



GProgrammer用户手册

版本： 3.1

发布日期： 2023-09-24

版权所有 © 2023 深圳市汇顶科技股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得对本手册内的任何部分擅自摘抄、复制、修改、翻译、传播，或将其全部或部分用于商业用途。

商标声明

GOODIX 和其他汇顶商标均为深圳市汇顶科技股份有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人持有。

免责声明

本文档中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。

深圳市汇顶科技股份有限公司（以下简称“GOODIX”）对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。GOODIX对因这些信息及使用这些信息而引起的后果不承担任何责任。

未经GOODIX书面批准，不得将GOODIX的产品用作生命维持系统中的关键组件。在GOODIX知识产权保护下，不得暗或以其他方式转让任何许可证。

深圳市汇顶科技股份有限公司

总部地址：深圳市福田区保税區腾飞工业大厦B座12-13层

电话：+86-755-33338828 邮编：518000

网址：www.goodix.com

前言

编写目的

本文档介绍了GProgrammer的安装过程以及各功能模块的操作使用，旨在帮助用户快速入门GProgrammer软件的使用。

读者对象

本文适用于以下读者：

- 芯片用户
- 开发人员
- 测试人员

版本说明

本手册为第17次发布，对应的软件版本为GProgrammer V1.2.41。

修订记录

版本	日期	修订内容
1.5	2020-05-30	首次发布
1.6	2020-06-30	<ul style="list-style-type: none"> • “芯片配置”章节sectors软件可修改。 • “命令行”章节GR551x_console.exe命令程序的erase和download命令新增一个参数，新增“GR551x_encrypt_signature.exe”和“用户自定义Windows脚本”章节。 • “加密加签”章节新增用于验证签名的公钥的Hash文件，变更加密加签后的文件后缀，支持加密加签或者仅加签。
1.7	2020-08-30	<ul style="list-style-type: none"> • “芯片选型”章节GR551x系列新增GR5515I0ND芯片。 • “固件”章节变更“Delete”和“Startup”图标。
1.8	2020-09-30	“下载固件”章节增加下载固件失败的说明。
1.9	2020-11-26	更新软件页面截图。
2.0	2021-01-05	更新芯片选型界面截图和固件界面截图。
2.1	2021-03-02	<ul style="list-style-type: none"> • “eFuse设置”章节补充生成的文件不能修改的说明。 • “导入与导出”章节补充文件导出说明。 • “设备日志”章节补充查看设备日志的前提操作说明。 • “eFuse展示”章节增加“IO_LDO_SEL”字段。 • “命令行”章节erase和download命令删除nvds参数。
2.2	2021-05-13	去除GMF03x系列芯片相关功能。
2.3	2021-07-16	更新芯片选型界面截图。
2.4	2021-09-06	更新芯片选型界面截图。

版本	日期	修订内容
2.5	2022-02-20	<ul style="list-style-type: none"> 增加适配GR5526系列芯片的描述。 “Firmware”和“Encrypt & Sign”界面，支持添加.hex格式固件文件。 “Firmware”界面，新增用户App固件导出功能按钮“Export”，支持导出芯片支持的.bin格式固件文件。 配置外部Flash时，新增SPI Type配置项“QSPI2”并修改相关配置项。 更新GR5xxx_console.exe和GR5xxx_encrypt_signature.exe支持的命令。
2.6	2023-01-19	删除GR5515I0ND。
2.7	2023-02-03	<ul style="list-style-type: none"> 更新GR5xxx_console.exe支持的指令，新增“device”、“load”指令。 GR5xxx_encrypt_signature.exe指令参数增加“rand_number”描述。 新增Linux版本 GProgrammer- Version.tar.bz2 说明。 “配置Flash”章节，新增外部Flash配置注意事项 新增芯片型号查看及重选功能。 SWD连接方式下，新增目标板S/N获取与选择功能。 下载数据至Flash时，支持添加多种文件类型，不限于“.bin”文件类型。 新增NVDS数据解析功能。
2.8	2023-03-30	<ul style="list-style-type: none"> UART连接新增支持GR5526的USB连接参数“Detect USB。” 优化GR5526下载数据到Boot区域。 新增NVDS参数ID范围配置。 更新GR5xxx_console.exe支持的指令，新增“dump”指令。 更新Feedback邮箱为开发者社区网址。
2.9	2023-04-20	<ul style="list-style-type: none"> 新增支持多款芯片的相关描述。 GR533x系列新增“OTP Layout”功能。
3.0	2023-09-22	<ul style="list-style-type: none"> 新增选型芯片。 更新“安装软件”、“硬件连接”章节少许描述。 更新“下载数据”章节关于“强制下载数据功能”的限制描述。 更新GR5xxx_console.exe指令参数。 新增ECDSA签名算法支持。 增加“常见问题”。
3.1	2023-09-24	<ul style="list-style-type: none"> 新增选型芯片。 GR533x 系列新增固件“加签”功能。

目录

前言.....	I
1 简介.....	1
2 安装指南.....	2
2.1 安装要求.....	2
2.2 安装软件.....	2
3 GProgrammer使用说明.....	4
3.1 硬件连接.....	4
3.2 芯片选型.....	5
3.3 主界面.....	6
3.4 连接管理.....	7
3.5 固件.....	10
3.5.1 下载固件.....	10
3.5.2 操作序列.....	12
3.6 Flash操作.....	13
3.6.1 内部Flash.....	14
3.6.1.1 配置Flash.....	14
3.6.1.2 擦除Flash.....	15
3.6.1.3 下载数据.....	17
3.6.1.4 Dump数据.....	18
3.6.2 外部Flash.....	18
3.6.2.1 配置Flash.....	18
3.6.2.2 Flash操作.....	20
3.7 加密加签.....	21
3.7.1 eFuse设置.....	21
3.7.2 eFuse下载.....	23
3.7.3 固件加密加签.....	24
3.8 eFuse展示.....	25
3.9 OTP展示.....	27
3.10 芯片配置.....	28
3.10.1 初始化NVDS.....	29
3.10.2 读取参数.....	30
3.10.3 写入参数.....	30
3.10.4 新增USER参数.....	31
3.10.5 修改参数.....	33
3.10.6 删除USER参数.....	34
3.10.7 导入与导出.....	34
3.10.8 解析NVDS数据.....	35

3.11 设备日志.....	36
3.12 命令行.....	37
3.12.1 GR5xxx_console.exe.....	37
3.12.2 GR5xxx_encrypt_signature.exe.....	40
3.12.3 用户自定义Windows脚本.....	41
3.13 帮助信息.....	42
3.14 常见问题.....	42
3.14.1 启动GProgrammer后，软件界面显示为空白.....	42

1 简介

GProgrammer是一款支持低功耗蓝牙GR5xx系列芯片的固件烧录工具，主要特性如下：

- 支持SWD与串口连接方式
- 下载固件
- Flash操作
- 制定产品信息
- 下载文件至eFuse
- 查看eFuse存储信息
- 查看OTP（One Time Programmable，一次性可编程）存储信息
- 固件加密加签
- 配置NVDS参数
- 显示设备日志
- 支持命令行操作

GProgrammer软件主界面如图 1-1所示。

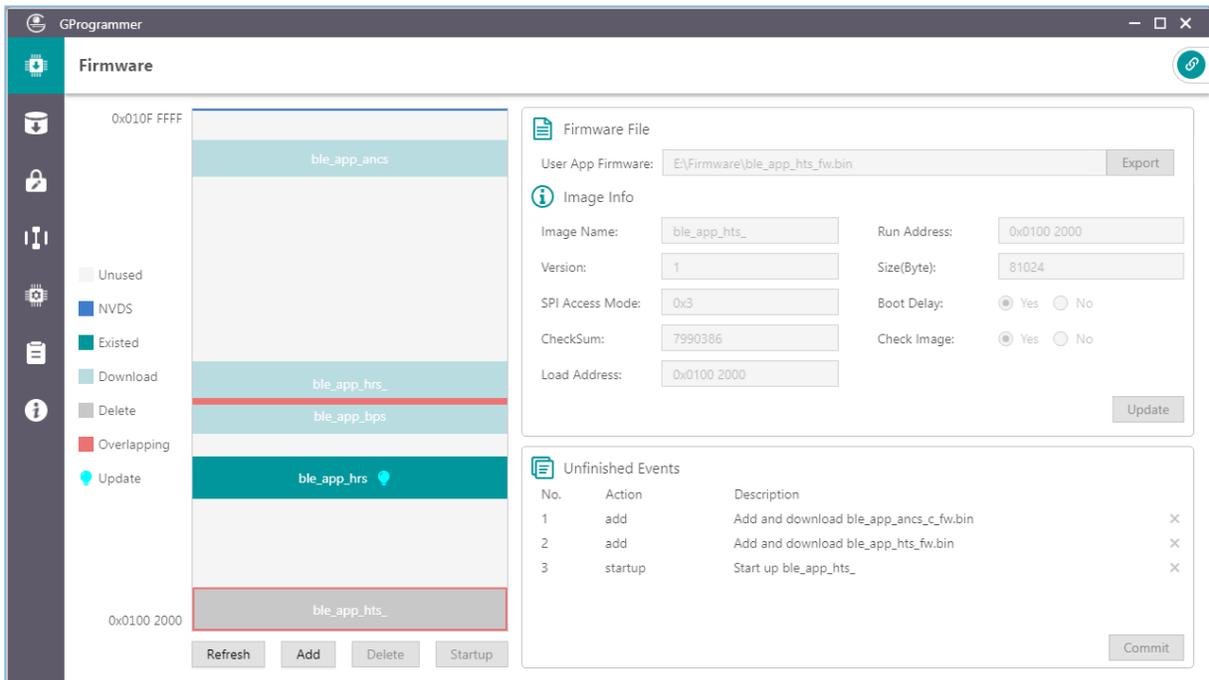


图 1-1 GProgrammer软件主界面

2 安装指南

本章主要介绍GProgrammer安装的环境要求以及安装过程。

2.1 安装要求

- 硬件环境

表 2-1 硬件要求

名称	描述
CPU	1.6 GHz及以上处理器
RAM	1 GB及以上RAM容量

- 操作系统

表 2-2 操作系统

名称	描述
Windows	Windows 7/Windows 10版本（32位/64位）
Linux	Ubuntu 22.04（64位）

2.2 安装软件

GProgrammer支持在Windows和Linux平台上安装使用。

- 安装文件
 - Windows平台：GProgrammer安装文件为可执行安装程序*GProgrammer_Windows_Version.exe*。
 - Linux平台：GProgrammer安装文件为免安装版本*GProgrammer_Linux_x64_Version.zip*。用户可直接解压此文件，再点击文件夹下的*gprogrammer*可执行文件启动软件。

说明:

Version表示GProgrammer软件版本号。

- 安装步骤

在Windows平台，用户可按照以下步骤安装GProgrammer：

1. 双击GProgrammer的可执行安装程序*GProgrammer_Windows_Version.exe*，进入GProgrammer安装界面（如图 2-1所示），然后按照安装向导提示逐步完成安装操作。



图 2-1 GProgrammer安装界面

2. GProgrammer安装完成后，将提示用户继续安装JLink驱动，如图 2-2所示。

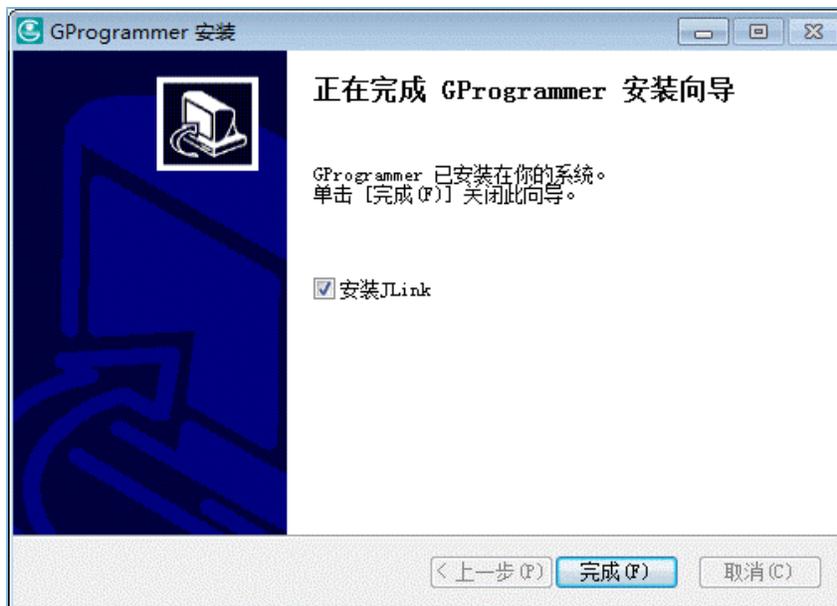


图 2-2 安装JLink提示

说明:

若用户在安装GProgrammer之前已在PC上安装了JLink驱动，则可取消勾选“安装JLink”。

3. JLink安装完成后，用户可通过桌面或开始菜单的快捷方式启动GProgrammer软件。

3 GProgrammer使用说明

本章主要介绍GProgrammer各功能模块的操作使用。

3.1 硬件连接

使用GProgrammer之前，请先完成主机（PC）与目标板的硬件连接，可选择SWD连接或串口连接方式。

- SWD连接

若用户选择SWD连接方式，则需通过J-Link仿真器连接PC与目标板：使用USB线将仿真器的一端与PC相连；使用杜邦线将仿真器的另一端与目标板上的对应芯片管脚相连。

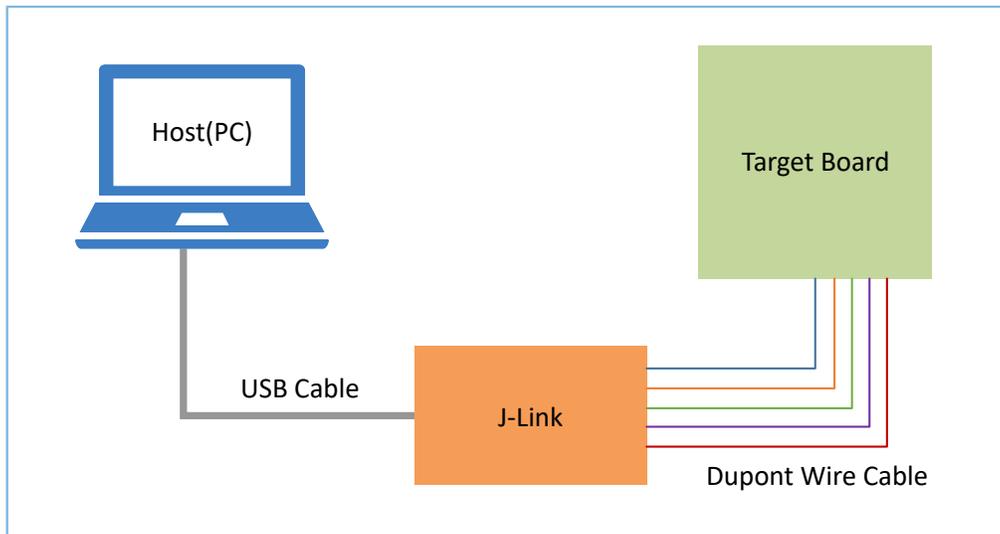


图 3-1 SWD硬件连接示意图

J-Link仿真器与芯片管脚的连接对应关系如下：

表 3-1 J-Link仿真器与芯片的管脚连接对应关系

J-Link仿真器管脚	芯片管脚
VCC	VCC
GND	GND
SWDIO	GPIO_1
SWCLK	GPIO_0

说明:

- 如果目标板上集成了J-Link仿真器芯片，用户可直接使用USB线连接PC与目标板。
- SK板默认集成的J-Link芯片不支持波特率“921600”，而芯片固化的ROM升级程序必须使用波特率“921600”。因此，无法通过SK板的J-link串口直接连接PC和烧录固件。

- 串口连接

若用户选择串口连接方式，则需通过USB转串口的转接板连接PC与目标板：使用USB线将转接板的一端与PC相连；使用杜邦线将转接板的另一端与目标板上的对应芯片管脚相连。

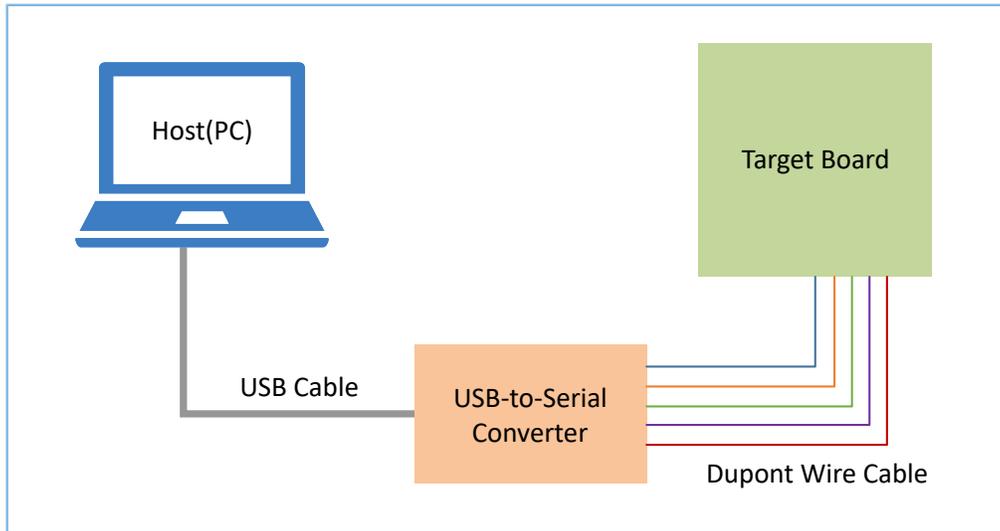


图 3-2 URAT硬件连接示意图

转接板与芯片管脚的连接对应关系如下：

表 3-2 转接板与芯片的管脚连接对应关系

转接板管脚	芯片管脚
VCC	VCC
GND	GND
TX	GPIO_1
RX	GPIO_0
RTS	CHIP_EN

说明：

- 如果目标板上集成了USB转串口芯片，用户可直接使用USB线连接PC与目标板。
- 当SWD关闭后，使用串口（UART）连接成功后，芯片Flash中的固件信息会被擦除。

3.2 芯片选型

使用GProgrammer时，必须先选择目标板的主控芯片型号，然后点击右下角的“OK”按钮，才能进入GProgrammer操作主界面。

说明：

启动GProgrammer后，默认进入芯片选型页面。

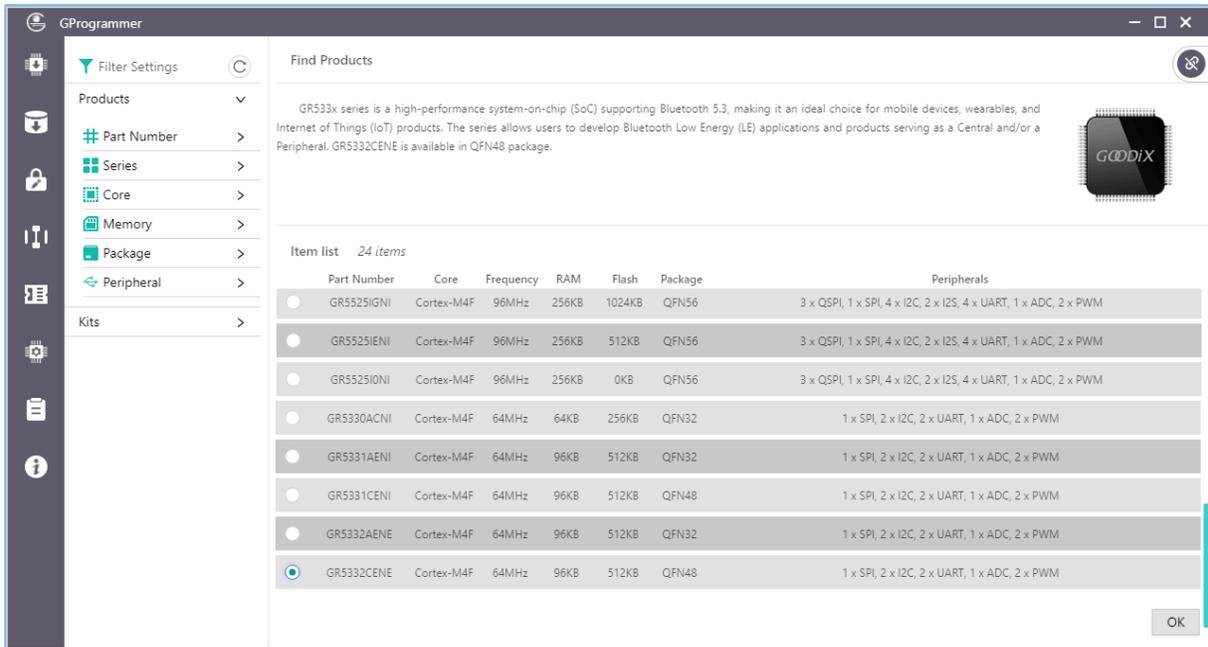


图 3-3 芯片选型界面

在芯片选型页面，左侧为筛选条件的设置面板；右侧为筛选结果的展示面板。用户可按照产品型号、产品系列、内核、内存大小、集成的外设、封装形式等条件进行产品筛选。

说明:

芯片选型页面上所展示的芯片外设，并非芯片的所有外设。如需获取芯片的全面信息，请参考该系列芯片的数据手册。

3.3 主界面

选择芯片后，将进入芯片对应的软件主界面，如图 3-4 所示。

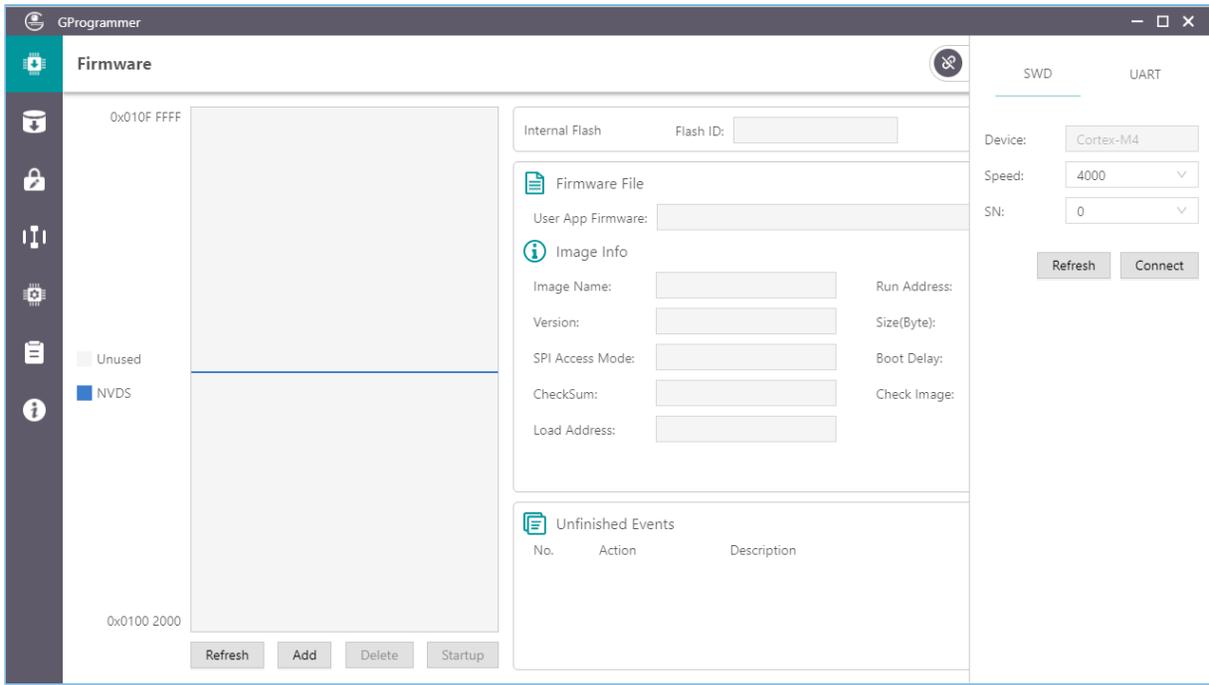


图 3-4 GProgrammer软件主界面

软件界面的左侧为功能导航栏（具体描述见表 3-3），右侧为功能操作区域。

表 3-3 功能导航栏选项

图标	功能名称	描述
	-	查看/重选芯片型号 <ul style="list-style-type: none"> 鼠标移入后，可查看当前选择的芯片型号 如需更换所选芯片型号，可点击此图标返回芯片选型页面。 重新选择芯片型号后，点击右下角“OK”按钮将断开已有设备连接
	Firmware	点击该图标，进入“固件”操作页面
	Flash	点击该图标，进入“Flash”操作页面
	Encrypt & Sign	点击该图标，进入“加密加签”操作页面
	eFuse Layout	点击该图标，进入“eFuse展示”操作页面
	Chip Configuration	点击该图标，进入“芯片配置”操作页面
	Device Log	点击该图标，进入“设备日志”操作页面
	Help	点击该图标，进入“帮助”操作页面

说明:

对于GR533x系列芯片，支持OTP，增加“OTP Layout”操作页面。

3.4 连接管理

用户可管理控制目标板与主机之间的连接。

点击软件界面右上角的，可展开连接管理面板；再次点击，可收起并隐藏该面板。

GProgrammer支持SWD和串口两种连接方式。

- SWD连接

在SWD连接方式下，用户配置下表参数后，即可点击“Connect”按钮连接目标板。

表 3-4 参数配置说明

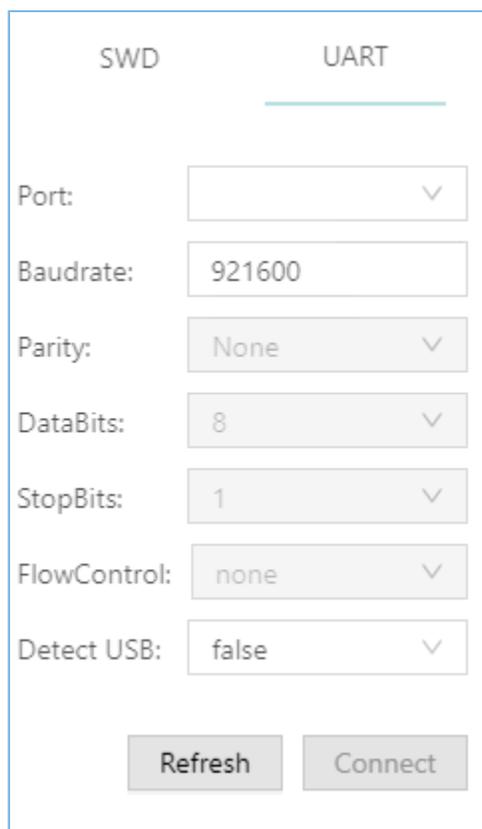
参数	描述
Device	目标板的芯片内核，默认为“Cortex-M4”，且不可更改。
Speed	传输速率，默认为“4000” kHz。
SN	目标板序列号，默认为“0”。 <ul style="list-style-type: none"> ◦ 当PC仅连接了一个目标板时，可直接使用默认值，或点击“Refresh”按钮获取当前目标板序列号。 ◦ 当PC连接多个目标板时，需点击“Refresh”获取各目标板对应序列号，再选择待连接的目标板序列号。 若直接使用默认值“0”进行连接，Windows环境下将弹窗提示选择目标板序列号。

图 3-5 SWD连接

- 串口（UART）连接

在串口（UART）连接方式下，用户可根据实际情况配置串口号“Port”（需点击“Refresh”按钮获取串口号列表，再选择正确的串口号）和波特率，其他参数默认配置不可修改。

参数配置完成后，点击“Connect”按钮连接目标板。



SWD	UART
	Port: <input type="text"/>
	Baudrate: <input type="text" value="921600"/>
	Parity: <input type="text" value="None"/>
	DataBits: <input type="text" value="8"/>
	StopBits: <input type="text" value="1"/>
	FlowControl: <input type="text" value="none"/>
	Detect USB: <input type="text" value="false"/>
	<input type="button" value="Refresh"/> <input type="button" value="Connect"/>

图 3-6 UART连接

说明:

“Detect USB”参数仅用于GR5526系列芯片的USB连接。设置为“true”，表示检测USB设备，点击“Refresh”按钮后，界面将提示用户复位设备，此时PC端将查找可用的USB设备（USB设备将枚举为串口设备）并将其加入“Port”列表中。设置为“false”，表示不检测USB设备，点击“Refresh”按钮，将获取所有串口设备加入到“Port”列表中。

目标板连接成功后，连接管理面板将自动收起并隐藏。同时，按钮将变成，表示当前的连接状态为“已连接”。

如需断开连接，再次点击，打开连接管理面板，然后点击“Disconnect”按钮即可。

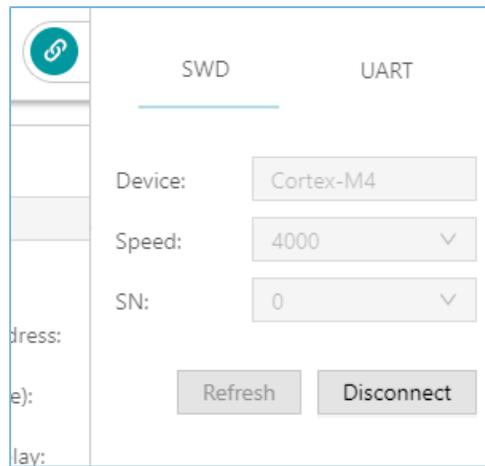


图 3-7 断开连接

3.5 固件

点击左侧功能导航栏的图标 , 进入“Firmware”（固件操作）页面。

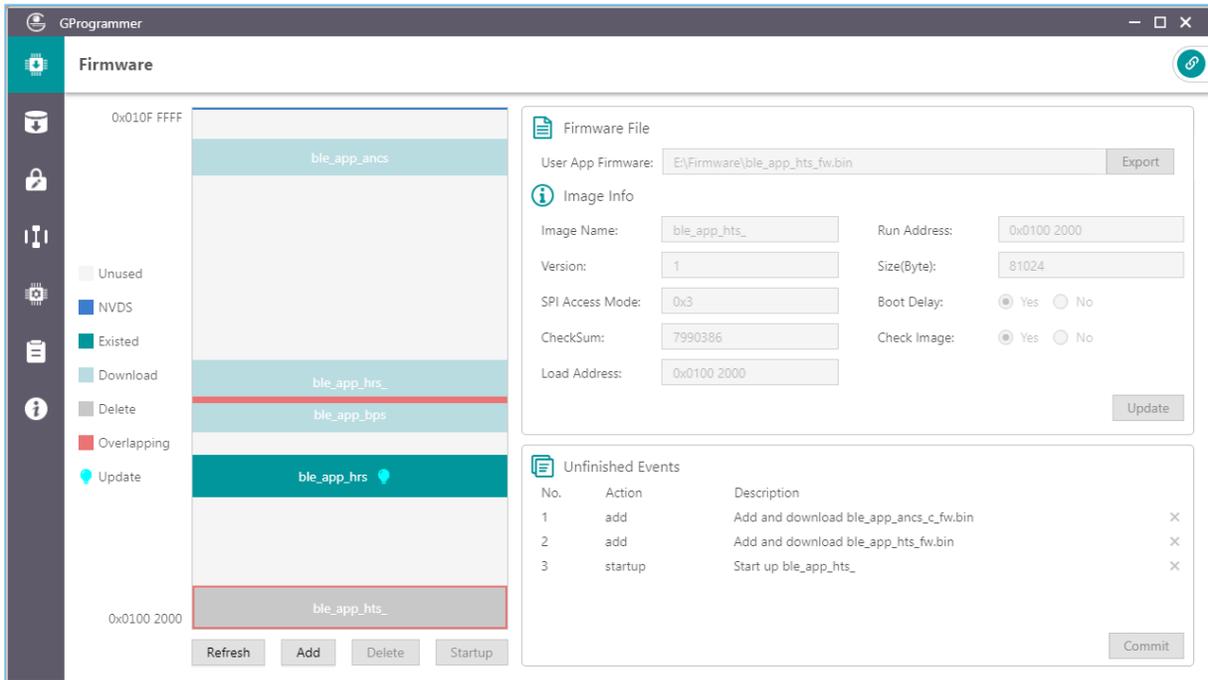


图 3-8 “Firmware” 页面

用户可将User APP固件下载至Flash中0x01002000 ~ 0x010FFFFF的连续空间。

说明:

Flash的起始地址和结束地址根据具体系列芯片的Flash大小有所差异。

3.5.1 下载固件

GProgrammer以图形化方式展示Flash Firmware Layout（如图 3-9所示），可帮助用户直观了解Flash中固件区域的占用情况。

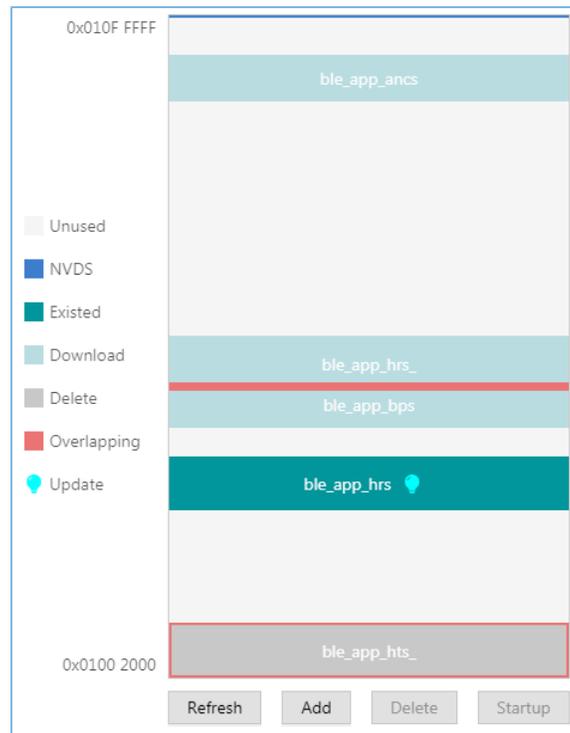


图 3-9 Flash Firmware Layout 示意图

- 浅灰色：可下载的Flash空间。
- 深蓝色：软件默认配置的NVDS区域。固件不能下载至该区域。
- 深灰色：待删除的固件，如ble_app_hts_。
- 浅绿色：待下载的固件，如ble_app_ancs。
- 深绿色：已存在于Flash中的固件，如ble_app_hrs。
- 红色：两个固件的占用空间的重叠区域，如ble_app_hrs_和ble_app_bps。

用户下载固件时，请按以下步骤操作：

1. 点击“Add”按钮，从本地添加需下载的固件文件（.hex/.bin格式）。固件文件添加后，可查看到该固件的详细信息，例如固件文件路径、固件Image Info信息。

点击“Firmware File”区域的“Export”按钮，可将导入的固件文件，转换为可供芯片使用的.bin格式文件（非加密）。

2. 点击“Commit”按钮，将固件文件下载到Flash。

下载完成后，若Layout示意图中的固件由“浅绿色”变为“深绿色”，则表示固件下载成功。

说明:

1. 连接成功后，将自动从连接的目标板的Flash中读取已存在的固件。
2. 下载固件时，如果连接JLink失败，开发板无法连接或者正常下载固件。芯片当前可能处于睡眠状态（即开启了睡眠模式的固件正在运行），可先按下开发板的“RESET”键，并间隔1秒左右重新下载固件。如果仍然无法下载固件，可擦除Flash后，重新下载固件。

3.5.2 操作序列

用户可一次性执行多个操作，例如：一次性下载多个固件，并将某个固件设置为启动项。用户执行的操作均为“预操作”，需点击“Commit”按钮后才会真正地执行，并且操作序列将在“Unfinished Events”栏中显示，如图 3-10所示。

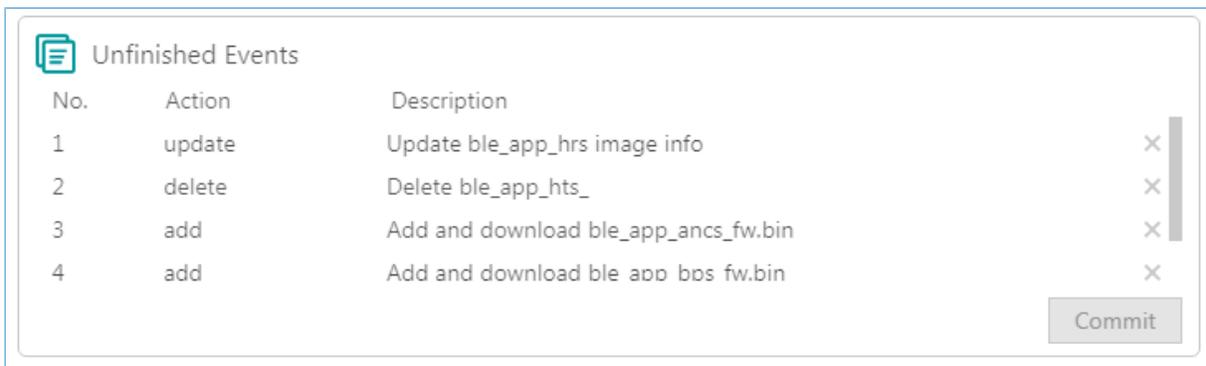


图 3-10 操作列表

用户可执行的固件操作主要包括：

表 3-5 固件操作

操作名称	操作按钮	描述
添加固件	Add	<p>点击“Add”按钮，从本地浏览添加固件文件。</p> <p>另外，也可直接从资源管理器拖拽文件。</p> <p>说明：</p> <p>拖拽文件时需直接拖拽，不可拖拽后点击“打开”。</p>
刷新固件	Refresh	<p>指获取当前目标板Flash中的固件信息。点击“Refresh”后，会被直接执行。</p> <p>如果“Unfinished Events”有针对当前目标板Flash中固件的未执行操作，如“开始运行”或“更新固件”，那么这些未执行操作将被撤回（对已存在的固件有效，只“Add”但未“Commit”的固件的Startup操作不会被撤销），已修改的参数也会被重置。</p>
删除固件	Delete	<p>指删除目标板Flash中已存在的固件。在Flash Layout示意图中选中待删除固件，然后点击“Delete”按钮，该固件会变成深灰色，且“Unfinished Events”中将增加一条Delete操作。</p> <p>说明：</p>

操作名称	操作按钮	描述
		“Delete”操作只删除对应固件的Image Info，不会删除存储区域实际的固件信息。
设置为启动项		指将固件设为启动项，即下载完成后立即运行该固件。在Flash Layout示意图中选中心固件，然后点击“Startup”按钮，该固件的右侧将显示  图标，且“Unfinished Events”中将增加一条startup操作。固件运行后，连接将会自动断开。
更新固件		指更新目标板Flash中已存在的固件的信息。在Flash Layout示意图中选中心待更新固件，然后修改固件信息（修改后的参数信息将变成红色），最后点击“Update”按钮，该固件的右侧将显示  图标，且“Unfinished Events”中将增加一条update操作。 执行update操作后，所有的参数选项都将被锁定，不允许再编辑。如需重新修改，则必须先撤回之前的update操作。

说明:

- 在操作序列中，点击操作项右侧的，可撤销该操作。
- 对于有关联关系的两项操作，撤销其中一项可能会自动撤销另一项。例如：添加一个固件文件并将其设置为启动项，即执行“添加”和“设置为启动项”两项操作。如果撤销“添加”操作，那么“设置为启动项”操作将被自动撤销。

另外，如果固件区域里存在重叠区域，“Commit”按钮将置灰不可用，冲突解决后才可提交并执行操作序列。

说明:

若两个固件完全重叠，单击可选中其中一个固件，再双击可选中另一个固件。

3.6 Flash操作

点击左侧功能导航栏的图标，进入“Flash”（Flash操作）页面。

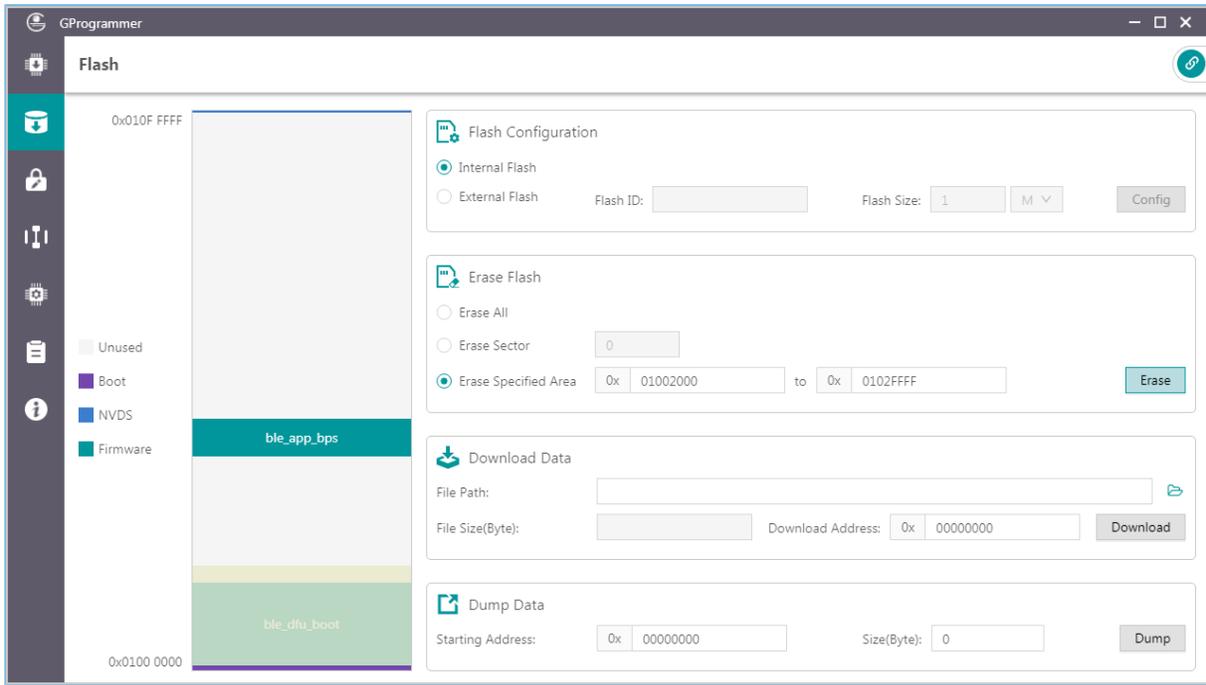


图 3-11 “Flash” 页面

用户可对芯片内部Flash和外部Flash进行操作，包括擦除Flash、下载数据至Flash、从Flash中Dump数据。与固件模块相似，该模块也采用图形化的方式展示Flash Layout信息。

- （浅灰色）：可用的Flash空间。
- （深蓝色）：配置的NVDS区域。
- （深紫色）：Boot Info区域（0x01000000 ~ 0x01002000，根据不同系列芯片有所差异），选择使用内部Flash时将自动加载并显示Boot Info区域。
- （深绿色）：已存在Flash中的固件，如ble_app_bps。
- （淡黄色）：待操作区域，如待擦除的Flash区域。

3.6.1 内部Flash

3.6.1.1 配置Flash

当用户操作内部Flash时，首先需在“Flash Configuration”栏选择“Internal Flash”。

页面左侧的Flash Layout示意图将同步“Firmware”模块的Firmware Layout信息，从而获取当前Flash中的固件区域以及NVDS区域，同时加载Boot Info区域。

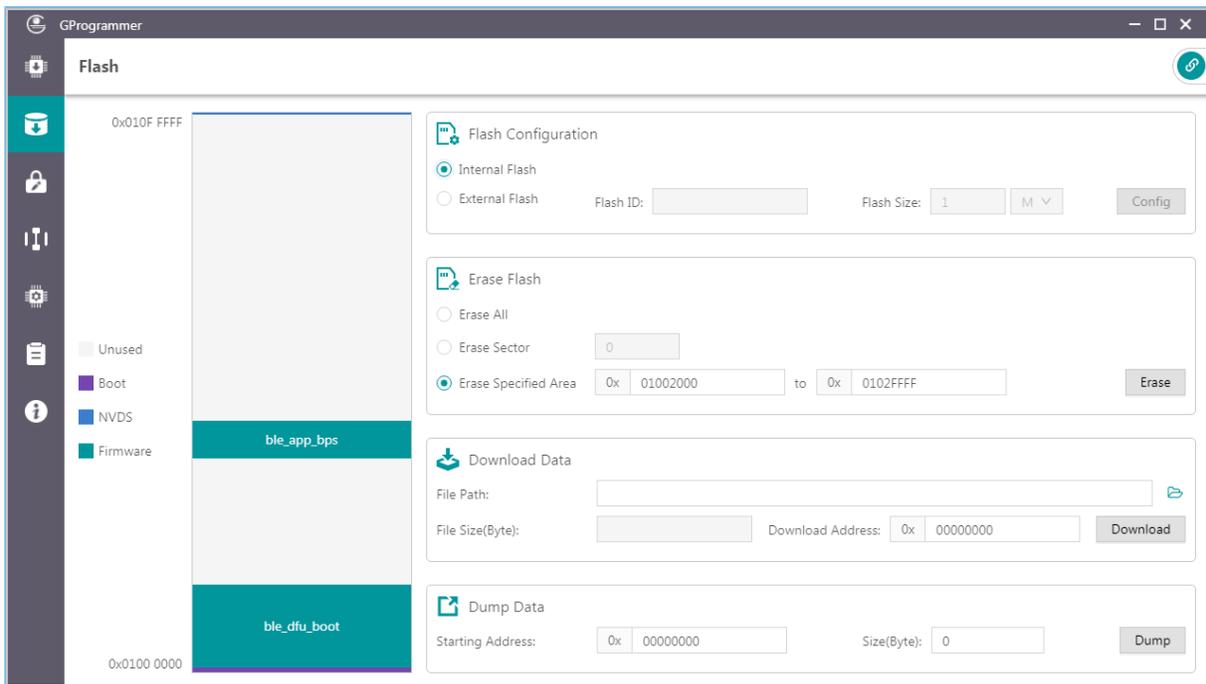


图 3-12 选择内部Flash

3.6.1.2 擦除Flash

GProgrammer支持三种Flash擦除方式：全片擦除、Sector擦除和区域擦除。

- 全片擦除（Erase All）

全片擦除，指整体擦除Flash空间。

擦除后，内部Flash中的Boot和NVDS区域内容将被清空，所有Firmware将被删除。

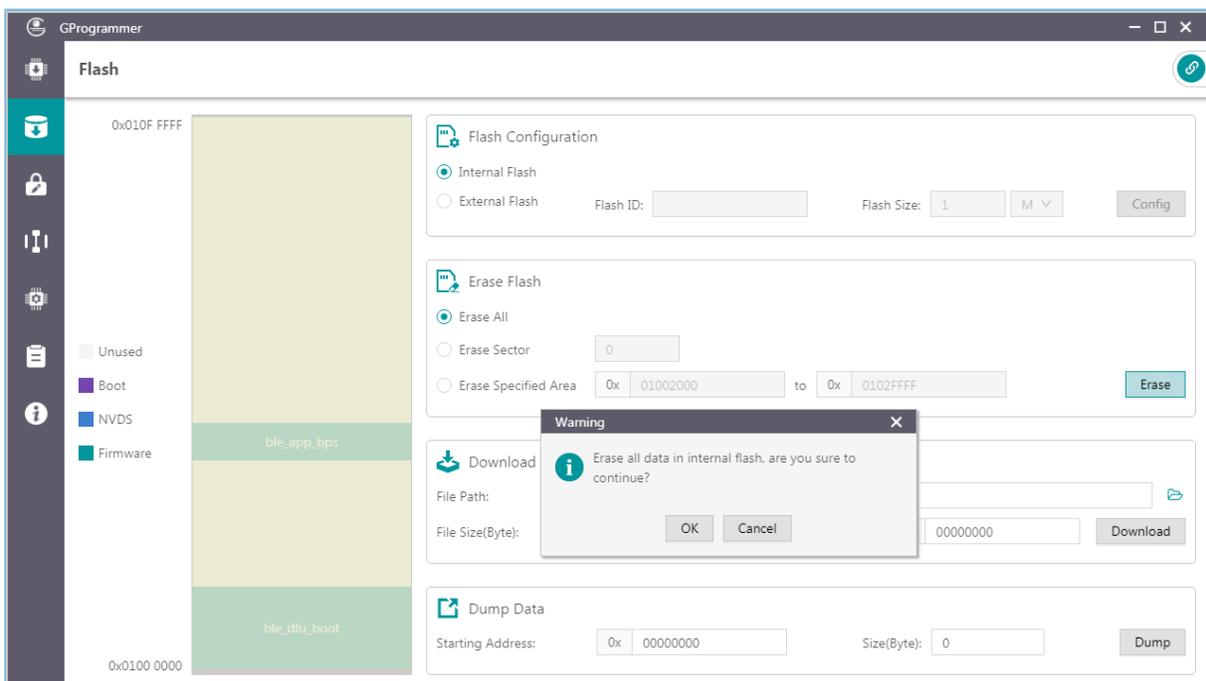


图 3-13 全片擦除

- Sector擦除（Erase Sector）
Sector擦除，指擦除指定的Sector区域（4 KB大小）。

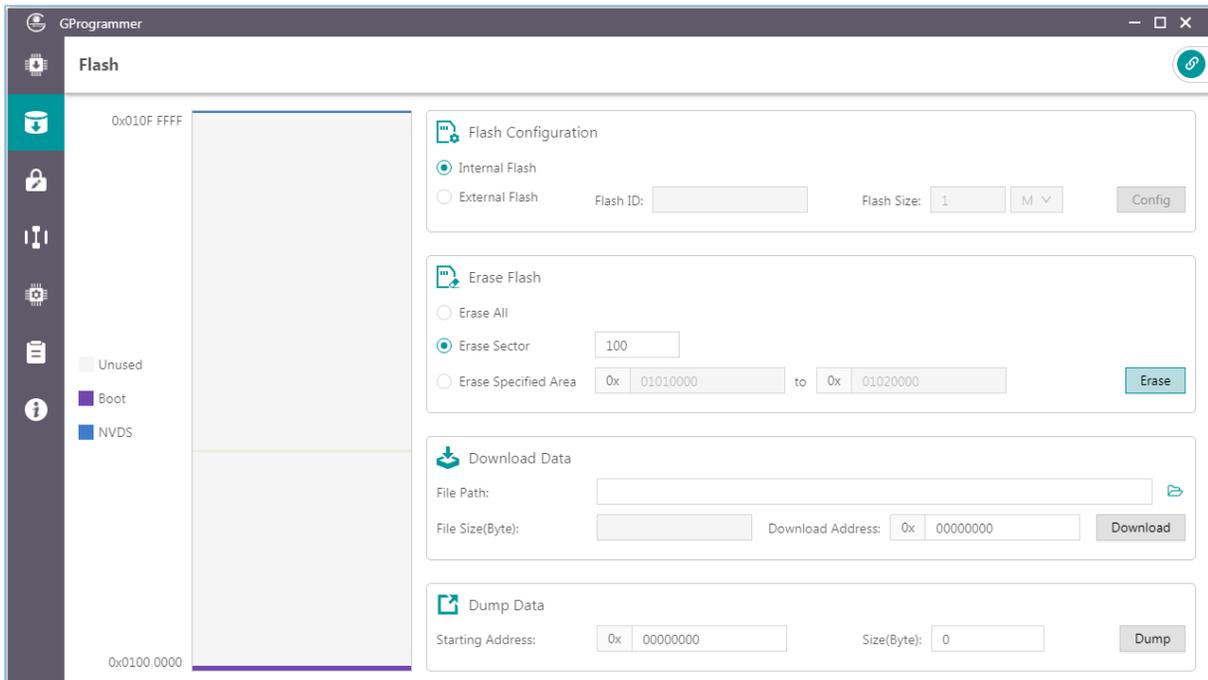


图 3-14 Sector擦除

- 区域擦除（Erase Specified Area）
区域擦除，指在指定的地址范围内按Sector逐一擦除。

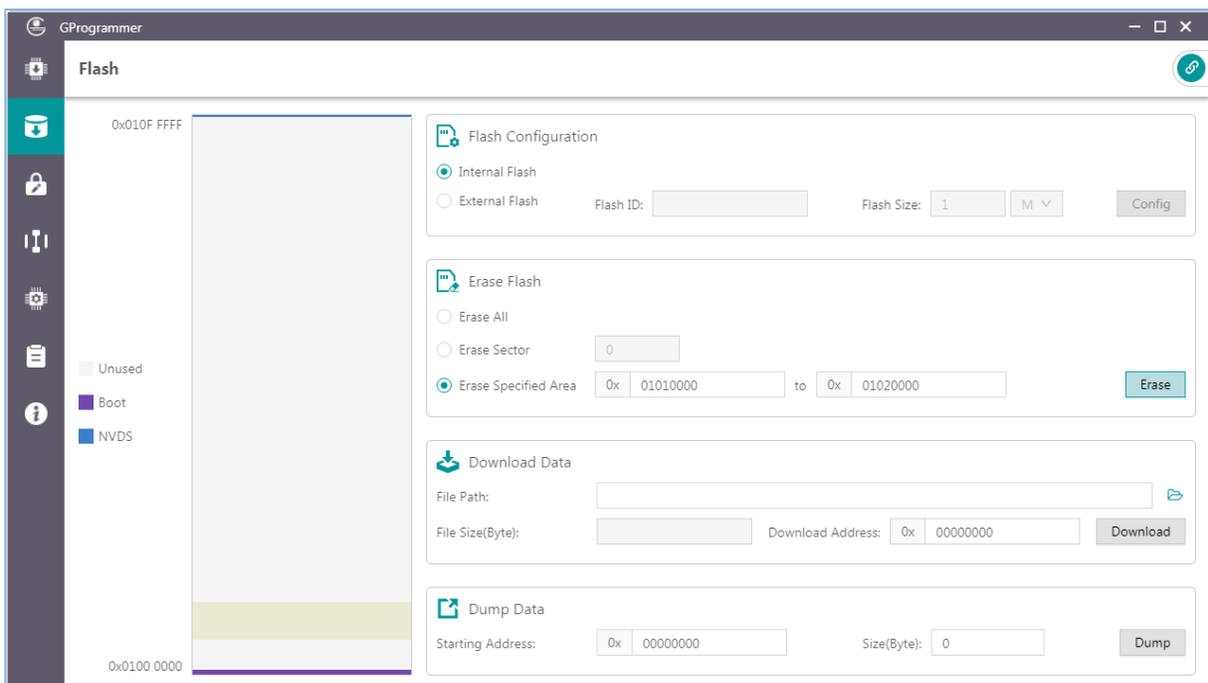


图 3-15 区域擦除

3.6.1.3 下载数据

下载数据至Flash时，用户只需浏览添加数据文件，然后设置下载起始地址即可。

 提示:

数据下载地址需以4 KB对齐。

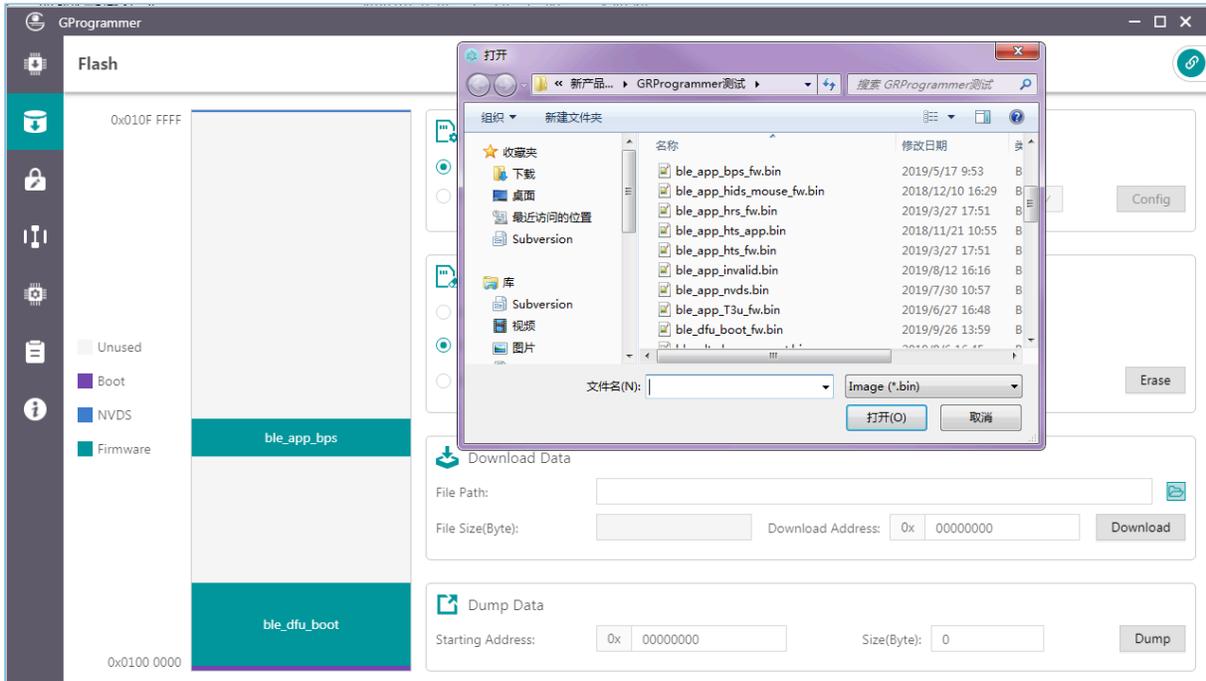


图 3-16 浏览选择数据文件

当下载的数据文件过大或设置的下载起始地址超出范围时，将导致Flash溢出报错。

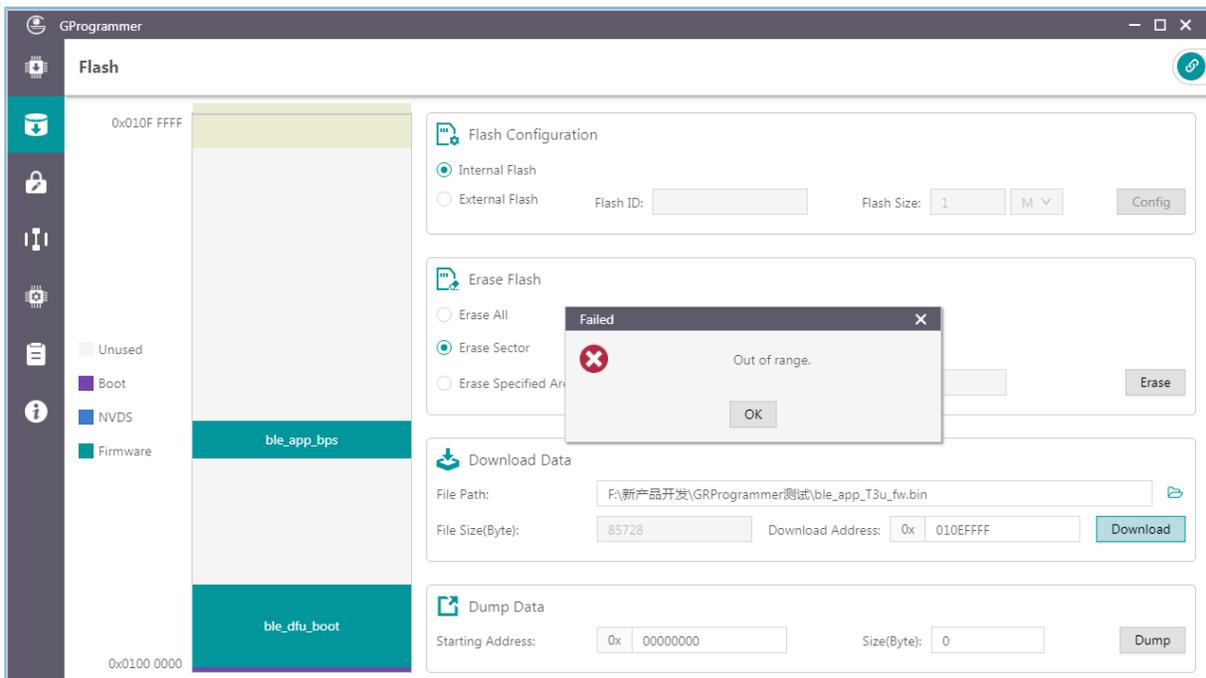


图 3-17 Flash 溢出

说明:

- 若使用SWD方式连接目标板，则可以强制下载数据到Boot区域（SCA区）。
- 若使用串口方式连接目标板，则无法强制下载数据到Boot区域（仅针对GR551x、GR5526系列芯片）。

3.6.1.4 Dump数据

用户可通过指定起始地址和数据大小，将Flash中的任意数据Dump到本地保存为文件。

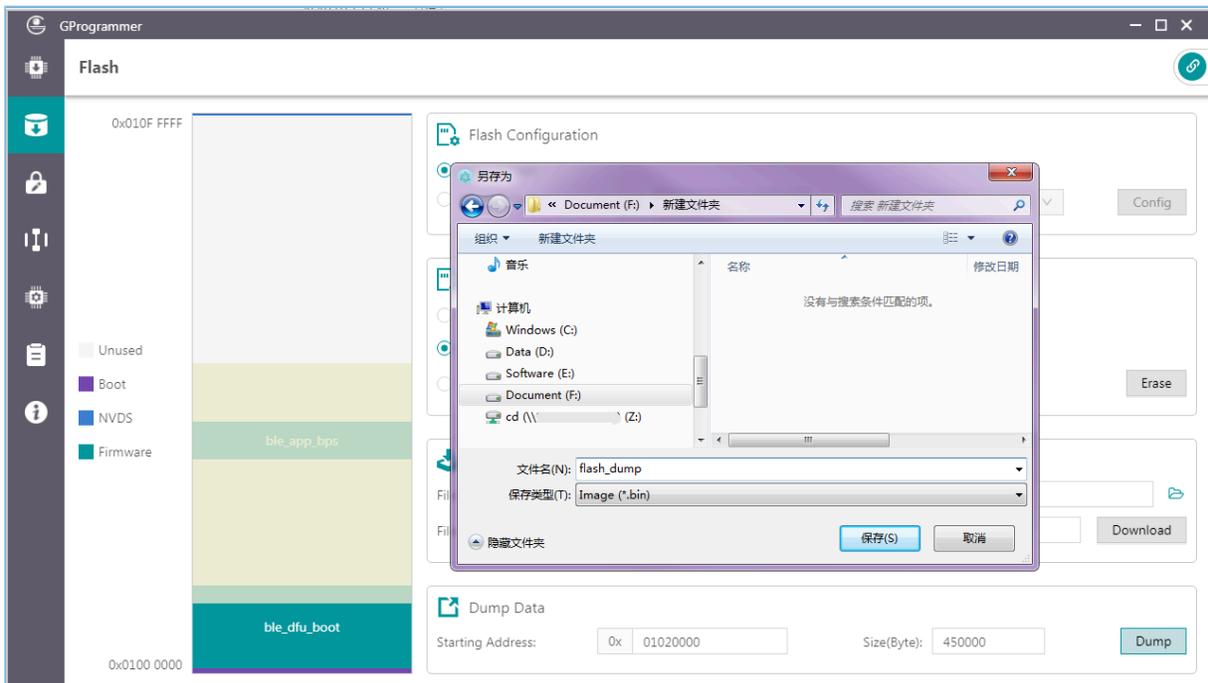


图 3-18 Dump数据

3.6.2 外部Flash

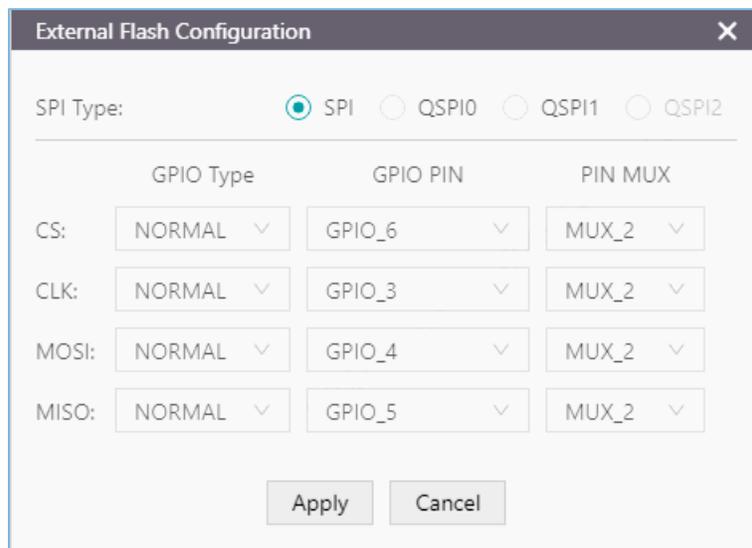
3.6.2.1 配置Flash

当用户操作外部Flash时，首先需在“Flash Configuration”栏选择“External Flash”；然后点击“Config”按钮，打开外部Flash配置窗口。用户可根据实际情况配置PIN脚。

配置完成后，点击“Apply”按钮应用配置信息。

说明:

- 点击“Apply”按钮之前，必须确保外部Flash按照引脚配置与目标板正确连接。否则，外部Flash无法与目标板正常通信。
- GPIO_0、GPIO_1用于设备连接，若配置为外部Flash PIN，将导致GProgrammer与目标板断开连接。

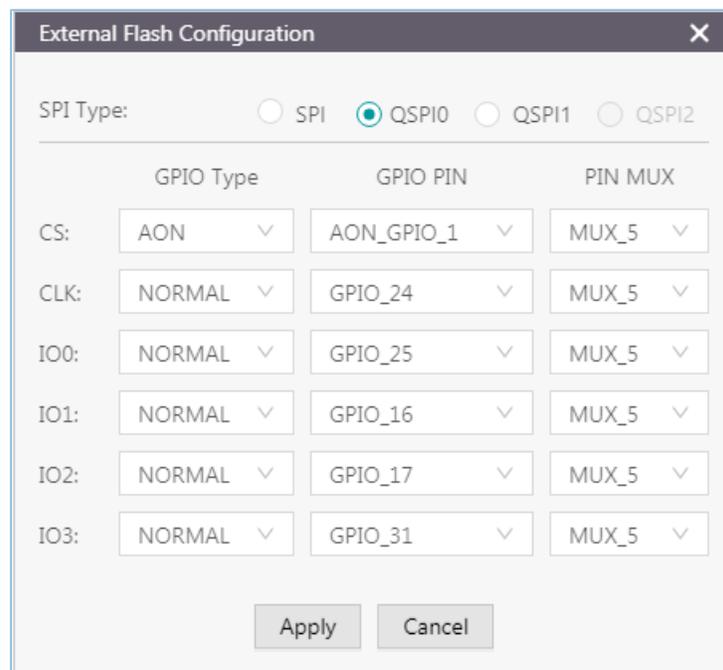


The dialog box titled "External Flash Configuration" shows the "SPI Type" set to "SPI". Below, there are three columns: "GPIO Type", "GPIO PIN", and "PIN MUX". The configurations are as follows:

	GPIO Type	GPIO PIN	PIN MUX
CS:	NORMAL	GPIO_6	MUX_2
CLK:	NORMAL	GPIO_3	MUX_2
MOSI:	NORMAL	GPIO_4	MUX_2
MISO:	NORMAL	GPIO_5	MUX_2

Buttons for "Apply" and "Cancel" are at the bottom.

图 3-19 SPI配置项



The dialog box titled "External Flash Configuration" shows the "SPI Type" set to "QSPI0". Below, there are three columns: "GPIO Type", "GPIO PIN", and "PIN MUX". The configurations are as follows:

	GPIO Type	GPIO PIN	PIN MUX
CS:	AON	AON_GPIO_1	MUX_5
CLK:	NORMAL	GPIO_24	MUX_5
IO0:	NORMAL	GPIO_25	MUX_5
IO1:	NORMAL	GPIO_16	MUX_5
IO2:	NORMAL	GPIO_17	MUX_5
IO3:	NORMAL	GPIO_31	MUX_5

Buttons for "Apply" and "Cancel" are at the bottom.

图 3-20 QSPI0配置项

- Flash Size 配置

用户完成PIN脚配置并应用后，软件将读取并显示外部Flash ID，并根据Flash ID自动配置Flash Size。若无法判定Flash Size，则用户需自行选择设置Flash Size。

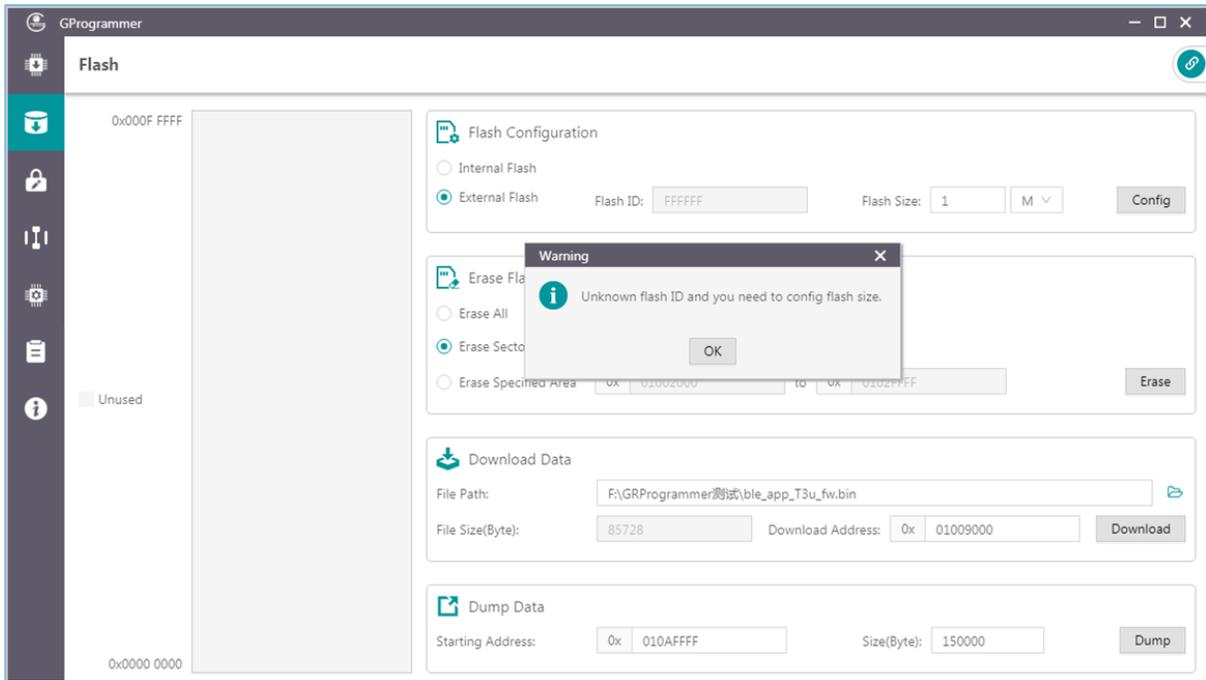


图 3-21 未知Flash

3.6.2.2 Flash操作

用户操作外部Flash（擦除Flash、下载数据、Dump数据）时，只要在有效地址区域之内，可任意操作，无特殊限制。

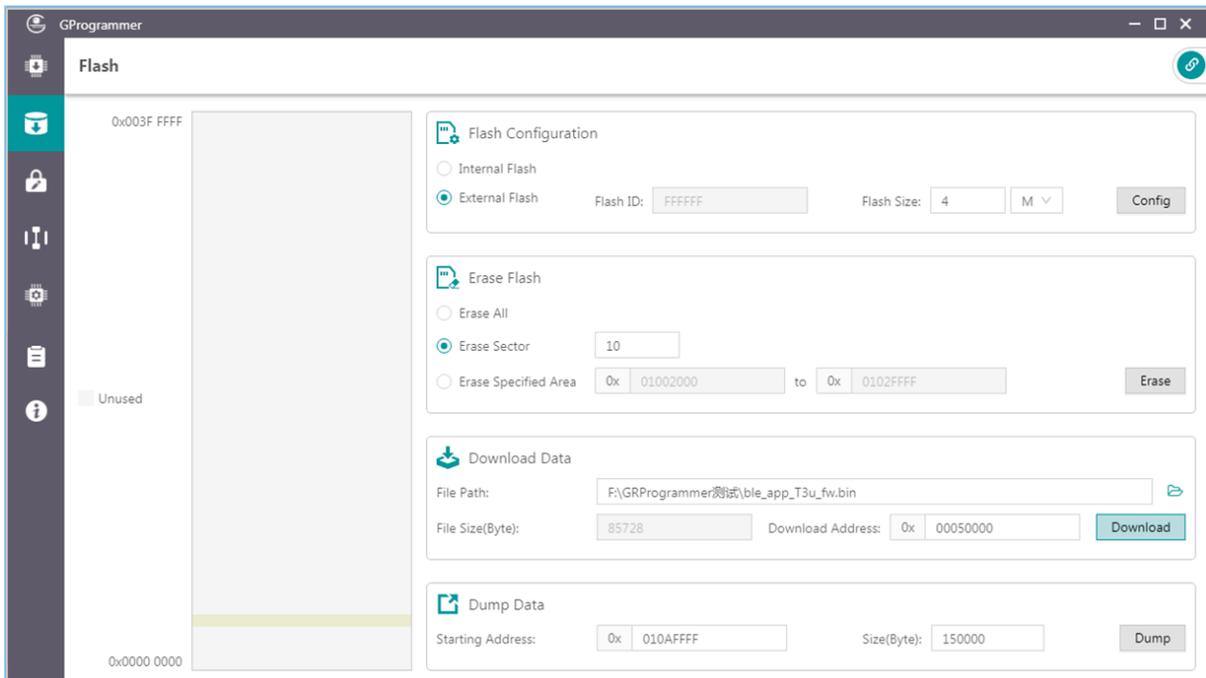


图 3-22 下载数据

说明:

完成PIN脚配置前，无法进行任何针对外部Flash的操作。

3.7 加密加签

点击左侧功能导航栏的图标 ，进入“Encrypt & Sign”（加密加签）页面。

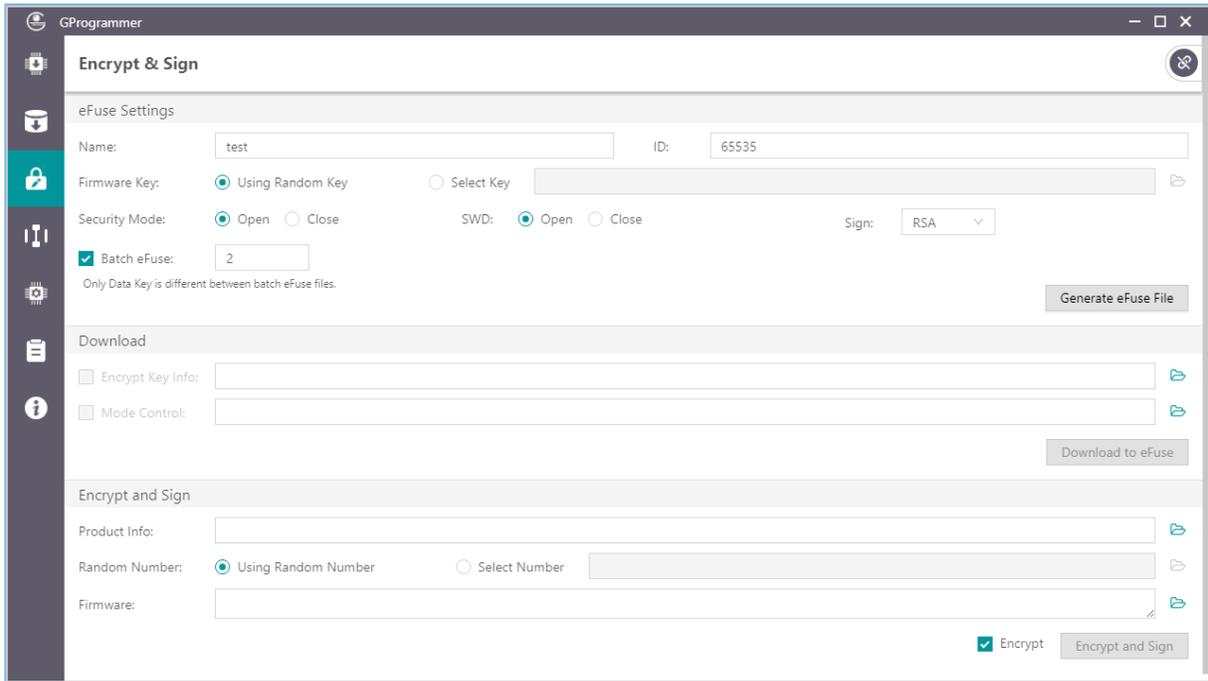


图 3-23 “Encrypt & Sign” 页面

芯片支持安全模式和非安全模式，由写入eFuse的产品信息中的Security Mode决定。在安全模式开启的情况下，固件需加密加签后才可以下载至Flash运行。

3.7.1 eFuse设置

eFuse是芯片内部的一个具有随机访问接口的一次性可编程存储器。eFuse中存放着产品配置信息、安全模式控制信息以及用于加密加签的各种密钥信息等。

用户可通过指定产品的Name、ID、Firmware Key、Security Mode和SWD接口生成存储在eFuse中的相关文件。

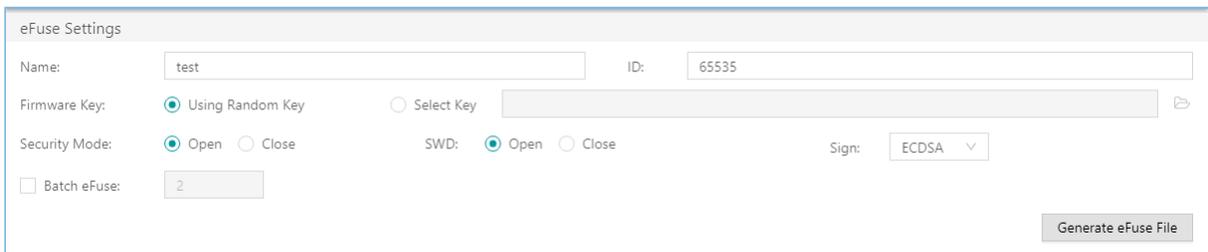


图 3-24 设置参数

说明:

- “Firmware Key”可使用软件自动生成的随机Key，也可使用用户自己选择添加的Key文件。
- “Security Mode”选择“Open”将启用芯片安全模式，该模式开启后不能被关闭。
- “SWD”选择“Close”将关闭SWD功能，但用户依然可以通过DFU对固件进行升级。
- RSA、ECDSA签名算法选择：若芯片为GR551x、GR5526系列，默认使用RSA（页面无选项配置）；若芯片为GR5525系列，可选择RSA或ECDSA；若芯片为GR533x系列，仅可选择ECDSA。

另外，GProgrammer还支持通过设置“Batch eFuse”选项批量生成多个*Encrypt_key_info.bin*文件，且生成的各文件具有唯一性，以保证“一机一密”。例如，勾选“Batch eFuse”选项并在文本框中设置文件数量为“3”，则会生成3个*Encrypt_key_info.bin*文件：*Encrypt_key_info.bin*，*2_Encrypt_key_info.bin*，*3_Encrypt_key_info.bin*。

生成的文件包括：

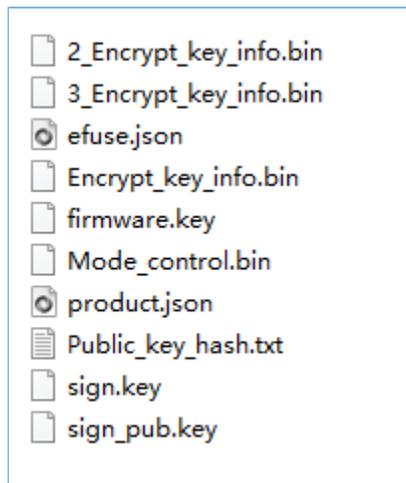


图 3-25 生成的文件

- efuse.json*: 临时文件。
- Encrypt_key_info.bin*、*2_Encrypt_key_info.bin*、*3_Encrypt_key_info.bin*: eFuse下载文件，包含产品及加密加签信息。此文件需下载至eFuse。
- firmware.key*: 用于加密固件的私钥。
- Mode_control.bin*: eFuse下载文件，包含Security Mode和SWD信息。此文件需下载至eFuse。
- product.json*: 产品信息文件。当固件加密加签时，需导入此文件。
- sign.key*: 用于生成签名的私钥。
- sign_pub.key*: 用于验证签名的公钥。
- Public_key_hash.txt*: 用于验证签名的公钥Hash。

说明:

以上文件请用户妥善保存，切忌泄露、丢失。后续的“eFuse下载”和“固件加密加签”操作需使用这些文件。

为了方便用户下载文件至eFuse或加密加签固件，生成的`Encrypt_key_info.bin`和`Mode_control.bin`文件的路径将自动被添加到“Download”区域，产品信息文件`product.json`的路径将自动被添加到“Encrypt and Sign”的“Product Info”栏中，如图 3-26所示。

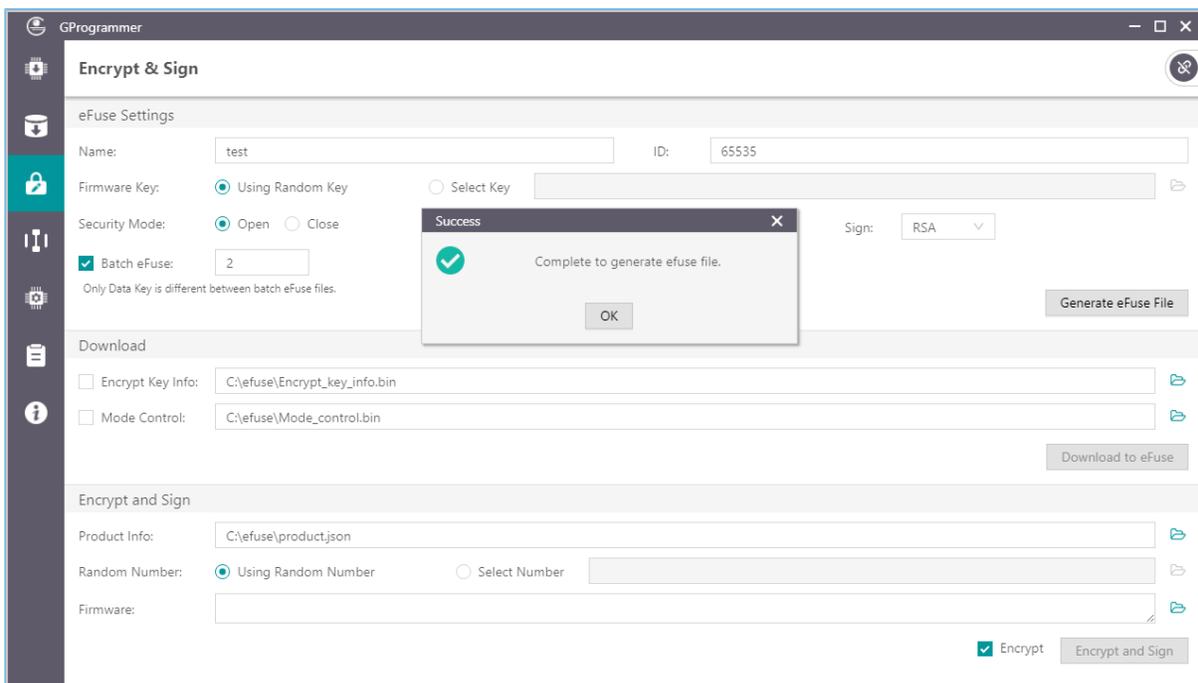


图 3-26 自动加载文件路径

说明:

不要修改eFuse中生成的任何文件，否则可能造成固件加密加签失败。

3.7.2 eFuse下载

若用户先执行“Generate eFuse File”操作生成了`Encrypt_key_info.bin`和`Mode_control.bin`文件，则可直接勾选“Encrypt Key Info”和“Mode Control”选项，点击“Download to eFuse”按钮将文件下载至eFuse。

否则，用户需先手动选择添加`Encrypt_key_info.bin`和`Mode_control.bin`文件，然后才可下载。

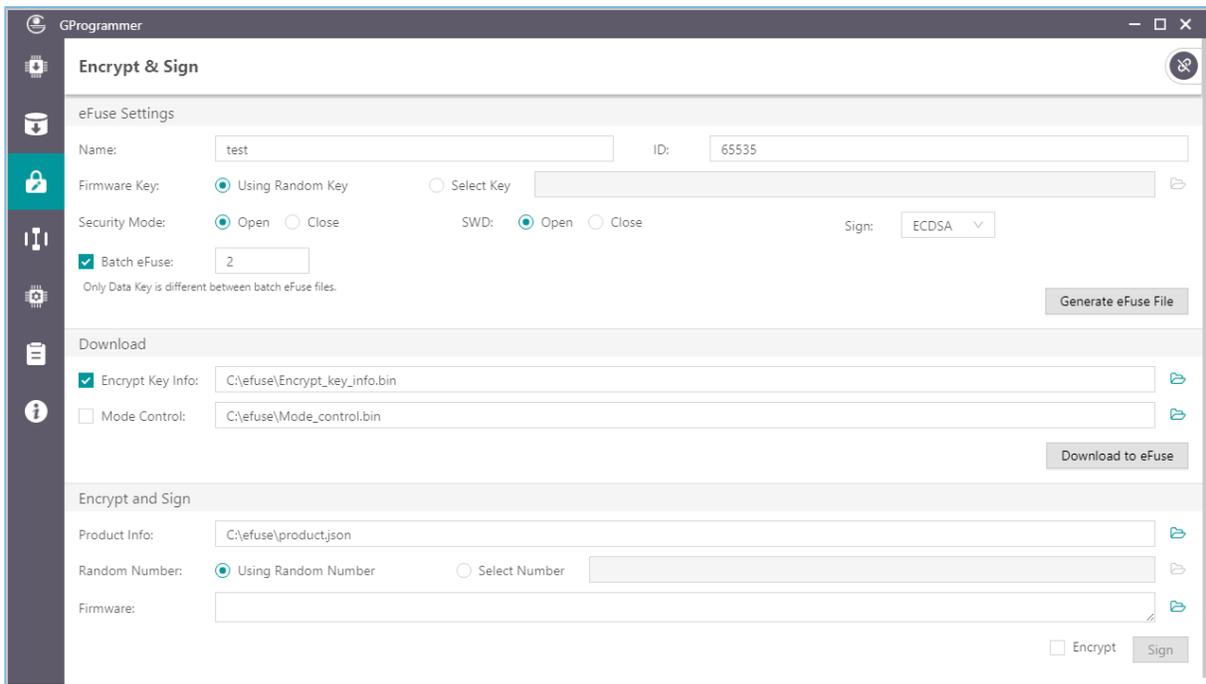


图 3-27 下载文件到eFuse

说明:

- eFuse不能重复下载。
- GR533x芯片不支持eFuse下载功能。

3.7.3 固件加密加签

GProgrammer可对固件（.hex/.bin格式）进行加密加签或者仅加签的操作，允许用户使用同一套产品信息和同一个Random Number操作一个或者多个固件文件。

- **Random Number:** 选择“Using Random Number”将使用软件自动生成的随机数进行加签，用户也可以点击“Select Number”选择自定义的数字bin文件进行加签。
- **Firmware:** 此栏中导入非加密固件。GProgrammer支持.hex格式和.bin格式的非加密固件导入，输出的加密加签或者仅加签的固件格式为.bin格式。添加多个固件时，各固件的路径之间用“;”隔开，如下图所示。

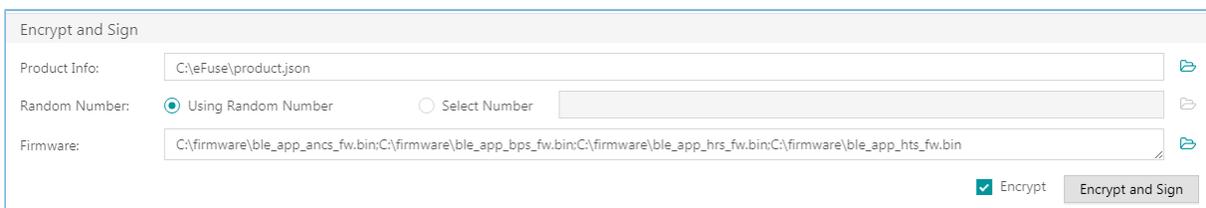


图 3-28 添加多个固件

- **Encrypt:** 如果勾选此复选框，则操作按钮变成“Encrypt and Sign”，点击操作按钮后将生成加密加签固件（.bin格式）；如果不勾选此复选框，则操作按钮变成“Sign”，点击操作按钮后将生成仅加签固件（.bin格式）。

- 加密加签后生成的文件包括：

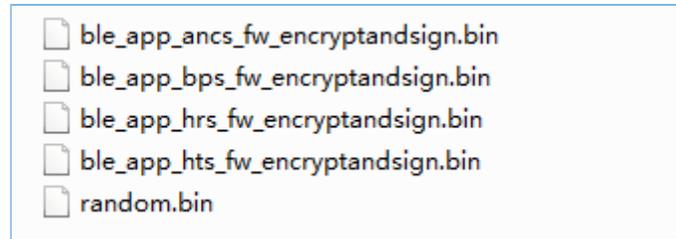


图 3-29 加密加签生成的文件

- 仅加签后生成的文件包括：

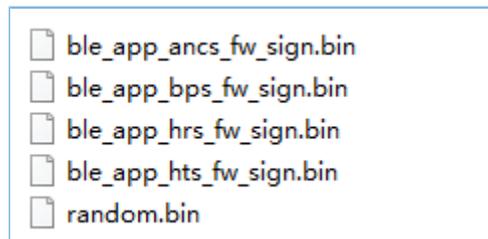


图 3-30 仅加签生成的文件

📖 说明:

软件随机生成的Random Number是一个用于加密算法的随机数。执行加密加签操作后，随机数文件（*random.bin*）会与加密加签后的固件文件保存在同一目录下。若用户需再次使用该随机数加密加签，则可浏览添加之前保存的随机数文件。

3.8 eFuse展示

点击左侧功能导航栏的图标 ，进入“eFuse Layout”（eFuse展示）页面。

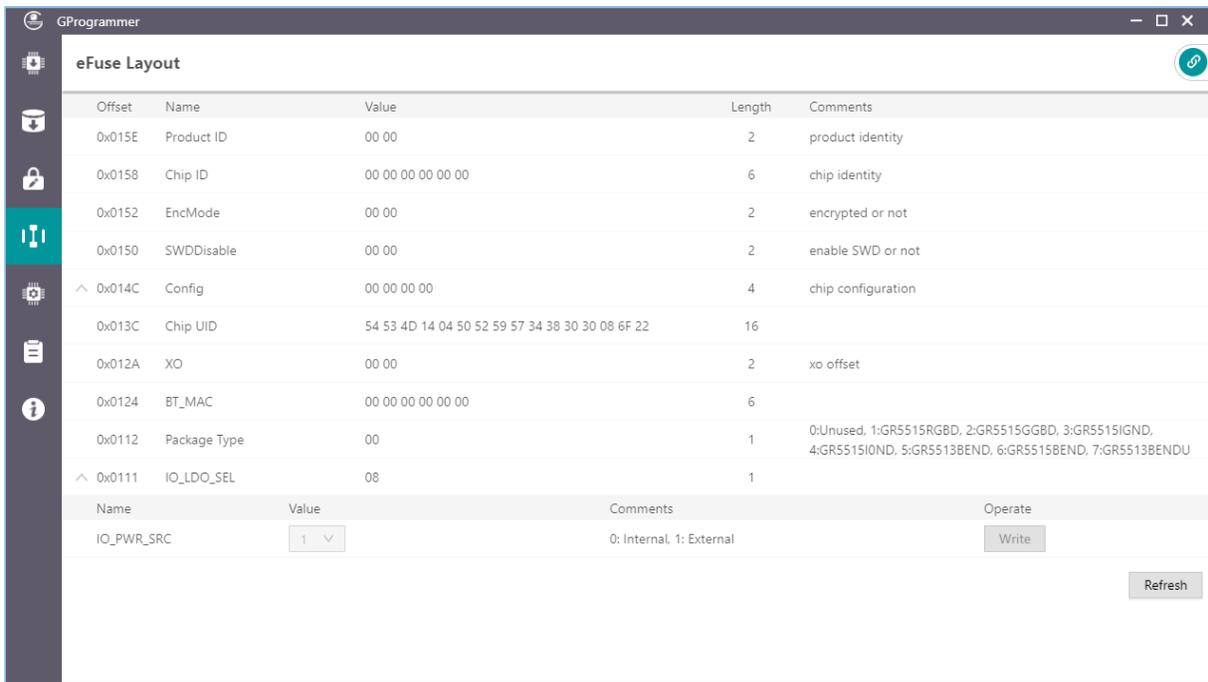


图 3-31 “eFuse Layout” 页面

用户可查看eFuse的存储布局信息，如：产品唯一标识符（“Product ID”）、芯片唯一标识符（“Chip ID”）、安全模式（“EncMode”）、SWD接口（“SWDDisable”）、配置信息（“Config”）字段的偏移地址、值、长度等信息。其中，“Config”和“IO_LDO_SEL”字段还包含多个位域。

各字段/位域的值信息，需点击“Refresh”按钮读取。

点击多位域“Offset”前的 ^ 图标，可展开其位域信息，如图 3-32 所示。如需收起位域信息，请点击 v 图标或直接双击“Config”和“IO_LDO_SEL”条目行。

用户可以更改“IO_LDO_SEL”字段中的“IO_PWR_SRC”位域，用于设置外设的供电方式。

说明:

只支持将0设置为1，不支持将1设置为0。

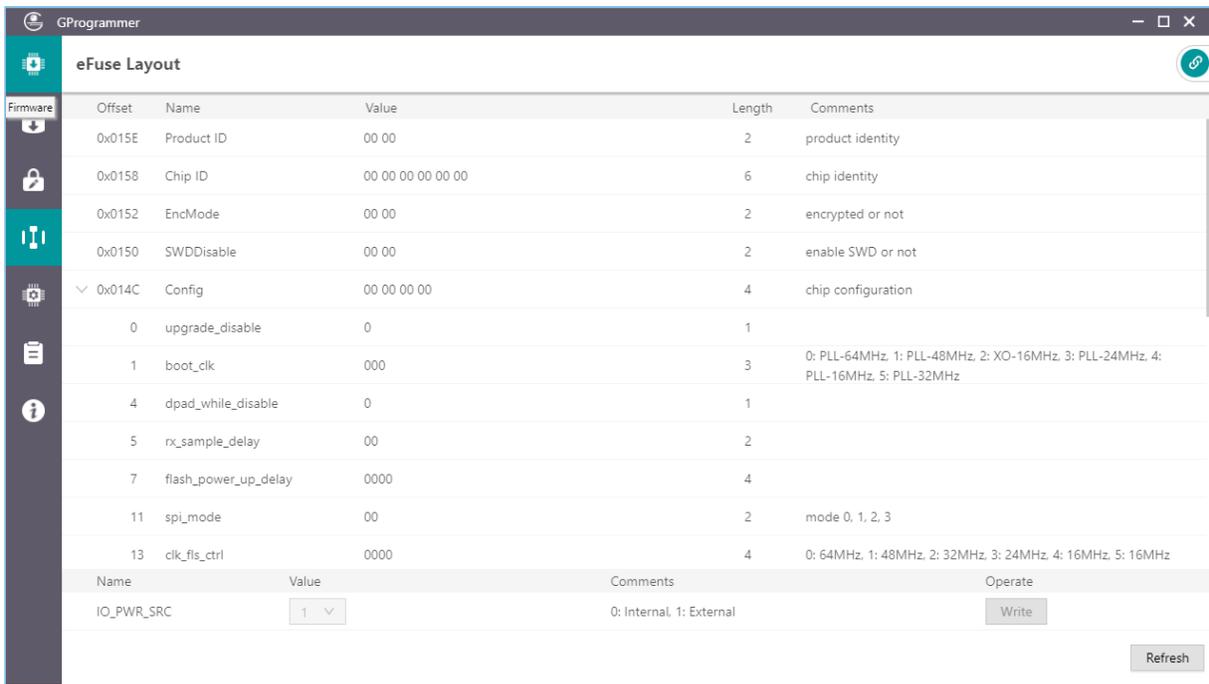


图 3-32 展开位域

说明:

列表中展示的字段和位域信息，均来自config文件夹下的`efuse_config.json`文件，并非芯片eFuse中存储的全部信息。

3.9 OTP展示

点击左侧功能导航栏的图标 ，进入“OTP Layout”（OTP展示）页面。

说明:

该操作页面仅适用于GR533x系列芯片。

列表中展示的字段和位域信息，均来自config文件夹下的`cairo_otp_config.json`文件，并非芯片OTP中存储的全部信息。

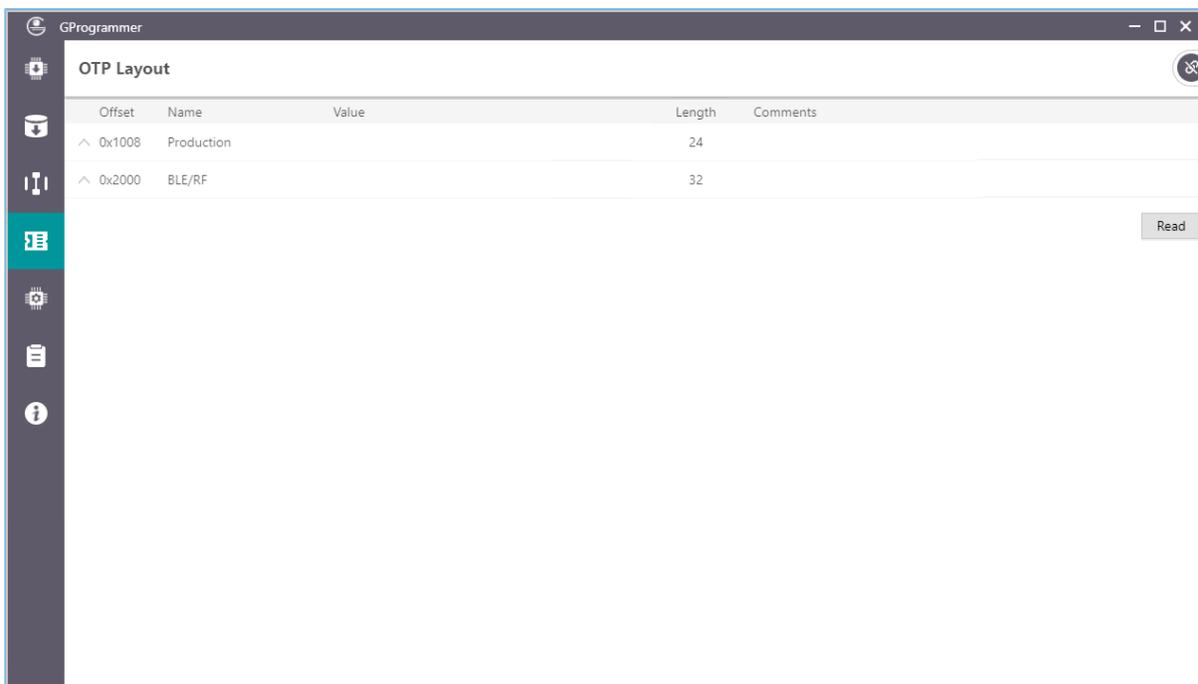


图 3-33 “OTP Layout” 页面

3.10 芯片配置

点击左侧功能导航栏的图标 , 进入“Chip Configuration”（芯片配置）页面。

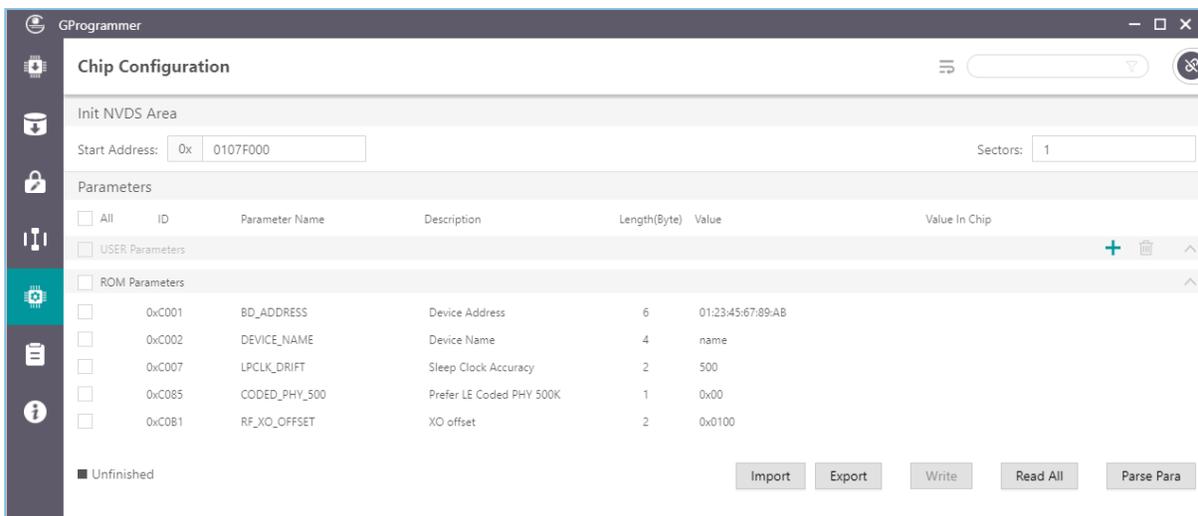


图 3-34 “Chip Configuration” 页面

用户可配置存储在芯片NVDS区域的参数，包括USER参数和ROM参数。

- **USER参数：**用户自定义的参数，允许用户新增、删除以及修改参数。
- **ROM参数：**芯片ROM中的参数，只允许用户修改参数值，不能新增或删除。

说明:

- 列表中默认显示的ROM参数，均来自config文件夹下的nvds_config.json文件，并非从芯片NVDS中实时读取。ROM参数的详细说明，参见表 3-6。
- 点击界面右上方的  按钮，可打开“自动换行”功能，列表将换行显示。
- 利用界面右上方的筛选功能，可快速查找参数。

表 3-6 NVDS ROM参数

ID	参数名称	描述
0xC001	BD_ADDRESS	设置蓝牙设备地址。
0xC002	DEVICE_NAME	设置设备名称。
0xC007	LPCLK_DRIFT	设置系统睡眠时钟精度等级SCA，取值范围为10 ~ 500，单位：ppm。
0xC085	CODED_PHY_500	设置默认Coded PHY，0为125 kbps，1为500 kbps。
0xC0B1	RF_XO_OFFSET	设置时钟校准参数，范围为0x000 ~ 0x1FF。

3.10.1 初始化NVDS

配置NVDS参数之前，用户需指定NVDS区域的起始地址（以4 KB大小对齐）和占用的Sector数量。

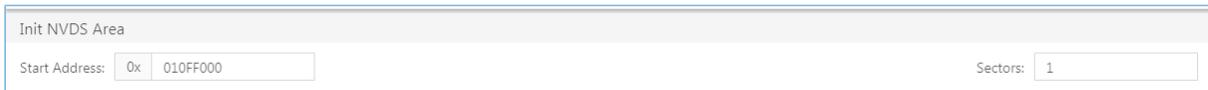


图 3-35 设置NVDS起始地址

若配置的NVDS区域与已存在的Firmware区域重叠，将无法完成NVDS初始化。

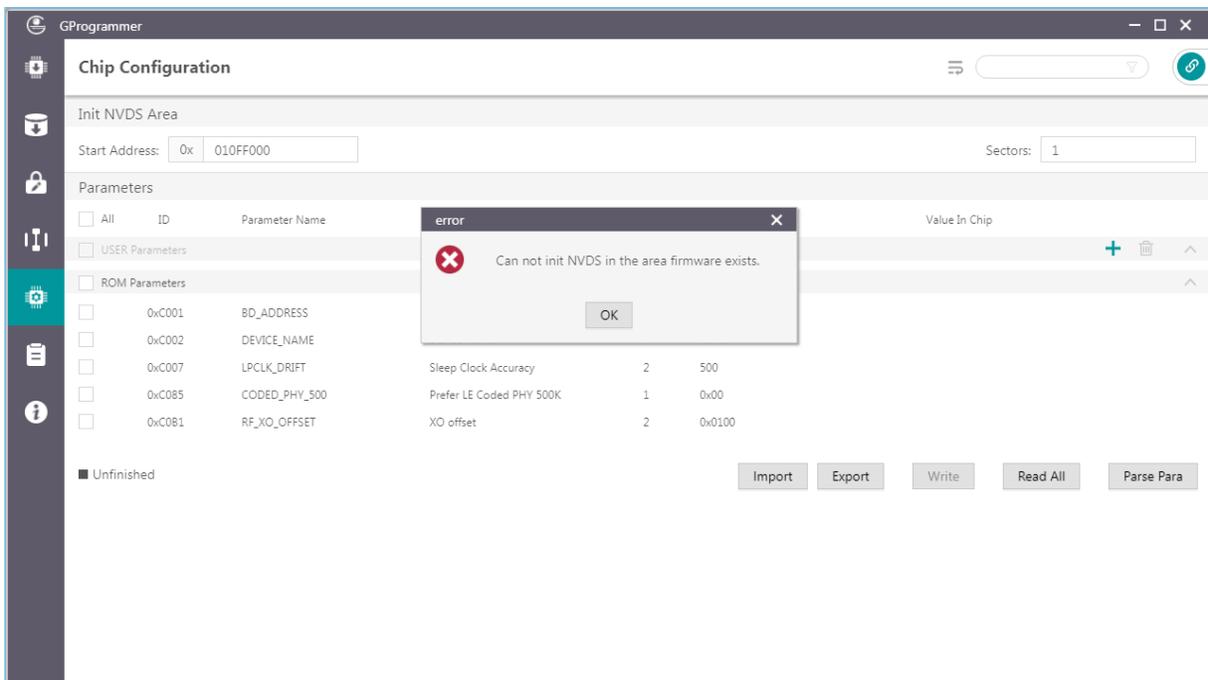


图 3-36 NVDS区域与固件区域重叠

3.10.2 读取参数

GProgrammer支持一次性读取当前芯片NVDS中的所有参数，并将读取的参数信息在“Parameters”列表中展示。

为了防止将芯片NVDS中已存在的参数错误覆盖，导致User APP无法正常运行，建议用户完成连接操作后，首先执行“Read All”操作，以确认参数是否已存在NVDS中。

GProgrammer以“Unfinished”、“Same”、“Different”表示参数状态，提示用户列表中的参数是否已存在NVDS中。

- “Unfinished”状态：字体为黑色，表示从NVDS中读取到列表中没有的参数（如图 3-37中ID为0x4000的参数），或者NVDS中已有此参数但长度不相等（如图 3-37中ID为0x4001的参数）。
- “Same”状态：字体为绿色，表示NVDS中有已有此参数，并且长度相等、值相同（如图 3-37中ID为0x4002的参数）。
- “Different”状态：字体为橙色，表示NVDS中已有此参数，长度相等但值不相同（如图 3-37中ID为0x4003的参数）。

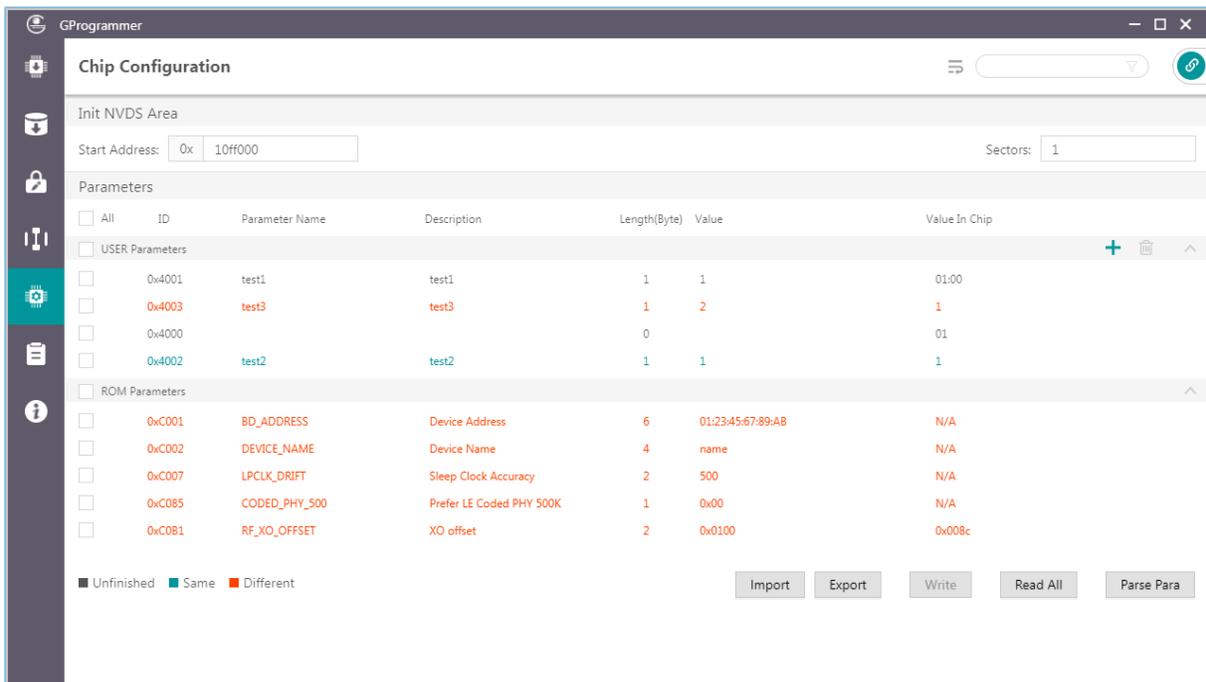


图 3-37 读取参数

3.10.3 写入参数

写入参数时，需先勾选待写入的参数项，然后点击“Write”按钮，将参数值写入NVDS。

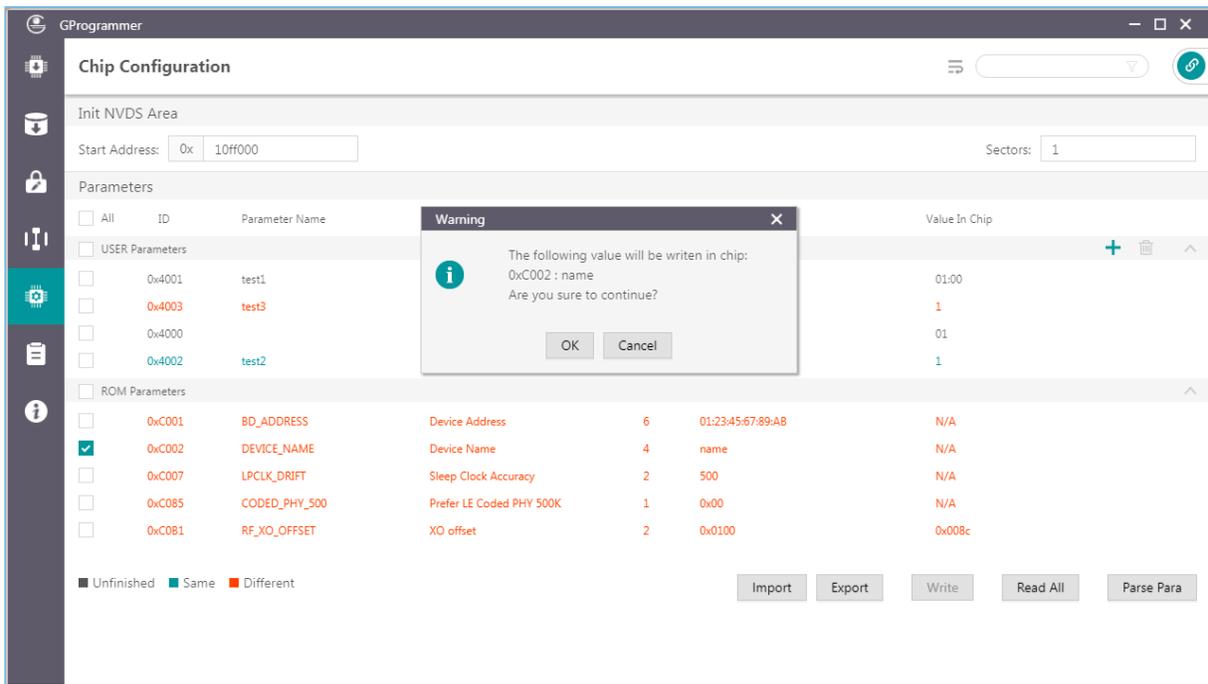


图 3-38 写入参数

说明:

- 若参数状态为“Unfinished”，则不允许直接写入芯片。
- 勾选多个参数，然后点击“Write”按钮，可实现参数的批量写入。
- 若勾选的多个参数中包含“Unfinished”状态的参数，则“Write”按钮将被禁用，不允许写入。

3.10.4 新增USER参数

新增USER参数时，请按以下步骤操作：

1. 点击+按钮，打开新增USER参数窗口。

2. 指定参数ID、名称、描述、类型、长度、值以及数据显示格式。

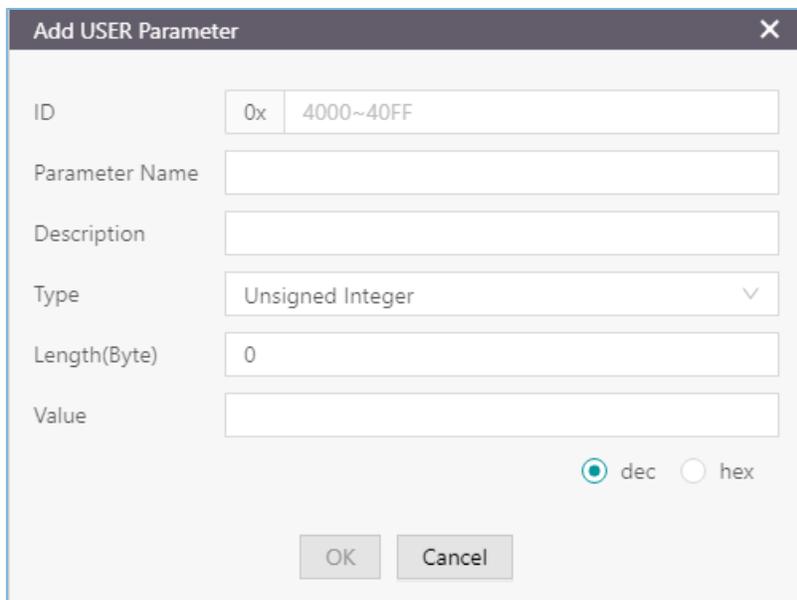


图 3-39 新增USER参数

3. 点击“OK”按钮确认，即完成新增参数。

说明:

- 新增参数的ID不能与列表中已有的参数ID相同。若相同，则将弹出警告提示信息，如图 3-40所示。
- 如果NVDS中无相同ID的参数，新增的参数被直接写入NVDS。
- 如果NVDS中有相同ID的参数，且参数长度相同，新增的参数值可被写入NVDS。
- 如果NVDS中有相同ID的参数，但参数长度不同，新增的参数值不会被写入NVDS，且参数状态显示为“Unfinished”。需修改参数长度，才能写入NVDS。

ID默认范围0x4000 ~ 0x40FF，可通过文件*nvds_common_config.json*自行调整有效范围为0x4000 ~ 0x7FFF，注意若范围配置过大将影响读取参数时间。

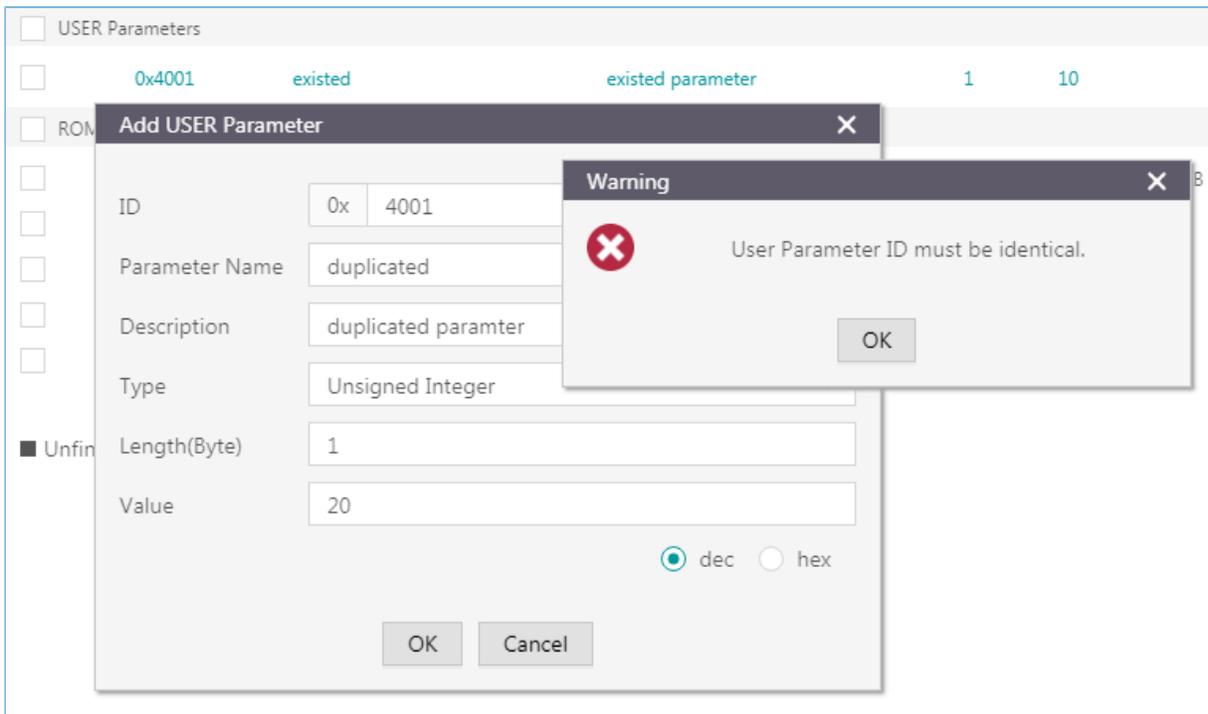


图 3-40 参数ID重复

3.10.5 修改参数

用户可修改NVDS中的USER参数和ROM参数。

ROM参数可修改参数名称、描述以及参数值，并且参数值的修改不会导致长度的变化（变长字符串类型除外）。

USER参数状态为“Same”或“Different”时，可修改的内容与ROM参数一致；USER参数状态为“Unfinished”时，还可修改参数类型和参数长度。

修改参数时，首先双击参数打开参数编辑窗口，编辑参数信息，然后点击“OK”按钮将修改后的参数信息写入NVDS中。

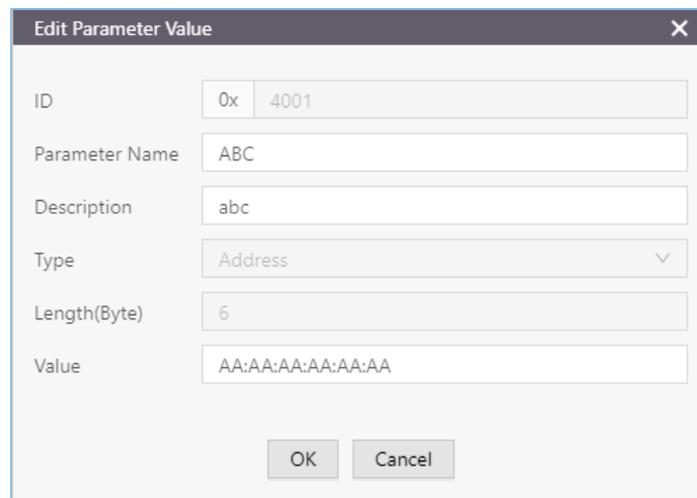


图 3-41 参数编辑窗口

说明:

若修改的参数状态为“Unfinished”，且修改后的长度与NVDS中的参数长度不一致，则执行写入操作后，参数仍将保持“Unfinished”状态，不会被写入NVDS。

3.10.6 删除USER参数

用户只可删除NVDS中的USER参数，不能删除ROM参数。

删除参数时，先勾选待删除参数，然后点击 **Delete** 图标，即可实现从NVDS中删除参数。

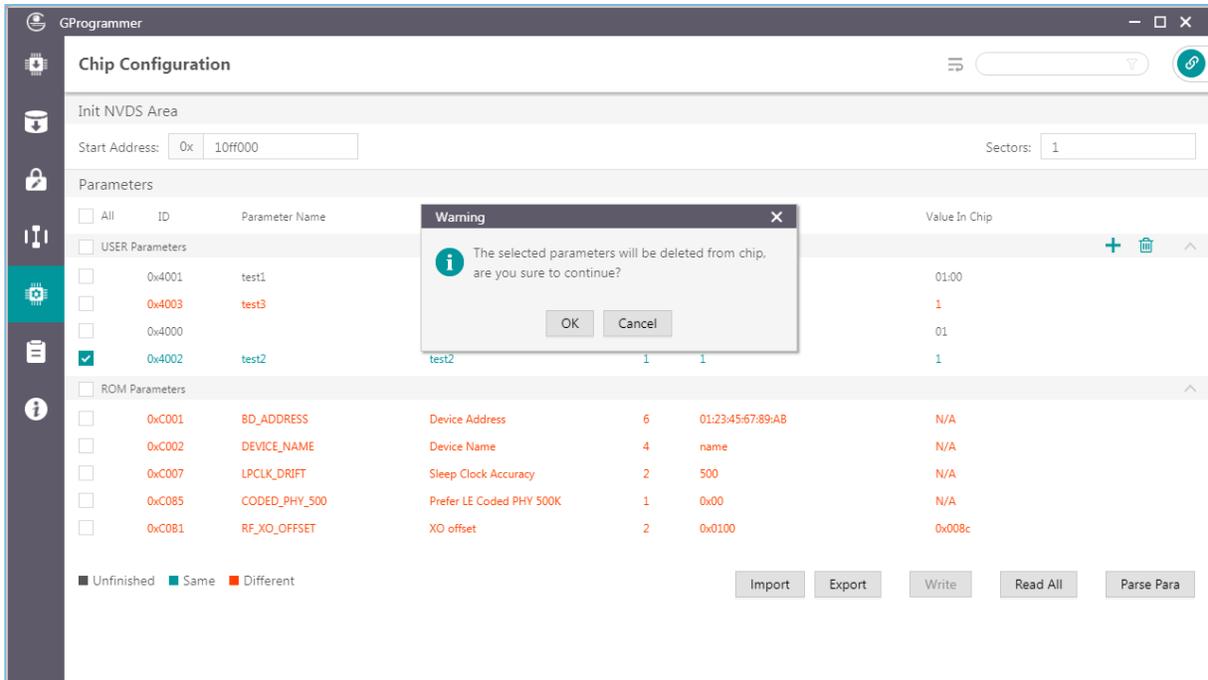


图 3-42 删除User参数

说明:

- 勾选多个User参数，然后点击 **Delete** 图标，可实现USER参数的批量删除。
- 如果勾选的多个参数中包含ROM参数，**Delete** 图标将被置灰（即“Remove”功能被禁用），不允许删除。

3.10.7 导入与导出

GProgrammer支持将参数列表中的所选数据（参数名称、描述、长度及值）导出为json格式的配置文件，以及从本地导入配置文件加载到列表显示。

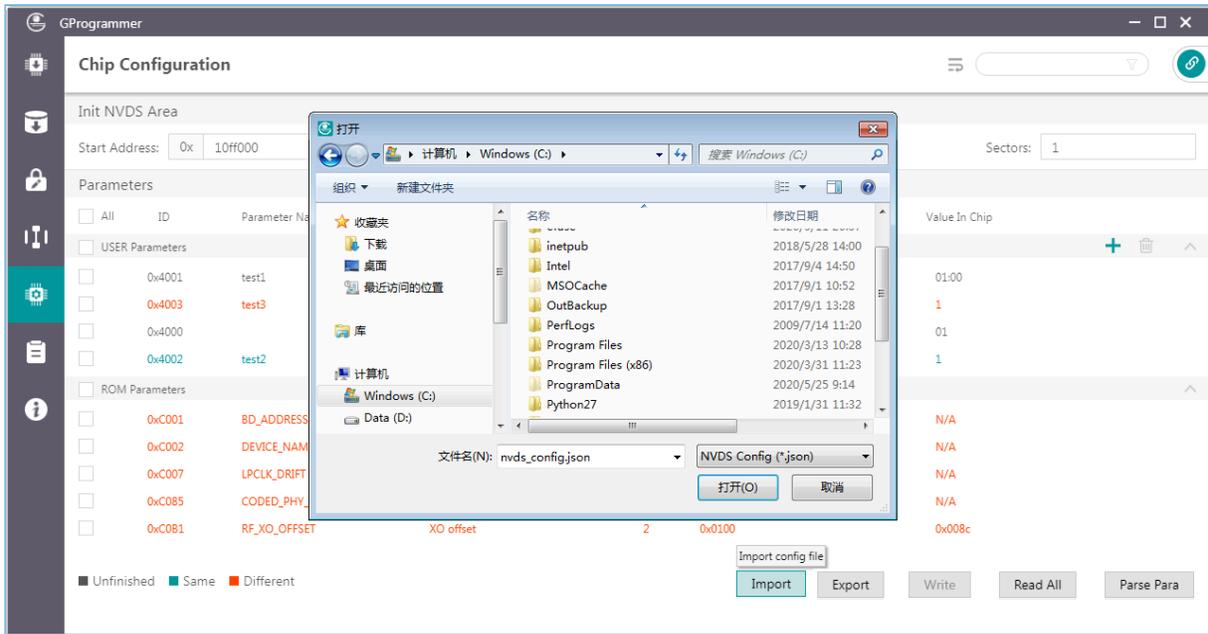


图 3-43 导入文件

说明:

- 导入配置文件后，文件中的参数会覆盖当前列表中的参数。
- 修改参数后，请及时导出配置文件并保存，以避免重复修改。
- 若存在“Unfinished”状态的参数，则“Export”按钮将被禁用，不允许导出。

3.10.8 解析NVDS数据

GProgrammer提供NVDS数据解析功能“Parse Para”，支持解析NVDS区域数据及文件数据，方便用户分析NVDS参数。

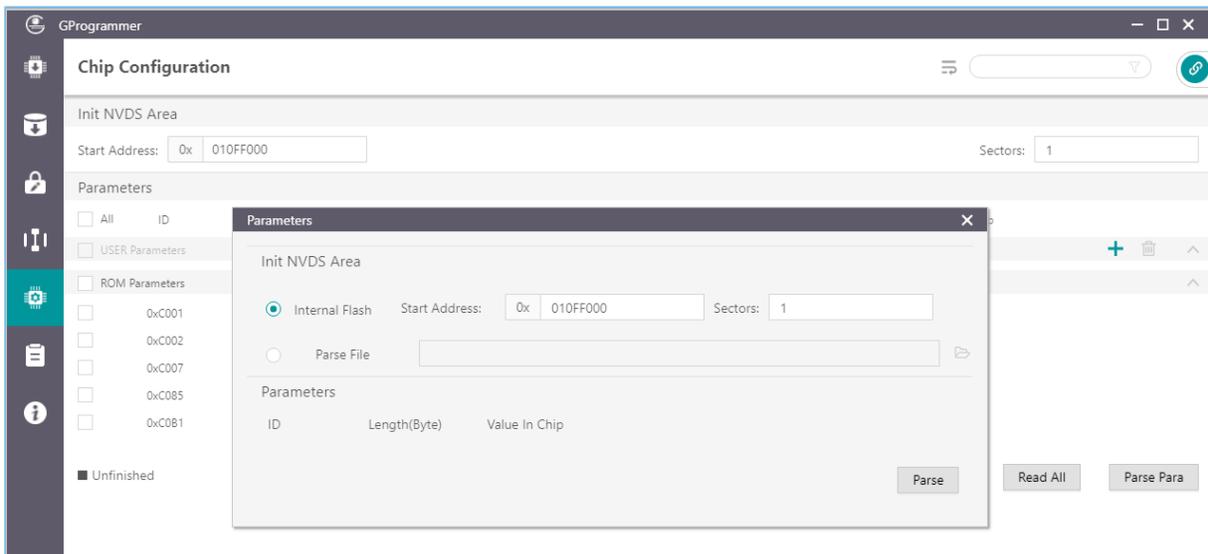


图 3-44 配置解析数据

- 解析NVDS区域数据：选择“Internal Flash”，并设置NVDS区域的起始地址（4 KB对齐）和占用的Sector数量。

待解析数据所在NVDS区域必须在配置的NVDS参数写入范围内（NVDS区域起始地址和占用Sector数量，详见3.10.1 初始化NVDS）。否则，会造成数据解析失败。

- 解析文件数据：选择“Parse File”，从本地添加已导出的NVDS区域数据文件。

说明:

- 仅支持解析非加密数据。
- 点击“Parse”按钮后，将按照Flash数据顺序进行参数解析与结果显示，参数值（Value in Chip）以小端模式显示。

3.11 设备日志

点击左侧功能导航栏的图标，进入“Device Log”（设备日志）页面。



图 3-45 “Device Log” 页面

用户可查看设备日志信息，一般为芯片运行中产生的错误信息。点击“Read”按钮，即可从芯片读取设备日志信息。

说明:

查看设备日志信息之前，需要确保已执行如下操作：

- 应用固件已将设备错误信息写入NVDS中（NVDS ID为A001~A010）。
- 在GProgrammer上正确初始化NVDS区域，与应用固件程序中定义一致。

另外，通过切换页面右上角的开关按钮，可选择日志信息的显示方式。

- ：开关按钮切换至“ascii”，表示以字符形式显示日志信息，如图 3-46 所示。
- ：开关按钮切换至“stream”，表示以字节流形式显示日志信息，如图 3-47 所示。

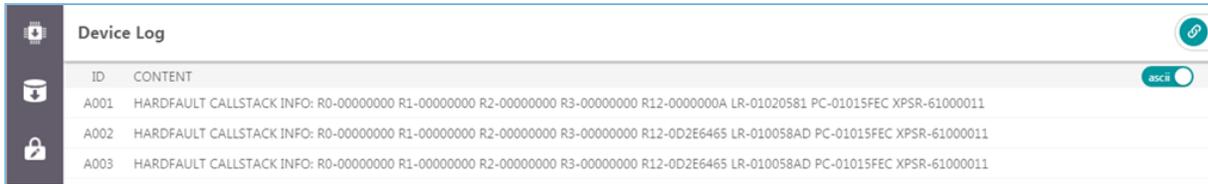


图 3-46 以字符形式显示日志

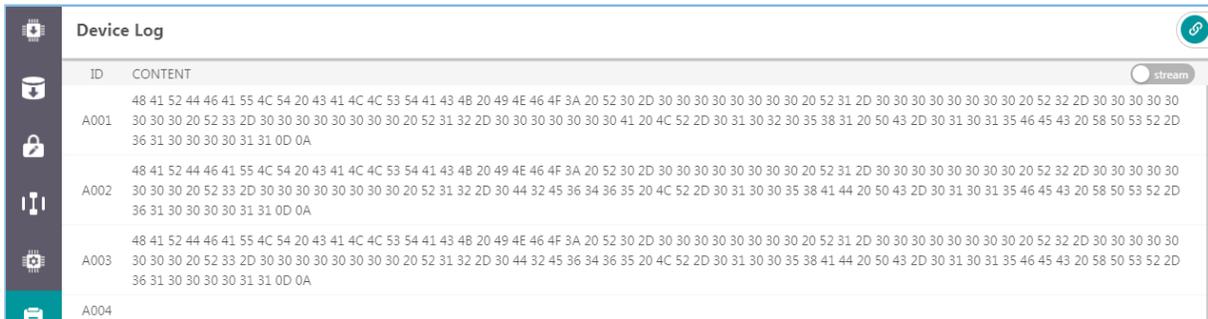


图 3-47 以字节流形式显示日志

3.12 命令行

GProgrammer提供2类命令程序：*GR5xxx_console.exe*和*GR5xxx_encrypt_signature.exe*。

说明：

GR5xxx表示芯片系列名称。

- *GR5xxx_console.exe*可方便用户直接使用命令行完成固件下载和Flash擦除等基本功能操作。
- *GR5xxx_encrypt_signature.exe*可方便用户直接使用命令行完成固件的加密加签或仅加签操作。

3.12.1 GR5xxx_console.exe

命令程序*GR5xxx_console.exe*的使用步骤如下：

1. 选择“开始菜单 > 命令提示符”，或者在运行对话框中输入“cmd”再回车，打开“命令提示符”窗口。
2. 使用cd命令进入GProgrammer安装目录。
3. 输入*GR5xxx_console.exe command*命令行完成相应操作。关于“command”具体描述，参见表 3-7。

表 3-7 GR5xxx_console.exe支持的命令

command	功能描述	命令格式及参数说明	备注
program	下载固件文件至芯片内部Flash	<pre>program <firmware file path> <run immediately:y n> <flash start address(hex)> <flash size> <product type> <S/N(optional)></pre> <p>参数说明:</p> <ul style="list-style-type: none"> • <firmware file path>: 待下载固件文件路径 • <run immediately:y n>: 下载完成后是否立即运行固件 	<p>命令中相同的参数，说明如下:</p> <ul style="list-style-type: none"> • <flash start address(hex)>: Flash起始地址，可输入值及含义如下: <ul style="list-style-type: none"> ◦ 0x01000000: GR551x Flash起始地址 ◦ 0x00200000: GR5526/GR533x/GR5525 Flash起始地址 • <flash size>: Flash大小（单位KB），即使用的芯片型号Flash大小，可查看图 3-3 “Flash” 列。 <p>说明:</p> <p>若使用的芯片Flash为0 KB，则写入使用的外挂Flash大小。</p> <ul style="list-style-type: none"> • <product type>: 芯片系列，可输入值及含义如下: <ul style="list-style-type: none"> ◦ 0: GR551x系列芯片 ◦ 1: GR5526系列芯片 ◦ 2: GR533x系列芯片 ◦ 4: GR5525系列芯片 • <S/N(optional)>: S/N号为可选参数，若PC端仅连接一个设备可不指定S/N号，若PC端连接多个设备，可使用S/N号指定需操作的设备。
erase	擦除指定地址范围内的芯片内部Flash数据	<pre>erase <start address<hex>> <end address<hex>><force erase when conflict with firmware/bootinfo:y n> <flash start address(hex)> <flash size> <product type> <S/N(optional)></pre> <p>参数说明:</p> <ul style="list-style-type: none"> • <start address<hex>>: 擦除区域的起始地址（16进制） • <end address<hex>>: 擦除区域的结束地址（16进制） • <force erase when conflict with firmware/bootinfo:y n>: 当和固件、bootinfo区域冲突之后，是否强制擦除 	
eraseall	擦除全部芯片内部Flash数据	<pre>eraseall <product type> <S/N(optional)></pre>	
download	下载数据文件至芯片内部Flash	<pre>download <data file path> <start address<hex>><force download when conflict with firmware/bootinfo:y n> <flash start address(hex)> <flash size> <product type> <S/N(optional)></pre> <p>参数说明:</p> <ul style="list-style-type: none"> • <data file path>: 待下载数据文件路径 • <start address<hex>>: 下载区域的起始地址（16进制） • <force download when conflict with firmware/bootinfo:y n>: 当和固件、bootinfo区域冲突之后，是否强制下载 	

command	功能描述	命令格式及参数说明	备注
writeefuse	将Encrypt Key Info和Mode Control文件写入eFuse	<pre>writeefuse <Encrypt Key Info file Path> <Mode Control file Path> <product type> <S/N(optional)></pre> 参数说明： <ul style="list-style-type: none"> • <Encrypt Key Info file Path>: Encrypt Key Info文件路径 • <Mode Control file Path>: Mode Control文件路径 	
readefuse	读取eFuse Layout信息	readefuse <S/N(optional)>	
reset	复位芯片系统	reset <product type> <S/N(optional)>	
generate	转换固件文件为芯片可用.bin格式文件	<pre>generate <input firmware file path> <output firmware file path> <flash start address(hex)> <flash size> <product type></pre> 参数说明： <ul style="list-style-type: none"> • <input firmware file path>: 输入文件路径（IDE工具编译所生成的.hex/.bin文件） • <output firmware file path>: 输出文件路径 	
device	获取J-Link设备信息 (输出设备名称, S/N号)	<pre>device <display count(optional)></pre> 参数说明： <ul style="list-style-type: none"> • <display count(optional)>: 显示设备信息数量（可选参数） 	
load	下载固件（支持Flash固件、RAM固件）	<pre>load <firmware file path> <erase type> <run immediately:y n> <S/N(optional)></pre> 参数说明： <ul style="list-style-type: none"> • <firmware file path>: 待下载固件文件路径 • <erase type>: 擦除类型 <ul style="list-style-type: none"> ◦ “sector”：擦除bootinfo区域、固件所用Flash区域 ◦ “chip”：全片擦除 ◦ “none”：无额外擦除，仅擦除固件所用Flash区域 • <run immediately:y n>: 下载完成后是否立即运行固件 	
dump	从内部Flash导出数据	dump <start address(hex)> <size(hex)> <file path> <S/N(optional)>	
help	查看所有命令帮助信息	help	

例如，使用program命令下载一个固件文件到GR551x系列芯片，并设置下载完成后立即运行该固件，输入的命令为：

```
GR5xxx_console.exe program "D:/test/test_fw.bin" y "0x01000000" 1024 0
```

示例命令，重点参数内容说明如下：

- "D:/test/test_fw.bin"：待下载的目标固件.bin所在路径。
- "0x01000000" 1024 0：依次为待烧录目标固件的Flash起始地址为0x01000000、Flash大小为1024 KB、芯片系列为GR551x。

program命令执行过程中会实时显示下载进度。

📖 说明：

GProgrammer命令程序GR5xxx_console.exe和界面程序无法同时使用。

3.12.2 GR5xxx_encrypt_signature.exe

命令程序GR5xxx_encrypt_signature.exe的使用步骤如下：

1. 选择“开始菜单 > 命令提示符”，或者在运行对话框中输入“cmd”后回车，打开“命令提示符”窗口。
2. 使用cd命令进入GProgrammer安装目录。
3. 输入GR5xxx_encrypt_signature.exe --parameter完成操作。

常用参数及对应描述，详见表 3-8，全部参数可使用GR5xxx_encrypt_signature.exe --help命令进行查看。

表 3-8 GR5xxx_encrypt_signature.exe的常用参数列表

参数	描述	备注
operation	操作类型，可输入值及含义如下： <ul style="list-style-type: none"> • encryptandsign: 加密加签 • sign: 仅加签 	
firmware_key	用于加密加签/仅加签的firmware.key文件所在路径	文件所在路径，对应3.7.1 eFuse设置章节，点击“Generate eFuse File”按钮后选择的文件保存路径。
signature_key	用于加密加签/仅加签的sign.key文件所在路径	
signature_pub_key	用于加密加签/仅加签的sign_pub.key文件所在路径	
product_json_path	用于加密加签/仅加签的product.json文件所在路径	
rand_number	用于加密加签/仅加签的random.bin文件所在路径	
ori_firmware	待加密加签/仅加签的原始固件路径	
output	加密加签/仅加签后的固件保存路径	
random_output	加密加签/仅加签使用的随机数的保存路径	
base_addr	Flash起始地址，可输入值及含义如下：	

参数	描述	备注
	<ul style="list-style-type: none"> • 0x01000000: GR551x Flash起始地址 • 0x00200000: GR5526/GR533x/GR5525 Flash起始地址 	
flash_size	<p>Flash大小（单位KB），即使用的芯片型号Flash大小，可查看图 3-3 “Flash” 列。</p> <p>说明</p> <p>若使用的芯片Flash为0 KB，则写入使用的外挂Flash大小。</p>	
product_type	<p>芯片类型，可输入值及含义如下：</p> <ul style="list-style-type: none"> • 0: GR551x系列芯片 • 1: GR5526系列芯片 • 2: GR533x系列芯片 • 4: GR5525系列芯片 	
help	显示帮助信息	

例如，在GR551x系列芯片上执行固件加密加签，输入的命令示例如下：

```
GR5xxx_encrypt_signature.exe --operation="encryptandsign" --firmware_key="D:/test/eFuse/firmware.key" --signature_key="D:/test/eFuse/sign.key" --signature_pub_key="D:/test/eFuse/sign_pub.key" --product_json_path="D:/test/eFuse/product.json" --rand_number="D:/test/eFuse/random.bin" --ori_firmware="D:/test/firmware/test_fw.bin" --output="D:/test/firmware_encryptAndSign/test_fw_encryptAndSign.bin" --random_output="D:/test/firmware_encryptAndSign/random.bin" --base_addr="0x01000000" --flash_size="1024" --product_type="0"
```

其中参数值路径“D:/test/eFuse/”为3.7.1 eFuse设置章节，点击“Generate eFuse File”按钮后用户选择的文件保存路径。更多参数介绍，请查看表 3-8。

- --ori_firmware="D:/test/firmware/test_fw.bin": 原始固件的路径。
- --output="D:/test/firmware_encryptAndSign/test_fw_encryptAndSign.bin": 保存加密加签后固件的路径。
- --base_addr="0x01000000" --flash_size="1024" --product_type="0": 依次表示待烧录目标固件的Flash起始地址为0x01000000、Flash大小为1024 KB、芯片系列为GR551x。
- --rand_number 参数使用请参见3.7.3 固件加密加签章节Random Number说明。

运行命令后，开始执行加密加签操作。

3.12.3 用户自定义Windows脚本

用户可自定义编写Windows脚本文件，来调用命令程序。在GProgrammer的安装目录下的GR5xxx_script目录中，存放了2个示例脚本文件。

📖 说明:

GR5xxx表示芯片系列名称。

`encryptAndSignatureFirmware.bat`脚本使用同级目录中`firmware_origin.bin`文件和eFuse目录，生成加密加签的固件保存到`firmware_encryptAndSign\firmware_encryptAndSign.bin`路径中。

`program_Firmware_EncryptAndSign.bat`脚本先擦除全部内部Flash，然后下载`firmware_encryptAndSign\firmware_encryptAndSign.bin`路径中的固件到内部Flash中。

3.13 帮助信息

点击左侧功能导航栏的图标 ，进入“Help”（帮助信息）页面。

GProgrammer为用户提供的帮助支持，包括：

- About GProgrammer
查看GProgrammer软件版本信息以及功能特性。
- Feedback
如在GProgrammer使用中有任何问题，可登录[开发者社区](#)反馈问题。
- About Goodix
如需了解更多Goodix的产品信息，可访问Goodix官网www.goodix.com。

3.14 常见问题

3.14.1 启动GProgrammer后，软件界面显示为空白

- 问题描述：
安装GProgrammer软件后，双击桌面图标启动软件。软件启动后，软件界面无任何内容，显示为空白。
- 问题分析：
导致该问题的原因可能为：
 - 原因1：GProgrammer安装在系统盘（如C盘），而系统盘对文件读写操作进行了保护，导致工具启动后界面显示异常。
 - 原因2：GProgrammer 安装文件丢失或受损（如文件被加密等），导致工具启动后界面显示异常。
- 处理方法：
用户可选择以下处理方法解决该问题：
 - 方案1：以管理员方式运行GProgrammer。
 - 方案2：重新安装GProgrammer，并将其安装在非系统盘。