

AutoTest 软件使用手册

修改日期:2024/1/18

当前版本:v1.0

Version No.	Description	Date
V1.0	Initial release	2024/01/18

目录

- 一. 软件使用须知..... 4
 - 1) 软件介绍..... 4
 - 2) 安装后端..... 4
 - 3) 运行软件..... 4
- 二. Main function 控制..... 6
 - 1) Main function 界面介绍..... 6
 - 2) Main function 选择和启动 6
 - 3) Battery lab..... 7
 - 4) Custom test 9
 - 5) Lithium test 11
 - 6) Jupiter test 11
 - 7) Natrium test 12
- 三. Instrument 控制 15
 - 1) 打开 Instrument 界面 15
 - 2) Instrument 界面介绍 15
 - 3) Instrument 连接..... 16
 - 4) Instrument 选择和设定 18
 - 5) Instrument 软复位 18
 - 6) Instrument 更多功能 19
- 四. MCU 控制..... 21
 - 1) 打开 MCU 界面..... 21
 - 2) 打开 MCU 界面..... 21
 - 3) MCU 连接与断开 22
 - 4) MCU 功能选择和设定 23
 - 5) Reg inversion 功能 24

一. 软件使用须知

1) 软件介绍

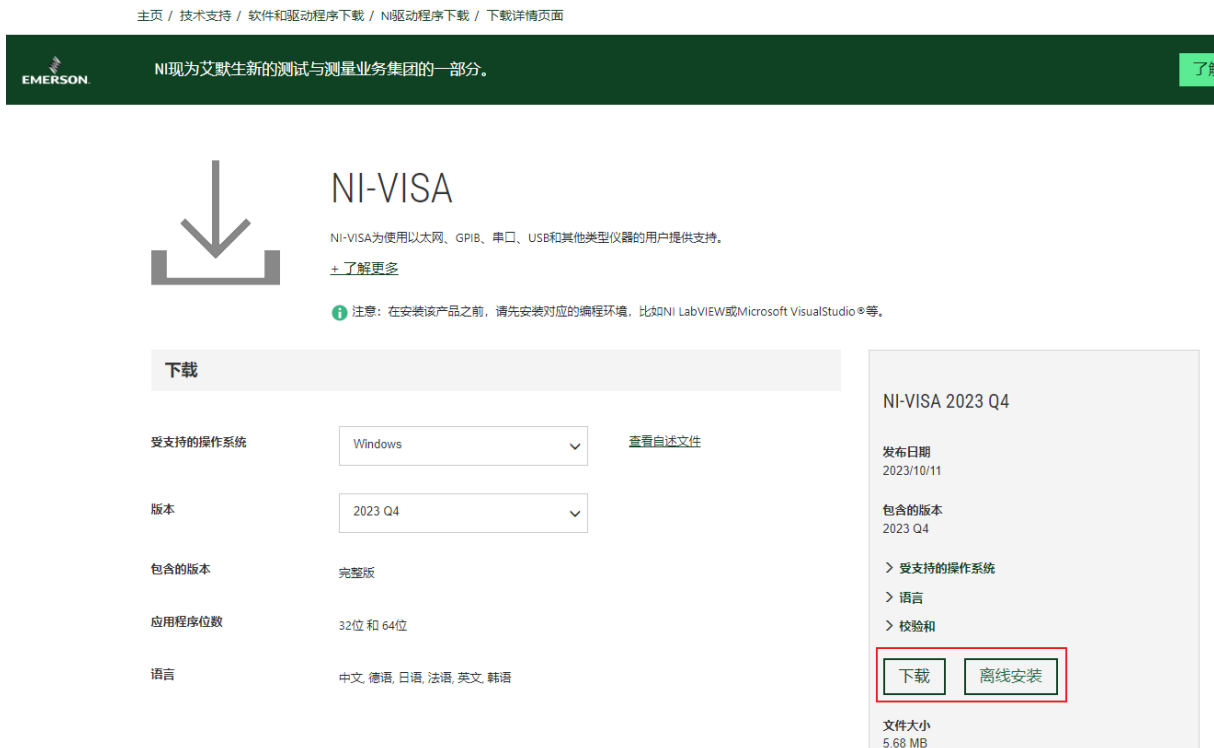
AutoTest 是一款拥有 GUI、支持多款仪器设备和多种通信协议的自动化测试软件。软件编写的代码种类为 python，版本为 3.8.2。

2) 安装后端

使用 AutoTest 软件需要安装底层驱动，驱动为 NI 公司的 NI-VISA，其下载地址为：

<https://www.ni.com/zh-cn/support/downloads/drivers/download.ni-visa.html#494653>

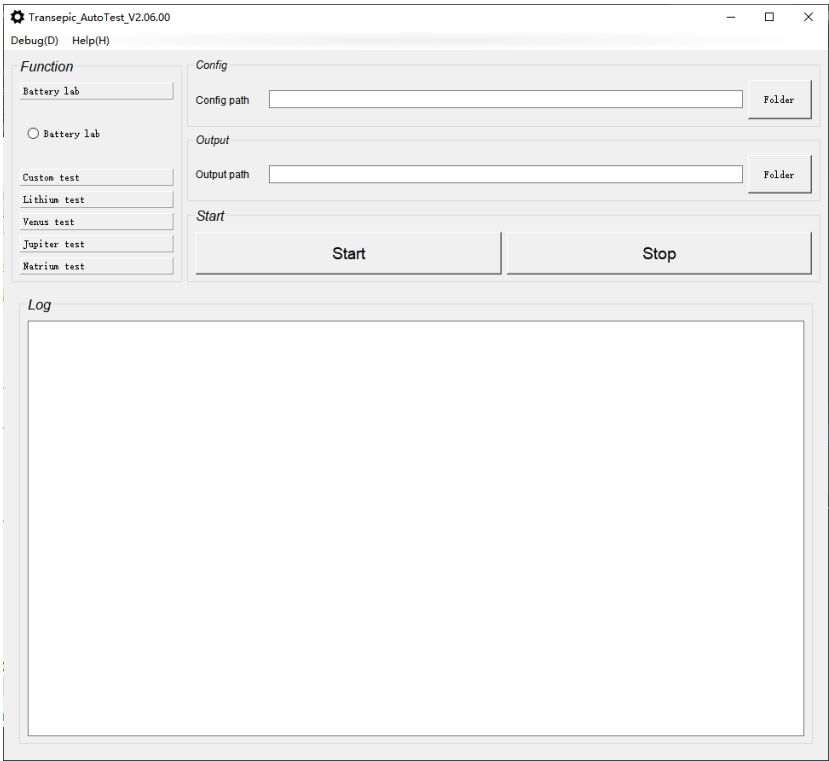
点击进入下载地址后，点击红色区域内的按钮下载并安装即可。



3) 运行软件

安装 NI-VISA 后，打开软件文件夹，点击 **AutoTest.exe** 运行软件。

名称	修改日期	类型	大小
_bz2.pya	2022/10/27 17:00	Python Extensio...	83 KB
_ctypes.pyd	2022/10/27 17:00	Python Extensio...	122 KB
_decimal.pyd	2022/10/27 17:00	Python Extensio...	264 KB
_elementtree.pyd	2022/10/27 17:00	Python Extensio...	173 KB
_hashlib.pyd	2022/10/27 17:00	Python Extensio...	45 KB
_lzma.pyd	2022/10/27 17:00	Python Extensio...	247 KB
_multiprocessing.pyd	2022/10/27 17:00	Python Extensio...	29 KB
_overlapped.pyd	2022/10/27 17:00	Python Extensio...	45 KB
_queue.pyd	2022/10/27 17:00	Python Extensio...	28 KB
_socket.pyd	2022/10/27 17:00	Python Extensio...	78 KB
_sqlite3.pyd	2022/10/27 17:00	Python Extensio...	86 KB
_ssl.pyd	2022/10/27 17:00	Python Extensio...	117 KB
AutoTest.exe	2024/1/12 15:40	应用程序	7,880 KB
base_library.zip	2024/1/12 15:39	压缩(zipped)文件...	1,005 KB
gear.ico	2023/11/10 17:10	ICO 文件	2 KB
libcrypto-1_1.dll	2022/10/27 17:00	应用程序扩展	3,303 KB
libffi-7.dll	2022/10/27 17:00	应用程序扩展	33 KB
libopenblas.FB5AE2TYXYH2IJRDKGD...	2022/10/27 17:00	应用程序扩展	34,859 KB
libssl-1_1.dll	2022/10/27 17:00	应用程序扩展	671 KB
old config.ini	2024/1/18 15:05	配置设置	1 KB
pyexpat.pyd	2022/10/27 17:00	Python Extensio...	186 KB

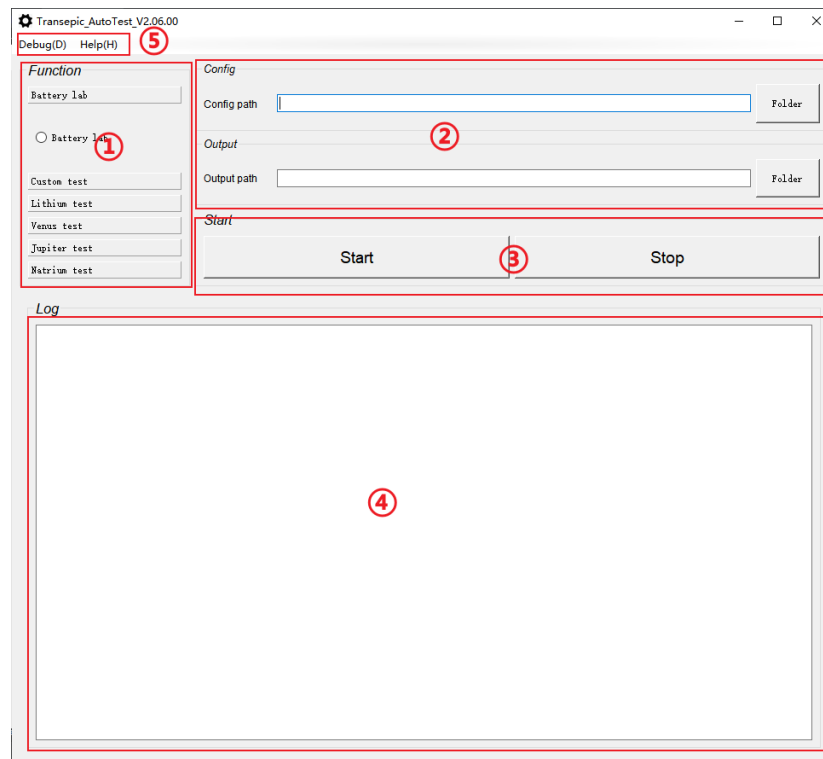


二. Main function 控制

1) Main function 界面介绍

Main function 界面共分为五个区域：

1. 功能选择区。选择项目和具体功能；
2. 配置区。为对应的功能选择 **Config**、**Output** 的路径；
3. 启动区。开启对应的功能测试；
4. 日志区。显示测试中收集的和报错的信息；
5. 其它功能区。进入其它功能。



2) Main function 选择和启动

1. 点击功能选择区中对应的项目，跳转到项目中；
2. 选择项目中的具体功能，功能为多选一；
3. 分别点击配置区中 **Config**、**Output** 的 **Folder** 按钮，分别为配置文件、结果输出选择对应的文件夹；
4. 点击启动区中的 **Start** 按钮，开始测试；

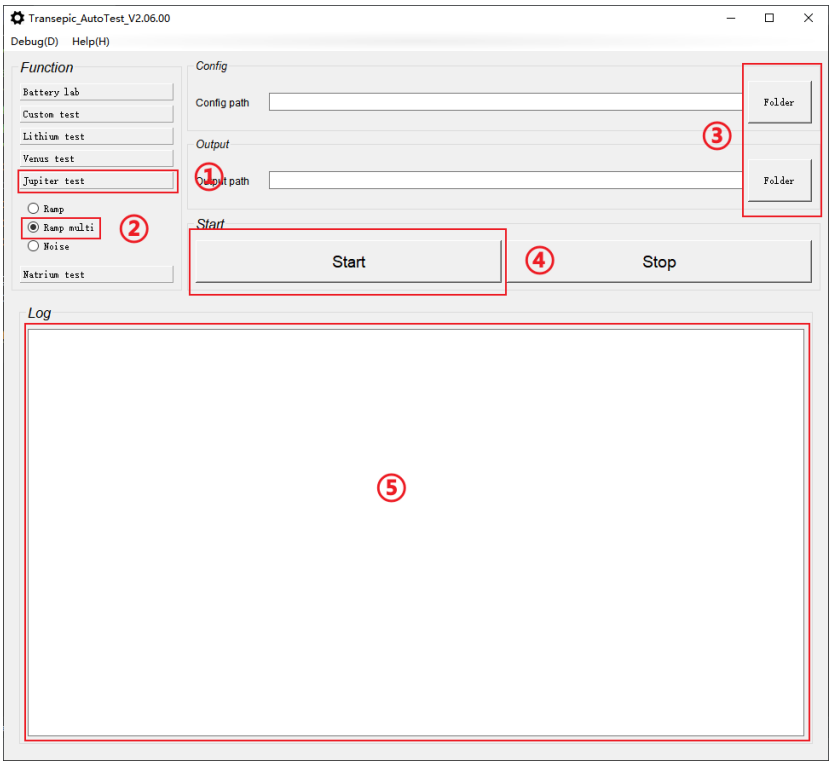
5. 查看 **Main** 界面的 **log** 区，检查测试信息。

注 1：当配置区中没有 **Output** 的路径时，会将 **Config** 的路径替代进去。

注 2：暂不开放启动区中 **Stop** 按钮的功能。

注 3：开始测试后，**Start** 按钮将下沉且无法再次点击，知道测试中断或完成。

注 4：在 **Config** 中写入的路径中若有多个配置文件，则会以文件名排序的第一个为准。



3) Battery lab

Battery lab 是一项对电池进行充电、放电的测试，并实时记录电池电压、电流和温度三个参数。有一项功能。Battery lab 配置文件类型为 ini。

Battery lab 功能测试参数需要如下：

Config item name	Description
Temperature_Setting_Instrument	设定温度设备名称
Temperature_Setting_Communication	设定温度设备通信方式
Temperature_Measurement_Instrument	测量温度设备名称
Temperature_Measurement_Communication	测量温度设备通信方式

Temperature_Measurement_Compensation	测量温度设备温度补偿值，单位：℃
Set_Temperature	温度设定值，单位：℃
Charge_Instrument	充电设备名称
Charge_Communication	充电设备通信方式
Charge_Voltage	充电电压，单位：mV
Charge_Voltage_Range	充电电压范围，单位：mV
Charge_Current	充电电流，单位：mA
Discharge_Voltage	放电电压，单位：mV
Discharge_Current_1	第一次放电电流，单位：mA
Discharge_Current_2	第二次放电电流，单位：mA
Discharge_Current_3	第三次放电电流，单位：mA
Charge_Voltage_Threshold	充满电时最低电压的阈值，单位：mV
Temperature_Time_Threshold	温度最少保持时长的阈值，单位：s
Relax_Current	休眠时电流，单位：mA
Charge_Time_Threshold	充满电时最少保持时长的阈值，单位：s
Charge_Current_Threshold	充满电时最高电流的阈值，单位：mA
Discharge_Time_Threshold	放电时最多时长的阈值，单位：s
Discharge_Voltage_Threshold_1	放电时判断电池电压的第一个阈值，单位：mV
Discharge_Voltage_Threshold_2	放电时判断电池电压的第二个阈值，单位：mV
Discharge_Voltage_Threshold_3	放电时判断电池电压的第三个阈值，单位：mV
Reset_Time_Threshold	电池等待最长保持时长的阈值，单位：s
Voltage_Judgement_1	第一电压判断阈值，电压<阈值，单位：mV
Voltage_Judgement_2	第二电压判断阈值，电压>阈值，单位：mV
Voltage_Judgement_3	第三电压判断阈值，电压<阈值，单位：mV
Voltage_Judgement_4	第四电压判断阈值，电压<阈值，单位：mV
Voltage_Judgement_5	第五电压判断阈值，电压<阈值，单位：mV
Voltage_Judgement_6	第六电压判断阈值，电压<阈值，单位：mV
Voltage_Judgement_7	第七电压判断阈值，电压<阈值，单位：mV

4) Custom test

Custom test 是一项自定义测试，可以根据用户自己的需求来配置测试项目，目前支持十几种设备的控制和几种单片机的通信。Custom test 支持多项功能，包括步进、循环、条件判断、跳转等。Custom test 配置文件类型为 `xlsx`。

Custom test 功能配置参数格式如下：

Step	Instrument	Function	Parameter	Time	Condition	Next
X	XXX	XXX	XXX	XX	XXX	X
..

如无特殊功能，测试将按照 `xlsx` 顺序从上到下依次进行。其中功能配置的列含义如下所示：

1. 列含义

- Step：表示该行操作的序列号，类型为 `int`，不同行之间的序列号不能相同；例：1
- Instrument：表示该行进行操作的设备名称，类型为 `string`；例：E36312A
- Function：表示该行进行的具体操作，类型为 `string` 或 `none`；例：open 或 operate=set_voltage
- Parameter：表示该行进行的具体操作的补充说明、参数设定，类型为 `string` 或 `none`；例：Voltage=2;Current=0.1;Channel=1 或 Channel=1
- Time：表示该行进行操作之前的等待时间，类型：float 或 `none`；例：0.01
- Condition：表示该行需要进行的条件判断，类型为 `string` 或 `none`；例：Voltage>10 或 Time<1
- Next：表示该行完成后需要进行的下一行的序列号，类型为 `int` 或 `string` 或 `none`；例：1

此外，Custom test 包括其它特殊功能以满足用户更多需求，其中有以下几项：

2. 步进功能

- 功能介绍：让设备按照想要的参数一步步进行操作。
- 使用规范：在 **Parameter** 中，其中一个参数包括下述格式。
- 格式（例）：Voltage=1:2:10 或 Voltage=1:10;Current=1
- 格式解释：
 - 格式有 `xx:yy:zz` 或 `xx:yy` 两种。
 - `xx:yy:zz` 中左边代表起始值，中间代表步进值，右边代表终止值。
 - `xx:yy` 中左边代表起始值，右边代表终止值。
 - 若多项参数中有一项参数为步进，则会将其它参数补充到每一步中。

3. 循环功能

- 功能介绍：将多行操作组成一个组，按照顺序优先完成组内操作。
- 使用规范：在 **Next** 中，填写下述格式的参数。
- 格式（例）：**{4,8,11}**
- 格式解释：
 - i. 格式为{x,y...}，个数无上限。
 - ii. 当 **Step=x** 行操作完成后，下一步直接跳转到 **Step=y** 行，以此类推。
 - iii. 当循环完成后，会跳回到 **Step=x+1** 行。
- 其它：循环优先级>步进优先级。当循环中的某一行包含步进功能时，会使步进中的每一步都拥有该循环。

4. 条件判断功能

- 功能介绍：判断该行是否满足条件，如未满足，则一直重复执行该行，直到满足后跳出。
- 使用规范：在 **Condition** 中，填写下述格式的参数。
- 格式（例）：**Voltage>=10** 或 **Voltage<=1,Current>0.1** 或 **(Voltage==1;Current!=0.1),Time<10**
- 格式解释：
 - i. 判断类型有<、>、<=、>=、==、!=共六种。
 - ii. 连接语句‘,’表示与，‘;’表示或，‘()’表示优先判断。
 - iii. 可同时支持最多三个判断。

5. 跳转功能

- 功能介绍：完成该行后，直接跳转到用户想要的一行进行下一步操作。
- 使用规范：在 **Next** 中，填写下述格式的参数。
- 格式（例）：**7**
- 格式解释：
 - i. 格式为 **y**。
 - ii. 当 **Step=x** 行操作完成后，下一步直接跳转到 **Step=y** 行。完成 **Step=y** 后，自动跳转到 **Step=y+1** 行。

5) Lithium test

Lithium test 是一项针对 Lithium 项目芯片的测试。Lithium test 配置文件类型为 ini。（待更新）

6) Jupiter test

Jupiter test 是一项针对 Jupiter 项目芯片的测试。有三项功能，**Ramp**，**Ramp multi**，**Noise**。Jupiter test 配置文件类型均为 ini。

1. Ramp

Ramp 是针对 Jupiter 单颗芯片的 INL 测试，其功能测试参数需要如下：

Config item name	Description
Temperature_Setting_Instrument	设定温度设备名称
Temperature_Setting_Communication	设定温度设备通信方式
Temperature_Setting_Flag	设定温度设备开关标志
Power_Instrument	电源设备名称
Power_Communication	电源设备通信方式
Power_Setting_Flag	电源设备开关标志
ADC_Setting_Instrument	ADC 输出设备名称
ADC_Setting_Communication	ADC 输出设备通信方式
ADC_Setting_Flag	ADC 输出设备开关标志
ADC_Measurement_Instrument	ADC 测量设备名称
ADC_Measurement_Communication	ADC 测量设备通信方式
ADC_Measurement_Flag	ADC 测量设备开关标志
Control_Instrument	MCU 设备名称
Control_Communication	MCU 设备通信方式
Control_Setting_Flag	MCU 设备开关标志
Set_Temperature	设定温度值，单位：℃
Power_Voltage_1	电源设备第一个电压设定值，单位：V
Power_Current_1	电源设备第一个电流设定值，单位：A

Power_Channel_1	电源设备第一个通道设定值
Power_Voltage_2	电源设备第二个电压设定值，单位：V
Power_Current_2	电源设备第二个电流设定值，单位：A
Power_Channel_2	电源设备第二个通道设定值
Power_Voltage_3	电源设备第三个电压设定值，单位：V
Power_Current_3	电源设备第三个电流设定值，单位：A
Power_Channel_3	电源设备第三个通道设定值
Temperature_Time_Threshold	温度保持时长最短阈值，单位：s
Start_Voltage	ADC 输出起始电压值，单位：V
Step_Voltage	ADC 输出步进电压值，单位：V
End_Voltage	ADC 输出终止电压值，单位：V
Start_Current	ADC 输出电流值，单位：A
Reg_Bus_Number	MCU 输出管脚序号
Reg_Slave	I2C Slave 地址
Reg_Address	I2C Reg 地址
Measurement_Period	测量周期等待时长，单位：s
Retest_Time	每次测量重复操作次数
Data_Average_Flag	单次测量数据合并标志

2. Ramp multi

Ramp multi 是针对 Jupiter 1~4 颗芯片的 INL 测试，可根据需求同时测量 1~4 颗芯片的 INL 数据。其功能测试参数与 **Ramp** 相同，只需要在 **Reg_Slave** 参数中写入多颗芯片的 Slave 地址即可。

3. Noise

Noise 是针对 Jupiter 1~4 颗芯片的 Noise 测试，可根据需求同时测量 1~4 颗芯片的 noise 数据。其功能测试参数与 **Ramp** 相同，只需要在 **Reg_Slave** 参数中写入多颗芯片的 Slave 地址即可。

7) Natrium test

Jupiter test 是一项针对 Jupiter 项目芯片的测试。有三项功能，**Ramp**，**Noise**，**Temperature**。Jupiter test 配置文件类型均为 ini。

1. Ramp

Ramp 是针对 Natrium 单颗芯片的 INL 测试，其功能测试参数与 Jupiter **Ramp** 相同。

2. Noise

Noise 是针对 Natrium 单颗芯片的 noise 测试，其功能测试参数与 Jupiter **Ramp** 相同。

3. Temperature

Temperature 是针对 Natrium 单颗芯片的 PTAT 测试，其功能测试参数与 Jupiter 大致相同，仅有几项区别，如下所示（增加项标记为蓝色，减少项标记为红色）：

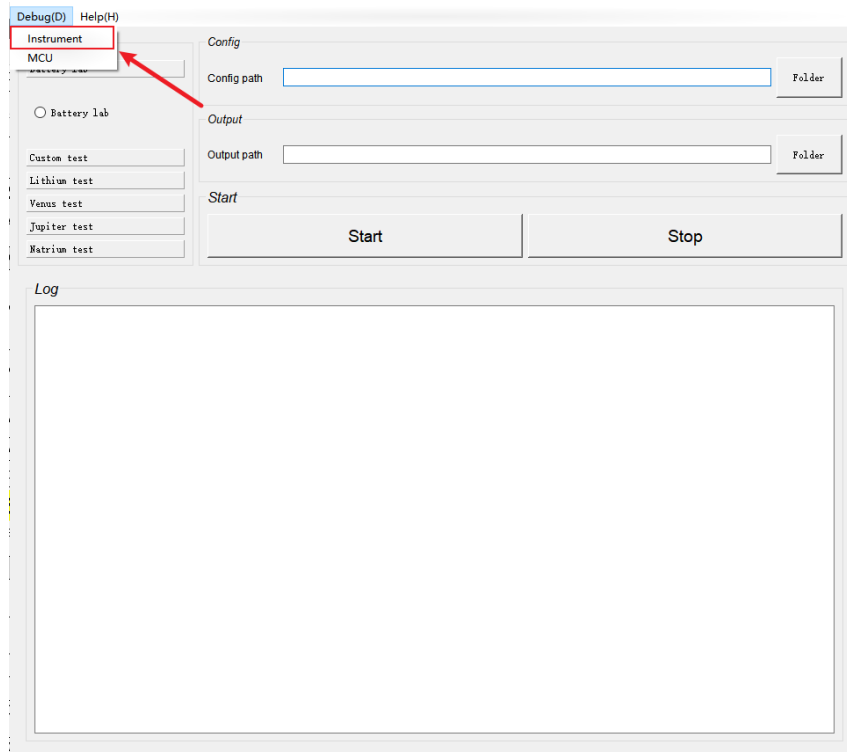
Config item name	Description
Temperature_Setting_Instrument	设定温度设备名称
Temperature_Setting_Communication	设定温度设备通信方式
Temperature_Setting_Flag	设定温度设备开关标志
Power_Instrument	电源设备名称
Power_Communication	电源设备通信方式
Power_Setting_Flag	电源设备开关标志
ADC_Setting_Instrument	ADC 输出设备名称
ADC_Setting_Communication	ADC 输出设备通信方式
ADC_Setting_Flag	ADC 输出设备开关标志
ADC_Measurement_Instrument	ADC 测量设备名称
ADC_Measurement_Communication	ADC 测量设备通信方式
ADC_Measurement_Flag	ADC 测量设备开关标志
Control_Instrument	MCU 设备名称
Control_Communication	MCU 设备通信方式
Control_Setting_Flag	MCU 设备开关标志
Set_Temperature	设定温度值，单位：℃
Power_Voltage_1	电源设备第一个电压设定值，单位：V
Power_Current_1	电源设备第一个电流设定值，单位：A
Power_Channel_1	电源设备第一个通道设定值
Power_Voltage_2	电源设备第二个电压设定值，单位：V
Power_Current_2	电源设备第二个电流设定值，单位：A

Power_Channel_2	电源设备第二个通道设定值
Power_Voltage_3	电源设备第三个电压设定值，单位：V
Power_Current_3	电源设备第三个电流设定值，单位：A
Power_Channel_3	电源设备第三个通道设定值
Temperature_Time_Threshold	温度保持时长最短阈值，单位：s
Start_Temperature	起始温度设定值，单位：℃
Step_Temperature	步进温度设定值，单位：℃
End_Temperature	终止温度设定值，单位：℃
Start_Voltage	ADC 输出起始电压值，单位：V
Step_Voltage	ADC 输出步进电压值，单位：V
End_Voltage	ADC 输出终止电压值，单位：V
Start_Current	ADC 输出电流值，单位：A
Reg_Bus_Number	MCU 输出管脚序号
Reg_Slave	I2C Slave 地址
Reg_Address	I2C Reg 地址
Measurement_Period	测量周期等待时长，单位：s
Retest_Time	每次测量重复操作次数
Data_Average_Flag	单次测量数据合并标志

三. Instrument 控制

1) 打开 Instrument 界面

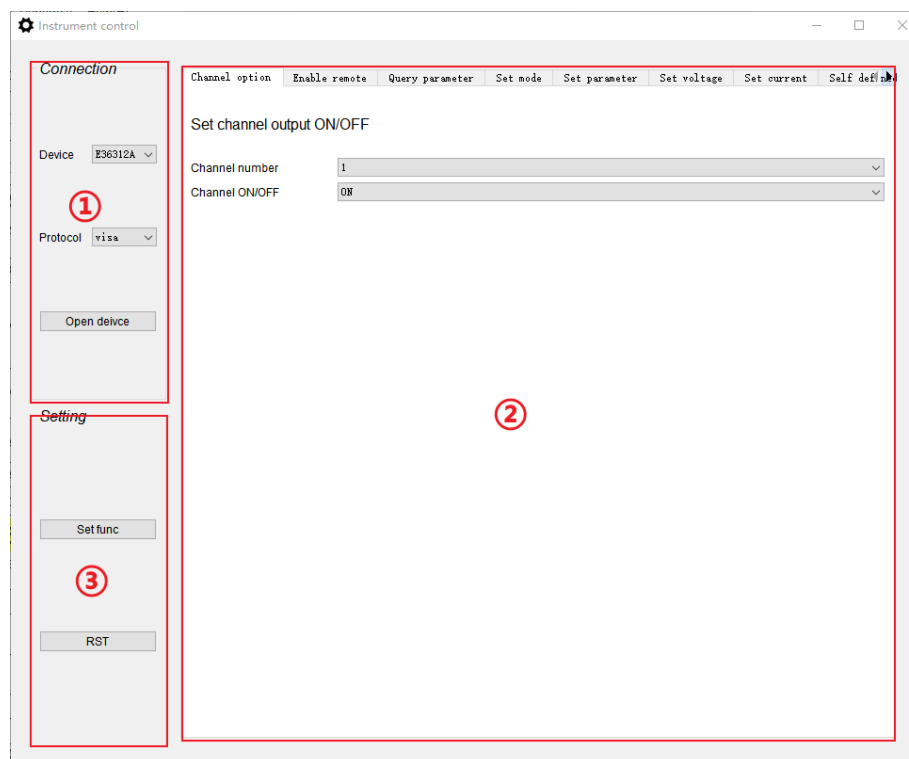
1. 点击左上角 **Debug** 按钮；
2. 点击 **Instrument** 按钮，进入 Instrument control 界面。



2) Instrument 界面介绍

Instrument 界面共分为三个区域：

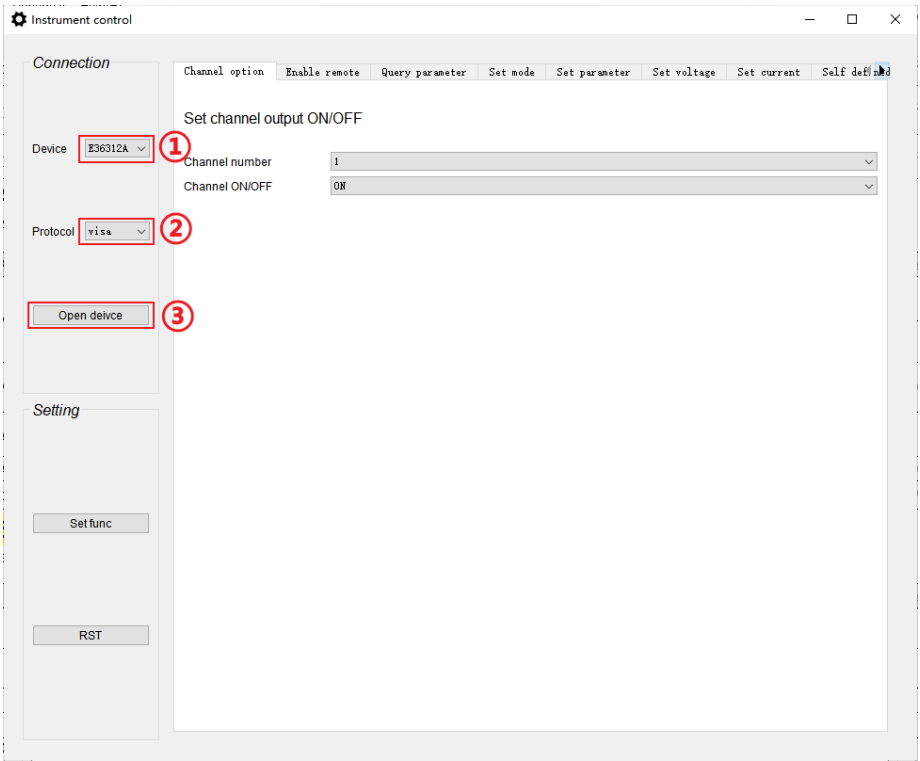
1. 型号选择区。选择 instrument 型号、通信方式，进行连接；
2. 功能选择区。选择对应 instrument 具体功能，设定相应参数；
3. 功能设定区。实施选择的 instrument 型号和功能，以及对 instrument 进行软复位。



3) Instrument 连接

打开 Autotest 软件后，需对 instrument 进行首次连接，才可进行操作。步骤如下：

1. 打开 **Device** 下拉列表，选择想要控制的 instrument；
2. 打开 **Protocol** 下拉列表，选择 instrument 的通信方式；
3. 点击 **Open device** 按钮，与 instrument 进行连接；
4. 查看 **Main** 界面的 **log** 区，确认 instrument 连接成功。



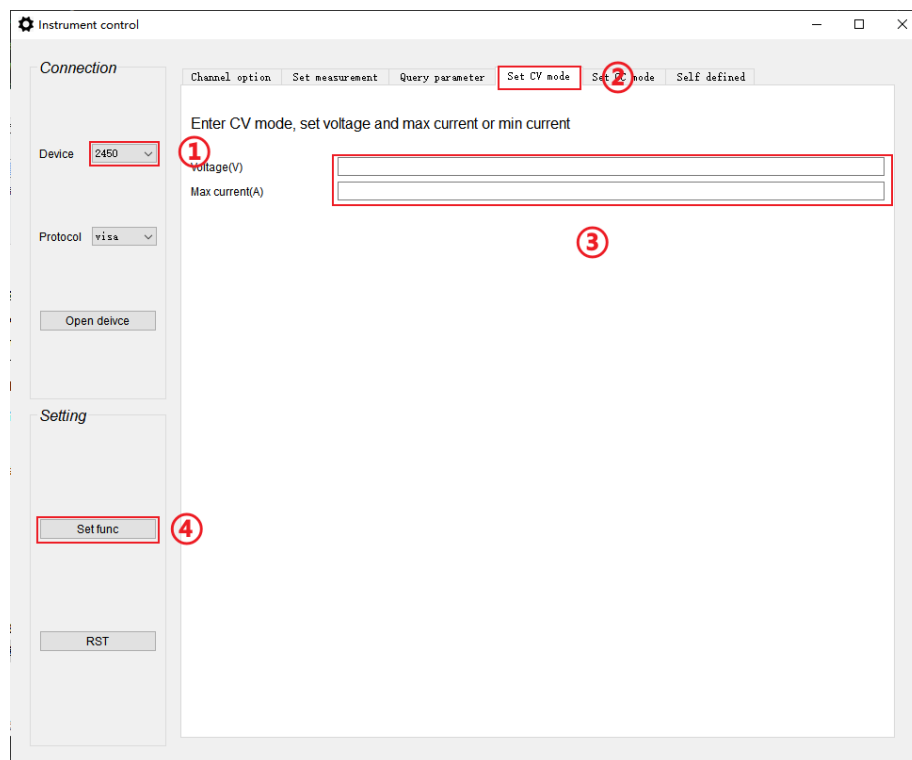
各 instrument 支持的通信方式如下：

Instrument name	Protocol type
E36312A	Visa
DMM7510	
2450	
B2912A	
DP832	
DG1062Z	
DP932	
IT8811	
DHT260	Tcp/Rtu
DL11B	Rtu
2400(无设备)	Visa

4) Instrument 选择和设定

通过 Device 下拉列表选择 instrument 后，会自动跳转到对应 instrument 的功能区，可选择功能和设定参数。步骤如下：

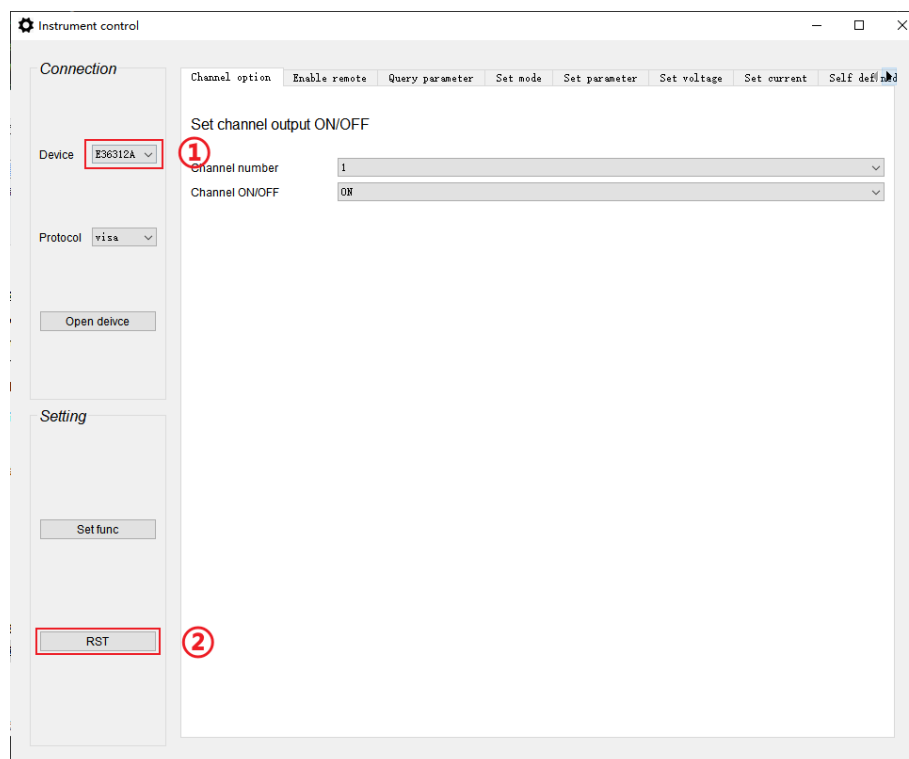
1. 打开 **Device** 下拉列表，选择想要控制的 instrument；
2. 选择功能选择区中的某项功能，进入该功能的参数设定页面；
3. 根据该功能的介绍和参数数目，写入想要的参数；
4. 点击功能设定区中的 **Set func** 按钮，控制 instrument 完成该功能；
5. 查看 **Main** 界面的 **log** 区，确认 instrument 操作已完成。



5) Instrument 软复位

支持对 instrument 进行软复位，去除之前对 instrument 的各项设定。步骤如下：

1. 打开 **Device** 下拉列表，选择想要控制的 instrument；
2. 点击功能设定区中的 **RST** 按钮，对 instrument 进行软复位；



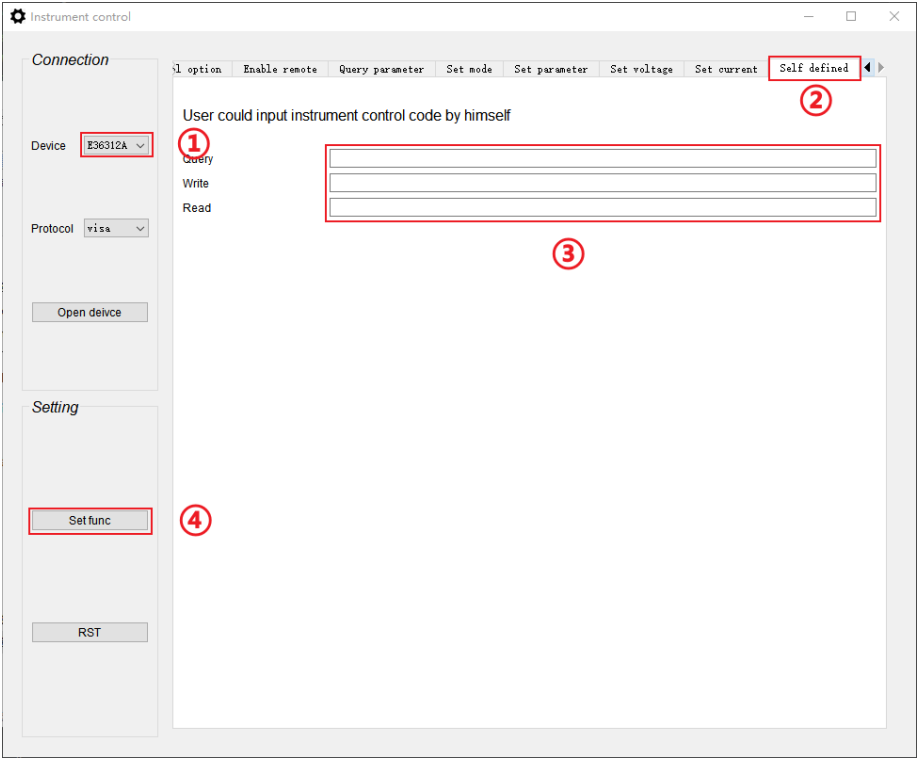
6) Instrument 更多功能

支持对 instrument 更多功能的操作。步骤如下：

1. 打开 **Device** 下拉列表，选择想要控制的 instrument；
2. 点击功能选择区中的 **Self defined** 页面，进入自定义界面中；
3. 根据需求写入想要的命令，只用写入一行；
4. 点击功能设定区中的 **Set func** 按钮，控制 instrument 完成该功能；

注 1：写入的命令格式由当前通信方式决定。

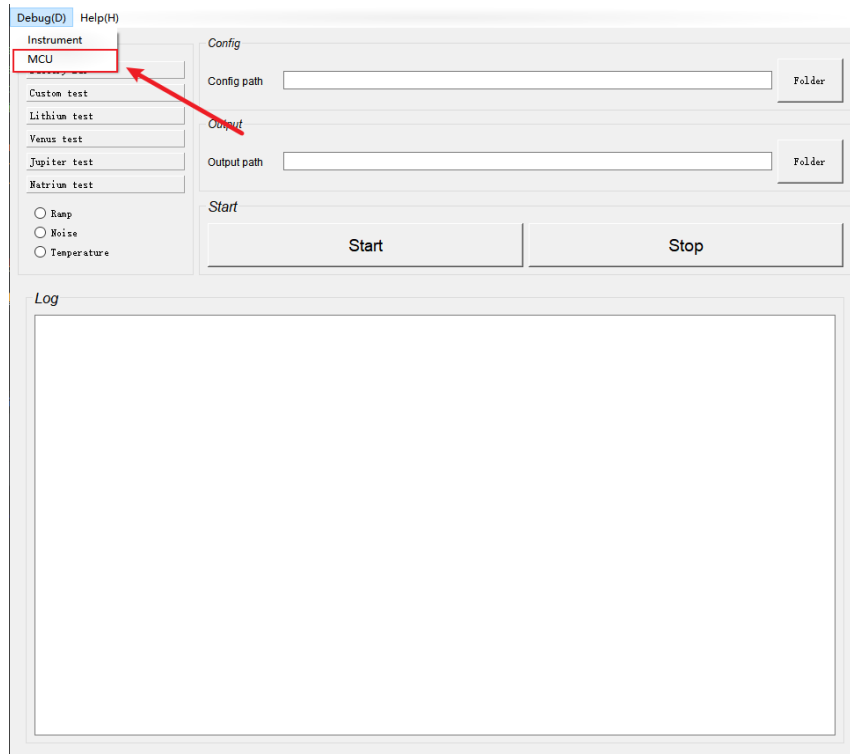
注 2：Query = Write + Read



四. MCU 控制

1) 打开 MCU 界面

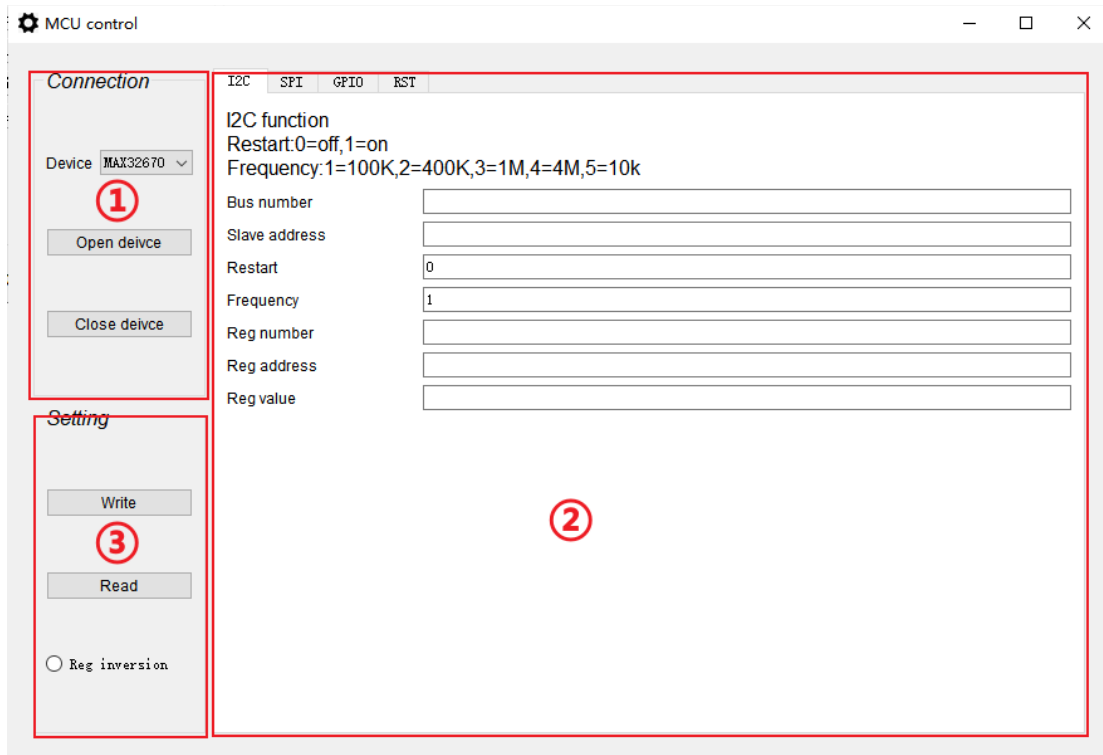
1. 点击左上角 **Debug** 按钮；
2. 点击 **MCU** 按钮，进入 MCU control 界面。



2) 打开 MCU 界面

MCU 界面共分为三个区域：

1. 型号选择区。选择 MCU 型号，进行连接，断开连接；
2. 功能选择区。选择对应 MCU 具体功能，设定相应参数；
3. 功能设定区。实施选择的 MCU 型号和功能，以及其它功能。



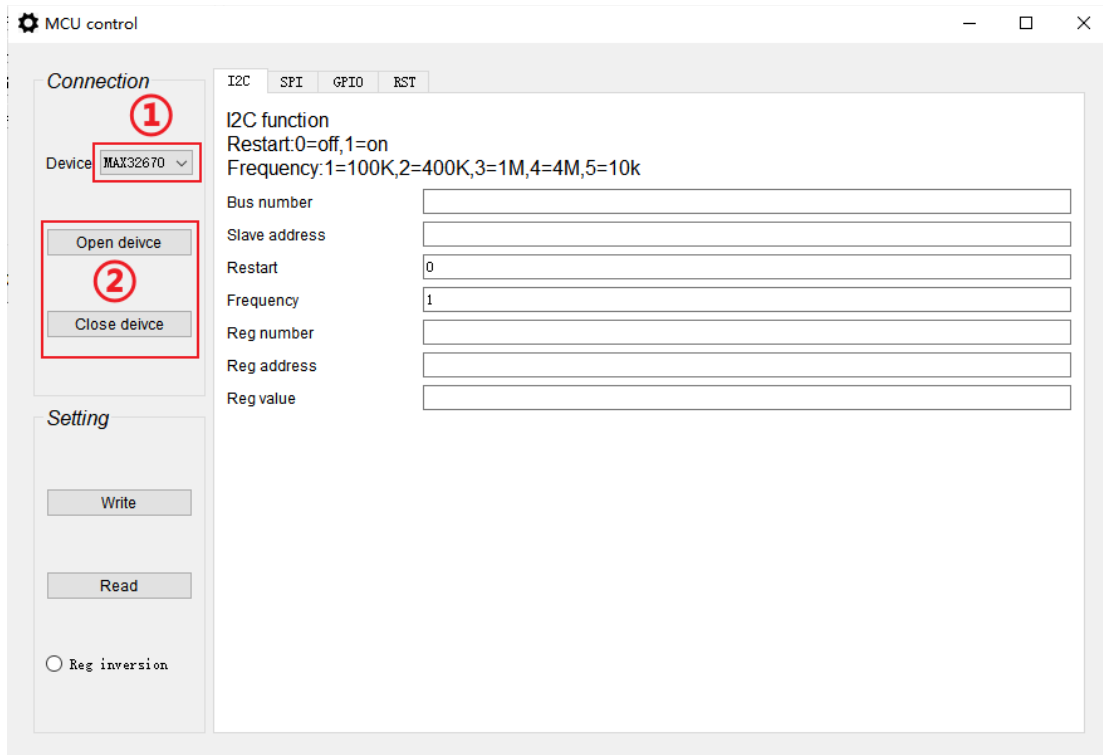
3) MCU 连接与断开

打开 Autotest 软件后，需对 MCU 进行首次连接，才可进行操作。步骤如下：

1. 打开 **Device** 下拉列表，选择想要控制的 MCU；
2. 点击 **Open device** 按钮，与 MCU 进行连接；
3. 查看 **Main** 界面的 **log** 区，确认 MCU 连接成功。

MCU 连接成功后，可以断开连接。步骤如下：

1. 打开 **Device** 下拉列表，选择想要控制的 MCU；
2. 点击 **Close device** 按钮，断开 MCU 连接；
3. 查看 **Main** 界面的 **log** 区，确认 MCU 断开成功。

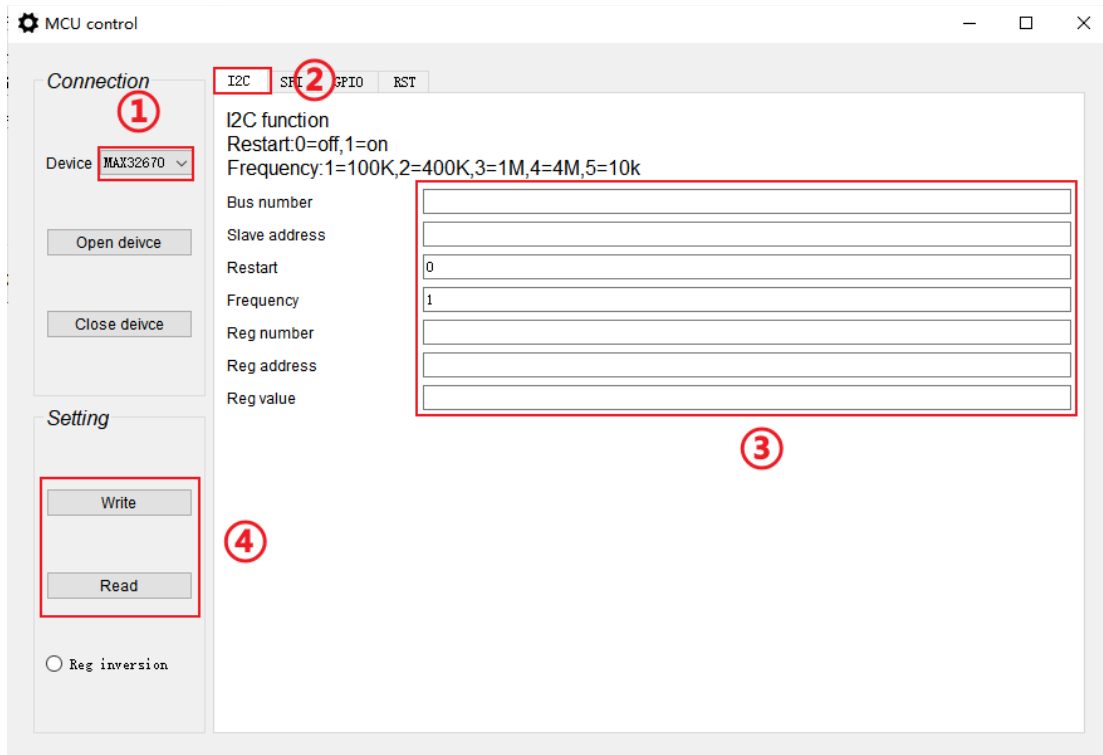


4) MCU 功能选择和设定

MCU 成功连接后，可根据需求选择功能并设定参数。步骤如下：

1. 打开 **Device** 下拉列表，选择想要控制的 MCU；
2. 选择功能选择区中的某项功能，进入该功能的参数设定页面；
3. 根据该功能的介绍和参数数目，写入想要的参数；
4. 点击功能设定区中的 **Write** 或 **Read** 按钮，控制 MCU 完成该功能；
5. 查看 **Main** 界面的 **log** 区，确认 MCU 操作已完成。

注 1：选择 RST 功能时，**Write** 或 **Read** 操作结果相同。



5) Reg inversion 功能

MCU 支持数据大小端变化功能，点击功能设定区中的 **Reg inversion** 按钮，将改变数据大小端。