



Esight & mPower1203U 用户手册

合肥晓鹰智能科技有限公司

Web: www.egteks.cn

Ver: 1.1

Date: 2025-12-05



关注公众号获取更多信息

版本记录

版本	日期	作者	描述
1.0	2025-01-10	Egteks	初版
1.1	2025-12-05	Egteks	1. 更新电流表接线方式 2. 更新一些工具小功能使用 tips 3. 论坛网址更新

目录

版本记录	1
1 简介	2
2 Windows 驱动安装	3
2.1. Windows10(免安装)	3
2.2. Windows7/8 安装步骤	3
2.3. NetFramework 安装(可选)	3
3 mPower1203U 接线说明	4
3.1. 电源模式	4
3.2. 电流表模式	4
3.3. mPower1203U 供电方式	5
4 Esight 工具操作	6
4.1. 设备连接 PC	6
4.2. 波形捕获和分析	6
4.3. 其他配置	10
4.4. 固件升级	17
4.5. 更多功能	18

1 简介

Esight 工具是用于 **mPower** 系列产品的上位机软件,支持 windows7 及其以上版本系统, 支持 x86(32 位)和 x64(64 位)系统, 二次开发 SDK 可以支持 windows/Linux(也支持 ARM 平台)/Mac os。 **mPower1203** 是一款集便携式直流程控供电、高精度电流测试、动态电流波形显示分析以及高性价比为一体的直流分析电源, 可以很好的在您进行功耗测试分析时提供动态测量分析和便携服务!

市场痛点:

- 手持设备/穿戴设备对低功耗的需求越来越高, 普通的大功率直流电源虽价格便宜, 但电流测试精度不能满足需求。
- 普通的大功率直流电源不能提供电流波形的显示, 保存, 回放以及分析功能。
- 专业的直流电源分析仪非常昂贵, 价格都是几万甚至几十万。而且比较笨重, 不方便高频率的搬运。很难做到户外无 220V 交流电环境下的可调供电以及电流测试。

我们能解决什么?

- **mPower** 系列产品体积只有常规智能手机大小, 供电容易, 携带和使用都方便。
- **mPower** 系列产品可支持 nA 级~A 级的动态电流测量范围, 测试精度可高达 30nA, 采样率可高达 30ksps, 可准确的反映电流动态情况, 并配合 **Esight** 工具进行电流波形的显示分析、存储和回放。
- **mPower** 系列产品设备可采用标配的适配器供电, 也可采用通用的手机充电器供电使用。户外情况可采用充电宝等直流储能设备供电使用。

2 Windows 驱动安装

2.1. Windows10 及以上系统(免安装)

无需安装，设备开机会自动安装驱动。

2.2. Windows7/8 安装步骤

驱动下载链接：

链接：<https://pan.baidu.com/s/1qsk-Lxt7TOiuZU9ow7B7wQ>

提取码：a0h0

解压 EgteksUsbDriver.rar

Win7 点击文件夹 Win7 中 w7_install.bat 安装即可。

Win8 点击文件夹 Win8 中 w8_install.bat 安装即可。

2.3. NetFramework 安装(可选)

默认绝大部分电脑都已经安装过 NetFramework.

Esight 工具免安装,如果双击打开工具提示电脑没有.net Framework 则需要手动安装 Microsoft.NET.exe.

下载链接：

链接：<https://pan.baidu.com/s/1qsk-Lxt7TOiuZU9ow7B7wQ>

提取码：a0h0

双击默认安装即可。

3 mPower1203U 接线说明

3.1. 电源模式

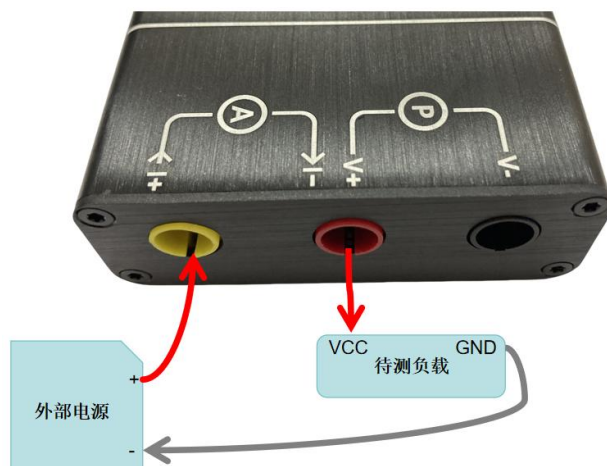
连接示意图如下：



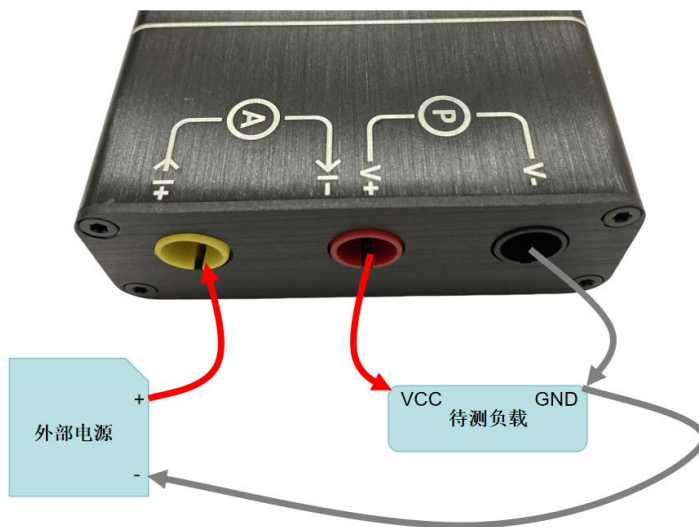
给负载提供调 0.6V~13.5V 的供电电压。
能够实时监测显示负载电流波形。

3.2. 电流表模式

不共地连接示意图如下：



共地连接示意图如下：



注意事项：

1. 电流表模式实时监测显示所串入电路的电流波形
2. 通常只需要使用**不共地**接法，电流通常从 I+ 流入, I- 流出，测量的是正电流，也可以反向接线则测量的是负电流，外部供电**不能超过 60V**。
3. 如果在电流表模式需要**同时测量电压**可以使用**共地**接法，将功耗仪的 V- 和待测负载地连接到一起，这种接法，外部供电**不能超过 13.5V**。

3.3. mPower1203U 供电方式

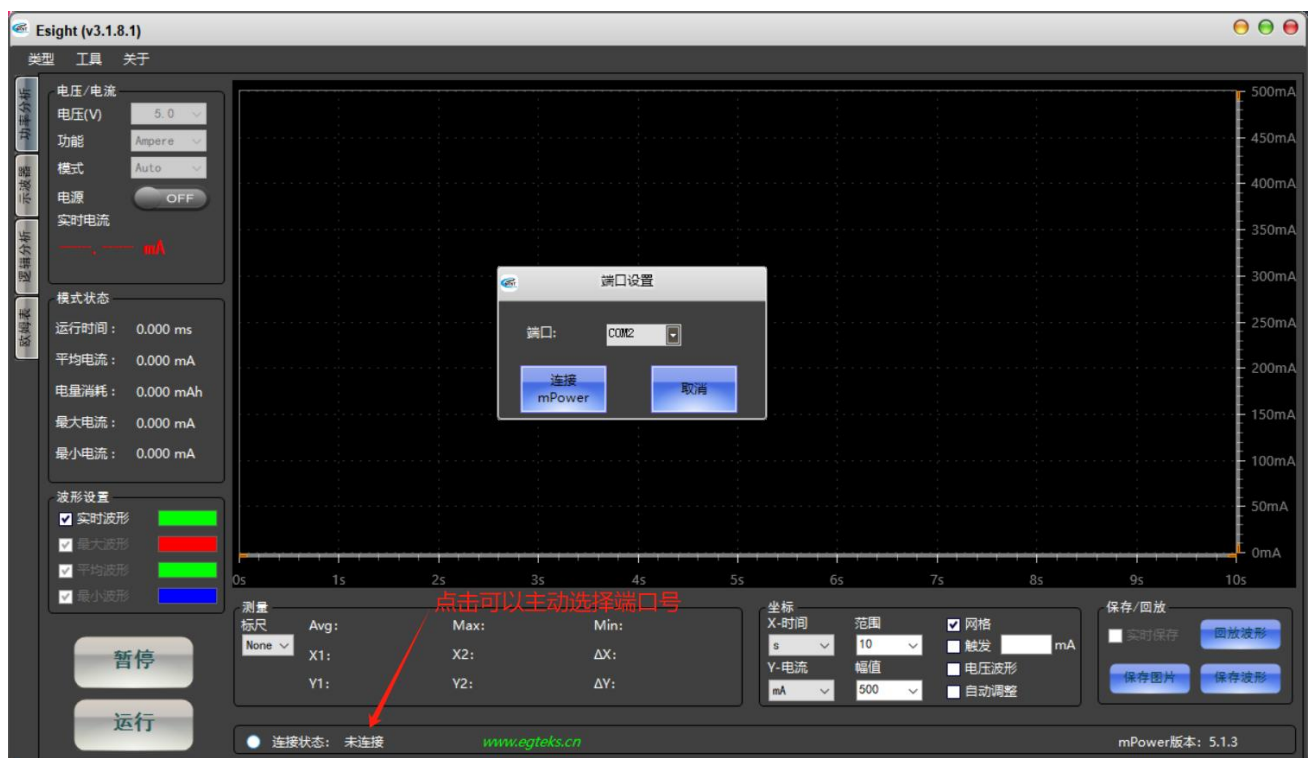
1. 有市电插座场景，推荐使用标配的 12V 配器供电。
2. 通过手机充电适配器和 Type-C 线也可以给 mPower 供电。
3. 外场无市电场景，可以使用充电宝配合 Type-C 线给 mPower 供电。

注意：mPower 的 Type-A 接口只能用于和 PC 通信，无法供电。

4 Esight 工具操作

4.1. 设备连接 PC

1. 工具默认是英文，可以通过【Tools】-【Language】菜单选择【Chinese】切换到中文界面
2. 默认 mPower 开机自动连接 Esight 工具。
3. 如果没有自动连接，可以点击左下角【未连接】或者点击菜单【工具】->【端口】选择端口连接。
注意：如果电脑有蓝牙端口，可能会导致工具卡顿无法自动找到 mPower 设备，稍等一会，手动连接一次即可，后面工具会记住这个端口下次再连接就不会卡顿了。



4.2. 波形捕获和分析

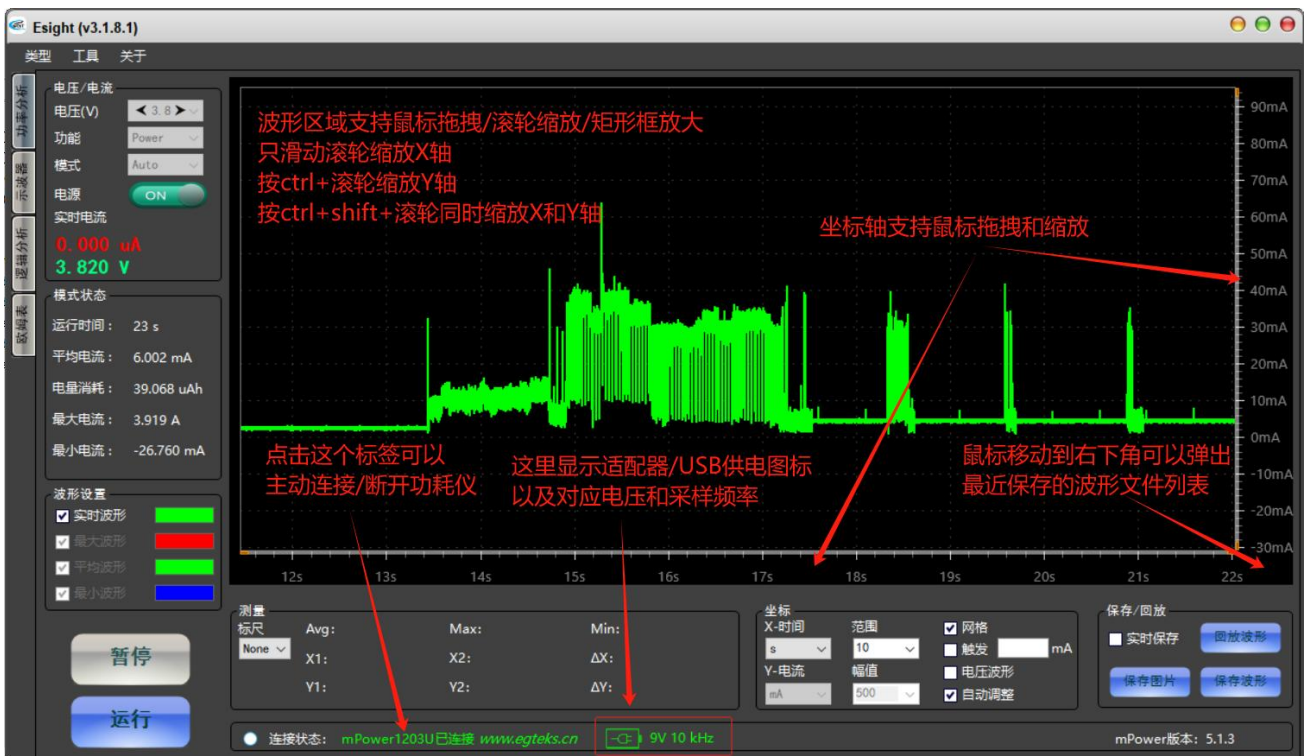
捕获：下面以电源模式 auto 档位为例,选择好模式后将【电源】按钮置为 ON 状态，点击【运行】按钮即可实时捕获波形。点击【停止】按钮停止捕获。点击屏幕左边“平均电流”标签可以切换到均方电流显示(用于电流表模式测试交流信号时查看),如果开启了实时电压功能，再次点击“平均电流”标签可以切换到平均电压显示。点击实时波形任意点，可以查看北京时间戳和对应点的电流信息，右击屏幕任意点取消显示。更多功能可以参考 <https://bbs.egteks.cn>

标尺：选择 X 标尺,X2 标尺会自动定位到屏幕显示区域最大值。将鼠标放在标尺线或者标签上按住鼠

标左键即可左右拖动标尺，释放鼠标即可停止。



- ①X1 和 X2 标尺区域之间波形的平均/最大/最小电流。点击 Avg 后面的数值可以切换电流和电量显示。
 ②Y1 和 Y2 是 X1 和 X2 点对应的电流（如果有 Y 标尺则是 Y 标尺对应的电流）。



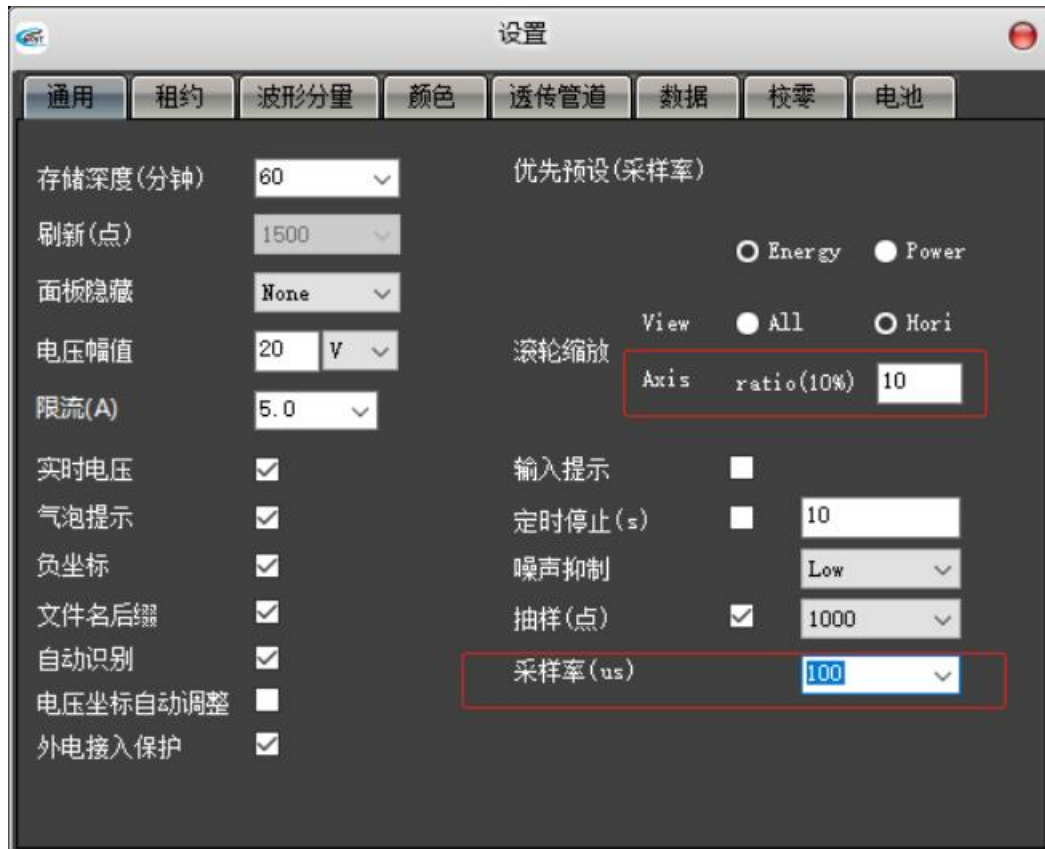
缩放：鼠标滚轮在波形区默认是水平缩放，按住 shift 按键配合鼠标滚轮也是水平缩放，按住 ctrl 按键

配合鼠标滚轮是垂直缩放，同时按住 **shift** 和 **ctrl** 配合滚轮则同时水平垂直缩放。鼠标左键拖动一个矩形框可以实现矩形区域放大。

平移：按住鼠标滚轮可以任意方向平移波形；在 X 坐标轴按住鼠标左键或者使用鼠标滚轮可以左右平移 X 轴；在 Y 坐标轴按住鼠标左键或者使用鼠标滚轮可以上下平移 Y 轴。

坐标轴的平移力度可以通过【工具】->【设置】中 **Axis ratio** 设置，默认为 10%。

采样率：按 mPower1203U 采样率可配置，范围[20KHz - 1KHz]，注意如果同时采集电压，最低采样率则是 10KHz；【采样率】输入框可以在[50us-1000us]任意设置，按回车生效。

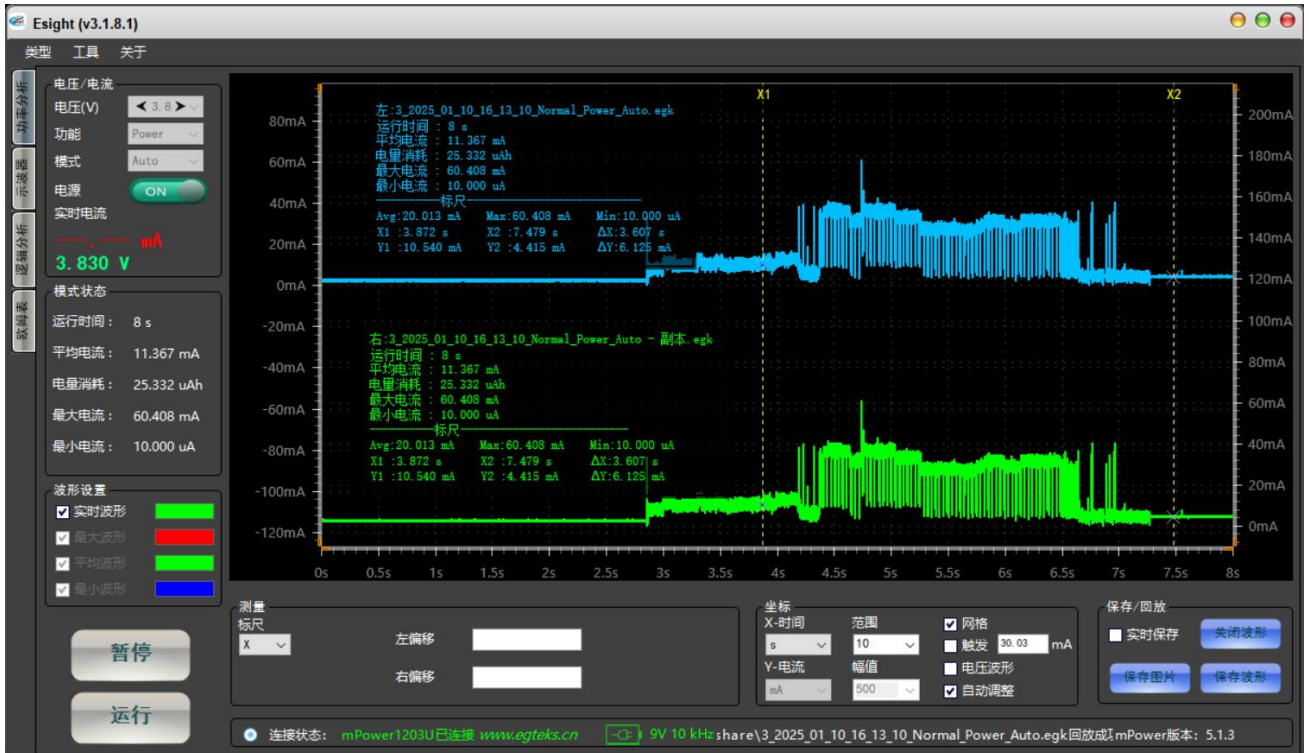


触发：勾选【触发】后通过移动触发线或者手动输入值调整触发线位置，当捕获电流达到触发值时自动停止捕获。



存储: 点击【保存波形】可以保存当前波形，也可以通过勾选【实时保存】实现长时间保存所有波形。鼠标移动到坐标轴右下角区域会自动弹出已保存波形对话框，双击即可回放波形。存储波形默认是 egk 文件，可以通过【工具】-【设置】-【数据】菜单功能转换为 txt 或者 csv 表格数据，csv 表格数据也可以导入到 Origin 数据分析软件中回放，参考 <https://bbs.egteks.cn/?/article/60>

双波形同时回放: 直接选中 2 个波形拖动到波形区域即可同时回放 2 个波形，或者点击【回放波形】弹出文件选择对话框，同时选择 2 个波形文件，点击打开即可。双波形的 X 轴偏移可以通过左偏移/右偏移配置，按回车键生效。



校零: 如果发现设备开机后底电流不是在 0 附近跳动或者对精度要求比较高可以点击【工具】->【校零】校零。

自动调整【自动调整】: 空载运行 5 秒, 点击【自动调整】按钮会自动更新数值到编辑框, 退出设置界面, 停止运行, 再次运行即可。

手动方式: 可以直接填入数值按回车生效。

注意: 因为每次电源 ON 校准值都有差异, 所以校准值在电源 OFF 后会自动清空。

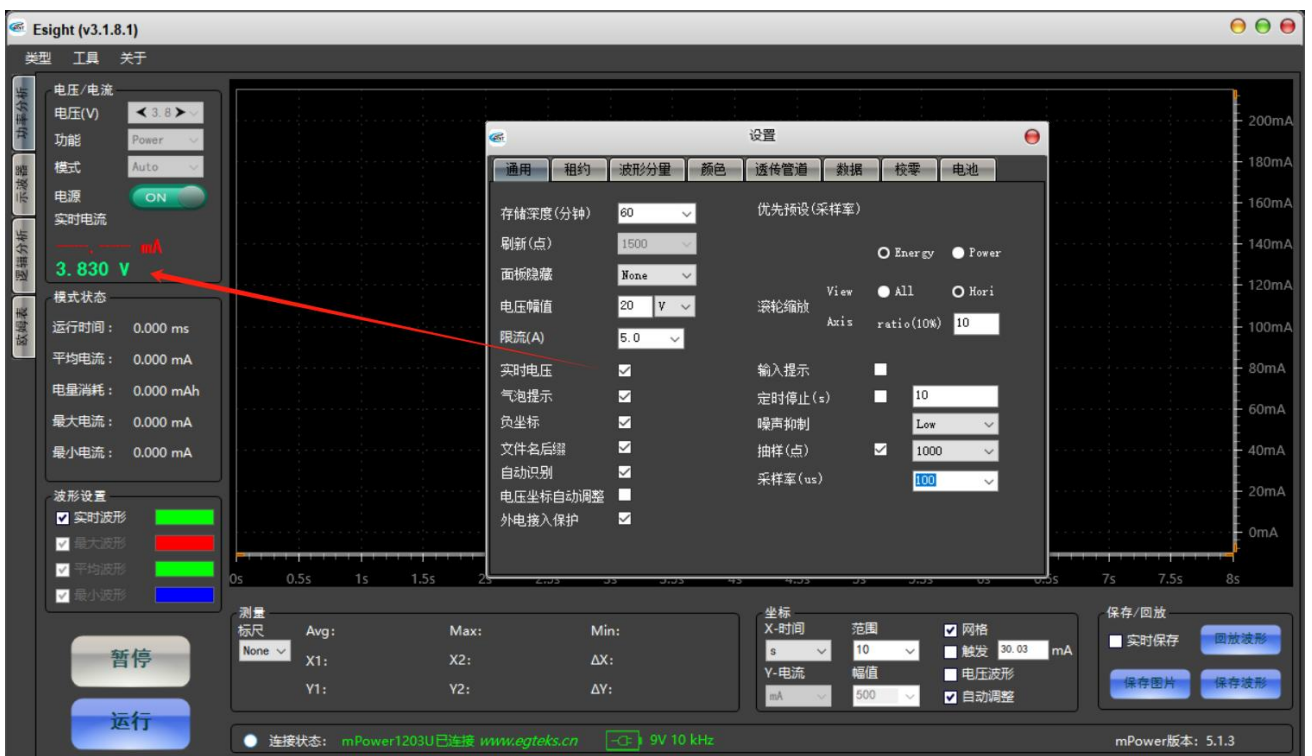
4.3. 其他配置

点击【工具】->【设置】打开配置选项。

面板隐藏: 为了最大化显示波形区域, 可以通过配置面板隐藏功能。



实时电压：勾选【实时电压】即可。



负坐标：因为容性负载的原因，可能会出现负电流，因此默认会显示负坐标，如果不关注负电流，可以去勾选【负坐标】取消负坐标的显示。

二次开发：第三程序也可以直接调用 SDK 中的 windows 的 dll 库或者 Linux/Mac OS 的 so 库提供的

API 接口控制 mPower1203 设备，SDK 中提供 python / C# / C++ / VB 等示例程序，理论上可以支持 128 路 mPower1203 设备同时采集数据,部分接口示例如下，更多接口 api 和示例请参考二次开发 SDK 包：

```

/*****
** 函数名:      Init
** 函数描述:    初始化 mPower1203 动态库
** 参数:        无
**
** 返回:        [out]结果码: 0-成功 其他-失败
*****/
int Init()

/*****
** 函数名:      OpenPort
** 函数描述:    打开一路 mPower1203 设备
** 参数:        [in]port:mPower1203 设备端口号(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:        [out]设备 id:通常是从 1 开始,依次累加,如果 ID 小于 0 表示打开失败
*****/
int OpenPort(string port)

/*****
** 函数名:      ClosePort
** 函数描述:    关闭一路 mPower1203 设备
** 参数:        [in]设备 id
**
** 返回:        [out]结果码: 0-成功 其他-失败
*****/
int ClosePort(int id)

/*****
** 函数名:      SetType
** 函数描述:    配置为电流表/电源类型
** 参数:        [in]设备 id
** 参数:        [in]type 取值参考 MPOWER_TYPE_E
**
** 返回:        [out]结果码: 0-成功 其他-失败
*****/
int SetType(int id, int type)

/*****
** 函数名:      SetMode
** 函数描述:    配置设备工作模式
** 参数:        [in]设备 id

```

** 参数: [in]mode 在电压表模式下取值参考 MPOWER_POWER_MODE_E,电流表模式下取值参考
MPOWER_AMPERE_MODE_E

**

** 返回: [out]结果码: 0-成功 其他-失败

*****/

int SetMode(int id, int mode)

/*****/

** 函数名: SetVoltage

** 函数描述: 电源模式下配置设备输出电压

** 参数: [in]设备 id

** 参数: [in]voltage 取值参考 MPOWER_VOLTAGE_E,支持 0.6V-13V 电压可调

**

** 返回: [out]结果码: 0-成功 其他-失败

*****/

int SetVoltage(int id, int voltage)

/*****/

** 函数名: SetPower

** 函数描述: 电源模式下打开/关闭设备电源

** 参数: [in]设备 id

** 参数: [in]on 取值参考 MPOWER_POWER_E

**

** 返回: [out]结果码: 0-成功 其他-失败

*****/

int SetPower(int id, int on)

/*****/

** 函数名: StartSample

** 函数描述: 启动数据采集

** 参数: [in]设备 id

**

** 返回: [out]结果码: 0-成功 其他-失败

*****/

int StartSample(int id)

/*****/

** 函数名: StopSample

** 函数描述: 停止数据采集

** 参数: [in]设备 id

**

** 返回: [out]结果码: 0-成功 其他-失败

*****/

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

```

/*****
** 函数名:      GetCurrent
** 函数描述:    获取电流采样数据，注意：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 输出等等，如果获取到相同时间戳的数
                据只保留一个即可
** 参数:        [in]设备 id
**
** 返回:        [out]电流数据数组，长度不固定，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) ... 依次类推
*****/
double[] GetCurrent(int id)

```

```

/*****
** 函数名:      GetFwVersion
** 函数描述:    获取 mPower1203 设备固件版本号
** 参数:        [in]设备 id
**
** 返回:        [out]版本号,[0]主版本号 [1]次版本号 [2]修订号
*****/
byte[] GetFwVersion(int id)

```

```

/*****
** 函数名:      GetDllVersion
** 函数描述:    获取 mPower1203.dll 版本号
** 参数:        无
**
** 返回:        [out]版本号,[0]主版本号 [1]次版本号 [2]修订号
*****/
byte[] GetDllVersion()

```

三台设备同时采集示例 python 程序如下：

```
#coding=utf-8
```

```
"""
```

本例程使用 python 控制 Egtks 公司的 mPower1203 设备,同时支持多路设备配置和电流数据采集
本例程演示同时采集 3 路 mPower1203 设备

本例程实测环境是 python v2.7.18

库文件 mPower1203.dll 提供的 API 接口说明参考 History 文件

```
"""
```

```

import os
import time
import clr

```



```

#常量数据定义
from collections import namedtuple
MPOWER_TYPE = namedtuple('MPOWER_TYPE','POWER AMPERE')._make(range(2))
MPOWER_POWER_MODE = namedtuple('MPOWER_POWER_MODE','AUTO NORMAL
PERCISION ULTRA')._make(range(4))
MPOWER_AMPERE_MODE = namedtuple('MPOWER_AMPERE_MODE','AUTO NORMAL
PERCISION')._make(range(3))
MPOWER_VOLTAGE = namedtuple('MPOWER_VOLTAGE','V0_6 V0_7 V0_8 V0_9 V1_0 V1_1
V1_2 V1_3 V1_4 V1_5 \
V1_6 V1_7 V1_8 V1_9 V2_0 V2_1 V2_2 V2_3 V2_4 V2_5 V2_6 V2_7 V2_8 V2_9 V3_0 V3_1 V3_2
V3_3 V3_4 V3_5 \
V3_6 V3_7 V3_8 V3_9 V4_0 V4_1 V4_2 V4_3 V4_4 V4_5 V4_6 V4_7 V4_8 V4_9 V5_0 V6_0 V7_0
V8_0 V9_0 V10 V11 V12 V13')._make(range(53))
MPOWER_POWER = namedtuple('MPOWER_POWER','OFF ON')._make(range(2))

#导入 mPower1203 动态库
clr.FindAssembly('mPower1203.dll')
from mPower1203 import *
mPower = ClassmPower()

#初始化动态库
Res = mPower.Init()
print Res

#配置第一路,COM 口需要根据实际端口号调整
Ch1 = mPower.OpenPort("COM5")
print Ch1
#配置为电源模式
Res = mPower.SetType(Ch1,MPOWER_TYPE.POWER)
print Res
#设置输出电压为 12v
Res = mPower.SetVoltage(Ch1,MPOWER_VOLTAGE.V12)
print Res
#打开电源输出开关
Res = mPower.SetPower(Ch1,MPOWER_POWER.ON)
print Res

#打开第二路,COM 口需要根据实际端口号调整
Ch2 = mPower.OpenPort("COM12")
print Ch2
#配置为电源模式

```

```

Res = mPower.SetType(Ch2,MPOWER_TYPE.POWER)
print Res
#设置输出电压为 12v
Res = mPower.SetVoltage(Ch2,MPOWER_VOLTAGE.V12)
print Res
#打开电源输出开关
Res = mPower.SetPower(Ch2,MPOWER_POWER.ON)
print Res

```

```

#打开第三路,COM 口需要根据实际端口号调整
Ch3 = mPower.OpenPort("COM13")
print Ch3
#配置为电源模式
Res = mPower.SetType(Ch2,MPOWER_TYPE.POWER)
print Res
#设置输出电压为 12v
Res = mPower.SetVoltage(Ch2,MPOWER_VOLTAGE.V12)
print Res
#打开电源输出开关
Res = mPower.SetPower(Ch2,MPOWER_POWER.ON)
print Res

```

```

#启动数据采样
Res = mPower.StartSample(Ch1)
print Res
time.sleep(1)
Res = mPower.StartSample(Ch2)
print Res
time.sleep(1)
Res = mPower.StartSample(Ch3)
print Res
time.sleep(1)

```

```
'''
```

说明:

- 1.这里为了演示采集数据所以使用 **log** 输出函数,正式使用场景下数据采集线程请勿输出 **log**,避免丢数据
- 2.因为 PC 采集 API 调用速度可能会比实际设备采集数据速度快,所以可能会出现接收数据中存在多个时间戳完全一样的数据,后续数据处理时丢弃掉重复的数据只保留一个即可

```
'''
```

```

while True:
    Data1 = mPower.GetCurrent(Ch1)

```

```

Data2 = mPower.GetCurrent(Ch2)
Data3 = mPower.GetCurrent(Ch3)

#输出 3 个通道每次接收的数据个数,
#print('%d,%d,%d'%(len(Data1),len(Data2),len(Data3)))

#输出 3 个通道每次接收的第一个数据的时间戳(us)和电流(uA)
print('%d,%.3f,%d,%.3f,%d,%.3f'%(Data1[0],Data1[1],Data2[0],Data2[1],Data3[0],Data3[1]))

#停止数据采样
Res = mPower.StopSample(Ch1)
print Res
Res = mPower.StopSample(Ch2)
print Res
Res = mPower.StopSample(Ch3)
print Res

#关闭通道
Res = mPower.ClosePort(Ch1)
print Res
Res = mPower.ClosePort(Ch2)
print Res
Res = mPower.ClosePort(Ch3)
print Res

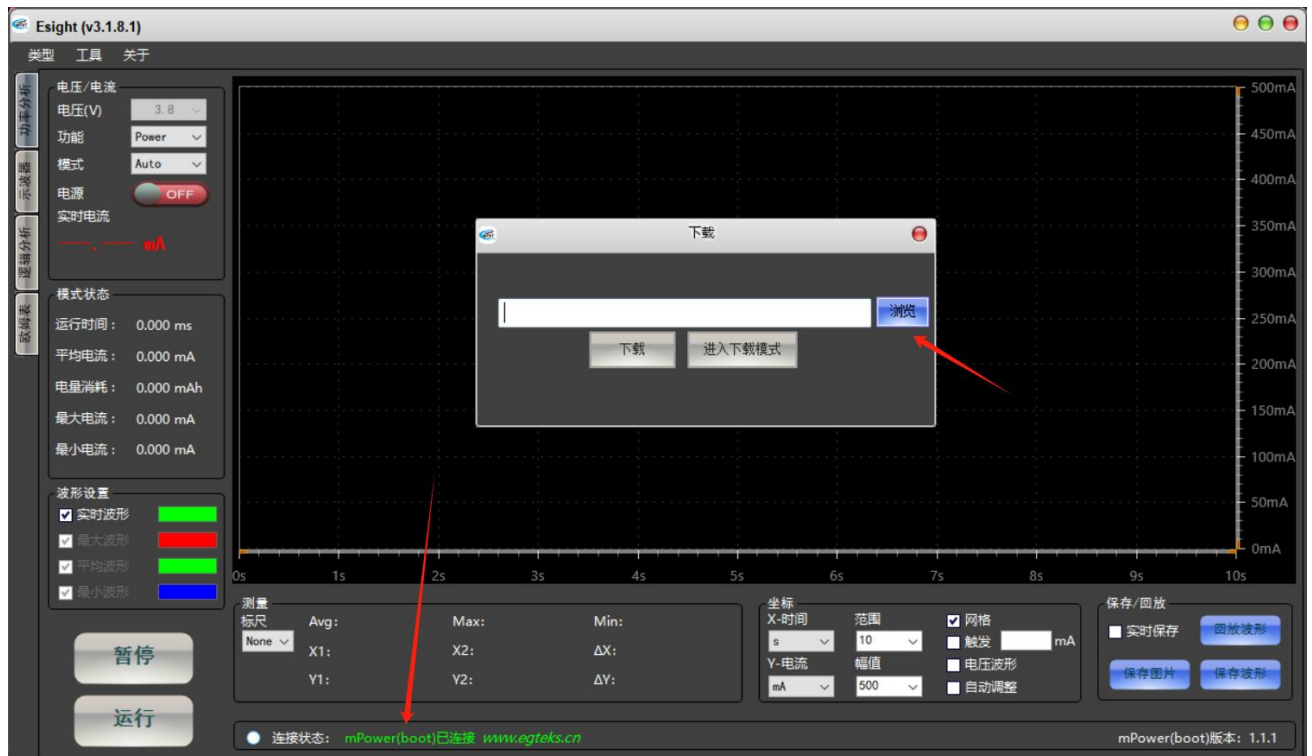
```

4.4. 固件升级

点击【工具】->【下载】打开下载对话框，点击【进入下载模式】进入下载模式，设备会自动关机，下载窗口不用关闭。



再次按 mPower 的电源键开机，左下角可以看到设备已经进入 boot 模式，点击【浏览】选择固件。



然后点击【下载】按钮即可下载，下载完成后设备会自动关机。

4.5. 更多功能

请参考 <https://bbs.egteks.cn> 和 www.egteks.cn 网站或者关注官方公众号/抖音：

