



Gowin Programmer

用户指南

SUG502-2.0,2025-02-28

版权所有 © 2025 广东高云半导体科技股份有限公司

GOWIN高云、、Gowin、高云、云源、小蜜蜂以及晨熙均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本文档内容的部分或全部，并不得以任何形式传播。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止反言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改文档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些文档进行适时的更新。

版本信息

日期	版本	说明
2017/04/06	1.0	初始版本。
2017/08/06	1.1	修改设备编程操作相关内容。
2019/10/28	1.2	● 增加配置Slave SPI 模式； ● 增加创建SVF文件； ● 增加User Flash初始化。
2020/02/17	1.3	增加Programmer安装与启动。
2022/05/30	1.4	● 新增第2.1节Programmer工具链简介； ● 新增第四章 Programmer_cli编程下载流程。
2023/06/08	1.5	● 新增Linux系统下cable权限配置描述； ● 新增Gowin USB Cable（GWU2X）下载线配置说明； ● 删除表3-1 Device操作描述中的SRAM Program JTAG 1149； ● 更新软件截图。
2024/05/09	1.6	● 更新3.6设备安全描述； ● 完善配置SRAM或Flash命令格式描述； ● 新增 I2C Slave 地址操作描述。
2024/06/28	1.7	新增状态码分析描述。
2024/10/25	1.8	● 新增MSPI 2nd Boot地址操作描述； ● 更新部分界面截图。
2024/12/31	1.9	新增章节3.13 JTAG状态机shift-ir宽度调整和3.14远程配置和烧录功能描述。
2025/02/28	2.0	● 3.1设置下载线章节新增管脚连接的注释。 ● 新增GAO-Bridge注释。

目录

目录	i
图目录.....	iii
表目录.....	v
1 关于本手册	1
1.1 手册内容.....	1
1.2 相关文档.....	1
1.3 术语、缩略语	1
1.4 技术支持与反馈.....	2
2 简介.....	3
2.1 Programmer 工具链简介	3
2.1.1 Programmer.exe	3
2.1.2 Programmer_cli.exe.....	3
2.1.3 JTAGLoading.exe	3
2.1.4 jtagserver.exe.....	3
2.1.5 Cable5.uid.up.exe	3
2.1.6 Gowin_USB_Cable_Installer.sh、Makefile.....	3
2.2 Programmer 工具链安装与启动	3
2.2.1 安装方式一	3
2.2.2 安装方式二	4
2.2.3 启动 Programmer 工具链	5
2.3 软件界面	7
2.4 查看软件版本	8
2.5 Linux 系统下 cable 的权限配置	8
2.5.1 Makefile.....	8
2.5.2 Gowin_USB_Cable_Installer.sh.....	9
3 图形界面 Programmer 编程下载流程.....	10
3.1 设置下载线	10
3.2 扫描设备菊花链.....	12
3.3 配置设备菊花链.....	12

3.3.1 添加设备	12
3.3.2 移除设备	13
3.3.3 修改设备在链中位置	13
3.4 配置设备编程操作	13
3.4.1 配置 SRAM 模式	15
3.4.2 小蜜蜂系列 FPGA 配置嵌入式 Flash 模式	15
3.4.3 配置外部 Flash 模式	15
3.4.4 Slave SPI 模式	16
3.5 编辑引脚状态	16
3.6 设备安全	17
3.7 编程下载	18
3.8 创建 SVF 文件	18
3.9 I2C Slave 地址操作	19
3.10 MSPI 2nd Boot 地址操作	19
3.11 状态码分析	20
3.12 User Flash 初始化	21
3.13 JTAG 状态机 shift-ir 宽度调整	22
3.14 远程配置和烧录功能	22
4 Programmer_cli 编程下载流程	27
4.1 预览和帮助	27
4.2 扫描 USB Cable 设备	30
4.3 指定 USB Cable 类型及端口	31
4.4 指定 USB Cable 位置或 UID	31
4.5 扫描 FPGA 设备	31
4.6 指定 Programmer 执行模式	32
4.7 配置 SRAM	34
4.8 配置小蜜蜂家族嵌入式 Flash	34
4.8.1 仅配置 Flash	34
4.8.2 配置 Flash 和 UserFlash 初始化文件	35
4.9 配置外部 SPI Flash	35

图目录

图 2-1 安装云源软件时默认安装 Programmer.....	4
图 2-2 Programmer 驱动安装	4
图 2-3 programmer.exe 启动	5
图 2-4 programmer_cli.exe 启动	6
图 2-5 云源软件中启动 Programmer 快捷键	6
图 2-6 Programmer Main Window.....	7
图 2-7 查看版本号	8
图 2-8 文件列表	8
图 2-9 Makefile 普通用户安装示例	9
图 2-10 Makefile root 用户安装示例	9
图 2-11 脚本安装示例	9
图 3-1 Gowin USB Cable (FT2CH)	11
图 3-2 LPT	11
图 3-3 Gowin USB Cable (GWU2X).....	12
图 3-4 Device Table	12
图 3-5 Device Configuration Interface	13
图 3-6 I/O State Editor	17
图 3-7 Security Configuration	17
图 3-8 Create SVF File.....	19
图 3-9 I2C 的 slave 地址操作.....	19
图 3-10 MSPI 2nd Boot 地址操作	19
图 3-11 状态码分析界面.....	20
图 3-12 User Flash Initialization	21
图 3-13 jtag 状态机的 shift-ir 的宽度调整	22
图 3-14 jtagserver 位置	22
图 3-15 jtagserver 启动状态	22
图 3-16 jtagserver 启动状态	23
图 3-17 Add GVC Server 选项.....	23
图 3-18 GVC 连接界面	23

图 3-19 连接成功界面	23
图 3-20 连接失败界面	24
图 3-21 远程操作输出	24
图 3-22 退出远程配置操作	24
图 4-1 CMD 中打开 programmer_cli	27
图 4-2 帮助信息	28
图 4-3 扫描 USB Cable 设备	30
图 4-4 帮助信息	31
图 4-5 帮助信息	31
图 4-6 扫描 FPGA 设备	31
图 4-7 帮助信息	32
图 4-8 帮助信息	33
图 4-9 示例图	34
图 4-10 示例图	35

表目录

表 1-1 术语、缩略语	1
表 3-1 Device 操作描述	14
表 3-2 User Flash 参考表	21
表 3-3 远程烧录配置功能与设备参考表	24

1 关于本手册

1.1 手册内容

本手册描述高云半导体编程工具 **Gowin Programmer** 的使用方法。有关本手册中的软件界面截图和支持的产品列表等信息，参见 1.9.11 版本。因软件版本更新，部分信息可能会略有差异，具体以用户软件版本信息为准。

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com.cn 可下载、查看以下相关文档：

- [SUG100, Gowin 云源软件用户指南](#)
- [TN653, Gowin FPGA 产品 JTAG 配置手册](#)
- [UG290, Gowin FPGA 产品编程配置手册](#)

1.3 术语、缩略语

本手册中的相关术语、缩略语及相关释义如表 1-1 所示。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
BSDL	Boundary Scan Description Language	边界扫描描述语言
FPGA	Field Programmable Gate Array	现场可编程门阵列
GAO	Gowin Analyzer Oscilloscope	在线逻辑分析仪
GAO-Bridge	The Bridge for JTAG to SPI	JTAG协议转为SPI协议
I/O	Input/Output	输入/输出
SRAM	Static Random Access Memory	静态随机存储器

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址：www.gowinsemi.com.cn

E-mail: support@gowinsemi.com

Tel: +86 755 8262 0391

2 简介

2.1 Programmer 工具链简介

2.1.1 Programmer.exe

图形化工具 Programmer.exe 是 Gowin FPGA 下载器软件，该软件提供图形操作界面，方便且直观的提供数据流配置或下载功能。

2.1.2 Programmer_cli.exe

Programmer_cli 是 Programmer 的命令行版本。

2.1.3 JTAGLoading.exe

Gowin SVF 命令行软件工具，目前仅有 windows 版，当前版本只支持高云 USB Cable Version 3.0 和 4.0。

2.1.4 jtagserver.exe

jtagserver.exe 及 jtagserver_lpt.exe\jtagserver_u2x.exe,属 GAO 工具链。

2.1.5 Cable5.uid.up.exe

Gowin USB Cable Version 5.0 UID 配置工具。

2.1.6 Gowin_USB_Cable_Installer.sh、Makefile

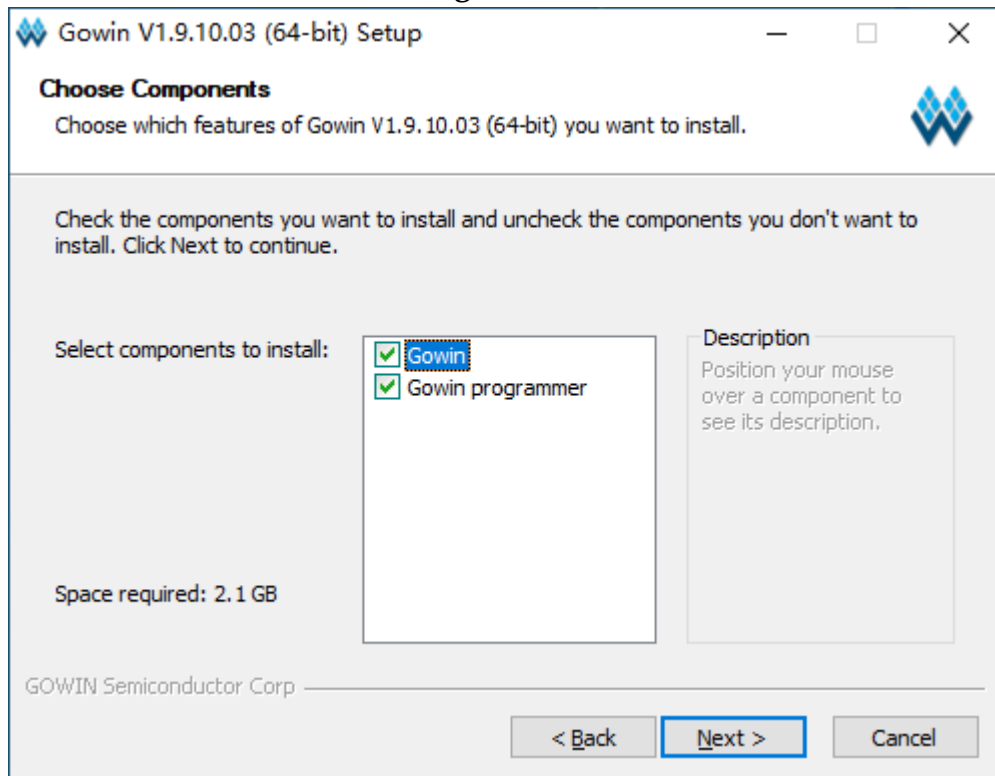
用于更改 linux 系统下 cable 的权限。

2.2 Programmer 工具链安装与启动

2.2.1 安装方式一

安装云源软件时按照默认安装方式，会安装组件 Gowin Programmer，如图 2-1 所示。云源软件的安装方法，请参考文档 [SUG100, Gowin 云源软件用户指南](#)。

图 2-1 安装云源软件时默认安装 Programmer



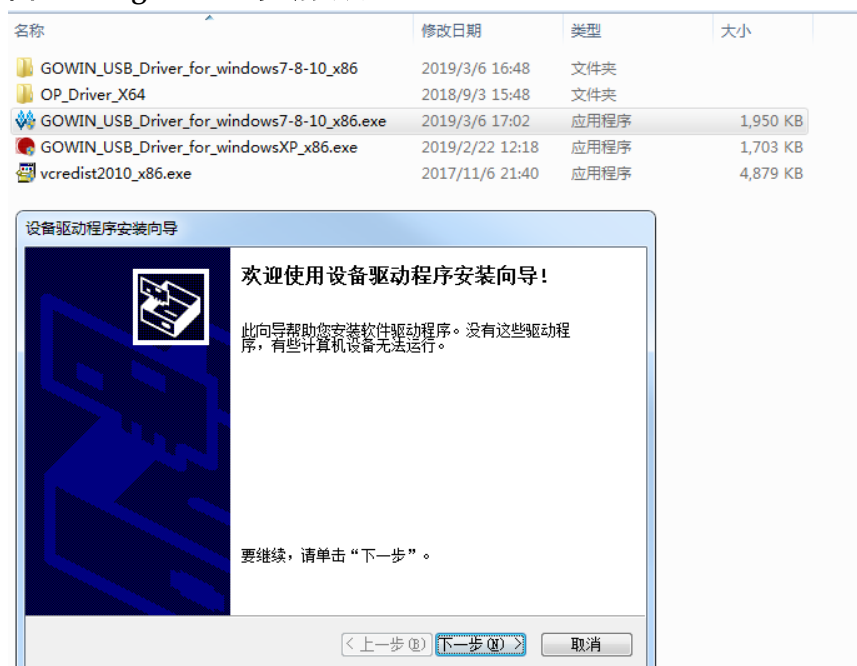
2.2.2 安装方式二

在官网(www.gowinsemi.com.cn/faq.aspx)下载单独的 Gowin Programmer 安装包, 安装包解压后打开 programmer2driver 目录, 如图 2-2 所示, 根据自己的电脑系统选择对应驱动安装即可。

注!

Windows XP 系统下安装 GWU2X 驱动时, 需要先插入对应 USB 设备, 否则无法正常安装。

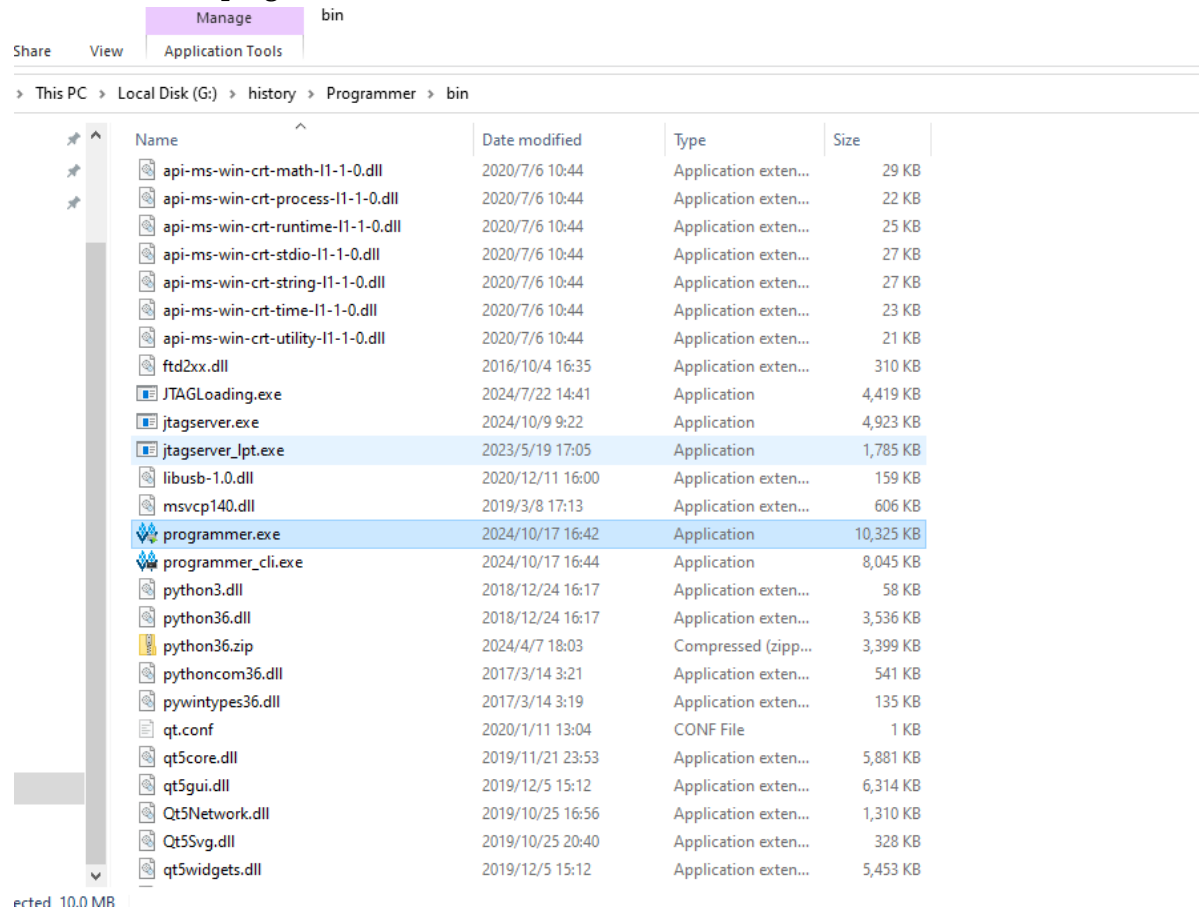
图 2-2 Programmer 驱动安装



2.2.3 启动 Programmer 工具链

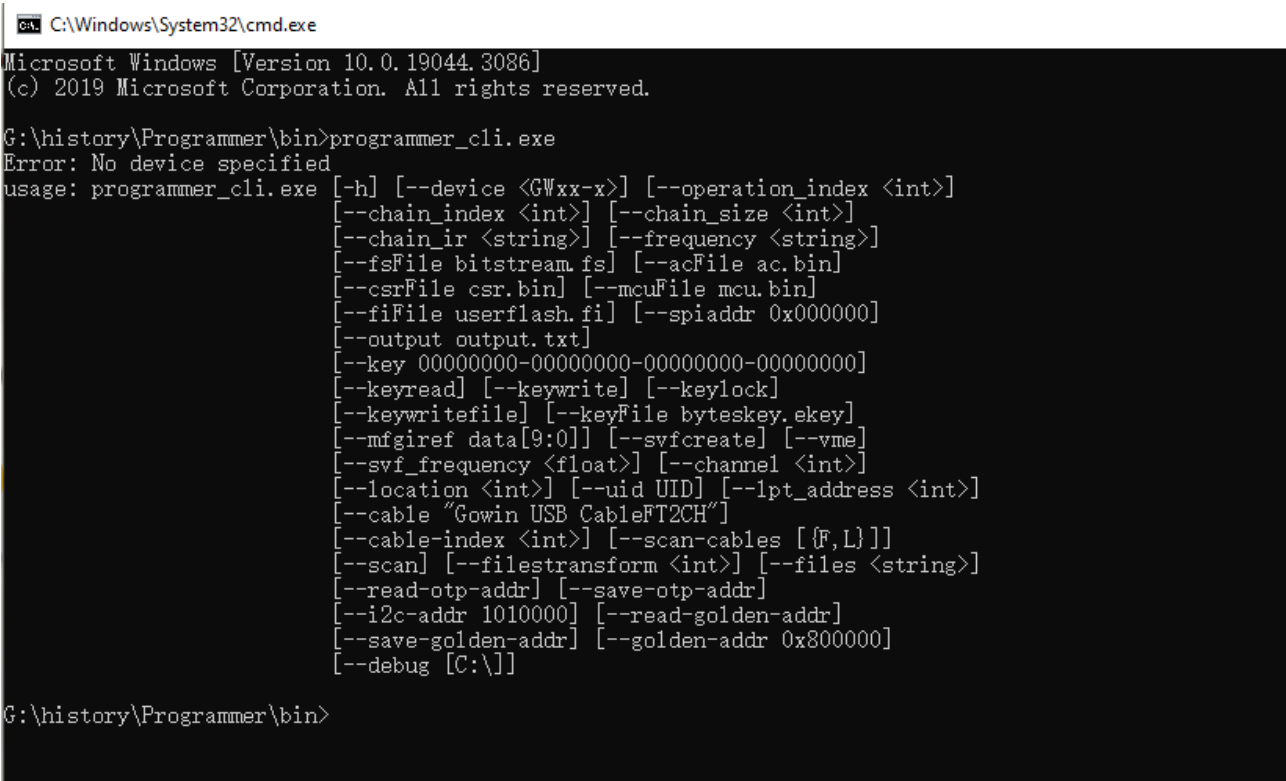
- Programmer 安装完成后执行文件在安装目录下 \x.x\Programmer\bin\; 双击 programmer.exe 可启动图形界面配置软件，如图 2-3 所示。

图 2-3 programmer.exe 启动



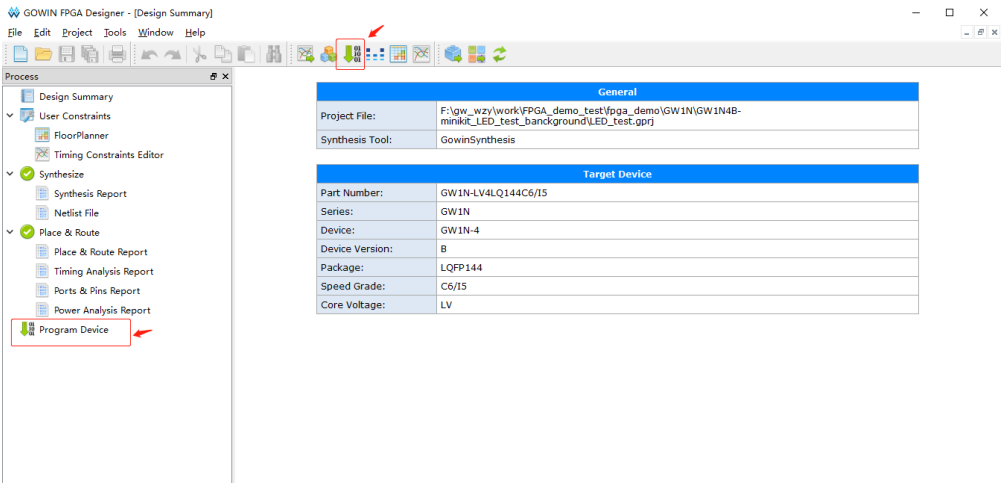
- 对于命令行类软件，请在 CMD 内打开，例如打开 programmer_cli.exe。

图 2-4 programmer_cli.exe 启动



- 可以通过云源软件的快捷键启动本软件，如图 2-5 所示。

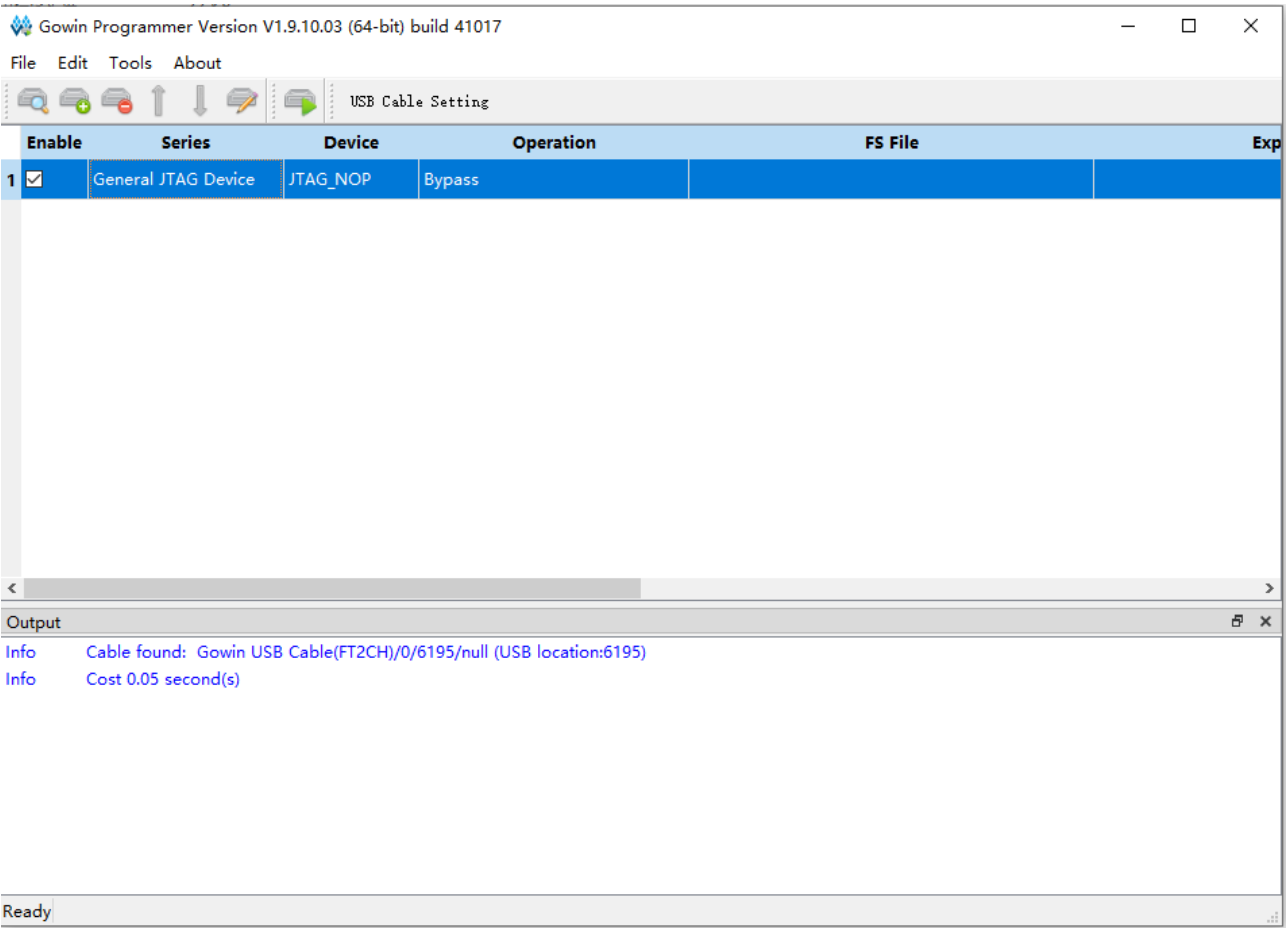
图 2-5 云源软件中启动 Programmer 快捷键



2.3 软件界面

高云编程器软件界面分为菜单栏、工具栏、设备表格和输出面板，如图 2-6 所示。

图 2-6 Programmer Main Window



设备表格显示了将要进行编程的菊花链中所有设备，这些设备可以是自动扫描获得，也可以通过手动配置获得。表格的每一行代表一个设备，可以通过 **Enable** 列来选择是否对设备进行编程。

设备表格包含 **Enable**, **Family**, **Device**, **Operation**, **FS File**, **Checksum**, **User Code** 和 **IDCODE** 选项。其中 **Enable**, **Family**, **Device**, **Operation** 和 **FS File** 是可编辑的，通过鼠标单击可进行编辑。双击其它不可编辑的选项可打开设备配置对话框（**Device Configuration Dialog**）进行编程操作的配置，详见 [3.4 配置设备编程操作](#)。

输出面板包含 **Output**、**Error**、**Warning**、**Info**，分别显示所有信息、错误信息、警告信息和指示信息。

注!

当设备设为 **Enable** 列不勾选时，编程器会将设备视为不在链中，设备所在行将不可编辑。

2.4 查看软件版本

Gowin Programmer 与 GOWIN IDE 具备独立的软件版本号, 可通过打开软件界面的“About”菜单查看版本号, 如下图 2-7 所示。

图 2-7 查看版本号



2.5 Linux 系统下 cable 的权限配置

如图 2-8 所示, Gowin_USB_Cable_Installer.sh 是一个脚本文件, Makefile 是一个文本文件。二者皆可用于 cable 权限的改变。

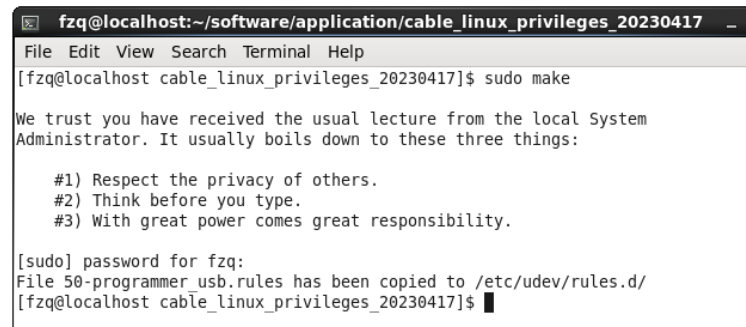
图 2-8 文件列表

- 50-programmer_usb.rules
- Gowin_USB_Cable_Installer.sh
- Makefile
- readme.txt

2.5.1 Makefile

打开终端, 输入 `sudo make` 指令或切换用户至 root 权限, 输入 `make` 指令, 显示 `File 50-programmer_usb.rules has been copied to /etc/udev/rules/d/` 表示安装成功 (部分 centos6 系统需要重启才能生效), 如图 2-9、图 2-10 所示。

图 2-9 Makefile 普通用户安装示例



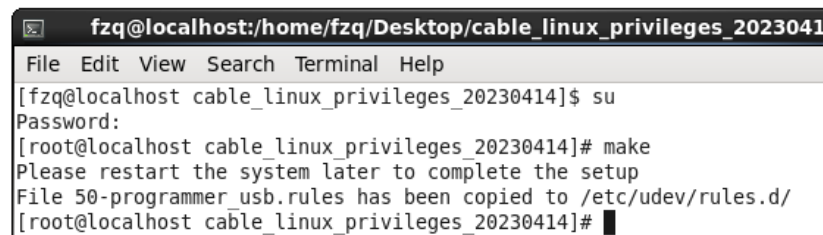
```
fzq@localhost:~/software/application/cable_linux_privileges_20230417
File Edit View Search Terminal Help
[fzq@localhost cable_linux_privileges_20230417]$ sudo make

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for fzq:
File 50-programmer usb.rules has been copied to /etc/udev/rules.d/
[fzq@localhost cable_linux_privileges_20230417]$
```

图 2-10 Makefile root 用户安装示例

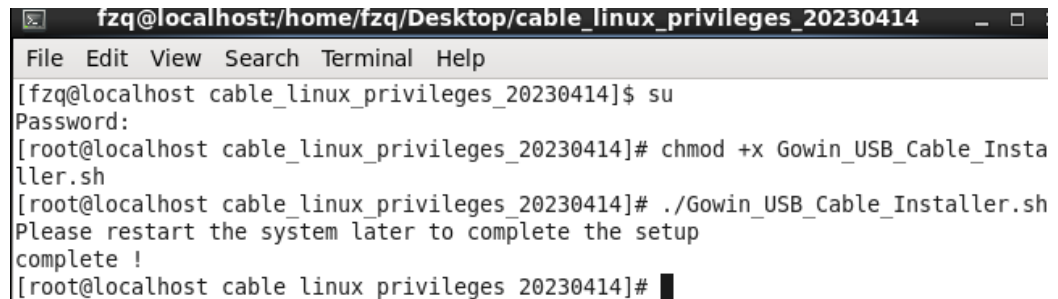


```
fzq@localhost:/home/fzq/Desktop/cable_linux_privileges_20230414
File Edit View Search Terminal Help
[fzq@localhost cable_linux_privileges_20230414]$ su
Password:
[root@localhost cable_linux_privileges_20230414]# make
Please restart the system later to complete the setup
File 50-programmer usb.rules has been copied to /etc/udev/rules.d/
[root@localhost cable_linux_privileges_20230414]#
```

2.5.2 Gowin_USB_Cable_Installer.sh

首先打开 `programmer` 所在文件夹, 检查图 2-8 中 4 个文件是否都存在。之后打开终端, 切换用户至 `root` 权限, 给 `Gowin_USB_Cable_Installer.sh` 赋予使用权限, 运行 `Gowin_USB_Cable_Installer.sh`, 显示 `complete` 表示安装成功, 如图 2-11 所示。

图 2-11 脚本安装示例



```
fzq@localhost:/home/fzq/Desktop/cable_linux_privileges_20230414
File Edit View Search Terminal Help
[fzq@localhost cable_linux_privileges_20230414]$ su
Password:
[root@localhost cable_linux_privileges_20230414]# chmod +x Gowin_USB_Cable_Installer.sh
[root@localhost cable_linux_privileges_20230414]# ./Gowin_USB_Cable_Installer.sh
Please restart the system later to complete the setup
complete !
[root@localhost cable_linux_privileges_20230414]#
```

3 图形界面 Programmer 编程下载流程

编程下载是指通过下载线将数据流文件传输到 FPGA 设备 SRAM，嵌入式 Flash 或外部 Flash 中的过程，在编程器中具体的流程如下：开始>设置下载线>扫描设备>配置设备编程操作>编程下载

1. 设置下载线（可选）：为编程下载选择合适的下载线的类型，端口，频率等。

注！

默认情况下，编程器会尝试使用第一个可见端口，默认频率选择为 2.5MHz。

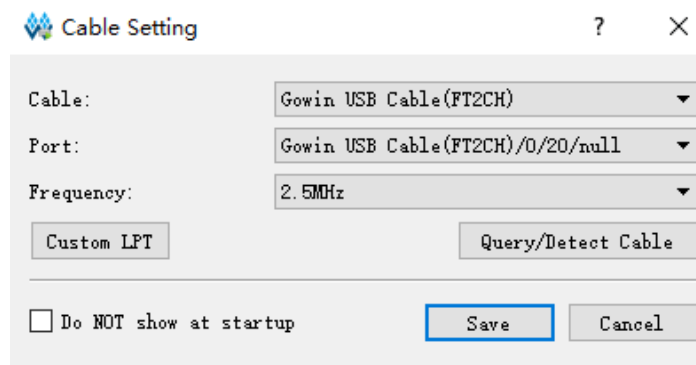
2. 配置菊花链和编程操作：将设备表格中的菊花链配置为符合物理上菊花链实际连接的状态，并为每个设备选择编程操作及所需的数据文件，最上层为靠近 Programmer 一端。
3. 编程下载：对配置完成的菊花链进行编程下载，最终结果会显示在输出面板中。

3.1 设置下载线

设置下载线是指选择可用的下载线的类型，端口和频率等信息。通过菜单栏“Edit > Setting > Cable Setting”打开“Cable Setting”对话框。当前支持三种类型的下载线：Gowin USB Cable（FTDI）、Gowin USB Cable（GWU2X）和 LPT。

1. Gowin USB Cable (FT2CH)如图 3-1 所示。
 - Cable: 下载线类型，选择 Gowin USB Cable。
 - Port: 下载线可用的端口，默认选择为第一个可用的端口。最后一个英文字符 A 表示下载器通道编号，有 S、A、B 三种通道。
 - Frequency: JTAG 频率，可选 2MHz, 2.5MHz,15MHz,10MHz 等，默认为 2.5MHz。

图 3-1 Gowin USB Cable (FT2CH)

**注!**

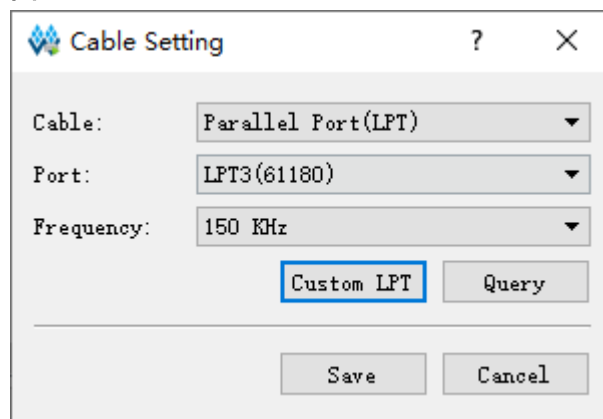
在使用 I2C 协议时，针对不同版本的下载器，TDI 与 TDO 管脚的连接方式有所不同，以下两个版本的下载器都需要将 TCK 管脚连接至 SCL 管脚。

- 4.0 版本下载器需要外部连接 TDI 和 TDO 管脚至 SDA 管脚。
- 4.1 版本下载器直接连接 TDI 管脚到 SDA 管脚，同时确保 FTDI 芯片内部的第 27 管脚被拉低。
- 4.1 版本下载器增加了 UART 协议的支持，连接要求如下：
 - TCK 管脚连接至 TX;
 - TDI 管脚连接至 RX;
 - 内部拉低 FTDI 芯片的第 17 管脚。

2. 并口（LPT）如图 3-2 所示。

- Cable: 下载线类型，选择 Parallel Port (LPT)。
- Port: 下载线可用的端口，依据计算机的设备管理器 PCI 属性选择。
- Frequency: 频率，可选 150KHz。

图 3-2 LPT

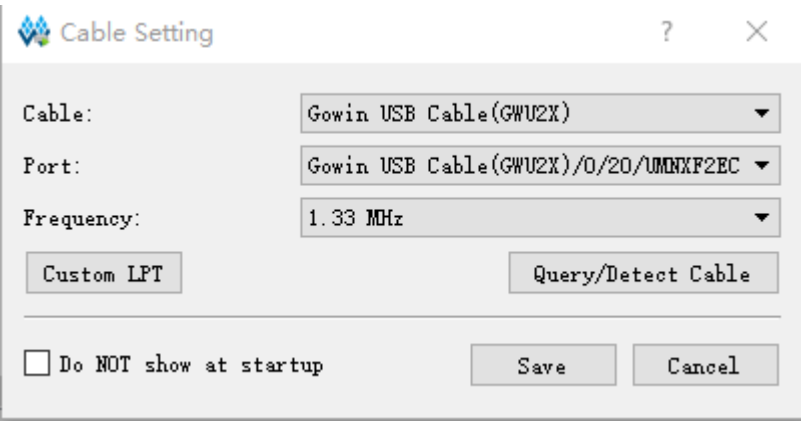


3. Gowin USB Cable (GWU2X)如图 3-3 所示。

- Cable: 下载线类型，选择 Gowin USB Cable(GWU2X)。
- Port: 下载线可用的端口，默认选择为第一个可用的端口。最后一个英文字符 A 表示下载器通道编号，有 S、A、B 三种通道。

- Frequency: 频率，默认为 1.33MHz。

图 3-3 Gowin USB Cable (GWU2X)



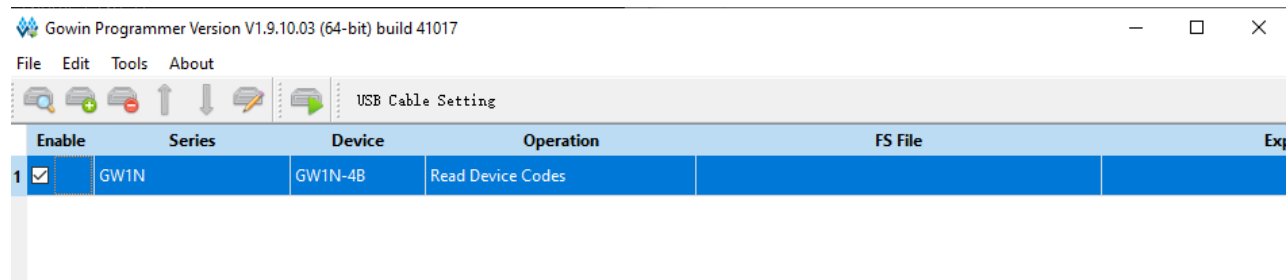
注!
在使用 I2C 协议时，针对不同版本的下载器，TDI 与 TDO 管脚的连接方式有所不同，以下两个版本下载器都需要将 TCK 管脚连接至 SCL 管脚。

- 5.0 版本下载器需要外部连接 TDI 和 TDO 管脚至 SDA 管脚。
- 5.1 版本下载器直接连接 TDI 管脚到 SDA 管脚,同时确保 U2X 芯片内部的 P14 管脚被拉低。

3.2 扫描设备菊花链

编程器支持自动扫描与计算机连接的设备菊花链。通过菜单栏工具栏“”扫描与计算机相连的菊花链。扫描完成后，所有的设备将按其所在链中顺序依次列于主界面的设备表格中，如图 3-4 所示。

图 3-4 Device Table



注!
部分设备具有相同的 ID(如 GW2A-18/GW2AR-18),扫描完成后根据提示指定对应的设备。

编程器支持手动配置设备菊花链，包括添加设备、移除设备、修改设备在链中位置等操作。

3.3 配置设备菊花链

3.3.1 添加设备

1. 通过菜单栏“Edit > Add Device”或工具栏“”添加一个新设备；
2. 单击“Family”单元格通过下拉菜单选择设备家族；

- 单击“Device”单元格通过下拉菜单选择设备具体型号。

注!

当有设备被选中时，新设备添加在选中位置，否则新设备添加在菊花链尾。

3.3.2 移除设备

选中设备所在行，通过菜单栏“Edit > Remove Device”或工具栏“”移除设备。

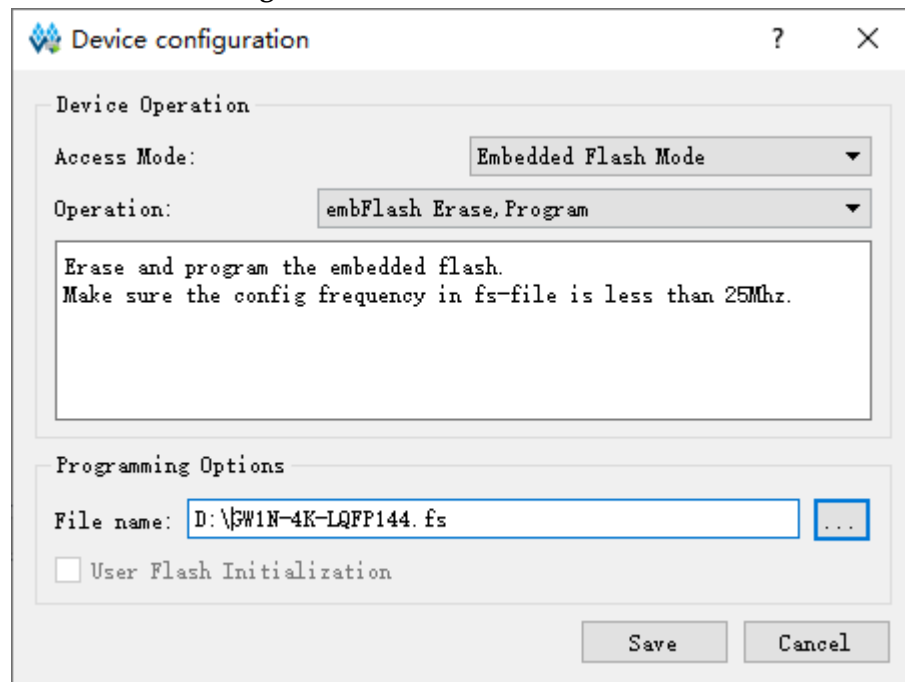
3.3.3 修改设备在链中位置

选中设备所在行，通过菜单栏“Edit > Up”（或 Down）或工具栏“”（或“”）调节设备在链中位置。

3.4 配置设备编程操作

选中设备所在行，通过菜单“Edit > Configure Device”或工具栏“”或双击“Operation”打开“Device Configuration”对话框，如图 3-5 所示。

图 3-5 Device Configuration Interface



- Access Mode: 选择设备的编程模式。
- Operation: 选择设备编程操作，详见表 3-1。
- Instruction Register Length: 当设备选择为 JTAG-NOP 时，选择设备指令寄存器长度。
- Programming File: 选择编程数据文件。
- Device: 当编程模式选择为 External Flash Mode 时，选择外部 Flash 型

号。

- **Start Address:** 当编程模式选择为 External Flash Mode 时，选择外部 Flash 起始地址。

表 3-1 Device 操作描述

Access Mode	Operation	Description
SRAM Mode	Bypass	Bypass
	Read Device Code	可读设备ID ,User Code, Status Code
	Read User Code	读出设备User Code
	Read Status Register	读出设备状态
	Reprogram	—
	SRAM Erase	擦除SRAM数据
	SRAM Program	配置数据流文件到FPGA SRAM
	SRAM program and Verify	将数据写入SRAM并进行验证
Embedded Flash Mode	embFlash Erase, Program	先擦除内嵌Flash，然后将数据写入。
	embFlash Erase, Program, Verify	先擦除内嵌Flash，然后将数据写入并进行验证。
	EmFlash Erase Only	仅擦除内嵌Flash
External Flash Mode	exFlash Erase, Program	先擦除外部Flash，然后将数据写入外部Flash。
	exFlash Erase, Program, Verify	先擦除外部Flash，然后将数据写入并进行验证。
	exFlash Program Without Erasure	将数据写到外部Flash，不擦除。
	exFlash Bulk Erase	擦除外部Flash
	exFlash Verify	验证外部Flash数据
	exFlash Erase, Program in bscan	使用bscan模式，先擦除外部Flash，然后将数据写入外部Flash。
	exFlash Erase, Program, Verify in bscan	使用bscan模式，先擦除外部Flash，然后将数据写入并进行验证。
	exFlash Verify in bscan	—
	exFlash Program in bscan without erasure.	使用bscan模式，将数据写到外部Flash，不擦除。
	exFlash Bulk Erase in bscan	使用bscan模式，验证外部Flash数据。
	exFlash C Bin Erase, Program	先擦除外部Flash，然后将RISC-V的bin文件，写入到外部Flash。
	exFlash C Bin Erase, Program, Verify	先擦除外部Flash，然后将RISC-V的bin文件，写入到外部Flash，并进行验证。
	exFlash C Bin Program	将RISC-V的bin文件，写入到外部Flash。
Slave SPI Mode	Slave SPI Read ID Code	SSPI模式下读设备ID
	Slave SPI Scan exFlash	SSPI模式下扫描外部Flash
	Slave SPI Program SRAM	SSPI模式下将数据写入SRAM

注!

GW2A/GW2AR 系列芯片没有 embedded Flash，不支持此模式。

3.4.1 配置 SRAM 模式

1. 选中设备所在行，通过菜单“Edit > Configure Device”或工具栏“”或双击“Operation”列单元格打开 Device Configuration 对话框；
2. Access Mode 下拉列表中选择 SRAM Mode；
3. Operation 下拉列表中选择期望进行的操作；
4. 如果设备非高云设备，需要手动指定指令寄存器的长度或指定设备的 BSDL 文件由编程器读取指令寄存器的长度。
5. 点击“Save”完成配置。

注!

非高云设备（JTAG-NOP）只支持 Bypass 操作。

3.4.2 小蜜蜂系列 FPGA 配置嵌入式 Flash 模式

GW1N/GW1NZ 系列设备具有嵌入式 Flash，编程模式可以选择为嵌入式 Flash 模式。

1. 选中设备所在行，通过菜单“Edit > Configure Device”或工具栏“”或双击“Operation”列单元格打开 Device Configuration 对话框；
2. Access Mode 下拉列表中选择“Embedded Flash Mode”；
3. Operation 下拉列表中选择期望进行的操作；
4. Programming File 中选择对应的编程数据流文件；
5. 点击“Save”完成配置。

3.4.3 配置外部 Flash 模式

高云编程器支持使用外部 Flash 进行编程，外部 Flash 模式配置过程如下：

1. 选中设备所在行，通过菜单“Edit > Configure Device”或工具栏“”或双击“Operation”列单元格打开 Device Configuration 对话框；
2. Access Mode 下拉列表中选择 External Flash Mode；
3. Operation 下拉列表中选择期望进行的操作；
4. 如果 Operation 选择 exFlash Program，需要在 Programming File 中选择对应的编程数据流文件；
5. 关于外部 Flash 型号，小蜜蜂和晨熙家族支持读指令为 0x03 或 0x0B 的 SPI Flash；
6. 如果菜单中没有所选 Flash，请选择 Generic Flash 进行自动烧录尝试；

7. 选择外部 Flash 的起始地址，当前默认是 0x000000；
8. 点击“Save”完成配置。

3.4.4 Slave SPI 模式

Slave SPI Mode 模式下需将下载线接到 SSPI 专用的 pin 上，请参考文档 [UG290, Gowin FPGA 产品编程配置手册](#)。

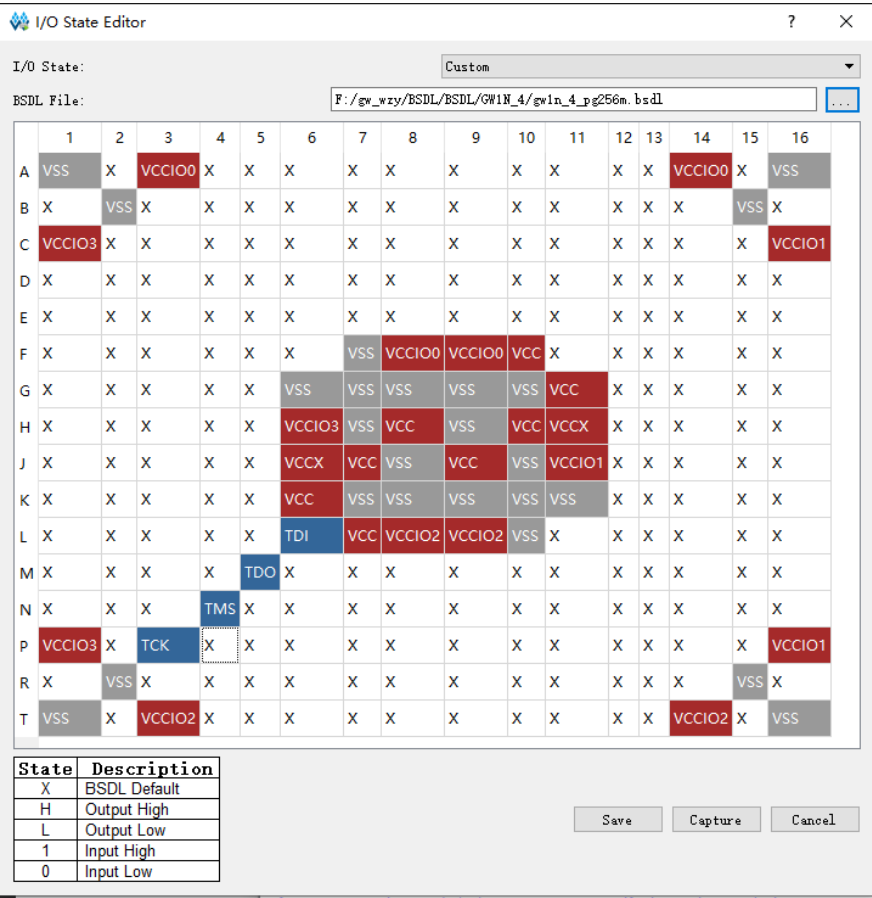
1. 选中设备所在行，通过菜单“Edit > Configure Device”或工具栏“”或双击“Operation”列单元格打开 Device Configuration 对话框；
2. Access Mode 下拉列表中选择 Slave SPI Mode；
3. Operation 下拉列表中选择期望进行的操作；
4. 如果 Operation 选择 Slave SPI Program SRAM，需要在 Programming File 中选择对应的编程数据流文件；
5. 点击“Save”完成配置。

3.5 编辑引脚状态

编程器使用 I/O State Editor 工具编辑输入输出引脚的值，可以设置引脚在编程下载之前的状态。

1. 选中设备所在行，通过菜单栏“Edit > I/O State”或右键单击“I/O State”打开 I/O State Editor；
2. 选择符合设备型号和封装的 BSM 文件；
3. 通过单击单元格位置改变对应引脚的状态或右键菜单为所有引脚设置同样的状态。

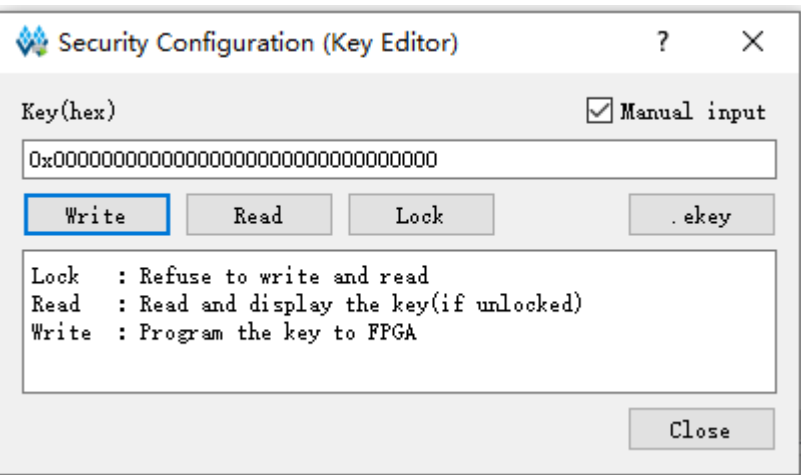
图 3-6 I/O State Editor



3.6 设备安全

当使用加密的比特流文件编程时，需要将比特流文件的解密 key 写入 FPGA。选择设备所在行，通过菜单栏 “Edit > Security Key Setting” 或右键单击 “Security Key Setting” 打开 “Security Configuration” 对话框，如图 3-7 所示。

图 3-7 Security Configuration



- Manual input: 是否明文输入密钥的选择项。

- **.ekey**: 打开密钥文件。
- **Write**: 将指定的 key 值写入 FPGA 中。
- **Read**: 在未锁定的情况下, 读出并显示 FPGA 内的 key 值。
- **Lock**: 锁定 FPGA 内的 key, 锁定后无法再进行读写。

解密密钥的写入有以下两种方式:

明文写入密钥

解密密钥写入成功后可以选择界面上的读取指令回读写入的密钥进行验证。

密钥写入成功后, 用户也可选择 **lock** 命令将密钥“锁死”在 FPGA 内部, 之后任何对密钥的读取和写入操作都将无效: 密钥值无法进行修改, 读取的密钥所有位全部为“1”。

密钥文件写入

打开密钥文件, 之后点击“Write”按钮, 会先将密钥写入开发板, 之后读取密钥进行校验, 如果校验成功, 则会将密钥锁死在 FPGA 内部。

设置解密密钥后, 加密的比特流数据只有与解密密钥匹配成功后才能唤醒。非加密比特流数据的配置不受密钥影响。

注!

高云半导体 FPGA 的密钥初始值所有位全部为 0, 当把密钥值的某一位修改为 1 后便无法改回 0。例如, 用户的某次操作中写入的密钥值 00000000-00000000-00000000-00000001, 之后修改此器件的密钥时最低位也必须是 1。详细信息请参考 [TN654, GW2A\(R\) 系列 FPGA 产品 AES 密钥编程指南](#)。

3.7 编程下载

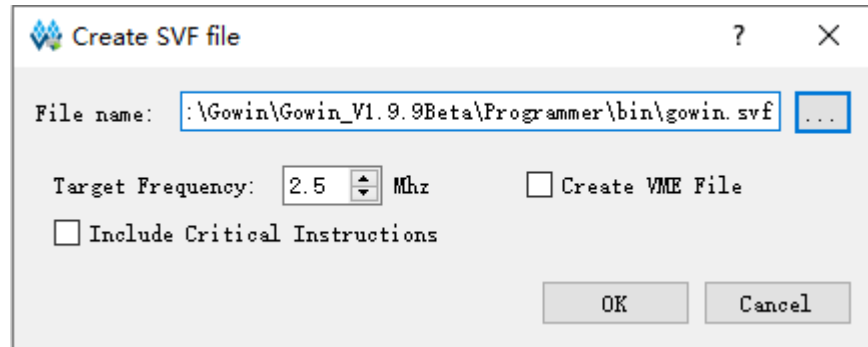
当下载线和菊花链都配置完成后, 通过菜单栏“Design > Run”或工具栏“”对设备进行编程下载, 最终结果显示在输出面板中。

3.8 创建 SVF 文件

支持使用 fs 文件创建 SVF 文件, 目前仅支持 GW1N-4 的 SVF 文件的创建。

1. 按照 GW1N/GW1NZ 配置嵌入式 Flash 模式配置, device 选择 GW1N4。
2. 选中设备链, 通过菜单栏“Edit > SVF File Create”或者右键单击“SVF File Create”打开 Create SVF file 对话框。
3. 在对话框中 File name 栏可命名生成 SVF 文件名, 可选择保存 svf 的路径, 如图 3-8 所示。
4. 单击“OK”, 完成 SVF 文件的创建。

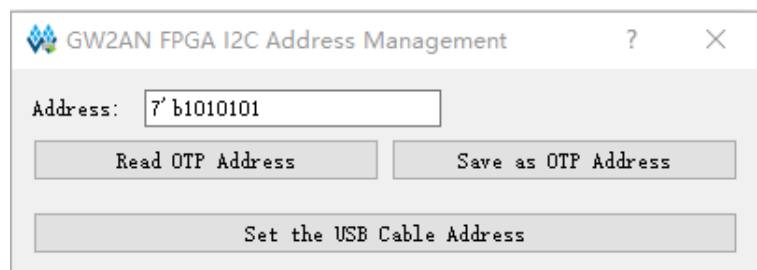
图 3-8 Create SVF File



3.9 I2C Slave 地址操作

支持 I2C slave 地址的读取、写入、设置操作，目前该操作仅支持 GW2AN-18X 和 GW2AN-9X 的设备。

图 3-9 I2C 的 slave 地址操作



- Read OTP Address: 读取 I2C slave 地址。
- Save as OTP Address: 将指定的 I2C slave 地址写入 FPGA 中。
- Set the USB Cable Address: 在使用 I2C 模式操作 FPGA 时，需要设置 I2C slave 地址。

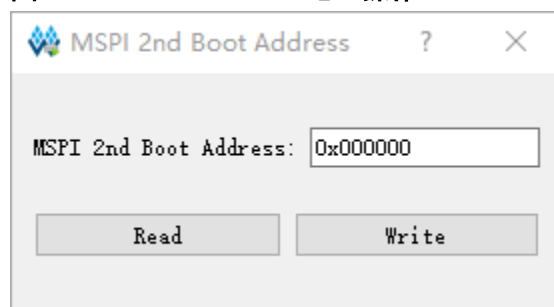
注！

I2C slave 地址仅支持 2 bits 配置，默认地址为 7'b1010**00**，只能将黄色字符 0 修改为 1，不能改回 0。

3.10 MSPI 2nd Boot 地址操作

支持 Golden Image 地址的读取、写入操作，范围是 0x000000-0xF00000，目前该操作仅支持 GW5A-25 A 的设备。

图 3-10 MSPI 2nd Boot 地址操作

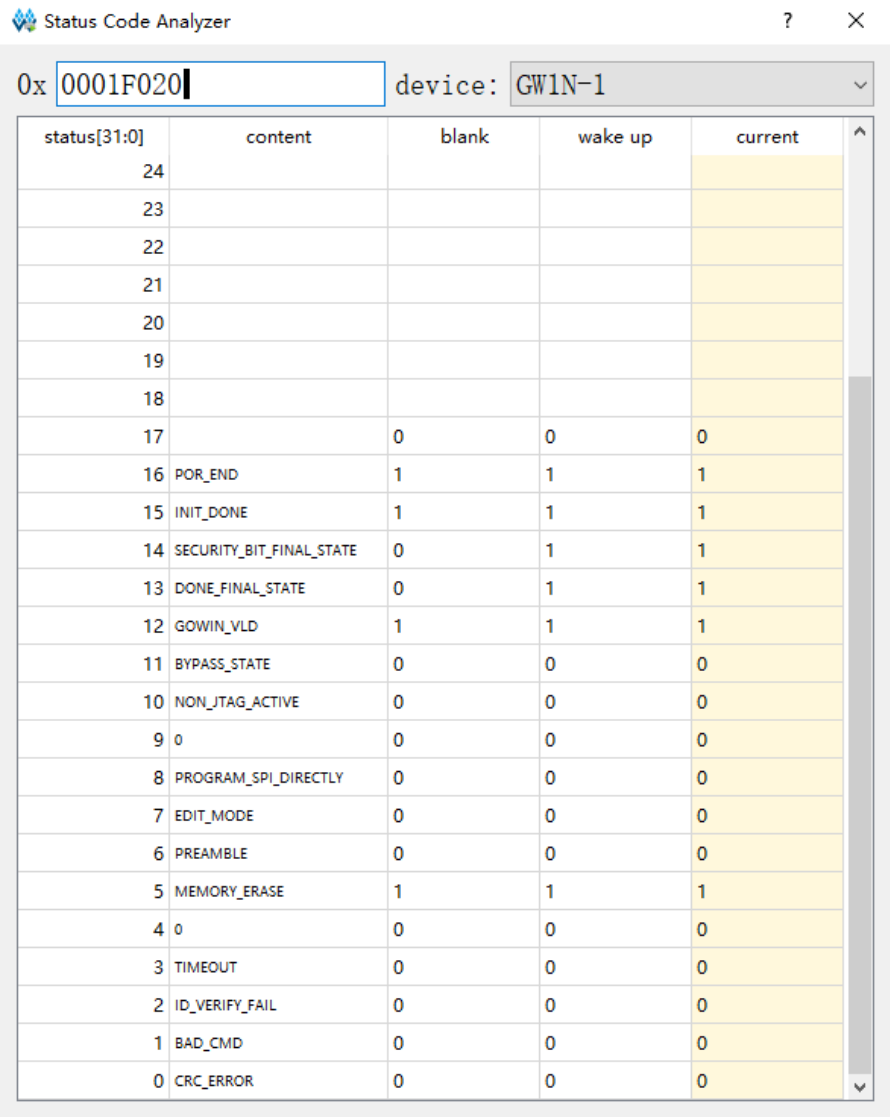


- Read: 读取 MSPI 2nd boot 地址。
- Write: 将指定的 MSPI 2nd boot 地址写入 FPGA 中。

3.11 状态码分析

当使需要查看当前开发板的状态时，可以通过分析状态码的方法，得到开发板的状态。通过点击菜单栏 “Tools > Analyzer Viewer” 选项或找到状态码显示行，右键单击 “Analyze Status Code”打开“Status Code Analyzer”对话框，如下图 3-11 所示。

图 3-11 状态码分析界面



首先选择合适开发板（**device** 列表可选），之后填入状态码，下方列表中会输出状态码的各个 **bit** 位的结果。

3.12 User Flash 初始化

小蜜蜂系列为用户提供了 User Flash 空间。User Flash 数据可通过 Programmer 在烧录内置 Flash 的同时可烧录 user flash 空间。从安全设计上考虑,在 Programmer 端的这个操作仅支持 User Flash 烧录,不支持回读。用户在烧录的同时,可选择以.fi 为文件后缀的 user flash 初始化文件,如图 3-12 所示。

图 3-12 User Flash Initialization

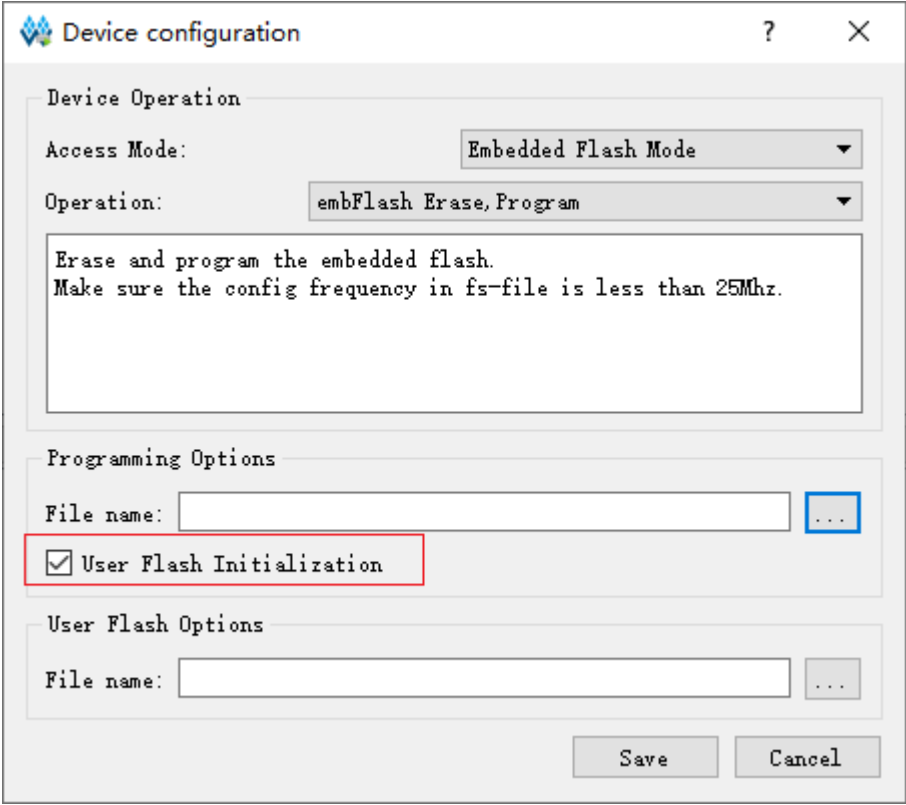


表 3-2 User Flash 参考表

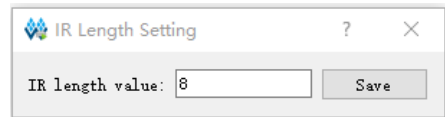
Series	Device	Flash Type	Address	Data Width
GW1N	GW1N-1	FLASH96K	48*64	32Bits
	GW1N-1S			
	GW1N-2	FLASH256K	128*64	
	GW1N-2B			
	GW1N-4			
	GW1N-4B			
	GW1N-6	FLASH608K	304*64	
	GW1N-9			
GW1NR	GW1NR-4	FLASH256K	128*64	
	GW1NR-4B			
	GW1NR-9	FLASH608K	304*64	

Series	Device	Flash Type	Address	Data Width
GW1NS	GW1NS-2	FLASH128K	32786	
GW1NSR	GW1NSR-2	FLASH128K	32786	
GW1NZ	GW1NZ-1	FLASH64KZ	32*64	

3.13 JTAG 状态机 shift-ir 宽度调整

支持调整 jtag 状态机的 shift-ir 的宽度，此功能只在菊花链配置时使用。

图 3-13 jtag 状态机的 shift-ir 的宽度调整



- IR length value: jtag 状态机 shift-ir 的宽度数值。
- Save: 将选定的数值写入 FPGA 中。

3.14 远程配置和烧录功能

支持远程烧录和配置功能，在使用此功能时，需要确定两台 pc 之间能够正常进行 tcp/ip 通信。

1. 需要在远端的 pc 启动 jtagserver.exe，在 programmer 同级目录下，找到 jtagserver 文件，双击打开如图 3-14 所示。

图 3-14 jtagserver 位置

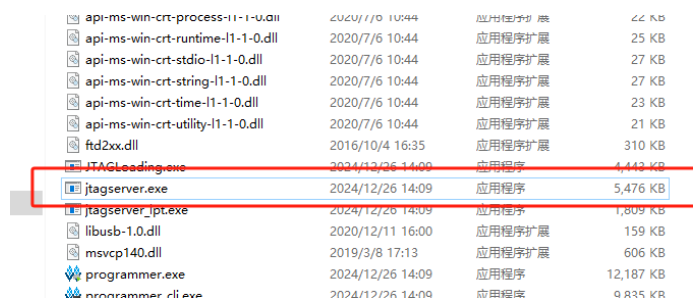
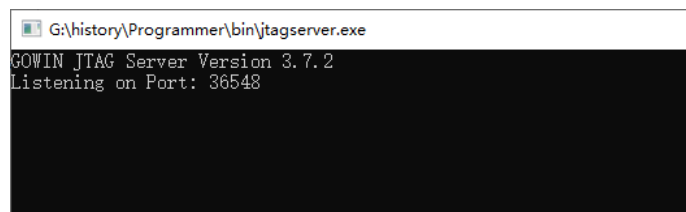


图 3-15 jtagserver 启动状态



jtagserver 会监听指定端口，用于支持客户端的链接，端口的默认值是 36548，如果想要使用其他端口号，需要在 programmer 的同级文件夹 data 里面找到 config 文件夹，里面有一个 gvc.ini，可以修改端口号。

图 3-16 jtagserver 启动状态

```
[remote]
host = 127.0.0.1
port = 36548
```

- host: 远端 ip 地址。
 - port: 监听的端口。
2. 需要在近端的 pc 启动 programmer.exe, 在“Tools”的目录下找到“Add GVC Server”选项, 单击打开, 弹出 GVC 界面, 如图 3-18 所示。

图 3-17 Add GVC Server 选项

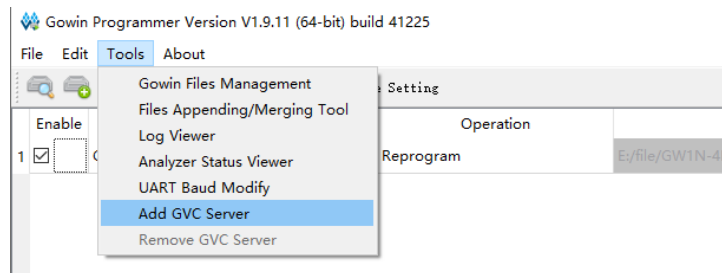
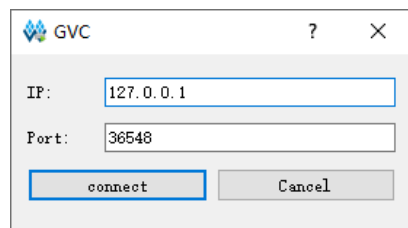


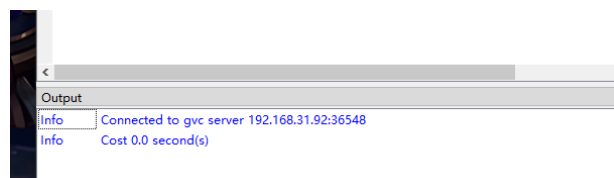
图 3-18 GVC 连接界面



- IP: 远程开启服务器的 ip 地址。
- Port: 远端服务器监听的端口。
- Connect: 连接远端服务器。
- Cancel: 取消操作, 退出该页面。

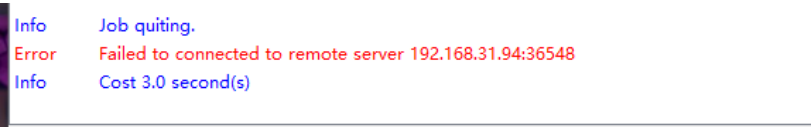
点击“Connect”后, 成功连接远端服务器会弹出如图 3-19 所示界面。

图 3-19 连接成功界面



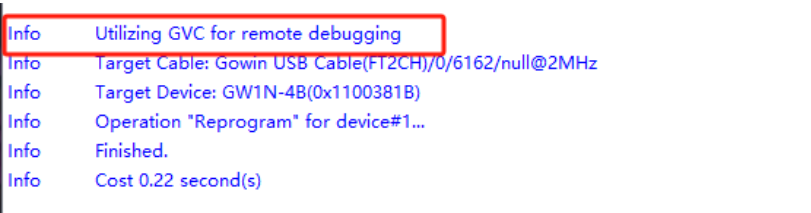
如果连接失败如图 3-20 所示。

图 3-20 连接失败界面



3. 在成功连接之后，所有的远程操作与本地操作一致，并无任何区别，只是在操作时会打印远程调试的信息加以区别。

图 3-21 远程操作输出



4. 操作完毕后，可以在“Tools”的目录下找到“Remove GVC Server”选项，点击之后即可退出远程配置模式。

图 3-22 退出远程配置操作

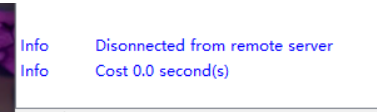


表 3-3 远程烧录配置功能与设备参考表

器件	功能
GW1N-1	'Read Device Codes'
GW1N-1S	'Read User Code'
GW1N-4	'Read Status Register'
GW1NSER-4C	"Reprogram"
GW1NR-4	"SRAM Erase"
GW1AN-1C	'SRAM Program'
	'SRAM Program JTAG 1149'
	'embFlash Erase Only'
	'embFlash Erase,Program'
	'embFlash Erase,Program,Verify'
	'exFlash Erase,Program thru GAO-Bridge'
	'exFlash Erase,Program,Verify thru GAO-Bridge'
	'exFlash C Bin Erase,Program thru GAO-Bridge'
	'exFlash C Bin Erase,Program,Verify thru GAO-Bridge'
	'exFlash Bulk Erase thru GAO-Bridge'
	'exFlash Verify thru GAO-Bridge'
	'exFlash Erase,Program'
	'exFlash Erase,Program,Verify'
	'exFlash C Bin Erase,Program'
	'exFlash C Bin Erase,Program,Verify'

器件	功能
	'exFlash Background Erase,Program' 'exFlash Background Erase,Program,Verify' 'exFlash Bulk Erase' 'exFlash Verify'
GW1N-1P5 GW1N-1P5B GW1N-1P5C GW1N-2 GW1N-2B GW1N-2C GW1N-4D GW1N-4B GW1N-9 GW1N-9C GW1NZ-1 GW1NZ-1C GW1NZ-2B GW1NZ-2C GW1NS-4 GW1NS-4C GW1NSR-4 GW1NSR-4C GW1NRF-4B GW1NR-2 GW1NR-2B GW1NR-2C GW1NR-4D GW1NR-4B GW1NR-9 GW1NR-9C	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" 'SRAM Program' 'SRAM Program JTAG 1149' 'embFlash Background Erase Only ' 'embFlash Background Erase,Program' 'embFlash Background Erase,Program,Verify' 'embFlash Erase Only'、 'embFlash Erase,Program' 'embFlash Erase,Program,Verify' 'exFlash Erase,Program thru GAO-Bridge' 'exFlash Erase,Program,Verify thru GAO-Bridge' 'exFlash C Bin Erase,Program thru GAO-Bridge' 'exFlash C Bin Erase,Program,Verify thru GAO-Bridge' 'exFlash Bulk Erase thru GAO-Bridge' 'exFlash Verify thru GAO-Bridge' 'exFlash Erase,Program' 'exFlash Erase,Program,Verify' 'exFlash C Bin Erase,Program' 'exFlash C Bin Erase,Program,Verify' 'exFlash Background Erase,Program' 'exFlash Background Erase,Program,Verify' 'exFlash Bulk Erase' 'exFlash Verify'
GW1AN-9A GW2AN-9X GW2AN-18X	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" 'SRAM Program' 'SRAM Program JTAG 1149' 'sFlash Erase,Program,Verify thru GAO-Bridge' 'sFlash Background Erase,Program,Verify thru GAO-Bridge' 'sFlash Erase,Program' ' sFlash Erase,Program,Verify' 'sFlash Background Erase,Program' 'sFlash Background Erase,Program,Verify' 'sFlash Bulk Erase'、 'sFlash Verify Only'
GW2A-55 GW2A-55C GW2A-18	'Read Device Codes' 'Read User Code' 'Read Status Register'

器件	功能
GW2A-18C GW2AN-55C GW2AR-18 GW2AR-18C GW2ANR-18C	"Reprogram" "SRAM Erase"、'SRAM Program' 'SRAM Program JTAG 1149' 'exFlash Erase,Program thru GAO-Bridge' 'exFlash Erase,Program,Verify thru GAO-Bridge' 'exFlash C Bin Erase,Program thru GAO-Bridge' 'exFlash C Bin Erase,Program,Verify thru GAO-Bridge' 'exFlash Bulk Erase thru GAO-Bridge' 'exFlash Verify thru GAO-Bridge' 'exFlash Erase,Program' 'exFlash Erase,Program,Verify' 'exFlash C Bin Erase,Program' 'exFlash C Bin Erase,Program,Verify' 'exFlash Background Erase,Program' 'exFlash Background Erase,Program,Verify' 'exFlash Bulk Erase' 'exFlash Verify'
GW5AT-15A GW5AT-15B GW5AT-60B GW5AT-60ES GW5AT-75B GW5AT-75C GW5AT-138 GW5AT-138B GW5AT-138C GW5AST-138B GW5AST-138C GW5AS-25A GW5AS-25B GW5AS-138B GW5AS-138C GW5A-60B GW5A-25A GW5A-25B GW5A-138B GW5A-138C GW5AR-25A GW5AR-25B GW5ART-15A GW5ART-15B GW5ANT-15A GW5ANRT-15A	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" 'SRAM Program' 'SRAM Program JTAG 1149' 'exFlash Erase,Program thru GAO-Bridge 5A' 'exFlash Erase,Program,Verify thru GAO-Bridge 5A' 'exFlash Verify thru GAO-Bridge 5A' 'exFlash Erase,Program 5A' 'exFlash Erase,Program,Verify 5A' 'exFlash C Bin Erase,Program 5A' 'exFlash C Bin Erase,Program,Verify 5A' 'exFlash Bulk Erase 5A' 'exFlash Verify 5A'

注！

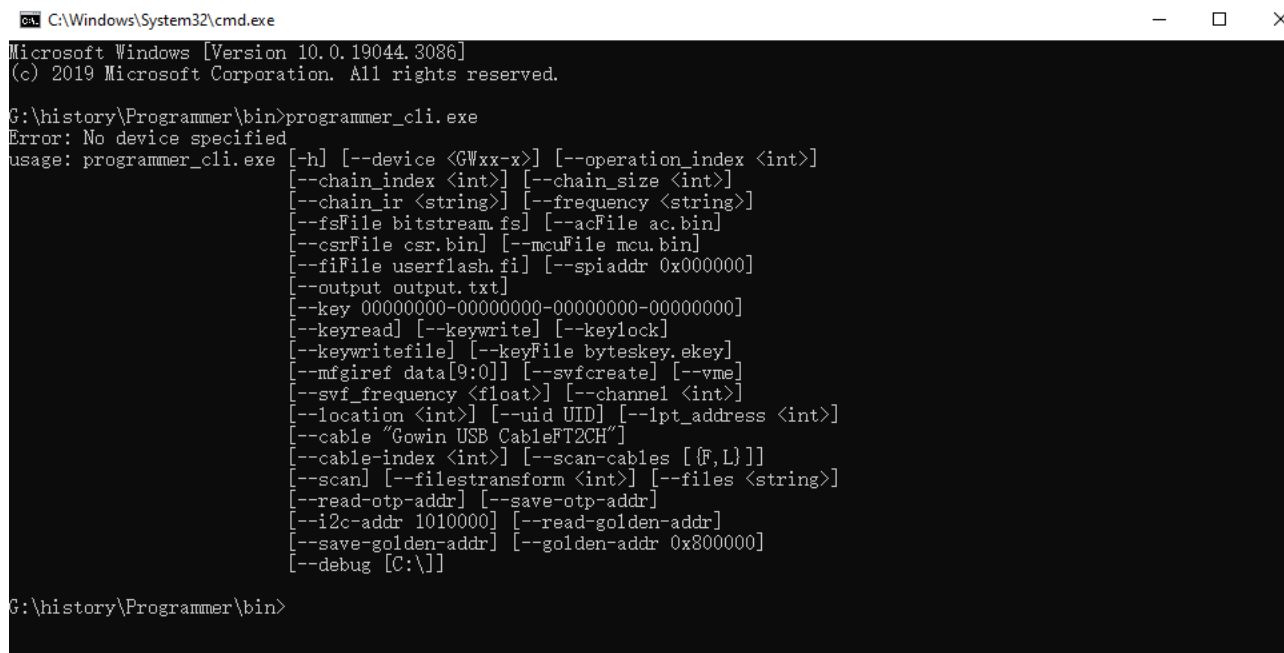
使用 GAO-Bridge 对外部 Flash 进行操作时，不必特意设置 mode pin 管脚。

4 Programmer_cli 编程下载流程

4.1 预览和帮助

在 CMD 中打开 programmer_cli 工具，不使用参数时会提示简要的帮助说明，并提示未指定设备。

图 4-1 CMD 中打开 programmer_cli



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.3086]
(c) 2019 Microsoft Corporation. All rights reserved.

G:\history\Programmer\bin>programmer_cli.exe
Error: No device specified
usage: programmer_cli.exe [-h] [--device <GWxx-x>] [--operation_index <int>]
                        [--chain_index <int>] [--chain_size <int>]
                        [--chain_ir <string>] [--frequency <string>]
                        [--fsFile bitstream.fs] [--acFile ac.bin]
                        [--csrFile csr.bin] [--mcuFile mcu.bin]
                        [--fiFile userflash.fi] [--spiaddr 0x0000000]
                        [--output output.txt]
                        [--key 00000000-00000000-00000000-00000000]
                        [--keyread] [--keywrite] [--keylock]
                        [--keywritefile] [--keyFile byteskey.ekey]
                        [--mfgiref data[9:0]] [--svfcreate] [--vme]
                        [--svf_frequency <float>] [--channel <int>]
                        [--location <int>] [--uid UID] [--lpt_address <int>]
                        [--cable "Gowin USB CableFT2CH"]
                        [--cable-index <int>] [--scan-cables [{F,L}]]
                        [--scan] [--filestransform <int>] [--files <string>]
                        [--read-otp-addr] [--save-otp-addr]
                        [--i2c-addr 1010000] [--read-golden-addr]
                        [--save-golden-addr] [--golden-addr 0x800000]
                        [--debug [C:\]]

G:\history\Programmer\bin>
```

使用参数 `-help` 可以获得更多帮助信息，如下图所示。

图 4-2 帮助信息

```

C:\Windows\System32\cmd.exe
Gowin FPGA Programmer command-line interface. Version V1.9.10.03 (64-bit) build(41017);
Copyright (C) 2014-2024 Gowin Semiconductor Corporation

optional arguments:
  -h, --help            show this help message and exit
  --device <GWxx-x>, -d <GWxx-x>
                        Define a GOWIN FPGA device from:
                        GW1AN-1C GW1AN-9A GW1N-1 GW1N-1P5
                        GW1N-1P5B GW1N-1P5C GW1N-1S GW1N-2
                        GW1N-2B GW1N-2C GW1N-4 GW1N-4B
                        GW1N-4D GW1N-9 GW1N-9C GW1NR-2
                        GW1NR-2B GW1NR-2C GW1NR-4 GW1NR-4B
                        GW1NR-4D GW1NR-9 GW1NR-9C GW1NRF-4B
                        GW1NS-4 GW1NS-4C GW1NSR-4C GW1NSR-4
                        GW1NSR-4C GW1NZ-1 GW1NZ-1C GW1NZ-2B
                        GW1NZ-2C GW2A-18 GW2A-18C GW2A-55
                        GW2A-55C GW2AN-18X GW2AN-55C GW2AN-9X
                        GW2ANR-18C GW2AR-18 GW2AR-18C GW5A-138B
                        GW5A-138C GW5A-25A GW5A-25B GW5ANRT-15A
                        GW5ANT-15A GW5AR-25A GW5AR-25B GW5ART-15A
                        GW5AS-138B GW5AS-138C GW5AS-25A GW5AS-25B
                        GW5AST-138B GW5AST-138C GW5AT-138 GW5AT-138B
                        GW5AT-138C GW5AT-15A GW5AT-60B GW5AT-60ES
                        GW5AT-75B GW5AT-75C
  --operation_index <int>, --run <int>, -r <int>
                        0: Read Device Codes;
                        1: Reprogram;
                        2: SRAM Program;
                        3: SRAM Read;
                        4: SRAM Program and Verify;
                        5: embFlash Erase,Program;
                        6: embFlash Erase,Program,Verify;
                        7: embFlash Erase Only;
                        8: exFlash Erase,Program;
                        9: exFlash Erase,Program,Verify;
                        10: exFlash Bulk Erase;
                        11: exFlash Verify;
                        12: exFlash Erase,Program in bscan;
                        13: exFlash Erase,Program,Verify in bscan;
                        14: exFlash Bulk Erase in bscan;
                        15: exFlash Verify in bscan;
                        16: SRAM Program JTAG 1149;
                        17: SRAM Program,Verify JTAG 1149;
                        18: bsd1 read;
                        19: embFlash 2nd Erase,Program;
                        20: embFlash 2nd Erase,Program,Verify;
                        21: embFlash 2nd Erase Only;
                        22: -R-;
                        23: Connect to JTAG of MCU;
                        24: SRAM Erase;
                        25: Authentication Code Erase,Program,Verify;
                        26: Authentication Code Read;
                        27: Firmware Erase,Program Securely;
                        28: Firmware Erase Only;
                        29: Firmware Erase,Program;
                        30: Firmware Erase,Program,Verify;
                        31: exFlash C Bin Erase,Program;
                        32: exFlash C Bin Erase,Program,Verify;
                        33: -R-;
                        34: MFG Write iRef;
                        35: CSR File Erase,Program,Verify;
                        36: exFlash Erase,Program thru GAO-Bridge;
                        37: exFlash Erase,Program,Verify thru GAO-Bridge;

```

```

35: CSR File Erase,Program,Verify;
36: exFlash Erase,Program thru GAO-Bridge;
37: exFlash Erase,Program,Verify thru GAO-Bridge;
38: exFlash C Bin Erase,Program thru GAO-Bridge;
39: exFlash C Bin Erase,Program,Verify thru GAO-Bridge;
40: DK-GoAI-GW1NSR4C_QN48 v1.1;
41: DK-GoAI-GW1NSR4C_QN48 v2.2;
42: DK-GoAI-GW2AR18_QN88P v1.1;
43: -R-;
44: sFlash Erase,Program;
45: sFlash Erase,Program,Verify;
46: sFlash Verify Only;
47: sFlash Bulk Erase;
48: sFlash Background Erase,Program;
49: sFlash Background Erase,Program,Verify;
50: sFlash Erase,Program,Verify thru GAO-Bridge;
51: exFlash Detect ID;
52: exFlash Bulk Erase 5A;
53: exFlash Erase,Program 5A;
54: exFlash Erase,Program,Verify 5A;
55: exFlash C Bin Erase,Program 5A;
56: exFlash C Bin Erase,Program,Verify 5A;
57: I2C Program SRAM;
58: I2C Program Flash;
59: I2C Erase Flash Only;
60: I2C Erase Flash Only thru I2C-SPI;
61: I2C Erase,Program Flash thru I2C-SPI;
62: EBR Read;
63: sFlash Background Erase,Program,Verify thru GAO-Bridge;
64: sFlash Bulk Erase in bscan;
65: sFlash Erase,Program in bscan;
66: exFlash Verify 5A;
67: exFlash Verify thru GAO-Bridge 5A;
68: exFlash Erase,Program thru GAO-Bridge 5A;
69: exFlash Erase,Program,Verify thru GAO-Bridge 5A;
70: embFlash Background Erase,Program;
71: embFlash Background Erase,Program,Verify;
72: embFlash Background Erase Only ;
73: Read User Code;
74: Read Status Register;
75: Set Flash QE For 9x/18x;
76: Set ExFlash QE For GW5A(T);
77: -R-;
78: -R-;
--chain_index <int>, -i <int>
    Define the device index on the chain. The default is 0.
    It must be used in combination with option: "--chain_size".
--chain_size <int>, -l <int>
    Define the device index on the chain. The minimum length is 1.
    It must be used in combination with option: "--chain_index".
--chain_ir <string>
    Define the IR_LENGTH of every device. example: 8,8,8,8
--frequency <string>, --freq <string>
    default is 2.5MHz. More options:
    2.5MHz; 2MHz; 15MHz; 10MHz; 1.5MHz; 1.1MHz; 0.9MHz; 0.75MHz; 0.5MHz; 0.3MHz; 0.4MHz; 0.1MHz; 0.02MHz;
--fsFile bitstream.fs, --fs bitstream.fs, -f bitstream.fs
    Define the .fs file path.
--acFile ac.bin, --ac ac.bin, -a ac.bin
    Define the Authentication-Code file path.
--csrFile csr.bin
    Define the CSR file path.
--mcuFile mcu.bin, --fw mcu.bin, --mcu mcu.bin, -m mcu.bin
    Define firmware file path of MCU.
--fiFile userflash.fi
    Define Userflash initialization file path.

```

```

--keywrite      Write key to FPGA
--keylock       Lock key setting
--keywritefile  Write key to FPGA through ekey file
--keyFile byteskey.ekey
                  Define the byteskey(.ekey) file path.
--mfgiref data[9:0] Write data[9:5] to tune iref;data[9:0]=itrim[9:5]+freq[4:0]
--svfcreate     Create SVF file only.
--vme          Create VME file after SVF file created.
--svf_frequency <float>
                  Define a frequency for SVF, default is 2.5 (MHz).
--channel <int>  Define download cable channel. Default is 0. Only works for Gowin USB Cable(FT2CH)
--location <int> Define location number of USB Cable.
                  when use location option, programmer will open the corresponding cable.
                  Default works for Gowin USB Cable(FT2CH) .
                  Will ignore --channel option
--uid UID, --unique-id UID
                  Define Unique-ID of USB Cable.
                  when use this option, programmer will open the corresponding cable.
                  Default works for Gowin USB Cable(FT2CH) .
                  Will ignore --location and --channel option
--lpt_address <int> Define GOWIN LPT cable address. Default is 0x0378.
--cable "Gowin USB Cable(FT2CH)"
                  Select a type of USB cable(including quotation marks):
                  "Gowin USB Cable(GWU2X)"
                  "Gowin USB Cable(FT2CH)"
                  "Parallel Port(LPT)"
                  "Diligent USB Device"
                  "USB Debugger A"
                  "Gowin USB Cable(WINUSB)"
                  Default cable is "Gowin USB Cable(FT2CH)"
--cable-index <int> Select a number for USB cable:
                  0: Gowin USB Cable(GWU2X);
                  1: Gowin USB Cable(FT2CH);
                  2: Parallel Port(LPT);
                  3: Diligent USB Device;
                  4: USB Debugger A;
                  5: Gowin USB Cable(WINUSB);
                  Higher priority than --cable, default cable-index is 0
--scan-cables [{F,L}], --show-channel [{F,L}]
                  List GOWIN USB download cables, F means using ft2xx driver, L means using libusb driver
--scan          Scan and list GOWIN FPGA devices
--filestransform <int>
                  1: Convert/Merge .fs to .bin;
                  2: Convert/Merge .bin(binary) to .hex(HEX);
                  3: Convert/Merge .bin(binary) to .h(hpp);
                  4: Convert .bin(binary) to .intelhex(Intel HEX);
                  5: Merge multiple ".fs" files to one ".fs" ;
                  6: Append User Flash Init File(.fi) to a BitStreamFile(.bin);
                  7: Append a MCU FW File(.bin) to a BitStreamFile(.bin);
                  8: Append GW1NS4C M3 Core File(.bin) to a BitStreamFile(.fs);
--files <string> Used with the parameter --filestransform together, multiple files are separated by ",",
                  such as: file1.fs, file2.fs
--read-otp-addr  Read OTP I2C Address
--save-otp-addr  Save as OTP I2C Address
--i2c-addr 1010000 Set the USB Cable address of I2C interface
--read-golden-addr Read golden image Address
--save-golden-addr Save as golden image Address
--golden-addr 0x800000
                  Set the address of the golden image , the range is 0x000000-0xF00000, the default is 0x800000
--debug [C:\]   Output address of the file during debugging
G:\history\Programmer\bin>

```

4.2 扫描 USB Cable 设备

扫描并展示已连接的 USB Cable 信息。

用法: programmer_cli.exe --scan-cables

图 4-3 扫描 USB Cable 设备

```

G:\history\Programmer\bin>programmer_cli.exe --scan-cables
Cable found: Gowin USB Cable(FT2CH)/0/99091/GW20LCRZ (USB location:99091) (SN: GW20LCRZ)
Cost 0.05 second(s)

```

- Gowin USB Cable(FT2CH)为 Cable 类型;
- /0/为 channel 号;
- 99091 为 USB location 信息;

- /GW20LCRZ 为 USB Cable UID
- (USB location:99091) (SN: GW20LCRZ) 为说明信息;

4.3 指定 USB Cable 类型及端口

在扫描或配置 FPGA 时,可以通过参数“--cable-index”指定 USB Cable 类型。当指定--cable-index 为 0 时, Cable 类型为 Gowin USB Cable (GWU2X);当指定--cable-index 为 1 时, Cable 类型为 Gowin USB Cable (FT2CH), 以此类推;默认情况下--cable-index 数值为 0;

当使用多端口 Cable 时,如 FTDI 类型 Cable 会有多个端口可选,可以使用参数“--channel”指定端口,默认--channel 数值为 0;详细帮助信息提示如下图所示。

图 4-4 帮助信息

```
--channel <int>      Define download cable channel. Default is 0. Only works for Gowin USB Cable(FT2CH)
--cable-index <int>  Select a number for USB cable:
                     0: Gowin USB Cable(GWU2X);
                     1: Gowin USB Cable(FT2CH);
                     2: Parallel Port(LPT);
                     3: Digilent USB Device;
                     4: Unverified FT Device;
                     Higher priority than --cable, default cable-index is 0
```

4.4 指定 USB Cable 位置或 UID

当使用多个 USB Cable 时,可以指定 USB 端口位置或 Cable UID 来指定具体的 USB Cable 设备,帮助信息如下图所示。

图 4-5 帮助信息

```
--location <int>      Define location number of USB Cable.
                     when use location option, programmer will open the corresponding cable.
                     Default works for Gowin USB Cable(FT2CH) .
                     Will ignore --channel option
--uid UID, --unique-id UID
                     Define Unique-ID of USB Cable.
                     when use this option, programmer will open the corresponding cable.
                     Default works for Gowin USB Cable(FT2CH) .
                     Will ignore --location and --channel option
```

4.5 扫描 FPGA 设备

使用如下指令扫描设备:

programmer_cli.exe -scan

图 4-6 扫描 FPGA 设备

```
G:\history\Programmer\bin>programmer_cli.exe --scan
Scanning!
Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
Device Info:
  Family: GW1NRF
  Name: GW1N-4D GW1NR-4D GW1N-4B GW1NR-4B GW1NRF-4B (One of them)
  ID: 0x1100381B
  1 device(s) found!
  Cost 0.08 second(s)
```

4.6 指定 Programmer 执行模式

使用参数 “--operation_index” 或 “--run” 或 “-r” 指定执行模式，如配置 SRAM、配置 Flash 等，帮助信息如下图所示。

图 4-7 帮助信息

```
--operation_index <int>, --run <int>, -r <int>
0: Read Device Codes;
1: Reprogram;
2: SRAM Program;
3: SRAM Read;
4: SRAM Program and Verify;
5: embFlash Erase, Program;
6: embFlash Erase, Program, Verify;
7: embFlash Erase Only;
8: exFlash Erase, Program;
9: exFlash Erase, Program, Verify;
10: exFlash Bulk Erase;
11: exFlash Verify;
12: exFlash Erase, Program in bscan;
13: exFlash Erase, Program, Verify in bscan;
14: exFlash Bulk Erase in bscan;
15: exFlash Verify in bscan;
16: SRAM Program JTAG 1149;
17: SRAM Program, Verify JTAG 1149;
18: bsd1 read;
19: embFlash 2nd Erase, Program;
20: embFlash 2nd Erase, Program, Verify;
21: embFlash 2nd Erase Only;
22: -R-;
23: Connect to JTAG of MCU;
24: SRAM Erase;
25: Authentication Code Erase, Program, Verify;
26: Authentication Code Read;
27: Firmware Erase, Program Securely;
28: Firmware Erase Only;
29: Firmware Erase, Program;
30: Firmware Erase, Program, Verify;
31: exFlash C Bin Erase, Program;
32: exFlash C Bin Erase, Program, Verify;
33: -R-;
34: MFG Write iRef;
```

配置 FPGA 时使用参数 “--device” 指定 FPGA Device 类型，帮助信息如下图所示。

图 4-8 帮助信息

```
--device <GWxx-x>, -d <GWxx-x>
Define a GOWIN FPGA device from:
GW1N-1 GW1N-1P5 GW1N-1P5B GW1N-1P5C
GW1N-1S GW1N-2 GW1N-2B GW1N-2C
GW1N-4 GW1N-4B GW1N-4D GW1N-9
GW1N-9C GW1NR-1 GW1NR-2 GW1NR-2B
GW1NR-2C GW1NR-4 GW1NR-4B GW1NR-4D
GW1NR-9 GW1NR-9C GW1NRF-4B GW1NS-2
GW1NS-2C GW1NS-4 GW1NS-4C GW1NSE-2C
GW1NSER-4C GW1NSR-2 GW1NSR-2C GW1NSR-4
GW1NSR-4C GW1NZ-1 GW1NZ-1C GW2A-18
GW2A-18C GW2A-55 GW2A-55C GW2AN-18X
GW2AN-55C GW2AN-9X GW2ANR-18C GW2AR-18
GW2AR-18C
```

配置 SRAM 或 Flash 一般使用如下命令格式：

```
programmer_cli.exe --device <GWxx-x> --run <int> --fsFile
<bitstream.fs> --cable-index <int> --location <int> --uid <UID>
--chain_index <int> --chain_size <int> --chain_ir <string> --frequency
<string>
```

- --frequency 指定 JTAG 频率，目前仅适用于 FTDI 类型 Cable；U2X 类型 Cable 频率固定为 1.33MHz；
- --chain_index <int> --chain_size <int> --chain_ir <string> 三者配合使用，用于指定目标设备在菊花链中的位置；
 - --chain_index <int> 用于指定目标设备在菊花链中的位置，例如：
--chain_index n 表示选中第几个器件，n 指的是第 n+1 个器件
 - --chain_size <int> 用于表示在菊花链中设备的总数，例如：
--chain_size n 表示菊花链中设备的总数为 n
 - --chain_ir <string> 用于指定该菊花链中 jtag 状态机中 ir 的长度，例如：
--chain_ir 8,8，表示该菊花链中两个设备的 ir 长度都是 8（如没有特殊情况，默认是 8）
- --location 用于指定目标设备所在 USB 端口，优先级高于 UID；
- --uid 用于指定目标设备所用 USB Cable；
- --cable-index <int>指定 USB Cable 类型；
- --fsFile <bitstream.fs>指定数据流文件路径；
- --run <int>同--operation_index，指定执行模式；
- --device <GWxx-x>指定目标设备名，区分大小写。

4.7 配置 SRAM

指定一个数据流文件、对应的设备和 SRAM 配置模式，可配置 FPGA SRAM，例如：

配置 SRAM Program，“SRAM Program”在--operation_index 参数中对应的数值为 2，所以此操作命令如下：

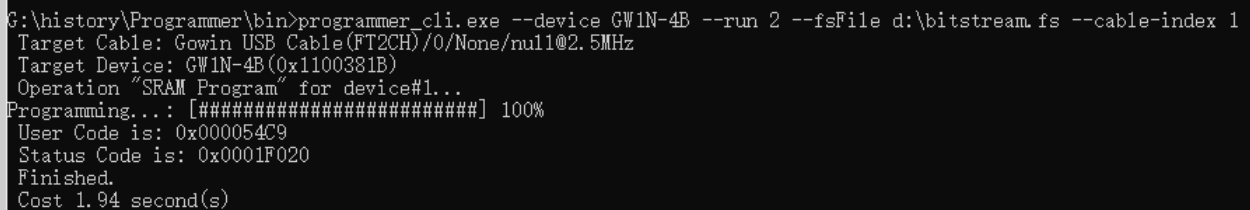
```
programmer_cli.exe --device <GWxx-x> --run <int> --fsFile
<bitstream.fs> --cable-index <int> --location <int>
```

其中--cable-index、--location 为指定时有默认数值，可以省略；

示例如下：

```
programmer_cli.exe --device GW1N-4B --run 2 --fsFile d:\bitstream.fs
--cable-index 1
```

图 4-9 示例图



```
G:\history\Programmer\bin>programmer_cli.exe --device GW1N-4B --run 2 --fsFile d:\bitstream.fs --cable-index 1
Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
Target Device: GW1N-4B(0x1100381B)
Operation "SRAM Program" for device#1...
Programming...: [#####] 100%
User Code is: 0x000054C9
Status Code is: 0x0001F020
Finished.
Cost 1.94 second(s)
```

其中 参数含义如下：

- --device GW1N-4B 指定目标 FPGA 设备，扫描时会打印器件名称；
- --fsFile d:\bitstream.fs 用于指定数据流文件 d:\bitstream.fs；
- --cable-index 1 用于指定 USB Cable 为“Gowin USB Cable(FT2CH)”。

4.8 配置小蜜蜂家族嵌入式 Flash

4.8.1 仅配置 Flash

烧录 Flash 的--operation_index 标号如下：

- 5: embFlash Erase,Program;
- 6: embFlash Erase,Program,Verify;
- 7: embFlash Erase Only;

示例如下：

```
programmer_cli --run 5 --fsFile d:\bitstream.fs --device GW1N-4B
--cable-index 1
```

图 4-10 示例图

```
G:\history\Programmer\bin>programmer_cli.exe --device GW1N-4B --run 5 --fsFile d:\bitstream.fs --cable-index 1
Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
Target Device: GW1N-4B(0x1100381B)
Operation "embFlash Erase,Program" for device#1...
Erasing embFlash ...: [#####] 100%
Programing...: [#####] 100%
Status Code is: 0x0001F020
User Code: 0x000054C9
Program Finished!
Finished.
Cost 3.18 second(s)
```

4.8.2 配置 Flash 和 UserFlash 初始化文件

通过参数“--fiFile userflash.fi”指定 UserFlash 初始化文件，在烧录 Flash 时同时配置 UserFlash，例如：

```
programmer_cli --run 5 --fsFile d:\bitstream.fs --fiFile d:userflash.fi
--device GW1N-4B --cable-index 1
```

4.9 配置外部 SPI Flash

烧录外部 SPI Flash 的--operation_index 标号如下，推荐使用“thru GAO-Bridge”的类型：

- 8: exFlash Erase,Program;
- 9: exFlash Erase,Program,Verify;
- 10: exFlash Bulk Erase;
- 11: exFlash Verify;
- 12: exFlash Erase,Program in bscan;
- 13: exFlash Erase,Program,Verify in bscan;
- 14: exFlash Bulk Erase in bscan;
- 15: exFlash Verify in bscan;
- 36: exFlash Erase,Program thru GAO-Bridge; （推荐）
- 37: exFlash Erase,Program,Verify thru GAO-Bridge;
- 38: exFlash C Bin Erase,Program thru GAO-Bridge;
- 39: exFlash C Bin Erase,Program,Verify thru GAO-Bridge;

示例如下：

```
programmer_cli --run 36 --fsFile d:\bitstream.fs --device GW1N-9
--cable-index 1
```

注！

更多请查看 programmer_cli -help。

