

Quectel Power Mini Portable DC Power Analyzer User Manual

Version: 1.0

Date: 2026-06-03

Status: Released



At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.

No. 8 Waipojing Road, Sijing Town, Songjiang District, Shanghai 201601, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local offices. For more information, please visit:

<https://www.quectel.com/contact/>.

For technical support, or to report documentation errors, please visit:

<https://www.quectel.com/tech-support/>.

Or email us at: support@quectel.com.

Legal Notices

We provide this document to support your product design. You are required to design your products based on the specifications and parameters set forth herein. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. You acknowledge and agree that we may add to, amend, or restate this document at any time at our sole discretion without any prior notice to you, and such additions, amendments, or restatements shall be binding upon you.

Use and Disclosure Restrictions

License Agreements

The recipient of any hardware, software, materials, or documentation provided by us shall keep such content confidential, unless expressly authorized by us. The recipient shall not disclose, access, or use any part of the received content for any purpose other than the execution and implementation of the intended project.

Copyright

Our and third-party products hereunder may contain copyrighted materials, including but not limited to protected content, hardware, software, and documentation owned by us or applicable third parties. Unless prior written consent is obtained, you shall not access, use, or disclose any documents or information provided by us, nor shall you copy, reproduce, republish, display, translate, distribute, merge, modify, or create derivative works from any such copyrighted materials. We and the applicable third party retain exclusive rights to all copyrighted materials. No license to any patents, copyrights, trademarks, or service marks shall be granted or transferred. For the avoidance of doubt, no form of purchase shall be construed as granting any license beyond a normal, non-exclusive, royalty-free license to use the product. We reserve the right to pursue legal action against any violation of confidentiality obligations, unauthorized

use, or any other unlawful or malicious use of the aforementioned documents and information.

Trademarks

Unless otherwise expressly provided, nothing in this document shall be construed as conferring any rights to use any trademark, trade name, name, abbreviation, or counterfeit thereof owned by us or any third party in advertising, publicity, or any other contexts.

Third-Party Rights

You understand that this document may refer to hardware, software, and/or documentation owned by one or more third parties (“third-party materials”). Use of such third-party materials is subject to all applicable restrictions and obligations set forth herein.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing, course of performance, or usage of trade.

Privacy Policy

To enable product functionality, certain device data may be uploaded to our or third-party servers, including those operated by carriers, chipset suppliers, or servers designated by you. We strictly comply with applicable laws and regulations and will retain, use, disclose, or otherwise process relevant data solely for the purpose of enabling product functionality, or as permitted by applicable laws. Before interacting with any third party regarding data exchange, please be informed of and understand their privacy and data security policies.

Disclaimer

- a) We shall not be liable for any damages resulting from failure to comply with applicable operational or design specifications.
- b) We shall bear no liability for any inaccuracies or omissions in this document, nor for any damages arising from the use of the information contained herein.
- c) While we make every effort to ensure the integrity, accuracy, and timeliness of the features and functions under development, errors or omissions may nevertheless occur. Unless otherwise provided in a valid written agreement, we make no warranties of any kind, express, implied, or statutory, and disclaim all liability for any loss or damage arising from the use of any features or functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage is foreseeable.
- d) We assume no legal responsibility for the accessibility, safety, accuracy, availability, legality, or completeness of any information, content, advertising, commercial offers, products, services, or materials on third-party websites or third-party resources.

Copyright © Quectel Wireless Solutions Co., Ltd. 2026. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2026-06-03	Dustin Wei	Creation of the document
1.0	2026-06-03	Dustin Wei	First official release

Safety Information

When operating and installing this equipment, you must follow the safety information in the product labels and this user manual.

The "DANGER", "WARNING", and "CAUTION" items mentioned in this document are only supplementary to the safety matters and do not represent all safety matters.

General

Installation

Personnel who install, operate, and maintain this product must understand all safety precautions and master the correct methods before installation, operation, and maintenance. Please follow these basic installation requirements:

- The product must be serviced by Quectel-authorized personnel.
- The replacement of parts must be performed by Quectel-authorized personnel.
- Strictly follow the factory assembly method after installation, otherwise it may affect the product performance.

Grounding

- Before installing the product, complete the grounding first; when dismantling the product, remove the ground wire last.
- It is strictly prohibited to perform any operation on the product when it is not or poorly grounded.
- Please ensure the product is well grounded before any operations.

Personal Safety

- It is strictly prohibited to install, operate, and maintain the product in dangerous environments, such as thunderstorms and flammable environments.
- Keep the product away from children.

Product Safety

- Keep the product away from fire and heat sources. Do not throw it into fire to avoid fire hazards or damage to the product.
- Avoid subjecting the product to severe vibration, impact, or compression to prevent damage to internal components or product failure.
- Pay attention to waterproof measures and ensure that the product interfaces are protected from water spray or immersion to prevent damage or malfunction.

- Do not attempt to disassemble or repair the product yourself, as this may cause safety risks or damage to the internal structure.
- Ensure the product is securely installed in a stable and reliable manner, whether on table, wall, or rack.
- Verify that the product is properly grounded before powering it on, and disconnect the ground wire only after shutting down.
- Operators must wear anti-static gloves during disassembly and assembly, and use proper tools to remove or install covers.
- Do not block or cover the product's ventilation openings.
- Removing product sealing plugs, such as antenna plugs, is prohibited.

Safety in Mobile Scenarios

- Full attention must be paid to driving at all times in order to reduce the risk of an accident. Using a mobile while driving (even with a handsfree kit) causes distraction and can lead to an accident. Please comply with laws and regulations restricting the use of wireless devices while driving.
- Switch off the terminal or mobile before boarding an aircraft. The operation of wireless appliances in an aircraft is forbidden to prevent interference with communication systems. If there is an Airplane Mode, it should be enabled prior to boarding an aircraft. Please consult the airline staff for more restrictions on the use of wireless devices on an aircraft.
- Wireless devices may cause interference on sensitive medical equipment, so please be aware of the restrictions on the use of wireless devices when in hospitals, clinics or other health-care facilities.
- Terminals or mobiles operating over radio signal and cellular network cannot be guaranteed to connect in certain conditions, such as when the mobile bill is unpaid or the (U)SIM card is invalid. When emergency help is needed in such conditions, use emergency call if the device supports it. In order to make or receive a call, the terminal or mobile must be switched on in a service area with adequate cellular signal strength. In an emergency, the device with emergency call function cannot be used as the only contact method considering network connection cannot be guaranteed under all circumstances.
- The terminal or mobile contains a transceiver. When it is ON, it receives and transmits radio frequency signals. RF interference can occur if it is used close to TV sets, radios, computers or other electric equipment.
- In locations with explosive or potentially explosive atmospheres, obey all posted signs and turn off wireless devices such as mobile phone or other terminals. Areas with explosive or potentially explosive atmospheres include fueling areas, below decks on boats, fuel or chemical transfer or storage facilities, and areas where the air contains chemicals or particles such as grain, dust or metal powders.

Electrical Safety

Adverse Weather

- Do not operate the product with AC power during thunderstorms, as it may pose a fatal risk.
- Do not mount the product on outdoor boxes or poles during thunderstorms, as this may lead to fatal

danger.

- Never install, operate, or maintain the product when powered by high-voltage supply, as the improper handling may result in fatal danger.

Power Supply

- Before connecting the power cable, verify the polarity of the cable (positive and negative).
- Always turn off the product before disconnecting the power cable.
- Disconnecting the power cable while the product is powered on is strictly prohibited, as it could cause electric sparks and potential injury.
- After connecting the power cable, ensure the connection is secure.

ESD Protection

- To prevent electrostatic discharge (ESD) damage from friction or movement, the operator must take appropriate ESD precautions before installing, operating, or maintaining the product.
- Before touching or handling the product, the operator must wear an anti-static wristband and connect it to a grounded surface.

Contents

About the Document	3
Safety Information	4
General.....	4
Electrical Safety	5
Contents	7
Table Index	9
Figure Index	10
1 Product Overview	11
2 Appearance & Interfaces	14
2.1 Appearance.....	14
2.2 Interfaces.....	15
3 Wiring Introduction	17
3.1 Wiring Diagram	17
3.2 Power Supply	17
4 Esight Overview	18
4.1 Windows Driver and .NET Framework Installation Guide	18
4.1.1 Installation for Windows 10	18
4.1.2 .NET Framework Dependency Installation (Optional)	18
4.2 Esight Tool Operations	18
4.2.1 Device-PC Connection.....	18
4.2.2 Waveform Capture and Analysis.....	19
4.2.3 Other Custom Configuration	23
4.3 System Upgrade Guide.....	25
5 Application Development Guide	27
5.1 API Overview	27
5.2 API Description	28
5.2.1 Init.....	28
5.2.2 OpenPort.....	28
5.2.3 ClosePort.....	29
5.2.4 SetType	29
5.2.4.1 MPOWER_TYPE	30
5.2.5 SetMode	30
5.2.5.1 MPOWER_POWER_MODE	31
5.2.5.2 MPOWER_AMPERE_MODE	31
5.2.6 SetVoltage	32
5.2.6.1 MPOWER_VOLTAGE_E.....	32
5.2.7 SetPower	35

5.2.7.1	MPOWER_POWER_SUPPLY	36
5.2.8	StartSample.....	36
5.2.9	StopSample.....	36
5.2.10	GetCurrent.....	37
5.2.11	GetFwVersion.....	37
5.2.12	GetDllVersion	38
5.3	Python Demo Code.....	38
6	Appendix References	42

Table Index

Table 1: Specifications	12
Table 2: Product Interfaces	16
Table 3: API Overview	27
Table 4: Terms and Abbreviations	42

Figure Index

Figure 1: Front and Rear Views	14
Figure 2: Left View	14
Figure 3: Right View	15
Figure 4: Product Interfaces	16
Figure 5: Wiring Diagram	17

1 Product Overview

Quectel Power Mini is a high-precision DC power analyzer specifically designed for IoT module communication and embedded software development scenarios. Integrating portable DC programmable power supply, high-precision current measurement, real-time dynamic current waveform analysis, and cost-effectiveness, Quectel Power Mini helps developers efficiently conduct precise power consumption testing and dynamic measurement analysis during product R&D phases.

Esight, the accompanying host application, is fully compatible with Windows 10 and above (x86/x64 architecture supported), providing an intuitive interface for data interaction and visual analysis.

Industry Pain Points & Challenges

- **Insufficient testing accuracy:** As low-power requirements for handheld and wearable devices become increasingly stringent, traditional high-power DC power supplies feature low costs yet fail to meet the testing demands of micro-power scenarios due to limited current measurement accuracy.
- **No dynamic analysis capability:** Conventional power supplies cannot realize real-time display, storage, playback and analysis of current waveforms. They are unable to accurately evaluate the dynamic power consumption characteristics of devices, and cannot satisfy refined testing requirements for module communication and similar applications.
- **Constraints on cost and portability:** Professional DC power analyzers are expensive (usually ranging from tens of thousands to hundreds of thousands of RMB) and bulky, making them extremely difficult to use for on-site debugging in outdoor environments without 220 V AC power or for frequent transportation.

Core Solutions & Advantages

- **Ultra-portable design & flexible deployment:** Quectel Power Mini is roughly the size of a regular smartphone for excellent portability. It can be powered by the standard adapter or USB adapter. For field use or environments without AC power, the device can be directly driven by a laptop (for both communication and power supply) or a power bank, fully adapting to on-site field testing.
- **High-precision dynamic current measurement:** Quectel Power Mini supports an ultra-wide dynamic current measurement range from nA to A, with a measurement accuracy up to 50 nA. Combined with the Esight tool, the device accurately captures and reflects dynamic current variations, and supports real-time waveform display, in-depth analysis, data storage and playback, enabling developers to quickly locate abnormal power consumption points.
- **High-efficiency automated integration:** Equipped with comprehensive secondary development APIs, Quectel Power Mini enables synchronous control of multiple power analyzers. The device can be easily integrated into existing automated test systems, greatly improving the efficiency of production line testing and R&D verification.

The detailed specifications of Quectel Power Mini are as follows:

Table 1: Specifications

Items	Description
Product Model	<ul style="list-style-type: none"> ● Quectel Power Mini
Product Name	<ul style="list-style-type: none"> ● Portable DC power analyzer
Physical Characteristics	<ul style="list-style-type: none"> ● Dimensions: 103 mm × 70 mm × 24 mm (L × W × H)
	<ul style="list-style-type: none"> ● Weight: approx. 170 g
	<ul style="list-style-type: none"> ● Housing material: metal
Hardware Interfaces	<ul style="list-style-type: none"> ● USB interfaces: Type-A, USB 2.0
	<ul style="list-style-type: none"> ● DUT output: red and black leads
	<ul style="list-style-type: none"> ● Power supply input: DC 12 V adapter, Type-C, power bank
Electrical Characteristics	<ul style="list-style-type: none"> ● DC input voltage: 12 V (typical); Up to 13 V
	<ul style="list-style-type: none"> ● Type-A input voltage: 5 V
	<ul style="list-style-type: none"> ● Type-C input voltage: 5 V, 9 V, 12 V
	<ul style="list-style-type: none"> ● Output voltage range: 0.6–6.5 V (50 mV step)
	<ul style="list-style-type: none"> ● Output voltage ripple: ≤ 30 mVpp
	<ul style="list-style-type: none"> ● Load regulation ¹: ≤ 5%
	<ul style="list-style-type: none"> ● Maximum output power: 25 W
	<ul style="list-style-type: none"> ● Current range: nA to 5 A
	<ul style="list-style-type: none"> ● Current resolution: nA to μA level
Operating Temperature	<ul style="list-style-type: none"> ● Sampling rate: 5 ksps
	<ul style="list-style-type: none"> ● 0–60 °C
	<ul style="list-style-type: none"> ● Precision mode/Auto mode/Normal mode
Measurement Modes	<ul style="list-style-type: none"> ● Precision mode/Auto mode/Normal mode
PC Tool	<ul style="list-style-type: none"> ● Esight (supports Windows 7 and above versions)
SDK Support	<ul style="list-style-type: none"> ● C#, Python, VB, C++, C, LabVIEW, etc.
Output Protection	<ul style="list-style-type: none"> ● Supported

¹ Test conditions: voltage = 4 V, load dynamic range = 10 mA to 2.0 A.

Firmware Upgrade	● Supported (via USB)
------------------	-----------------------

2 Appearance & Interfaces

This section displays images that represent the appearance and interfaces. These exhibited images serve illustrative purpose only and may not accurately reflect the actual product. To view the authentic appearance and label, please refer to the product you have received from Quectel.

2.1 Appearance



Figure 1: Front and Rear Views

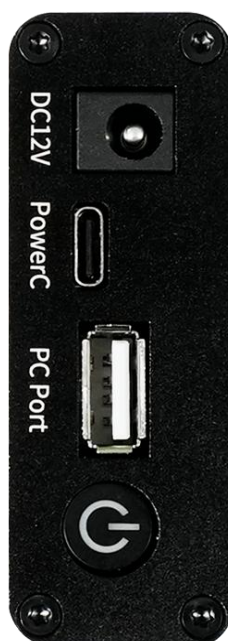


Figure 2: Left View



Figure 3: Right View

2.2 Interfaces

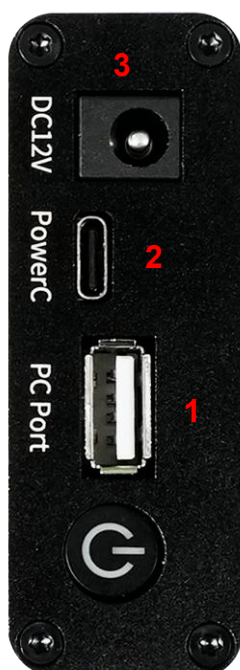




Figure 4: Product Interfaces

Table 2: Product Interfaces

No.	Name	Description
1	Type-A USB	USB 2.0, used for PC communication and power supply
2	Type-C	Power supply input, supporting 5 V, 9 V, 12 V charger or power bank
3	DC power supply interface	DC 12 V adapter input
4	Red lead	DUT output (positive)
5	Black lead	DUT output (negative)

3 Wiring Introduction

3.1 Wiring Diagram

The wiring diagram is shown below:

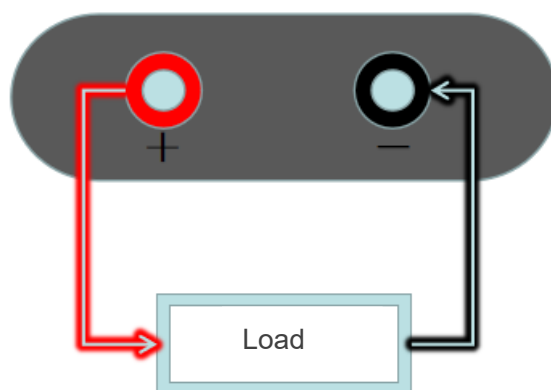


Figure 5: Wiring Diagram

- It is required to supply the load with an adjustable voltage ranging from 0.6 V to 6.5 V.
- The real-time waveform of load current and voltage can be monitored and displayed.

3.2 Power Supply

- In scenarios with mains electricity outlets, it is recommended to power the Quectel Power Mini via a 12 V adapter or USB (supporting Type-C fast charging).
- In scenarios without mains electricity outlets, the Quectel Power Mini can be directly powered by a laptop or a power bank.

NOTE

The Type-A port of the Quectel Power Mini supports both power supply and PC communication. When the device is powered via a Type-C cable (one end connected to a PC and the other to the device) or an adapter, it automatically selects the power supply input from either the Type-C or the adapter.

4 Esight Overview

4.1 Windows Driver and .NET Framework Installation Guide

To ensure proper communication between the Quectel Power Mini and the Esight tool, and to achieve optimal performance, please follow the instructions below to complete driver deployment and runtime environment configuration.

4.1.1 Installation for Windows 10

Windows 10 features automatic hardware driver detection. Once the device is connected to the host via USB, the system will automatically search for, match and install the required driver without manual operation. Once recognition is completed, the device is ready for communication and use.

4.1.2 .NET Framework Dependency Installation (Optional)

Esight is a portable standalone application. Most office and R&D PCs come with the corresponding version of .NET Framework pre-installed, allowing direct execution.

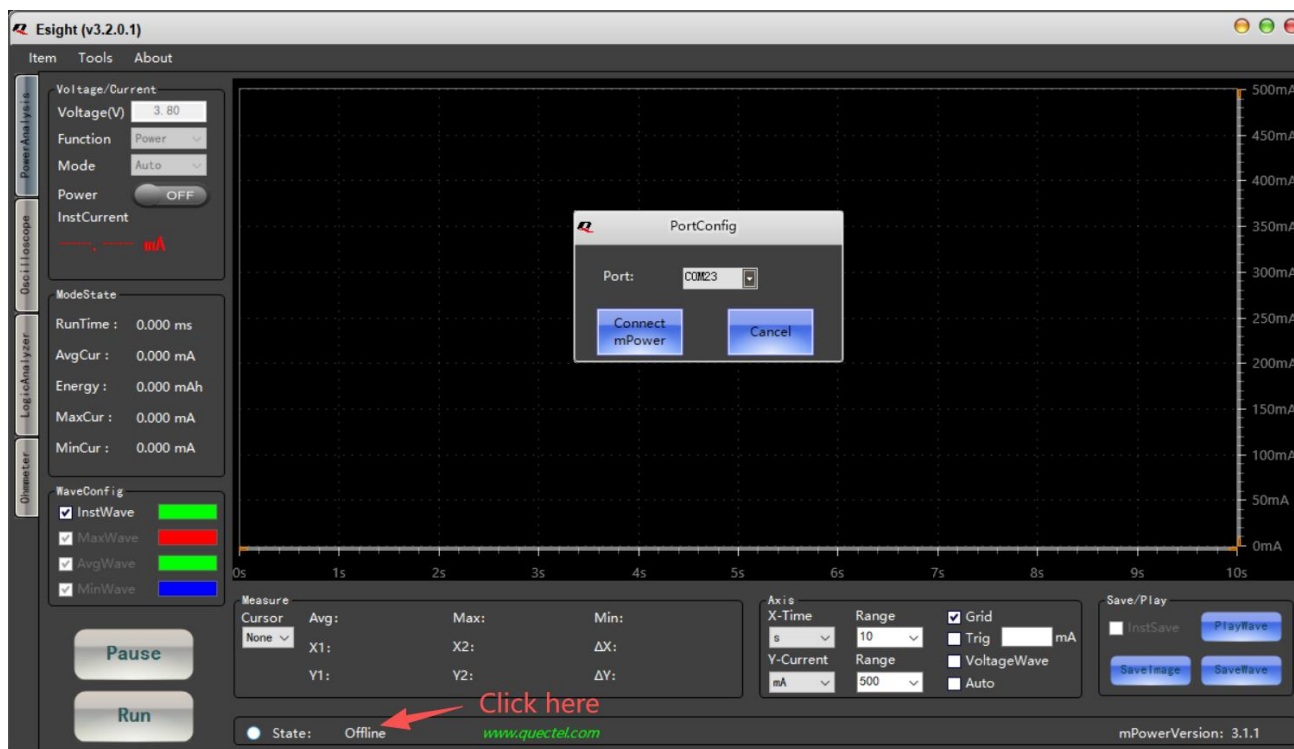
If double-clicking Esight prompts a missing .NET Framework runtime environment error, it means the corresponding dependency library needs to be installed manually.

1. Download *Microsoft.NET.exe* from https://developer.quectel.com/doc/files/quectel_power_mini/sources.zip.
2. Double-click the installer and follow default prompts to finish deployment, then run Esight normally.

4.2 Esight Tool Operations

4.2.1 Device-PC Connection

1. The Quectel Power Mini will automatically connect to the Esight tool by default when it is turned on.
2. If the automatic connection fails, click “**State: Offline**” at the lower-left corner, or navigate to the menu: “**Tools**”, then click “**PortConfig**” to select a port for connection.

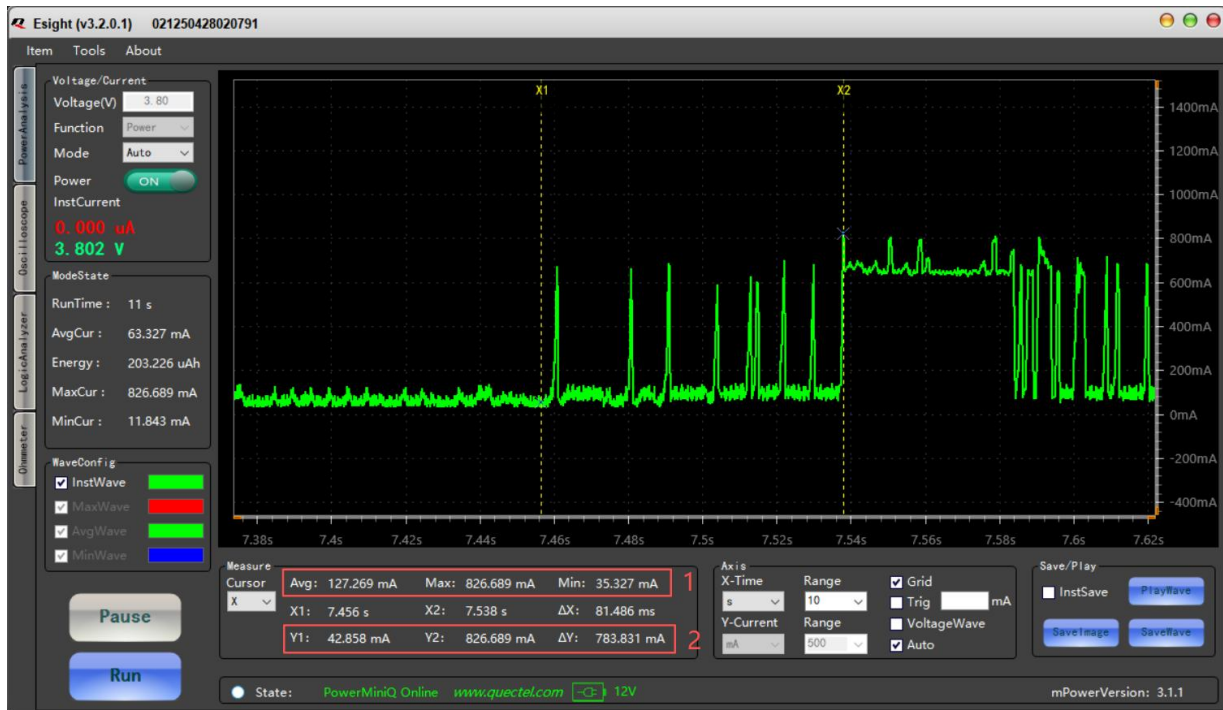


- For current-limiting protection: In power supply mode, if the current exceeds the current-limit threshold, the device will prompt an over-current warning by default and cut off power automatically. You can also set a current-limit value to enable the automatic cut-off function once the load current reaches the preset threshold, so as to protect the load from exceeding the set current.

4.2.2 Waveform Capture and Analysis

Capture: Taking the “Auto” mode as an example. After selecting the mode, set the “Power” to “ON”, and click “Run” to capture the real-time waveforms. Then click “Pause” to stop capturing.

Measure Cursor: Select “X” measure cursor, the X2 cursor will be automatically positioned at the maximum value of the displayed screen area. Place the mouse cursor over the cursor lines or the X1/X2 points on the horizontal axis, hold down the left mouse button to drag the measure cursor horizontally and release the button to stop dragging.

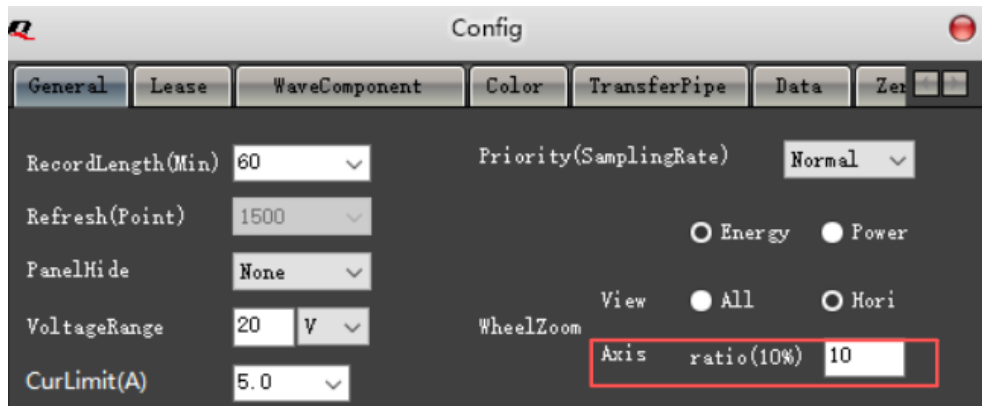


- Label 1: The average, maximum and minimum current of the waveform between the X1 and X2 cursors.
- Label 2: Y1 and Y2 represent the currents corresponding to points X1 and X2 on the horizontal axis (or the current corresponding to the Y measure cursor if Y measure cursor is available).

Zoom: Scrolling the mouse wheel over the waveform area performs horizontal zooming by default. Holding the “**Shift**” keypad while scrolling also enables horizontal zooming. Holding the “**Ctrl**” keypad while scrolling enables vertical zooming. Holding both “**Shift**” and “**Ctrl**” keypads while scrolling enables simultaneous horizontal and vertical zooming. Dragging a rectangle with the left mouse button allows zooming into the selected rectangular area.

Pan: Press and hold the mouse wheel to pan the waveform in any direction. On the X-axis, press and hold the left mouse button or use the mouse wheel to pan horizontally. On the Y-axis, press and hold the left mouse button or use the mouse wheel to pan vertically.

The panning sensitivity of the axes can be adjusted via “**Tools**”, then “**Config**” under “**Axis ratio**”, with the default set to 10%.



Trigger: After checking “Trig”, adjust the trigger threshold line position by dragging it or manually inputting a value. The system automatically stops waveform capture when the captured current reaches the trigger threshold.



Storage: Click “**SaveWave**” to save the current waveform. Long-term saving of all waveforms can also be achieved by checking the “**InstSave**”.

Hover the mouse cursor over the lower-right area of the coordinate axes, and a dialog box of saved waveforms will pop up automatically. Double-click to replay the waveform.



Simultaneous Dual-Waveform Playback: Click “Play Waveform” to bring up the file selection dialog, select two waveform files simultaneously, then click “Open”. The X-axis offset of dual waveforms can be configured via “Left Offset”/“Right Offset”, and the setting takes effect upon pressing the “Enter” keypad.



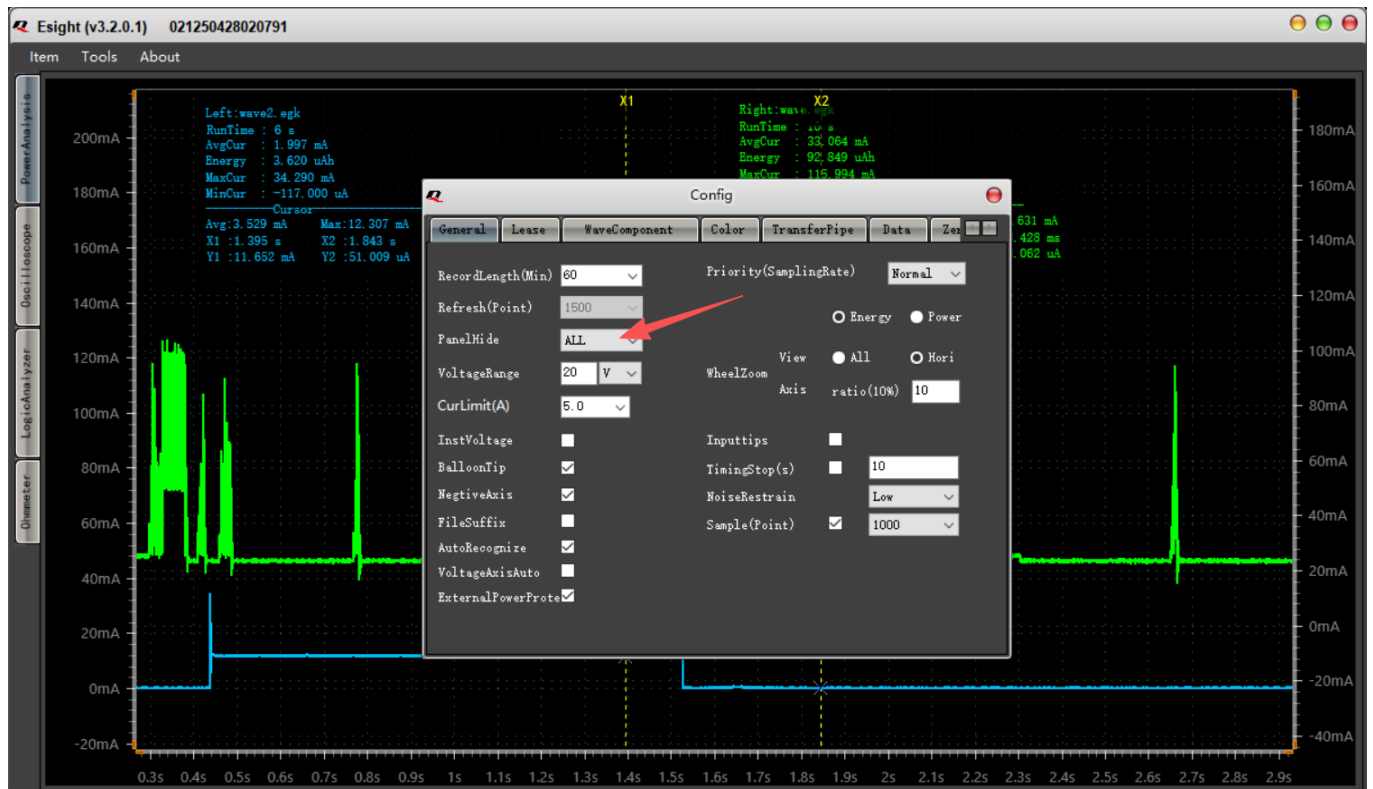
Zero Calibration: If the base current fluctuates far from zero after the device is turned on, perform manual zero calibration by clicking “Tools”, then “Zero Calibration”.

4.2.3 Other Custom Configuration

Esight provides abundant customized configuration options to meet requirements of diverse test scenarios. Click "**Tools**" → "**Config**" on the menu bar to open the parameter configuration panel for fine adjustment.

1. Panel Hiding

Set "PanelHide" to "ALL" to hide redundant setting panels, maximize the waveform display area and optimize observation for low-power consumption test data.



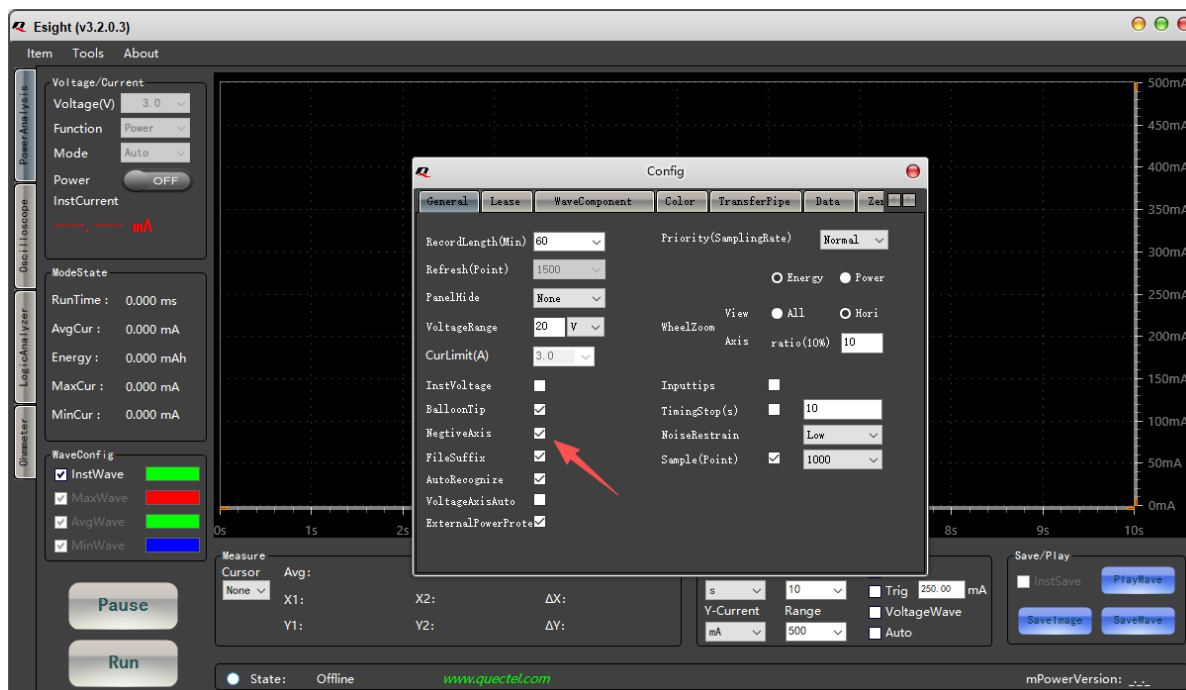
2. Dual Waveform Display (Voltage & Current)

Check "**VoltageWave**" to enable synchronous collection and display of voltage and current waveforms, facilitating joint analysis of power supply status and power consumption variations.



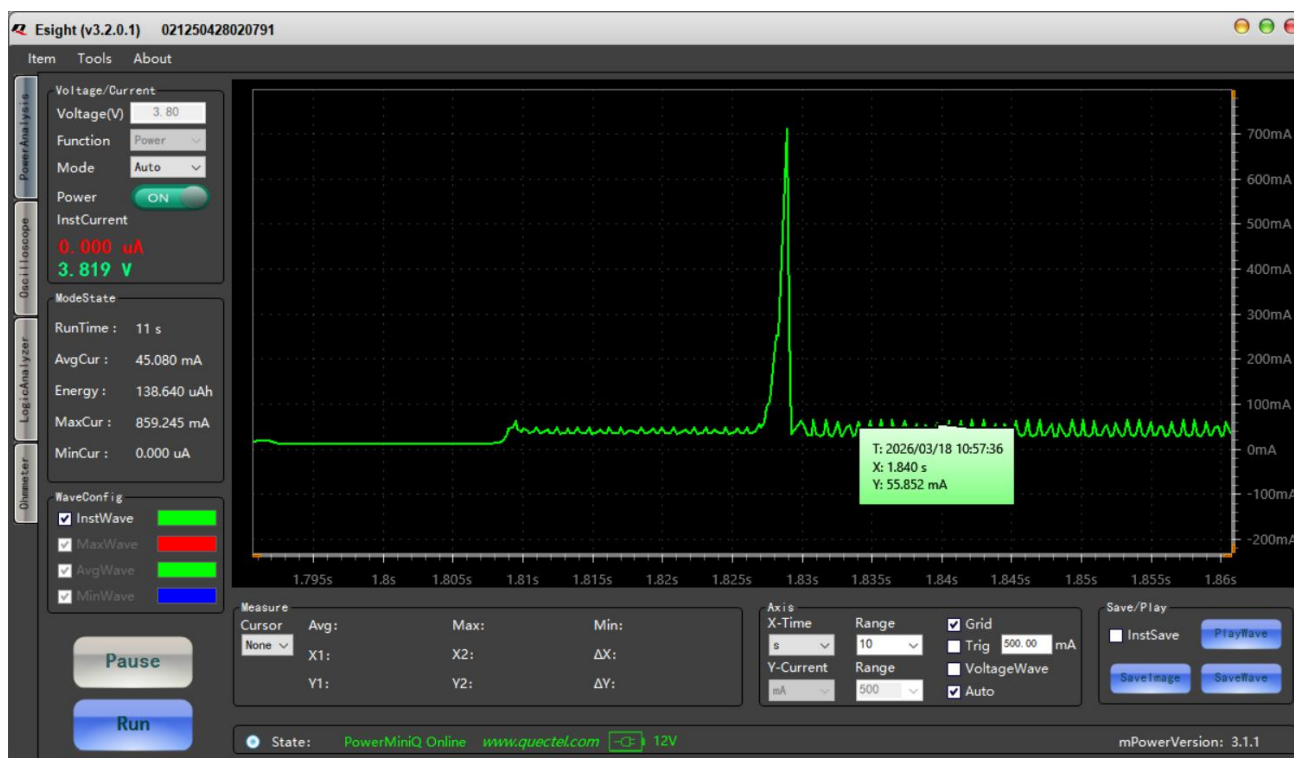
3. Negative Axis Display

Due to capacitive load characteristics, negative current may occur during testing. Esight enables negative axis display by default. Uncheck "**NegativeAxis**" to disable this function if negative current data is not required.



4. Timestamp

Click any position on the waveform to view the real-time Beijing timestamp and corresponding current information of the test point. Timestamps will be saved along with waveform data, enabling full restoration of time information for all test points during playback. Left-click blank area to close the display.



5. Acquisition Priority Configuration

The device only supports "Normal" sampling mode with a default sampling rate of 5 kps. Esight displays average electric quantity by default, with the option to switch to average power display.

4.3 System Upgrade Guide

To ensure complete functionality and operational stability, the Quectel Power Mini supports firmware upgrades via the Esight tool, enabling the latest features and optimizations. The upgrade procedure is as follows:

Step 1: Launch Esight, click **"Tools"** → **"Download"** on the menu bar to pop up the firmware download dialog.

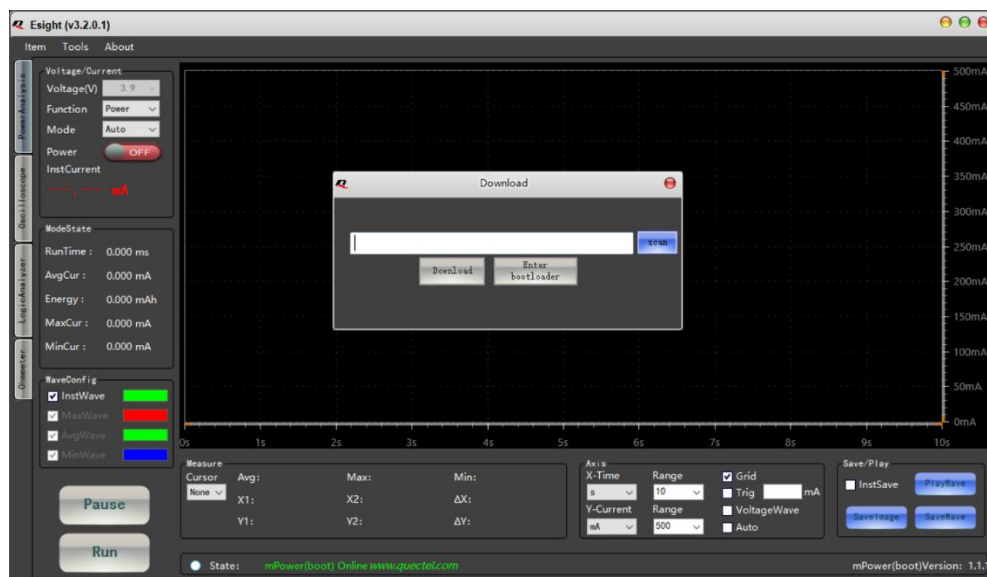
Step 2: Click **"Enter bootloader"** to enter the download mode. The device will power off automatically. Keep the download dialog open.



Step 3: Press the device power button to power it on. The software will indicate in the bottom-left corner that the device has entered Boot mode.

Step 4: Click "Scan" and select the local firmware file matching the device version.

Step 5: Click "Download" to start firmware upgrade. Do not disconnect the USB cable during the process.



Step 6: Once firmware download is complete, the device will automatically power off, and the upgrade process is finished.

5 Application Development Guide

Quectel Power Mini provides comprehensive secondary development support to satisfy the demands of automated testing and production line integration. Third-party programs can directly call the APIs encapsulated in the Windows DLL to control the device. The system supports synchronous data acquisition of up to 128 Quectel Power Mini units in theory. This chapter describes core APIs and provides Python demo codes. Refer to *API Description.txt* in the SDK package for complete interface definitions.

NOTE

You can download all supporting development materials from https://developer.quectel.com/doc/files/quectel_power_mini/sources.zip or by scanning the QR code on the device packaging.

5.1 API Overview

Table 3: API Overview

Function	Description
<i>Init()</i>	Initializes mPower1203 dynamic library.
<i>OpenPort()</i>	Establishes a connection with one mPower1203 device.
<i>ClosePort()</i>	Disconnects one mPower1203 device.
<i>SetType()</i>	Configures the device as power supply mode or ammeter mode.
<i>SetMode()</i>	Configures operating accuracy or range mode of the device.
<i>SetVoltage()</i>	Configures output voltage of the device under power supply mode.
<i>SetPower()</i>	Turns on/off device power output under power supply mode.
<i>StartSample()</i>	Starts data acquisition.
<i>StopSample()</i>	Stops data acquisition.
<i>GetCurrent()</i>	Retrieves continuous current sampling data.
<i>GetFwVersion()</i>	Retrieves firmware version of mPower1203 device.
<i>GetDllVersion()</i>	Retrieves the version of mPower1203.dll.

5.2 API Description

5.2.1 Init

This function initializes the mPower1203 dynamic library.

- **Prototype**

```
int Init()
```

- **Parameter**

None

- **Return Value**

0	Successful execution
Other values	Failed execution

5.2.2 OpenPort

This function establishes a connection with one mPower1203 device.

- **Prototype**

```
int OpenPort(string port)
```

- **Parameter**

port:

[In] Port number of the mPower1203 device.

NOTE

After connecting the device to PC via USB, check the corresponding port number in the PC's Device Manager.

- **Return Value**

Device ID (positive integer)	Successful execution
Less than 0	Failed execution

5.2.3 ClosePort

This function disconnects one mPower1203 device.

- **Prototype**

```
int ClosePort(int id)
```

- **Parameter**

id:

[In] Device ID.

- **Return Value**

0 Successful execution

Other values Failed execution

5.2.4 SetType

This function configures the device as power supply mode or ammeter mode.

- **Prototype**

```
int SetType(int id, int type)
```

- **Parameter**

id:

[In] Device ID.

type:

[In] Device operating mode. See **Chapter 5.2.4.1** for details.

- **Return Value**

0 Successful execution

Other values Failed execution

5.2.4.1 MPOWER_TYPE

The enumeration of device operating modes:

```
typedef enum {
    MPOWER_POWER          0
    MPOWER_AMPERE         1
}MPOWER_TYPE;
```

- **Member**

Member	Description
<i>MPOWER_POWER</i>	Power supply mode: The device outputs voltage to power the DUT.
<i>MPOWER_AMPERE</i>	Ammeter mode: The device only measures current without power output.

5.2.5 SetMode

This function configures operating accuracy or range mode of the device.

- **Prototype**

```
int SetMode(int id, int mode)
```

- **Parameter**

id:

[In] Device ID.

mode:

[In] Operating accuracy or range mode of the device. See **Chapter 5.2.5.1** and **Chapter 5.2.5.2** for details.

- **Return Value**

0 Successful execution

Other values Failed execution

5.2.5.1 MPOWER_POWER_MODE

The enumeration of operating accuracy in power supply mode:

```
typedef enum {
    MPOWER_POWER_AUTO          0
    MPOWER_POWER_NORMAL        1
    MPOWER_POWER_PERCISION     2
    MPOWER_POWER_ULTRA         3
}MPOWER_POWER_MODE;
```

● Member

Member	Description
<i>MPOWER_POWER_AUTO</i>	Auto mode.
<i>MPOWER_POWER_NORMAL</i>	Normal mode.
<i>MPOWER_POWER_PERCISION</i>	Precision mode.
<i>MPOWER_POWER_ULTRA</i>	Ultra-high precision mode.

5.2.5.2 MPOWER_AMPERE_MODE

The enumeration of range in ammeter mode:

```
typedef enum {
    MPOWER_AMPERE_AUTO          0
    MPOWER_AMPERE_NORMAL        1
    MPOWER_AMPERE_PERCISION     2
}MPOWER_AMPERE_MODE;
```

● Member

Member	Description
<i>MPOWER_AMPERE_AUTO</i>	Auto mode.
<i>MPOWER_AMPERE_NORMAL</i>	Normal mode.
<i>MPOWER_AMPERE_PERCISION</i>	Precision mode.

5.2.6 SetVoltage

This function configures output voltage of the device under power supply mode.

● Prototype

```
int SetVoltage(int id, int voltage)
```

● Parameter

id:

[In] Device ID.

voltage:

[In] Output voltage. Range: 0.6–6.05. Unit: V. See **Chapter 5.2.6.1** for details.

● Return Value

0 Successful execution

Other values Failed execution

5.2.6.1 MPOWER_VOLTAGE_E

The enumeration of output voltage:

```
typedef enum {
    V0_6,      0
    V0_65,    1
    V0_7,      2
    V0_75,    3
    V0_8,      4
    V0_85,    5
    V0_9,      6
    V0_95,    7
    V1_0,      8
    V1_05,    9
    V1_1,     10
    V1_15,    11
    V1_2,     12
    V1_25,    13
    V1_3,     14
    V1_35,    15
    V1_4,     16
    V1_45,    17
}
```

V1_5,	18
V1_55,	19
V1_6,	20
V1_65,	21
V1_7,	22
V1_75,	23
V1_8,	24
V1_85,	25
V1_9,	26
V1_95,	27
V2_0,	28
V2_05,	29
V2_1,	30
V2_15,	31
V2_2,	32
V2_25,	33
V2_3,	34
V2_35,	35
V2_4,	36
V2_45,	37
V2_5,	38
V2_55,	39
V2_6,	40
V2_65,	41
V2_7,	42
V2_75,	43
V2_8,	44
V2_85,	45
V2_9,	46
V2_95,	47
V3_0,	48
V3_05,	49
V3_1,	50
V3_15,	51
V3_2,	52
V3_25,	53
V3_3,	54
V3_35,	55
V3_4,	56
V3_45,	57
V3_5,	58
V3_55,	59
V3_6,	60
V3_65,	61

V3_7,	62
V3_75,	63
V3_8,	64
V3_85,	65
V3_9,	66
V3_95,	67
V4_0,	68
V4_05,	69
V4_1,	70
V4_15,	71
V4_2,	72
V4_25,	73
V4_3,	74
V4_35,	75
V4_4,	76
V4_45,	77
V4_5,	78
V4_55,	79
V4_6,	80
V4_65,	81
V4_7,	82
V4_75,	83
V4_8,	84
V4_85,	85
V4_9,	86
V4_95,	87
V5_0,	88
V5_05,	89
V5_1,	90
V5_15,	91
V5_2,	92
V5_25,	93
V5_3,	94
V5_35,	95
V5_4,	96
V5_45,	97
V5_5,	98
V5_55,	99
V5_6,	100
V5_65,	101
V5_7,	102
V5_75,	103
V5_8,	104
V5_85,	105

```
V5_9,      106
V5_95,     107
V6_0,      108
V6_05      109
}MPOWER_VOLTAGE;
```

● Member

Member	Description
V0_6	0.6 V
V0_65	0.65 V
V0_7	0.7 V
V0_75	0.75 V
...	...(Step: 0.05 V, follow the rule in sequence)
V6_0	6 V
V6_05	6.05 V

5.2.7 SetPower

This function turns on or off device power output under power supply mode.

● Prototype

```
int SetPower(int id, int on)
```

● Parameter

id:

[In] Device ID.

on:

[In] Power switch operation. See **Chapter 5.2.7.1** for details.

● Return Value

0 Successful execution

Other values Failed execution

5.2.7.1 MPOWER_POWER_SUPPLY

The enumeration of power switch operations:

```
typedef enum {
    MPOWER_POWER_OFF        0
    MPOWER_POWER_ON         1
}MPOWER_POWER_SUPPLY;
```

- **Member**

Member	Description
<i>MPOWER_POWER_OFF</i>	Turn off device power output.
<i>MPOWER_POWER_ON</i>	Turn on device power output.

5.2.8 StartSample

This function starts data acquisition.

- **Prototype**

```
int StartSample(int id)
```

- **Parameter**

id:
[In] Device ID.

- **Return Value**

0 Successful execution
Other values Failed execution

5.2.9 StopSample

This function stops data acquisition.

- **Prototype**

```
int StopSample(int id)
```

- **Parameter**

id:

[In] Device ID.

- **Return Value**

0 Successful execution

Other values Failed execution

5.2.10 GetCurrent

This function retrieves continuous current sampling data.

- **Prototype**

```
double* GetCurrent(int id)
```

- **Parameter**

id:

[In] Device ID.

- **Return Value**

Array of current data, length variable but always even. Data output format:

[0]: timestamp(us) [1]: current(uA) [2]: timestamp [3]: current(uA) ...

NOTE

Due to the high data acquisition rate, it is recommended to call this function in an independent thread and avoid time-consuming operations such as delay functions and log output. If duplicate timestamps are acquired, retain only one record.

5.2.11 GetFwVersion

This function retrieves firmware version of mPower1203 device.

- **Prototype**

```
char* GetFwVersion(int id)
```

- **Parameter**

id:

[In] Device ID.

- **Return Value**

Version string: [0] Major version [1] Minor version [2] Revision
-1

Successful execution
Failed execution

5.2.12 GetDllVersion

This function retrieves the version number of mPower1203.dll.

- **Prototype**

```
char* GetDllVersion()
```

- **Parameter**

None

- **Return Value**

DLL version string: [0] Major version [1] Minor version [2] Revision [3] Build number
-1

Successful execution
Failed execution

5.3 Python Demo Code

The following code demonstrates how to call DLL APIs via Python to control two devices simultaneously for data acquisition:

```
import os
import time
import clr
import sys
from datetime import datetime, timedelta
from collections import namedtuple

# Customized sampling duration (unit: second).
SAMPLE_TIME = 5
# COM ports of two power analyzers; modify according to actual configuration.
CH1_PORT_NAME = "COM10"
```

```
CH2_PORT_NAME = "COM8"
```

```
MPOWER_TYPE = namedtuple('MPOWER_TYPE','POWER AMPERE')._make(range(2))
MPOWER_POWER_MODE = namedtuple('MPOWER_POWER_MODE','AUTO NORMAL
PERCISION')._make(range(3))
MPOWER_VOLTAGE = namedtuple('MPOWER_VOLTAGE','V0_6 V0_65 V0_7 V0_75 V0_8 V0_85
V0_9 V0_95 V1_0 V1_05 \
V1_1 V1_15 V1_2 V1_25 V1_3 V1_35 V1_4 V1_45 V1_5 V1_55 V1_6 V1_65 V1_7 V1_75 V1_8 V1_85
V1_9 V1_95 V2_0 V2_05 \
V2_1 V2_15 V2_2 V2_25 V2_3 V2_35 V2_4 V2_45 V2_5 V2_55 V2_6 V2_65 V2_7 V2_75 V2_8 V2_85
V2_9 V2_95 V3_0 V3_05 \
V3_1 V3_15 V3_2 V3_25 V3_3 V3_35 V3_4 V3_45 V3_5 V3_55 V3_6 V3_65 V3_7 V3_75 V3_8 V3_85
V3_9 V3_95 V4_0 V4_05 \
V4_1 V4_15 V4_2 V4_25 V4_3 V4_35 V4_4 V4_45 V4_5 V4_55 V4_6 V4_65 V4_7 V4_75 V4_8 V4_85
V4_9 V4_95 V5_0 V5_05 \
V5_1 V5_15 V5_2 V5_25 V5_3 V5_35 V5_4 V5_45 V5_5 V5_55 V5_6 V5_65 V5_7 V5_75 V5_8 V5_85
V5_9 V5_95 V6_0 V6_05')._make(range(110))
MPOWER_POWER = namedtuple('MPOWER_POWER','OFF ON')._make(range(2))
```

```
def format_run_time(seconds):
    """Convert seconds to xh ym zs format"""
    hours = int(seconds // 3600)
    remaining_seconds = seconds % 3600
    minutes = int(remaining_seconds // 60)
    seconds = int(remaining_seconds % 60)

    # Concatenate the non-zero parts.
    parts = []
    if hours > 0:
        parts.append(f"{hours}h")
    if minutes > 0:
        parts.append(f"{minutes}m")
    if seconds > 0 or not parts: # Make sure to display at least the seconds (even if all are 0).
        parts.append(f"{seconds}s")

    return ' '.join(parts)
```

```
def print_t(*args, **kwargs):
    """
    Print function with timestamp, the same function as print; add [YYYY-MM-DD HH:MM:SS] prefix
    before log
    """
    # Obtain the current timestamp (accurate to the second), and format it as [Year-Month-Day
```



```
Hour:Minute:Second]
    timestamp = datetime.now().strftime("[%Y-%m-%d %H:%M:%S]")

    # Pass the timestamp as the first parameter and concatenate all the parameters passed by the user.
    # Use builtins.print to call the original print function to avoid recursive calls.
    print(timestamp, *args, **kwargs)

# Load mPower1203 dynamic library. Use AddReference if FindAssembly fails, no .dll suffix required.
clr.AddReference('mPower1203')
from mPower1203 import *
mPower = ClassmPower()

# Initialize dynamic library
Res = mPower.Init()
print_t(f"init:{Res}")

# Open the first device.
Ch1 = mPower.OpenPort(CH1_PORT_NAME)
if Ch1 < 0:
    print_t(f"OpenPort {CH1_PORT_NAME} fail:{Ch1}")
    sys.exit(Ch1)

print_t(f"OpenChannel:{Ch1}")

# Configure the first device as power supply mode.
mPower.SetType(Ch1,MPOWER_TYPE.POWER)

# Set output voltage to 3 V.
mPower.SetVoltage(Ch1,MPOWER_VOLTAGE.V3_0)

# Open the second device.
Ch2 = mPower.OpenPort(CH2_PORT_NAME)
if Ch2 < 0:
    print_t(f"OpenPort {CH2_PORT_NAME} fail:{Ch2}")
    sys.exit(Ch2)
print_t(f"OpenChannel:{Ch2}")

# Configure the second device as power supply mode.
mPower.SetType(Ch2,MPOWER_TYPE.POWER)

# Set output voltage to 4 V.
mPower.SetVoltage(Ch2,MPOWER_VOLTAGE.V4_0)
```

```
# Turn on device power output.
mPower.SetPower(Ch1,MPOWER_POWER.ON)
mPower.SetPower(Ch2,MPOWER_POWER.ON)
time.sleep(1)

# Start data acquisition.
mPower.StartSample(Ch1)
mPower.StartSample(Ch2)
time.sleep(1)

start_time = time.perf_counter()
test_cnt = 0
while True:
    test_cnt += 1
    print_t(f"-----On the {test_cnt}th test, the running time was
{format_run_time(time.perf_counter() - start_time)}-----")

# Stop data acquisition.
Res = mPower.StopSample(Ch1)
Res = mPower.StopSample(Ch2)
time.sleep(1)

# Close communication ports.
Res = mPower.ClosePort(Ch1)
print_t(f"CloseChannel:{Ch1}")
Res = mPower.ClosePort(Ch2)
print_t(f"CloseChannel:{Ch2}")
```

6 Appendix References

Table 4: Terms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
DC	Direct Current
DUT	Device Under Test
LPWA	Low-Power Wide-Area Network
OTA	Over-the-air
PC	Personal Computer
USB	Universal Serial Bus