



## GProgrammer用户手册

版本： 3.2

发布日期： 2024-08-21

版权所有 © 2024 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：深圳市福田区腾飞工业大厦B座13层

电话：+86-755-33338828      邮编：518000

网址：[www.goodix.com](http://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
- 技术支持工程师

## 版本说明

本手册为第18次发布，对应的软件版本为GProgrammer V2.0.1。

## 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"><li>• “芯片配置”章节sectors软件可修改。</li><li>• “命令行”章节GR551x_console.exe命令行程序的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。</li><li>• “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。</li></ul> |
| 1.7 | 2020-08-30 | <ul style="list-style-type: none"><li>• “芯片选型”章节GR551x系列新增GR5515I0ND芯片。</li><li>• “固件”章节变更“Delete”和“Startup”图标。</li></ul>                                                                                                                                   |
| 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"><li>• “eFuse设置”章节补充生成的文件不能修改的说明。</li><li>• “导入与导出”章节补充文件导出说明。</li><li>• “设备日志”章节补充查看设备日志的前提操作说明。</li><li>• “eFuse展示”章节增加“IO_LDO_SEL”字段。</li><li>• “命令行”章节erase和download命令删除nvds参数。</li></ul>                              |
| 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。                                                                                                                                                                                                                                                 |

| 版本  | 日期         | 修订内容                                                                                                                                                                                                                                                                                                                                                                                          |
|-----|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2.4 | 2021-09-06 | 更新芯片选型界面截图。                                                                                                                                                                                                                                                                                                                                                                                   |
| 2.5 | 2022-02-20 | <ul style="list-style-type: none"><li>• 增加适配GR5526系列芯片的描述。</li><li>• “Firmware”和“Encrypt &amp; Sign”界面，支持添加.hex格式固件文件。</li><li>• “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。</li><li>• 配置外部Flash时，新增SPI Type配置项“QSPI2”并修改相关配置项。</li><li>• 更新GR5xxx_console.exe和GR5xxx_encrypt_signature.exe支持的命令。</li></ul>                                                                     |
| 2.6 | 2023-01-19 | 删除GR5515I0ND。                                                                                                                                                                                                                                                                                                                                                                                 |
| 2.7 | 2023-02-03 | <ul style="list-style-type: none"><li>• 更新GR5xxx_console.exe支持的指令，新增“device”、“load”指令。</li><li>• GR5xxx_encrypt_signature.exe指令参数增加“rand_number”描述。</li><li>• 新增Linux版本 GProgrammer- Version.tar.bz2 说明。</li><li>• “配置Flash”章节，新增外部Flash配置注意事项</li><li>• 新增芯片型号查看及重选功能。</li><li>• SWD连接方式下，新增目标板S/N获取与选择功能。</li><li>• 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。</li><li>• 新增NVDS数据解析功能。</li></ul> |
| 2.8 | 2023-03-30 | <ul style="list-style-type: none"><li>• UART连接新增支持GR5526的USB连接参数“Detect USB。”</li><li>• 优化GR5526下载数据到Boot区域。</li><li>• 新增NVDS参数ID范围配置。</li><li>• 更新GR5xxx_console.exe支持的指令，新增“dump”指令。</li><li>• 更新Feedback邮箱为开发者社区网址。</li></ul>                                                                                                                                                              |
| 2.9 | 2023-04-20 | <ul style="list-style-type: none"><li>• 新增支持多款芯片的相关描述。</li><li>• GR533x系列新增“OTP Layout”功能。</li></ul>                                                                                                                                                                                                                                                                                          |
| 3.0 | 2023-09-22 | <ul style="list-style-type: none"><li>• 新增选型芯片。</li><li>• 更新“安装软件”、“硬件连接”章节少许描述。</li><li>• 更新“下载数据”章节关于“强制下载数据功能”的限制描述。</li><li>• 更新GR5xxx_console.exe指令参数。</li><li>• 新增ECDSA签名算法支持。</li><li>• 增加“常见问题”。</li></ul>                                                                                                                                                                            |
| 3.1 | 2023-09-24 | <ul style="list-style-type: none"><li>• 新增选型芯片。</li><li>• GR533x 系列新增固件“加签”功能。</li></ul>                                                                                                                                                                                                                                                                                                      |
| 3.2 | 2024-08-21 | <ul style="list-style-type: none"><li>• 新增选型芯片。</li><li>• 更新GR5xxx_console命令描述。</li><li>• 更新“常见问题”描述。</li></ul>                                                                                                                                                                                                                                                                               |

# 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..... 14

            3.6.1.3 下载数据..... 16

            3.6.1.4 Dump数据..... 17

        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.....                    | 37 |
| 3.12.1.1 通用参数.....                            | 38 |
| 3.12.1.2 外部Flash配置参数.....                     | 38 |
| 3.12.1.3 帮助命令.....                            | 39 |
| 3.12.1.4 生成烧录固件.....                          | 40 |
| 3.12.1.5 合并固件.....                            | 40 |
| 3.12.1.6 解锁/上锁Flash.....                      | 41 |
| 3.12.1.7 复位设备.....                            | 41 |
| 3.12.1.8 烧录固件.....                            | 41 |
| 3.12.1.9 写入Flash.....                         | 42 |
| 3.12.1.10 更新Flash.....                        | 43 |
| 3.12.1.11 读取Flash.....                        | 44 |
| 3.12.1.12 擦除Flash.....                        | 44 |
| 3.12.1.13 写入NVDS.....                         | 45 |
| 3.12.1.14 读取NVDS.....                         | 46 |
| 3.12.1.15 擦除NVDS.....                         | 46 |
| 3.12.1.16 写入eFuse.....                        | 47 |
| 3.12.1.17 读取eFuse.....                        | 47 |
| 3.12.1.18 写入OTP.....                          | 48 |
| 3.12.1.19 读取OTP.....                          | 48 |
| 3.12.1.20 擦除OTP.....                          | 49 |
| 3.12.1.21 展示设备列表.....                         | 49 |
| 3.12.2 GR5xxx_encrypt_signature.....          | 50 |
| 3.12.3 用户自定义Windows脚本.....                    | 51 |
| 3.13 帮助信息.....                                | 52 |
| 3.14 常见问题.....                                | 52 |
| 3.14.1 启动GProgrammer后，软件界面显示为空白.....          | 52 |
| 3.14.2 GProgrammer连接失败.....                   | 53 |
| 3.14.3 Linux环境下启动GProgrammer时，加载模块失败.....     | 53 |
| 3.14.4 Linux环境下GProgrammer软件界面显示为空白或连接失败..... | 53 |

## 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。

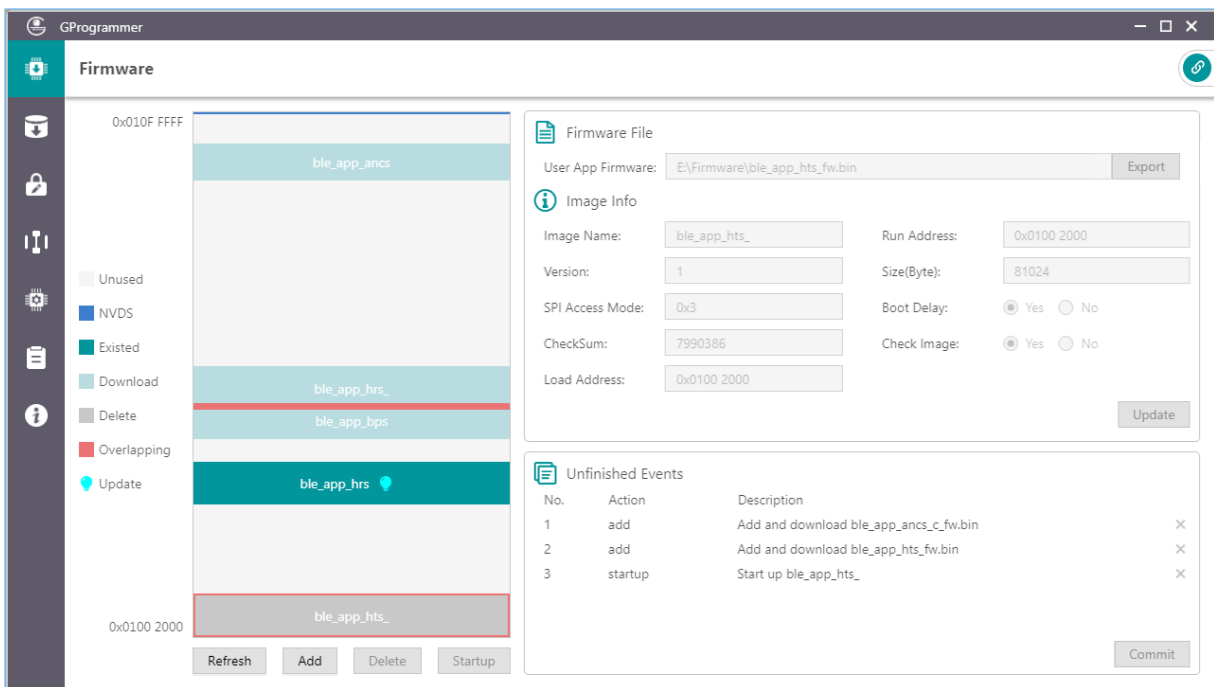


图 1-1 GProgrammer软件主界面

## 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.tar.bz2*。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。

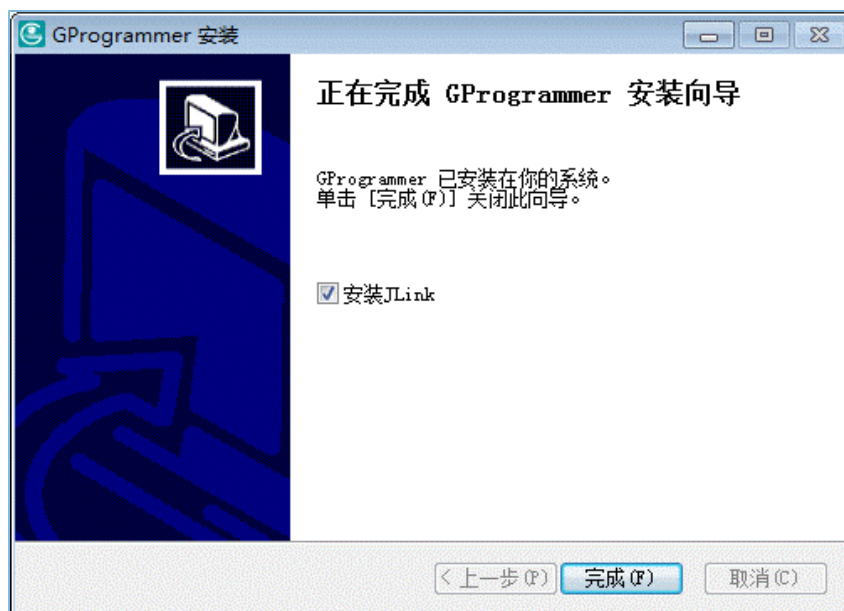


图 2-2 安装JLink提示

#### 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。

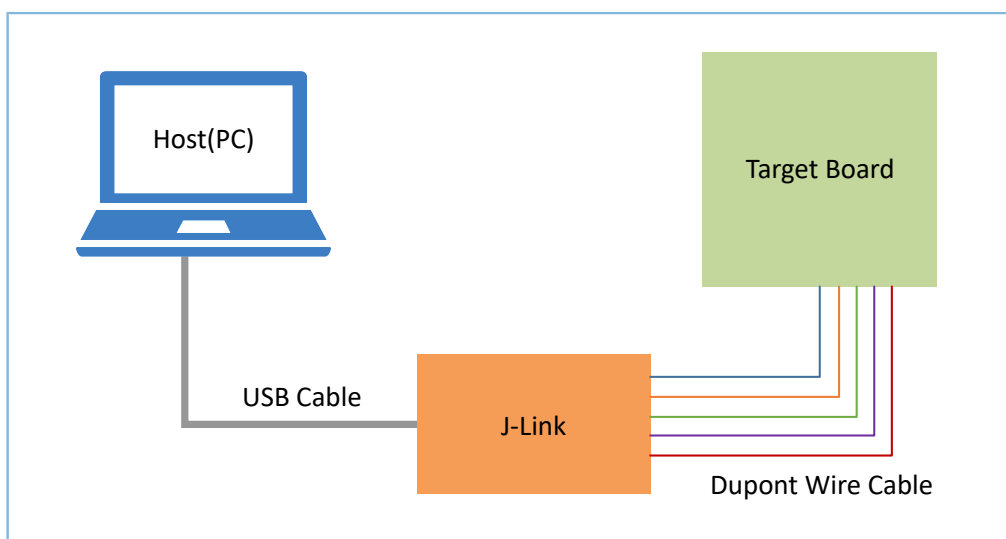


图 3-1 SWD硬件连接示意图

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。

- 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。

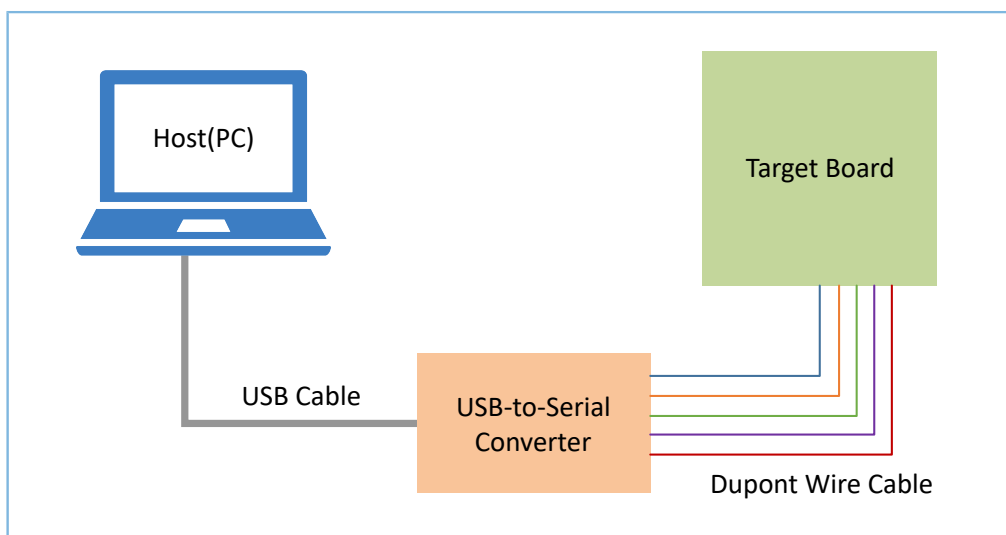


图 3-2 URAT硬件连接示意图

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。

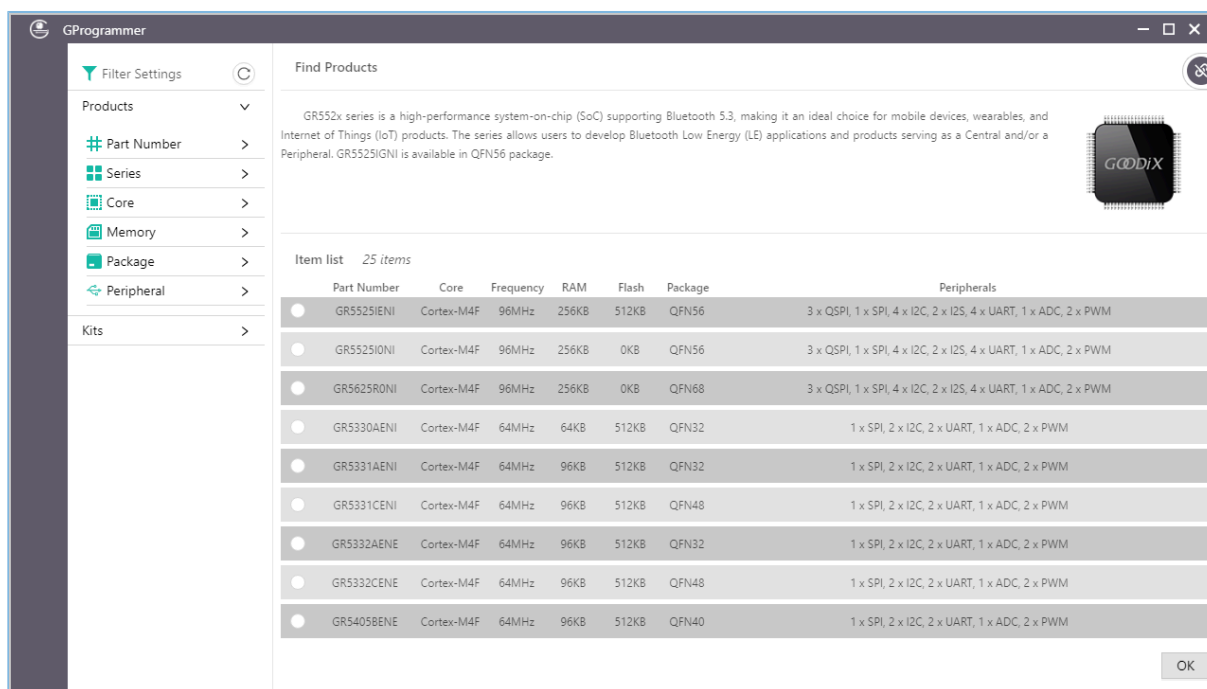


图 3-3 芯片选型界面

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。

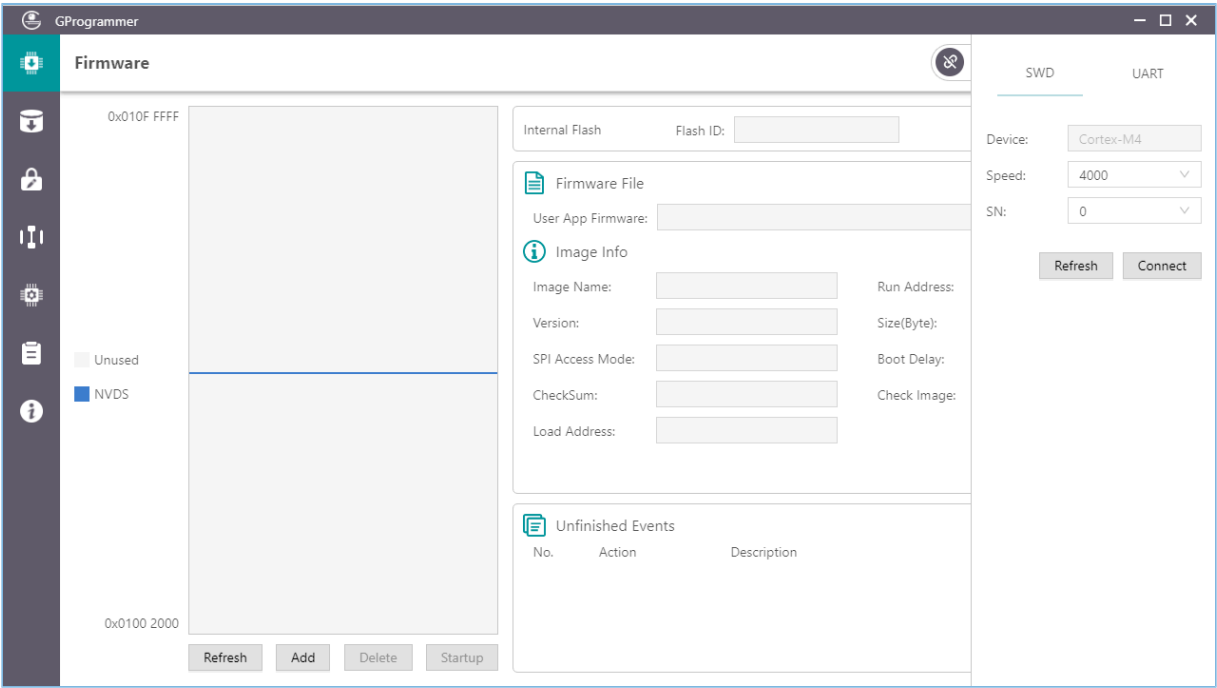


图 3-4 GProgrammer软件主界面

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<br>• 鼠标移入后，可查看当前选择的芯片型号<br>• 如需更换所选芯片型号，可点击此图标返回芯片选型页面。<br>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系列和GR5405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”。<br><ul style="list-style-type: none"><li>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。</li><li>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。</li></ul> 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。 |

SWD

UART

Device:

Cortex-M4

Speed:

4000

▼

SN:

0

▼

Refresh

Connect

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                                  |
|--------------|---------------------------------------|
|              | <div></div>                           |
| Port:        | <div></div>                           |
| Baudrate:    | <div>921600</div>                     |
| Parity:      | <div>None</div>                       |
| DataBits:    | <div>8</div>                          |
| StopBits:    | <div>1</div>                          |
| FlowControl: | <div>none</div>                       |
| Detect USB:  | <div>false</div>                      |
|              | <div>Refresh</div> <div>Connect</div> |

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。

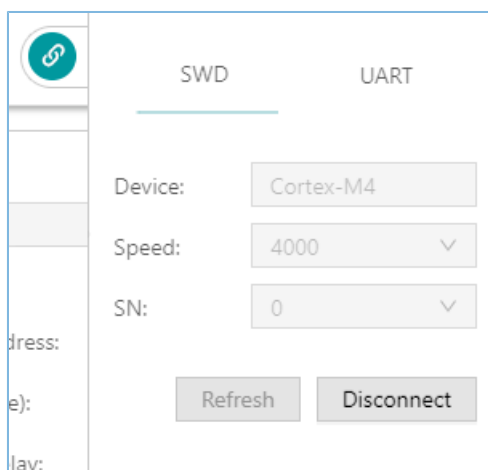


图 3-7 断开连接

## 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。

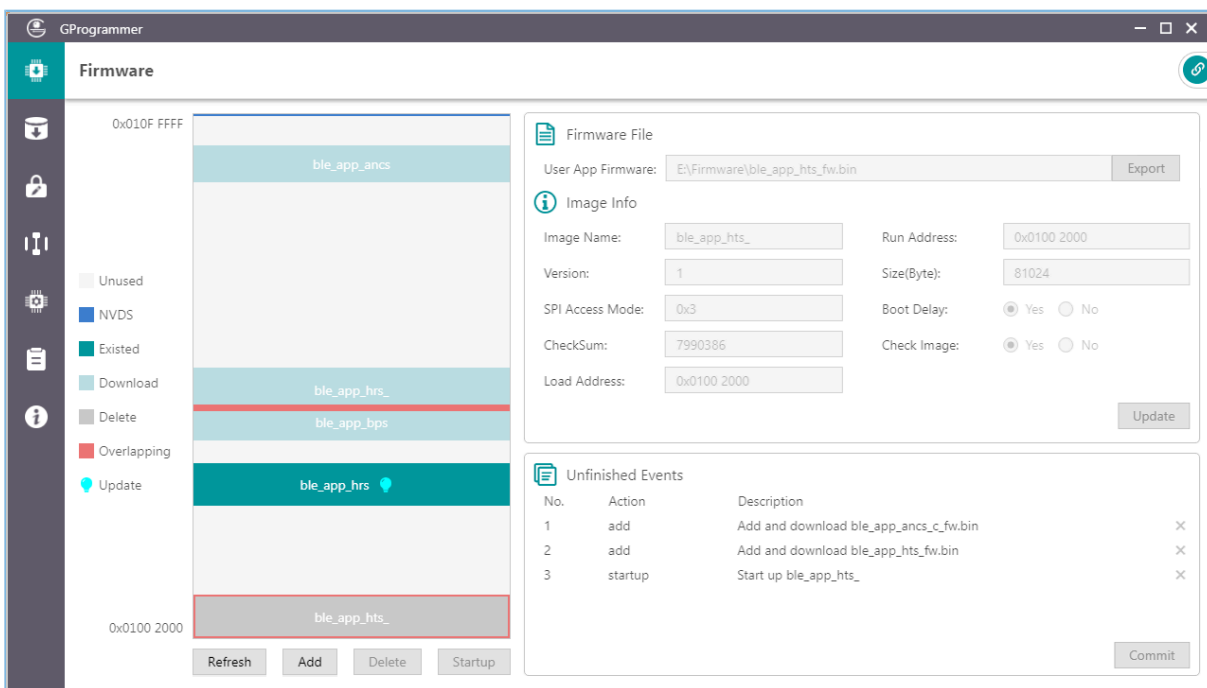


图 3-8 “Firmware” 页面

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。



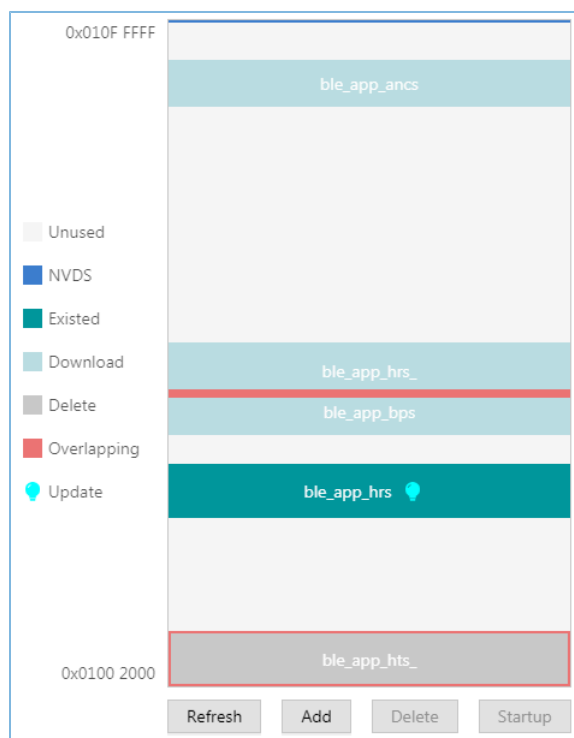


图 3-9 Flash Firmware Layout 示意图

- （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:

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。

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。

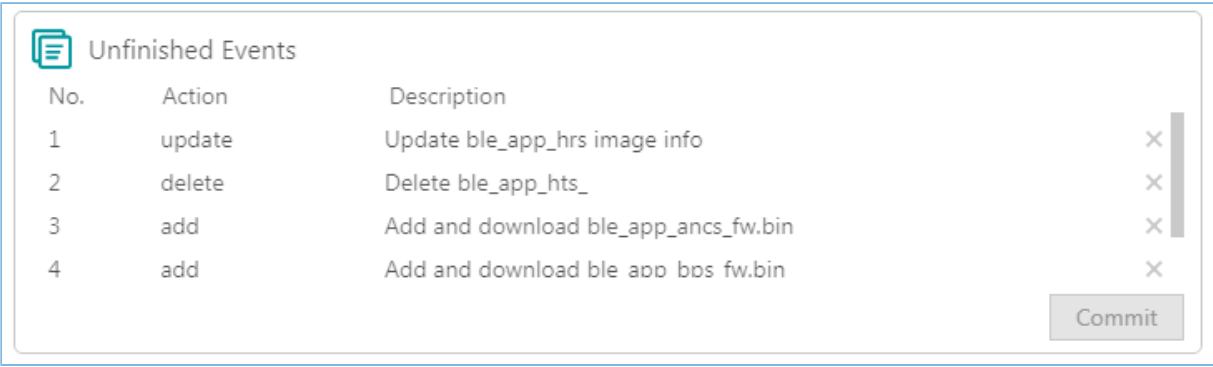


图 3-10 操作列表

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><b>说明：</b></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><b>说明：</b></p> <p>“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。</p>     |
| 设置为启动项 | Startup | <p>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。</p>                                                      |

| 操作名称 | 操作按钮                                                                              | 描述                                                                                                                                                                                                                                                                                             |
|------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 更新固件 |  | <p>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。</p> <p>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。</p> |

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。

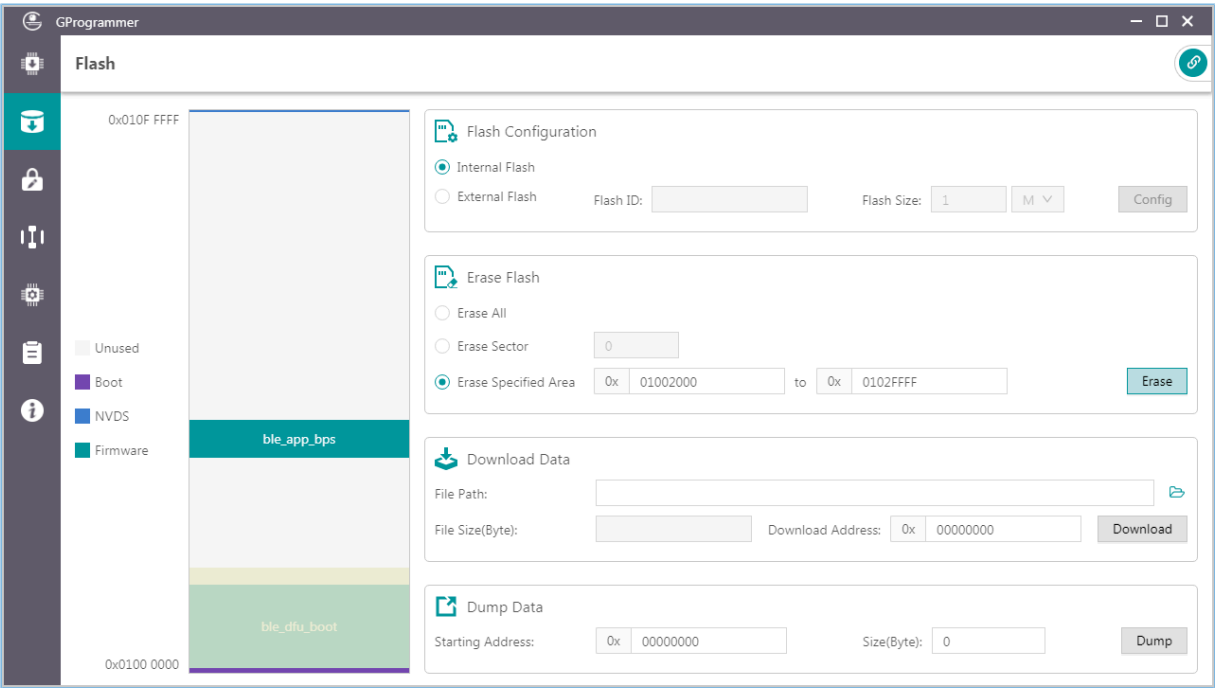


图 3-11 “Flash” 页面

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。

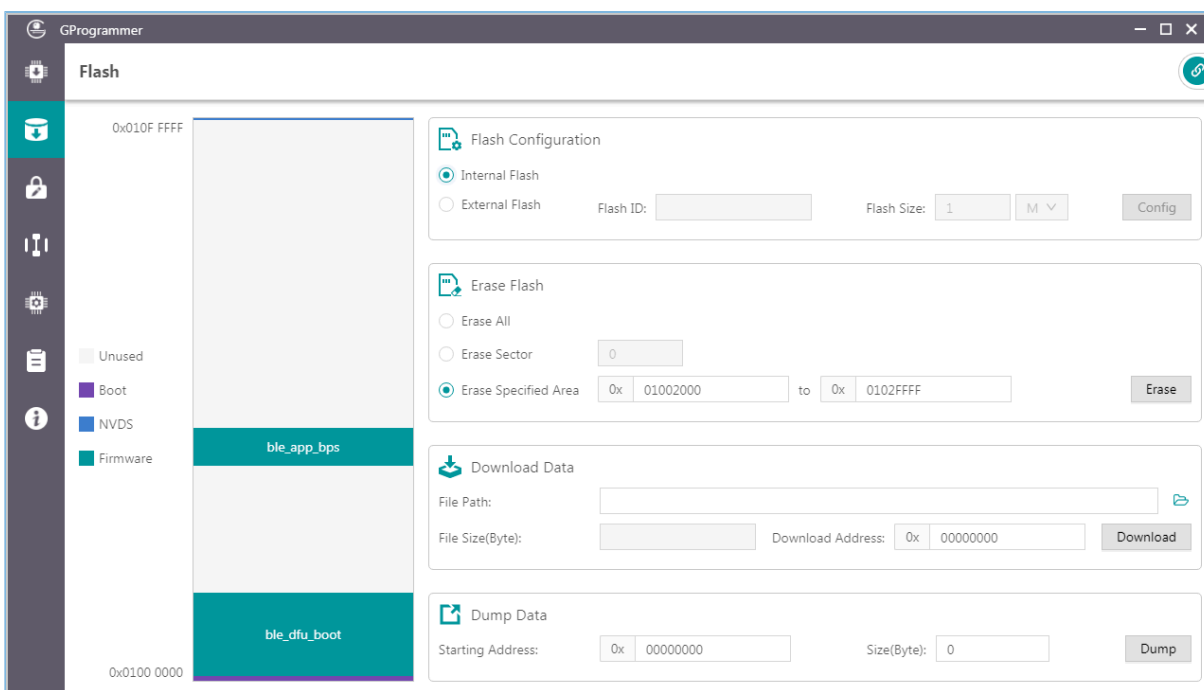


图 3-12 选择内部Flash

### 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。

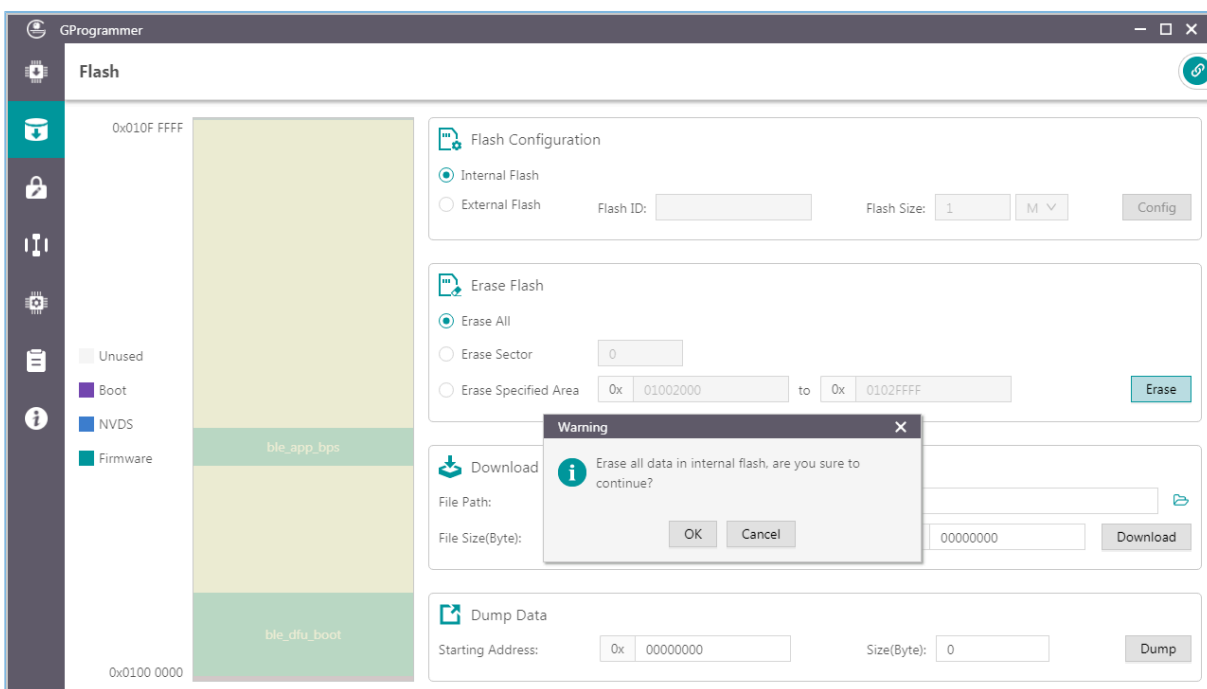


图 3-13 全片擦除

- Sector擦除（Erase Sector）

Sector擦除，指擦除指定的Sector区域（4 KB大小）。

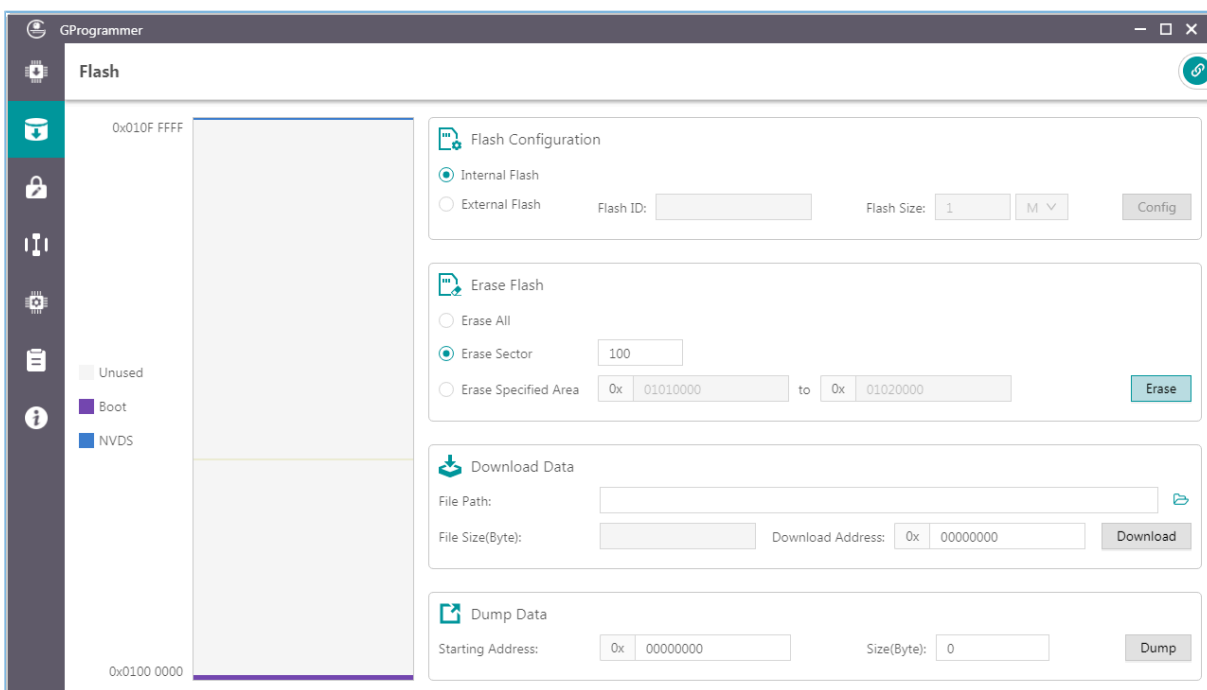


图 3-14 Sector擦除

- 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。

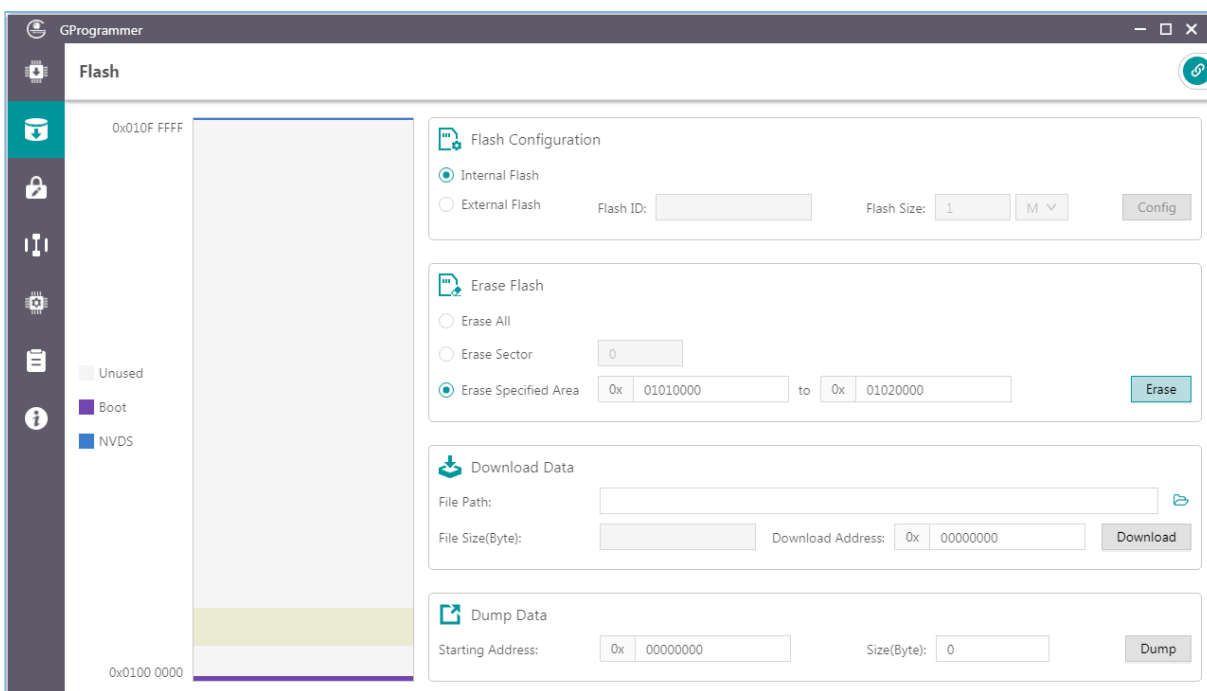


图 3-15 区域擦除

### 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。

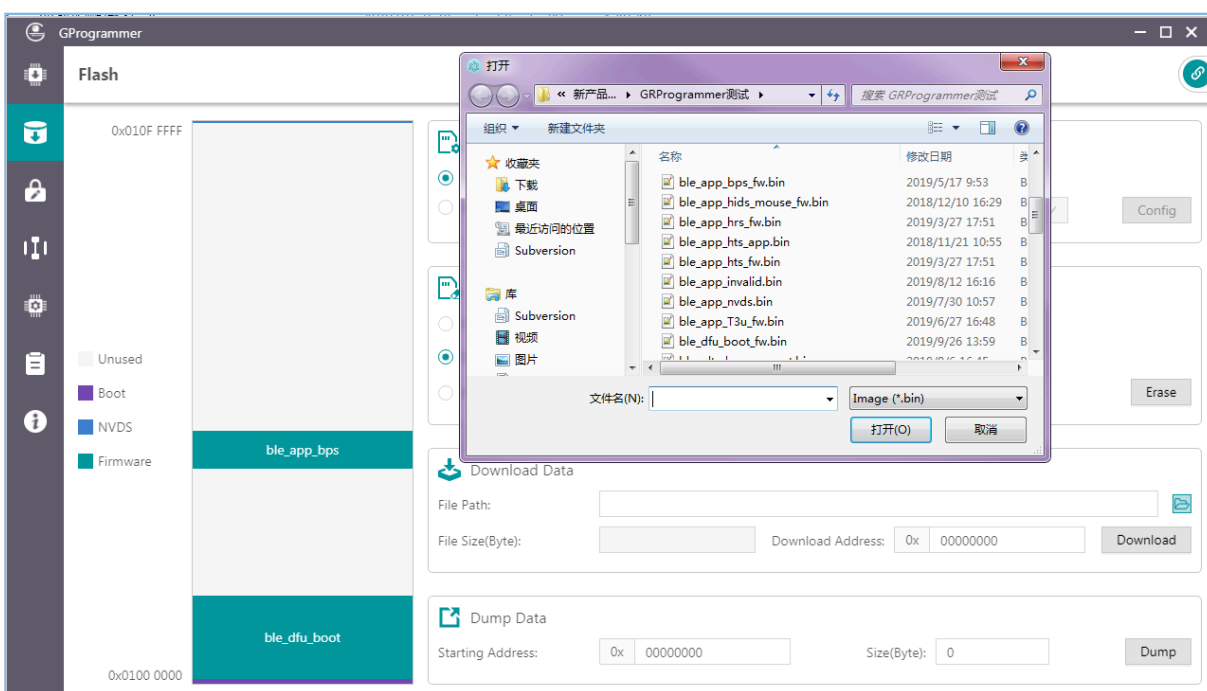


图 3-16 浏览选择数据文件

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。

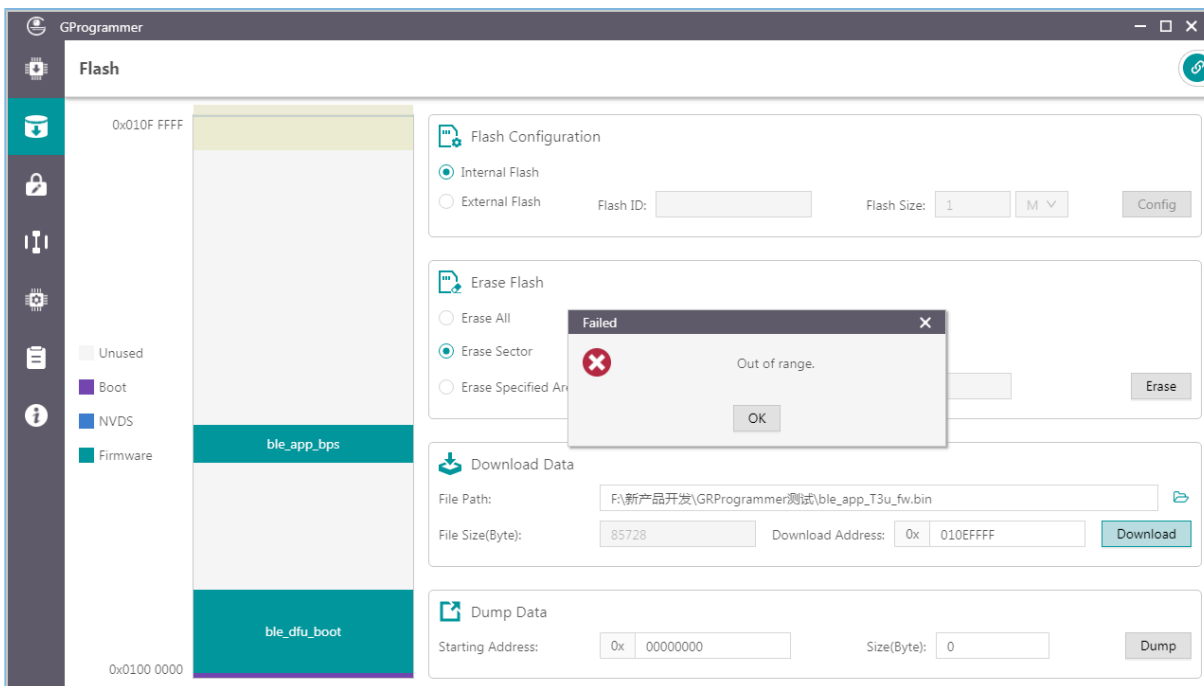


图 3-17 Flash 溢出

#### 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。

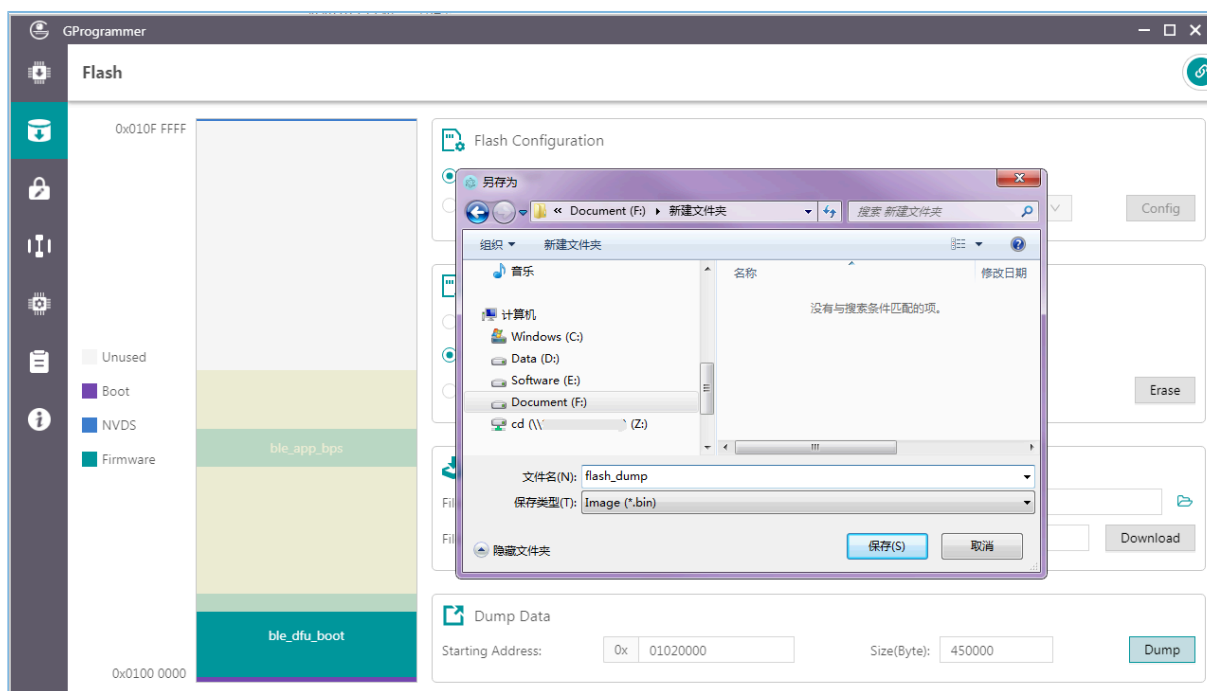


图 3-18 Dump数据

## 3.6.2 外部Flash

### 3.6.2.1 配置Flash

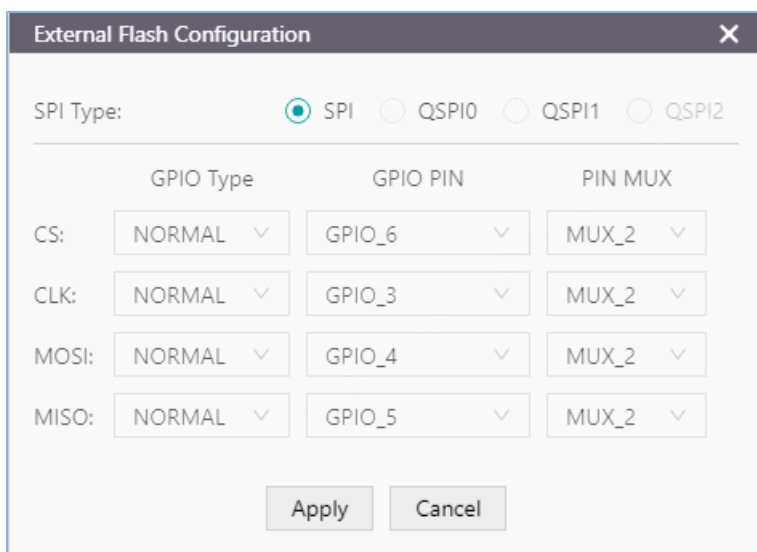
当用户操作外部Flash时，首先需在“Flash Configuration”栏选择“External Flash”；然后点击“Config”按钮，打开外部Flash配置窗口。用户可根据实际情况配置PIN脚。

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。

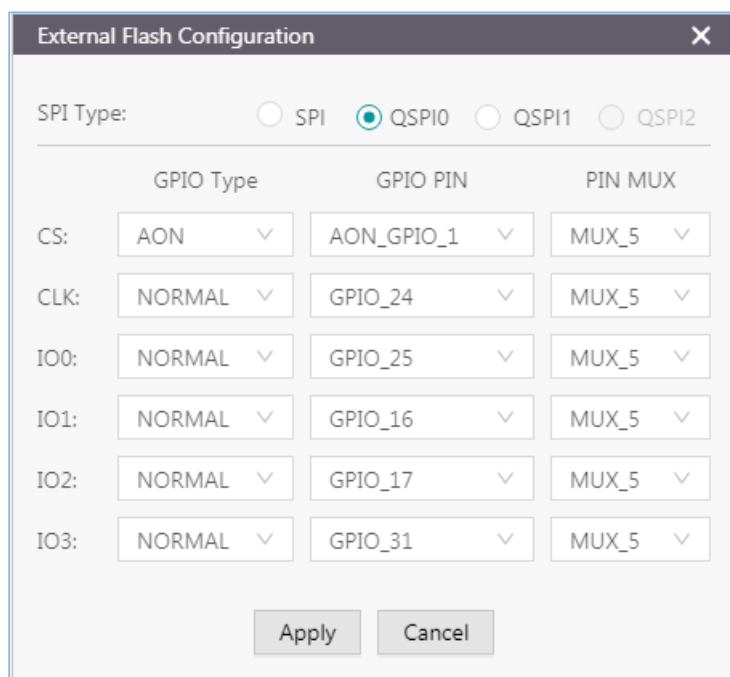




The dialog box titled "External Flash Configuration" shows the "SPI Type" set to "SPI". Below, a table lists the GPIO configurations for CS, CLK, MOSI, and MISO. All GPIO Types are set to "NORMAL", and all PIN MUX values are set to "MUX\_2".

|       | 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   |

图 3-19 SPI配置项



The dialog box titled "External Flash Configuration" shows the "SPI Type" set to "QSPI0". Below, a table lists the GPIO configurations for CS, CLK, IO0, IO1, IO2, and IO3. CS is set to "AON", while others are "NORMAL". All PIN MUX values are set to "MUX\_5".

|      | 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   |

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。

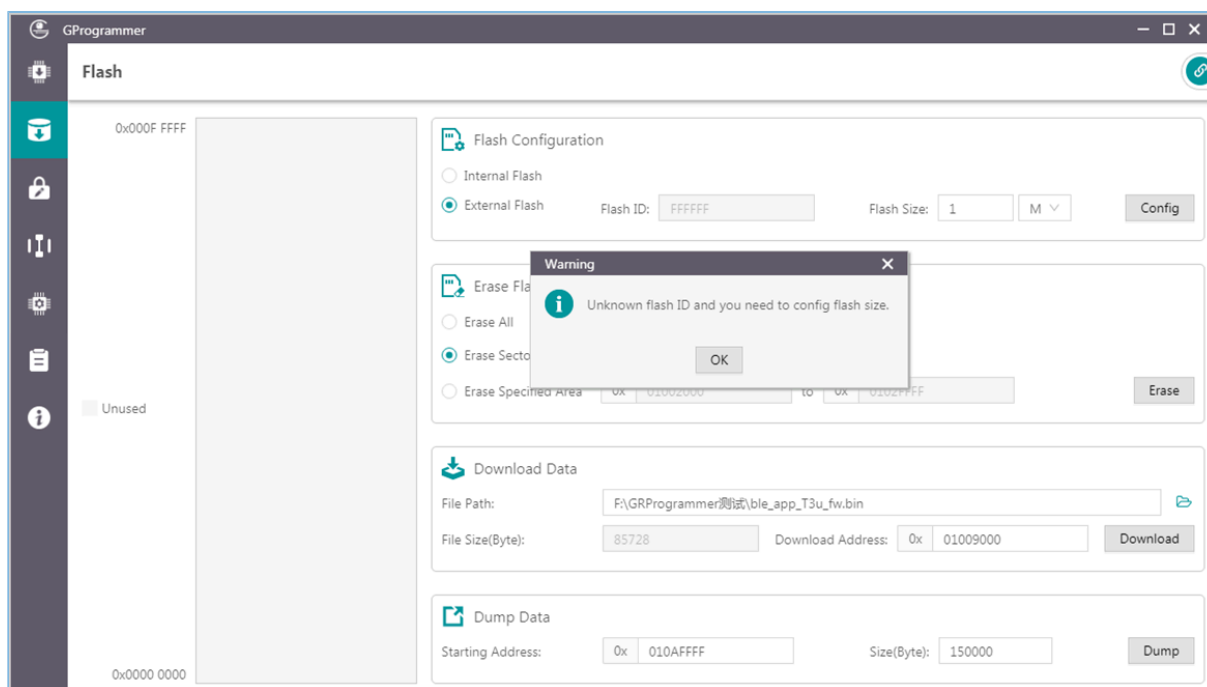


图 3-21 未知Flash

### 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。

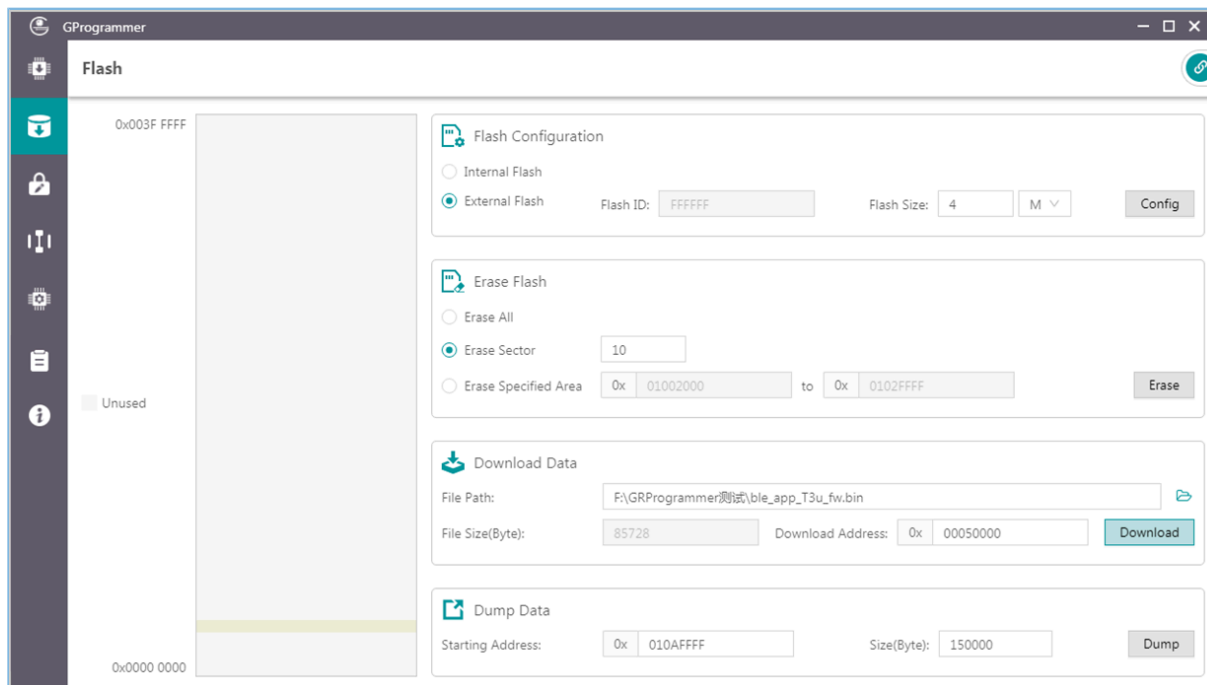


图 3-22 下载数据

#### 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。

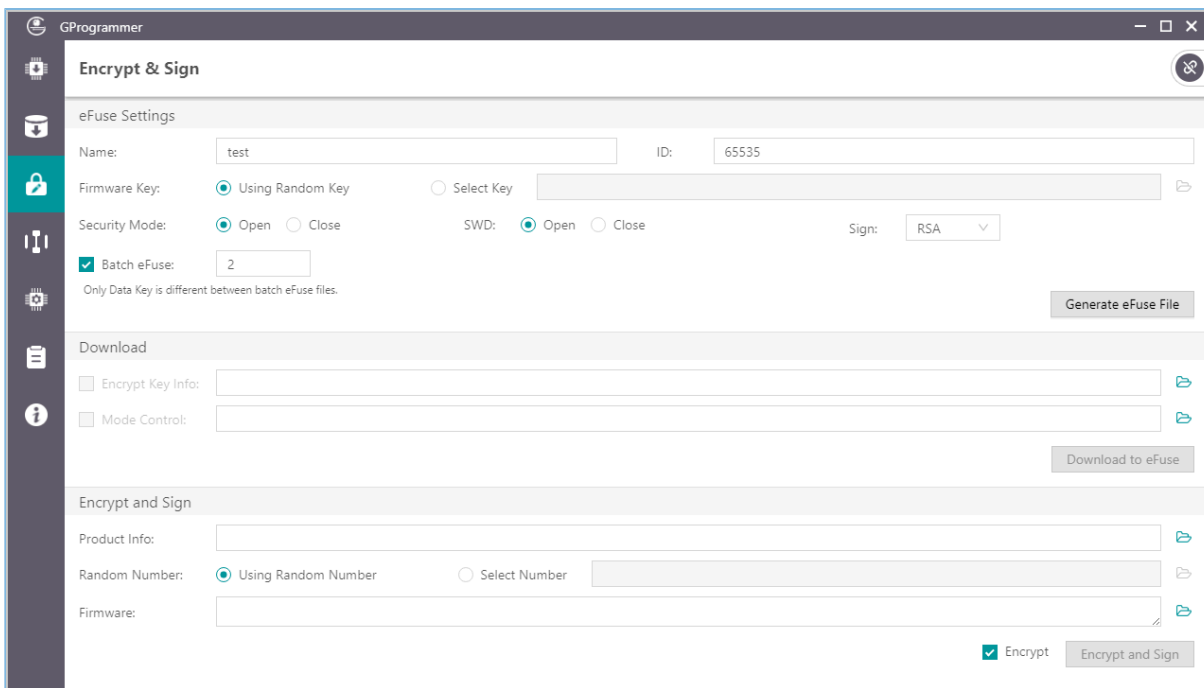


图 3-23 “Encrypt & Sign” 页面

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。

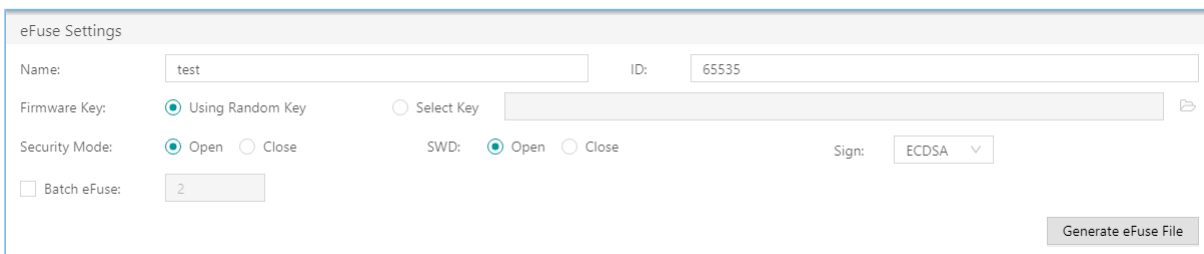


图 3-24 设置参数

 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系列或GR5405，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：

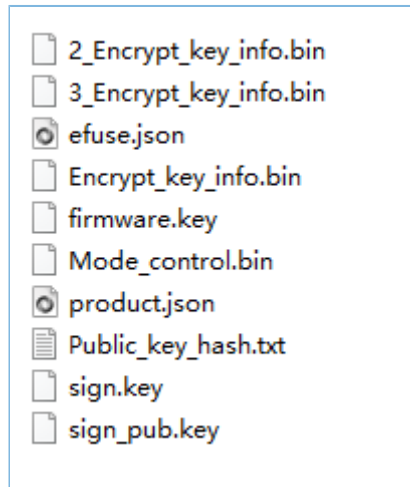


图 3-25 生成的文件

- *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。

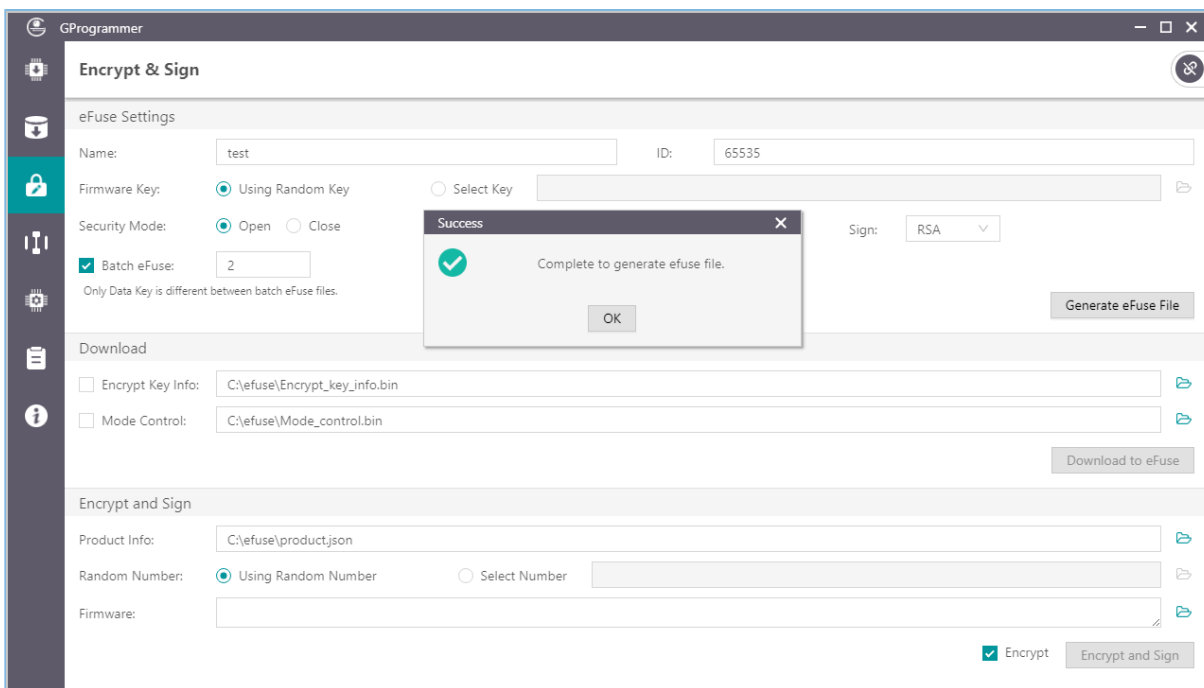


图 3-26 自动加载文件路径

### 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。

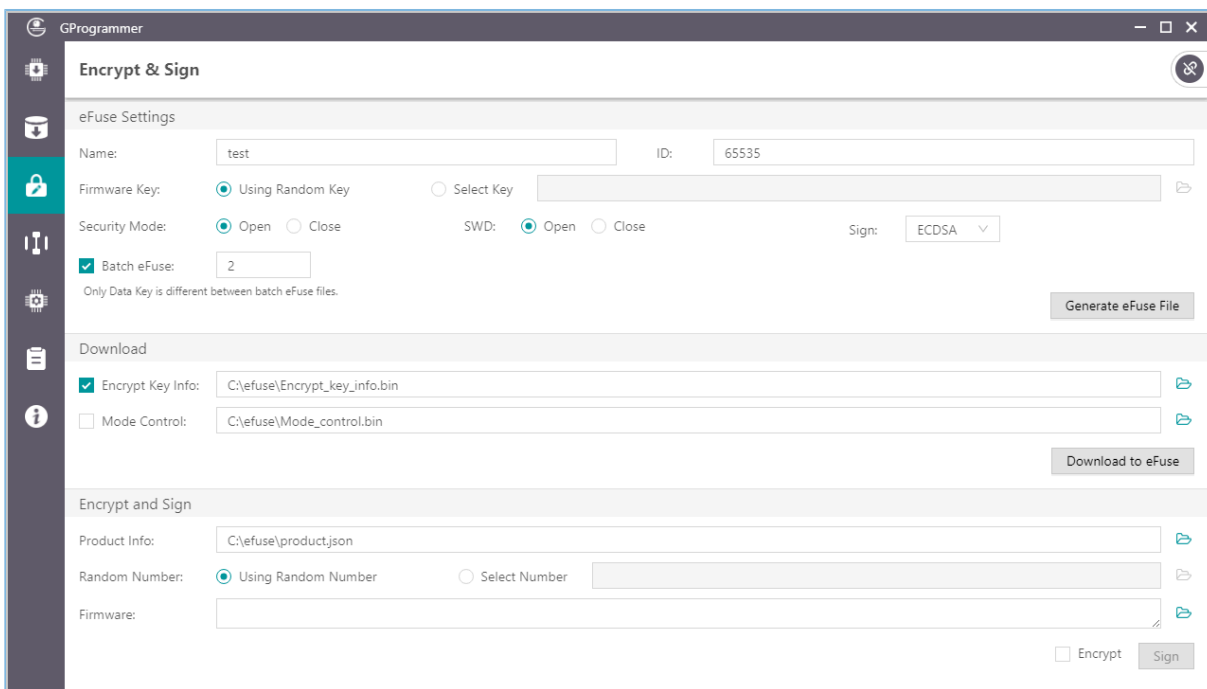


图 3-27 下载文件到eFuse

#### 说明:

- eFuse不能重复下载。
- GR533x系列、GR5405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。

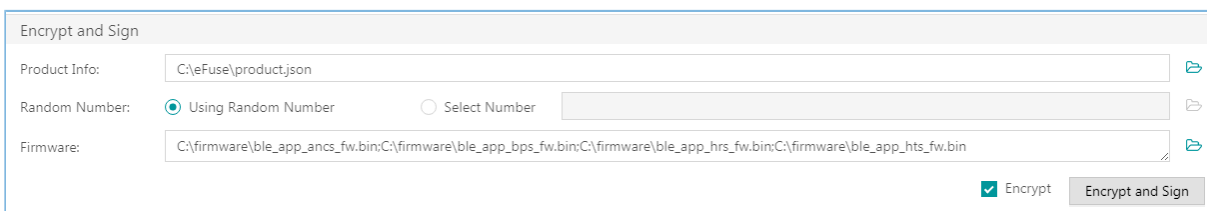


图 3-28 添加多个固件

- **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：

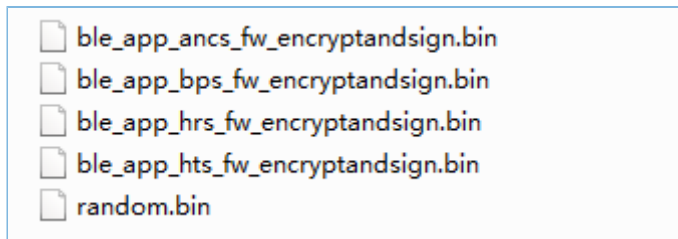


图 3-29 加密加签生成的文件

- 仅加签后生成的文件包括：

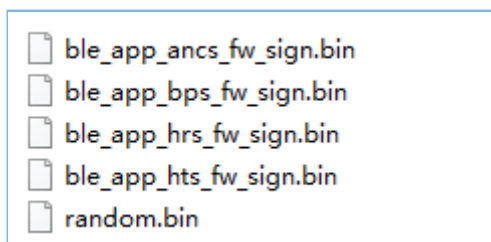


图 3-30 仅加签生成的文件

#### 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。

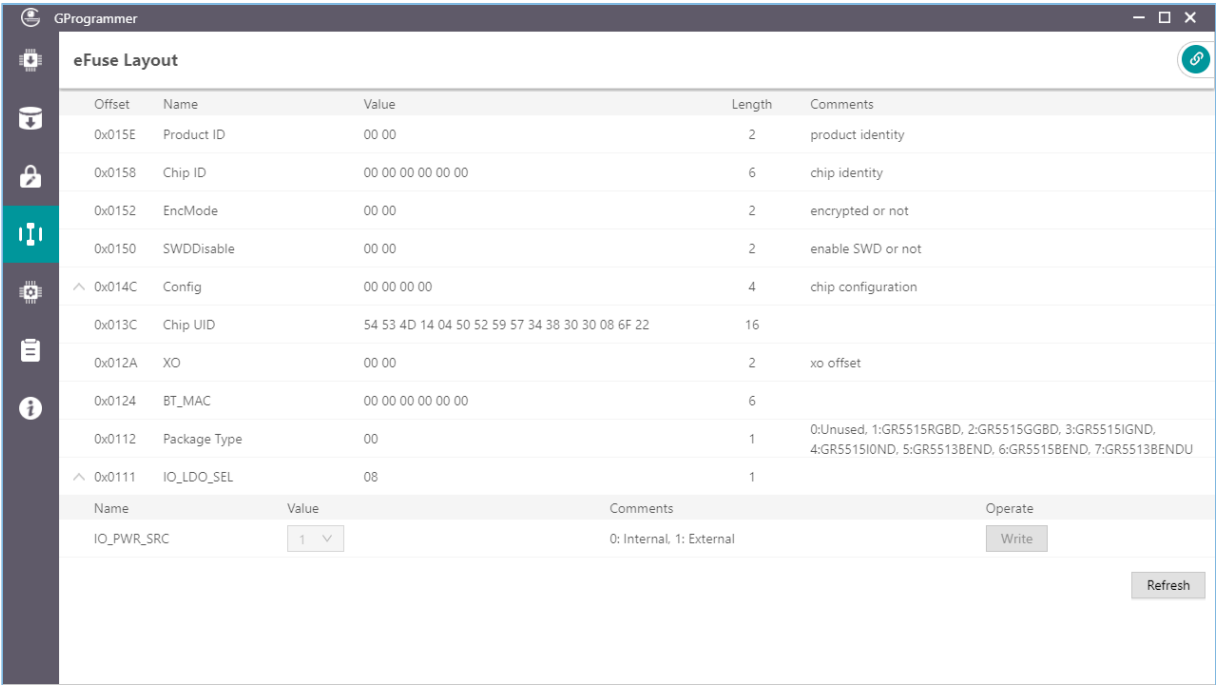


图 3-31 “eFuse Layout” 页面

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。



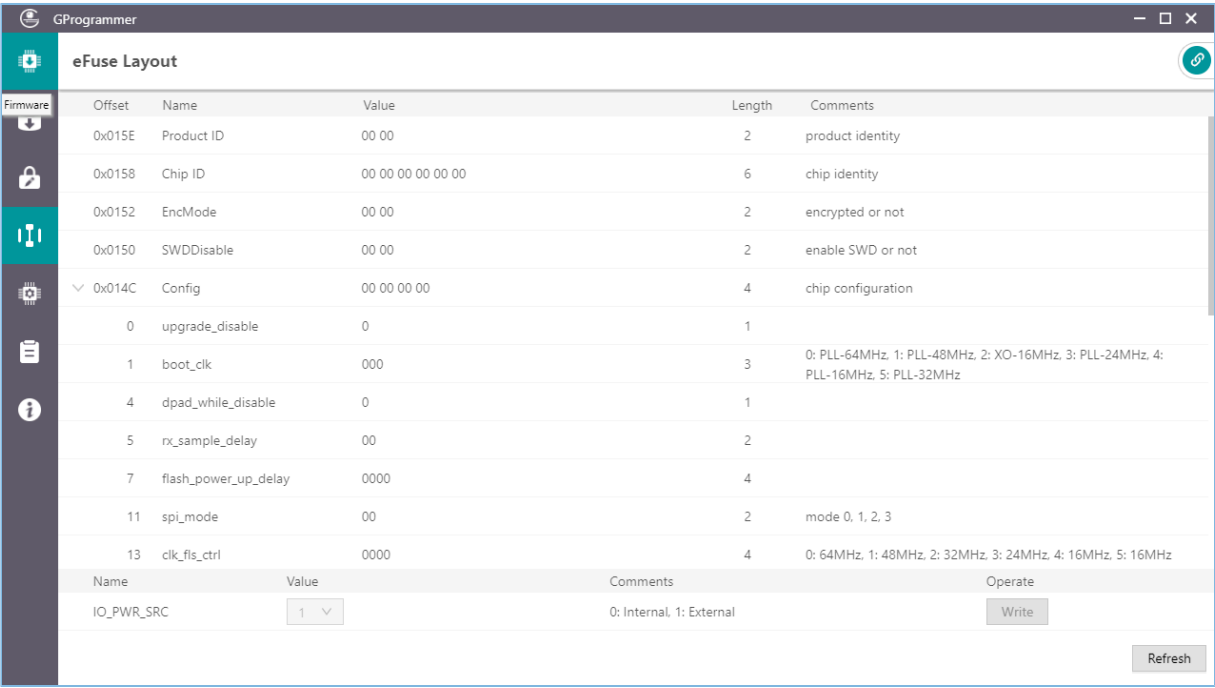


图 3-32 展开位域

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、GR5405芯片。

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

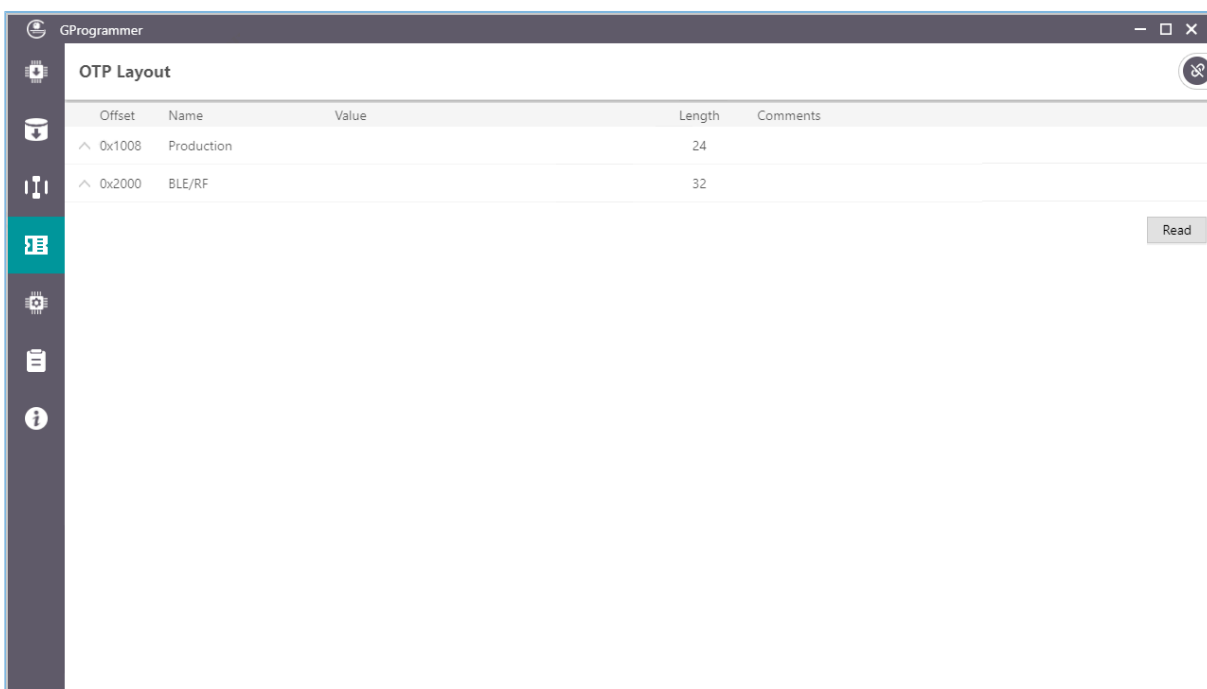


图 3-33 “OTP Layout” 页面

### 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。

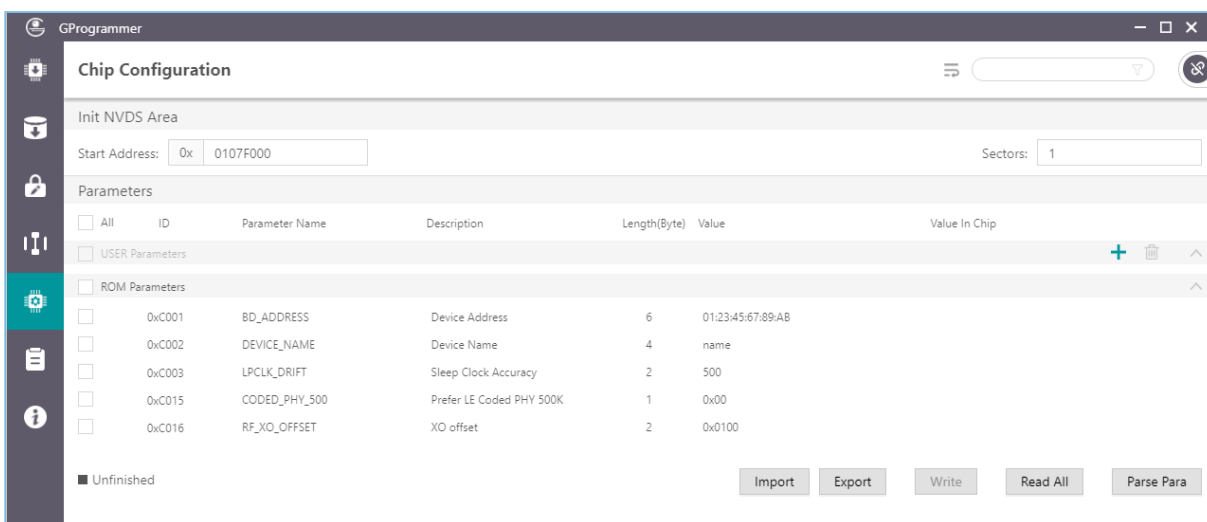


图 3-34 “Chip Configuration” 页面

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。                               |
| 0xC003 | 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。 |
| 0xC015 | CODED_PHY_500 | 设置默认Coded PHY，0为125 kbps，1为500 kbps。  |
| 0xC016 | 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。

Init NVDS Area

Start Address: 0x 010FF000

Sectors: 1

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。

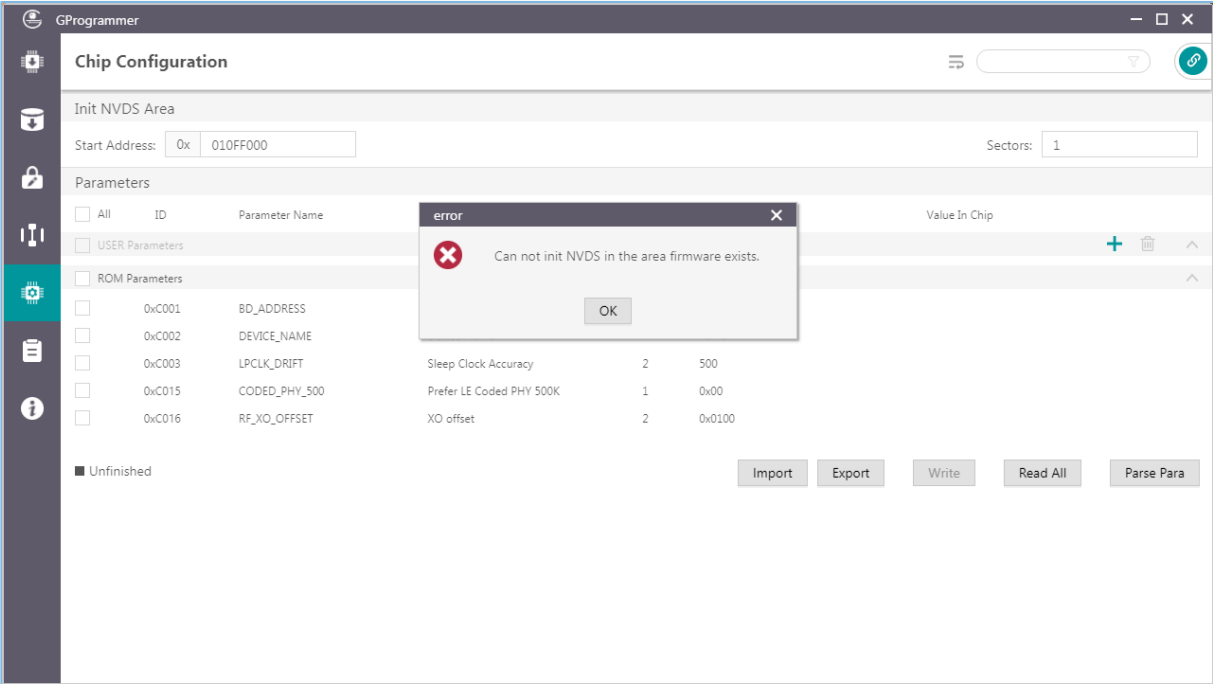


图 3-36 NVDS区域与固件区域重叠

### 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）。

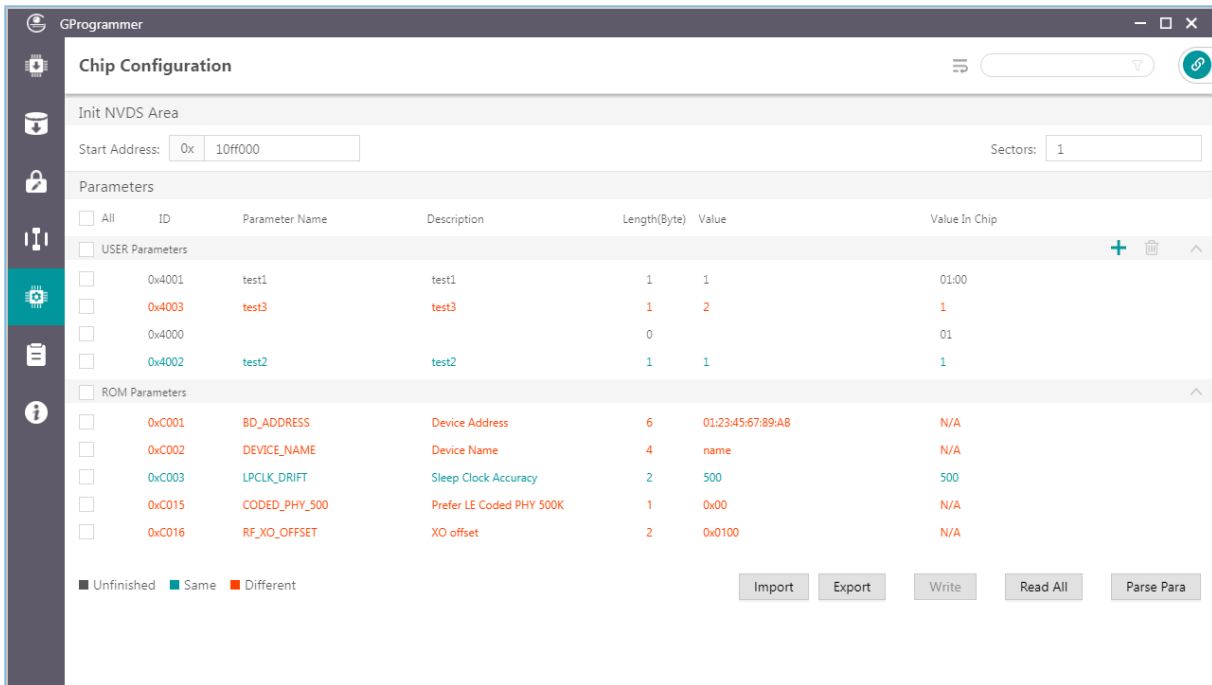


图 3-37 读取参数

### 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。

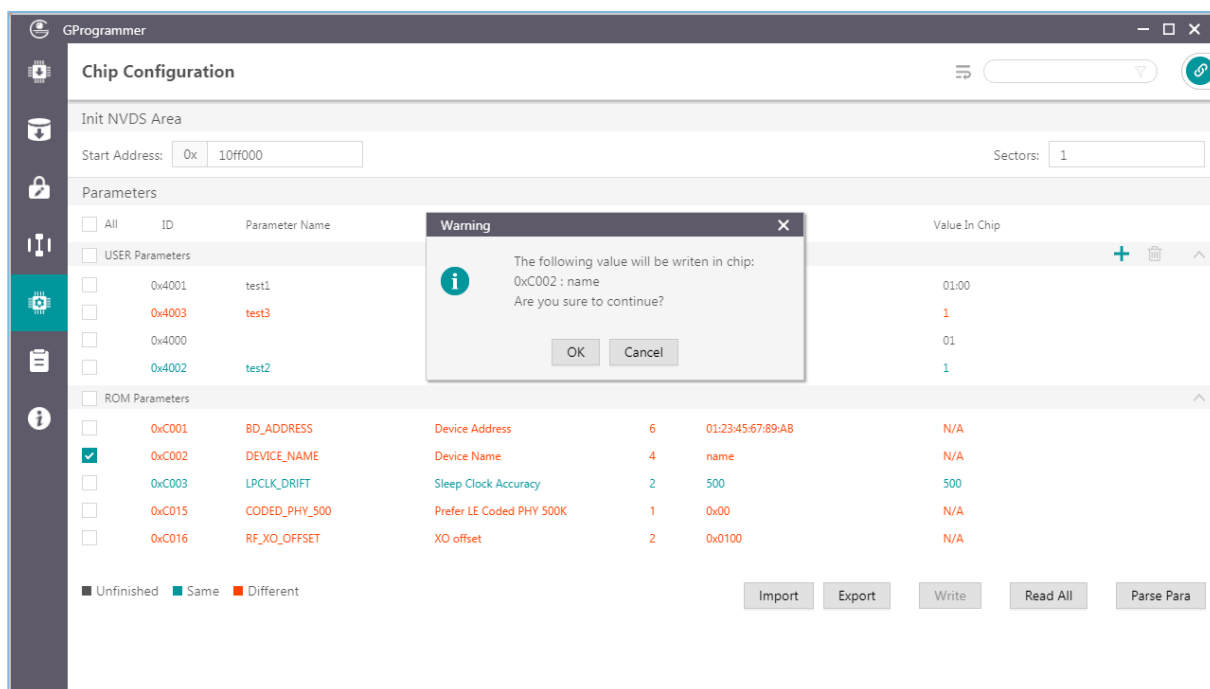


图 3-38 写入参数

#### 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。

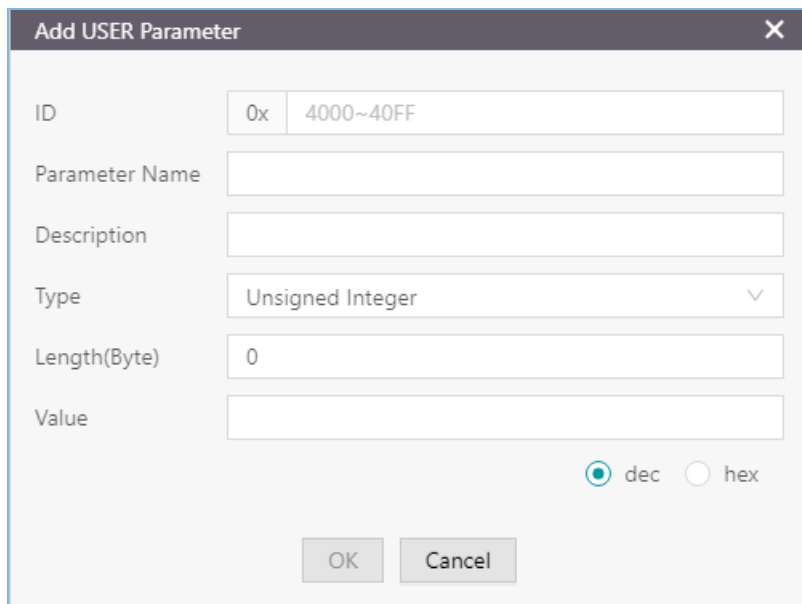
A dialog box titled "Add USER Parameter" with a close button (X) in the top right corner. It contains several input fields: "ID" with a prefix "0x" and a range "4000~40FF"; "Parameter Name"; "Description"; "Type" with a dropdown menu showing "Unsigned Integer"; "Length(Byte)" with the value "0"; and "Value". At the bottom right, there are two radio buttons labeled "dec" (selected) and "hex". At the bottom center, there are "OK" and "Cancel" buttons.

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。

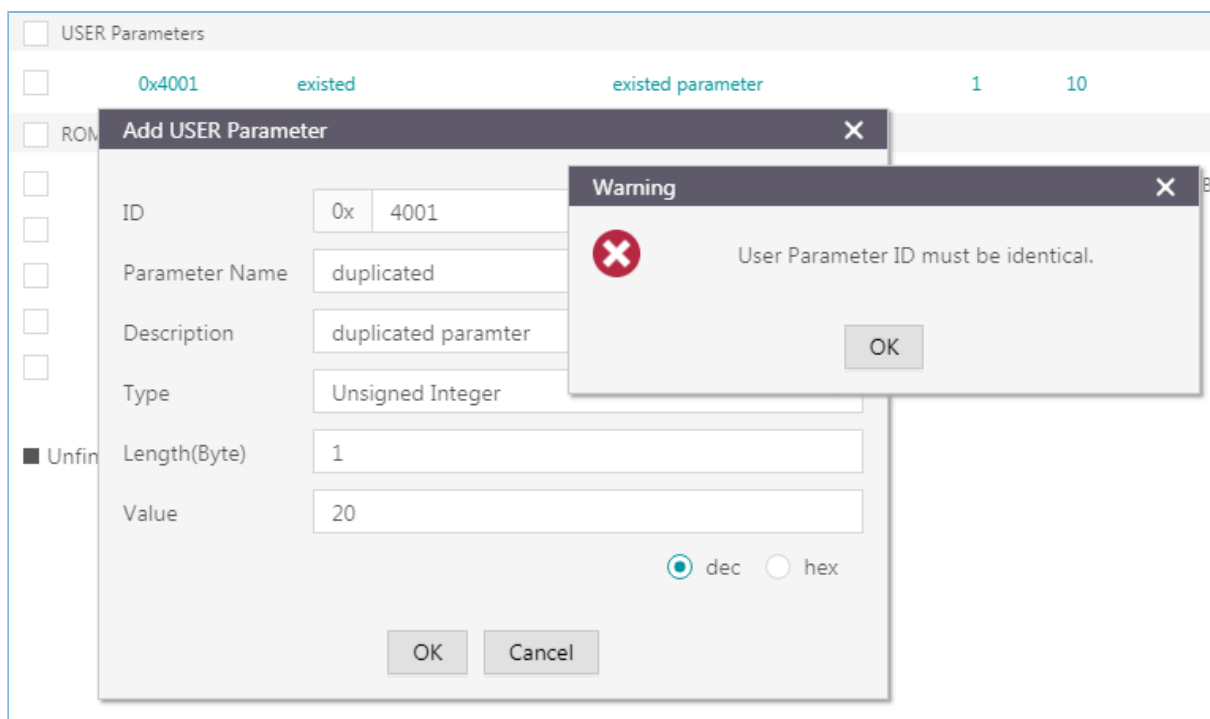


图 3-40 参数ID重复

### 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。

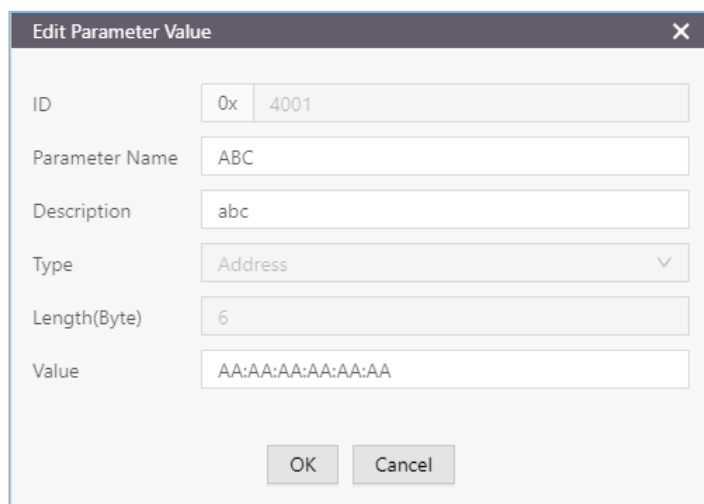


图 3-41 参数编辑窗口

### 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  图标，即可实现从NVDS中删除参数。

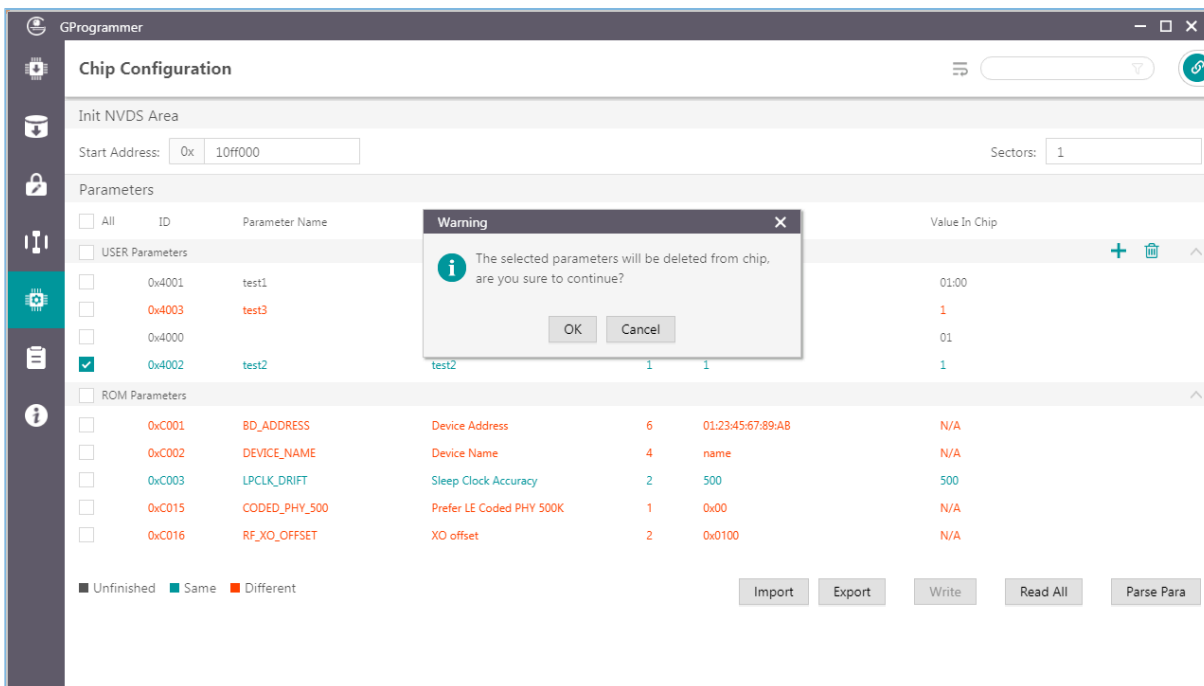


图 3-42 删除User参数

### 说明:

- 勾选多个User参数，然后点击  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， 图标将被置灰，即不允许删除。

## 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。



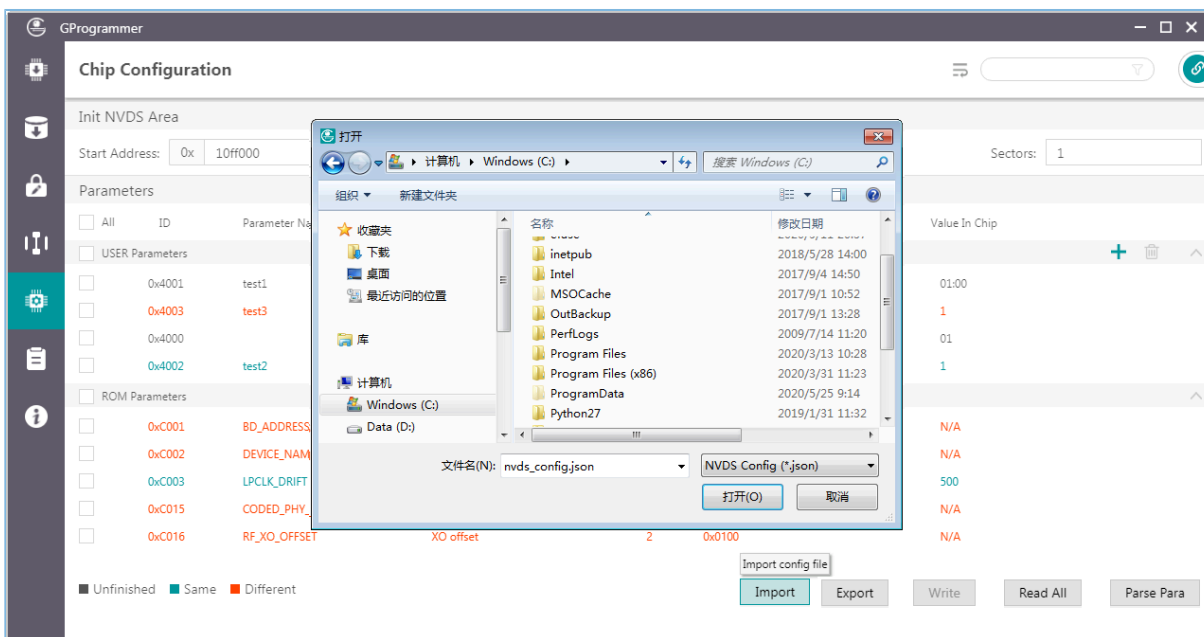


图 3-43 导入文件

#### 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。

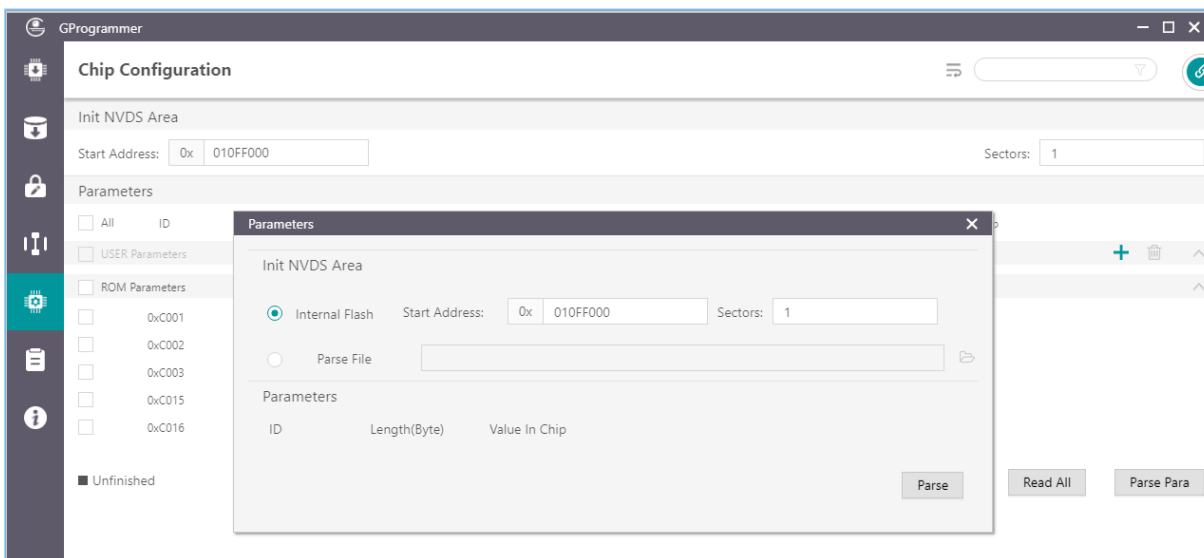


图 3-44 配置解析数据

- 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。

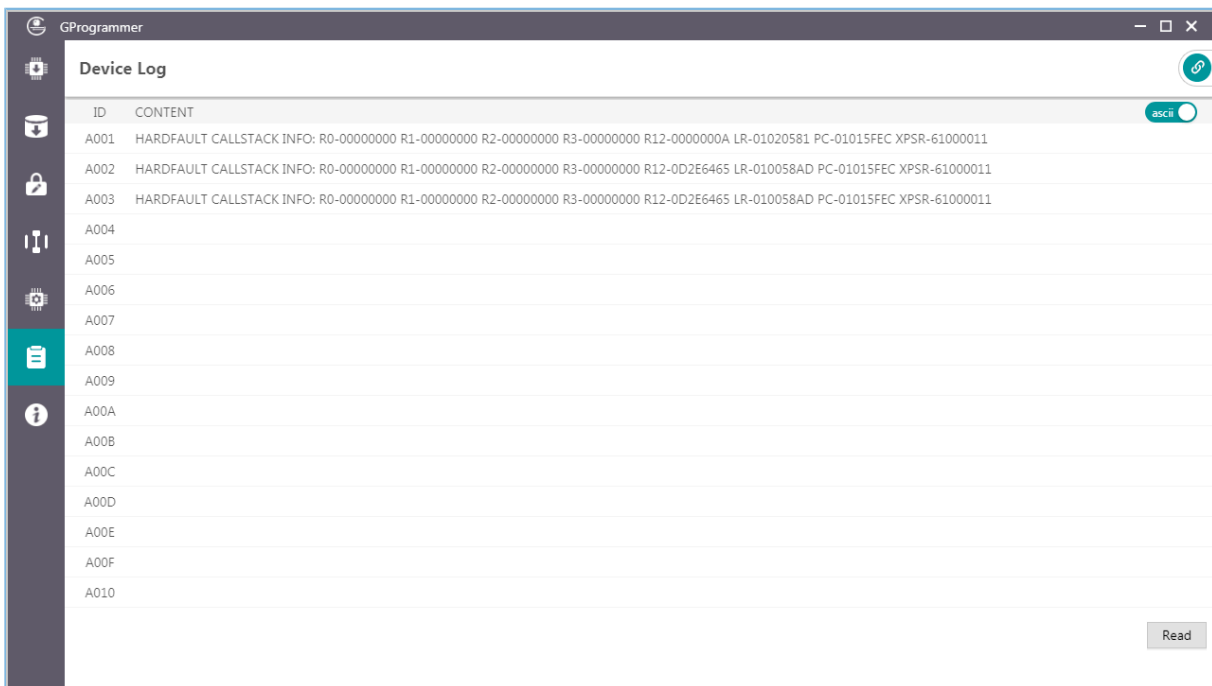


图 3-45 “Device Log” 页面

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。

-  **ascii**：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。
-  **stream**：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。

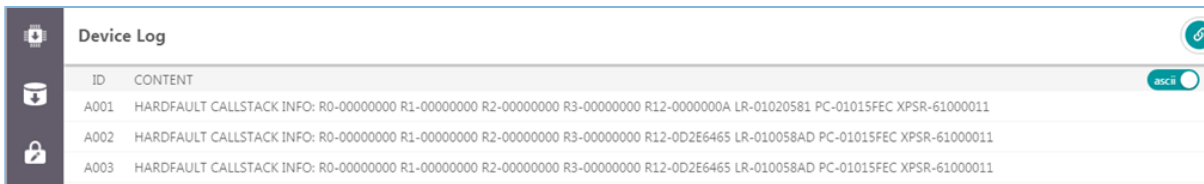


图 3-46 以字符形式显示日志

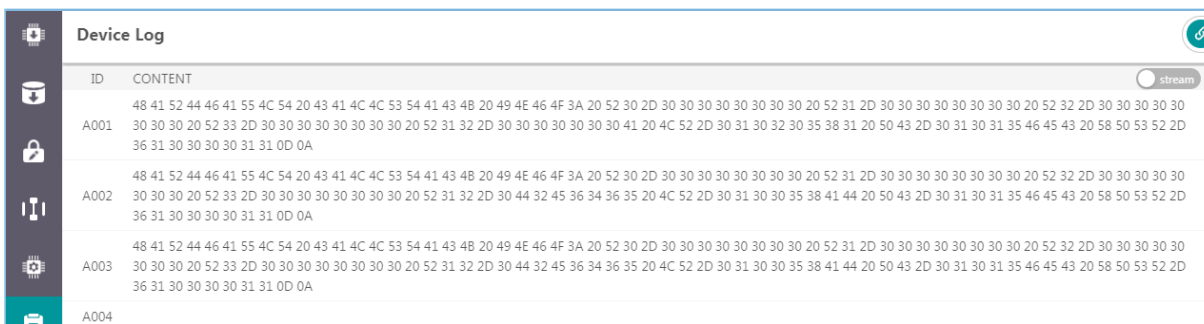


图 3-47 以字节流形式显示日志

### 3.12 命令行

GProgrammer提供2类命令行程序：GR5xxx console和GR5xxx encrypt signature。

 说明:

GR5xxx表示芯片系列名称。

- GR5xxx\_console可方便用户直接使用命令行完成固件烧录、Flash/NVDS/OTP/eFuse操作。
- GR5xxx\_encrypt\_signature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

- Windows平台

在Windows平台上，命令程序GR5xxx console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 help，查看命令列表。
4. 输入GR5xxx\_console.exe help <command>（如GR5xxx\_console.exe help program），查看该命令详细描述和使用方法。

- Linux平台
- 在Linux平台上，命令程序GR5xxx\_console的使用步骤如下：
1. 打开终端。
  2. 使用cd命令进入GProgrammer安装目录。
  3. 输入./GR5xxx\_console help，查看命令列表。
  4. 输入./GR5xxx\_console help <command>（如./GR5xxx\_console help program），查看该命令详细描述和使用方法。

 提示：

如果需要使用旧版本（V2.0.0之前）命令，可输入GR5xxx\_console.exe help jlink（Windows平台）或./GR5xxx\_console help jlink（Linux平台）查看命令列表。建议使用最新版本，以获得更好的使用体验。

3.12.1.1 通用参数

通用参数适用于所有需连接设备的命令，不包括无需与设备通信的本地命令，如help、merge、generate和device等。这些参数主要包括连接参数和芯片类型等。

表 3-7 通用参数描述

| 参数名称           | 参数说明                                            |
|----------------|-------------------------------------------------|
| --uart         | 指定 UART通信方式。参数名后跟UART名称，如：COM32、ttyUSB0。        |
| --baudrate（可选） | 当以UART方式通信时，指定通信波特率，如921600、2000000。            |
| --jlink        | 指定J-Link通信方式。参数名后跟J-Link序列号（SN）。当只有一个设备时，SN可为0。 |
| --chip         | 指定芯片类型，如GR5515、GR5525、GR5332、GR5405等。           |

3.12.1.2 外部Flash配置参数

执行外部Flash操作命令（writeFlash、updateFlash、readFlash、eraseFlash）时，需要额外配置外部Flash参数。

**GPIO\_TYPE 参数说明：**

- GR551x :
  - 00 - Normal
  - 01 - AON
  - 02 - MSIO
- GR5525/GR5526 :
  - 00 - GPIOA（GPIO0 - GPIO15）

- 01 - GPIOB （GPIO16 - GPIO31）
- 02 - GPIOC （GPIO32 - GPIO33）
- 03 - AON （AON\_GPIO\_0 - AON\_GPIO\_7）
- 04 - MSIOA （MSIO\_A0 - MSIO\_A7）

表 3-8 外部Flash配置参数描述

| 参数名称                | 参数说明                                                      |
|---------------------|-----------------------------------------------------------|
| --spi_type          | SPI 类型： QSPI、SPI                                          |
| --gpio_cs_type      | GPIO CS类型， 参见上述 <a href="#">GPIO_TYPE参数说明</a>             |
| --gpio_cs_pin       | GPIO CS PIN                                               |
| --gpio_cs_mux       | GPIO CS MUX                                               |
| --gpio_clk_type     | GPIO Clock类型， 参见上述 <a href="#">GPIO_TYPE参数说明</a>          |
| --gpio_clk_pin      | GPIO Clock PIN                                            |
| --gpio_clk_mux      | GPIO Clock MUX                                            |
| --mosi_io0_type     | GPIO IO0(or MOSI) 类型， 参见上述 <a href="#">GPIO_TYPE参数说明</a>  |
| --mosi_io0_pin      | GPIO IO0(or MOSI) PIN                                     |
| --mosi_io0_mux      | GPIO IO0(or MOSI) MUX                                     |
| --miso_io1_type     | GPIO IO1(or MISO) 类型， 参见上述 <a href="#">GPIO_TYPE参数说明</a>  |
| --miso_io1_pin      | GPIO IO1(or MISO) PIN                                     |
| --miso_io1_mux      | GPIO IO1(or MISO) MUX                                     |
| --gpio_io2_type（可选） | GPIO IO2类型（参见上述 <a href="#">GPIO_TYPE参数说明</a> ）， 用于配置QSPI |
| --gpio_io2_pin（可选）  | GPIO IO2 PIN， 用于配置QSPI                                    |
| --gpio_io2_mux（可选）  | GPIO IO2 MUX， 用于配置QSPI                                    |
| --gpio_io3_type（可选） | GPIO IO3类型（参见上述 <a href="#">GPIO_TYPE参数说明</a> ）， 用于配置QSPI |
| --gpio_io3_pin（可选）  | GPIO IO3 PIN， 用于配置QSPI                                    |
| --gpio_io3_mux（可选）  | GPIO IO3 MUX， 用于配置QSPI                                    |
| --qspi_id（可选）       | QSPI module ID， 用于配置QSPI                                  |

3.12.1.3 帮助命令

- 命令名称： help
- 命令描述： 列出命令，并展示命令的使用方法。
- 参数说明：

表 3-9 help命令参数描述

| 参数名称 | 参数说明                  |
|------|-----------------------|
| 命令名称 | 1. help不跟任何参数，列出所有命令。 |

| 参数名称 | 参数说明                               |
|------|------------------------------------|
|      | 2. help后跟某条命令名称，可查看命令的描述、参数及使用方法等。 |

- 命令示例：
  - 列出所有命令。

```
help
```

- 查看program命令的描述和使用方法。

```
help program
```

3.12.1.4 生成烧录固件

- 命令名称：generate
- 命令描述：将开发编译的Bin/Hex文件生成可烧录的Bin固件。
- 参数说明：

表 3-10 generate命令参数描述

| 参数名称     | 参数说明                                 |
|----------|--------------------------------------|
| --chip   | 指定芯片类型，如GR5515、GR5525、GR5332、GR5405等 |
| --input  | 开发编译的Bin或Hex文件                       |
| --output | 生成的可烧录的二进制固件                         |

- 命令示例：

将开发编译的Bin文件app\_bootloader.bin生成可烧录的固件app\_bootloader\_fw.bin。

```
generate --chip gr5525 --input app_bootloader.bin --output app_bootloader_fw.bin
```

3.12.1.5 合并固件

- 命令名称：merge
- 命令描述：将多个子固件打包合并成一个固件，以便一次性烧录。

说明:

目前，merge命令合并的固件仅支持使用program命令进行烧录，无法通过GProgrammer界面工具烧录。

- 参数说明：

表 3-11 merge命令参数描述

| 参数名称    | 参数说明                                  |
|---------|---------------------------------------|
| --files | 指定待合并的非启动固件，多个文件之间用空格分开。              |
| --boot  | 指定一个“启动”固件。合并固件时，通常需要指定一个固件程序为“启动”固件。 |
| --out   | 合并后的二进制固件的保存路径。                       |

- 命令示例：  
合并1个应用固件和1个启动固件。

```
merge --files ble_app_template_dfu_fw.bin --boot ble_dfu_boot_fw.bin --out out.bin
```

3.12.1.6 解锁/上锁Flash

- 命令名称：lockFlash
- 命令描述：对内部Flash进行解锁和上锁。
- 参数说明：

表 3-12 lockFlash命令参数描述

| 参数名称   | 参数说明                                                                                   |
|--------|----------------------------------------------------------------------------------------|
| 通用参数   | 参考 3.12.1.1 通用参数。<br><br>说明：<br>仅UART通信方式下支持此命令。                                       |
| --mode | <ul style="list-style-type: none"><li>0：解锁</li><li>1：解锁并擦除Flash</li><li>2：上锁</li></ul> |

- 命令示例：  
解锁GR5405 Flash。

```
lockFlash --chip gr5405 --mode 0 --uart COM30
```

3.12.1.7 复位设备

- 命令名称：reset
- 命令描述：复位设备。
- 参数说明：

表 3-13 reset命令参数描述

| 参数名称 | 参数说明              |
|------|-------------------|
| 通用参数 | 参考 3.12.1.1 通用参数。 |

- 命令示例：  
复位设备。

```
reset --chip gr5332 --jlink 0
```

3.12.1.8 烧录固件

- 命令名称：program
- 命令描述：烧录固件至目标设备的内部Flash。

- 参数说明:

表 3-14 program命令参数描述

| 参数名称    | 参数说明                                                                                                                    |
|---------|-------------------------------------------------------------------------------------------------------------------------|
| 通用参数    | 参考 3.12.1.1 通用参数。                                                                                                       |
| --file  | 指定待烧录固件的文件路径。<br>待烧录固件可以是单个固件，也可以是由merge命令生成的合成固件。                                                                      |
| --run   | 固件烧录完成后是否立即运行程序。 <ul style="list-style-type: none"><li>◦ true: 立即执行</li><li>◦ false: 不处理</li></ul>                      |
| --erase | 固件烧录前是否擦除Flash。 <ul style="list-style-type: none"><li>◦ 0: 不擦除</li><li>◦ 1: 擦除整个Flash</li><li>◦ 2: 擦除BootInfo</li></ul> |

- 命令示例:

- 烧录固件app\_bootloader\_fw.bin，且烧录前先擦除整个Flash，烧录完成后立即运行程序。

```
program --jlink 0 --chip GR5405 --file app_bootloader_fw.bin --erase 1 --run 1
```

- 生成可烧录固件并烧录。

```
generate --chip GR5405 --input app_bootloader.bin --output app_bootloader_fw.bin
program --uart COM10 --chip GR5405 --file app_bootloader_fw.bin --erase 1 --run 1
```

- 合并固件并烧录。

```
merge --files ble_app_template_dfu_fw.bin --boot ble_dfu_boot_fw.bin --out out.bin
program --uart COM10 --chip GR5405 --file out.bin --erase 1 --run 1
```

3.12.1.9 写入Flash

- 命令名称: writeFlash
- 命令描述: 写入数据至Flash。
- 参数说明:

表 3-15 writeFlash命令参数描述

| 参数名称            | 参数说明                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------|
| 通用参数            | 参考 3.12.1.1 通用参数。                                                                            |
| 外部Flash配置参数（可选） | 当“flashType”选择外部Flash时，需要输入外部Flash配置参数，具体参考3.12.1.2 外部Flash配置参数章节。                           |
| --flashType（可选） | Flash 类型 <ul style="list-style-type: none"><li>◦ 0：内部Flash（默认）</li><li>◦ 1：外部Flash</li></ul> |



| 参数名称            | 参数说明                                                                                                                             |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------|
| --flashSize（推荐） | Flash容量大小，单位为Kbytes。                                                                                                             |
| --file          | 文件数据，参数名后跟文件路径。                                                                                                                  |
| --hex           | Hex数据，参数名后跟Hex字符串。                                                                                                               |
| --address       | 待写入数据的目标地址（16进制）                                                                                                                 |
| --force         | 如果数据写入地址与当前NVDS、BOOT_INFO 或IMAGE地址冲突，是否强制写入。 <ul style="list-style-type: none"><li>◦ true：强制写入</li><li>◦ false：不执行写入操作</li></ul> |

命令示例：

1. 写入文件数据至内部Flash指定地址。

```
writeFlash --chip GR5525 --file bootloader.bin --address 0x00210000 --force false --uart COM32
```

2. 写入Hex数据至内部Flash指定地址。

```
writeFlash --chip GR5405 --hex 0x11223344 --address 0x00210000 --force true --jlink 0
```

说明：

writeFlash可能会擦除写入目标地址范围外的数据，如需避免此情况，请使用updateFlash命令。

3.12.1.10 更新Flash

- 命令名称：updateFlash
- 命令描述：更新Flash数据。
- 参数说明：

表 3-16 updateFlash命令参数描述

| 参数名称            | 参数说明                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------|
| 通用参数            | 参考 3.12.1.1 通用参数。                                                                            |
| 外部Flash配置参数（可选） | 当“flashType”选择外部Flash时，需要输入外部Flash配置参数，具体参考3.12.1.2 外部Flash配置参数章节。                           |
| --flashType（可选） | Flash 类型 <ul style="list-style-type: none"><li>◦ 0：内部Flash（默认）</li><li>◦ 1：外部Flash</li></ul> |
| --flashSize（推荐） | Flash容量大小，单位为Kbytes。                                                                         |
| --file          | 文件数据，参数名后跟文件路径。                                                                              |
| --hex           | Hex数据，参数名后跟Hex字符串。                                                                           |
| --address       | 待更新数据的目标地址（16进制）                                                                             |

命令示例：

1. 更新文件数据至内部Flash指定地址。

```
updateFlash --chip GR5405 --file save.bin --address 0x00210000 --jlink 0
```

2. 更新Hex数据至内部Flash指定地址。

```
updateFlash --chip GR5405 --hex 0x11223344 --address 0x00210000 --force true --jlink 0
```

### 3.12.1.11 读取Flash

- 命令名称: readFlash
- 命令描述: 从Flash指定地址读取数据。
- 参数说明:

表 3-17 readFlash命令参数描述

| 参数名称            | 参数说明                                                                                     |
|-----------------|------------------------------------------------------------------------------------------|
| 通用参数            | 参考 <a href="#">3.12.1.1 通用参数</a> 。                                                       |
| 外部Flash配置参数（可选） | 当“flashType”选择外部Flash时，需要输入外部Flash配置参数，具体参考 <a href="#">3.12.1.2 外部Flash配置参数</a> 章节。     |
| --flashType（可选） | Flash 类型 <ul style="list-style-type: none"><li>0：内部Flash（默认）</li><li>1：外部Flash</li></ul> |
| --flashSize（推荐） | Flash容量大小，单位为Kbytes。                                                                     |
| --save          | 保存读取的数据至文件。若未输入此参数，则将在终端以Hex格式显示前1K数据。                                                   |
| --address       | 待读取数据的目标地址                                                                               |
| --size          | 读取的数据大小，单位为字节。                                                                           |

- 命令示例:  
从内部Flash指定地址读取数据，并将数据保存到文件。

```
readFlash --chip GR5515 --address 0x01070000 --size 10240 --save save.bin --uart COM32
```

### 3.12.1.12 擦除Flash

- 命令名称: eraseFlash
- 命令描述: 擦除Flash。
- 参数说明:

表 3-18 eraseFlash命令参数描述

| 参数名称 | 参数说明                               |
|------|------------------------------------|
| 通用参数 | 参考 <a href="#">3.12.1.1 通用参数</a> 。 |

| 参数名称            | 参数说明                                                                                                                               |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|
| 外部Flash配置参数（可选） | 当“flashType”选择外部Flash时，需要输入外部Flash配置参数，具体参考 <a href="#">3.12.1.2 外部Flash配置参数</a> 章节。                                               |
| --flashType（可选） | Flash 类型 <ul style="list-style-type: none"><li>0：内部Flash（默认）</li><li>1：外部Flash</li></ul>                                           |
| --flashSize（推荐） | Flash容量大小，单位为Kbytes。                                                                                                               |
| --start         | 待擦除Flash区域的起始地址（16进制）。实际擦除操作将从该地址所在的Sector开始执行。                                                                                    |
| --end           | 待擦除Flash区域的结束地址（16进制）。实际擦除操作将执行到该地址所在的Sector结束。                                                                                    |
| --force         | 进行非全片擦除时，若擦除地址与当前NVDS、BOOT_INFO 或IMAGE地址冲突，是否强制擦除。 <ul style="list-style-type: none"><li>true：强制擦除</li><li>false：不执行擦除操作</li></ul> |

命令示例：

1. 全片擦除内部Flash。

```
eraseFlash --all true --chip GR5332 --uart COM77
```

2. 擦除内部Flash指定区域。

```
eraseFlash --start 0x00210000 --end 0x00212000 --chip GR5405 --jlink 0
```

3.12.1.13 写入NVDS

- 命令名称：writeNvds
- 命令描述：写入数据至NVDS。
- 参数说明：

表 3-19 writeNvds命令参数描述

| 参数名称      | 参数说明                               |
|-----------|------------------------------------|
| 通用参数      | 参考 <a href="#">3.12.1.1 通用参数</a> 。 |
| --address | NVDS起始地址（16进制）                     |
| --sectors | NVDS Sector总数目，等于NVDS容量除以4K。       |
| --tag     | 待写入数据的目标Tag。                       |
| --hex     | 以Hex字符串方式输入待写入的数据。                 |

命令示例：

写入数据至NVDS。

```
writeNvds --address 0x00270000 --sectors 1 --tag C001 --hex 0123456789BB --jlink 0 --chip GR5332
```

3.12.1.14 读取NVDS

- 命令名称: readNvds
- 命令描述: 从NVDS 指定Tag读取数据。
- 参数说明:

表 3-20 readNvds命令参数描述

| 参数名称       | 参数说明                                  |
|------------|---------------------------------------|
| 通用参数       | 参考 3.12.1.1 通用参数。                     |
| --address  | NVDS起始地址（16进制）                        |
| --sectors  | NVDS Sector总数目，等于NVDS容量除以4K。          |
| --tag      | 待读取数据的目标Tag。                          |
| --save（可选） | 保存读取的数据至文件。若未输入此参数，则以Hex格式显示前1K数据至终端。 |

- 命令示例:  
从NVDS指定Tag读取数据。

```
readNvds --address 0x00270000 --sectors 1 --tag C001 --chip GR5332 --jlink 0
```

 说明:

在读取NVDS 之前，请确保NVDS已写入数据，否则将读取失败。

3.12.1.15 擦除NVDS

- 命令名称: eraseNvds
- 命令描述: 删除NVDS 指定TAG数据。
- 参数说明:

表 3-21 eraseNvds命令参数描述

| 参数名称      | 参数说明                         |
|-----------|------------------------------|
| 通用参数      | 参考 3.12.1.1 通用参数。            |
| --address | NVDS起始地址（16进制）               |
| --sectors | NVDS Sector总数目，等于NVDS容量除以4K。 |
| --tag     | 待删除数据的目标Tag。                 |

- 命令示例:  
删除NVDS 指定TAG数据。

```
eraseNvds --address 0x00270000 --sectors 1 --tag C001 --chip GR5332 --jlink 0
```

3.12.1.16 写入eFuse

- 命令名称: writeEfuse
- 命令描述: 写入数据至eFuse的User Region区域（地址范围为0x00 ~ 0x1F）。

说明:

该命令仅适用于GR551x、GR5525、GR5526系列芯片。

- 参数说明:

表 3-22 writeEfuse命令参数描述

| 参数名称     | 参数说明                               |
|----------|------------------------------------|
| 通用参数     | 参考 3.12.1.1 通用参数。                  |
| --offset | 待写入数据的目标偏移地址（0x00 ~ 0x1C），需以4字节对齐。 |
| --file   | 文件数据，参数名后跟文件路径，且文件大小需以4字节对齐。       |
| --hex    | Hex数据，参数名后跟十六进制字符串，需以8个字符对齐。       |

- 命令示例:  
写入Hex数据至eFuse。

```
writeEfuse --offset 0 --hex 12345678 --chip GR5525 --uart COM77
```

3.12.1.17 读取eFuse

- 命令名称: readEfuse
- 命令描述: 从eFuse读取数据。
- 参数说明:

表 3-23 readEfuse命令参数描述

| 参数名称       | 参数说明                                |
|------------|-------------------------------------|
| 通用参数       | 参考 3.12.1.1 通用参数。                   |
| --offset   | 待读取数据的目标偏移地址，需以4字节对齐。               |
| --size     | 读取的数据长度（单位为字节），需以4字节对齐。             |
| --save（可选） | 保存读取的数据至文件。若未输入此参数，则将数据以Hex格式输出到终端。 |

- 命令示例:  
从eFuse读取数据。

```
readEfuse --offset 0 --size 32 --chip GR5332 --uart COM77
```

3.12.1.18 写入OTP

- 命令名称: writeOtp
- 命令描述: 写入数据至OTP指定地址。

说明:

该命令仅适用于GR533x系列、GR5405芯片。

- 参数说明:

表 3-24 writeOtp命令参数描述

| 参数名称      | 参数说明               |
|-----------|--------------------|
| 通用参数      | 参考 3.12.1.1 通用参数。  |
| --address | 待写入数据的目标地址（16进制）   |
| --file    | 文件数据，参数名后跟文件路径。    |
| --hex     | Hex数据，参数名后跟Hex字符串。 |

- 命令示例:  
写入数据至OTP。

```
writeOtp --address 0x2008 --hex 12345678 --uart COM77 --chip GR5332
```

3.12.1.19 读取OTP

- 命令名称: readOtp
- 命令描述: 从OTP指定地址读取数据。

说明:

该命令仅适用于GR533x系列、GR5405芯片。

- 参数说明:

表 3-25 readOtp命令参数描述

| 参数名称       | 参数说明                                |
|------------|-------------------------------------|
| 通用参数       | 参考 3.12.1.1 通用参数。                   |
| --address  | 待读取数据的目标地址（16进制）                    |
| --size     | 读取的数据长度，单位为字节。                      |
| --save（可选） | 保存读取的数据至文件。若未输入此参数，则将数据以Hex格式输出到终端。 |

- 命令示例:  
从OTP读取数据。

```
readOtp --address 0x2008 --size 0x10 --uart COM77 --chip GR5332
```

3.12.1.20 擦除OTP

- 命令名称: eraseOtp
- 命令描述: 删除OTP指定地址数据。

说明:

该命令仅适用于GR533x系列、GR5405芯片。

- 参数说明:

表 3-26 eraseOtp命令参数描述

| 参数名称      | 参数说明              |
|-----------|-------------------|
| 通用参数      | 参考 3.12.1.1 通用参数。 |
| --address | 待删除数据的目标地址（16进制）  |

- 命令示例:  
删除OTP数据。

```
eraseOtp --address 0x2008 --uart COM77 --chip GR5332
```

3.12.1.21 展示设备列表

- 命令名称: device
- 命令描述: 展示设备列表。
- 参数说明:

表 3-27 device命令参数描述

| 参数名称        | 参数说明                                                  |
|-------------|-------------------------------------------------------|
| --type      | 展示的设备类型，包括：uart、jlink或all。其中，all表示同时展示uart和jlink设备列表。 |
| --count（可选） | 展示的设备数目（不超过50），默认值为50                                 |

- 命令示例:  
1. 展示所有uart设备。

```
device --type uart
```

- 2. 展示所有jlink设备。

```
device --type jlink
```

- 3. 展示所有uart和jlink设备。

```
device --type all
```

### 3.12.2 GR5xxx\_encrypt\_signature

- Windows平台

在Windows平台上，命令程序GR5xxx\_encrypt\_signature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 3-28，全部参数可使用GR5xxx\_encrypt\_signature.exe -h命令进行查看。

- Linux平台

在Linux平台上，命令程序GR5xxx\_encrypt\_signature的使用步骤如下：

1. 打开终端。
2. 使用cd命令进入GProgrammer安装目录。
3. 输入./GR5xxx\_encrypt\_signature --parameter完成操作。

常用参数及对应描述，详见表 3-28，全部参数可使用./GR5xxx\_encrypt\_signature --help命令进行查看。

表 3-28 GR5xxx\_encrypt\_signature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"><li>• encryptandsign: 加密加签</li><li>• sign: 仅加签</li></ul>                                                     |                                                               |
| firmware_key      | 用于加密加签/仅加签的firmware.key文件所在路径                                                                                                                                  |                                                               |
| signature_key     | 用于加密加签/仅加签的sign.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_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： <ul style="list-style-type: none"><li>• 0x01000000: GR551x Flash起始地址</li><li>• 0x00200000: GR5526/GR533/GR5405/GR5525 Flash起始地址</li></ul> |                                                               |



| 参数           | 描述                                                                                                                                                        | 备注 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 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。<br><br>说明<br>若使用的芯片Flash为0 KB，则写入使用的外挂Flash大小。                                                          |    |
| product_type | 芯片类型，可输入值及含义如下：<br><ul style="list-style-type: none"><li>0: GR551x系列芯片</li><li>1: GR5526系列芯片</li><li>2: GR533x系列、GR5405芯片</li><li>4: GR5525系列芯片</li></ul> |    |
| 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 3-28。

- --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，来调用命令行程序。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 Tool  
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](http://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。
- 原因分析：  
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。
- 解决方法：  
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。

### 3.14.2 GProgrammer连接失败

- 问题描述:

GProgrammer与设备连接失败。

- 原因分析和解决方法:

请按照以下步骤进行问题排查和解决:

1. 确认GProgrammer连接之前是否选择了正确的芯片型号。
2. 检查硬件接线是否正确。若芯片正在运行已使能睡眠功能的固件程序，则需确保J-Link仿真器的Reset管脚或UART转接板的RTS管脚连接至芯片的Chip\_En管脚。若不方便连接两管脚，可试图手动复位：在GProgrammer点击“Connect”按钮后，立即按下开发板上的“Reset”键，但手动方式成功率会下降。
3. 如果插入了多个J-Link设备导致频繁弹出窗口而无法连接成功，请先拔掉暂时不使用的设备，然后再尝试重新连接。
4. 若J-Link连接时始终显示等待框或命令行执行无响应，可能是程序卡死在J-Link驱动程序中了。请尝试重新插入设备，然后重启GProgrammer，再次连接。

### 3.14.3 Linux环境下启动GProgrammer时，加载模块失败

- 问题描述:

在Linux环境下启动GProgrammer时，报错：Failed to load module “canberra-gtk-module”。

- 原因分析:

导致该问题的原因是系统缺少canberra-gtk-module。

- 解决方法:

安装canberra-gtk-module即可。例如，ubuntu下可通过以下命令安装：

```
sudo apt-get install libcanberra-gtk-module
```

### 3.14.4 Linux环境下GProgrammer软件界面显示为空白或连接失败

- 问题描述:

在Linux环境下，启动GProgrammer软件，软件界面无任何内容显示为空白，或连接设备失败。

- 原因分析:

可能是由于部分文件无访问权限，导致程序无法正确加载。

- 解决方法:

请确保GProgrammer目录下的所有文件对于当前用户均为可访问的，或以root权限启动GProgrammer。例如，在ubuntu下执行以下命令启动GProgrammer：

```
sudo ./GProgrammer
```