Open Energy Profiler Toolset

Drive innovations in the field of low-powered technologies

Documentation

User Guide

30. June 2025

Revision History

Version	Date	Description
1.0	30.06.2025.	Initial draft

CONTENT

1. Introduction

The Open Energy Profiler Toolset (OpenEPT) is a powerful tool designed to support energy-aware development workflows by providing high-speed voltage and current measurement, along with the ability to generate arbitrary current profiles. With sampling rates of up to 1 MSps per channel and precise control capabilities, the system enables developers to accurately monitor, analyze, and optimize the power behavior of battery-powered and low-power embedded systems. This is particularly valuable for tasks such as battery profiling, power consumption optimization, and real-time energy debugging. By seamlessly integrating both data acquisition and controlled signal generation into a single, unified hardware module, OpenEPT serves as an indispensable tool for the design, validation, and optimization of modern embedded applications.

This documentation serves as a comprehensive technical guide for understanding, setting up, and fully utilizing the OpenEPT system from a user's perspective. It provides detailed instructions on hardware setup for first-time use, an in-depth overview of the graphical user interface (GUI), and step-by-step explanations on configuring workspaces and defining acquisition parameters. Users will also find guidance on configuring the EPP device, adjusting ADC settings, tuning performance parameters, and interpreting measurement results effectively. The GUI is designed to offer flexibility and modularity, supporting various hardware configurations and allowing customization of sampling periods, packet collection strategies, and plotting parameters to balance system performance and resource utilization.

Special emphasis is placed on real-time monitoring and data visualization. For example, the **Track Last Samples** feature enables live tracking of incoming data, which is essential for immediate system feedback but should be disabled for detailed data analysis such as zooming or panning. The system is capable of streaming data at up to 2 MSps (32 Mbps total), with raw data transmission over Ethernet ensuring high fidelity. However, plotting and storing such high volumes of data can place significant demands on CPU and memory resources, making careful tuning of parameters such as sampling period (**TS**), number of packets (**P**), and number of samples per packet (**N**) crucial for optimal performance.

Based on extensive testing on modern laptops (e.g., machines equipped with 11th generation Intel Core i-series processors, 32 GB of RAM, and 100 Mbps Ethernet), it has been shown that resource utilization is highly dependent on the chosen sampling rate and system configuration. Users are encouraged to monitor system load and adjust these parameters to ensure stable and efficient operation.

In addition to component-level descriptions, this documentation includes a dedicated section on **Typical Use Cases**, illustrating practical, step-by-step workflows that integrate all aspects of the system. These scenarios are designed to help users transition from understanding individual features to confidently applying them in real-world tasks. By following these guided examples, both novice and experienced users can maximize the capabilities of the OpenEPT system, ensuring reliable data acquisition, efficient analysis, and effective optimization of their devices.

Ultimately, this documentation aims to bridge the gap between theoretical configuration and hands-on usage, empowering users to fully leverage the advanced functionalities of OpenEPT in their development and research projects.

2. Setup hardware

The equipment required to connect OpenEPT EPP Schield board with host machine is presented on Figure 2.1.

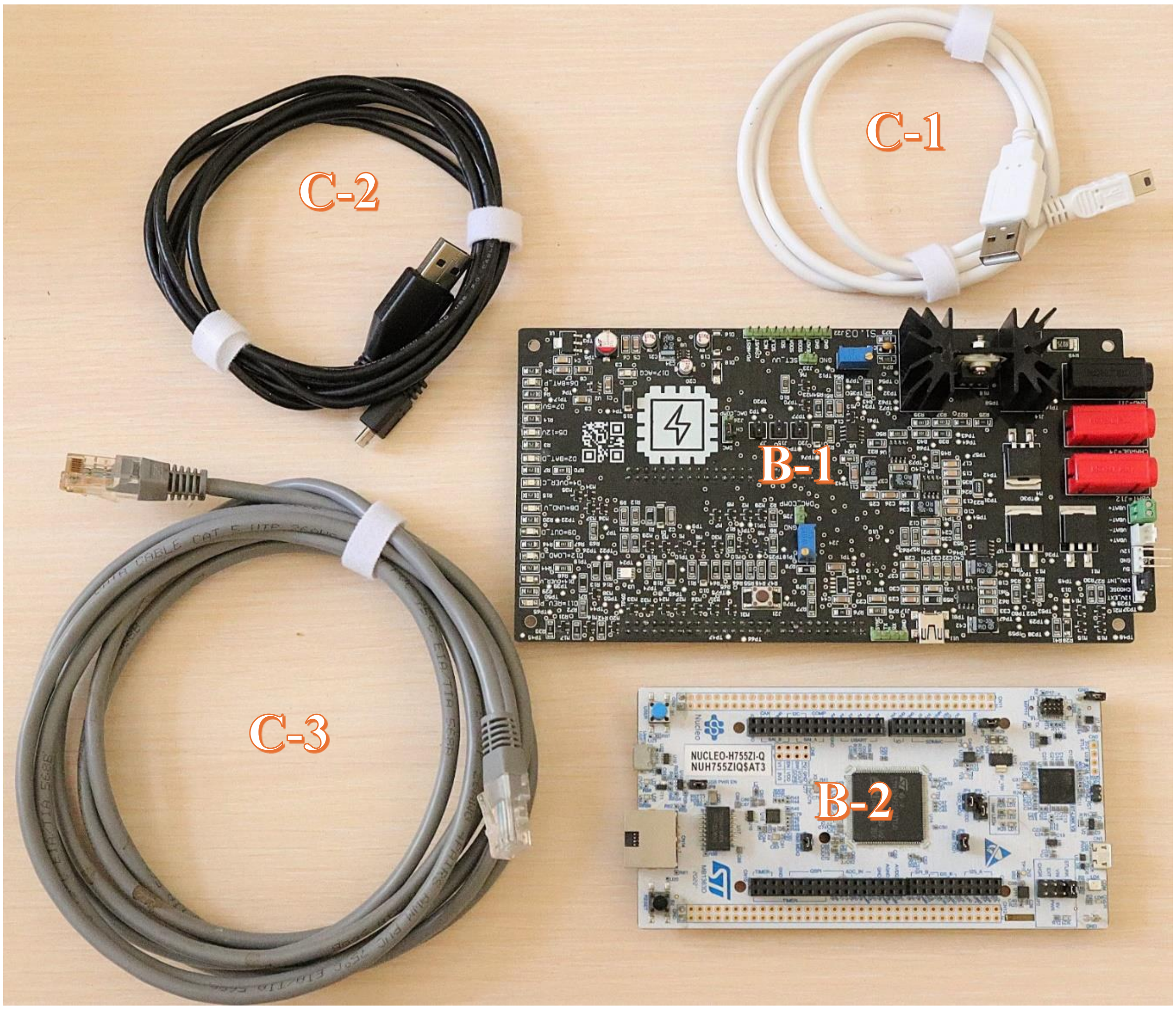


Figure 2.1 - Hardware components

Components presented on Figure 2.1 are:

(**C-1**) USB Cable Type A to Mini USB – Used to power OpenEPT EPP Shield board
(**C-2**) USB Cable Type A to Micro B – Used to connect Nucleo-H755ZIQ board with Host machine for programming
(**C-3**) LAN Cable – To connect Nucleo-H755ZIQ board with Host machine for data stream
(**B-1**) OpenEPT EPP Schield board and Nucleo-H755ZIQ development board.
Instead of a Schield board, Full board can be used
(**B-2**) Nucleo-H755ZIQ Development board

Following steps are required to make board fully operational:

Step 1: Connect OpenEPT EPP Schield board with Nucleo Board

Connect OpenEPT EPP Schield Board (**5**) with Nucleo Board (**4**) as it is illustrated on Figure 2.2.

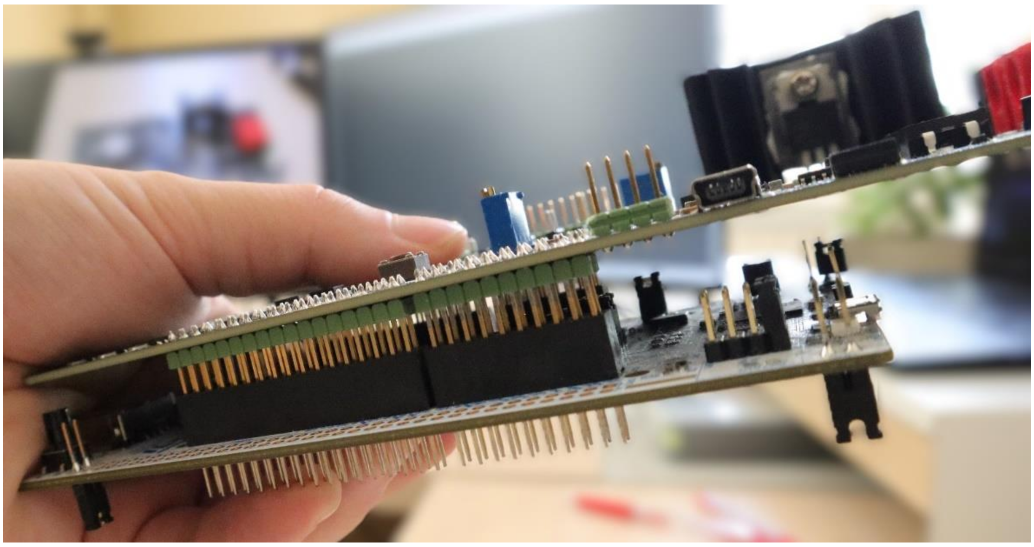


Figure 2.2 - Connect OpenEPT EPP Schield Board with Nucleo Board

Once the boards are connected, the top view of the OpenEPT EPP board appears as shown in the Figure 2.3.

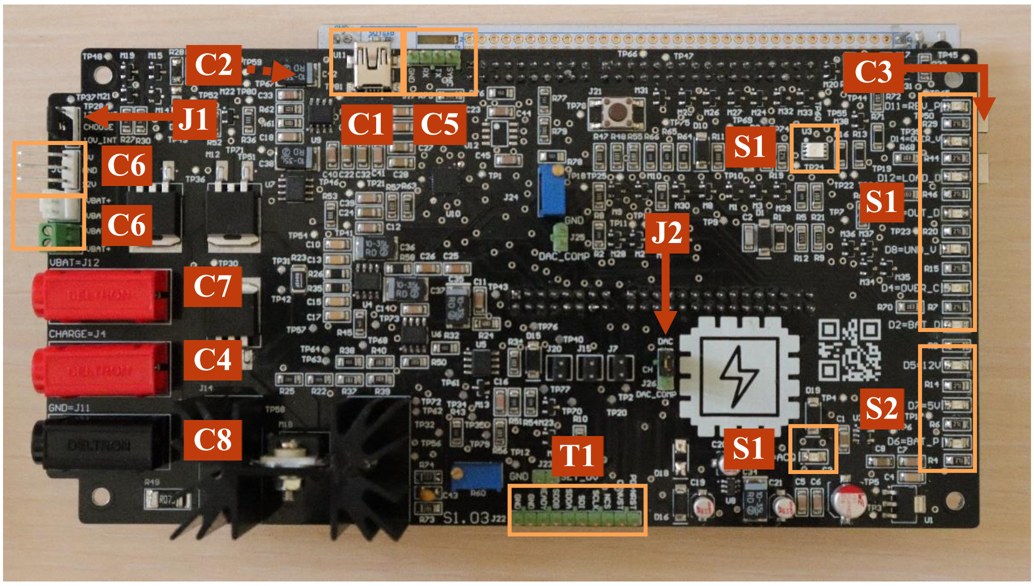


Figure 2.3 - Top view of the OpenEPT EPP board highlighting key components.

Name of the key components of the OpenEPT EPP board are listed in Table 2.1.

Table 2.1 - OpenEPT EPP Board's Components

C1	5V Power Supply	**C2**	Nucleo USB Connector (*visible from bottom side*)
C3	RJ45 Connector	**C4**	Charger connector
C5	Energy Debugging Interface	**C6**	Battery connectors
C7	Load connector	**C8**	Load and Charger Ground
T1	ADC Test Points		
J2	ADC Selector: Int or **Ext** (*default*)	**J1**	Power supply selector: **5V** (*default*) or 12V
S2	Power supply indicators	**S1**	Protection and switches status

Step 2: Connect OpenEPT board with Host machine

Take the Micro-B connector of the USB cable (**C-2**) shown in Figure 2.1 and connect it to the **C2** connector on the OpenEPT EPP board, as illustrated in Figure 2.4. Then, connect the other end of the cable to your PC. This cable will be used to program the STM32 Nucleo board and to monitor logs sent from the OpenEPT EPP board.

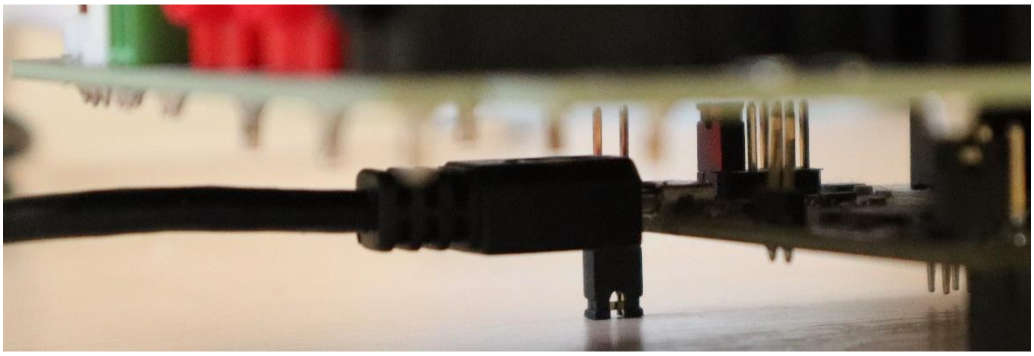


Figure 2.4 - OpenEPT EPP Board with connected cable for programming STM Nucleo board

Step 3: Obtain the latest firmware binary

Visit the official project website, navigate to the **Download** section, and click the download button to obtain the latest OpenEPT EPP firmware binary.

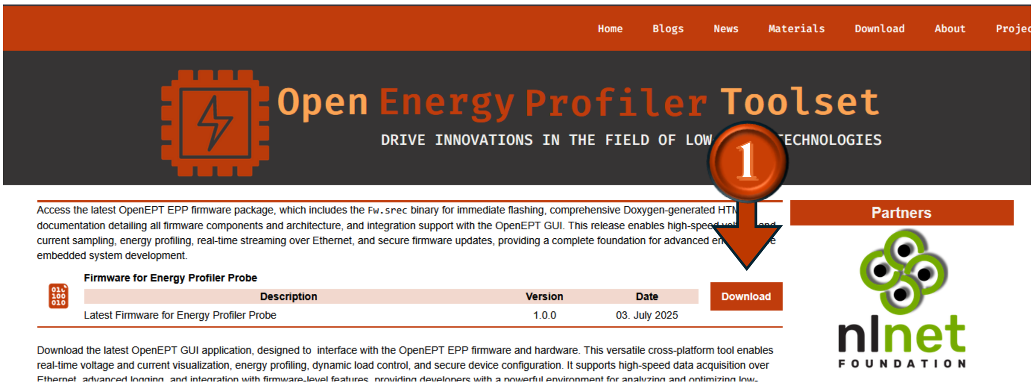


Figure 2.5 - Page for downloading the OpenEPT EPP firmware binary

After clicking the **Download** button, you will be redirected to the GitHub Releases page, where you can download the *OpenEPT_Fw_<latest version>.srec* binary file.

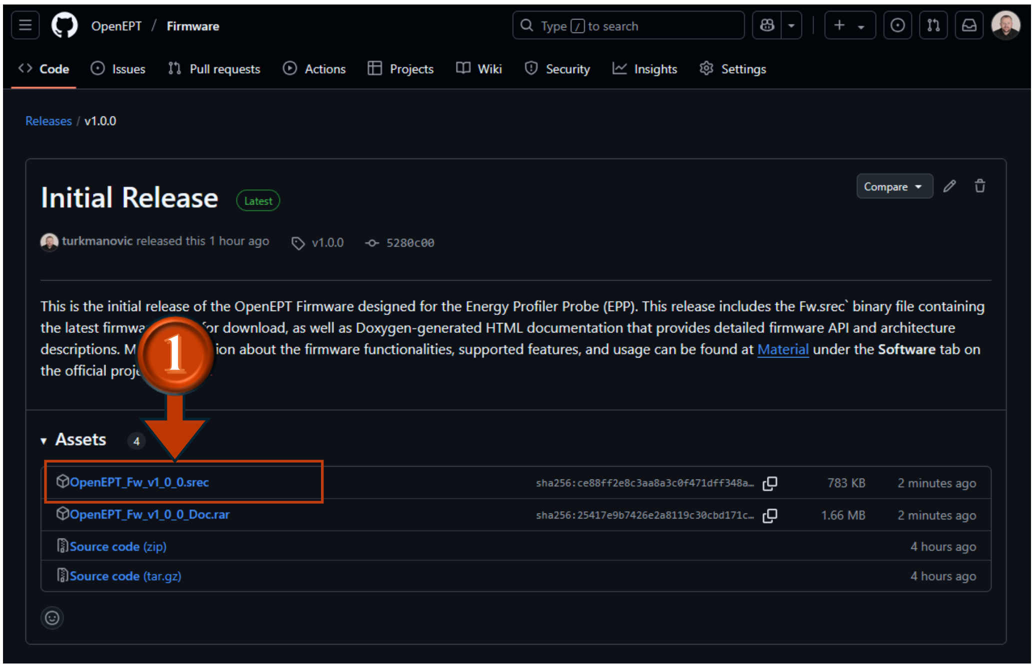


Figure 2.6 - Latest OpenEPT EPP Firmware release on GitHub

Step 4: Flash the firmware onto the STM32 board using STM32CubeProgrammer

After connecting the OpenEPT EPP board to your PC, open the STM32CubeProgrammer application. In STM32CubeProgrammer, click Connect to establish a connection with the OpenEPT EPP board (**1** in Figure 2.7).

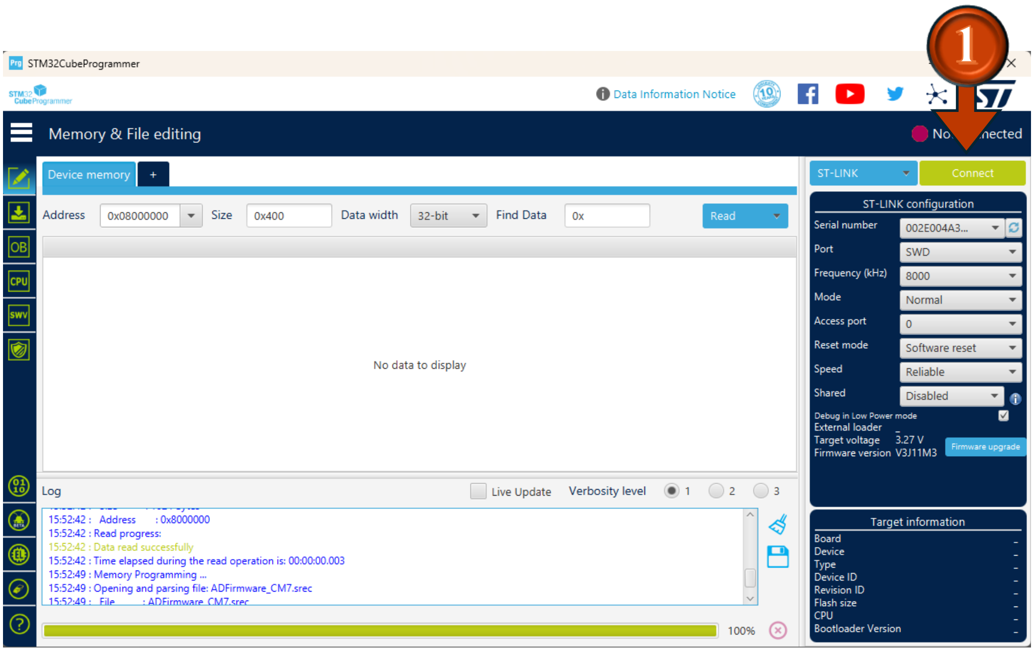


Figure 2.7 - STM32Cube Programmer after start

Once the connection is successfully established, click the button on the left side of the window to navigate to the page for erasing and programming the flash memory of the OpenEPT EPP's STM32 Nucleo board (**1** in Figure 2.8).

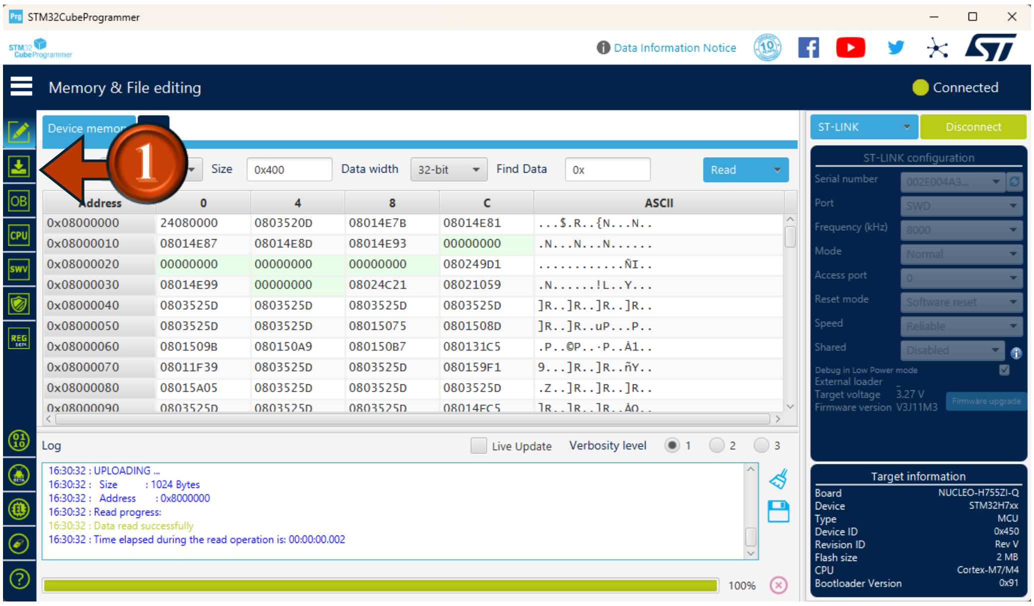


Figure 2.8 - STM32CubeProgrammer after connection with the board is established

As the first step in the erasing and programming process, perform a full chip erase (**1** in Figure 2.9). This is important to ensure that all bits in the flash memory are set to their default values before programming. Once the erase is complete, click the **Browse** (**2** in Figure 2.9) button and navigate to the previously downloaded .srec file. Finally, start programming the board by clicking the **Start Programming** button (**3** in Figure 2.9).

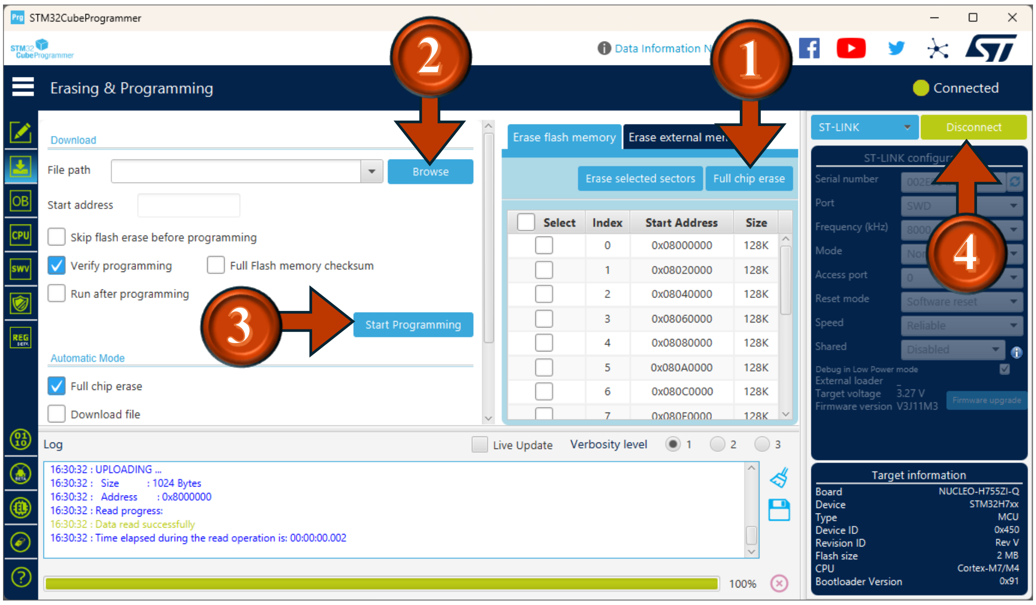


Figure 2.9 - Erase and programming

After programming is successfully completed, click the Disconnect button (**4** in Figure 2.9) and ***unplug the USB cable from (C-2) the PC***. Once this is done, you are ready to proceed to the next step.

Step 5: Power on OpenEPT Schield board, Connect LAN cable and Reset MCU

After board is successfully programmed, connect:

1. Connect OpenEPT EPP Nucleo Board with PC via USB cable (**C-2**) (see Figure 2.1) connected to **C2** (see Figure 2.3)
2. Power OpenEPT EPP Schield Board with USB cable (**C-1**) (see Figure 2.1) connected to **C1** (see Figure 2.3)
3. Connect LAN cable (**C-3**) (see Figure 2.1) to OpenEPT EPP Nucleo Board **C3** Connector (see Figure 2.3)
4. [*Optional*] Open Serial Terminal and connect to STLink COM port. Communication parameters are: 115200-8N1
5. Reset OpenEPT board (**1** in Figure 2.10)

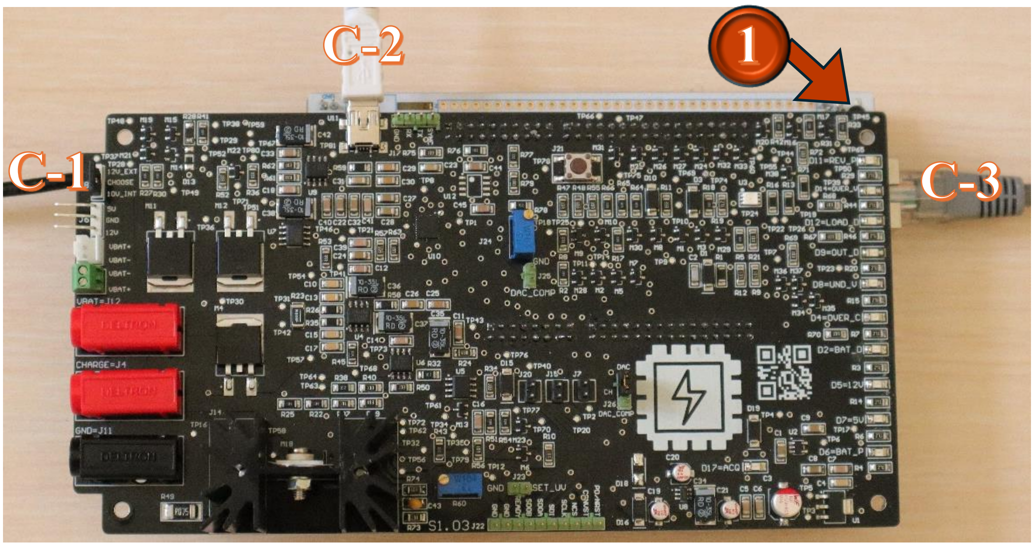


Figure 2.10 - OpenEPT EPP Board after all cables are connected

After the board is reset and all cables are connected as shown in Figure 2.10, the device begins sending messages indicating the initialization status of its services (refer to the Firmware Developer Guide for more details).

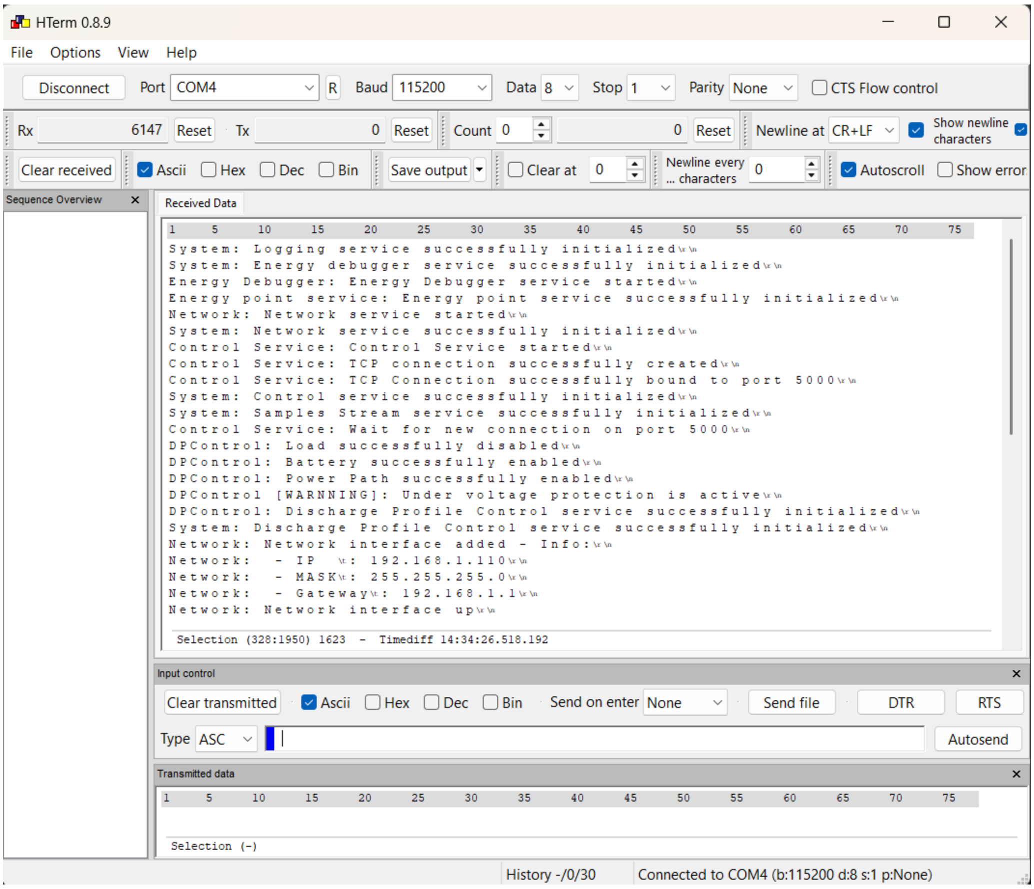


Figure 2.11 - Log messages after successful startup of the OpenEPT device

If the device is successfully flashed, all drivers and services are properly initialized, and it is ready for use, two blue LEDs on the board will turn on, as shown in the Figure 2.12.

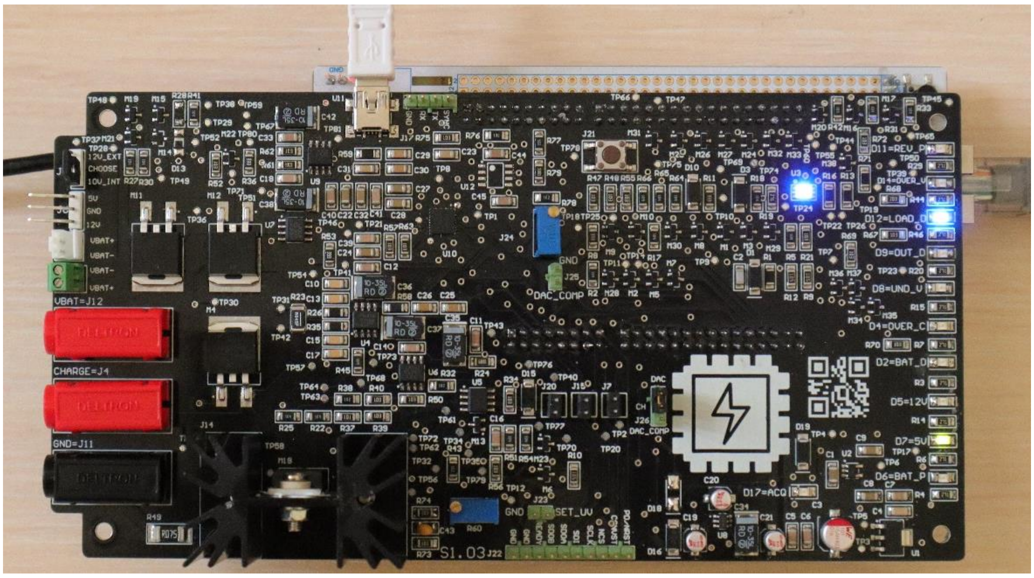


Figure 2.12 - The two blue LEDs indicate that the OpenEPT EPP is ready for use.

3. GRAPHICAL USER INTERFACE

3.1. Download and run

Visit the official project website, navigate to the **Download** section, and click the download button to obtain the latest OpenEPT GUI application installer.

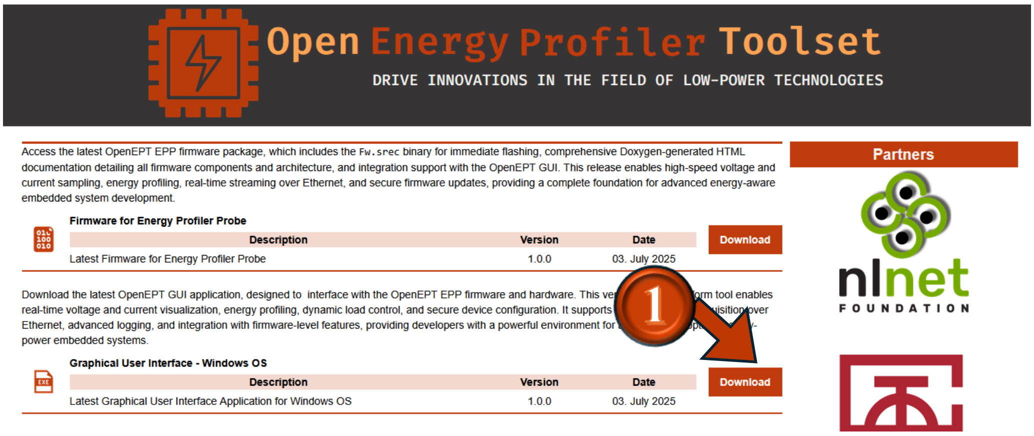


Figure 3.1 - Page for downloading the OpenEPT GUI Installation

After clicking the download button, you will be redirected to the GitHub Releases page, where you can download the *OpenEPT_GUI_v<latest version>_Install.exe* installer file.

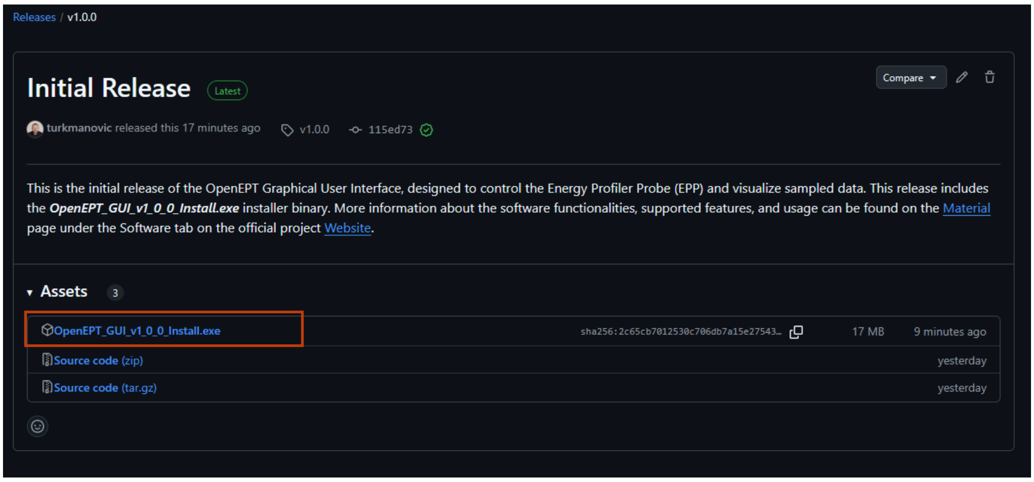


Figure 3.2 - Latest OpenEPT GUI release on GitHub

After installer is downloaded, you can run it and install OpenEPT GUI Application by following installation steps.

3.2. Define Workspace

When the OpenEPT GUI application is launched, the first window that appears prompts the user to specify the workspace path. The layout of this window is shown in **Figure 3.3**.

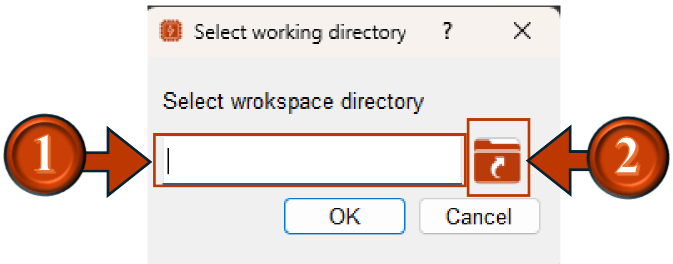


Figure 3.3 - Workspace directory selection

Within this window, the user can either manually enter the working directory path (**1**) or select it by clicking the directory selection button (**2**). When the workspace is defined, the user can proceed to the main application by clicking OK.

The workspace is a directory where all information related to the current acquisition session is stored. In the current software version, the workspace is only required if the sampled acquisition data will be saved. Otherwise, the workspace path can be left blank, and the user can proceed to the application by clicking ***OK****.*

3.3. Add New Device

A new device can be added by clicking the symbol ⊕ in the main OpenEPT GUI application, as shown in Figure 3.4.

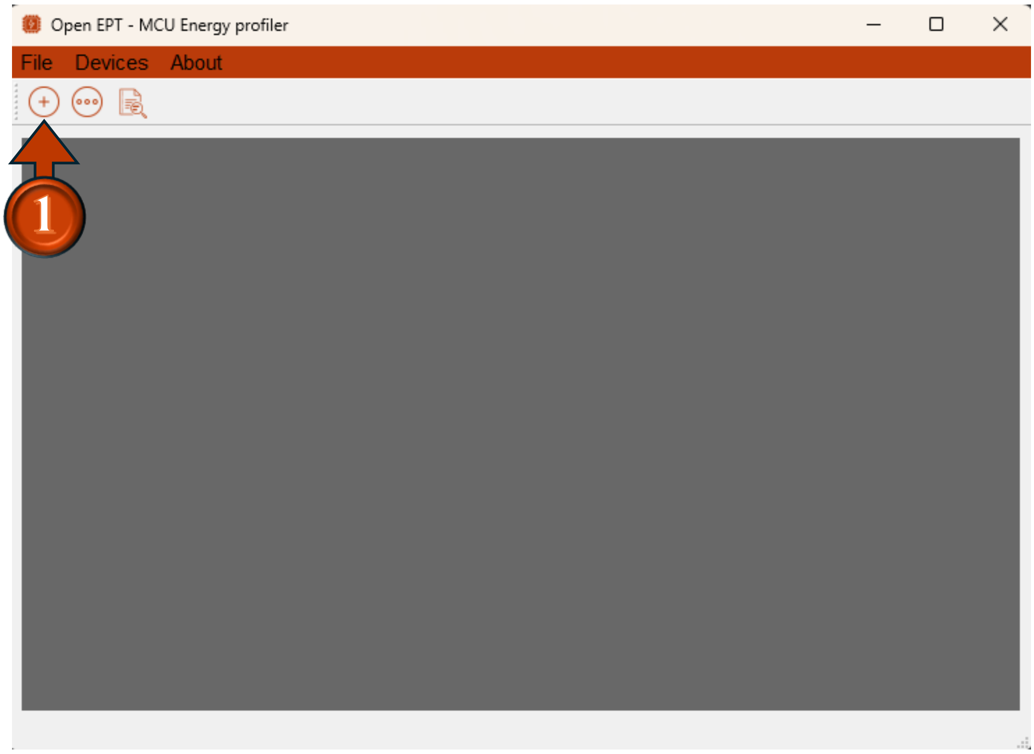


Figure 3.4 - OpenEPT's main GUI layout without added device

After clicking the symbol ⊕ in the main OpenEPT GUI application (**1** in Figure 3.4), new dialog is opened (Figure 3.5) where we should enter device IP address and port for communication. The default IP address for the EPP device is 192.168.1.100, and the default port is 5000 (see [2] for instructions on how to change the IP address and compile a new raw binary).

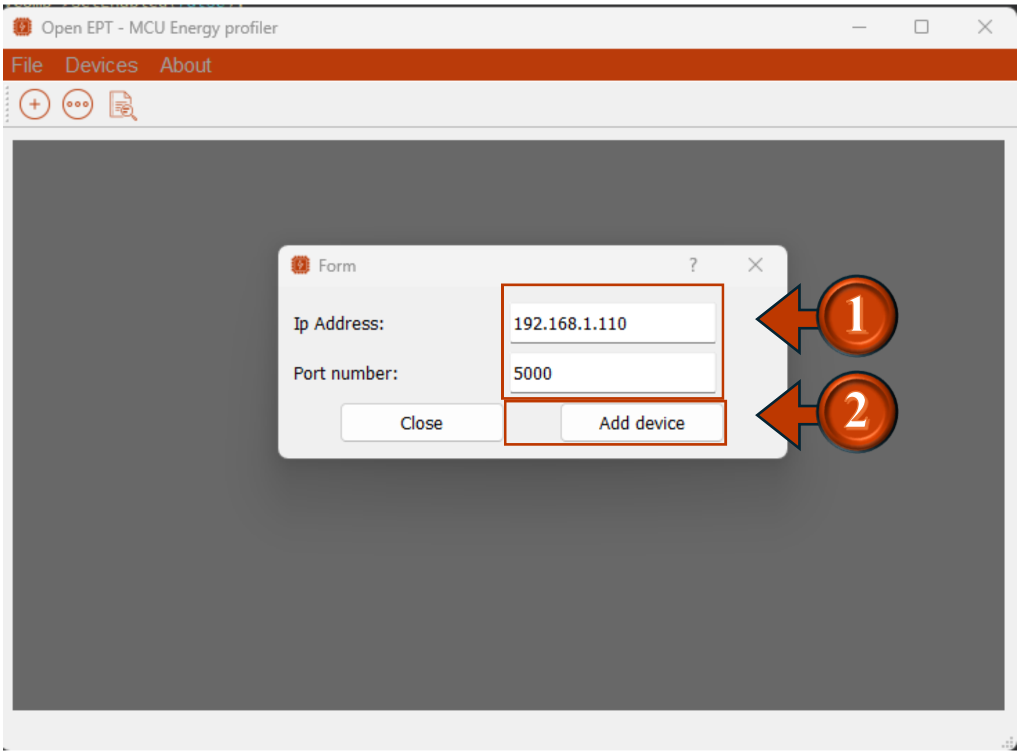


Figure 3.5 – Add Device Dialog

After entering the device IP address (**1** in Figure 3.5), the user can add the new device by clicking Add Device (**2** in Figure 3.5). If the device is added successfully, a dialog window stating "Device successfully added" will appear, as shown in Figure 3.6.

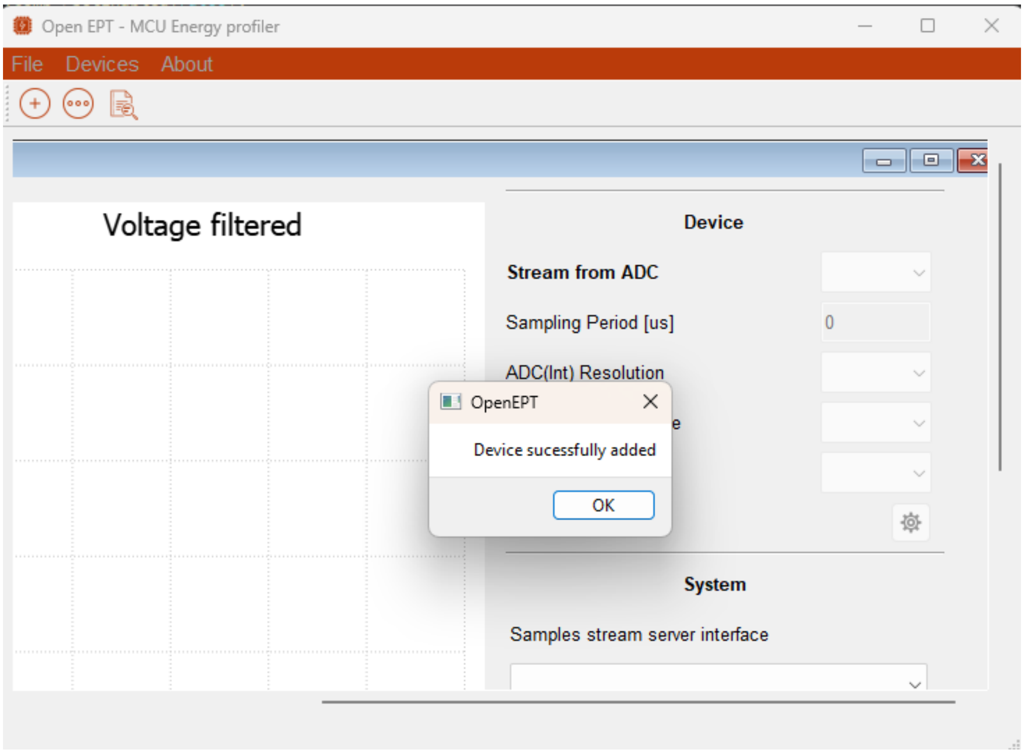


Figure 3.6 - Device is successfully added

3.4. Device Control

Device control is managed through the **Device Window**, shown in Figure 3.7, which is created after a device is successfully added. This window consists of three main sections:

1) **Plot**
   Includes Voltage, Current and Consumption plotting.
2) **Log**
   Presents messages from EPP or GUI
3) **Configuration**
   Includes all options used to configure device

Figure 3.7 - Device Window

3.4.1. Plot

The plot section consists of three subwindows: Voltage, Current, and Consumption. The Voltage subwindow displays samples acquired from the EPP's voltage channel, expressed in volts. The Current subwindow shows samples from the EPP's current channel, expressed in milliamperes (mA). The Consumption subwindow presents values calculated from the voltage and current data, representing the system's consumption.

Each subwindow shares the same layout, as shown in Figure 3.8. It consists of two main sections: (**1**) the plot control section and (**2**) the graphical display section. The graphical section displays the samples, while the control section is used to manage the plot. The available controls (listed from top to bottom) include:

- Zum In
- Zum Out
- Expand graph to show all sampled values
- Select specific part of the graph
- Move graph
- Track last samples

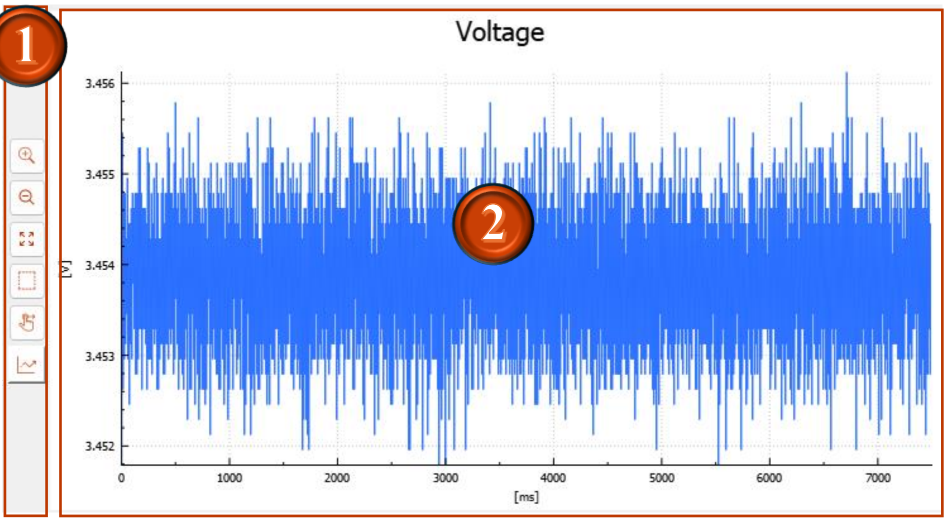
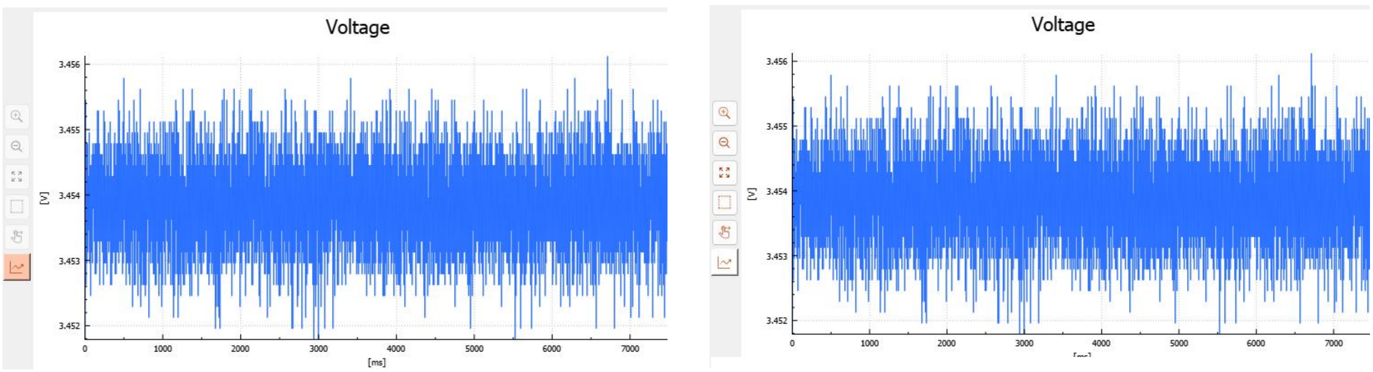


Figure 3.8 - Plot subwindow

The *Track Last Samples* control enables or disables all other plot controls. When this option is active (Figure 3.9-**a**), it allows real-time monitoring and live visualization of incoming samples. However, if the user wants to analyse the data using features such as Zoom in, Zoom out, and other navigation tools, the track option should be disabled (Figure 3.9-**b**).



a) b)

Figure 3.9 - Plot subwindow with a) enabled tracking option; b) disabled tracking option

3.4.2. Log

The Log section displays all activities relevant to the user that occur during program execution. There are two sources of messages: Application and Device. When the source is Application, the message is generated by the GUI itself, whereas messages from Device represent asynchronous notifications sent from the EPP. Both sources support three message levels: Info, Warning, and Error. Info messages provide general information, Warning messages indicate issues that do not affect software execution, and Error messages are critical and can be fatal to program operation. Each message type has its own text and background colour for clear identification. Figure 3.10 illustrates a typical example of the Log section after the device has been connected and configured.

```
[Application] [16:00:42]:Ep link ( port=8000 ) sucessfully created:
[Application] [16:00:42]:Sampling time sucessfully obained and presented
[Application] [16:00:42]:Load state sucessfully obained and presented
[Application] [16:00:42]:Load current sucessfully obained and presented
[Application] [16:00:42]:Battery state sucessfully obained and presented
```

Figure 3.10 - Log window

3.4.3. Configuration

The Configuration section is used to set up the EPP device and acquisition parameters within the GUI application. The layout of this section is shown in Figure 3.11, and its main options are described in Table 3.1.

Some options in this section are intended solely for development purposes and are not recommended for use in regular working scenarios. The rows corresponding to these options are highlighted in colour .

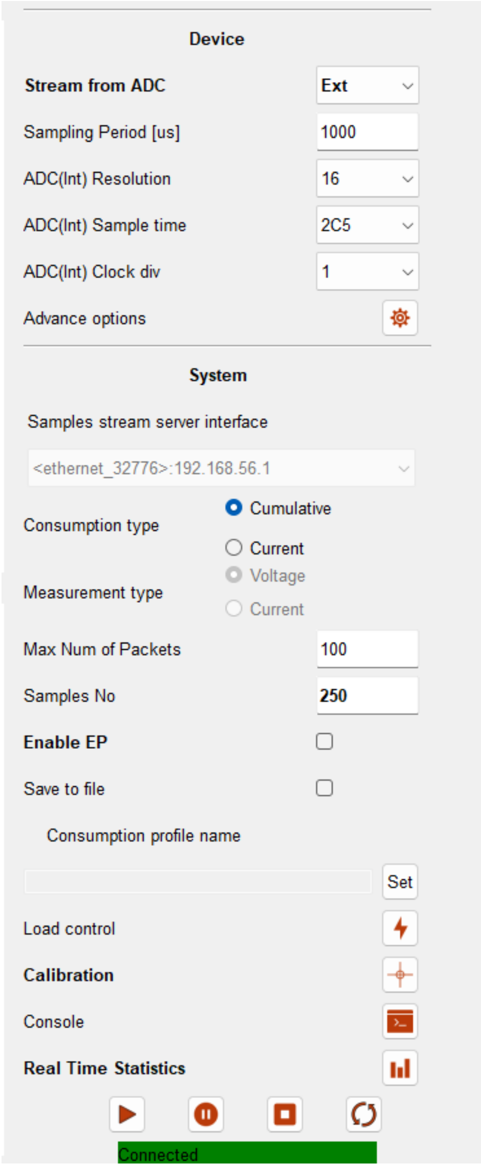


Figure 3.11 - Configuration section

Table 3.1 - Configuration Options

Name	Value	Description
Stream from ADC	Int[1], Ext	Selects whether to use the internal or external ADC for acquisition.
Sampling period	[1, x] (μs)	Specifies the sampling period, expressed in microseconds.
ADC (int) Resolution	10, 12, 14, 16	Sets the resolution of the internal ADC.
ADC (int) Sample time	[1, x]	Defines the sample time for the internal ADC.

[1] The internal ADC is intended solely for development purposes and is not meant to be used during regular operation.

ADC (int) Clock div	2, 4, 8, 16, 32, 64, 128, 256, 512	Sets the clock divider for the internal ADC.
Advanced Options		
Sample stream server interface	One of the detected Ethernet interfaces	Selects which Ethernet interface on the host machine running the GUI will be used to receive the sample data.
Consumption Type	Current[2], Cumulative	When Current is selected, only P × N samples (where P is the number of packets and N the number of samples per packet) of either voltage or current (as defined by Measurement Type) are displayed without logging all data over time. When Cumulative is selected, all samples over time are logged.
Measurement Type	Voltage, Current	If Current is chosen as the consumption type, this option specifies which channel's samples will be presented.
Max Num of Packets	[1, P]	Defines the number of stream data packets to collect before plotting.
Samples No	[1, N]	Specifies the number of samples per channel contained in a single stream data packet.
Enable EP	Checked, Unchecked	Enables or disables the Energy Point feature.
Save to file	Checked, Unchecked	Determines whether the data should be saved to a file.
Consumption profile name		Defines the name of the consumption profile. This name will be used to create a corresponding folder in the workspace path, where voltage, current, consumption, and other acquisition-related data will be stored.
Load control		Provides options for controlling the load.
Calibration		Opens calibration settings.
Console		Provides access to the console for advanced commands and logs.
Real-Time Statistics		Displays live statistical information during acquisition.

[2] Current consumption type is intended solely for development purposes and is not meant to be used during regular operation.

3.4.3.1. Network interfaces

On the host machine running the GUI application, multiple Ethernet-based communication interfaces may exist (for example, LAN and Wi-Fi). Therefore, one of the first steps required is to select the interface that will be used to receive sampled data from the EPP device. Figure 3.12 illustrates the interface selection process.

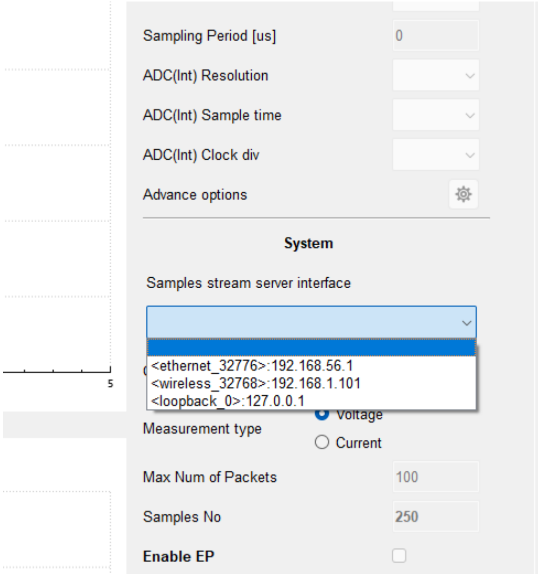


Figure 3.12 - Interface selection

3.4.3.2. Sampling period, Number of Packets and Samples No

The EPP is capable of streaming data at 2 MSps, which corresponds to approximately 32 Mbps (each channel acquires data at 1 MSps, with each sample being 16 bits). The data are streamed in raw format, meaning that the 32 Mbps bandwidth is maintained over Ethernet and received at the same rate on the host side. Data from both channels are organized in data stream packets which maximal size is 500 samples (250 samples per channel).

Receiving and processing this data is not a heavy load for modern CPUs. However, plotting and storing such large amounts of data can pose a significant burden on most CPUs. The CPU load caused by plotting depends primarily on the refresh rate and the amount of data visualized in each update cycle. Meanwhile, RAM usage over time is determined by both the sampling rate and the duration of the acquisition session.

To make the application modular and adaptable to different hardware configurations, three configurable parameters are provided:

- Sampling period (**Ts**)
  Minimal value is 1uS
- Number of stream data packets collected before plotting (**P**)
  Minimal value is 1
- Number of samples per stream data packet (**N**)
  Maximal value is 250

If **Ts** is small, both **P** and **N** should be set to higher values. For example, **N** can be set to a maximum of 250, while **P** should be adjusted to minimize CPU load (this can be done by monitoring CPU usage in Task Manager or similar tools, running a short acquisition, and fine-tuning the parameter). On the other hand, if **Ts** is large, **N** should be set to its minimum value of 1, and **P** should be adjusted to achieve an acceptable real-time plot update interval.

We have already performed tuning on our laptop, which is equipped with an 11th generation Intel Core i-series processor (4 cores, 2.8 GHz), 32 GB of DDR memory, and a 100 Mbps Ethernet connection. The results are summarized in **Table 3.2**.

Table 3.2 - CPU, RAM and Network utilization overview for typical **T_s**, **N** and **P** values

Sampling Period [uS]	P	N	CPU [%]	RAM [MB]	Network [Mbps]
1000000	1	1	< 1.5	50MB	< 0.1 Mbps
1000000	10	1	< 1.5	50 MB	< 0.1 Mbps
1000	1	250	< 1.5	50MB	< 0.1 Mbps
1000	10	250	< 1.5	50MB	< 0.1 Mbps
1000	25	250	< 1.5	50MB	< 0.1 Mbps
1000	25	250	< 1.5	50MB	< 0.1 Mbps
1000	25	100	< 1.5	50MB	< 0.1 Mbps
1000	25	10	< 1.5	50MB	< 0.1 Mbps
10	100	250	8	523MB	3.3 Mbps
10	200	250	6	523MB	3.3 Mbps
10	500	250	5	523MB	3.3 Mbps
1	100	250	> 33%	> 1.5GB	33.3 Mbps
1	200	250	> 33%	> 1.5GB	33.3 Mbps
1	500	250	> 33%	> 1.5GB	33.3 Mbps

Based on results presented in Table 3.3, it is evident that resource utilization on our machine is strongly influenced by the selected sampling rate. Additionally, our machine was unable to sustain long acquisition sessions when a small sampling period was chosen, as resource usage, especially CPU load, became extremely high.

3.4.3.3. Consumption profile name

During acquisition, beside real-time plotting of data there is an option to save data into corresponding files. Data are saved on a path defined with *<workspace_path>/<consumption_profile_name>*. *<workspace_path>* is defined within a first window started after GUI application is run (see *Define Workspace*), while *<consumption profile_name>* is possible to be defined inside Control section of Device Window as it is presented on Figure 3.13.

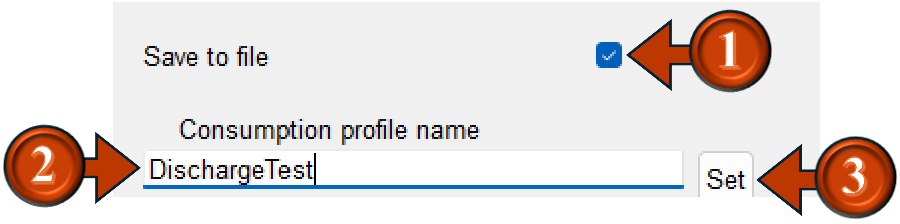
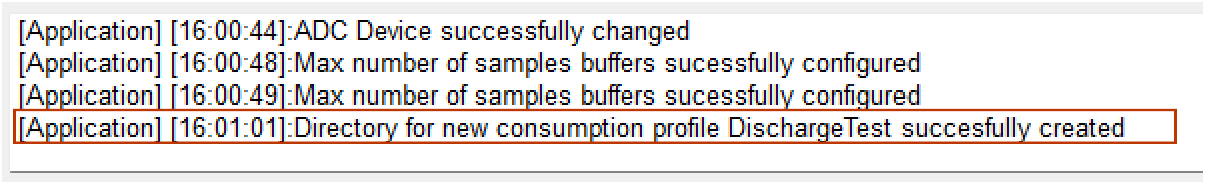


Figure 3.13 - Configure consumption profile name.

Therefore, as the first step, the **"Save to file"** option must be checked (**1** in Figure 3.13), which will enable the **"Consumption profile name"** input field. In this field, the user should enter the desired consumption profile name (**2** in Figure 3.13) and confirm it by clicking the **Set** button (**3** in Figure 3.13). If the Consumption profile name is configured, inside Log will be printed message as it is presented on Figure 3.14.



[Application] [16:00:44]:ADC Device successfully changed
[Application] [16:00:48]:Max number of samples buffers sucessfully configured
[Application] [16:00:49]:Max number of samples buffers sucessfully configured
[Application] [16:01:01]:Directory for new consumption profile DischargeTest succesfully created

Figure 3.14 - Log after consumption profile name is set

When acquisition end and **"Save to file"** option was enabled, on path defined with *<workspace_path>/<consumption profile_name>* will be located 4 files which description is presented in Table 3.4.

Table 3.4 - Files created for consumption profile

Name of the file	Content
vc.csv	Voltage and Current samples
ep.csv	Energy Point samples
cons.csv	Consumption samples
OpenEPT.txt	Acquisition related information

3.4.3.4. Acquisition control and network status

Acquisition control buttons are located at the bottom of the Control section of the Device window. Below acquisition control is located “Connection status” label which is green when device is connected and red when device is disconnected. These buttons with “Connection status” label are presented on **Figure 3.15**.

Figure 3.15 - Acquisition control and Connection status

Acquisition control buttons are:

- Start acquisition
- Pause acquisition
- Stop acquisition
- Acquire parameters

The Start Acquisition button is used to send the "Start Acquisition" command to the EPP device, after which it begins transmitting voltage and current samples over Ethernet. The Pause Acquisition and Stop Acquisition buttons both halt the current acquisition; however, unlike Stop, the Pause function does not clear the plot buffers, allowing the user to resume without losing existing data.

3.4.3.5. Console

All actions that depend on the EPP board are executed by sending ASCII-based commands to the board. These commands are encapsulated in Command messages, which are transmitted over the TCP stack to the EPP device. In the GUI application, all commands are automatically generated and sent to the EPP board in response to specific user actions — for example, when the Start Acquisition or Stop Acquisition buttons are pressed, or when querying the current protection status.

However, in the Console Window (shown on Figure 3.16), which can be opened by clicking the Console button in the Control section of the Device window, it is possible to send raw control messages directly to the device and view their responses.

The Console Window is divided into two main sections: the command line (**1**) and the command log (**2**).

- The command line allows users to manually enter specific commands. Commands can be sent by clicking the Send button or pressing the Enter key. This input field supports autocomplete, and previous commands can be browsed using the up and down arrow keys.
- The command log displays all commands sent from the GUI as well as the responses received from the EPP device.

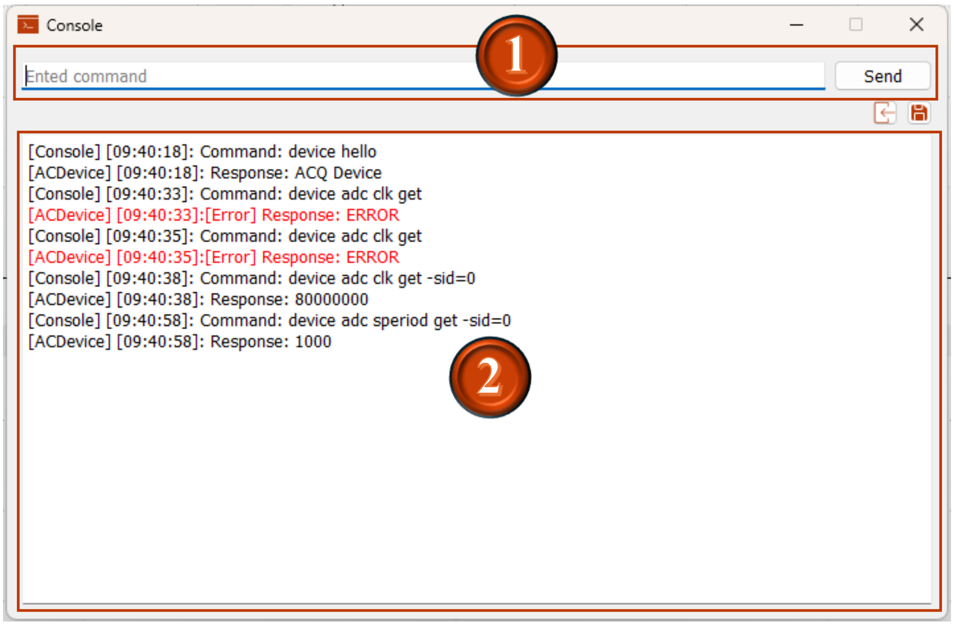


Figure 3.16 - Console Window

A list of currently supported commands can be found in this document.

One of the main strengths of the OpenEPT tools is their flexibility and ability to adapt to specific project requirements. Thanks to the Console Window and the Command Parser service on the EPP side (see EPP Developer Guide), users can introduce custom commands and easily test them directly through the console before integrating them into GUI workflows. This approach allows for rapid prototyping and fine-tuning of new features tailored to each application.

3.4.3.6. Load control

Inside Control Section, after clicking on Load Control button, it is opened Load Control Window which layout is presented on Figure 3.17.

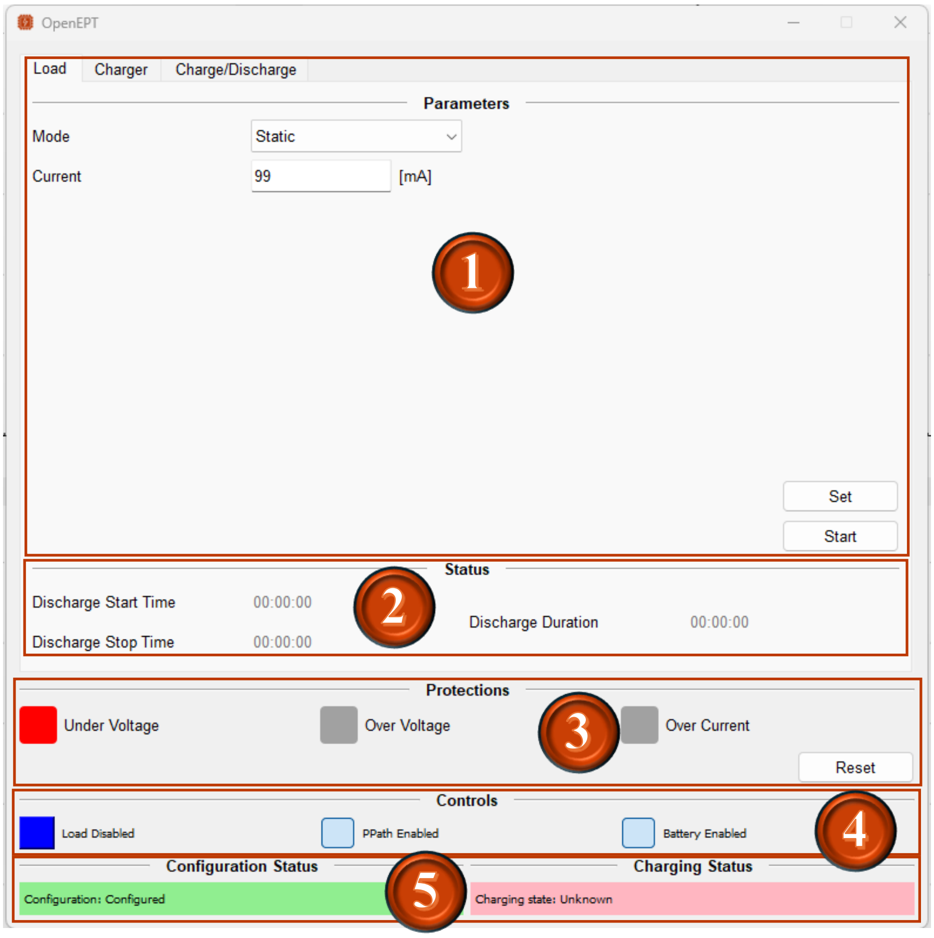


Figure 3.17 - Load Control Window

This window consists of following sections:

1) Load Control
2) Load Status
3) Protection status
4) Power Paths Controls
5) Acquisition status

In the Load Control section (**1**), it is possible to configure the load current. This configuration includes two fields: Mode and Current. Currently, the only supported mode is Static, which represents a constant (fixed) current. The Current field allows you to specify the desired current amplitude, expressed in milliamperes (mA). After selecting the mode and entering the desired amplitude, click the **Set** button to apply the configuration. If the current is successfully set, a confirmation message will appear in the Log window. Once the current is configured, the load can be activated by pressing the Start button.

Before starting the load, make sure that all protections are disabled (this can be verified in the Protections section) and that the load is enabled (visible in the Controls section). The indicator status before starting the load is shown in Figure 3.18

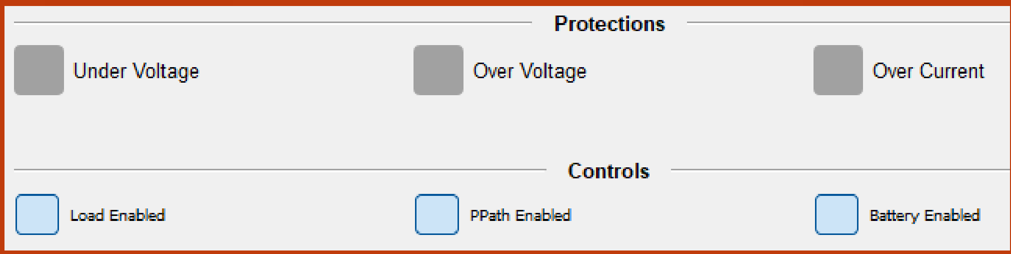


Figure 3.18 - Protections and Power Path Controls before starting load

The Status section (**2**) provides information about the duration of the load mode, including the absolute time when the load started, the total duration of the load mode, and the absolute time when the load was stopped. There are two conditions that can cause the load to return to 0: when the user presses the Stop button or when a protection is triggered.

The Protections section (**3**) displays the status of all hardware protections available on the EPP board. These protections include Under Voltage, Over Voltage, and Over Current. If a protection is active, its corresponding square indicator will appear red; if inactive, it will appear grey. When any protection is triggered on the EPP board, it remains latched to prevent potential hardware damage. Before resetting a protection, make sure to investigate and resolve the root cause. Once the cause has been addressed, you can restore the protection to its initial state by pressing the **Reset** button.

In the Power Path Controls section (**4**), it is possible to enable or disable the corresponding circuit paths on the EPP board. The available controllable paths are shown in Figure 3.19.

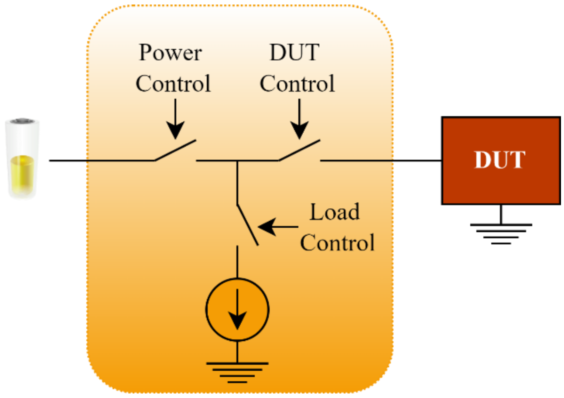


Figure 3.19 - EPP Circuit's paths that are controllable from software

The **PPath** control in the Power Path Controls section (**4**) manages the Power Path, the **Load** control manages the Load, and the **Battery** control corresponds to the DUT (Device Under Test).

The final section in the Load Control window (**5**) contains two status fields: Configuration and Charging.

- The Configuration field indicates whether the device is properly configured or not.
- The Charging status shows one of the possible battery states: Charging (when a negative current value is detected), Discharging (when a positive current value is present), or Idle (when the current is approximately zero).

3.4.3.7. Calibration

Each time a new session is started, it is necessary to perform a calibration of the EPP board. This is done through the Calibration Data window, which can be opened by pressing the Calibration button located in the Control section of the Device Window. The layout of the Calibration Data window is shown in Figure 3.20.

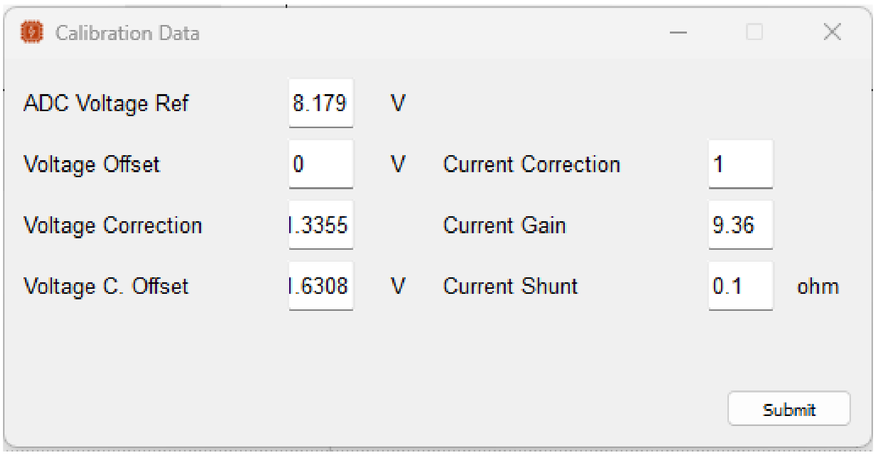


Figure 3.20 - Calibration Data window

Each field inside this window affects a specific channel, either voltage or current. To understand how these parameters influence the raw measurements, it is important to first understand the formula used to convert the raw voltage and current readings into their final, calibrated values.

The formula for calculating current reflects several layers of calibration logic. The raw ADC value is scaled by a gain and shunt resistance, then adjusted with the current correction factor to account for known deviations or hardware inaccuracies. This ensures that small mismatches in hardware (e.g., non-ideal resistor values or amplifier gains) are corrected in software, yielding high-quality, consistent results regardless of board-to-board variation. Equations (E.1) and (E.2) are used to calculate final voltage value.

$$currentInc = \left(\frac{V_{ref}}{2^{Res}}\right) \tag{E.1}$$

$$I = \left(\frac{rawCurrent \cdot currentInc - V_{offset\_current}}{R_{shunt} \cdot G_{amp}}\right) \cdot 1000 \cdot K_{current} \tag{E.2}$$

Similarly, voltage data is compensated using a combination of offset and linear scaling based on the supplied correction values. The calibrated voltage is then checked against minimum and maximum values to support statistical analysis, which will later be reported or visualized. The processed values are stored in circular buffers and used to derive instantaneous as well as cumulative statistics like average voltage, peak current, and total energy consumption. Equations (E.3) and (E.4) are used to calculate final voltage value.

$$voltageInc = \left(\frac{V_{ref}}{2^{Res}}\right) \cdot K_{voltage} \tag{E.3}$$

$$V = V_{offset} + rawVoltage \cdot voltageInc \tag{E.4}$$

The calibration has a significant impact on downstream processing. For instance, the accuracy of cumulative consumption measurements (expressed in mAh) depends on how precisely the current readings are corrected and integrated over time. Inaccurate gain or shunt values would distort the integration result, leading to underestimated or overestimated energy consumption. This is particularly critical in battery-driven applications, where fine-grained consumption data is vital for energy management.

4. Typical Use Cases

This section is dedicated to illustrating practical usage scenarios and typical workflows for the OpenEPT system, providing users with clear, step-by-step guidance on how to apply the tool in real-world contexts.

While previous sections focused on describing individual components, setup procedures, and configuration options from a functional perspective, this section integrates those elements into cohesive, actionable examples. The purpose is to help users understand how to efficiently start and configure data acquisition, define load profiles, adjust measurement parameters, and perform advanced analysis tasks.

By following these scenarios, users can gain confidence in operating both the hardware and the software, ensuring reliable and reproducible results. Detailed examples also highlight best practices for optimizing system performance, managing resource utilization, and avoiding common pitfalls. Furthermore, these workflows demonstrate how to leverage various features of the GUI, such as plotting controls, workspace management, and calibration tools, in a practical and intuitive manner.

Some of the scenarios that are covered for now are:

- Start data acquisition with no logging to file
- Start data acquisition with file logging enabled
- Analyse stored data
- EPP Calibration
- Energy breakpoint scenario
- Generate load profile

4.1. Plot Voltage and Current values acquired with 1ms

In this use case, we demonstrate how to configure the EPP board using the OpenEPT GUI application to acquire voltage and current samples at a 1 ms sampling interval and plot these values in real time. In this demonstration, the samples will only be displayed live and will not be stored on the host machine.

For each use case, it is assumed that the EPP device is flashed with the latest firmware as described in the corresponding section and is connected to the LAN network with the IP address 192.168.1.110.

Step 1: Start application and leave WS path empty

When the application starts, the Workspace Path Selection window appears (for more details, see the corresponding section). Since data will not be saved to a file in this scenario, the workspace path can be left empty. In the Workspace Path Selection window, leave the path line editor blank as shown in Figure 4.1, and click **OK**.

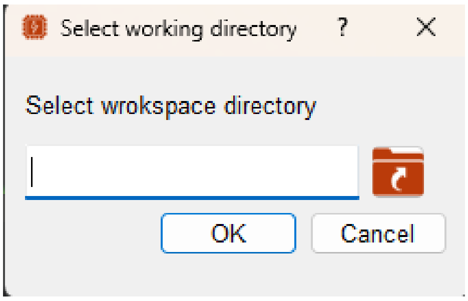


Figure 4.1 - Workspace Selection window with empty Workspace path

When OK button is pressed, OpenEPT GUI application main layout appears as it is presented Figure 4.2.

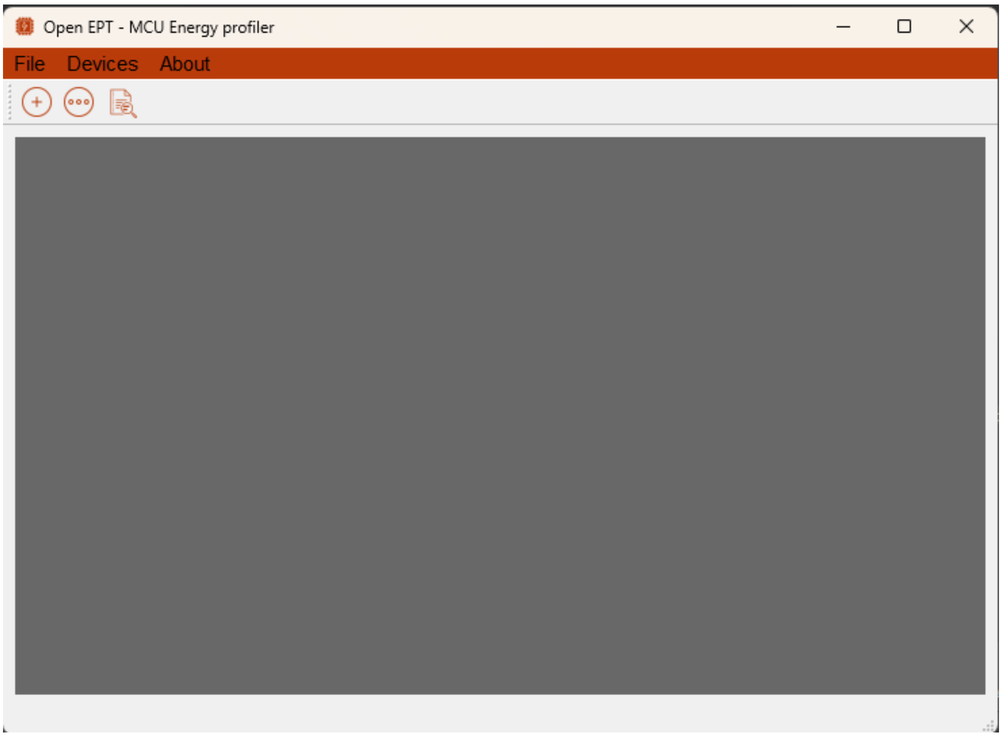


Figure 4.2 - OpenEPT main window

Step 2: Add EPP device

In the main window of the OpenEPT GUI, click the Add Device button ⊕ to open the Add New Device window (**1** in **Figure 4.3**). Enter the device IP address (by default: 192.168.1.100) (**2**). Once the IP address has been entered, click **Add Device** to continue (**3**).

Make sure that EPP and HOST machine to which EPP device is connected are within the same subnet mask.

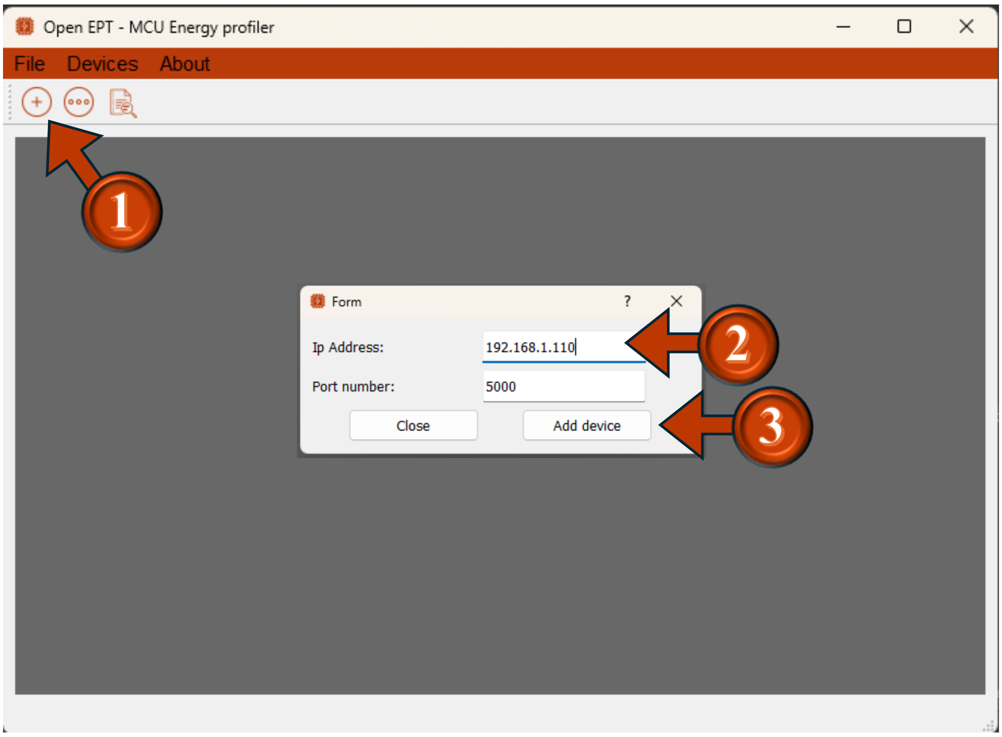


Figure 4.3 - Add New Device

Step 3: Select Interface

If the device is successfully added, the Device Window will appear (see the relevant section for more details about the Device Window and its configuration parameters), as shown in Figure 4.4. Initially, all options will be greyed out because the data reception interface has not yet been selected.

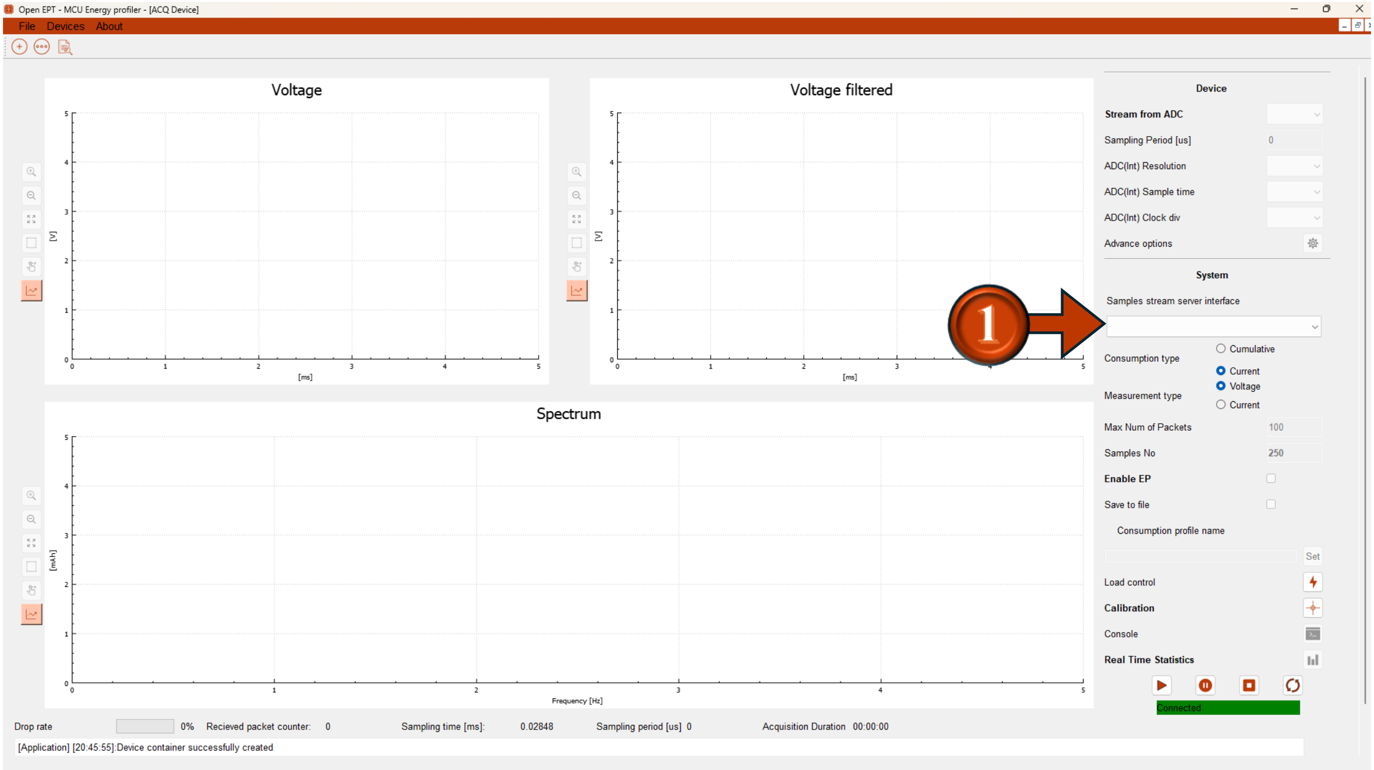


Figure 4.4 - Device window without selected interface and with default configuration

In the Device Window, select the interface over which the sampled data will be received. In this example, we choose the WiFi adapter with IP address 192.168.1.101. The appearance of the Device Window after selecting the interface is shown in Figure 4.5.

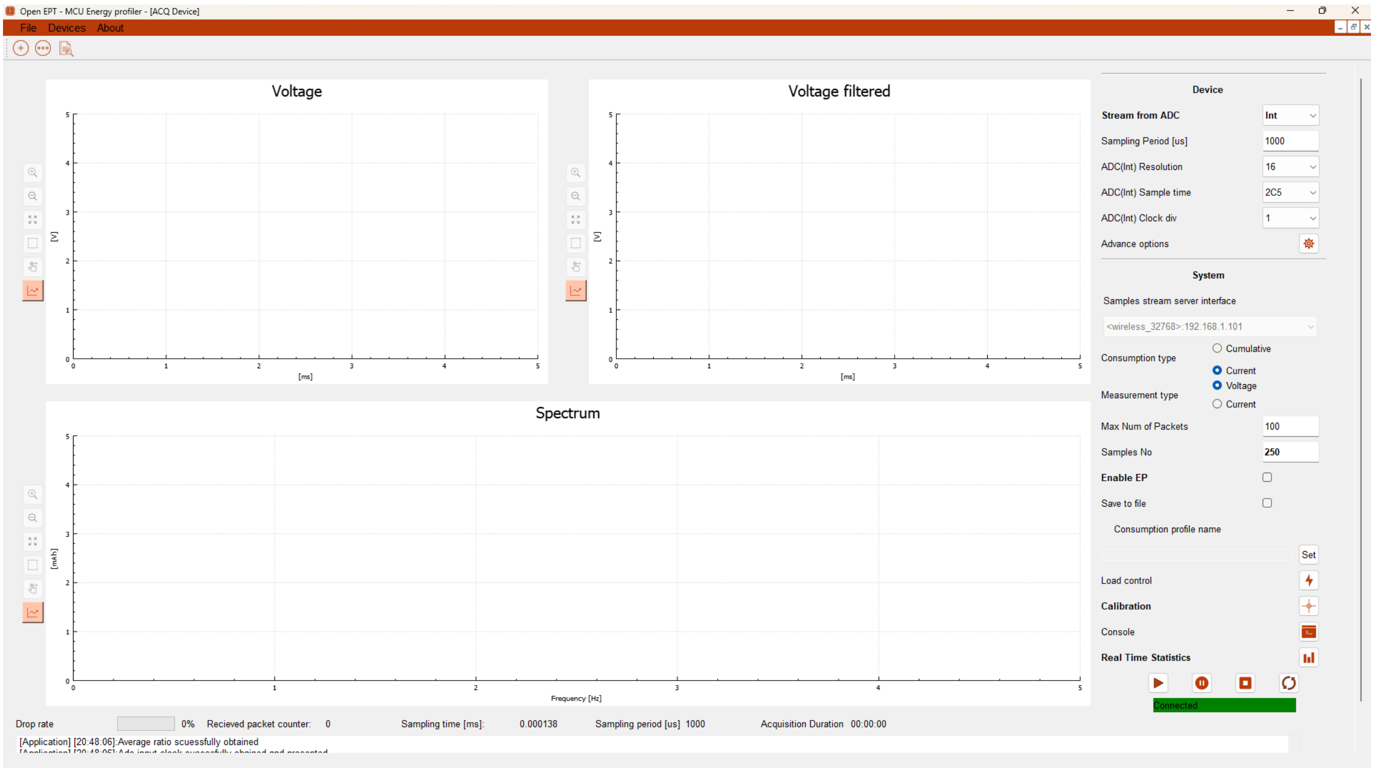


Figure 4.5 - Device window after interface is selected.

After Interface is selected, blue diode changes its state to Green as it is presented on Figure 4.6.

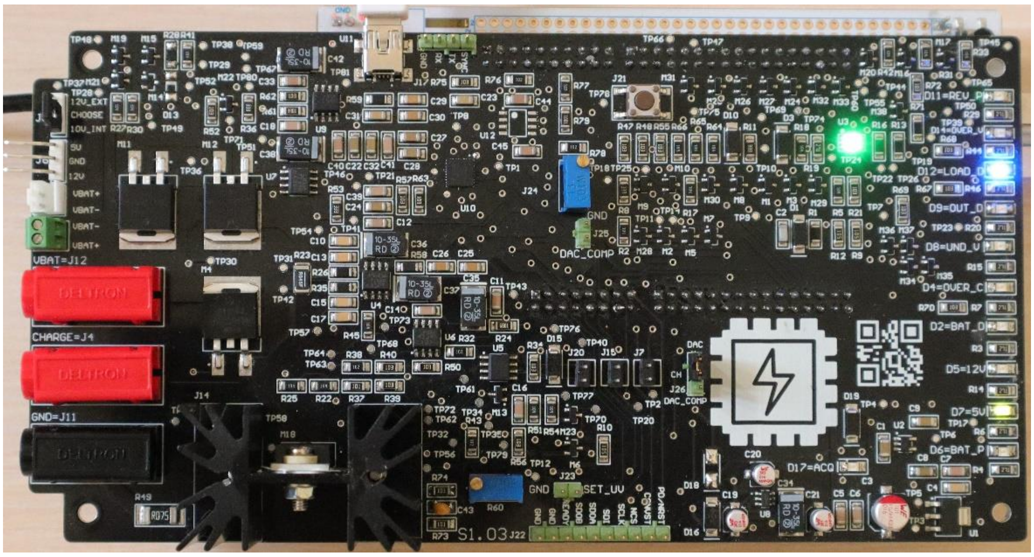


Figure 4.6 - OpenEPT EPP Board after interface is selected

Step 4: Configure EPP and Acquisition parameters

Table 4.1 list all parameters that should be changed.

Table 4.1 - Parameters configured for Use Case 1

Parameter Name	Parameter Value	Description
Stream from ADC	Ext	We use external ADC
Consumption type	Cumulative	We want to monitor samples over time
Max Number of samples	5	See

The Configuration section of the Device Window, after applying the parameters from Table 4.1, is shown in Figure 4.7.

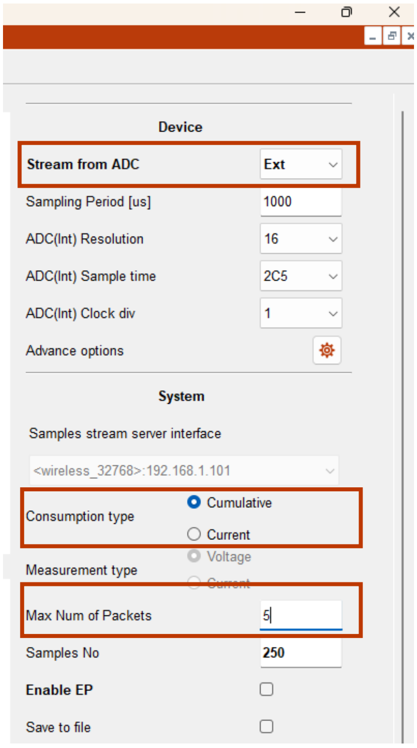


Figure 4.7 - Configuration after parameters value from Table 4.1 are applied

Step 5: Control Acquisition

To start the acquisition, press the Start Acquisition button ▶. If the parameters are correctly set, data should appear on the corresponding plots after a few seconds, as shown in Figure 4.8 .

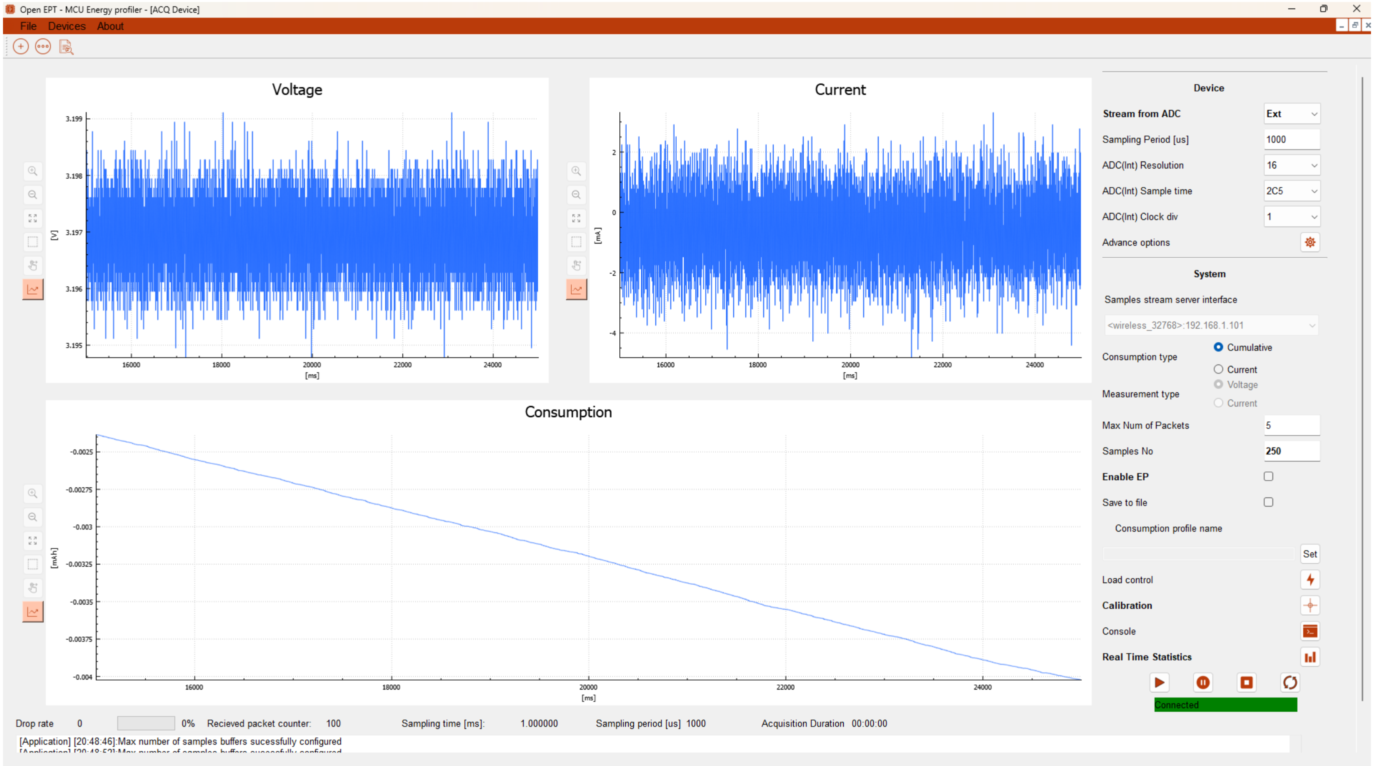


Figure 4.8 - Device window after acquisition is started

After the acquisition starts, the blue LED located at the bottom right corner of the board will turn on (**1** in Figure 4.9).

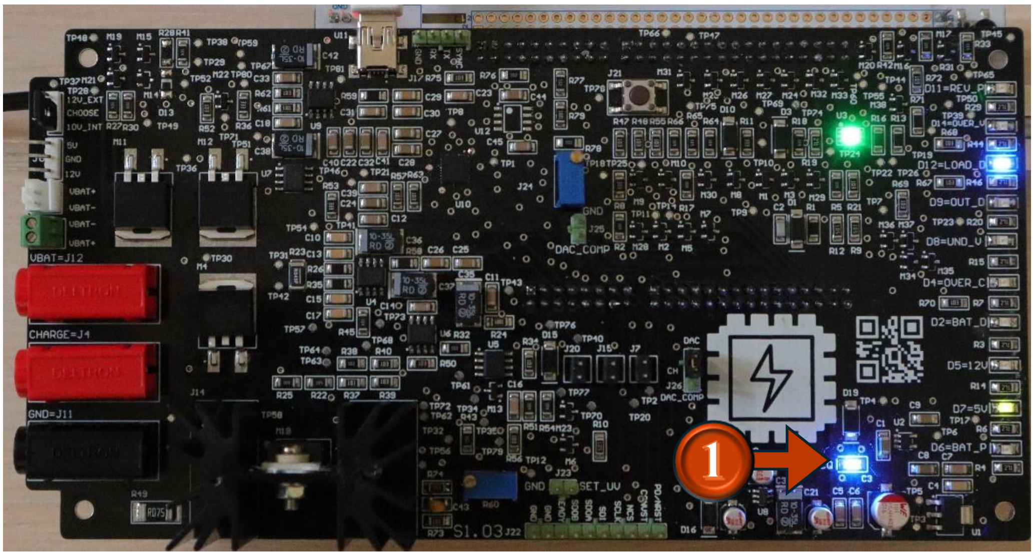


Figure 4.9 - OpenEPT EPP Board after the acquisition starts

At any time, you can stop or pause the acquisition by pressing the ■ or ⏸ buttons, respectively.

4.2. Plot Voltage and Current values acquired with 1ms and save to file

In this use case, we demonstrate how to configure the EPP board using the OpenEPT GUI application to acquire voltage and current samples at a 1 ms sampling interval, plot these values in real time, and additionally save the samples to corresponding files (see *3.4.3.3 Consumption profile name*). The configuration and steps presented in this demonstration are largely similar to those described in *4.1 Plot Voltage and Current values acquired with 1ms*. Therefore, throughout this demonstration, we will frequently refer to the steps already outlined in that section.

Step 1: Start application and set WS path

When the application starts, the Workspace Path Selection window appears (for more details, see *3.2 Define Workspace*). Since data will be saved to a file in this scenario, the workspace path needs to be configured. In the Workspace Path Selection window, click the **Select Directory** button (**1** in Figure 4.1) and navigate to your desired workspace directory. Once the directory is selected, its path will be displayed in the Workspace Path Selection window (**2** in Figure 4.1). Click **OK** to proceed to the next step.

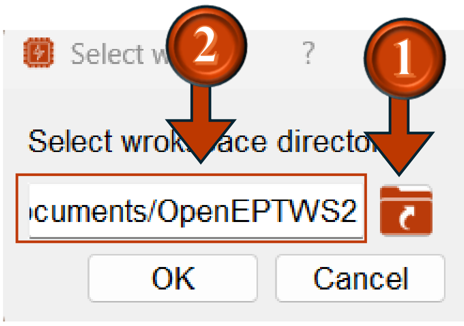


Figure 4.10 - Select workspace directory

Step 2: Add EPP device

Same as **Step 2** in *4.1 Plot Voltage and Current values acquired with 1ms*

Step 3: Select Interface

Same as **Step 3** in *4.1 Plot Voltage and Current values acquired with 1ms*

Step 4: Configure EPP and Acquisition parameters

Configure all parameters that are listed in **Step 4** in *4.1 Plot Voltage and Current values acquired with 1ms*. Additionally, Within Configuration section of Device Window “Save to file option” should be selected (**1** in Figure 4.11). Then, enter the Consumption Profile name (**2** in Figure 4.12) and finally click the **Set** button (**3** in Figure 4.13).

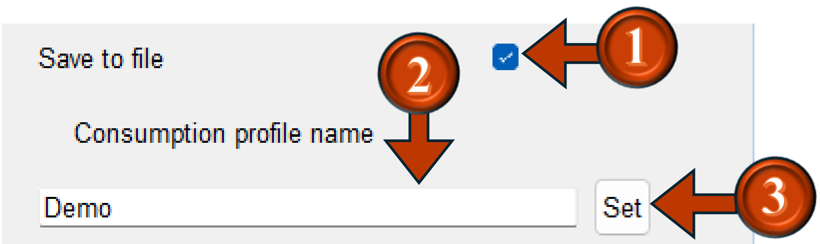


Figure 4.14 - Save to file part of Configuration section

If profile name is successfully configured, corresponding Log message will be printed as it is presented on Figure 4.15.

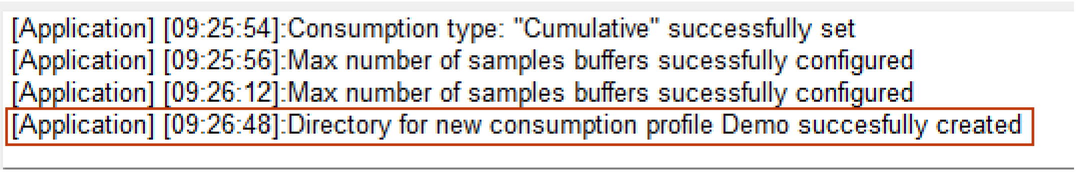


Figure 4.15 - Log message after consumption profile name is configured

Step 5: Control Acquisition

Same as **Step 5** in *4.1 Plot Voltage and Current values acquired with 1ms*

Step 6: Check for saved file on Workspace path

After Acquisition is done, on *<workspace_path>/<consumption_name>* path should be created 4 files as it is presented on Figure 4.16. Name and content of the files are listed in Table 3.4.

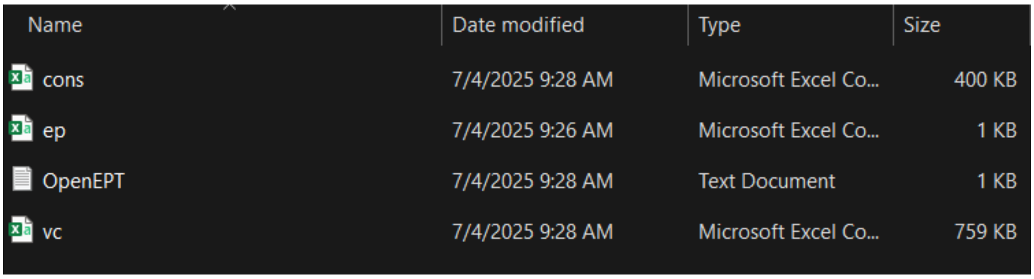


Figure 4.16 - Content of *<workspace_path>/<consumption_name>* directory after the end of the acquistion

LIST OF FIGURES

References

[b1] ADS92x4R Dual, Low-Latency, Simultaneous-Sampling SAR ADC Datasheet, June 2019

[1] Command list document