


HexWriter_SD502 是一款程序烧写软件，是通用仿真器 SD502 的配套软件，可供芯片烧写人员和开发人员对  电能计量 SoC 芯片进行 Flash 脱机烧写和在线烧写程序。

V6.1

2016-8-12


特点


- 适用于所有  电能计量 SoC 芯片的程序烧写
- 支持 Flash 脱机烧写和在线烧写程序
- 支持 CRC 校验
- 支持电表参数配置
- 支持页操作
- 支持芯片 Flash 内容读取和擦除
- 支持 RTC 参数配置支持
- 支持烧写数量控制和烧写数量不受限两种控制方式
- 支持 800k 和 3.2M 两种 MCU 频率下的调试
- 支持 Flash 脱机烧写（Download）、RTC 秒脉冲校正（RTC Cal）和 RTC 时间设置（Set Time）三种脱机操作（可以进行三种操作的随意组合）
- 支持连续烧写和手动烧写两种启动模式
- 支持晶体停振检测和晶体停振不检测两种方式
- 支持 hex 加密
- 支持对比 HexWriter、SD502 及芯片的 Hex 码比较
- 支持烧写数量统计
- 在连续烧写模式下并且使用万高提供的专用 SOCKET 小板（VI_Socket_FPQ_V9811_V1.1）进行裸片烧写模式下，具有合盖检测功能，减少生产中漏烧比例

目录

特点	1
目录	2
1. 适用范围	3
2. 启动 HexWriter_SD502	3
3. HexWriter_SD502 界面介绍	3
3.1 型号选择	4
3.2 打开/配置 Hex 文件	4
3.3 数据表格	6
3.4 参数配置	6
3.5 页操作	8
3.6 CRC 校验	9
3.7 加密设置	11
3.8 代码长度	12
3.9 在线编程操作	12
3.10 脱机烧写操作	13
3.11 脱机烧写功能配置	13
3.12 配置信息保存	16
3.13 Hex 加密烧写操作	16
3.14 读 Flash	18
3.15 RTC 相关	21
3.16 帮助	23
3.17 退出	24
4. 系统提示说明	24
5. 附录_CRC 校验	25
5.1 CRC 算法	25
5.2 CRC 校验例程	26
版本更新说明	32

1. 适用范围

HexWriter_SD502 与 SD502 配套使用，该软件适用于所有  电能计量 SoC 芯片，包括 V98xx/V9311/V9011/V9401/V9401A/V9003/V9103/V99xx。

当做 RTC 秒脉冲校正和 RTC 时间设置时，软件仅适用于  电能计量 SoC 芯片 V98xx 和 V99xx。

2. 启动 HexWriter_SD502

启动 **HexWriter_SD502** 之前需要先安装 USB 驱动，USB 驱动安装方法见，SD502 用户手册中驱动安装描述方法，一台电脑只需要安装一次。双击 **HexWriter_SD502.exe** 文件，即可启动 **HexWriter_SD502**。

3. HexWriter_SD502 界面介绍

如图 1 所示，HexWriter_SD502 操作界面由 7 部分组成：

- ① **菜单栏**，包括“File（文件）”、“Online Operation（在线编程操作）”、“Offline operation（脱机烧写操作）”、“ReadFlash（读 Flash）”、“Parameter（参数配置）”、“RTC（RTC 相关）”和“Help（帮助）”等选项；
- ② **快速工具栏**，该工具栏提供快速方式进行菜单栏里提供的所有操作，不包括“Parameter（参数配置）”和“RTC（RTC 相关）”操作除外；
- ③ **配置区**：在配置区，用户可完成包括“Device（芯片型号）”和“.hex Editor（hex 类型）”的选择、“CRC Verification（CRC 校验）”、“Page Operation（页操作）”、“Encryption（加密设置）”和“CodeBank（代码长度配置）”，以及“File Operation（打开并配置 hex 文件）”等一系列操作；
- ④ **进度条**：点击“AutoLoad（自动下载）”或“OffLoad（脱机下载）”时会显示下载进度；
- ⑤ **信息区**：显示用户每次操作的结果。点击 Clear（清空）按钮可清除该信息区内的所有信息；
- ⑥ **数据表格**：显示打开的 hex 文件或配置文件（cfg 文件，.ini）内容；
- ⑦ **状态栏**：与信息区一样，显示用户每次操作的结果。

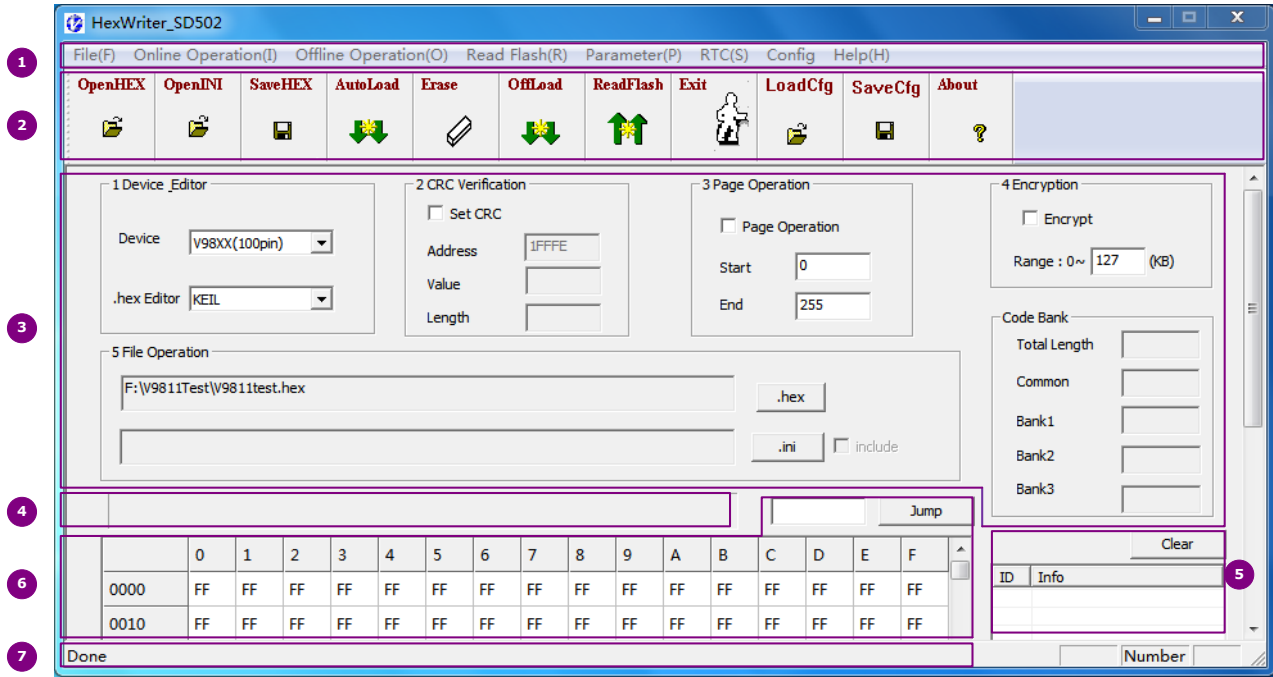


图 1 HexWriter_SD502 操作界面

3.1 型号选择

启动 HexWriter_SD502 后,在 **Device_Editor** 框内选择需要处理的目标板型号和要下载的 hex 文件类型。

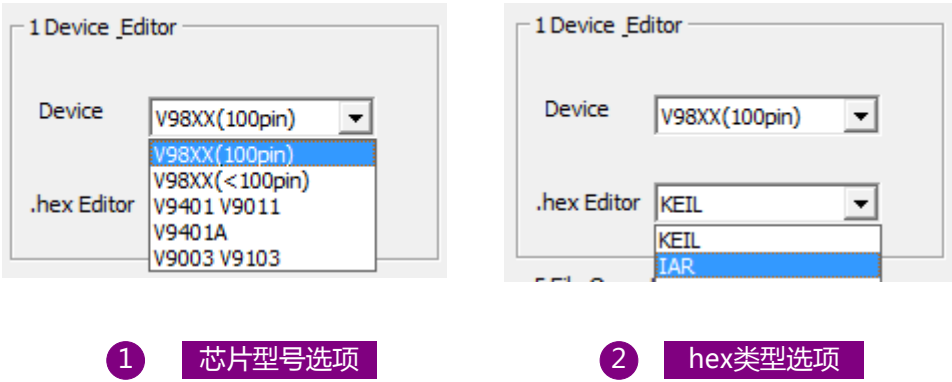
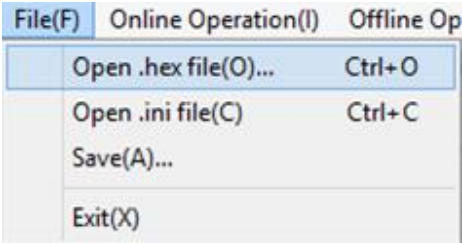


图 2 “Device Editor（型号选择）”选项

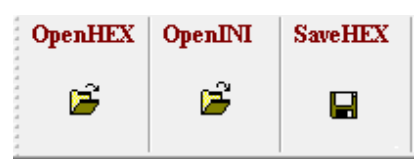
3.2 打开/配置 Hex 文件

如图 3 所示,在 HexWriter_SD502 操作界面上,用户可以通过菜单栏、快速工具栏和配置区打开 Hex 文件、打开配置文件或保存配置文件。

1 菜单栏“文件”选项



2 快速工具栏中对应的操作键



3 配置区中对应的操作



图 3 打开/配置 Hex 文件

- 1. **.Hex/OpenHEX/Open.hex file:** 即打开用户需要下载的.hex 文件，其内容会显示在**数据表格**中。
- 2. **.ini/OpenINI/Open .ini file:** 打开.ini 文件，文件内容将显示在数据表格中。用户可使用万高科技提供的 function.ini 文件，也可根据需要按 function.ini 文件的规范自行编写.ini 文件，用于电表参数配置。用户可通过图 3 所示的三种方式打开配置文件后，.ini 文件会自动加载入数据表格内，起始地址是.ini 文件中设置的 **BaseAddress** (**基地址**，详见图 6)。如果用户需要修改电表参数配置，可以通过参数配置窗口进行电表参数修改，详见参数配置操作说明。

必须先打开 hex 文件，再打开.ini 文件。如果先打开.ini 文件，再打开 hex 文件，已加载入数据表格中的.ini 文件会被清空。

- 3. **Save:** 将**数据表格**中的所有内容，或者根据用户设置的页码范围，将**数据表格**中相应范围内的数据保存为.txt 文件。.txt 文件格式与数据表格的格式一致，显示了 Flash 数据的范围和内容。

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7000	02	17	B5	02	15	AC	22	22	01	22	22	02	15	CA	22	22
7010	22	22	22	02	15	E8	22	22	22	22	02	16	06	90	28	
7020	68	E0	30	E5	03	7F	01	22	7F	00	22	02	16	24	90	28
7030	78	E4	F0	90	28	7D	F0	D2	84	22	22	02	14	4C	22	22
7040	22	22	22	02	09	AC	22	22	22	22	02	0A	76	22	22	
7050	22	22	22	02	0D	71	7D	78	7C	28	8D	82	8C	83	EF	F0
7060	22	FF	FF	02	16	42	BB	01	06	89	82	8A	83	F0	22	50
7070	02	F7	22	BB	FE	01	F3	22	F8	BB	01	0D	E5	82	29	F5
7080	82	E5	83	3A	F5	83	E8	F0	22	50	06	E9	25	82	C8	F6
7090	22	BB	FE	05	E9	25	82	C8	F2	22	EF	8D	F0	A4	A8	F0
70A0	CF	8C	F0	A4	28	CE	8D	F0	A4	2E	FE	22	BC	00	0B	BE
70B0	00	29	EF	8D	F0	84	FF	AD	F0	22	E4	CC	F8	75	F0	08

地址

数据

图 4 .txt 文件格式

3.3 数据表格

数据表格显示所有读取到的，或即将被烧写入目标板的 Flash 数据，表格中的数据地址和目标板 Flash 的地址一一对应。如图 5 所示，

- ① 为每一行数据的起始地址；
- ② 显示每一个 Flash 地址对应的数据，如地址 0xFC06 的数据为 0x02；
- 在 ③ 中填入 Flash 地址，再点击 **Jump**，可直接跳转到**数据表格**中的相应位置。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	02	09	80	FF	FF	FF	FF	FF	01	FF	FF	FF	FF	FF	FF	FF
0010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

图 5 数据表格

3.4 参数配置

正确连接 SD502 与 PC 后，如果用户需要修改电表参数配置，可以点击菜单栏中的 **Parameter**，进入如图 6 所示的 **Configure** 窗口。该窗口是一个.ini 文件的用户界面：

1. 点击 **Clear**，清除各配置项的 **Setting**（设置数据）；
2. 点击 **Reset**，各配置项的 **Setting**（设置数据）被恢复为默认值；
3. 点击 **Open**，选择载入一个.ini 文件。如果.ini 文件与 HexWriter_SD502.exe 在同一目录下，点击 **Configure** 时，打开的窗口自动显示待配置的参数。如果.ini 文件与 HexWriter_SD502.exe 不在同一目录下，点击 **Configure** 时，将打开一个空窗口，在打开的窗口中，点击 **Open** 载入用户所需的.ini 文件。
4. 点击 **Save**，将配置后的内容保存为一个.ini 文件；
5. 点击 **OK**，将配置好的参数载入 HexWriter_SD502 的**数据表格**中。
6. **BaseAdress (hex)** 处显示配置参数的基地址，反映在**数据表格**中，即从基地址（如图 6 中的 0xFC00）开始的表格中显示的即为配置参数的信息。该地址信息可修改。

用户可使用万高科技提供的多功能电能表参数配置文件 **function.ini**，也可根据需求按 **function.ini** 文件的规范编写.ini 文件。



图 6 参数配置

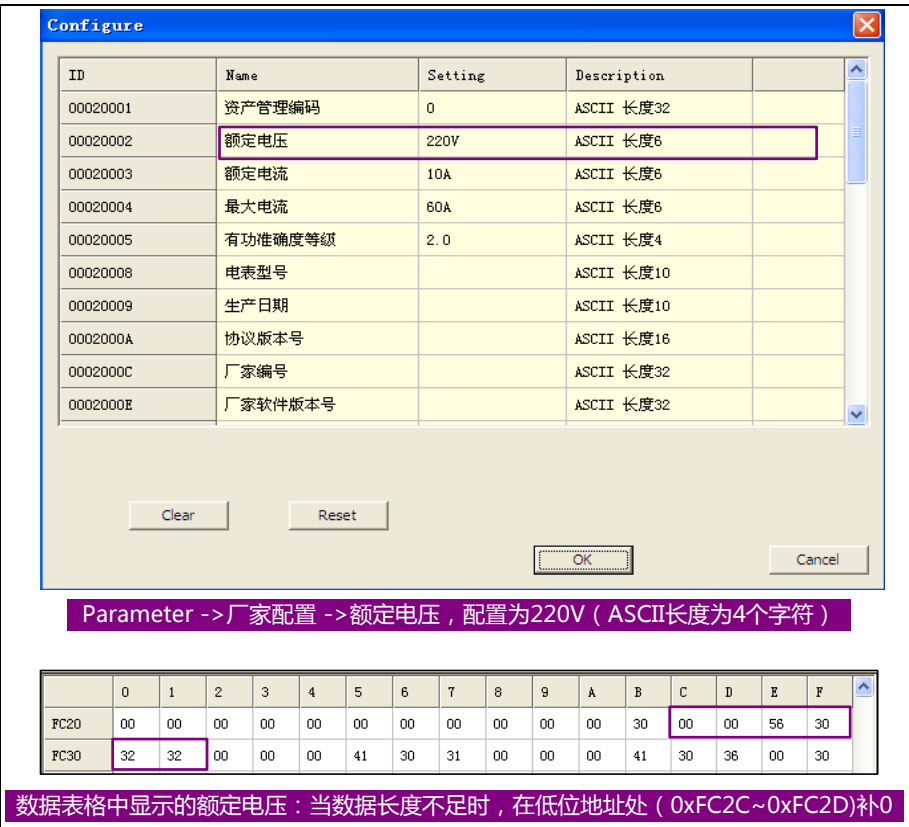


图 7 Parameter →ASCII 码显示补 0 说明

注意：ASCII 设置项中，输入@符号代表空格，如图 8 所示。

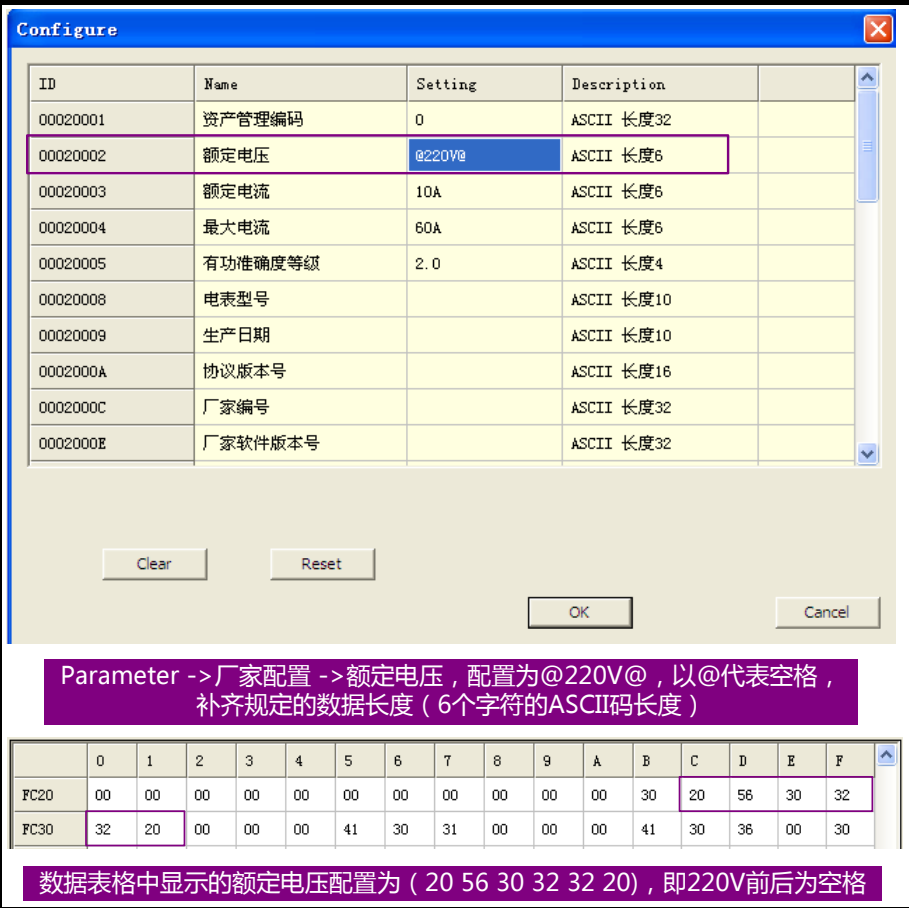


图 8 Parameter →ASCII 码空格输入/显示说明

3.5 页操作

HexWriter_SD502 支持按页操作 Flash 存储内容，每页最多存放 512 字节数据。以 V9401 为例，32KB 的 Flash 可被分为 64 页，分别命名为 0 页到 63 页。

表格 1 V9401 的 Flash 页对应的起止地址

页码	Flash 开始地址（hexadecimal）	Flash 结束地址（hexadecimal）
0	0000	01FF
1	0200	03FF
2	0400	05FF
3	0600	07FF
...
60	7800	79FF
61	7A00	7BFF
62	7C00	7DFF
63	7E00	7FFF

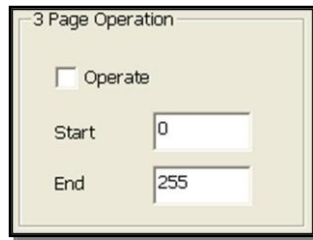


图 9 Page Operation（页操作）

如图 9 所示，如果已勾选 **Operate**，并设置 **Start**（起始页）和 **End**（末页）信息，用户将操作相应范围内的 Flash 内容。页操作时，实际操作的是起始页 Flash 开始地址到末页的前一页的 Flash 结束地址，例如，起始页设为“2”，末页设为“60”，则，

1. 如果点击 **Erase**，则仅擦除目标板中 0400h~77FFh 范围内的 Flash 内容；
2. 如果点击 **AutoLoad** 或 **OffLoad**，则数据表格中 0400h~77FFh 范围内的内容被烧写入目标板，或下载至 SD502 中；
3. 如果点击 **ReadFlash**，则目标板 0400h~77FFh 范围内的 Flash 内容会显示在数据表格中；
4. 如果点击 **Save**，则数据表格中 0400h~77FFh 范围内的内容被保存为.txt 文件。

未勾选 **Operate** 时，用户将操作目标芯片 Flash 的所有地址，即，此时，

1. 如果点击 **Erase**，则擦除目标板全部 Flash 内容；
2. 如果点击 **AutoLoad** 或 **OffLoad**，则数据表格中所有内容均被烧写入目标板，或下载至 SD502；
3. 如果点击 **ReadFlash**，则读取目标板 Flash 中全部内容并显示在数据表格中；
4. 如果点击 **Save**，则数据表格中所有内容均保存为.txt 文件。

如果退出 HexWriter_SD502 前未取消页操作，也未将起止页码恢复为原来的值，则下次启动 HexWriter_SD502 时，页操作对话框仍显示该设置。

按页进行自动或脱机下载数据表格内容时，HexWriter_SD502 不再支持 **CRC Verification**（CRC 校验）功能。如图 9 和图 10 所示，如果同时勾选 Operate 和 SetCRC，则系统会报警 **“In Page Operation mode, CRC Verification is not available. Do you want to download the unverified data?”**（页操作时，CRC 校验功能不能用，请确定是否需要下载？）此时，如果选是，则系统自动取消 CRC 检验功能，继续下载操作；如果选否，则需要重新操作。

3.6 CRC 校验

除了密码保护字节（0x0400）或该字节所在页和 Flash 存储器的最后 6 或 8 个字节外，所有的 Flash 内容均通过 16-bit 的 CRC 码进行校验，在上位机和下位机上，均采用查表法进行校验，详见附录_CRC 校验。

V98xx/V99xx: 除了 0x0400 所在的页和 Flash 存储器最后 6 个字节外，所有的 Flash 内容均需要进行 CRC-16 校验；

V9401/V9011/V9401A: 除了 0x0400 和 Flash 存储器最后 6 个字节外，所有的 Flash 内容均需要进行 CRC-16 校验；

V9003/V9103: 除了 0x0400、Flash 存储器最后 8 个字节外，所有的 Flash 内容均需要进行 CRC-16 校验。0xFFFF8~0xFFFFD 用于存储 CRC 值和 CRC 长度；0xFFFFE~0xFFFF 预存了芯片相关的信息，用户不可进行操作。

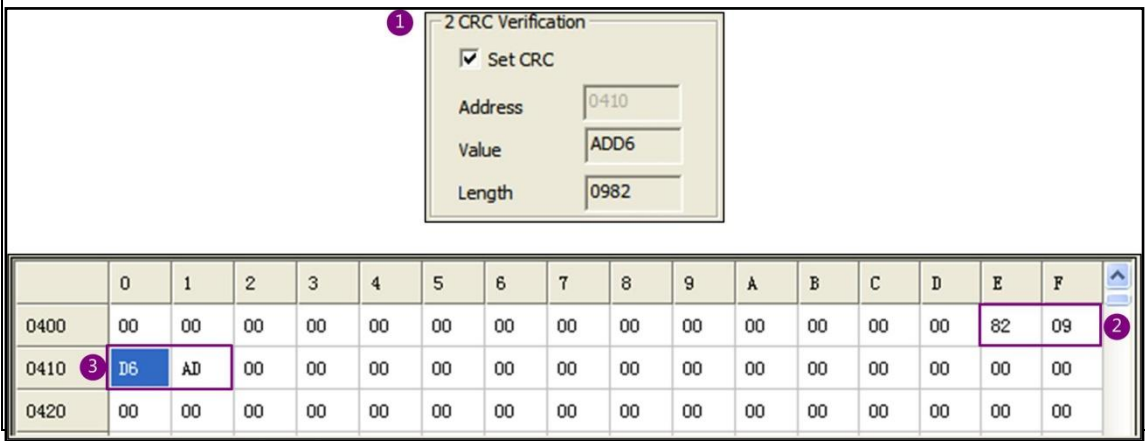


图 10 CRC Verification（CRC 校验）

用户可根据需要选择是否采用 CRC-16 进行 Flash 内容校验。正确连接 SD502 后，勾选 **Set CRC**，再打开 hex 文件或.ini 文件后，图 10 所示的 CRC 校验对话框 ① 会显示相应的 CRC 检验相关信息，其中，

1. **Address**: 存储 CRC 校验值的起始地址。
2. **Value**: 根据 CRC 长度（Length）计算得到的 CRC 校验值，将存于 **Address** 及后一地址处，以小端模式存储。烧写程序时，这项数据会烧写到目标板 Flash 的相应地址中。打开 Hex 文件或.ini 文件后该值不会直接显示在**数据表格**中，只有将**数据表格**中的内容全部正确烧写入目标板 Flash 中，再读 Flash 时，**数据表格**的相应位置才会显示 CRC 值。图 10 中的 ② 为显示在**数据表格**中的 CRC 值。
3. **Length**（CRC 长度）: 所选芯片的 Flash 总容量减去六个字节，将存于 **Address** 前四个地址处，以小端模式存储。烧写程序时，这项数据会烧写到目标板 Flash 中。打开 Hex 文件时该值不会直接显示在**数据表格**中，只有将该 Hex 文件全部正确烧写入目标板 Flash 中，再读 Flash 时，**数据表格**的相应位置才会显示 Length。图 10 中的 ③ 为显示在**数据表格**中的 Length。


CRC 校验相关的信息只能存于目标板 Flash 的最后几个字节中，如表格 2 所示，如果打开的 Hex 文件占用了 CRC 校验信息存储地址，系统会报错 hex 超界。

表格 2 各产品对应的 CRC 校验范围和 CRC 值及 CRC 长度的存储地址

型号	Value 存储地址		Length 存储地址		不参与 CRC 校验的字节
	起始地址	字节数	起始地址	字节数	
V98xx/V99xx	0x1FFFE	2	0x1FFFA	4	0x0400~0x05FF, 0x1FFFA~0x1FFFF
V9401/V9011	0x7FFE	2	0x7FFA	4	0x0400, 0x7FFA~0x7FFF
V9401A	0x5FFE	2	0x5FFA	4	0x0400, 0x5FFA~0x5FFF
V9003/V9103	0xFFFC	2	0xFFF8	4	0x0400, 0xFFF8~0xFFFF

如果未正确连接 SD502 或 SD502 即打开 Hex 文件，则系统会报错 USB 通讯错误。按页进行自动或脱机下载数据表格内容时，HexWriter_SD502 不再支持 CRC 校验功能。如图 9 和图 10 所示，如果同时勾选 **Operate** 和 **SetCRC**，则系统会报警 “**In Page Operation mode, CRC Verification is not available. Do you want to download the unverified data?**”（页操作时，**CRC 校验功能不能用，请确定是否需要下载？**）此时，选**是**，则系统自动取消 CRC 检验功能，继续下载操作；选**否**，则需要重新操作。

3.7 加密设置

 电能计量 SoC 芯片片上 Flash 的 0x0400 字节的 bit0 为程序加密位。设置该加密位，会影响 SD502 对目标板 Flash 的 ISP 操作。

在 HexWriter_SD502 上，用户可在 **Encryption** 框内勾选 **Encrypt** 操作程序加密位，并自行设置加密的范围，从而影响 JTAG 对目标板 Flash 的访问。

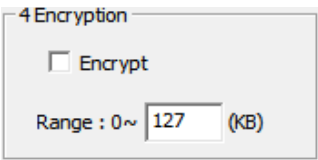


图 11 Encryption（加密设置）

自动下载或脱机下载时，勾选 **Encrypt**，通过 JTAG 对目标板 Flash 全地址范围进行擦除和写操作不受限制，仅影响读操作和页擦除。加密对 ISP 的影响详见表格 3。

自动下载或脱机下载时，不勾选 **Encrypt**，通过 JTAG 可对目标板 Flash 进行无限制访问。

表格 3 读加密对 ISP 操作的影响

目标板型号	Flash 地址	读加密			
		全擦除	页擦除	写操作	读操作
V98xx/V99xx	0x00000~0x17FFF	√	×	√	×
	0x18000~0x1FFFF	√	×	√	√
V9011/V9401	0x0000~0x5FFF	√	×	√	×
	0x6000~0x7FFF	√	×	√	√
V9401A	0x0000~0x3FFF	√	×	√	×
	0x4000~0x5FFF	√	×	√	√
V9003/V9103	0x0000~0xBFFF	√	×	√	×
	0xC000~0xFFFF	√	×	√	√

注意：勾选读加密进行自动下载或脱机下载后，点击 **Erase** 对目标板 Flash 进行全擦除，则 Flash 的 0x0400~0x0600 范围内的信息会丢失。

3.8 代码长度

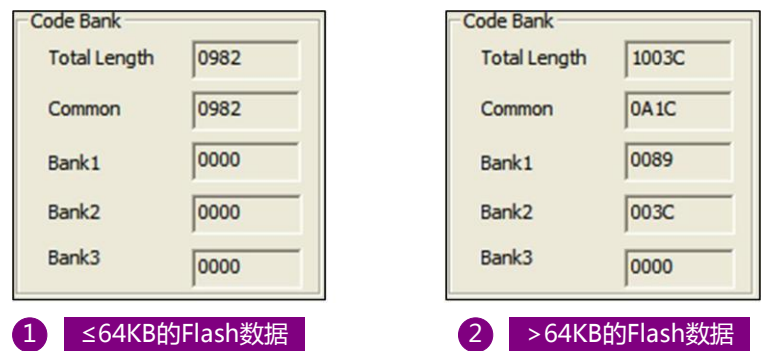


图 12 Code Bank（代码长度）

图 12 所示为 HexWriter_SD502 操作界面上的 **CodeBank**（代码长度）设置区。当打开一个 hex 文件时，**Total Length** 显示该文件的代码总长度。

图 12 的①所示，如果打开的 hex 文件为一个不大于 64KB 代码，则代码总长度与 COMMON 的数值相同。

图 12 的②所示，如果打开的 hex 文件为一个大于 64KB 代码，则 Bank1/Bank2/Bank3 显示代码分页后每个代码段的大小。

3.9 在线编程操作

Online Operation（在线编程操作）包括**自动下载（Auto Download）**和**擦除芯片（Erase Chip）**两种操作。如图 13 所示，在 HexWriter_SD502 操作界面上，用户可以通过菜单栏和快速工具栏进行相应的操作。

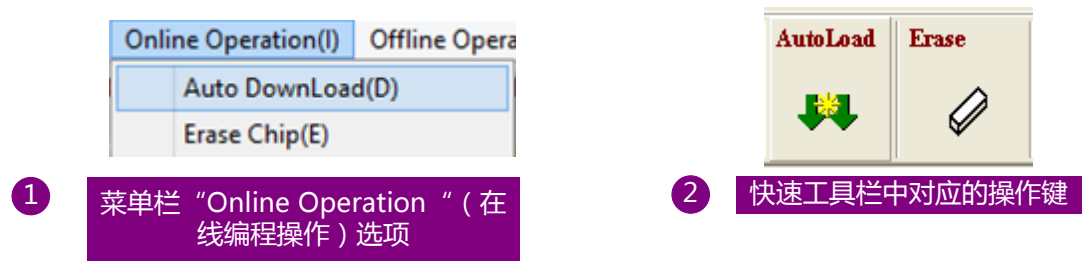


图 13 Online Operation（在线编程操作）

- 1. **Auto Download（自动下载）**：根据用户设置的页操作起止地址，擦除芯片内部相应范围的 Flash 数据，提取**数据表格**中相应范围内的数据，将其烧写入芯片 Flash 并验证芯片 Flash 内容。如果用户未设置页操作，则，点击 **AutoLoad** 即可完成以下操作：擦除芯片内部所有的 Flash 数据，提取**数据表格**中所有数据，并将其烧写入目标板并验证 Flash 内容。

自动下载操作的是**数据表格**中的内容。如果数据表格中存在不需要被下载的内容，那么，用户应重新打开 Hex 文件及/或.ini 文件。

- 2. **Erase Chip（擦除芯片）**：根据用户设置的页操作起止地址，擦除芯片内部相应范围的 Flash 内容。如果用户未设置页操作，则，点击 **Erase** 将擦除 Flash 内存储的所有数据。

注意：

对 V98xx/V99xx 的 Flash 进行擦除操作时，0x0400~0x05FF 范围内的数据不能被擦除，而且 0x0000~0x0008 内的数据被更新为 0x020000020000020000；

对其它芯片的 Flash 进行擦除操作时，除了 0x0000~0x0008 内的数据被更新为 0x020000020000020000，其它数据均会被擦除。

3.10 脱机烧写操作

脱机烧写操作的过程如下：根据用户设置的页操作起止地址，提取**数据表格**中相应范围内的数据，并将其脱机下载至 SD502 中，再通过 SD502 上的 **Start** 按钮，将上述数据烧写入目标板 Flash 中，操作流程详见“SD502 用户手册”。

如图 14 所示，在 HexWriter_SD502 操作界面上，用户可以通过菜单栏 **Offline Operation→Download** 和快速工具栏 **OffLoad** 进行脱机烧写操作。

如果用户未设置页操作，则脱机烧写操作将**数据表格**中的所有内容烧写入目标板。



图 14 Offline Operation（脱机烧写操作）

脱机下载操作的是**数据表格**中的内容。如果**数据表格**中存在不需要被下载的内容，那么，用户应根据需要设置页操作或重新打开 Hex 文件及/.ini 文件。

3.11 Flash 脱机烧写功能配置

点击 **Offline Operation→Setting** 进入 **Special settings** 对话框，配置脱机烧写功能，页面如图 15：有 5 个配置项目。

注意：只要点击 **Set**，SD502 中存储的目标芯片的 Hex 文件就会被销毁，此时，如果直接复位 SD502，SD502 无法通过自检。所以，完成 **Special settings** 的配置后，一定要重新下载 Hex 文件。如果没有用于当前生产的 Hex 文件时，用户不应点击该对话框的 **Set** 按键，但是可以点击 **Read** 按键读取当前 SD502 的配置信息。

1. 用户应根据以下步骤进行 **Special settings** 对话框的配置：点击 **Read**，读取 SD502 的配置值，包括烧写数量控制、自动连续烧写、调试频率切换和 RTC 秒脉冲校正、RTC 时间设置和晶体停振检测等信息；
2. 根据实际需要开启特殊功能：
 - 1) 开启烧写数量控制：勾选 **Quantity control**，设定 SD502 单次烧写的起始数量 **StartingNumber** 和结束数量 **Ending Number**；
 - 2) 开启自动连续烧写：勾选 **Continuous Programming**；

- 3) 选择调试频率：800k 或 3.2M；
 - 4) RTC 控制: Flash 脱机烧写 (**Download**)、RTC 秒脉冲校正 (**RTC Cal**) 和 RTC 时间设置 (**Set Time**)，以及以上操作的随意组合；
 - 5) 开启晶振停振检测：勾选 **Check Crystal**；
3. 点击 **Set** 将上述修改过的配置信息写入 SD502。
- 1) 如果 HexWriter_SD502 提示写入成功，那么重新点击 **Read**，读出新的 SD502 的配置值，检查配置是否正确。应保证上述配置均已完成；
 - 2) 如果 HexWriter_SD502 提示写入失败，建议重新连接 SD502 和 PC，再重复 1~2 步骤操作，直至 HexWriter_SD502 提示写入成功。
4. 退出 **Special settings** 对话框；
5. 在 HexWriter_SD502 主界面上，点击 **OffLoad** 将 Hex 文件下载至 SD502 中；
6. 复位 SD502，如果自检通过，表明 SD502 可以正常工作。

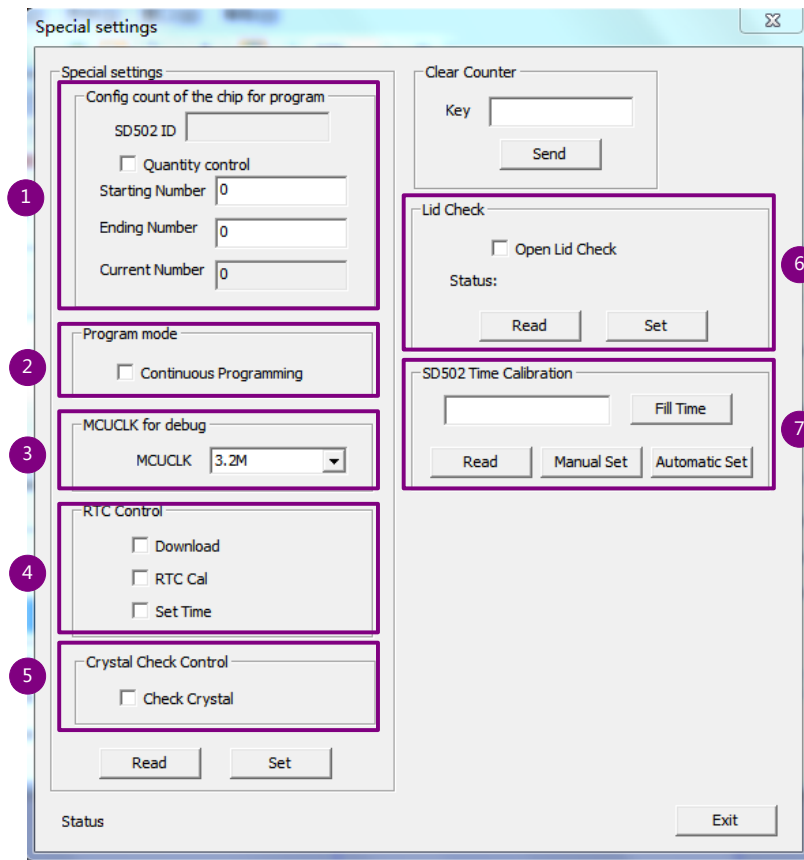


图 15 脱机烧写功能配置操作

3.11.1 烧写数量控制

每次完成烧写数据控制的设置后，如果当前烧写数量小于起始数量，当前数量变化，会和起始数量保持同一个数。设置时候一定要用户保证，起始数量大于等于当前数量。用户可以点击 **Read** 再次读取 SD502 的配置参数进行查看。系统默认不限制烧写数量。

烧写次数的统计方法:

1. 手动或者自动方式启动开始烧写;
2. 烧写成功后, 烧写数量加 1, 如果烧写失败, 烧写数量不累加。一次烧写成功后, 如果 SD502 和目标板一直处于连接状态, 那么 SD502 不会对该目标板自动进行二次烧写。此时, 如果需要对目标板进行第二次烧写, 用户可以按 SD502 的 **Start** 按钮启动第二次脱机烧写准备。

重复 1) ~2), 直到当前烧写数量达到允许烧写的最大数量。达到允许烧写的最大数量后, 会自动将 SD502 内部存储的用户 hex 销毁。如果需要再次脱机烧写, 需要修改烧写数量控制配置参数并重新加载用户 Hex 文件。

3.11.2 自动连续烧写

开启自动连续烧写功能时, 按下 SD502 的 **Start** 按钮, SD502 能连续自动检测目标板的连接状态并向目标板下载 Hex 文件, 连续两次烧写操作之间无需再按 **Start** 按钮。

自动连续烧写有个特殊配置: 合盖检测功能。开启方法: 连接 SD502 与 PC, 打开 hexwriter_SD502 进入 **Special settings** 对话框, 勾选 **Open Lid Check**, 点击 **set** 按钮, **status** 提示成功或失败。

关于自动连续烧写, 详见“SD502 用户手册”。

3.11.3 调试频率切换

如果需要调整调试频率, 可以通过 **MCUCLK** 选项进行选择, 目前支持两种 MCU 时钟频率 (800k 和 3.2M)。在选择调试频率时, 应注意以下几点:

1. 如果目标芯片 MCU 时钟频率为 3.2M 及以上, 应将 **MCUCLK** 设为 3.2M;
2. 如果目标芯片 MCU 时钟频率为 800k~3.2M 之间, 应将 **MCUCLK** 设为 800k;
3. 如果目标芯片实际的 MCU 时钟频率为 3.2M 及以上, 但是 **MCUCLK** 设为 800k, 那么, 用户能正常在线调试, 但是调试速度会降低;
4. 如果目标芯片实际的 MCU 时钟频率为 800k~3.2M 之间, 但是 **MCUCLK** 设为 3.2M, 则很可能不能正常进行在线调试。
5. 如果目标芯片 MCU 实际的时钟频率为 800k 以下, 那么无论 **MCUCLK** 设为何值都可能不能正常进行在线调试。

3.11.4 RTC 秒脉冲校正控制

当 SD502 对的电能表进行 RTC 秒脉冲校正时, 必须开启 RTC 秒脉冲校正功能, 即勾选 **RTC Cal**。

开启 RTC 秒脉冲校正功能时, 用户需要设置 RTC 秒脉冲校正相关的信息, 详见 RTC 相关。

RTC 秒脉冲校正步骤, 详见“SD502 用户手册”。

3.11.5 晶体停振检测

晶体停振检测功能可以开启或关闭。如果开启该功能且晶体检测结果为晶振停振, 那么与芯片相关的所有操作都会失败。修复晶振问题后, 才能对芯片进行正常操作。如果目标板方案为无晶振方案, 就不要勾选晶体停振检测项目, 否则无法调试和下载。

3.11.6 SD502 时间校正

正确连接 PC 和 SD502, 用户可以手动或自动对 SD502 的时间进行校正。

- 1) 手动设置: 点击 **Fill Time**(读取 PC 当前的时间)或 **Read**(读取 SD502 的时间), 在读取的时间信息中手动修改, 再点击 **Manual Set**, 左下角提示 “Manually set time success” 即设置成功。
- 2) 自动设置: 点击 **Automatic Set**, 将 PC 当前的时间直接写到 SD502 中, 提示“Automaticlly set time success” 即设置成功。

操作完成后可再次读取 SD502 中的时间检验是否完成正确校正。

3.12 配置信息保存

以上所有配置完成后, 点击 **SaveCfg** 按钮, 保存所有配置到用户选中的路径的 ini 文件中。

3.13 Hex 加密烧写操作

由于某些应用中, 客户希望对 hex 进行加密, 使用加密的 hex 文件进行传输, 这样可以防止他人盗用。如何对 hex 进行加密?

3.13.1 Hex 加密

The screenshot shows the 'Encryption' dialog box. It has a yellow title bar with a red close button. The dialog is divided into several sections:

- Hex File Info:** Contains a '.hex Editor' dropdown menu currently set to 'KEIL' and a 'Select Hex file' button.
- Setting Number Info:** Contains a 'Quantity control' checkbox (unchecked), 'Starting Number' and 'Ending Number' input fields, both containing the value '0'.
- Get Current Number:** Contains a 'Current Number' input field and a 'Send' button.
- ID/Key Operation:** Contains a 'Select key Folder' button, a 'Get ID' button, and a 'Generate key file' button.
- Bottom Buttons:** 'Generate cipher file' and 'Cancel' buttons.

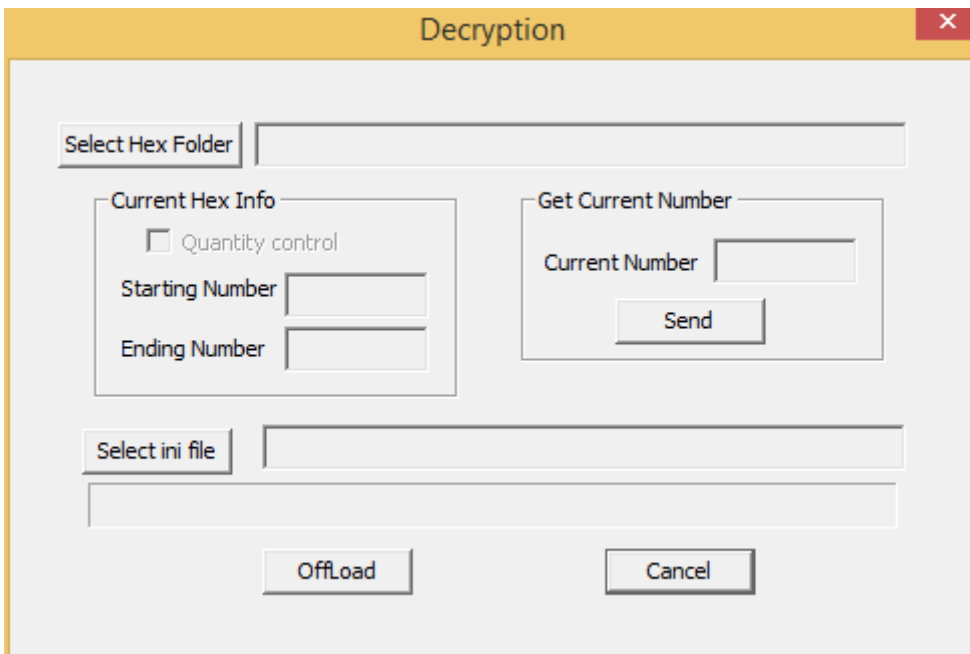
实现功能:

将编译器生成的 hex 文件、烧写数量配置、批量密钥文件通过一定的算法生成批量的密文文件。可以将这些密文文件发送给生产方进行生产。

操作步骤:

- 1) 选择 hex 是使用哪种编译器软件进行编译的;
- 2) 点击 “Select Hex file “选择 hex 文件;
- 3) 选中或不选 “Quantity control “，控制是否进行烧写数量控制;
- 4) 如果需要烧写数量控制，需要填写烧写起始数量 (Starting Number) 和结束数量 (Ending Number)。需要注意的是，起始数量一定要大于等于当前数量 (Current Number)，才能保证烧写总数量符合要求。如果不知道当前数量的话，可以通过连接 SD502，点击 “Send “进行获取。
- 5) 选择密钥文件。直接点击选择密钥文件 “Select key files “，选择密钥文件存在的文件夹。
该文件夹下有多个密钥文件就会相应生成多个密文文件，支持多个密钥产生多个密文，即批量产生密文。
如果当前没有密钥文件，通过以下途径制作密钥文件：
直接将 SD502 的 ID 号填写入 ID 编辑框内或通过 USB 接口获取 SD502 的 ID 号 (点击 “Get ID “按钮)，点击 “Generate key file “，选择存储路径和文件名，生成密钥文件。
- 6) 点击 “Generate cipher file “，选择生成密文的路径和输入密文文件名，生成最终批量密文文件。

3.13.2 Hex 脱机导入 SD502



实现功能:

将客户发送过来的批量密文文件和配置文件导入对应的 SD502，用于之后的脱机批量烧写。SD502 会自行选择适用于自身的密文文件。此过程中也会显示一些信息，比如烧写起始数量和结束数量，以及是否进行烧写数量控制。请仔细核对，特别对于需要烧写数量控制部分，一定要使起始数量 (Starting Number) 大于等于当前数量 (Current number)，并保证结束数量 (Ending Number) - 起始数量 (Starting Number) \geq 对方要求的烧写总数量。

步骤:

1. 点击 “Select Hex Folder”，选择批量密文存在的文件夹;
2. 点击 “Select ini file”，选择 ini 配置文件;

3. 点击“OffLoad”，将以上信息烧写入 SD502；
4. 核对烧写数量控制配置，烧写起始数量和结束数量，以及读取当前数量。如果有问题，与 Hex 来源方沟通；没有问题的话，可以复位 SD502 后，批量烧写芯片。

3.14 读 Flash

ReadFlash 是指，根据用户设置的页操作起止地址，读取与 SD502 连接的目标板 Flash 中相应范围内的数据，并按地址顺序显示在数据表格中。如果未设置页操作，点击 **ReadFlash** 将直接读取目标板 Flash 的全部内容；如图 16 所示，在 HexWriter_SD502 操作界面上，用户可以通过菜单栏和快速工具栏进行读 Flash 操作。

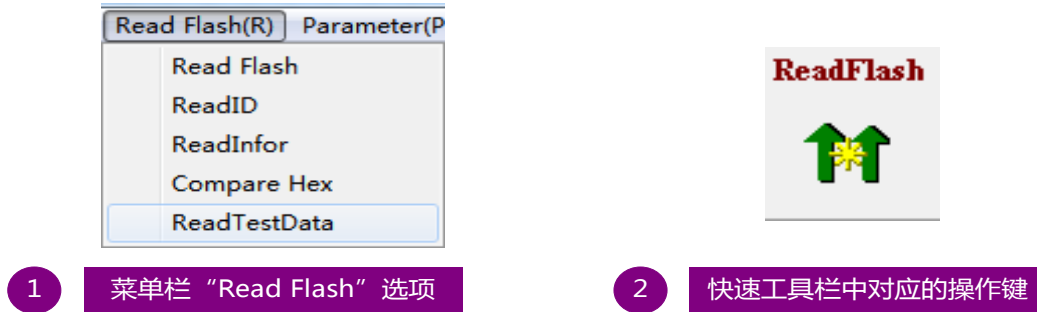


图 16 ReadFlash（读 Flash）

3.14.1 对比 Hex 码

这个功能需要勾选 CRC 功能。它主要判断：

1. 对比 SD502 中的 hex 码和 Hexwriter 中装载的 hex 码是否一致（需要连接 SD502 并且 HexWriter 要装载 Hex 码）；
2. 对比 SD502 中的 hex 码和芯片中的 Hex 码是否一致（需要连接 SD502 和芯片）；
3. 对比 Hexwriter 中装载的 hex 码和芯片中 Hex 码是否一致（HexWriter 要装载 Hex 码并且需要连接芯片）。

点击 Read Flash→Compare Hex 进入 Compare Hex 对话框，页面如图 17：

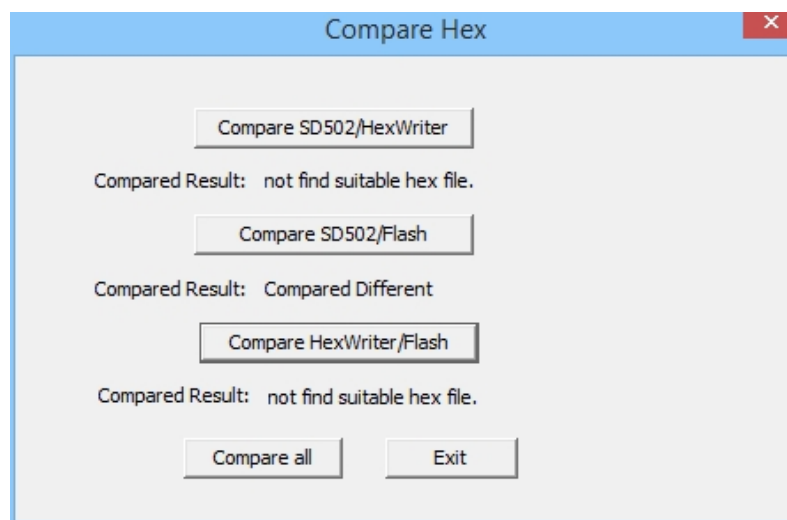


图 17 Compare Hex（对比 Hex）

如果进入 Compare Hex 对话框之前弹出提示 “not find suitable hex file”，则表示 HexWriter 没有加载 Hex 码。图 18 显示 HexWriter 加载 Hex 码的界面（加载之前自行配置相关信息）：

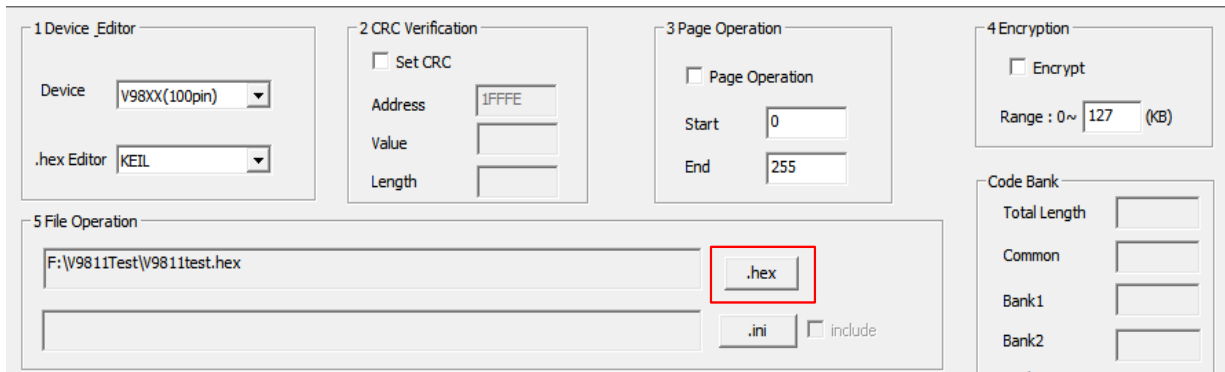


图 18 HexW 加载 Hex 界面图

点击 “Compare SD502/HexWriter”，将会在紧接着的 “Compared Result” 右侧显示对比结果，有 4 种结果：

1. not find suitable hex file. （HexWriter 没有加载 Hex 码时出现）
2. USB communication failed. （SD502/芯片连接失败时出现）
3. ComparedSame(对比结果一致)
4. ComparedDifferent(对比结果不一致)

点击 “Compare SD502/Falsh”，将会在紧接着的 “Compared Result” 右侧显示对比结果，有 5 种结果：

1. USB communication failed. （SD502/芯片连接失败时出现）
2. No connection to target board. （无法连接目标芯片时出现）
3. Initializing target board failed. （无法初始化目标芯片时出现）
4. ComparedSame (对比一致)
5. ComparedDifferent (对比不一致)

点击 “Compare HexWriter/Flash”，将会在紧接着的 “Compared Result” 右侧显示对比结果，有 5 种结果：

1. not find suitable hex file. （HexWriter 没有加载 Hex 码时出现）
2. USB communication failed. （SD502/芯片连接失败时出现）
3. No connection to target board. （无法连接目标芯片时出现）
4. Initializing target board failed. （无法初始化目标芯片时出现）
5. ComparedSame (对比一致)
6. ComparedDifferent (对比不一致)

点击 “Compare all”，将会在上述三个 “Compared Result” 右侧显示对比结果，显示的结果为上述各种结果的可能组合。

3.14.2 烧写数量统计

1. 进入 ReadTestData 子界面:

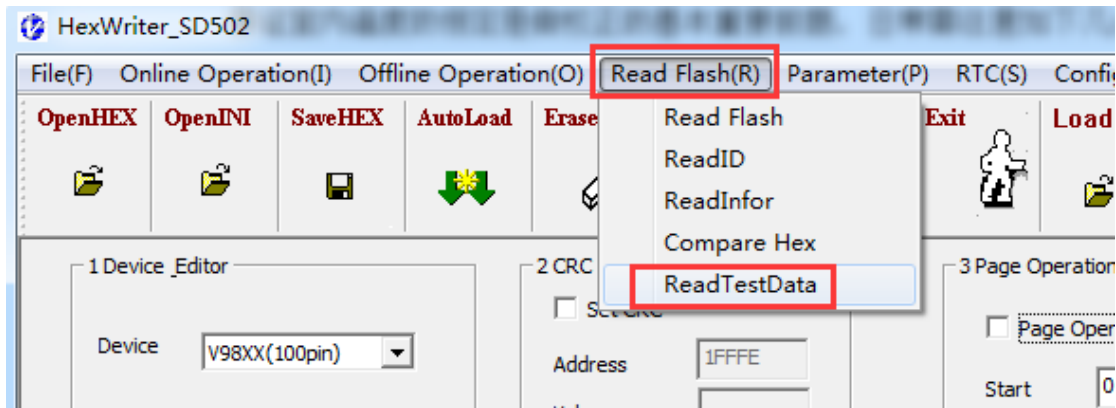


图 19 烧写数量统计子界面进入图

2. 子界面主要分成三部分：SD502 访问区、本地文件访问区以及信息展示区。

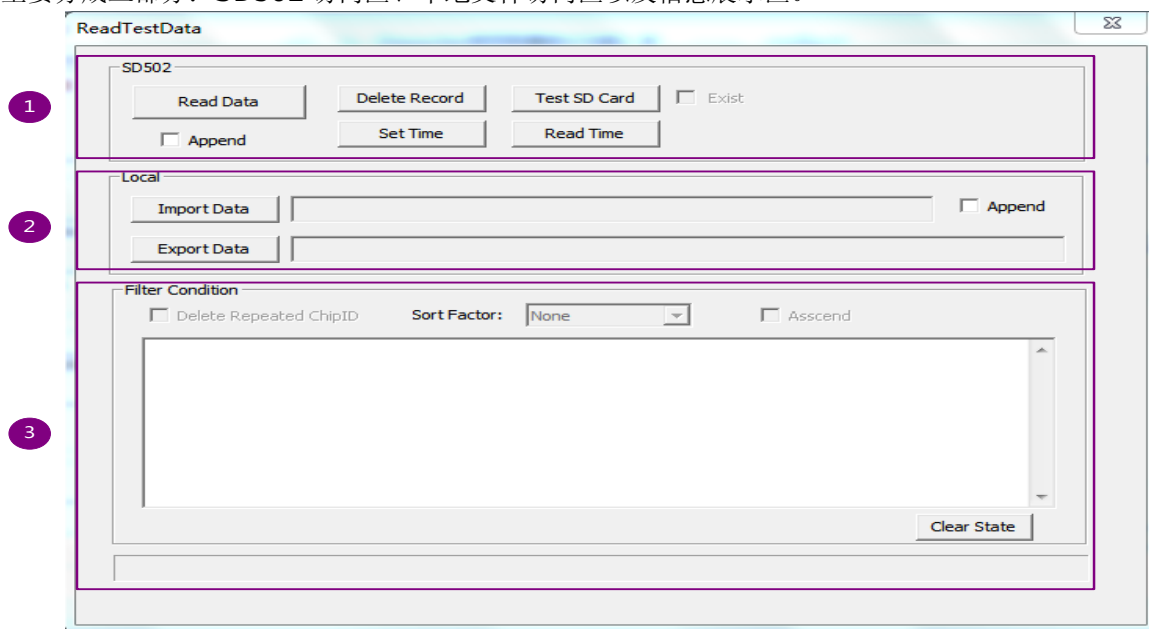


图 20“ReadTestData”界面显示图

- 1) 操作 SD502 访问区（此操作需要连接芯片）

SD502 内部会有 SD 卡，SD502 会根据测试情况生成记录信息。记录信息有四列，分别表示 SD502 的 ID 号、芯片烧写时间、芯片 ID 号和芯片烧写结果（0 表示成功，非 0 表示失败）。

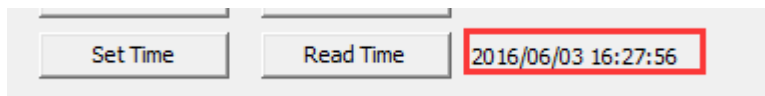
Read Data 功能：读取 SD 卡的记录。此功能会根据 Append 是否选中来确定如何读取 SD 卡记录。如果选中 Append 复选框，新读取的数据会追加到以前读取的数据，反之，新读取的数据会覆盖以前读取的数据。

Delete Record 功能：清除 SD 卡的记录。

Test SD Card 功能：测试是否存在 SD 卡，测试状态会显示到右侧 Exist 中。

Set Time 功能：设置 SD502 内部时间。设置的时间为当前电脑上的时间。这个时间会显示到右侧。

Read Time 功能：获取 SD502 内部时间。获取的时间会显示到右侧。



2) 本地文件访问区

Import Data 功能：导入 csv 格式的数据。支持读取文件夹。如果正在读取的 csv 文件的记录格式不合法，将会舍弃当前正在读取的 csv 文件，继续读取下一个文件，直到读完当前文件夹中的 csv 文件。此功能会根据 Append 是否选中来确定如何读取 csv 记录。如果选中 Append 复选框，新读取的数据会追加到以前读取的数据，反之，新读取的数据会覆盖以前读取的数据。

Export Data 功能：导出 csv 格式的数据。此功能会导出通过 Import Date 功能或者 Read Data 功能读取的数据。选择需要保存的 csv 路径，此路径用于保存完整的记录信息。此时会生成另外一个用于保存去掉重复数据的文件，这个文件名和用户选择的文件名只多了一个_uniquechipID 后缀。此时会生成一个 summary.txt 文件，summary 文件用于存储信息展示区中的信息显示框中的概要信息。

3) 信息展示区

Delete Repeated ChipID 复选框：去掉重复数据。暂未使用。

Sort Factor 下拉框：构建排序规则。暂未使用。

Asscend 复选框：是否按照升序排序。暂未使用。

信息显示框：显示所有操作产生的概要信息。

Clear State：清空显示框中的信息。

滚动条：显示当前操作状态的进展情况。在需要时间的操作中（如 Read Data、Import Data、Export Data），此滚动条会动态的表示操作情况。

3. 注意事项

- 1) 批量生产前确认 SD 卡是否存在，是否读写正常。通过点击 ReadFlash，打开 ReadTestData 对话框，点击 Test SD Card，查看返回状态。一般不配 SD 卡，如果 SD 卡没有或已损坏，可以自行采购。
- 2) 批量生产前确认是否将历史数据清除干净。在 ReadTestData 对话框中，点击 Delete Record，清除 SD 中的烧写数量记录。
- 3) 批量烧写前确认 SD502 中的时间是不是正确，通过 ReadTime 确认。如果不准确可以通过 Set Time 进行校正，校正到当前 PC 时间。如果发生掉电后，时间丢失的话，可能是电池没有电了，可以打开盒子，更换电池。
- 4) 批量烧写完成后，读取某位员工使用的 SD502 中的测试记录，如果使用多个 SD502，可以通过勾选追加将多个 SD502 的数据进行整体统计。读 SD502 中测试记录方法：勾选追加选项 APPEND，点击 ReadData。读取完毕后，信息对话框中会显示相应统计信息。有四列信息，分别是 SD502 的 ID 号，芯片烧写时间，芯片的 ID 号，芯片烧写结果（0 表示烧写成功，非 0 表示烧写失败）。

3.15 RTC 相关

RTC 设置仅用于对基于 V98xx 和 V99xx 的电表进行 RTC 秒脉冲校正和 RTC 时间设置。

在 HexWriter_SD502 操作界面上，用户可通过点击菜单栏中的 **RTC 进入 RTC settings** 对话框进行 RTC 秒脉冲校正相关设置，设置界面如 21 所示。

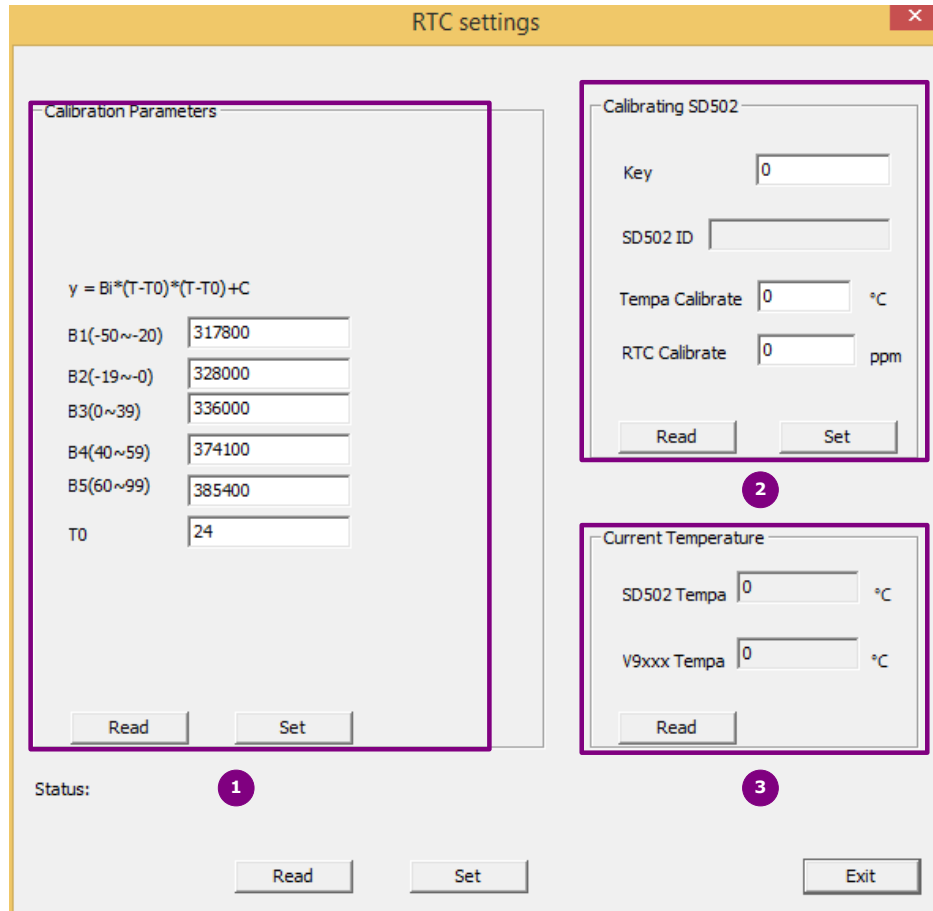


图 21 RTC setting (RTC 设置)

注意：RTC 配置统一采用 **ADD33** 进行校验。

① 温度曲线系数设置和晶体参数设置：

- ParaA~ParaE 为目标芯片的温度 AD 曲线系数，V98xx/V99xx 采用同样的温度 AD 曲线系数，该数据由万高科技提供；
- B1~B5 为晶体在不同温度范围内的二次校准系数，由晶体决定。如果使用万高科技提供的晶体，必须采用万高科技提供的数据；
- T0 为晶体的顶点温度，由晶体决定。如果使用万高科技提供的晶体，必须采用万高科技提供的数据；

完成上述信息配置后，点击 **Set**，将配置写入 SD502。点击 **Read**，用户可读取保存在 SD502 中温度曲线系数和晶体参数。

② 校正 SD502：校正 SD502 自身的系统偏差，出厂前已完成校正，以后每年返厂校正一次。已加密，用户无法操作。

③ 点击 **Read** 读取当前 SD502 温度和目标板温度：

- SD502 温度：因为 SD502 内部 DC-DC 发热较为厉害，所以它的内部温度一般比环境温度要高；
- 目标板温度：连接 SD502、PC 和目标板，点击 **Read**，读取目标板温度。如果目标板温度与环境温度差别较大，可能是因为目标板没有经过温度校正，或者校正有问题。

3.16 帮助

在 HexWriter_SD502 操作界面上,用户可点击菜单栏或快速工具栏中的 **Help** 和 **About?** 按键进入 **About** 对话框, 如图 20 所示。

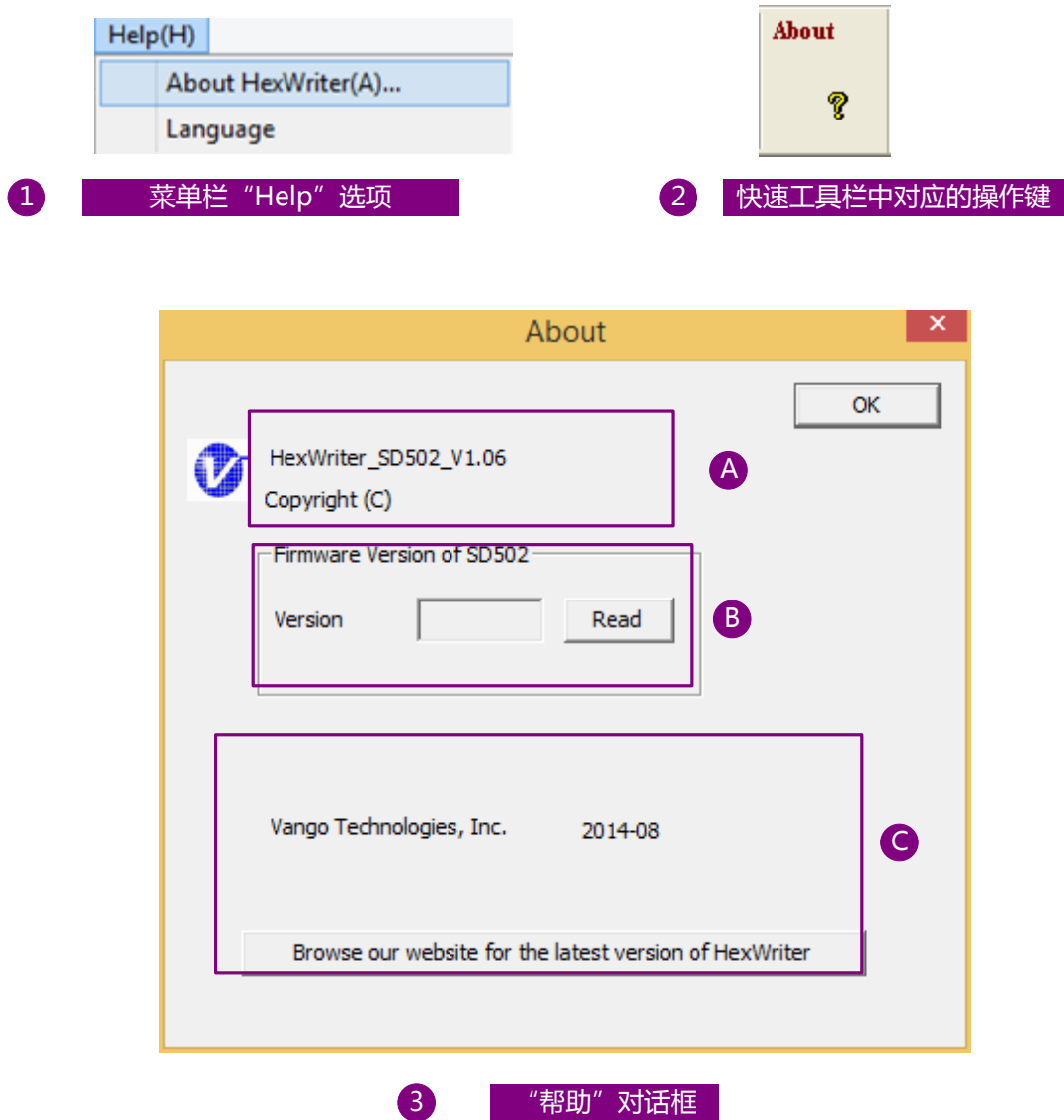


图 22 Help（帮助）：读取固件版本号，获取客户支持信息

如图 20 中的对话框所示，

Ⓐ 为 HexWriter_SD502 软件的版本号；

Ⓑ 点击读版本按键 **Read**，获得 SD502 的固件版本号；

Ⓒ 显示软件发布（更新）时间以及公司网址。用户可点击按键 **Browse our website for the latest version of HexWriter_SD502** 访问万高科技网站上最新的 HexWriter_SD502 相关信息。

3.17 退出

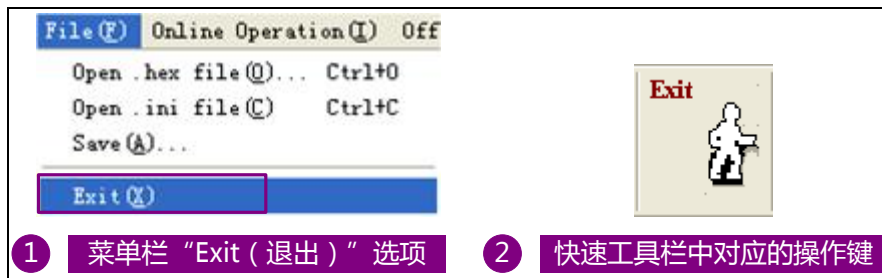


图 23 Exit (退出)

在 HexWriter_SD502 操作界面上，用户可通过菜单栏点击 **File→Exit**，或快速工具栏中的 **Exit** 按钮退出 HexWriter_SD502 程序。这 2 种退出方式可以保存用户的设置。

用户也可以通过点击窗口的关闭按钮退出 HexWriter_SD502 程序。但是这种操作方法无法保存用户的设置。

4. 系统提示说明

1. USB communication failed (USB 通讯错误)

SD502 未与 PC 正确连接时，用户如果在 HexWriter_SD502 界面上进行自动下载、脱机下载、擦除、读 Flash、写 RTC、读固件版本等操作，系统会报错 USB 通讯错误。

在 SD502 未与 PC 正确连接时，如果直接打开参数配置页面进行参数配置，系统也会报错 USB 通讯错误。

2. Address is out of range (Hex 超界)

目标板 Flash 的最后 6 个字节被用于存储 CRC 检验相关的信息。所以，如果打开的 Hex 文件中这几个字节已经被占用，则系统会报错 hex 超界；另外，如果数据表格这几个字节有数据，则点击脱机下载或自动下载时，系统会报错 hex 超界。

3. No connection to target board Or Initializing target board failed. (无法连接目标芯片/无法初始化目标芯片)

出现该提示时，请检查 SD502 与目标板的连接是否正常；如果连接正常，则检查 SD502 上的给目标板供电的电源灯是否常亮；如果电源灯灯灭，则表明目标板有短路。

4. Addresses from 0x0400 to 0x0600 is occupied. The content will not be downloaded to Flash memory. Do you want to continue? (0x400 到 0x600 非空，但是该部分数据不会被下载到 Flash，请确定是否需要下载?)

因为 V98xx/V99xx 的 Flash 存储器的 0x0400~0x05FF 范围内预存了芯片相关的信息，所以，数据表格中该地址范围内不应有应用程序代码。当芯片型号选择为 V98XX 时，如果数据表格中的 0x0400~0x0600 范围内有数据，则点击脱机下载或自动下载时，系统会出现该提示。此时，点击是，则数据表格中该范围内的数据不会被下载到目标板 Flash 或下载器中；点击否，则重新进行操作。

5. In Page Operation mode, CRC Verification is not available. Do you want to download the unverified data? (页操作时，CRC 校验功能不能用，请确定是否需要下载?)

按页进行自动或脱机下载数据表格内容时，HexWriter_SD502 不再支持 CRC 校验功能。如果此时同时勾选页操作和 CRC 校验，则系统会出现该提示。此时，如果选是，则系统自动取消 CRC 检验功能，继续下载操作；如果选否，则需要重新操作。

5. 附录_CRC 校验

5.1 CRC 算法

表格如下：

```
const uint16 code CrcTab[256]={ // X16+X12+X5+1 余式表
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
    0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
    0xdbfd, 0xcdbc, 0xfbff, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
    0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
    0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
    0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
    0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
    0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
    0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
    0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
    0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
    0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
    0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
    0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
    0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
```

```
0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0
};
```

CRC 函数:

```
uint16 do_CRC_root(uint16 initcrc,uint32 nStartAddr,uint32 len)
{
    uint8 data ucData;
    uint16 data crc,nlen;
    uint8 code *point;

    nlen = len;
    point = (uint8 code *)nStartAddr;
    crc = initcrc;

    while(nlen--)
        crc = (crc<< 8) ^ CrcTab[((crc>>8) ^(*point++)) & 0xff];

    return crc;
}
```

5.2 CRC 校验例程

必须为 Windows 32-bit 系统。采用查表法。

以下为 V9401/V9811/V9801 的 CRC 校验例程，具体工程详见学习板工程的 Keil 文件夹中的 V9401CRCtest_keil、V9811CRCtest_keil 和 V9801CRCtest_keil。

5.2.1 CRC 校验例程_V9401

具体工程参见文件：**V9401CRCtest_keil**，主要代码 Func.c 如下：

```
uint16 GetCRCData(void)
{
    uint16 u16Ret;
    uint8 code *point;
    point = (uint8 code *)0x7fff;
    u16Ret = *point;
    u16Ret = u16Ret<<8;
    point = (uint8 code *)0x7ffe;
    u16Ret += *point;
    return u16Ret;
}
```

```

uint32 GetCRCLen(void)
{
    uint32 u32Ret;
    uint8 code *point;
    point = (uint8 code *)0x7ffd;
    u32Ret = *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0x7ffc;
    u32Ret += *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0x7ffb;
    u32Ret += *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0x7ffa;
    u32Ret += *point;
    return u32Ret;
}

uint8 checkCRC(void)
{
    uint16 u16CrcData,u16CalCRCData;
    uint32 u32CrcLen;
    u16CrcData = GetCRCData();
    u32CrcLen = GetCRCLen();
    if(u32CrcLen<0x400)
    {
        u16CalCRCData = do_CRC_root(0x5555,0,u32CrcLen);
    }
    else if(u32CrcLen<0x8000)
    {
        u16CalCRCData = do_CRC_root(0x5555,0,0x400);
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x401,u32CrcLen-0x401);
    }
    else
    {
        return 0;
    }

    if(u16CalCRCData!=u16CrcData)

```

```

{
    return 0;
}
else
{
    return 1;
}
}

```

调用函数（checkCRC）。如果返回 0，表明 CRC 校验失败；返回 1，表明校验成功。

5.2.2 CRC 校验例程_V9811

具体工程参见文件：**V9811CRCtest_keil**，主要代码如下：

```

//放在 root 区
uint16 GetCRCData(void)
{
    uint16 u16Ret;
    uint8 code *point;
    point = (uint8 code *)0xffff;
    u16Ret = *point;
    u16Ret = u16Ret<<8;
    point = (uint8 code *)0xfffe;
    u16Ret += *point;
    return u16Ret;
}

//放在 root 区
uint32 GetCRCLen(void)
{
    uint32 u32Ret;
    uint8 code *point;
    point = (uint8 code *)0xfffd;
    u32Ret = *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0xfffc;
    u32Ret += *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0xfffb;
    u32Ret += *point;
    u32Ret = u32Ret<<8;
}

```

```

    point = (uint8 code *)0xfffa;
    u32Ret += *point;
    return u32Ret;
}

//放在 root 区
uint8 checkCRC(void)
{
    uint16 u16CrcData,u16CalCRCData;
    uint32 u32CrcLen;
    CBANK = 3;
    u16CrcData = GetCRCData();
    u32CrcLen = GetCRCLen();

    if(u32CrcLen<0x20000&&u32CrcLen>0x18000)
    {
        CBANK = 1;
        u16CalCRCData = do_CRC_root(0x5555,0,0x400);
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x600,0x8000-0x600);
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x8000,0x8000);
        CBANK = 2;
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x8000,0x8000);
        CBANK = 3;
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x8000,u32CrcLen-0x18000);
    }
    else
    {
        return 0;
    }
    if(u16CalCRCData!=u16CrcData)
    {
        return 0;
    }
    else
    {
        return 1;
    }
}

```


由于 V9811 是通过分 bank 实现 128KB Flash 的访问，所以 CRC 校验相关代码必须放在 root 区，保证能访问所有 Flash 地址。调用函数（checkCRC）。如果返回 0，表明 CRC 校验失败；返回 1，表明校验成功。

5.2.3 CRC 校验例程_V9801

具体工程参见文件：V9801CRCTest_keil，主要代码如下：

```
//放在 root 区
uint16 GetCRCData(void)
{
    uint16 u16Ret;
    uint8 code *point;
    point = (uint8 code *)0xffff;
    u16Ret = *point;
    u16Ret = u16Ret<<8;
    point = (uint8 code *)0xfffe;
    u16Ret += *point;
    return u16Ret;
}

//放在 root 区
uint32 GetCRCLen(void)
{
    uint32 u32Ret;
    uint8 code *point;
    point = (uint8 code *)0xfffd;
    u32Ret = *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0xfffc;
    u32Ret += *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0xfffb;
    u32Ret += *point;
    u32Ret = u32Ret<<8;
    point = (uint8 code *)0xfffa;
    u32Ret += *point;
    return u32Ret;
}

//放在 root 区
uint8 checkCRC(void)
```

```

{
    uint16 u16CrcData,u16CalCRCData;
    uint32 u32CrcLen;
    CBANK = 3;
    u16CrcData = GetCRCData();
    u32CrcLen = GetCrcLen();
    if(u32CrcLen<0x20000&&u32CrcLen>0x18000)
    {
        CBANK = 1;
        u16CalCRCData = do_CRC_root(0x5555,0,0x400);
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x600,0x8000-0x600);
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x8000,0x8000);
        CBANK = 2;
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x8000,0x8000);
        CBANK = 3;
        u16CalCRCData = do_CRC_root(u16CalCRCData,0x8000,u32CrcLen-0x18000);
    }
    else
    {
        return 0;
    }
    if(u16CalCRCData!=u16CrcData)
    {
        return 0;
    }
    else
    {
        return 1;
    }
}

```

由于 V9801 是通过分 bank 实现 128KB Flash 的访问，所以 CRC 校验相关代码必须放在 root 区，保证能访问所有 Flash 地址。

调用函数（checkCRC）。如果返回 0，表明 CRC 校验失败；返回 1，表明校验成功。

版本更新说明

时间	版本	说明
2014-09-17	V1.0	初稿
2015-4-22	V2.0	增加晶体停振检测选项
2015-8-17	V3.0	修改烧写数量控制的方法和 hex 加密烧写流程
2015-12-1	V5.0	增加合盖检测功能
2016-7-1	V6.0	增加烧写数量统计、Flash 加密范围和支持 V99xx
2016-8-12	V6.1	增加 SD502 时间校正和 RTC 时间设置功能

万高（杭州）科技有限公司保留对本手册所涉及的产品及相关的技术信息进行补正或更新的权利。使用本手册时，请您从我们的销售渠道或登录公司网站 <http://www.vangotech.com> 获取最新信息。