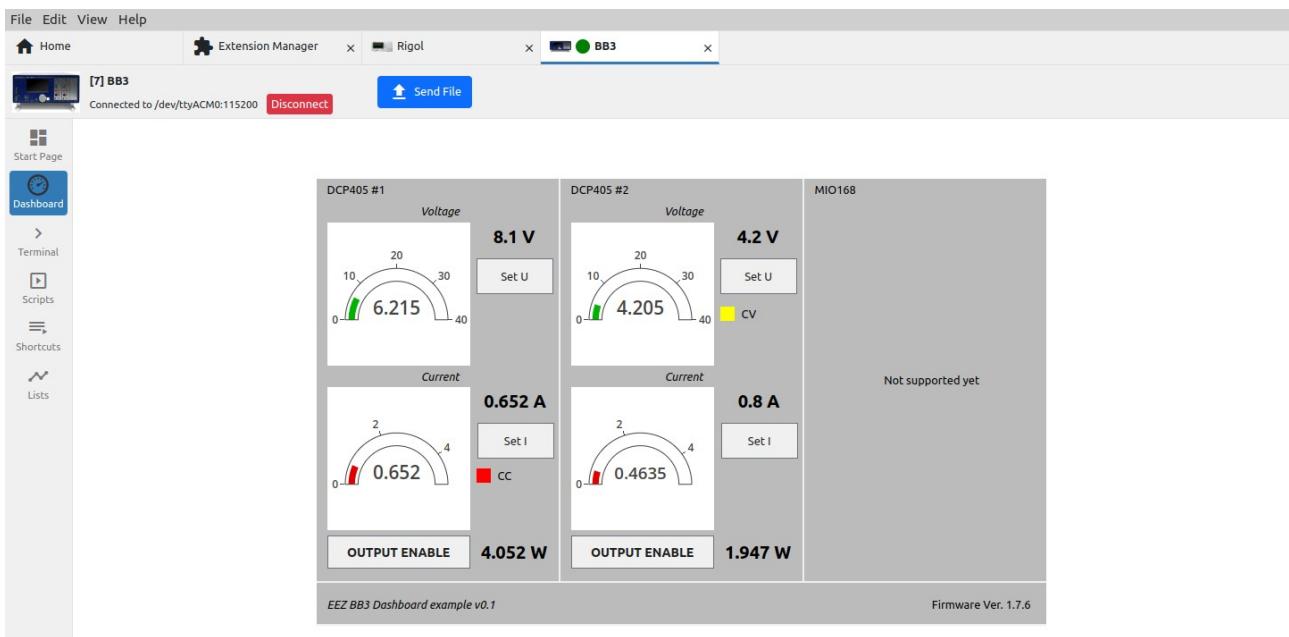


Fig. 34: Instrument dashboard example

7.3. Terminal

The *terminal* allows interaction with the instrument, which is primarily based on the SCPI specification.

The number of SCPI commands varies greatly between instruments, and IEXT can also include help for easier finding of the desired SCPI command or query that will be displayed at the bottom of the screen (7).

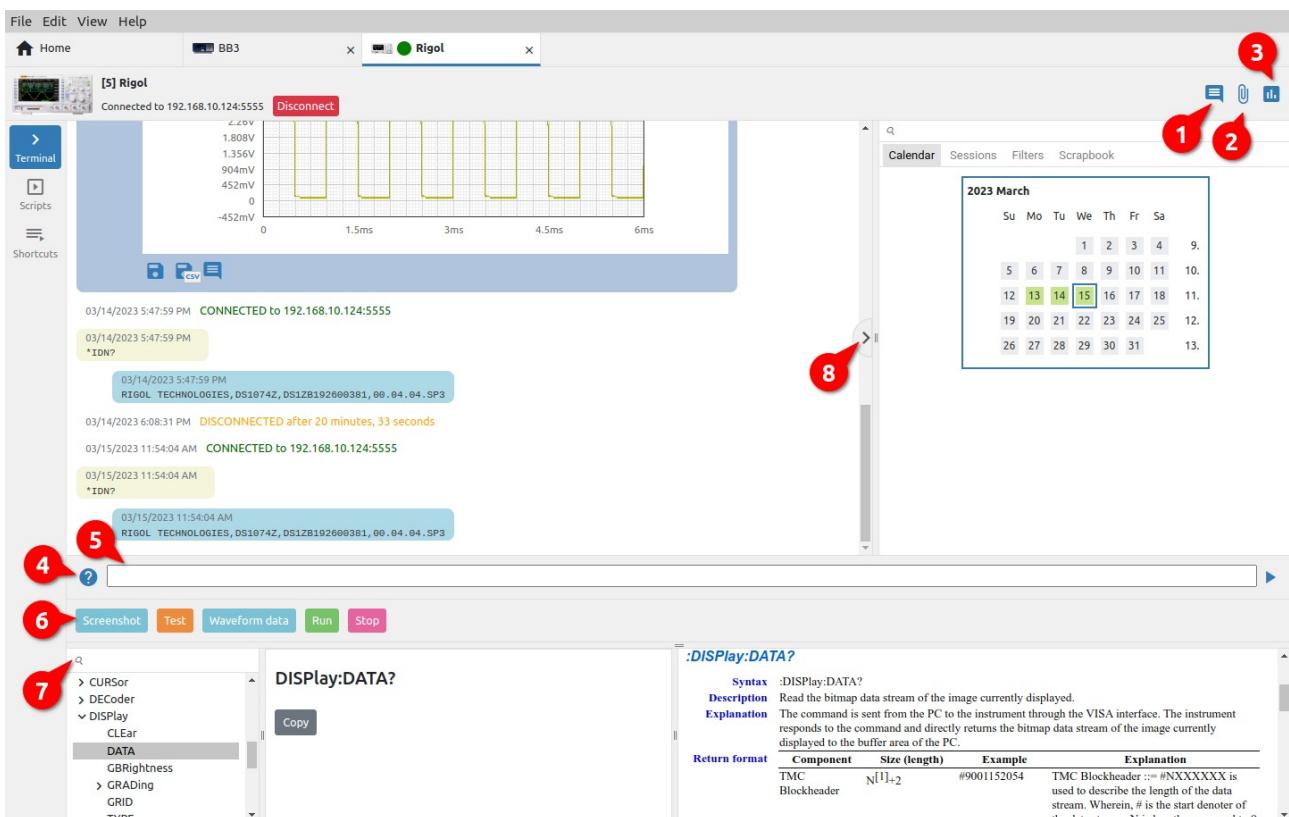


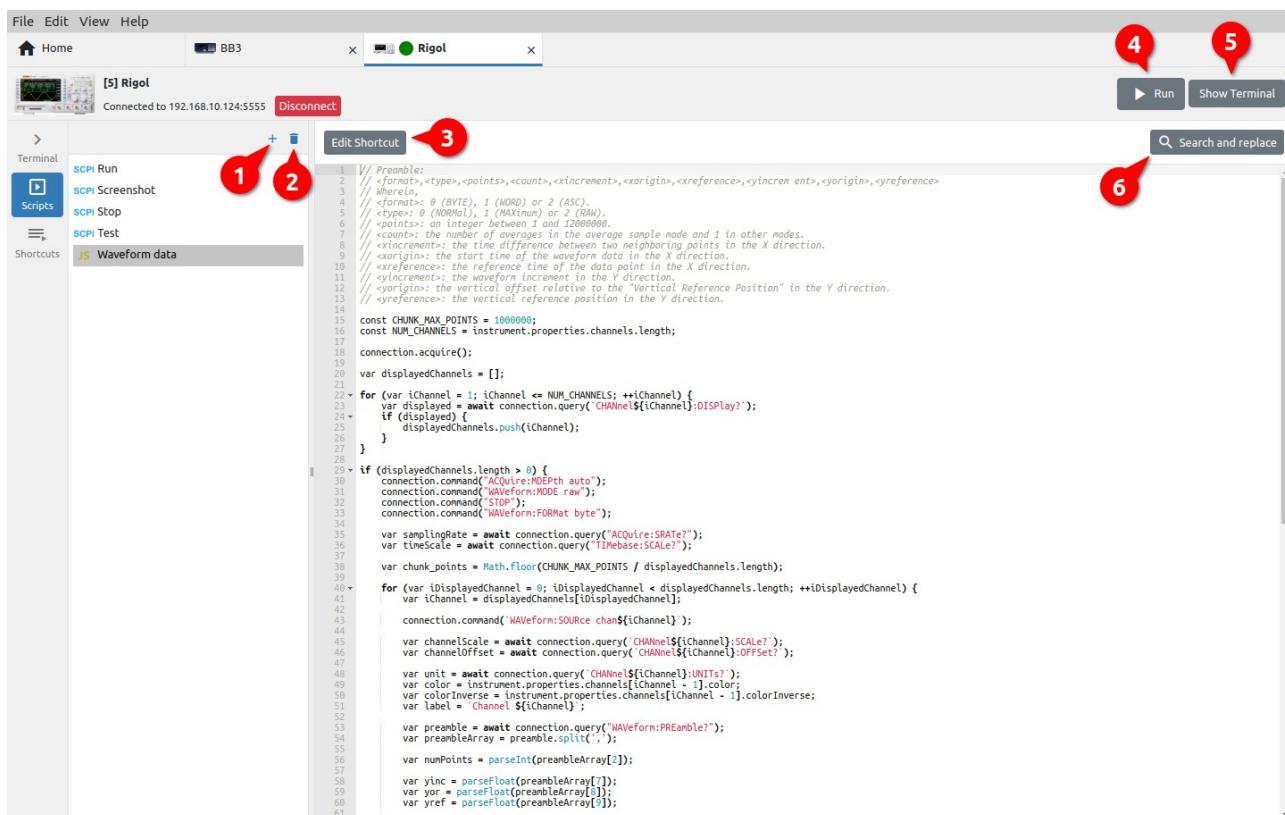
Fig. 35: Instrument terminal

| # | Option | Description |
|---|----------------------------|--|
| 1 | Add note | Adds a note (see Section 6.3). |
| 2 | Attach file | Attaching a file (see Section 6.3). |
| 3 | Add chart | Creating a new chart from two or more charts (see Section 6.3). |
| 4 | Show/hide commands catalog | Show or hide the help section for instrument commands at the bottom of the <i>Terminal</i> view. Command help will only be displayed if it is defined in IEXT for the selected instrument. Help for each command contains an explanation and syntax of how the command is used with the option to copy it to the command line (5). |
| 5 | Command line | Prompt line for sending a command to the instrument. |
| 6 | Shortcuts bar | IEXT imported and user defined shortcuts. |
| 7 | Command search | Commands help search. |
| 8 | Show/hide Side bar | Show or hide sidebar with history search options. |

7.4. Scripts

Scripts can be used to automate communication with the instrument (configuration, data collection, test sequences, etc.). Three types of scripts are supported: SCPI commands, JavaScript (JS) code and MicroPython (EEZ BB3 only) script. The number of scripts is unlimited and can be defined in IEXT or created by the user. A shortcut can be added to the script for easier launch.

In addition to containing complex programming procedures, a JS script can also contain GUI elements for communication with the user (entry forms, info or error messages, etc.).



The screenshot shows the EEZ Studio interface with the following numbered callouts:

- 1: A red circle points to the "Scripts" icon in the left sidebar.
- 2: A red circle points to the "New Script" button (+) in the left sidebar.
- 3: A red circle points to the "Edit Shortcut" button in the top right of the script editor.
- 4: A red circle points to the "Run" button in the top right of the interface.
- 5: A red circle points to the "Show Terminal" button in the top right of the interface.
- 6: A red circle points to the "Search and replace" bar in the top right of the interface.

The main area displays a JavaScript script for waveform data processing. The script uses promises to interact with the instrument's connection. It defines variables like `CHUNK_MAX_POINTS` and `NUM_CHANNELS`, and loops through channels to set up waveforms. It also handles scaling and offset calculations based on the number of displayed channels.

```

1 // Preamble:
2 // <format>, <type>, <points>, <count>, <xincrement>, <xorigin>, <xreference>, <yincrement>, <yorigin>, <yreference>
3 // <format>: 0 (BYTE), 1 (WORD) or 2 (ASC).
4 // <type>: 0 (NORMAL), 1 (MAXIMUM) or 2 (RAW).
5 // <points>: an integer between 1 and 12800000.
6 // <count>: the number of data points to average sample mode and 1 in other modes.
7 // <xincrement>: the time difference between two neighboring points in the X direction.
8 // <xorigin>: the start time of the waveform data in the X direction.
9 // <xreference>: the reference time of the data point in the X direction.
10 // <yincrement>: the waveform increment in the Y direction.
11 // <yorigin>: the offset relative to the "Vertical Reference Position" in the Y direction.
12 // <yreference>: the vertical reference position in the Y direction.
13
14 const CHUNK_MAX_POINTS = 1000000;
15 const NUM_CHANNELS = instrument.properties.channels.length;
16
17 connection.acquire();
18
19 var displayedChannels = [];
20
21 for (var iChannel = 1; iChannel <= NUM_CHANNELS; ++iChannel) {
22     var displayed = await connection.query(`CHANnel${iChannel}:DISPlay?`);
23     if (displayed) {
24         displayedChannels.push(iChannel);
25     }
26 }
27
28 if (displayedChannels.length > 0) {
29     connection.command("ACQuire:DEPth auto");
30     connection.command("WAveform:MODE raw");
31     connection.command("STOP");
32     connection.command("WAveform:FORMAT byte");
33
34     var samplingRate = await connection.query("ACQuire:SRATE?");
35     var timeScale = await connection.query("TImebase:SCALE?");
36
37     var chunk_points = Math.floor(CHUNK_MAX_POINTS / displayedChannels.length);
38
39     for (var iDisplayedChannel = 0; iDisplayedChannel < displayedChannels.length; ++iDisplayedChannel) {
40         var iChannel = displayedChannels[iDisplayedChannel];
41
42         connection.command(`WAveform:SOURce chAnel${iChannel}`);
43
44         var channelscale = await connection.query(`CHANnel${iChannel}:SCALE?`);
45         var channelOffset = await connection.query(`CHANnel${iChannel}:OFFSet?`);
46
47         var unit = await connection.query(`CHANnel${iChannel}:UNITS?`);
48         var color = instrument.palette.channels[iChannel - 1].color;
49         var colorInverse = instrument.properties.channels[iChannel - 1].colorInverse;
50         var label = `CHANnel ${iChannel}`;
51
52         var preamble = await connection.query("WAveform:PREamble");
53         var preambleArray = preamble.split(",");
54
55         var numPoints = parseInt(preambleArray[0]);
56
57         var yinc = parseFloat(preambleArray[1]);
58         var yor = parseFloat(preambleArray[2]);
59         var yref = parseFloat(preambleArray[3]);
60
61         if (unit === "V") {
62             yinc *= 1000000;
63             yor *= 1000000;
64             yref *= 1000000;
65         }
66     }
67 }

```

Fig. 36: Instrument scripts

| # | Option | Description |
|---|------------|--|
| 1 | Add script | Creating a new script. It will be necessary to define the name and type: SCPI, JS or MicroPython (EEZ BB3 only). |

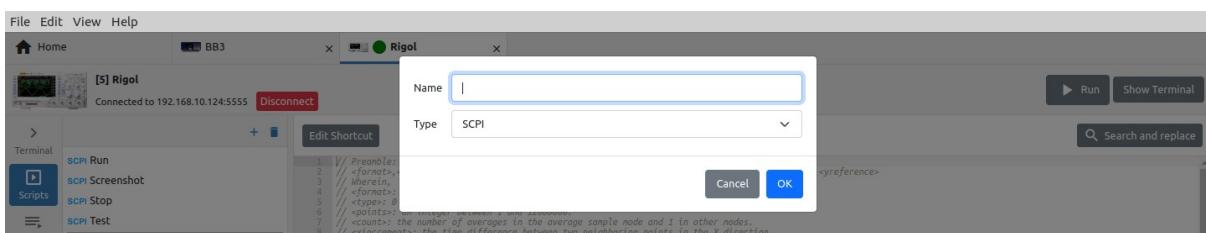


Fig. 37: Adding a new script

The content of the script is entered in the editor (Fig. 38).

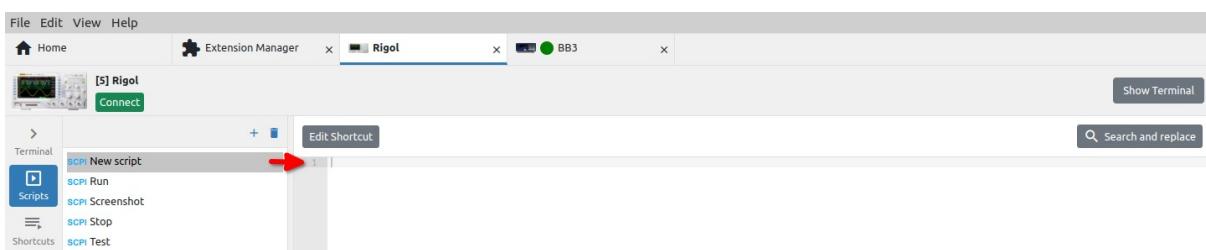


Fig. 38: Script editing

| | | |
|---|----------------------|--|
| 2 | Delete script | Deleting the selected script. |
| 3 | Edit shortcut | Editing a script shortcut (see Section 7.4.1.) |
| 4 | Run | Runs the script on the instrument. This option is only displayed if the connection to the instrument is established. |
| 5 | Show / Hide terminal | Show / hide Terminal on the right. |
| 6 | Search and replace | Script editor function for searching and replacing text in the script. By default, only the search field is displayed. To replace the found text, it will be necessary to click on the "+" sign. |

7.4.1. Edit script shortcut

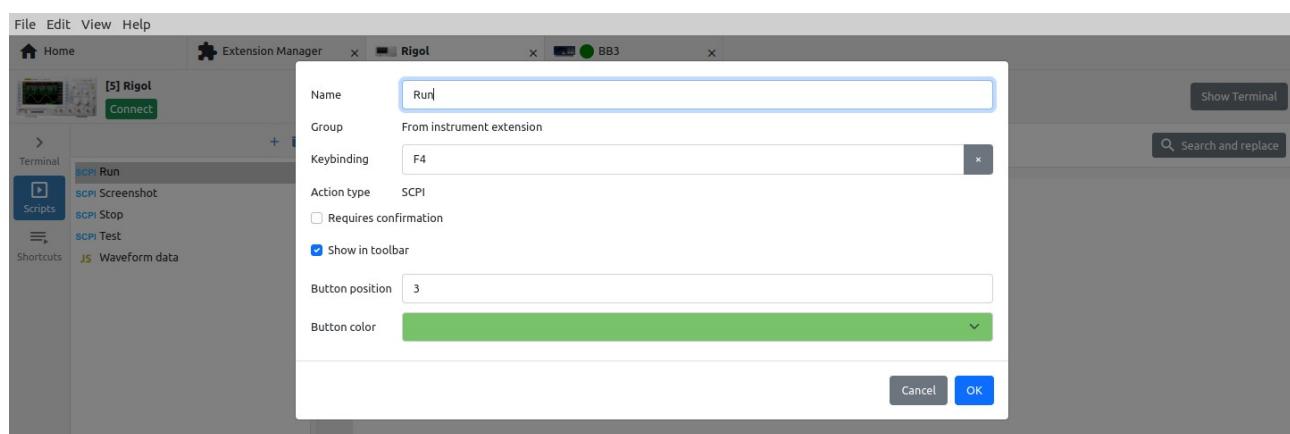


Fig. 39: Script shortcut editing

| Option | Description |
|--------|--|
| Name | The name of the script shortcut as it will be displayed in the shortcut bar. |
| Group | The name of the group to which the shortcut belongs. If the shortcut is defined in IEXT, the label <i>From instrument extension</i> will be displayed. |

| | |
|------------------------------|--|
| <i>Keybinding</i> | A key or a combination of several keys (e.g. with Shift, Alt, Ctrl) that will start the execution of the script. |
| <i>Action type</i> | Script type: SCPI, JS or MicroPython (EEZ BB3 only). |
| <i>Requires confirmation</i> | Displays a dialog box to confirm the execution of the script. |
| <i>Show in Shortcuts bar</i> | Determines whether the shortcut button will be displayed in the <i>Terminal's Shortcuts bar</i> . |
| <i>Button position</i> | The position of the shortcut button in the <i>Shortcuts bar</i> . When displaying, the shortcut with a lower value will be displayed first. If there are multiple shortcuts with the same value, they will be sorted alphabetically. |
| <i>Button color</i> | Color coding of shortcut button. |

7.5. Shortcuts

Shortcuts are used to simplify the execution of scripts and can be defined in IEXT or user defined.

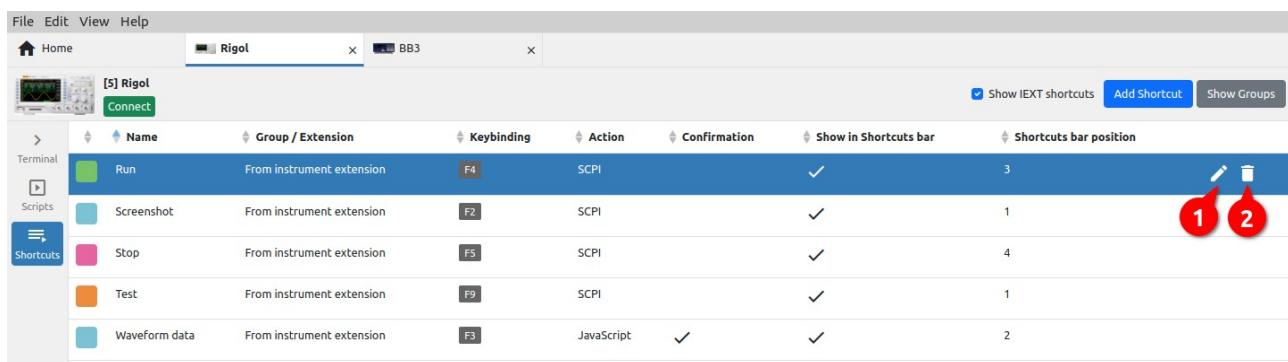


Fig. 40: Instrument shortcuts

| # | Option | Description |
|---|----------------------------|--|
| 1 | <i>Edit shortcut</i> | Editing the shortcut (see Section 7.4.1.) |
| 2 | <i>Delete shortcut</i> | Deleting an existing shortcut. |
| | <i>Show IEXT shortcuts</i> | Filters the display of Shortcuts belonging to the installed Instrument Extension (IEXT). |

Add Shortcut Adding a new shortcut opens the entry form as shown in Fig. 41.

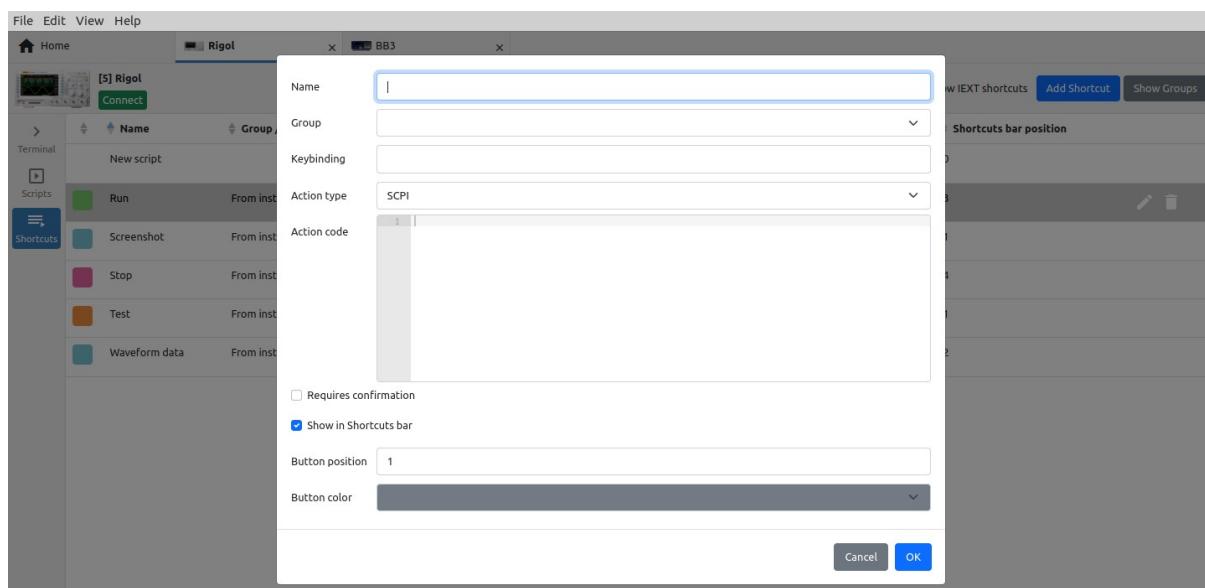


Fig. 41: Add new shortcut

7. Instrument activity bar

Show Groups / Show Shortcuts

Toggle between displaying a list of shortcuts and groups (Fig. 42) of shortcuts.

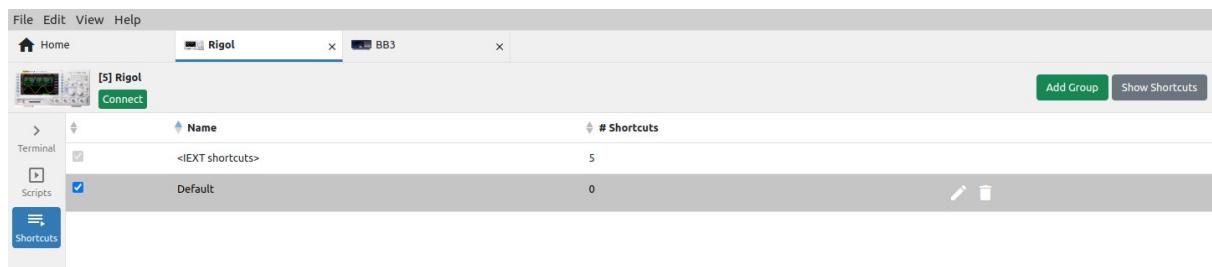


Fig. 42: Instrument shortcut groups

7.6. Lists

Lists are used to program parameters for instruments that support SCPI list commands. Lists for programming value and duration of output voltage and current for EEZ BB3 will be described below.

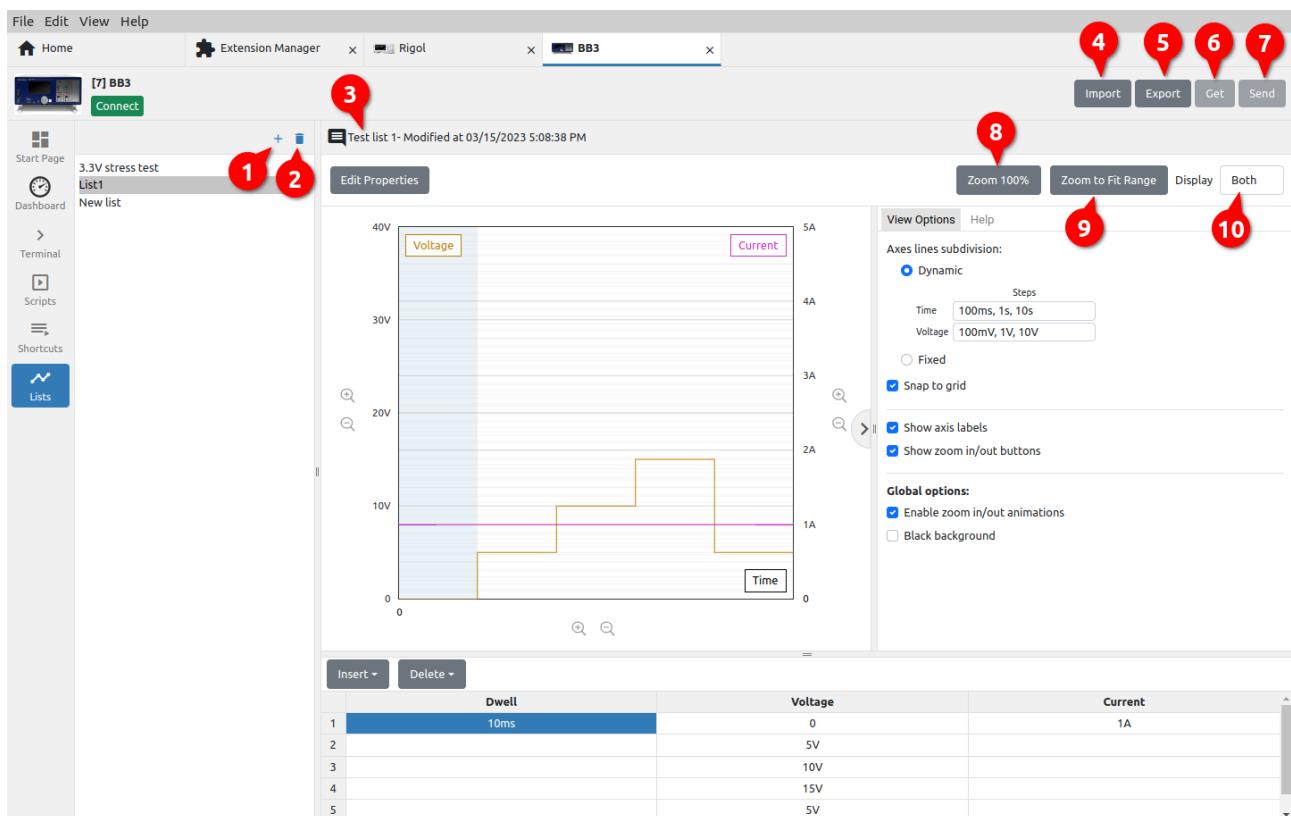


Fig. 43: Instrument programming lists

Option

1 *Add list*

Description

Creating a new list. The parameters of the list can be specified through a table (Fig. 43) or by defining envelope points that show the change of the parameter value over time.
In addition to the list *Type*, it will be necessary to define a *Name* and optionally a *Description*.

2 *Remove list*

Deleting the list (use *Undo* from the *Edit* menu to restore).

3 *List info*

List description and datetime of last changes.

4 *Import*

Import list from local storage. Opens a new dialog box for selecting the folder and name.

5 *Export*

Export list to local storage. Opens a new dialog box in which a list file can be selected.

6 Get

Receiving a list from the instrument. The option will be disabled if connection is not established with the instrument.
Opens a menu (Fig. 44) where you can choose the source (e.g. channel) from which the list will be received. For the imported list, it is necessary to enter the name and description (Fig. 45).

If the selected source does not have a defined list, an empty list will be imported.

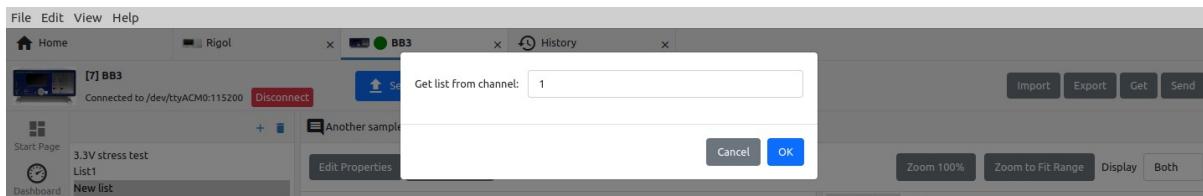


Fig. 44: List source selection

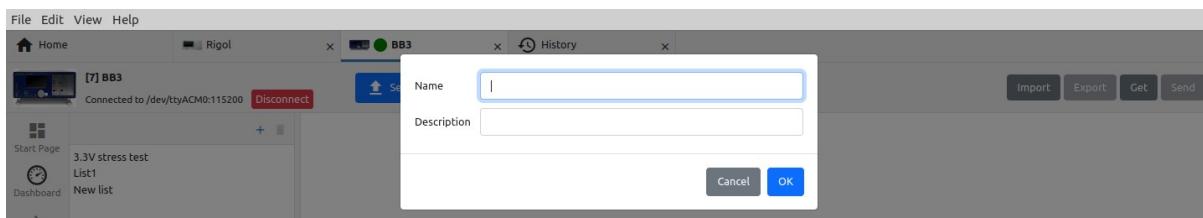


Fig. 45: Imported list parameters

7 Send

Sending the list to the instrument. The option will be disabled if connection is not established with the instrument.

8 Zoom 100%

Display graph without scaling.

9 Zoom to Fit Range

Graph display scaled according to the largest defined value.

10 Display

Selection of graphs to be displayed (e.g. voltage only, current only, both).

7.6.1. Editing a list using a table

Editing the list via the table is shown in Fig. 43. The program parameters graph is drawn simultaneously with editing the table at the bottom of the graph. In the case shown, the list contains two program parameters: *Voltage* and *Current*, for which values should be entered as well as duration (*Dwell*). To define the value, it is possible to use the units prefix, e.g. ms for dwell, mV for voltage, and mA for current.

In Fig. 46 and Fig. 47 shows all options for inserting new lines and deleting existing ones.

| Insert | Delete | Dwell | Voltage | Current |
|------------------|--------|-------|---------|---------|
| Insert row above | | 10ms | 0 | 1A |
| Insert row below | | | 5V | 2A |
| | | | 10V | 1A |
| | | | 15V | 1A |
| | | | 5V | 2A |

Fig. 46: Table insertion options

| Insert | Delete | Voltage | Current |
|--------|-------------------------------|---------|---------|
| | Delete row | 0 | 1A |
| | Clear column from cursor down | 5V | 2A |
| | Delete all from cursor down | 10V | 1A |
| | Delete all | 15V | 1A |
| | | 5V | 2A |

Fig. 47: Table deletion options

7.6.2. Editing a list using an envelope

In contrast to the previously mentioned editing of the list, where it is necessary to define program points through a table, envelope mode allows program points to be defined directly on the curve of the parameter being edited. This can simplify and speed up the whole process.

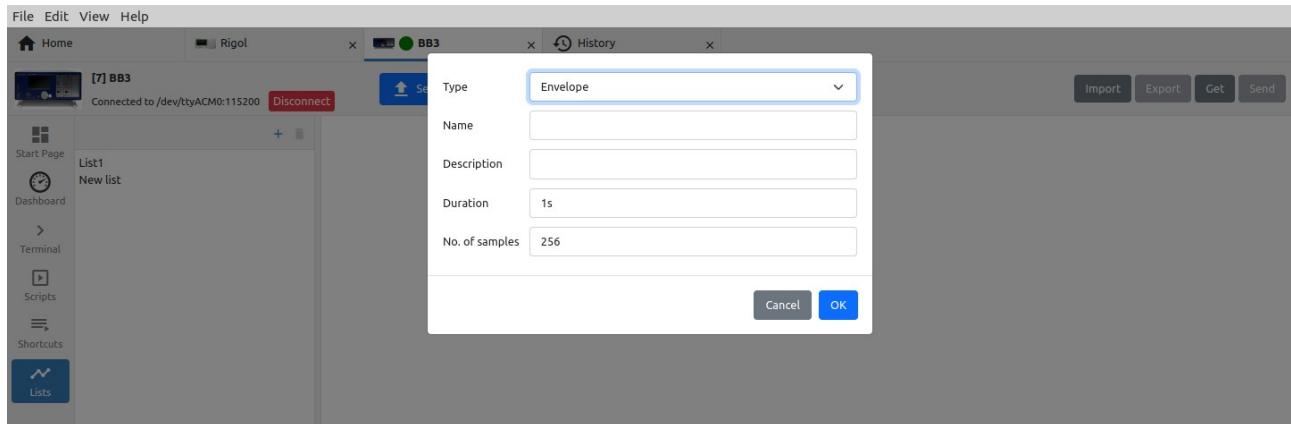


Fig. 48: Adding a new list in envelope mode

When creating a new list in envelope mode, it will be necessary to set two more parameters: the total duration of the program sequence and the number of samples (Fig. 48). The former is needed to be able to display the duration in the graph, and the later is needed to know how many points should be generated in total when sending the list to the instrument.

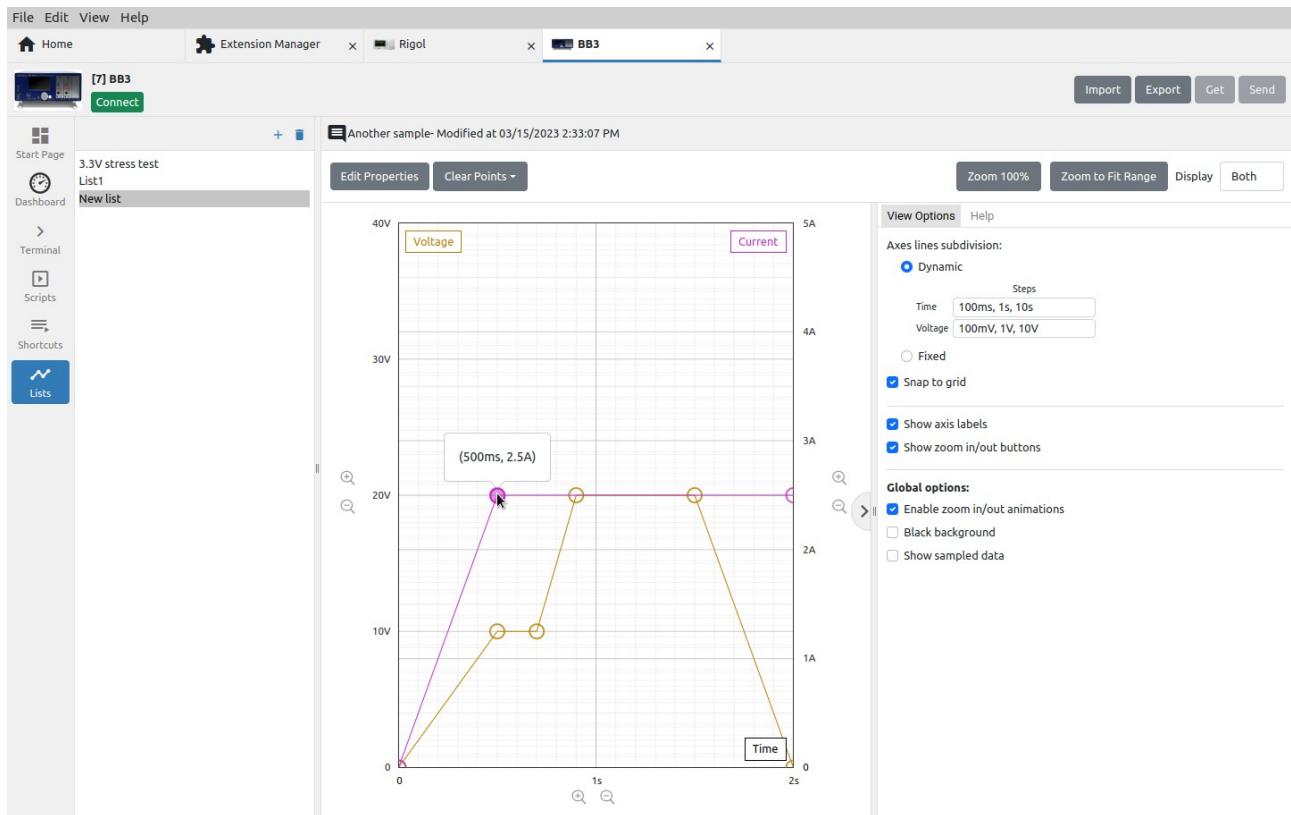


Fig. 49: Graph editing in envelope mode

The example in Fig. 49 contains 6 programming points for setting the voltage (light brown) and 3 for setting the current (magenta).

Adding a new point is simple: you only need to position the cursor somewhere in the graph and click, and a new point will appear, which will be automatically connected to two adjacent ones.

If we want to move the point in any direction, it will be necessary to position the cursor on it again and drag&drop it to a new position somewhere in the graph.

EEZ Studio User manual

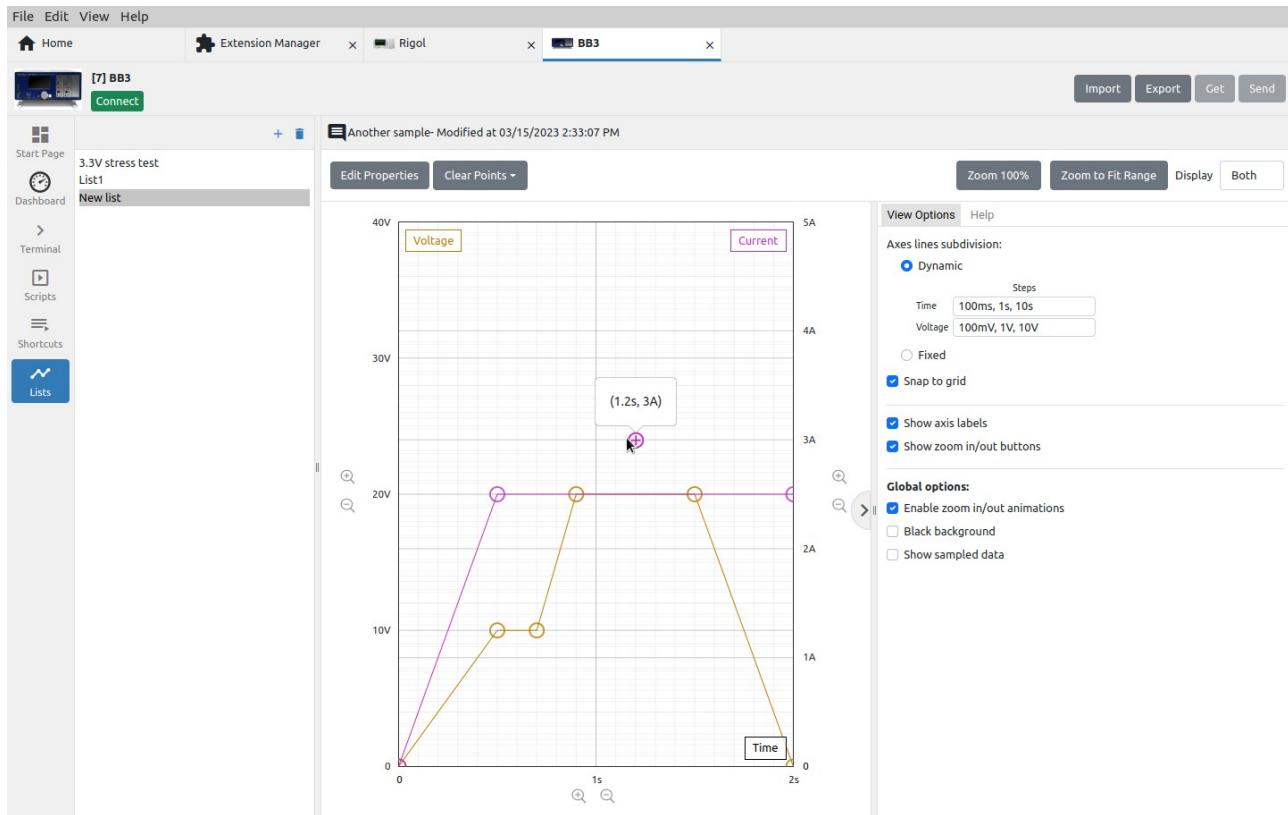


Fig. 50: Adding a new point in envelope mode

If you want to delete an existing point or manually edit its parameters after you have positioned yourself on it, you only need to click once more with the mouse when a dialog box will appear as shown in Fig. 51.

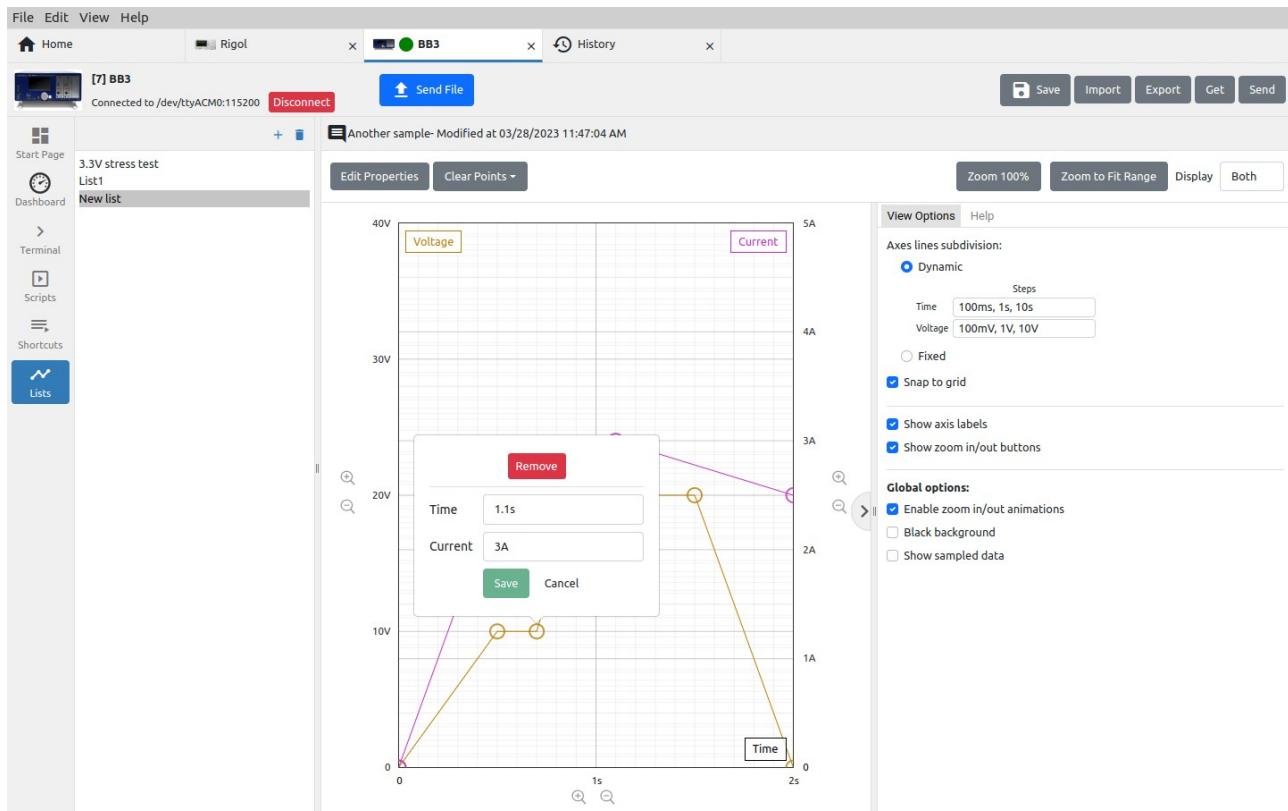


Fig. 51: Program point editing in envelope mode

7.6.3. List view options

The display of the graph can be dynamically changed (Fig. 43) depending on the resize of the window or the number of graticules can be fixed (Fig. 52).

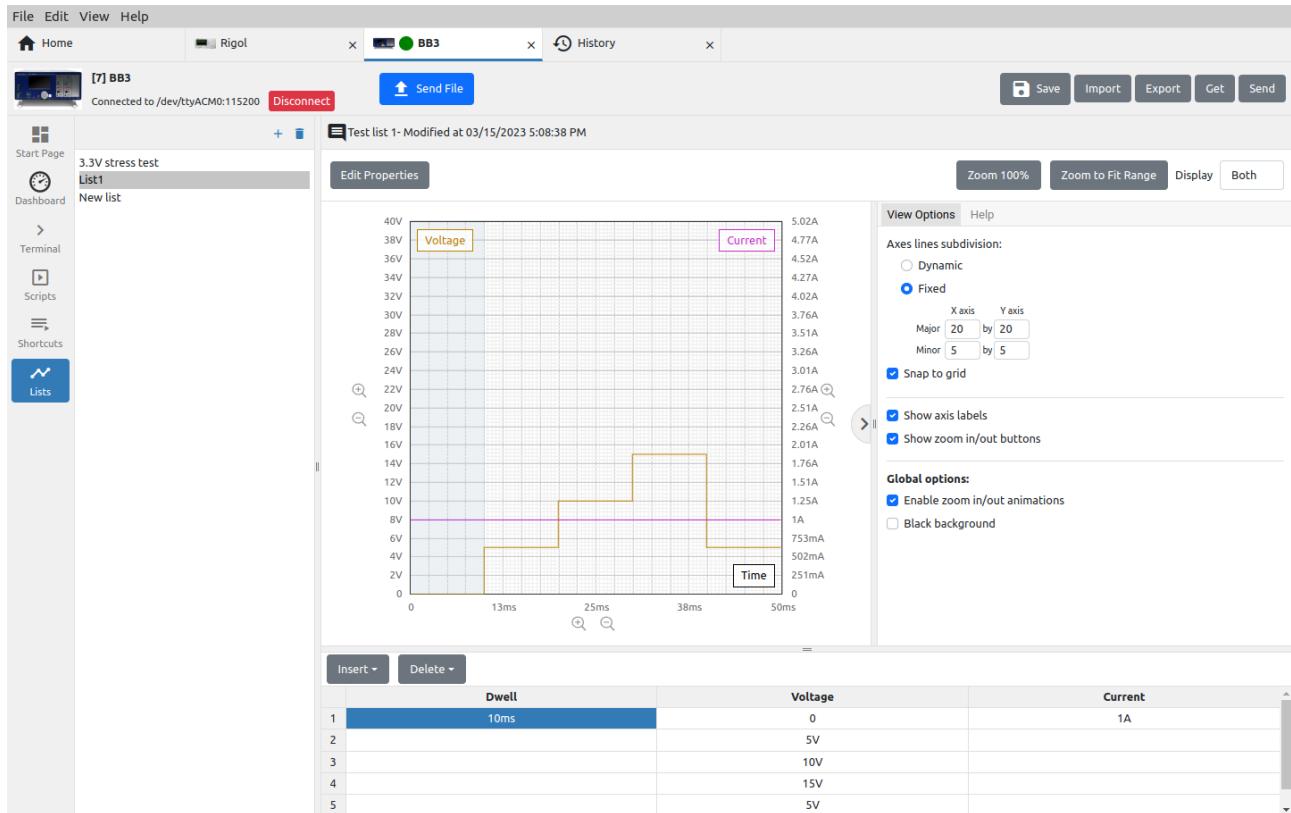


Fig. 52: Fixed graph view

7.6.4. List help

For zooming and navigating the graph, in addition to the zoom options located next to the x- and y-axes of the graph ("+" and "-" magnifier signs), a combination of mouse keys and control keys can be used. These additional options are shown in the Help tab as in Fig. 53.

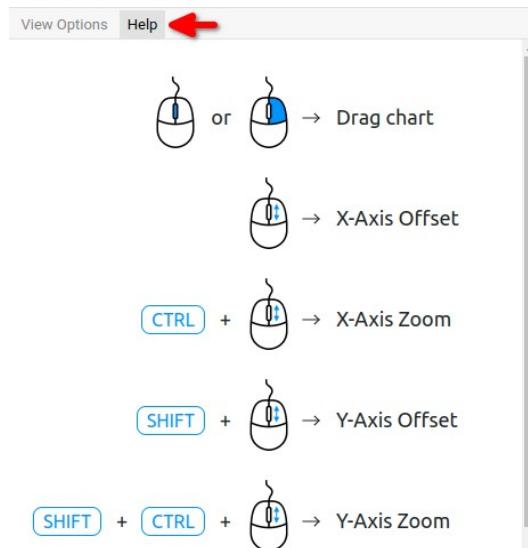


Fig. 53: Graph navigation and zoom help

*EEZ Studio
Project*

8. Home page project sections

One of the important features of EEZ Studio is that it enables the creation of projects for different target platforms using different technologies, which will be described below. The *Projects* section of the home page is shown in Fig. 54. which displays a searchable Recent Project List (RPL).

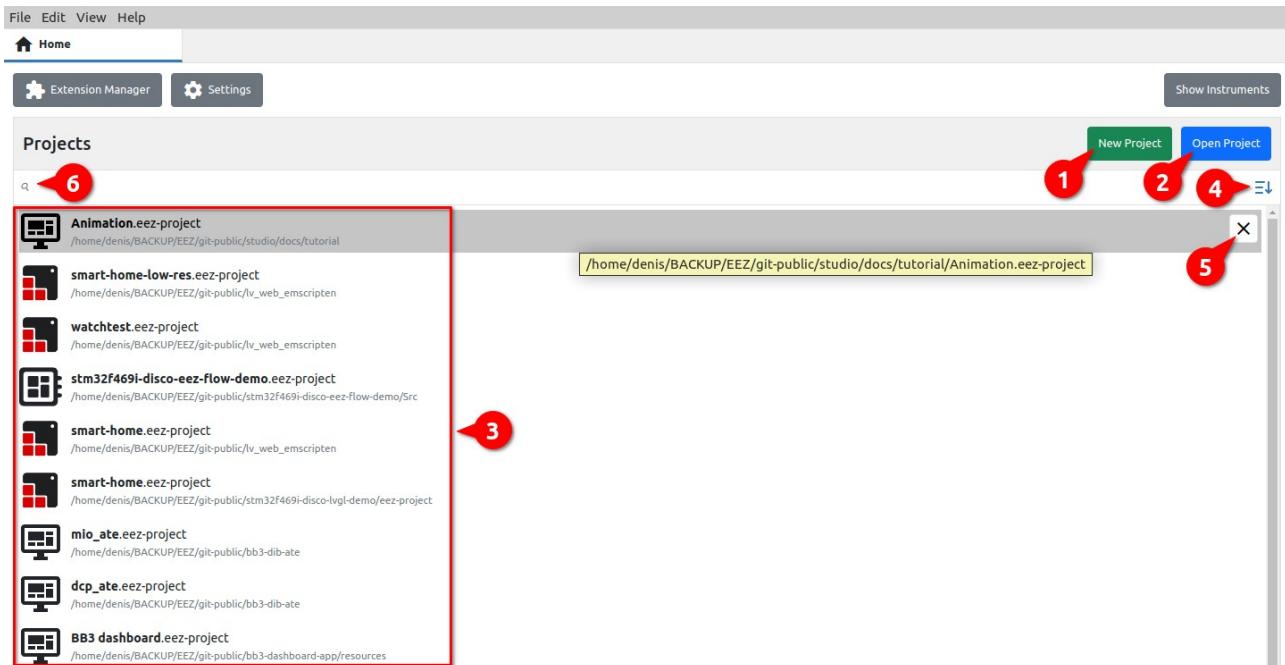


Fig. 54: Home page project options

| # | Option | Description |
|---|---------------------------|---|
| 1 | New project | Creating a new project. |
| 2 | Open project | Opening an existing project (will be added to RPL after successful loading). |
| 3 | Recent Project List (RPL) | List of all successfully loaded projects after the first run. |
| 4 | RPL sort order | Sorting order of projects in RPL: It can be <i>Show most recent first</i> or <i>Sort alphabetically</i> . |
| 5 | Remove from RPL | Removing the project from RPL. |
| 6 | Search RPL | RPL search by project name. |

8.1. EEZ Studio project types

EEZ Studio offers the creation of the following project types:

- **Dashboard** – desktop application. GUI applications can be quickly and easily created thanks to the drag&drop of available widgets and the import of multiple fonts and ready-made bitmaps prepared by the designer. The animation editor allows adding simple animations to the desired sections of the page or navigation between pages. Finally, the flowchart method of defining program logic instead of programming in one of the programming languages will further speed up prototyping and creation of the final application. The implemented debugger will shorten the application development process and help in more efficient error detection.
- **EEZ-GUI** – embedded GUI application that uses the EEZ-GUI framework. This is a native EEZ Studio framework that was initially developed to speed up and simplify embedded GUI development for [EEZ BB3](#) firmware.
- **LVGL** – embedded GUI application that uses LVGL (Light and Versatile Graphics Library) framework. LVGL is a popular open source project that supports a large number of target platforms. For more information visit <https://lvgl.io/>
- **Applet** – GUI application that can be run on EEZ BB3. Program logic is created using EEZ Flow (flowchart-based programming).

- **MicroPython script** – GUI application that can be run on EEZ BB3. Program logic is created using MicroPython scripting.
- **Templates from gitea repository** – Completed projects located in the gitea.io repository (mostly based on the EEZ-GUI framework). They can be used as a starting point for creating new projects.
- **Empty** – Creating an empty project for advanced users who want to configure everything themselves from the start.

8.2. Create new project

When creating a new project, the first step is a dialog as shown in Fig. 55.

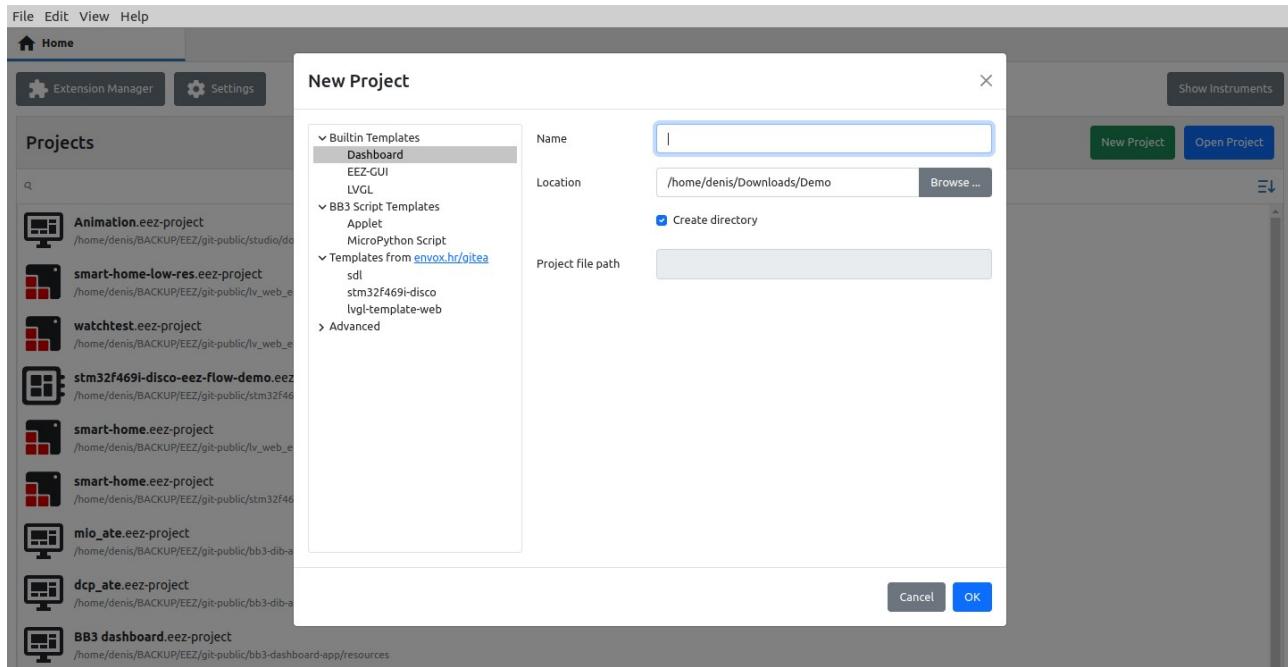


Fig. 55: Create new project

New project type

List of available project types as mentioned in the previous section.

Name

The name of the new project.

Location

The location where the project files will be stored.

Create directory

If selected, a subdirectory (at Location) with the name of the project will be created.

Project file path

Information field (read-only) showing the resulting path in which the new project will be created.

8.3. Additional steps for creating EEZ BB3 projects

New *Applet* and *MicroPython script* projects require access to the EEZ BB3 firmware master project from which exported styles, fonts and themes are used to make the GUI of the newly created application compatible with the EEZ BB3 on which it will be executed.

The necessary EEZ BB3 master project can be downloaded from GitHub (Fig. 56) when creating a new project or set a reference to a local copy of the repository (Fig. 57).

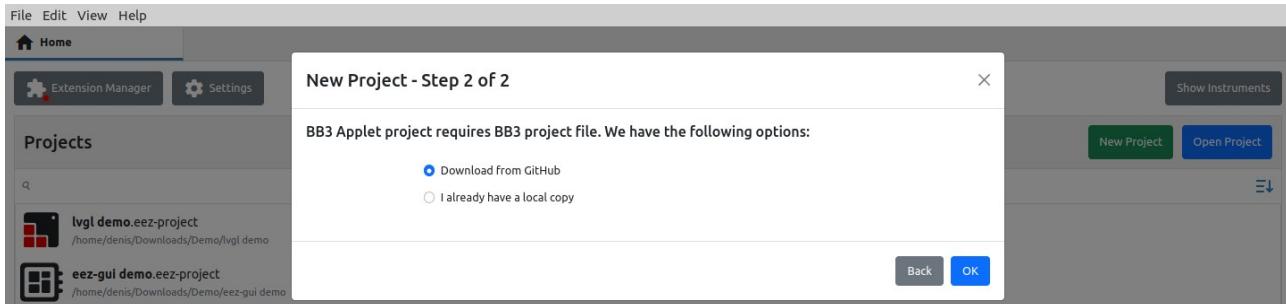


Fig. 56: Selection of EEZ BB3 project for a new project from GitHub

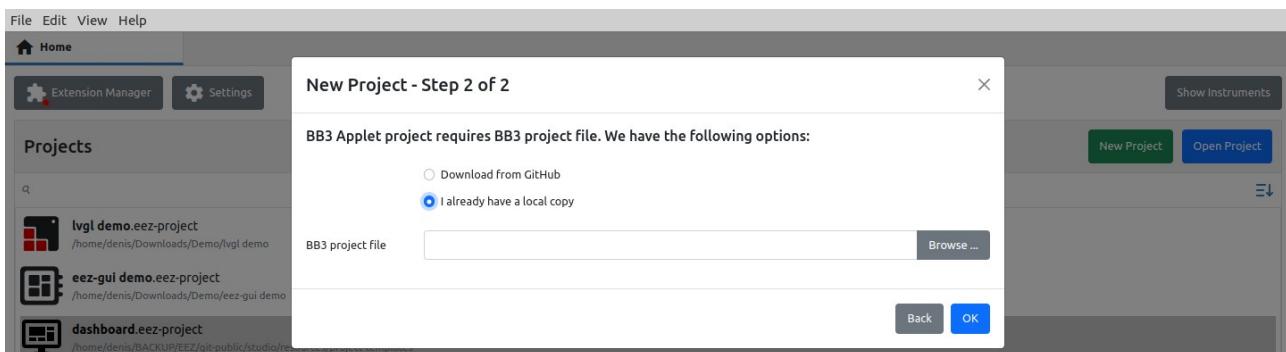


Fig. 57: Selection of EEZ BB3 project for a new project from the local repository

When creating a *MicroPython script* project, it will be necessary in the third step to define which firmware version is used on the target EEZ BB3 in order to create the corresponding resource file during the build (Fig. 58).

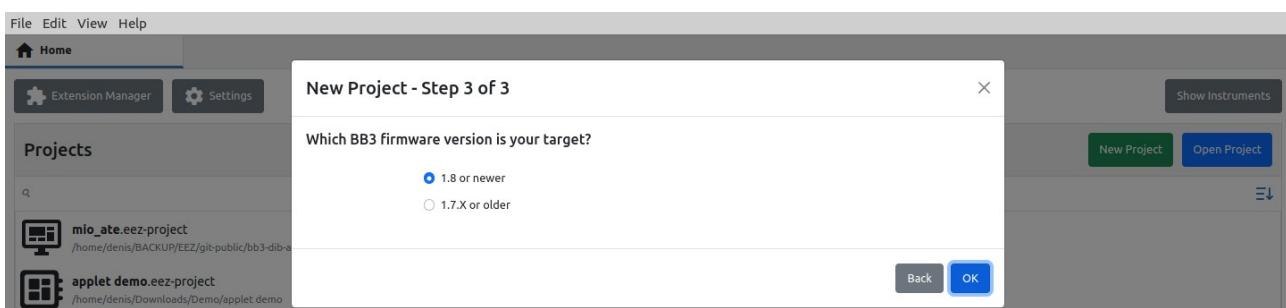


Fig. 58: Selection of EEZ BB3 firmware version for a new project

After all steps have been confirmed, the new project will be displayed in the project editor in *Edit* mode. Fig. 59 shows a new project for the *EEZ BB3 applet*. An overview of the project editor can be found in the next chapter.

The newly created project has the minimum required to be able to execute it in simulation (*Run* or *Debug* mode) or after build on the target platform.

EEZ Studio User manual

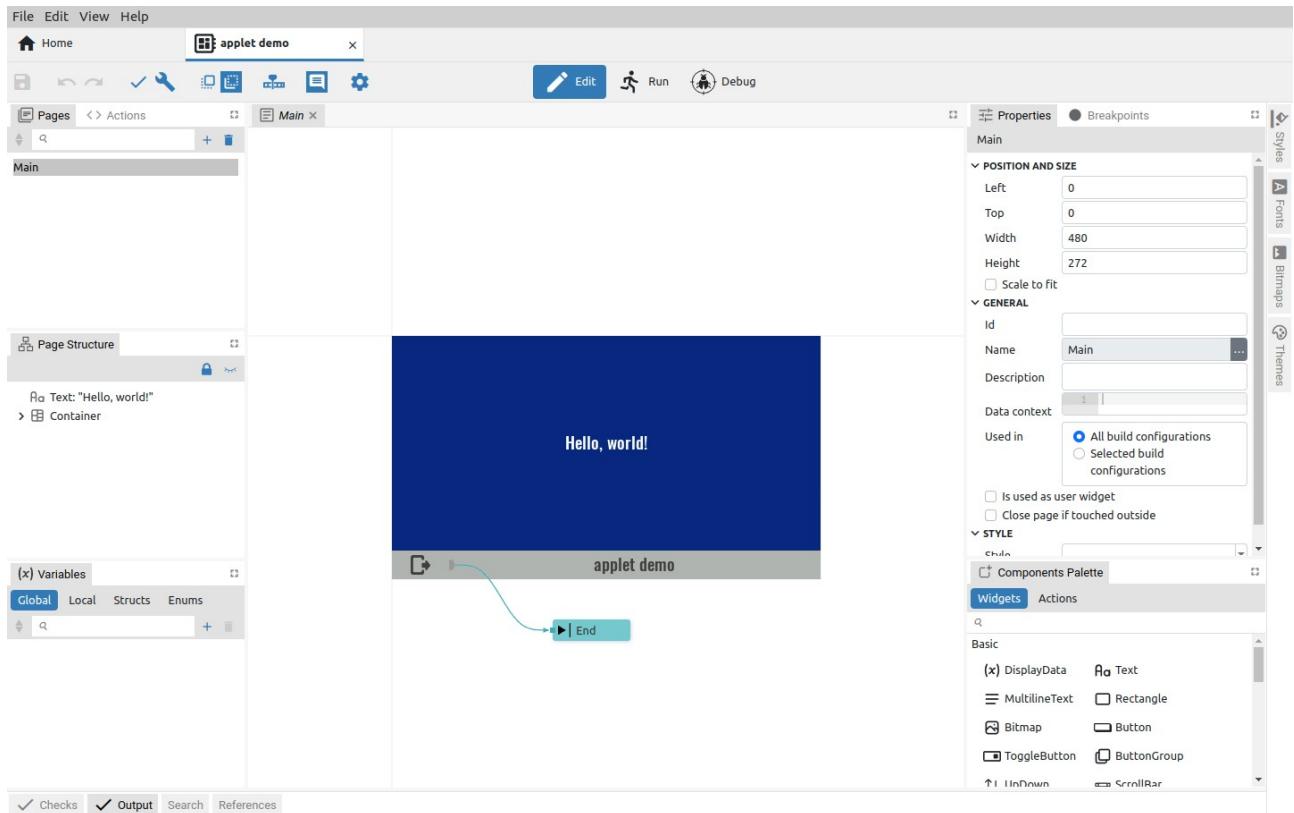


Fig. 59: Newly created project in Edit mode

The basic project settings set by the New project wizard can be seen by clicking on the *Settings* option (1) when the project *Settings* will open in a new tab (2) as shown in Fig. 60.

There you can also see *Project features* that have been added and are mandatory, so the *Remove* option is disabled (3), added and can be removed (4) and others that have not yet been added (5).

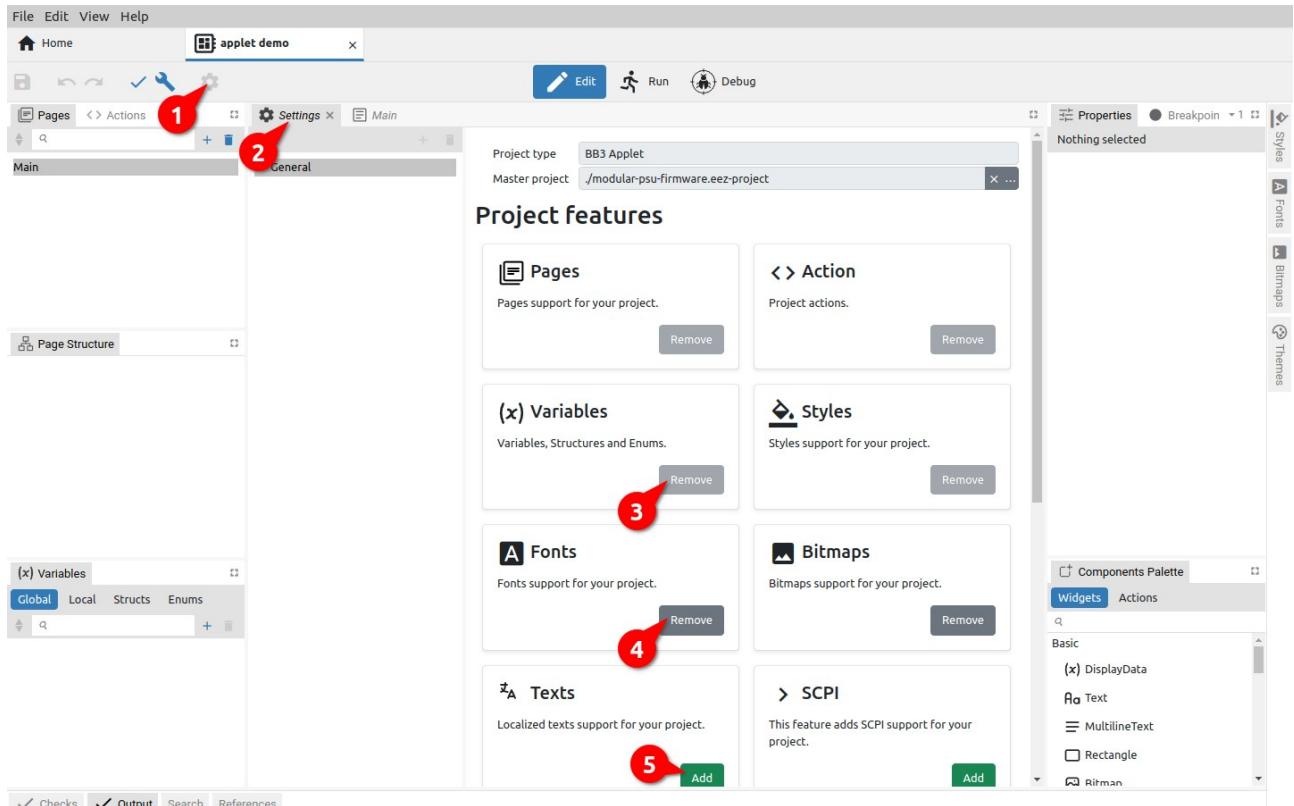


Fig. 60: Newly created project settings

For more info visit: www.envox.eu
File repository: <https://github.com/eez-open>

Version: M14 DRAFT
Date: 2023-05-11