

# Quectel Power Mini 便携式直流功耗分析仪用户手册

版本：1.0

日期：2026-06-03

状态：受控文件



上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司  
上海市松江区泗泾镇外婆泾路 8 号 邮编：201601  
电话：+86 21 5108 6236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：<https://www.quectel.com.cn/contact>。

如需技术支持或反馈我司技术文档中的问题，请随时登录网址：  
<https://www.quectel.com.cn/contact?tab=t> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

移远通信提供该文档内容以支持您的产品设计。您须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。您知悉并同意，移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档，增加、修改或重述后的文档对您具有约束力。

## 使用和披露限制

### 许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

### 版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，任何形式的购买都不可被视为授予除正常的非独家、免版税的产品使用许可之外的任何许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法、侵权行为，移远通信有权追究法律责任。

### 商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

### 第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

## 隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

## 免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何明示、暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 ©上海移远通信技术股份有限公司 2026，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2026.**

# 文档历史

## 修订记录

版本	日期	作者	变更表述
-	2026-06-03	Dustin Wei	文档创建
1.0	2026-06-03	Dustin Wei	受控版本

## 安全须知

在操作、安装本产品时，必须仔细阅读并严格遵守产品标识以及用户手册中的安全须知。

本文档中提及的“危险”、“警告”、“注意”事项，仅作为对安全事项的补充，并不代表所有安全事项。

## 通用安全

### 安装与使用

安装、操作、维护本产品的人员须了解所有安全注意事项，并在掌握正确的方法后进行安装、操作和维护。请遵循以下基本安装要求：

- 必须由移远通信授权人员检修产品。
- 必须由移远通信授权人员更换产品部件。
- 安装后，请严格按照出厂时的装配方式进行操作，否则可能会影响产品的性能表现。

### 接地

- 接地操作必须符合国家或地区的安装规范和相关标准。
- 安装产品前，必须先完成接地操作；拆除产品时，最后拆除地线。
- 操作产品前请进行接地检查，确保产品接地良好。严禁在产品未接地或接地不良的情况下对产品进行任何操作。

### 人身安全

- 严禁在危险环境下安装、操作和维护产品，如雷雨天气、易燃环境。
- 确保产品远离儿童，防止意外发生。

### 产品安全

- 确保产品远离火源和热源，严禁将其投入火中，防止产品损坏或发生火灾。
- 严禁对产品进行剧烈振动、冲击或挤压，以避免内部元件受损或产品故障。
- 注意防水措施，确保产品接口免遭水淋或水浸，以防止产品损坏或发生电路故障。
- 严禁用户自行拆卸或维修产品，以免损坏产品内部结构或引发安全问题。
- 保证产品在任何一种安装方式下的牢固可靠，如桌面、挂墙或机架安装。
- 对产品进行开机操作前，确保接地良好；关机操作完成后再拆除地线。
- 拆装操作必须佩戴防静电手套，请使用适当的工具拆卸或安装产品。
- 严禁遮挡或堵塞产品通风面板。
- 严禁拔除产品密封胶塞，如天线胶塞。

## 移动场景安全

- 开车时请勿使用手持终端设备，即使设备支持免提功能。请先停车，再打电话。
- 飞行前请关闭终端设备。若设备具备“飞行模式”，请在飞行前开启该模式以禁用所有无线功能，以免干扰飞机通讯系统。飞行过程中禁止开启或使用任何无线功能，违反该提示可能影响飞行安全，甚至触犯法律。关于航班上电子设备使用的额外限制，请向航空公司工作人员咨询。
- 出入医院或健康看护场所时，请注意是否存在终端设备使用限制。射频干扰可能会导致医疗设备运行失常，请根据要求关闭终端设备。
- 终端设备并不保障在任何情况下均能进行有效连接，例如在设备欠费或(U)SIM卡无效时。若设备支持紧急呼叫功能，请使用紧急呼叫，同时请确保设备开机并且位于信号强度足够的区域。由于网络连接存在不确定性，严禁将本设备作为紧急情况下的唯一通讯工具。
- 终端设备在开机时会接收和发射射频信号。当靠近电视、收音机、电脑或者其他电子设备时都会产生射频干扰。
- 确保终端设备远离易燃易爆品。当靠近加油站、油库、化工厂或爆炸作业场所时，请关闭终端设备。在任何有潜在爆炸危险的场所操作电子设备均存在安全隐患。

## 电气安全

### 恶劣环境

- 严禁在雷雨天气对产品进行交流电操作，否则会有致命危险。
- 严禁在雷雨天气对产品进行室外设备箱体、抱杆等作业，否则会有致命危险。
- 严禁在高压电源供电情况下，不规范地进行安装、操作和维护，否则会有致命危险。

### 供电电源

- 连接产品电源线前，必须确认电源电压、频率和容量等参数符合产品要求。
- 连接产品电源线前，必须优先确认电源线正负极标识正确再进行连接。
- 拆除产品电源线前，必须优先关闭电源使产品处于关机状态。
- 严禁在产品上电状态下，带电拆除电源线，避免产生电火花导致身体损伤。
- 产品电源线与产品连接后，确保连接位置牢固不松动。
- 电池供电时，请注意使用经测试合格的电池包。

### 静电

- 尽可能在配备防静电工作台和防静电地垫的区域内操作。
- 为防止人体摩擦或移动产生的静电损坏产品，安装、操作和维护产品前，先进行静电放电操作。
- 触摸、手持产品前，必须佩戴防静电手环，并将手环的接地端子接地。

# 目录

文档历史 .....	3
安全须知 .....	4
通用安全 .....	4
电气安全 .....	5
目录 .....	6
表格索引 .....	8
图片索引 .....	9
<b>1 产品概述</b> .....	<b>10</b>
<b>2 外观及接口</b> .....	<b>12</b>
2.1 外观 .....	12
2.2 接口 .....	13
<b>3 接线说明</b> .....	<b>15</b>
3.1 接线图 .....	15
3.2 供电方式 .....	15
<b>4 Esight 使用指导</b> .....	<b>16</b>
4.1 Windows 驱动及 NetFramework 安装指南 .....	16
4.1.1 Windows 10 系统安装说明 .....	16
4.1.2 .NET Framework 依赖组件安装（可选配置） .....	16
4.2 Esight 工具操作说明 .....	16
4.2.1 设备-PC 连接 .....	16
4.2.2 波形捕获及分析 .....	17
4.2.3 其他自定义配置 .....	20
4.3 系统升级指导 .....	23
<b>5 应用开发指导</b> .....	<b>25</b>
5.1 函数概览 .....	25
5.2 函数详解 .....	26
5.2.1 Init .....	26
5.2.2 OpenPort .....	26
5.2.3 ClosePort .....	27
5.2.4 SetType .....	27
5.2.4.1 MPOWER_TYPE .....	28
5.2.5 SetMode .....	28
5.2.5.1 MPOWER_POWER_MODE .....	29
5.2.5.2 MPOWER_AMPERE_MODE .....	29
5.2.6 SetVoltage .....	30
5.2.6.1 MPOWER_VOLTAGE_E .....	30
5.2.7 SetPower .....	33

5.2.7.1	MPOWER_POWER_SUPPLY .....	34
5.2.8	StartSample.....	34
5.2.9	StopSample.....	34
5.2.10	GetCurrent.....	35
5.2.11	GetFwVersion.....	35
5.2.12	GetDllVersion .....	36
5.3	Python 示例代码.....	36
<b>6</b>	<b>附录 术语缩写.....</b>	<b>40</b>



## 表格索引

表 1: 规格参数 .....	10
表 2: 接口描述表 .....	14
表 3: 函数概览 .....	25
表 4: 术语缩写 .....	40

图片索引

图 1：正面图和背面图 ..... 12

图 2：左视图..... 12

图 3：右视图..... 13

图 4：接口示意图 ..... 14

图 5：接线示意图 ..... 15

# 1 产品概述

Quectel Power Mini 是一款专为物联网模组通信及嵌入式软件开发场景打造的高精度直流功耗分析仪。该设备集便携式直流程控供电、高精度电流测试、动态电流波形实时分析及高性价比于一体，能够高效协助开发人员在产品研发阶段进行精准的功耗测试与动态测量分析。

配套的 Esight 上位机软件完美兼容 Windows 10 及其以上版本系统（支持 x86/x64 架构），提供直观的数据交互与可视化分析界面。

## 行业痛点与挑战：

- **测试精度不足：**随着手持及穿戴设备对低功耗要求的日益严苛，传统大功率直流电源虽成本低廉，但其电流测试精度已无法满足微功耗场景的测试需求。
- **缺乏动态分析能力：**常规电源设备无法提供电流波形的实时显示，存储、回放及分析功能，无法支撑设备功耗动态特性的精准研判，适配不了模组通信等场景的精细化测试需求。
- **设备成本与便携性受限：**专业级直流电源分析仪价格昂贵（通常数万至数十万元），且体积笨重，极难满足户外无 220 V 交流电环境下的移动调试与高频次搬运需求。

## 核心解决方案与优势：

- **极致便携与灵活部署：**Quectel Power Mini 体积仅为常规智能手机大小，极大提升了设备的携带便捷性。支持标配适配器、USB 适配器供电；在户外或无交流电场景下，可直接通过笔记本电脑（同时实现通信与供电）或移动电源（充电宝）驱动，完美适配野外实地测试。
- **高精度动态电流测量：**设备支持 nA 级至 A 级的超宽动态电流测量范围，测试精度高达 50 nA。配合 Esight 工具，可精准捕捉并反映电流的动态变化，支持波形的实时显示、深度分析、存储与回放，帮助开发人员快速定位异常功耗点。
- **高效的自动化集成：**提供完善的二次开发 API 接口，支持同时控制多台功耗仪，可轻松接入客户现有的自动化测试系统，大幅提升产线测试或研发验证的效率。

Quectel Power Mini 的规格参数如下：

表 1：规格参数

参数	说明
产品型号	<ul style="list-style-type: none"> <li>● Quectel Power Mini</li> </ul>
产品名称	<ul style="list-style-type: none"> <li>● 便携式直流功耗分析仪</li> </ul>
物理参数	<ul style="list-style-type: none"> <li>● 尺寸：103 mm × 70 mm × 24 mm（长 × 宽 × 高）</li> </ul>
	<ul style="list-style-type: none"> <li>● 重量：约 170 g</li> </ul>
	<ul style="list-style-type: none"> <li>● 外壳材质：金属</li> </ul>

硬件接口	● USB 接口: Type-A、USB 2.0
	● 被测设备输出: 红黑测试线
	● 电源输入: DC 12 V 适配器、Type-C、移动电源
电气参数	● DC 输入电压: 12 V (典型值), 最高达 13 V
	● Type-A 输入电压: 5 V
	● Type-C 输入电压: 5 V、9 V、12 V
	● 输出电压范围: 0.6~6.5 V (50 mV 步进)
	● 输出电压纹波: $\leq 30 \text{ mVpp}$
	● 负载调整率 <sup>1</sup> : $\leq 5\%$
	● 最大输出功率: 25 W
	● 电流量程: nA ~ 5 A
工作温度范围	● 电流分辨率: nA ~ $\mu\text{A}$ 级
	● 采样率: 5 ksps
	● 0~60 °C
测量模式	● 精密模式/自动模式/普通模式
PC 工具	● Esight (支持 Windows 7 及以上系统)
SDK 支持	● C#、Python、VB、C++、C、LabVIEW 等
输出保护	● 支持
固件升级	● 支持 (通过 USB)

<sup>1</sup> 测试条件: 电压 = 4 V, 负载动态范围 = 10 mA ~ 2.0 A。

## 2 外观及接口

本章节提供产品外观及接口图。图片仅供参考，实际的产品外观和标签信息，请参照移远通信的产品实物。

### 2.1 外观



图 1：正面图和背面图

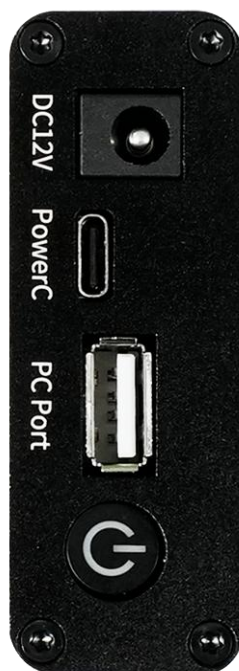


图 2：左视图



图 3: 右视图

## 2.2 接口

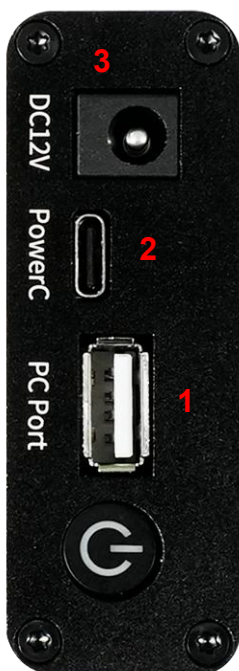




图 4：接口示意图

表 2：接口描述表

序号	接口名称	说明
1	Type-A USB	USB 2.0，用于连接 PC 通信及供电
2	Type-C	电源输入，支持 5 V、9 V、12 V 充电器或移动电源
3	DC 电源接口	DC 12 V 适配器输入
4	红色测试线	被测设备输出（正极）
5	黑色测试线	被测设备输出（负极）

## 3 接线说明

### 3.1 接线图

Quectel Power Mini 接线示意图如下：

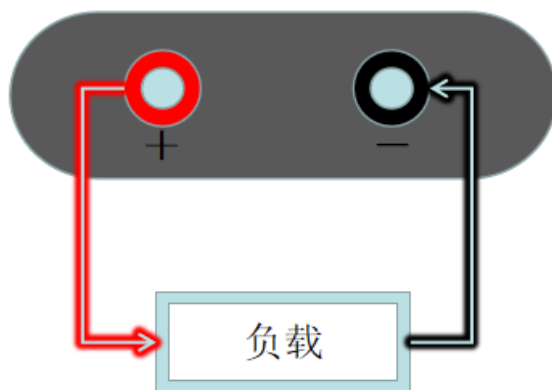


图 5：接线示意图

- 建议为负载提供 0.6~6.5 V 的可调电压。
- 可实时监测并显示负载电压、电流的波形。

### 3.2 供电方式

- 具备市电插座的环境下，推荐使用 12 V 适配器或者 USB（支持 Type-C 快充）为 Quectel Power Mini 供电。
- 户外无市电环境下，可通过笔记本电脑或移动电源直接为 Quectel Power Mini 供电。

#### 备注

Quectel Power Mini 的 Type-A 接口具备供电和 PC 通信功能；如果设备插入 Type-C 供电口（Type-C 一端接入电脑、一端接入设备）或适配器，将自动选择 Type-C 或适配器供电。



## 4 Esight 使用指导

### 4.1 Windows 驱动及 NetFramework 安装指南

为确保 Quectel Power Mini 设备能够与上位机软件 Esight 正常通信并发挥最佳性能，请参考以下说明完成驱动程序的部署及运行环境的配置。

#### 4.1.1 Windows 10 系统安装说明

Windows 10 操作系统具备自动硬件驱动识别能力，设备通过 USB 接口连接至上位机后，系统将自动检索并完成驱动匹配安装，无需人工手动干预，等待系统识别完成即可正常通信使用。

#### 4.1.2 .NET Framework 依赖组件安装（可选配置）

Esight 上位机工具为绿色免安装程序，日常主流办公及研发电脑均已预装对应版本 .NET Framework 运行库，可直接启动运行。

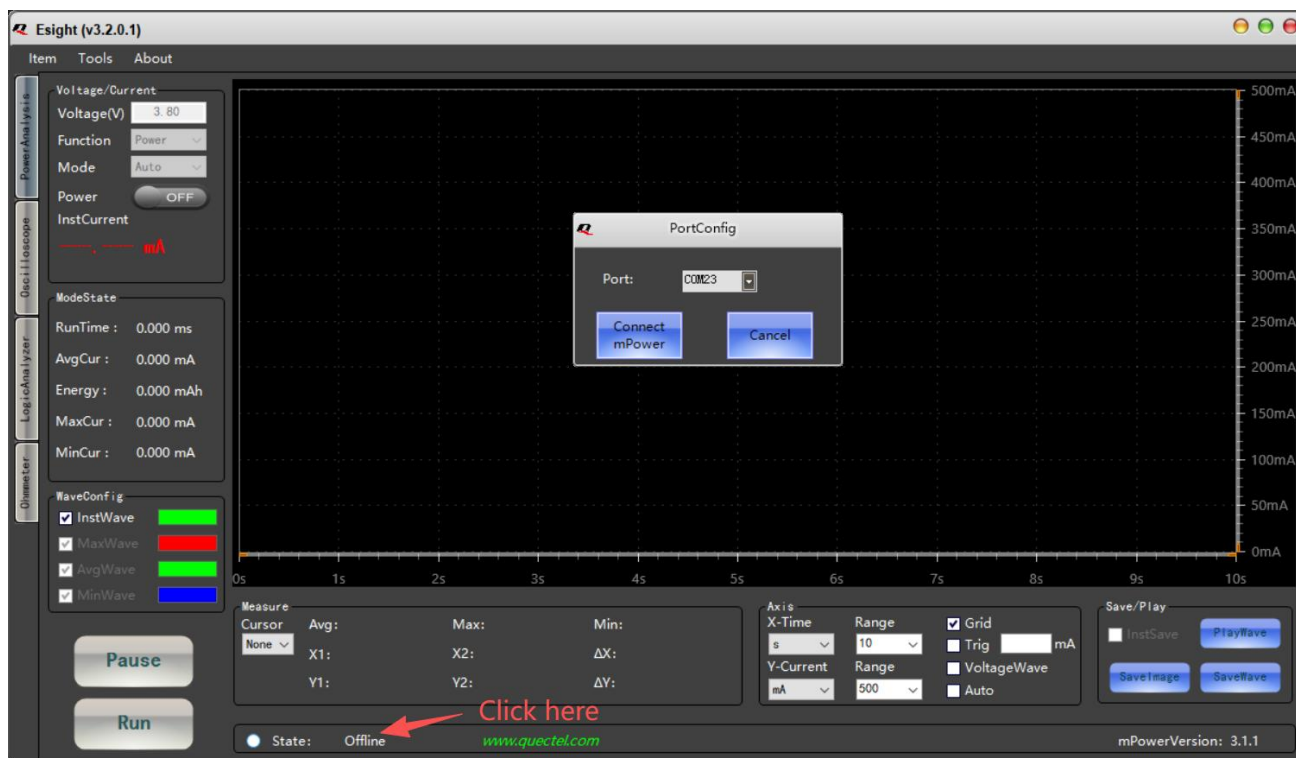
若在双击启动 Esight 工具时，系统弹出缺少 .NET Framework 运行环境异常提示，说明当前系统未部署对应依赖组件，需手动安装适配运行库。

访问 [https://developer.quectel.com/doc/files/quectel\\_power\\_mini/sources.zip](https://developer.quectel.com/doc/files/quectel_power_mini/sources.zip) 获取 *Microsoft.NET.exe* 安装程序后，双击并按照默认指引完成环境部署，随后即可顺利运行 Esight 工具。

### 4.2 Esight 工具操作说明

#### 4.2.1 设备-PC 连接

1. Quectel Power Mini 开机后默认自动连接至 Esight 工具。
2. 若未自动连接，可点击左下角“**State: Offline**”或点击菜单中的“**Tools**”，在“**PortConfig**”中选择对应端口完成连接。

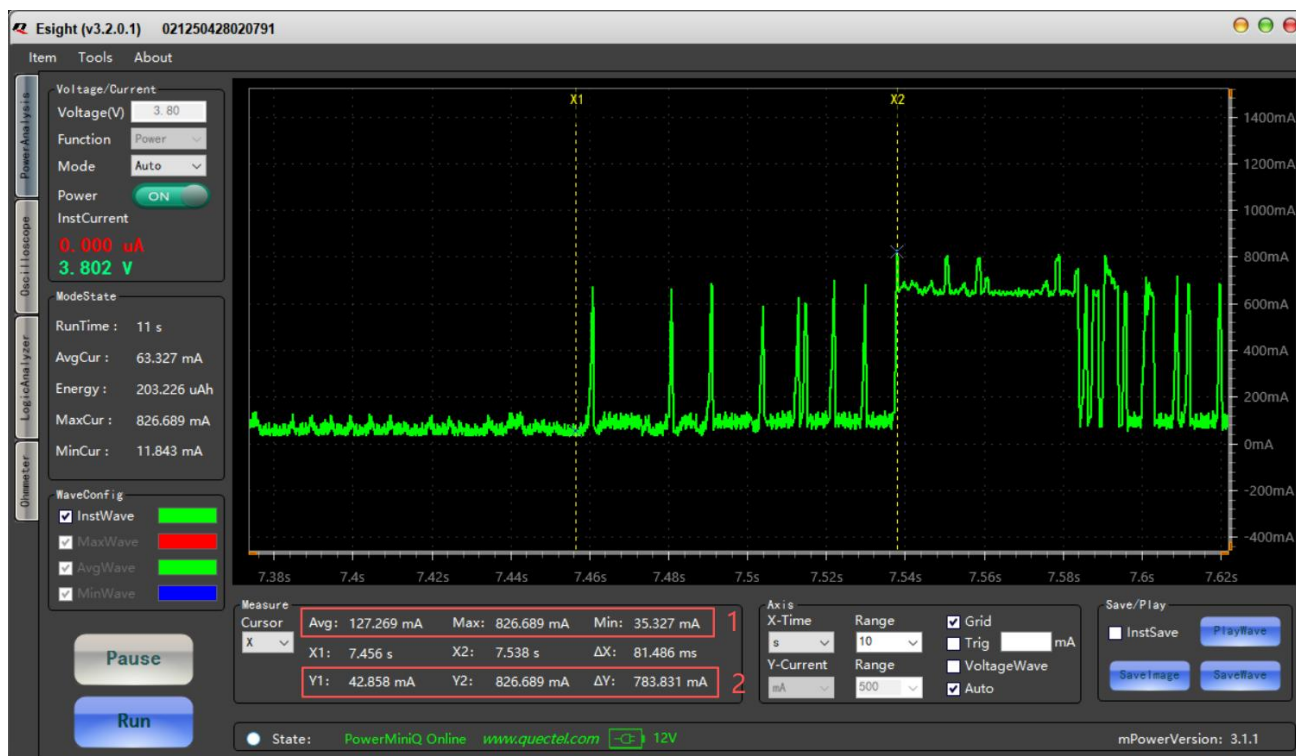


- 关于限流保护：电源模式下，当遇到超过限流阈值的电流时，设备默认触发过流提示并自动切断电源；也可通过设置限流值，在负载电流达到设定值后，实现自动断电，确保负载不超过额定电流。

#### 4.2.2 波形捕获及分析

**捕获：**下图以“Auto”模式为例。选择模式后，将“Power”设置为“ON”，点击“Run”即可实时捕获波形。点击“Pause”停止捕获。

**标尺：**选择“X”标尺后，X2 标尺将自动定位到屏幕显示区域的最大值。将鼠标光标移至标尺线或横轴 X1、X2 点位处，按住鼠标左键即可左右拖动标尺，松开鼠标即可停止拖动。

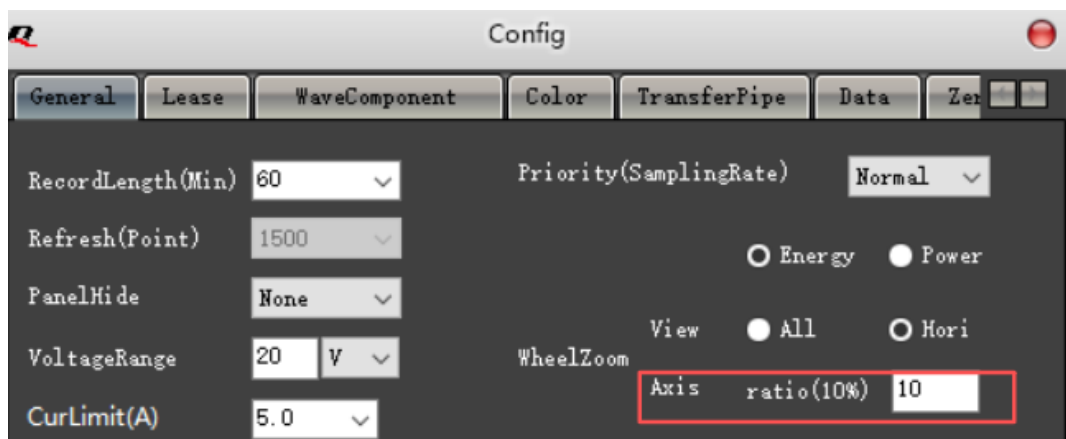


- 标签 1: 指 X1 和 X2 标尺区间内波形的平均、最大及最小电流。
- 标签 2: Y1、Y2 为横轴 X1、X2 点位对应的电流（若采用 Y 标尺，则显示 Y 标尺对应的电流）。

**缩放:** 波形区域内默认通过鼠标滚轮实现水平缩放，按住“**Shift**”键配合鼠标滚轮也可进行水平缩放；按住“**Ctrl**”键配合鼠标滚轮为垂直缩放；同时按住“**Shift**”和“**Ctrl**”键配合鼠标滚轮，可同时进行水平与垂直缩放。鼠标左键拖动绘制矩形框，可对选定区域放大。

**平移:** 按住鼠标滚轮，可沿任意方向平移波形；在 X 轴上按住鼠标左键或使用鼠标滚轮，可左右平移 X 轴；在 Y 轴上按住鼠标左键或使用鼠标滚轮，可上下平移 Y 轴。

坐标轴的平移力度可通过“**Tools**”->“**Config**”中“**Axis ratio**”设置，默认为“**10%**”。



**触发：**勾选“**Trig**”后，可通过拖动触发线或手动输入数值调整其位置；当捕获电流达到设定触发值时，系统将自动停止捕获。



**存储：**点击“**SaveWave**”可保存当前波形；勾选“**InstSave**”，可实现长时间保存全部波形。

将鼠标移至坐标轴右下角区域，将自动弹出已保存波形对话框，双击对应波形即可回放。



**双波形同时回放：**点击“Play Waveform”，弹出文件选择窗口，同时选中两个波形文件并点击“Open”即可。可通过“Left Offset” / “Right Offset” 设置双波形的 X 轴偏移，按回车键生效。



**校零：**若设备开机后，底电流未在 0 附近波动，可点击“Tools” -> “Zero Calibration” 进行手动校零。

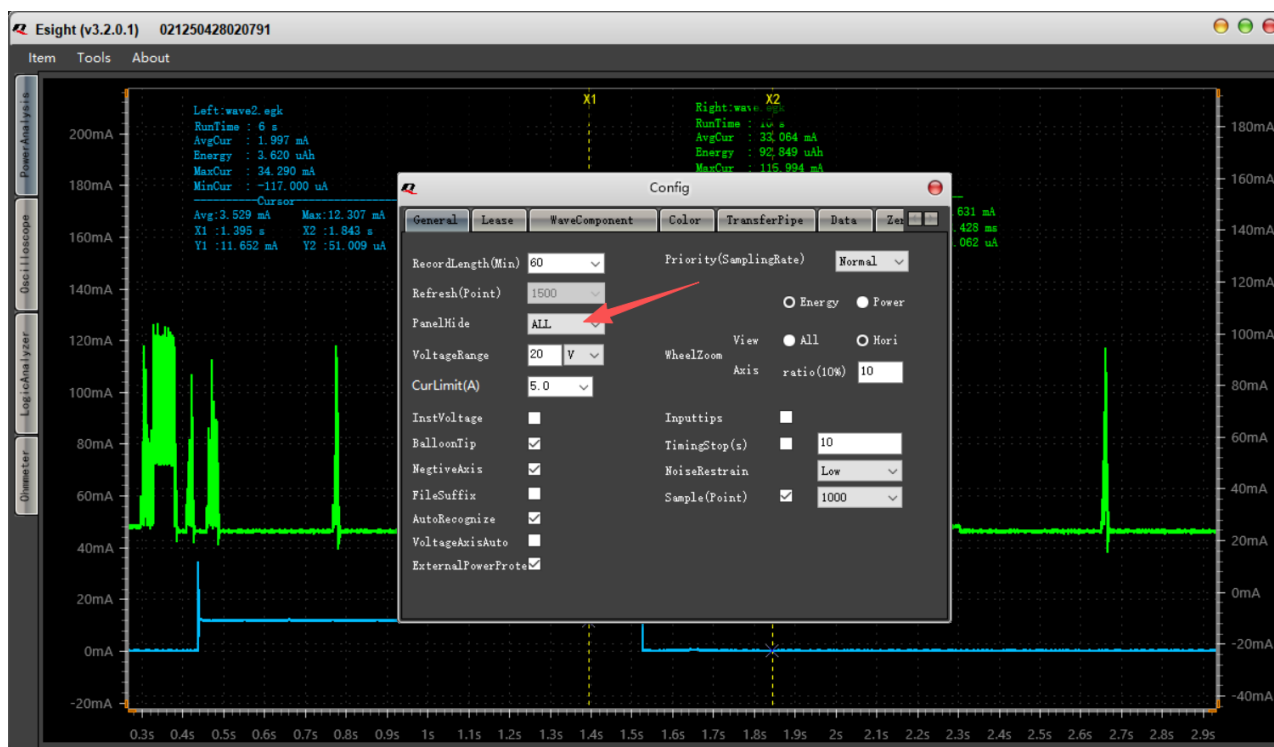
### 4.2.3 其他自定义配置

为满足不同测试场景的个性化需求，Esight 工具提供了丰富的自定义配置选项。用户可通过点击菜单栏“Tools” → “Config”，打开参数配置面板，进行以下参数的精细化调整：

#### 1. 面板隐藏

将“PanelHide”项配置为“ALL”，隐藏非必要设置面板，最大化扩展波形显示区域，提升低功耗测试数据观测体验。





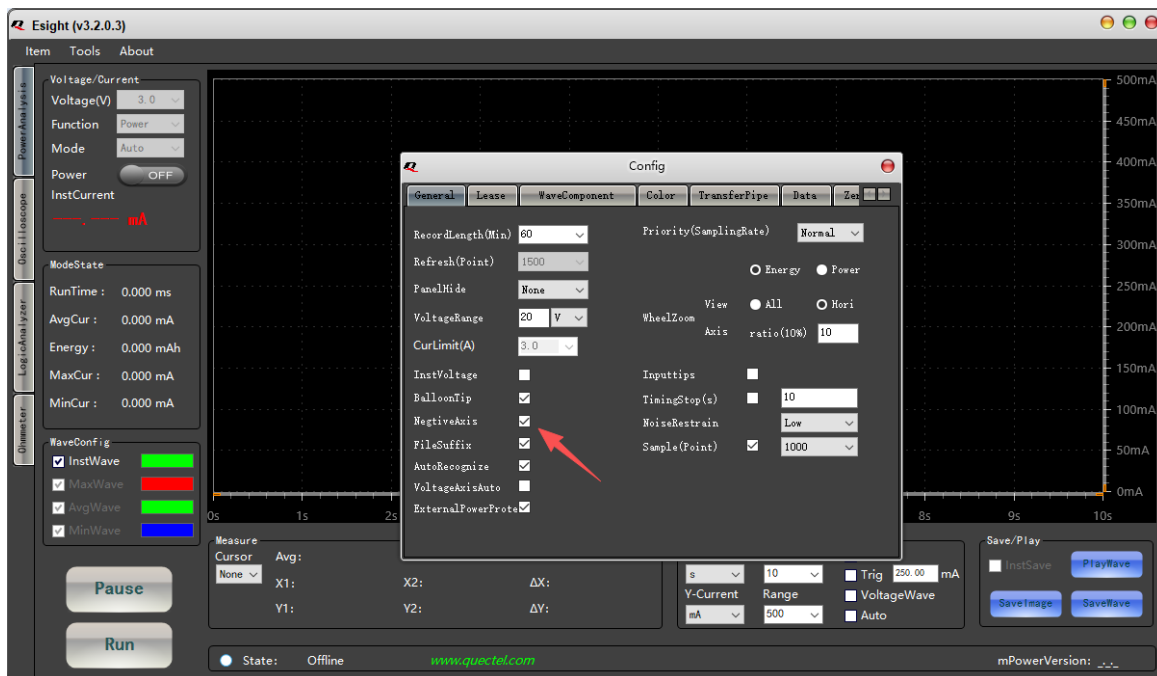
## 2. 电压/电流双波形显示

勾选“**VoltageWave**”，软件将同步采集并展示电压、电流双路波形，便于用户联合分析设备供电状态与功耗变化。



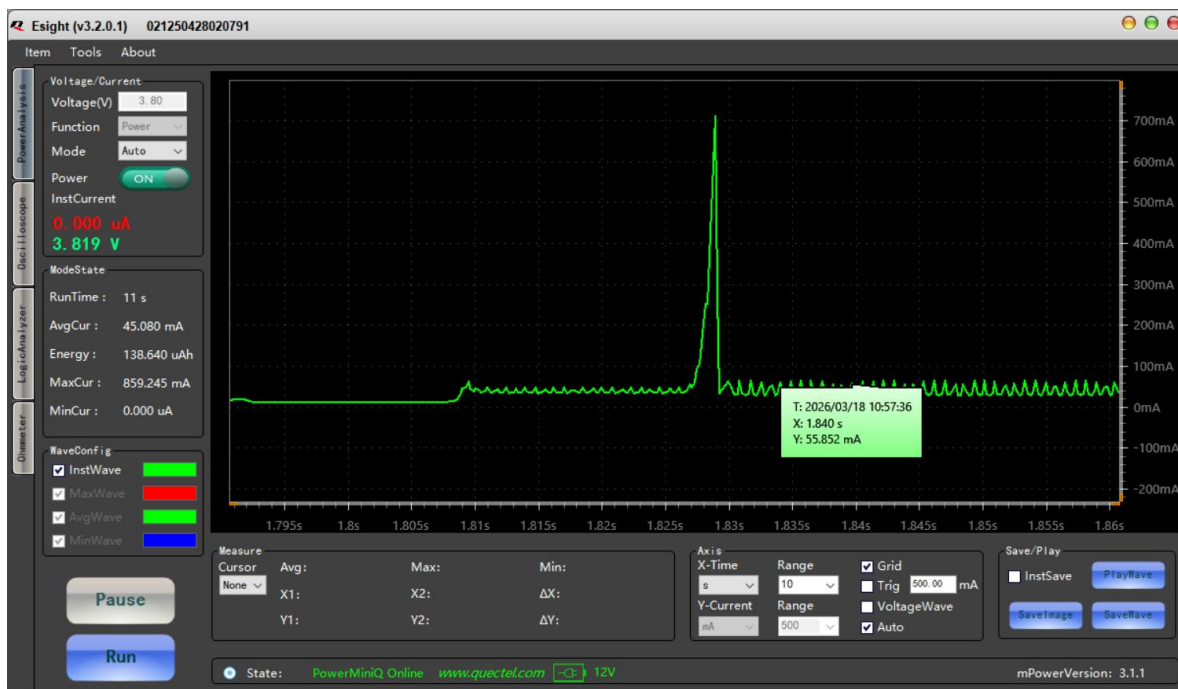
### 3. 负坐标显示

受容性负载特性影响，测试过程中可能会出现负电流，因此软件默认开启负坐标显示；若无需观测负电流，可取消勾选“**NegativeAxis**”，关闭负坐标显示。



### 4. 时间戳

单击波形任意位置，可显示该测试点的实时北京时间戳及对应的电流等详细信息。此外，波形存储时会同步保存时间戳，确保在后续回放数据时，能够精准还原所有测试点的详细时间信息。左键单击空白处即可关闭显示。



## 5. 采集优先级配置

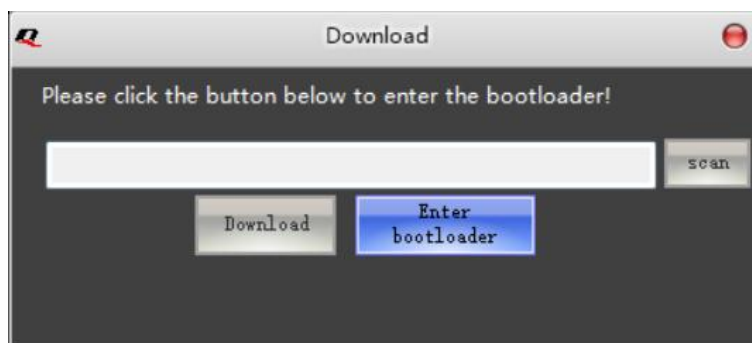
设备采样率当前仅支持“Normal”模式，默认为 5 ksps。软件默认显示平均电量，支持切换配置为平均功率展示。

## 4.3 系统升级指导

为保证设备功能完整性与稳定性，Quectel Power Mini 支持通过 Esight 工具进行固件在线升级，更新最新功能与优化项。升级操作流程如下：

步骤一：打开 Esight 工具，点击菜单栏“Tools”→“Download”，弹出固件下载对话框；

步骤二：点击“Enter bootloader”进入下载模式，设备将自动关机，保持下载窗口不关闭；

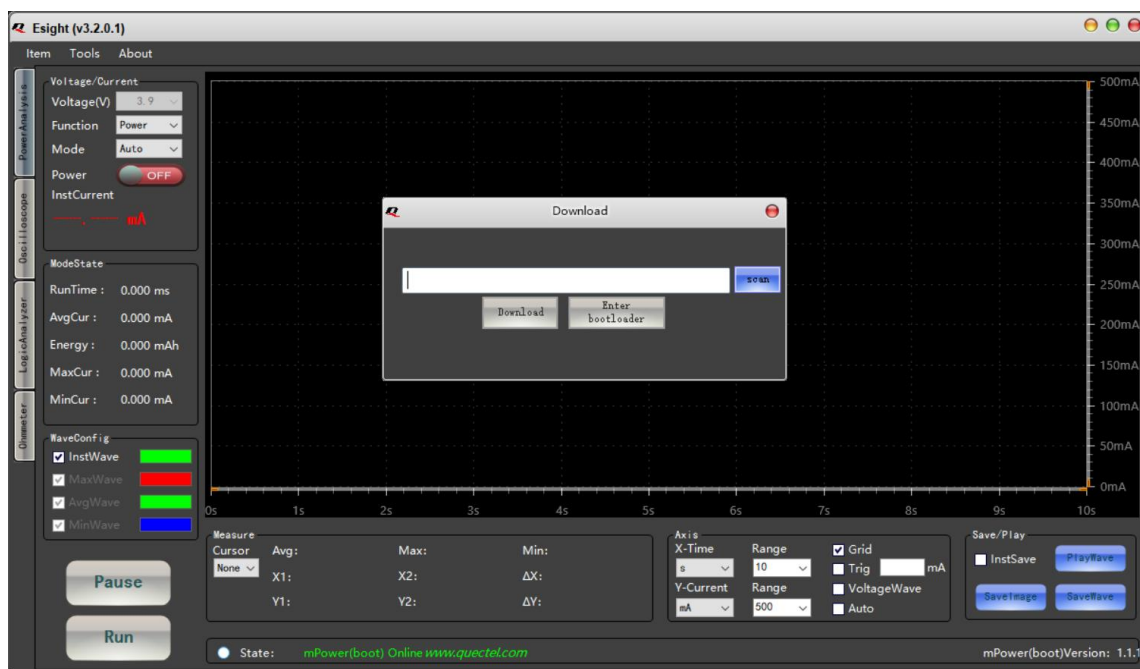


步骤三：重新按下设备电源键开机，软件左下角提示设备进入 Boot 模式；



步骤四：点击“Scan”，选择本地对应版本的固件文件；

步骤五：点击“Download”按钮开始固件升级，升级过程请勿断开 USB 连接；



步骤六：固件下载完成后，设备将自动关机，升级流程完成。

## 5 应用开发指导

为满足自动化测试及产线集成需求，Quectel Power Mini 提供了完善的二次开发支持。第三方可直接调用 Windows DLL 库提供的 API 接口来控制设备。理论上，系统支持多达 128 路 Quectel Power Mini 设备同时采集数据。本章节介绍核心 API 接口的部分说明及 Python 调用示例，详细的接口定义请参考 SDK 包内 *API 接口说明.txt* 文档。

### 备注

用户可访问 [https://developer.quectel.com/doc/files/quectel\\_power\\_mini/sources.zip](https://developer.quectel.com/doc/files/quectel_power_mini/sources.zip) 或扫描设备包装上的二维码下载所有配套的开发资料。

### 5.1 函数概览

表 3：函数概览

函数	说明
<i>Init()</i>	初始化 mPower1203 动态库
<i>OpenPort()</i>	打开一路 mPower1203 设备
<i>ClosePort()</i>	关闭一路 mPower1203 设备
<i>SetType()</i>	配置设备为电源模式或电流表模式
<i>SetMode()</i>	配置设备的工作精度或量程模式
<i>SetVoltage()</i>	电源模式下配置设备的输出电压
<i>SetPower()</i>	电源模式下打开或关闭设备电源
<i>StartSample()</i>	启动数据采集
<i>StopSample()</i>	停止数据采集
<i>GetCurrent()</i>	获取电流连续采样数据
<i>GetFwVersion()</i>	获取 mPower1203 设备固件版本号
<i>GetDllVersion()</i>	获取 mPower1203.dll 版本号

## 5.2 函数详解

### 5.2.1 Init

该函数用于初始化 mPower1203 动态库。

- 函数原型

```
int Init()
```

- 参数

无

- 返回值

0	函数执行成功
其他值	函数执行失败

### 5.2.2 OpenPort

该函数用于打开一路 mPower1203 设备。

- 函数原型

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

- 参数

*port*:

[In] mPower1203 设备端口号。

#### 备注

设备使用 USB 连接 PC 后可以在 PC 的设备管理器中查看对应设备的端口号。

- 返回值

设备 ID（正整数）	函数执行成功
小于 0	函数执行失败

### 5.2.3 ClosePort

该函数用于关闭一路 mPower1203 设备。

- 函数原型

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

- 参数

*id*:

[In] 设备 ID。

- 返回值

0                  函数执行成功

其他值            函数执行失败

### 5.2.4 SetType

该函数用于配置设备为电源模式或电流表模式。

- 函数原型

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

- 参数

*id*:

[In] 设备 ID。

*type*:

[In] 设备工作模式：详见第 5.2.4.1 章。

- 返回值

0                  函数执行成功

其他值            函数执行失败

#### 5.2.4.1 MPOWER\_TYPE

设备工作模式的枚举定义如下：

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

- 成员

成员	描述
<i>MPOWER_POWER</i>	电源模式：设备对外输出电压，为被测物供电
<i>MPOWER_AMPERE</i>	电流表模式：设备仅测量电流，不对外供电

#### 5.2.5 SetMode

该函数用于配置设备的工作精度或量程模式。

- 函数原型

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

- 参数

*id*:

[In] 设备 ID。

*mode*:

[In] 设备工作精度或量程模式；详情请见第 5.2.5.1 章和第 5.2.5.2 章。

- 返回值

0            函数执行成功

其他值      函数执行失败

### 5.2.5.1 MPOWER\_POWER\_MODE

设备在电源模式下的工作精度枚举定义如下：

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

#### ● 成员

成员	描述
<i>MPOWER_POWER_AUTO</i>	自动模式
<i>MPOWER_POWER_NORMAL</i>	普通模式
<i>MPOWER_POWER_PERCISION</i>	精密模式
<i>MPOWER_POWER_ULTRA</i>	超高精度模式

### 5.2.5.2 MPOWER\_AMPERE\_MODE

设备在电流表模式下的量程枚举定义如下：

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

#### ● 成员

成员	描述
<i>MPOWER_AMPERE_AUTO</i>	自动模式
<i>MPOWER_AMPERE_NORMAL</i>	普通模式
<i>MPOWER_AMPERE_PERCISION</i>	精密模式

## 5.2.6 SetVoltage

该函数用于在电源模式下配置设备的输出电压。

### ● 函数原型

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

### ● 参数

*id*:

[In] 设备 ID。

*voltage*:

[In] 设备的输出电压；范围：0.6~6.05；单位：V；详见第5.2.6.1章。

### ● 返回值

0 函数执行成功

其他值 函数执行失败

### 5.2.6.1 MPOWER\_VOLTAGE\_E

设备输出电压的枚举定义如下：

```
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;
```

#### ● 成员

成员	描述
V0_6	0.6 V
V0_65	0.65 V
V0_7	0.7 V
V0_75	0.75 V
...	...（以 0.05V 为步长，依此类推）
V6_0	6 V
V6_05	6.05 V

### 5.2.7 SetPower

该函数用于在电源模式下打开或关闭设备电源。

#### ● 函数原型

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

#### ● 参数

*id*:

[In] 设备 ID。

*on*:

[In] 设备电源开关操作；详见第 5.2.7.1 章。

#### ● 返回值

0 函数执行成功

其他值 函数执行失败

### 5.2.7.1 MPOWER\_POWER\_SUPPLY

设备电源开关操作的枚举定义如下：

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

- 成员

成员	描述
<i>MPOWER_POWER_OFF</i>	关闭电源
<i>MPOWER_POWER_ON</i>	打开电源

### 5.2.8 StartSample

该函数用于启动数据采集。

- 函数原型

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

- 参数

*id*:  
[In] 设备 ID。

- 返回值

0                函数执行成功  
其他值        函数执行失败

### 5.2.9 StopSample

该函数用于停止数据采集。

- 函数原型

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

- 参数

*id:*

[In] 设备 ID。

- 返回值

0 函数执行成功

其他值 函数执行失败

## 5.2.10 GetCurrent

该函数用于获取电流连续采样数据。

- 函数原型

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

- 参数

*id:*

[In] 设备 ID。

- 返回值

电流数据数组，长度不固定，但一定是偶数，数据输出格式如下：

[0]:时间戳(us) [1]:电流(uA) [2]:时间戳 [3]:电流(uA) ... 依此类推。

### 备注

因为数据采集速率非常快，建议单独线程采集数据，且尽量不要有耗时操作比如延时函数或者 log 输出等。如果获取到相同时间戳的数据，只保留一个即可。

## 5.2.11 GetFwVersion

该函数用于获取 mPower1203 设备固件版本号。

- 函数原型

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

- 参数

*id:*

[In] 设备 ID。

- 返回值

固件版本号字符串，格式：[0]主版本号 [1]次版本号 [2]修订号  
-1

函数执行成功  
函数执行失败

### 5.2.12 GetDllVersion

该函数用于获取 mPower1203.dll 版本号。

- 函数原型

```
char* GetDllVersion()
```

- 参数

无

- 返回值

DLL 库版本号字符串，格式：[0]主版本号 [1]次版本号 [2]修订号 [3]构建号  
-1

函数执行成功  
函数执行失败

## 5.3 Python 示例代码

以下代码演示了如何通过 Python 调用 DLL 接口，同时控制两台设备进行数据采集：

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

# 自定义采集时长，单位：秒。
SAMPLE_TIME = 5
# 两路功耗仪设备 COM 口，根据实际端口号调整。
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):
    """将秒数转换为 xh ym zs 格式"""
    hours = int(seconds // 3600)
    remaining_seconds = seconds % 3600
    minutes = int(remaining_seconds // 60)
    seconds = int(remaining_seconds % 60)

    # 拼接非零部分
    parts = []
    if hours > 0:
        parts.append(f'{hours}h')
    if minutes > 0:
        parts.append(f'{minutes}m')
    if seconds > 0 or not parts: # 确保至少显示秒（即使都为 0）
        parts.append(f'{seconds}s')

    return ' '.join(parts)

def print_t(*args, **kwargs):
    """
    带日期时间戳的打印函数，功能与 print 一致，日志前自动添加[YYYY-MM-DD HH:MM:SS]前缀
    """
    # 获取当前时间戳（精确到秒），格式化为 [年-月-日 时:分:秒]
    timestamp = datetime.now().strftime("[%Y-%m-%d %H:%M:%S]")

    # 将时间戳作为第一个参数，拼接用户传入的所有参数
    # 可以使用 builtins.print 调用原始 print 函数，避免递归调用
    print(timestamp, *args, **kwargs)

# 导入 mPower1203 动态库，若新版本 FindAssembly 方法不生效，请更换为 AddReference 方法，

```

```

不加扩展名.dll
clr.AddReference('mPower1203')
from mPower1203 import *
mPower = ClassmPower()

# 初始化动态库
Res = mPower.Init()
print_t(f"init:{Res}")

# 打开第一路设备
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}")

# 配置第一路设备为电源模式
mPower.SetType(Ch1,MPOWER_TYPE.POWER)

# 设置输出电压为 3 V
mPower.SetVoltage(Ch1,MPOWER_VOLTAGE.V3_0)

# 打开第二路设备
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}")

# 配置第二路设备为电源模式
mPower.SetType(Ch2,MPOWER_TYPE.POWER)

# 设置输出电压为 4 V
mPower.SetVoltage(Ch2,MPOWER_VOLTAGE.V4_0)

# 打开电源输出开关
mPower.SetPower(Ch1,MPOWER_POWER.ON)
mPower.SetPower(Ch2,MPOWER_POWER.ON)
time.sleep(1)

```

```
# 启动数据采样
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"----- 第 {test_cnt} 次测试, 运行时长: {format_run_time(time.perf_counter() -
start_time)}-----")

# 停止数据采样
Res = mPower.StopSample(Ch1)
Res = mPower.StopSample(Ch2)
time.sleep(1)

# 关闭通道
Res = mPower.ClosePort(Ch1)
print_t(f"CloseChannel:{Ch1}")
Res = mPower.ClosePort(Ch2)
print_t(f"CloseChannel:{Ch2}")
```



## 6 附录 术语缩写

表 4：术语缩写

缩写	英文全称	中文全称
API	Application Programming Interface	应用程序编程接口
DC	Direct Current	直流电
LPWA	Low-Power Wide-Area Network	低功耗广域网络
PC	Personal Computer	个人电脑
USB	Universal Serial Bus	通用串行总线