Gowin Programmer

User Guide

SUG502-2.5.2E, 09/22/2026

Copyright © 2026 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN is the trademark of Guangdong Gowin Semiconductor Corporation and are registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Date	Version	Description
04/06/2017	1.0E	Initial version published.
08/06/2017	1.1E	Device programming operation modified.
10/28/2019	1.2E	● Slave SPI Mode added. ● SVF File Creation added. ● User Flash Initialization added.
02/17/2020	1.3E	The description of installing and starting Programmer added.
06/01/2022	1.4E	● Section 2.1 Introduction to Programmer Tool Chain added. ● Chapter 4 Programmer_cli Programming Download Flow added.
06/08/2023	1.5E	● The description of cable privilege configuration in Linux system added. ● The description of Gowin USB Cable (GWU2X) configuration added. ● SRAM Program JTAG 1149 removed from Table 3-1 Device Operations Description. ● Software screenshots updated.
05/09/2024	1.6E	● Descriptions of Section 3.6 Device Security updated. ● Descriptions of the SRAM and Flash command format improved. ● Descriptions of I2C slave address operation added.
06/28/2024	1.7E	Status code analyzer added.
10/25/2024	1.8E	● Descriptions of MSPI 2nd Boot address operation added. ● Some interface screenshots updated.
12/31/2024	1.9E	Sections 3.13 Adjustment of JTAG State Machine Shift-IR Width and 3.14 Remote Configuration and Programming added.
02/28/2025	2.0E	● Pin connection notes added in the Section 3.1 Cable Setting. ● GAO-Bridge note added.
03/28/2025	2.1E	● Descriptions of Gowin USB Cable(WINUSB) and USB Debugging A cable added. ● Section 3.4.7 MCU Mode Configuration added. ● Section 3.4.8 SecureFPGA Mode Configuration added. ● Section 3.4.10 Debugging Mode Configuration added. ● Section 3.4.11 GoConfigIP Mode Configuration added. ● Section 3.4.12 Background Programming Mode Configuration added. ● Section 4.9.3 External Flash Programming of Arora V Devices added.
04/30/2025	2.2E	● Section 3.15 File Conversion Function added. ● Section 3.16 Bitstream File Merging Function added. ● Section 3.17 Log Viewer Function added. ● Section 3.18 Baud Rate Modification Function added.
06/27/2025	2.2.1E	Descriptions of pin note in the Section 3.1 Cable Setting updated.
08/29/2025	2.3E	● "embFlash Background Erase, Program without Reprogram", "embFlash Background Erase, Program, Verify without Reprogram", "exFlash Erase Sectors About File", "exFlash Erase Sectors About File thru GAO-Bridge", "exFlash Erase Sectors About File Arora V", "exFlash Erase Sectors About File thru GAO-Bridge Arora V", "sFlash Erase Sectors About File", and "sFlash Erase Sectors About File thru GAO-Bridge" operations added in Table 3-1 Device Operations Description.

Date	Version	Description
		● Section 4.11 MSPI 2nd Boot Address Operation added. ● Section 4.12 Remote Configuration and Programming added. ● Section 4.13 Device Security added. ● Section 4.14 Create an SVF File added. ● Section 4.15 File Conversion Function added. ● Section 4.16 I2C Slave Address Operation added.
12/31/2025	2.4E	● Updated descriptions of Internal Flash Mode in Table 3-1 Device Operations Description. ● Added an example of the MSPI 2nd Boot address write operation. ● Section 3.4.4 Sipped Flash Configuration updated to describe the configuration procedures categorized by device characteristics. ● Section 4.9.2 Sipped Flash Programming updated to describe the configuration procedures categorized by device characteristics.
03/09/2026	2.5E	● Updated the note descriptions in the 3.10 MSPI 2nd Boot Address Operation section. ● "exFlash Verify thru JTAG IEEE 1149", "exFlash Bulk Erase thru JTAG IEEE 1149", "exFlash Erase, Program thru JTAG IEEE 1149", "exFlash Erase, Program, Verify thru JTAG IEEE 1149", "exFlash C Bin Erase, Program thru JTAG IEEE 1149", "exFlash C Bin Erase, Program, Verify thru JTAG IEEE 1149" operations added in Table 3-1 Device Operations Description. ● Added section 3.19 Interface Theme Switching.
06/17/2026	2.5.1E	● Added GW1AN-3B/GW1AN-5B devices. ● Added descriptions of FPGA programming restrictions in JTAG pin multiplexing scenarios.
09/22/2026	2.5.2E	Added a description specifying that the directory containing programmer.exe is used as the base directory for relative paths.

Contents

Contents i
List of Figures iv
List of Tables vi
1 About This Guide 1
1.1 Purpose 1
1.2 Related Documents 1
1.3 Terminology and Abbreviations 2
1.4 Support and Feedback 2
2 Introduction 3
2.1 Introduction to Programmer Tool Chain 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 Gowin_USB_Cable_Installer.sh and Makefile 3
2.2 Install and Launch Programmer Tool Chain 3
2.2.1 The First Method to Install 3
2.2.2 The Second Method to Install 4
2.2.3 Launch Programmer Tool Chain 5
2.3 Software Interface 6
2.4 Software Version 7
2.5 Cable Privilege Configuration in Linux System 8
2.5.1 Makefile 8
2.5.2 Gowin_USB_Cable_Installer.sh 9
3 Programming Download Flow 10
3.1 Cable Setting 10
3.2 Scan Daisy Chain 13
3.3 Daisy Chain Configuration 14
3.3.1 Add a Device 14
3.3.2 Delete a Device 14
3.3.3 Modify a Device Position in Chain 14

3.4 Device Configuration 14
3.4.1 SRAM Configuration 21
3.4.2 Embedded Flash Configuration of LittleBee Family FPGA Products 21
3.4.3 External Flash Configuration 22
3.4.4 Sipped Flash Configuration 22
3.4.5 External Flash of Arora V Devices Configuration 23
3.4.6 Slave SPI Mode 24
3.4.7 MCU Mode Configuration 24
3.4.8 SecureFPGA Mode Configuration 24
3.4.9 I2C Mode Configuration 25
3.4.10 Debugging Mode Configuration 25
3.4.11 GoConfigIP Mode Configuration 25
3.4.12 Background Programming Mode Configuration 26
3.5 Edit Pin State 26
3.6 Device Security 27
3.7 Programming Download 28
3.8 SVF File Creation 28
3.9 I2C Slave Address Operation 29
3.10 MSPI 2nd Boot Address Operation 29
3.11 Status Code Analyzer 30
3.12 User Flash Initialization and Data Retention 31
3.13 Adjustment of JTAG State Machine Shift-IR Width 34
3.14 Remote Configuration and Programming 34
3.15 File Conversion Function 38
3.16 Bitstream File Merging Function 39
3.17 Log Viewer Function 40
3.18 Baud Rate Modification Function 41
3.19 Interface Theme Switching 41
4 Programmer_cli Programming Download Flow 42
4.1 Preview and Help 42
4.2 Scan USB Cable Device 45
4.3 Specify USB Cable Type and Port 46
4.4 Specify USB Cable Location or UID 46
4.5 Scan FPGA Device 46
4.6 Specify Programmer Operation Mode 47
4.7 Configure SRAM 49
4.8 Configure Embedded Flash of LittleBee Family 50
4.8.1 Configure Flash Only 50
4.8.2 Configure Flash and UserFlash Initialization File 50

4.9 Configure External SPI Flash 51
4.9.1 External Flash Programming of LittleBee Family, GW2A(R) and GW3A Devices 51
4.9.2 Sipped Flash Programming 52
4.9.3 External Flash Programming of Arora V Devices 54
4.10 Configure I2C 55
4.11 MSPI 2nd Boot Address Operation 55
4.12 Remote Configuration and Programming 56
4.13 Device Security 56
4.14 Create an SVF File 57
4.15 File Conversion Function 57
4.16 I2C Slave Address Operation 57

List of Figures

Figure 2-1 Install Programmer 4
Figure 2-2 Install Programmer Driver 4
Figure 2-3 Launch programmer.exe 5
Figure 2-4 Launch programmer_cli.exe 6
Figure 2-5 Launch Programmer Using Shortcut Key 6
Figure 2-6 Programmer Interface 7
Figure 2-7 Version Number 8
Figure 2-8 File List 8
Figure 2-9 Makefile Installation for a Regular User 8
Figure 2-10 Makefile Installation for a Root User 9
Figure 2-11 Script Installation 9
Figure 3-1 Gowin USB Cable (FT2CH) 11
Figure 3-2 LPT 11
Figure 3-3 Gowin USB Cable (GWU2X) 12
Figure 3-4 Gowin USB Cable (WINUSB) 12
Figure 3-5 USB Debugger A 13
Figure 3-6 Device Table 13
Figure 3-7 Device Configuration Interface 14
Figure 3-8 I/O State Editor 27
Figure 3-9 Security Configuration 27
Figure 3-10 Create SVF File 29
Figure 3-11 I2C Slave Address Operation 29
Figure 3-12 MSPI 2nd Boot Address Operation 30
Figure 3-13 Status Code Analyzer 31
Figure 3-14 User Flash Initialization 32
Figure 3-15 Retained User Flash Data 33
Figure 3-16 Adjustment of JTAG State Machine Shift-IR Width 34
Figure 3-17 jtagserver Location 34
Figure 3-18 jtagserver in Running State 34
Figure 3-19 jtagserver in Running State 34
Figure 3-20 Add GVC Server Option 35
Figure 3-21 GVC 35

Figure 3-22 Connection Succeed 35
Figure 3-23 Connection Failed 35
Figure 3-24 Remote Operation Output 36
Figure 3-25 Exit Remote Configuration Mode 36
Figure 3-26 File Conversion Interface 39
Figure 3-27 Bitstream File Merging Interface 39
Figure 3-28 Log Viewer Interface 40
Figure 3-29 Baud Rate Modification Interface 41
Figure 3-30 Theme Switching 41
Figure 4-1 Open programmer_cli in CMD 42
Figure 4-2 Help Information 43
Figure 4-3 Scan USB Cable Device 45
Figure 4-4 Help Information 46
Figure 4-5 Help Information 46
Figure 4-6 Scan FPGA Device 47
Figure 4-7 Help Information 47
Figure 4-8 Help Information 48
Figure 4-9 SRAM Configuration Example 49
Figure 4-10 Embedded Flash Programming of LittleBee Family Example 50
Figure 4-11 External Flash Programming of LittleBee Family Example 51
Figure 4-12 Example for Sipped Flash Programming of GW2AN-9X Devices 52
Figure 4-13 Example for Sipped Flash Programming of GW2AN-18C Devices 53
Figure 4-14 External Flash Programming of Arora V Devices Example 54
Figure 4-15 I2C Configuration Example 55
Figure 4-16 MSPI 2nd Boot Address Operation Example 55
Figure 4-17 MSPI 2nd Boot Address Operation Example 56
Figure 4-18 Remote Configuration and Programming Example 56
Figure 4-19 Device Security Example 57
Figure 4-20 Creating an SVF File Example 57
Figure 4-21 File Conversion Function Example 57
Figure 4-22 I2C Slave Address Operation Example 58

List of Tables

Table 1-1 Terminology and Abbreviations 2
Table 3-1 Device Operations Description 15
Table 3-2 User Flash Reference 33
Table 3-3 Reference Table for Device and Remote Programming Function 36
Table 4-1 SRAM Function Configuration Index 49
Table 4-2 Function Configuration Index for LittleBee Family 50
Table 4-3 Function Configuration Index for LittleBee Family and GW2A(R) Devices 51
Table 4-4 Device Function Configuration Index 52
Table 4-5 Device Function Configuration Index 53
Table 4-6 Function Configuration Index for Arora V Devices 54
Table 4-7 Function Configuration Index for I2C 55

1 About This Guide

1.1 Purpose

This guide describes how to use Gowin Programmer. The software screenshots and the supported products listed in this guide are based on 1.9.11.01. As the software is subject to change without notice, some information may not remain relevant and may need to be adjusted according to the software that is in use.

1.2 Related Documents

The latest user guides are available on GOWINSEMI Website. You can find the related documents at www.gowinsemi.com:

- SUG100, Gowin Software User Guide
- TN653, Gowin FPGA Products JTAG Programming and Configuration Manual
- UG290, Gowin FPGA Products Programming and Configuration Manual
- UG704, Arora V 138K FPGA Product Programming and Configuration Guide
- UG714, Arora V 25K FPGA Products Programming and Configuration Guide
- UG718, Arora V 60K FPGA Products Programming and Configuration Guide
- UG720, Arora V 15K FPGA Products Programming and Configuration Guide

1.3 Terminology and Abbreviations

Table 1-1 shows the abbreviations and terminology used in this guide.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Meaning
FPGA	Field Programmable Gate Array
SRAM	Static Random Access Memory
I/O	Input/Output
BSDL	Boundary Scan Description Language
GAO	Gowin Analyzer Oscilloscope
GAO-Bridge	The Bridge for JTAG to SPI

1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly by the following ways.

Website: www.gowinsemi.com

E-mail:support@gowinsemi.com

2 Introduction

2.1 Introduction to Programmer Tool Chain

2.1.1 programmer.exe

The graphical tool programmer.exe is Gowin FPGA downloader, which provides a graphical operation interface and provides bitstream configuration or download functions.

2.1.2 programmer_cli.exe

programmer_cli.exe is the command line version of Programmer.

2.1.3 JTAGLoading.exe

Gowin SVF command line software tool, currently only Windows version is available; and the current version only supports Gowin USB Cable Version 3.0 and 4.0.

2.1.4 jtagserver.exe

jtagserver.exe belongs to GAO tool chain.

2.1.5 Gowin_USB_Cable_Installer.sh and Makefile

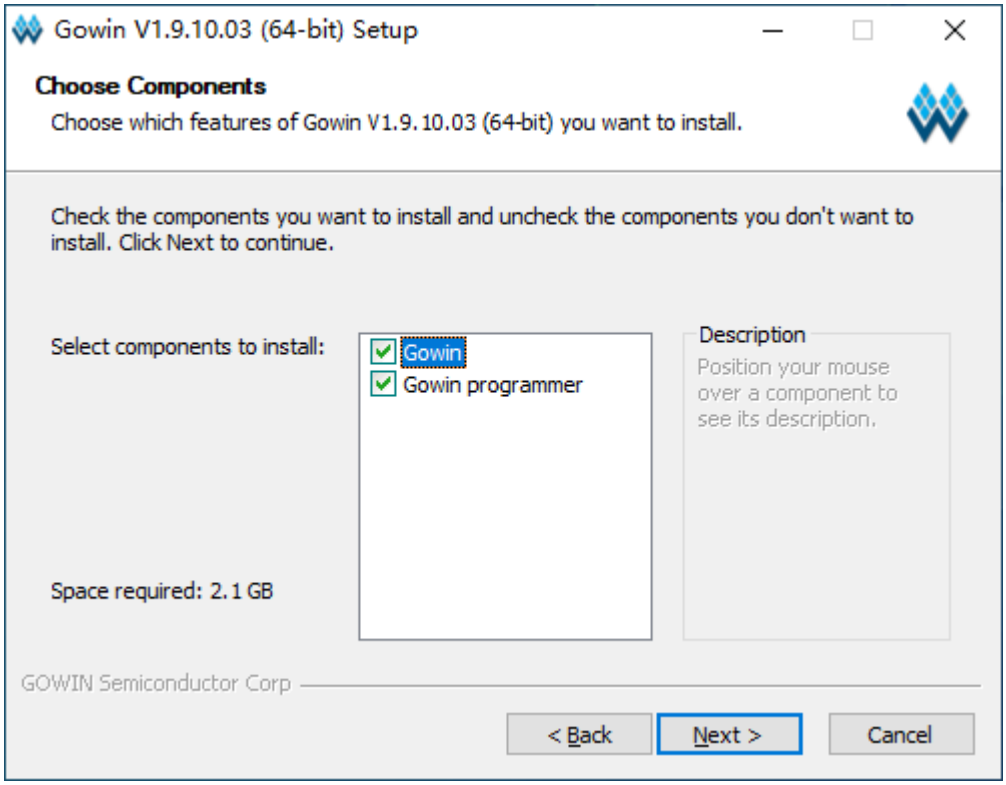
Used to modify the privileges of a cable in Linux system

2.2 Install and Launch Programmer Tool Chain

2.2.1 The First Method to Install

When installing Gowin Software using the default installation method, the component Gowin Programmer will be installed, as shown in Figure 2-1. For details on the software installation and application, see *SUG100, Gowin Software User Guide*.

Figure 2-1 Install Programmer



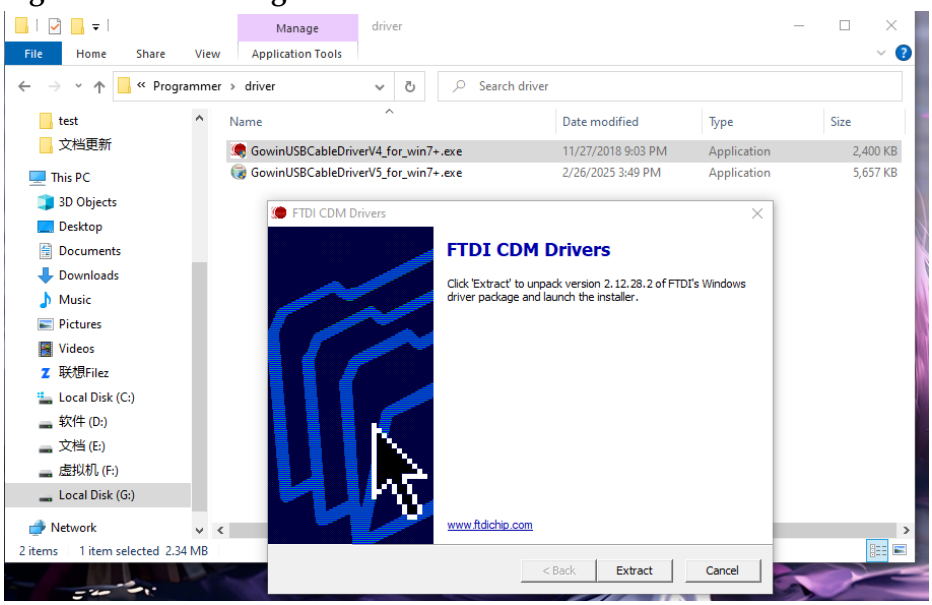
2.2.2 The Second Method to Install

Download Gowin Programmer installation package at official website: https://www.gowinsemi.com/en/support/download_eda/. After unzipping the installation package, open directory of programmer\driver. Choose corresponding driver according to your own computer system, as shown in Figure 2-2.

Note!

When installing the GWU2X driver on a Windows XP system, make sure to first insert the corresponding USB device, otherwise the installation cannot proceed normally.

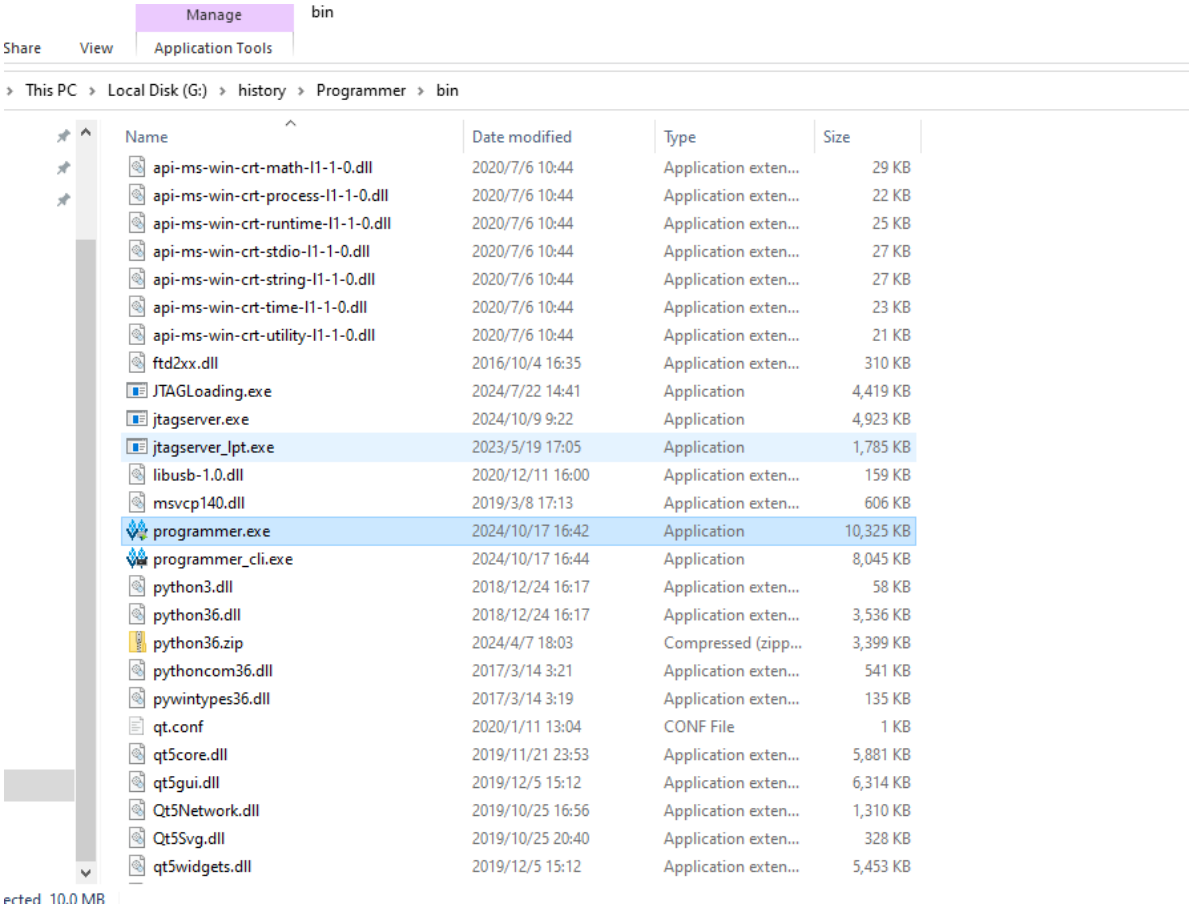
Figure 2-2 Install Programmer Driver



2.2.3 Launch the Programmer Tool Chain

- After Programmer is installed, the executable file is located in the \x.x\Programmer\bin\ directory under the installation directory. Double-click programmer.exe to launch the software, as shown in Figure 2-3. All relative paths used by the software are relative to the directory containing programmer.exe.

Figure 2-3 Launch programmer.exe



- For command line software, please open it in CMD; for example, open programmer_cli.exe.

Figure 2-4 Launch programmer_cli.exe

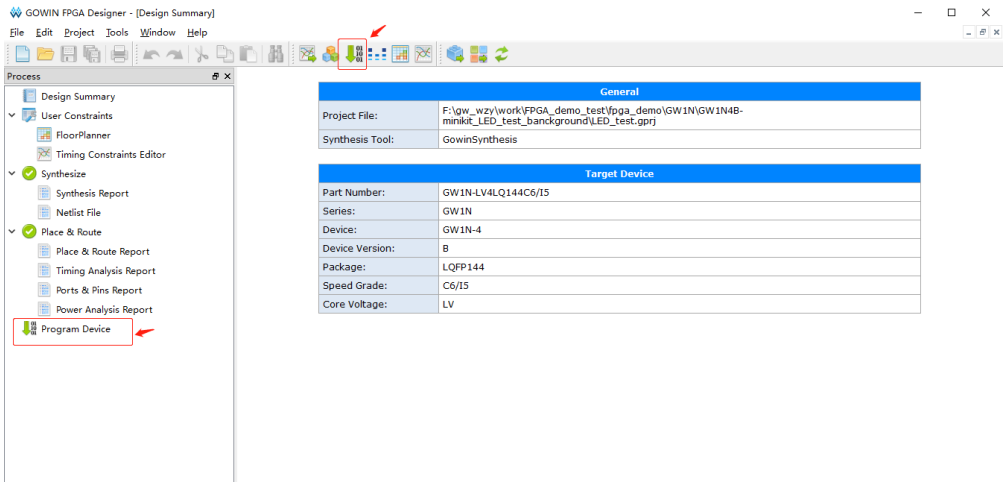
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.3086]
(c) 2019 Microsoft Corporation. All rights reserved.

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 0x000000]
                          [--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] [--1pt_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>
```

- This software can be started by the shortcut key in Gowin Software, as shown in Figure 2-5.

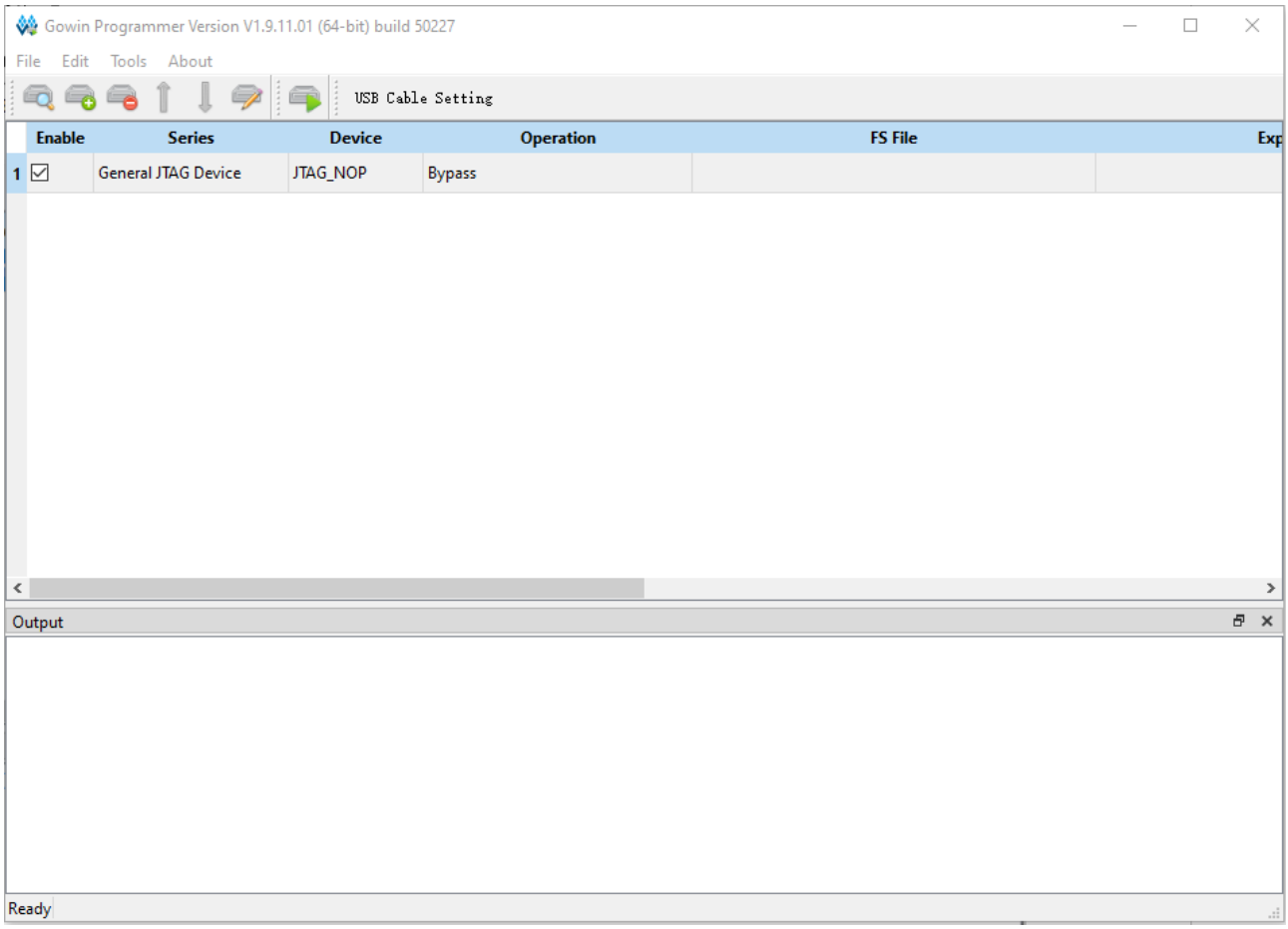
Figure 2-5 Launch Programmer Using Shortcut Key



2.3 Software Interface

Gowin Programmer interface includes menu bar, tool bar, device table, output area, as shown in Figure 2-6.

Figure 2-6 Programmer Interface



In the device table, all the devices that will be programmed in daisy chain are displayed through automatically scanning or manual configuration. Each row of the table represents a device, which can be programmed or not by selecting Enable column.

The device table includes Enable, Series, Device, Operation, FS File, Export File, User Code, and IDCODE options. Enable, Family, Device, Operation, and FS files are editable and can be edited with clicking. Double-click to open Device Configuration dialog box to configure the other options. See 3.4 Device Configuration for the details.

Output area includes Output, Error, Warning, and Info, which respectively displays all information, error information, warning information, and instructions information.

Note!

If "Enable" is not checked, the Programmer will consider the device as not being in the chain.

2.4 Software Version

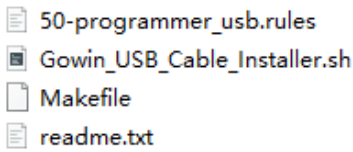
Gowin Programmer and Gowin Software have separate software version numbers, which can be viewed by clicking "About" in software interface, as shown in Figure 2-7.

Figure 2-7 Version Number

2.5 Cable Privilege Configuration in Linux System

As shown in Figure 2-8, Makefile is a text file and Gowin_USB_Cable_Installer.sh is a script file, and both of them can be used to modify the cable privileges.

Figure 2-8 File List



2.5.1 Makefile

Open a terminal, enter "sudo make" command or switch to root privileges; then enter make command, the display shows "File 50-programmer_usb.rules has been copied to /etc/udev/rules/d/" to indicate successful installation (some centos6 systems need to restart to complete the setup), as shown in Figure 2-9 and Figure 2-10.

Figure 2-9 Makefile Installation for a Regular User

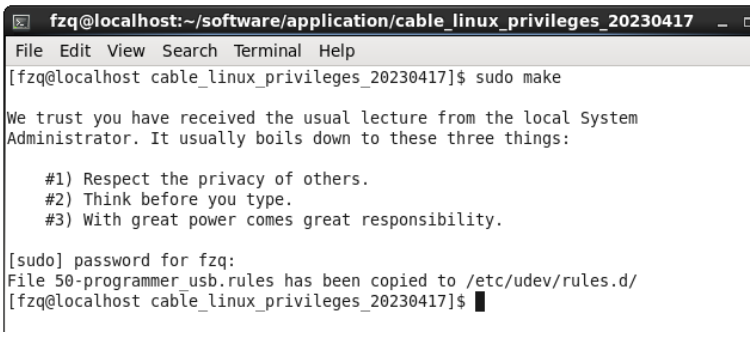


Figure 2-10 Makefile Installation for a Root User

```
fzq@localhost:/home/fzq/Desktop/cable_linux_privileges_2023041
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

Open the folder where the programmer is located and check if all the above 4 files exist in Figure 2-8. After that, open a terminal, switch to root permission, and give the executable permissions to Gowin_USB_Cable_Installer.sh; then run Gowin_USB_Cable_Installer.sh, and it will show "complete" to indicate successful installation, as shown in Figure 2-11.

Figure 2-11 Script Installation

```
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_Insta
ller.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 Programming Download Flow

Programming download is the process of downloading the bitstream files to SRAM, embedded Flash or external Flash of FPGA through download cable, and the programming download process is as follows: Start > Setting download cable > Scanning device > Device programming configuration > Downloading.

1. Download Cable Setting (optional): Select the download cable type, port, and frequency for the programming download.

Note!

The first available port will be selected by default, and the default frequency is 2 MHz.

2. Daisy chain configuration and programming: Configure daisy chain in the device table to match the actual physical connection of daisy chain, and select the programming operation and required bitstream file for each device with the top level being near the Programmer.
3. Programming download: Download the daisy chain that has been configured, and the final result will be displayed in the output area.

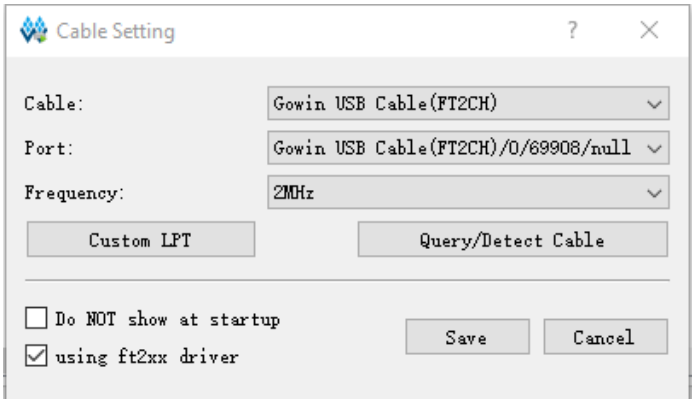
3.1 Cable Setting

The Cable Setting allows users to select the available download cable type, port, and frequency for the programming download. Select "Edit > Setting > Cable Setting" in the menu bar to open "Cable Setting". Five types of cables are supported currently: Gowin USB Cable (FTDI), Gowin USB Cable (GWU2X), LPT, Gowin USB Cable (WINUSB), and USB Debugger A.

1. Gowin USB Cable (FT2CH) is as shown in Figure 3-1.
   - Cable: Download cable; you can select Gowin USB Cable (FT2CH)
   - Port: Available port for the download cable; the first available port will be selected by default.
   - Frequency: JTAG frequency, options include 2 MHz, 2.5 MHz, 15 MHz, 10 MHz, etc., with a default of 2 MHz.
   - Do not show at startup: Whether to display this page when the software is launched.

- Using ft2xx driver: Check to use the ftd2xx.xx to operate Gowin USB Cable (FTDI).

Figure 3-1 Gowin USB Cable (FT2CH)



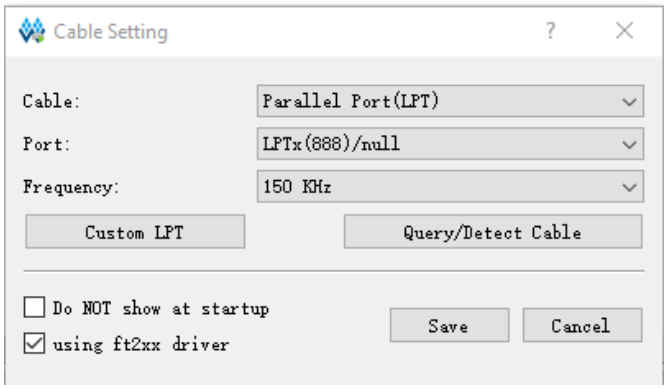
Note!

When using the I2C protocol, the connection for the TDI and TDO pins varies depending on the version of Programmer. For both versions mentioned below, the TCK pin must be connected to the SCL pin.

- For PL-USB-Cable V4.0, the TDI and TDO pins need to be externally connected to the SDA pin.
- For PL-USB-Cable V4.1, directly connect the TDI pin to the SDA pin, while ensure that pin 27 of the FTDI chip is pulled down internally.
- PL-USB-Cable V4.1 supports the UART protocol, with the following connection requirements:
  - The TCK pin is used as TX.
  - The TDI pin is used as RX.
  - Internally pull down the pin 17 of the FTDI chip.

2. LPT is as shown in Figure 3-2.
   - Cable: Download cable; you can select Parallel Port (LPT).
   - Port: The available port for the download cable, selected according to the PCI property of the computer.
   - Frequency: 150KHz

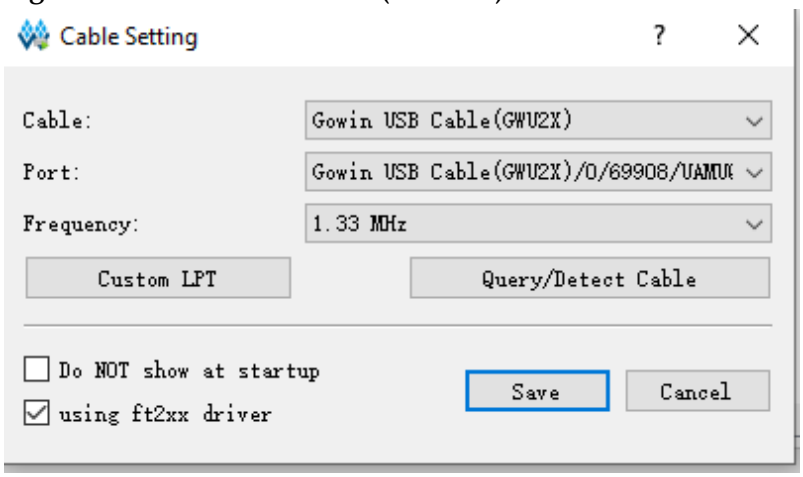
Figure 3-2 LPT



3. Gowin USB Cable (GWU2X) is as shown in Figure 3-3.
   - Cable: Download cable; you can select Gowin USB Cable(GWU2X).

- Port: Available port for the download cable; the first available port will be selected by default.
- Frequency: 1.33MHz by default

Figure 3-3 Gowin USB Cable (GWU2X)



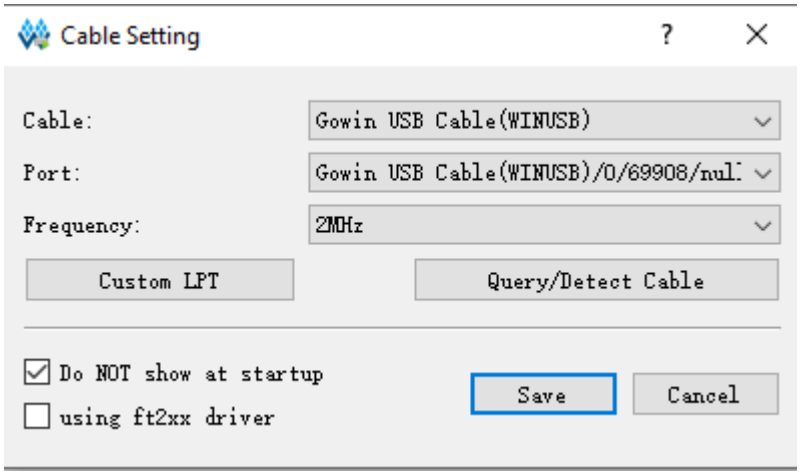
Note!

When using the I2C protocol, the connection for the TDI and TDO pins varies depending on the version of Programmer. For both versions mentioned below, the TCK pin must be connected to the SCL pin.

- For PL-U2X-Cable V5.0, the TDI and TDO pins need to be externally connected to the SDA pin.
- For PL-U2X-Cable V5.1, directly connect the TDI pin to the SDA pin, while ensure that pin P14 of the U2X chip is pulled down.

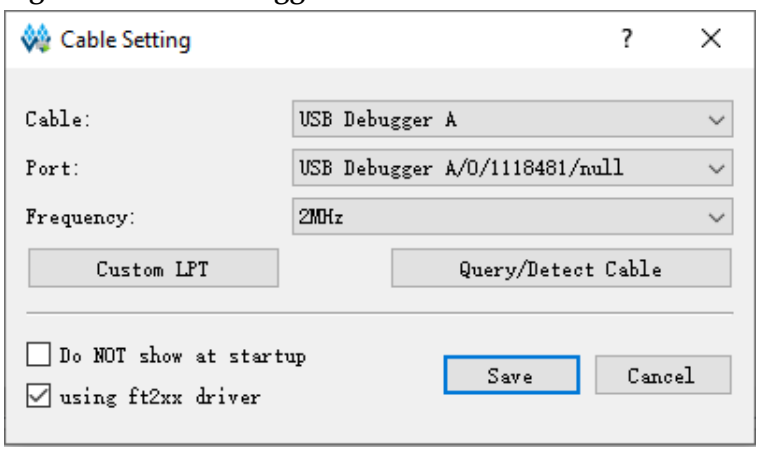
4. Gowin USB Cable (WINUSB) is as shown in Figure 3-4.
   - Cable: Download cable; you can select Gowin USB Cable (WINUSB).
   - Port: Available port for the download cable; the first available port will be selected by default.
   - Frequency: JTAG frequency, options include 2MHz, 2.5MHz, 15MHz, 10MHz, etc. The default is 2MHz.
   - Using ft2xx driver: If unchecked, libusbxx.xx is used to operate Gowin USB Cable (WINUSB).

Figure 3-4 Gowin USB Cable (WINUSB)



5. USB Debugger A is as shown in Figure 3-5.
   - Cable: Download cable; you can select USB Debugger A.
   - Port: Available port for the download cable; the first available port will be selected by default.
   - Frequency: The default value is 2MHz.

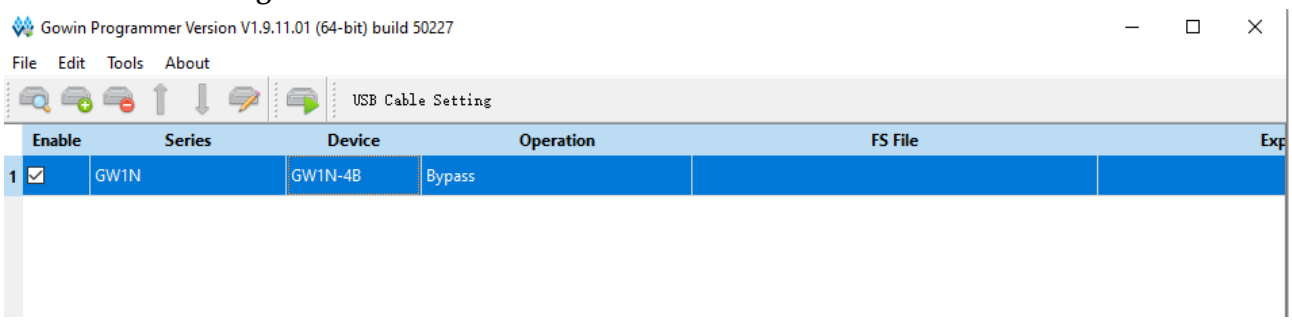
Figure 3-5 USB Debugger A



3.2 Scan Daisy Chain

Programmer automatically scans the daisy chain connected to the computer. Click " " to scan daisy chain connected to the computer. After scanning, all devices are shown in device table in the order of chain, as shown in Figure 3-6.

Figure 3-6 Device Table



Note!

Some devices have the same ID (such as GW2A-18/GW2AR-18), which requires users to manually specify the corresponding device after scanning.

Programmer supports the manual configuration of daisy chain. It includes the operations of adding device, removing device, and modifying the position of the device in the chain.

3.3 Daisy Chain Configuration

3.3.1 Add a Device

1. Select "Edit > Add Device" or click " " in menu to add a new device.
2. Click "Series" to select the device series from the drop-down menu.
3. Click "Device" to select a part number from the drop-down menu.

Note!

When selected, the new device is added to the selected location or the end of the daisy chain.

3.3.2 Delete a Device

Select the device row and delete a device by clicking "Edit > Delete Device" or the " ".

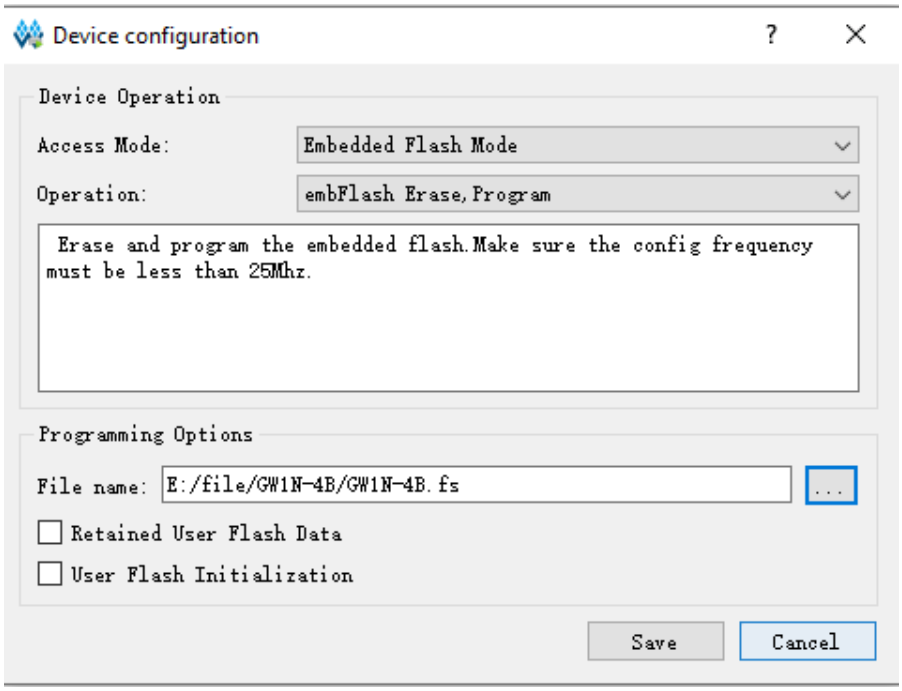
3.3.3 Modify a Device Position in Chain

Select the device row and modify the device position in the chain by clicking "Edit > Move Device Up (or Down) " or the " " or " ".

3.4 Device Configuration

Select the device row and open the "Device configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation", as shown in Figure 3-7.

Figure 3-7 Device Configuration Interface



- Access Mode: Select programming mode.

- Operation: Select programming operation, see Table 3-1 for details.
- File name: Select programming data file.
- Device: When the programming mode is selected as External Flash Mode, select External Flash.
- Start Address: When the programming mode is selected as External Flash Mode, select initial address of External Flash.

Table 3-1 Device Operations Description

Access Mode	Operation	Description
SRAM Mode	Bypass	Bypass
	Read Device Code	Read device ID Code, User Code, Status Code.
	Read User Code	Read Device User Code.
	Read Status Register	Read Device Status.
	Reprogram	Used to reload bitstream files from Flash to SRAM
	SRAM Erase	Erase SRAM data.
	SRAM Program	Configure the bitstream file to FPGA SRAM
	SRAM program JTAG 1149	JTAG write in pure data mode, without support for CRC, encryption, or compressed bitstream files.
Embedded Flash Mode	embFlash Erase, Program	Erase embFlash, then write data.
	embFlash Erase, Program, Verify	Erase embFlash, then write data and verify it.
	embFlash Erase Only	Erase embFlash only.
Embedded Flash Background Mode	embFlash Background Erase, Program	Perform background programming; erase the embedded Flash first, and then write the data.
	embFlash Background Erase, Program, Verify	Perform background programming; erase the embedded Flash first, then write data and verify it.
	emFlash Background Erase Only	Perform background programming; erase the embedded Flash.
	embFlash Background Erase, Program without Reprogram	Perform background programming; first erase the embedded Flash, then write the data without issuing the reprogram command.
	embFlash Background Erase, Program, Verify without Reprogram	Perform background programming: first erase the embedded Flash, then write and verify the data without issuing the reprogram command.
External Flash Mode	exFlash Erase, Program	Erase external Flash, then write data to external flash.
	exFlash Erase, Program, Verify	Erase external Flash, then write data and verify it.
	exFlash Bulk Erase	Perform bulk erase of external Flash.
	exFlash Verify	Verify external Flash data.

Access Mode	Operation	Description
	exFlash Background Erase, Program	Background programming; erase the external Flash first, and then write the data to the external Flash.
	exFlash Background Erase, Program, Verify	Background programming; erase the external Flash first, then write the data and verify it.
	exFlash Erase, Program in bscan	Erase external Flash and write data to external Flash in bscan mode.
	exFlash Erase, Program, Verify in bscan	Erase external Flash, then write data to external flash and verify in bscan mode.
	exFlash Verify in bscan	Read external Flash data and verify it against the bitstream file using bscan mode.
	exFlash Bulk Erase in bscan	Perform bulk erase of the external Flash data in bscan mode.
	exFlash C Bin Erase, Program	Erase external Flash, then write RISC-V bin files to the external Flash.
	exFlash C Bin Erase, Program, Verify	Erase external Flash, then write RISC-V bin files to the external Flash and verify it.
	exFlash Erase, Program thru GAO-Bridge	Use GAO-Bridge mode to first erase the external Flash, then write data to the external Flash.
	exFlash Erase, Program，Verify thru GAO-Bridge	Use GAO-Bridge mode to first erase the external Flash, then write data to the external Flash and verify it.
	exFlash C Bin Erase, Program thru GAO-Bridge	Use GAO-Bridge mode to first erase the external Flash, then write the RISC-V binary file to the external Flash.
	exFlash C Bin Erase, Program, Verify thru GAO-Bridge	Use GAO-Bridge mode to first erase the external Flash, then write the RISC-V binary file to the external Flash and verify it.
	exFlash Bulk Erase thru GAO-Bridge	Use GAO-Bridge mode to perform bulk erase of the external Flash.
	exFlash Verify thru GAO-Bridge	Use GAO-Bridge mode to read external Flash data and verify against the bitstream file.
	exFlash Export thru GAO-Bridge	Use GAO-Bridge mode to read external Flash data and save it as a binary file.
	exFlash Import thru GAO-Bridge	Use GAO-Bridge mode to read external Flash data and save it as a binary file.
	exFlash Erase Sectors About File	Erase the external Flash sectors corresponding to the file size.
	exFlash Erase Sectors About File thru GAO-Bridge	Erase the required external Flash sectors corresponding to the file size via GAO-Bridge.
	exFlash Verify thru JTAG IEEE 1149	Follow the JTAG IEEE 1149 protocol to read data from the external Flash and compare it with the bitstream file for verification.
	exFlash Bulk Erase thru JTAG IEEE 1149	Follow the JTAG IEEE 1149 protocol to perform a bulk erase of the external Flash.

Access Mode	Operation	Description
	exFlash Erase,Program thru JTAG IEEE 1149	Follow the JTAG IEEE 1149 protocol to first erase the external Flash, then program data into the external Flash.
	exFlash Erase,Program,Verify thru JTAG IEEE 1149	Follow the JTAG IEEE 1149 protocol to first erase the external Flash, then program the data and perform verification.
	exFlash C Bin Erase,Program thru JTAG IEEE 1149	Follow the JTAG IEEE 1149 protocol to first erase the external Flash, then program the RISC-V binary (.bin) file into the external Flash.
	exFlash C Bin Erase,Program,Verify thru JTAG IEEE 1149	Follow the JTAG IEEE 1149 protocol to first erase the external Flash, then program the RISC-V binary (.bin) file into the external Flash and perform verification.
Slave SPI Mode	Slave SPI Read ID Code	Read ID Code in SSPI mode.
	Slave SPI Scan exFlash	Scan exFlash in SSPI mode.
	Slave SPI Program SRAM	Write data to SRAM in SSPI mode.
GoConfigure IP Mode	embFlash Background Ese,Program,thru IP-JTAG	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM, then erase the embedded Flash, and write data to it.
	embFlash Background Erase,Program,Verify thru IP-JTAG	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the embedded Flash, then write data and verify it.
	embFlash Background Erase Only thru IP-JTAG	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM, then erase the embedded Flash.
	embFlash Background Program,Hotboot thru JTAG-JTAG	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the embedded Flash, write data and verify it, then perform a hot boot.
	embFlash Background Erase Only thru IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM, then erase the embedded Flash.
	embFlash Background Erase,Program,thru IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the embedded Flash, and write data.
	embFlash Background Erase,Program,Verify thru IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the embedded Flash, then write data and verify it.
	Read ID Register thru IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM, then read the ID

Access Mode	Operation	Description
		Code.
	Read User Code thru IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM, then read the User Code.
	Read Status Code thru IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM, then read the Status Code.
	exFlash Erase,Program,thru UART-IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the external Flash, and write data.
	exFlash Erase,Program,Verify thru UART-IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the external Flash, then write data and verify it.
	SRAM Reprogram thru UART-IP-SPI	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM, then reboot the device.
	embFlash Background Erase,Program,thru IP-I2C	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the embedded Flash, and write data.
	embFlash Background Erase,Program,Verify thru IP-I2C	Use GoConfig mode for background programming; first download the IP bitstream file into SRAM; erase the embedded Flash, then write data and verify it.
External Flash Mode Arora V	exFlash Erase,Program Arora V	First erase the external Flash of the Arora V devices, then write data into the external Flash.
	exFlash Erase,Program,Verify Arora V	First erase the external Flash of the Arora V devices, then write data into the external Flash, and verify it.
	exFlash Verify Arora V	Verify the data in the external Flash of the Arora V devices.
	exFlash Bulk Erase Arora V	Perform bulk erase of the external Flash data of the Arora V devices.
	exFlash C Bin Erase,Program Arora V	First erase the external Flash of the Arora V devices, then write the RISC-V binary file into the external Flash.
	exFlash C Bin Erase,Program,Verify Arora V	First erase the external Flash of the Arora V devices, then write the RISC-V binary file into the external Flash, and verify it.
	exFlash Background Erase,Program Arora V	Background programming; first erase the external Flash of the Arora V devices, then write data into the external Flash.
	exFlash Background Erase,Program,Verify Arora	Background programming; first erase the external Flash of the Arora V devices, then

Access Mode	Operation	Description
	V	write data and verify it.
	exFlash Erase,Program thru GAO-Bridge Arora V	Use GAO-Bridge mode to erase the external Flash of the Arora V devices, then write data into it.
	exFlash Erase,Program,Verify thru GAO-Bridge Arora V	Use GAO-Bridge mode to erase the external Flash of the Arora V devices, then write data and verify it.
	exFlash Verify thru GAO-Bridge Arora V	Use GAO-Bridge mode to verify the external Flash data of the Arora V devices.
	exFlash Export thru GAO-Bridge Arora V	Establish a JTAG-to-SPI bridge and read the external Flash data of the Arora V devices, saved as a binary file.
	exFlash Erase Sectors About File Arora V	Erase the external Flash sectors corresponding to the file size for Arora V devices.
	exFlash Erase Sectors About File thru GAO-Bridge Arora V	Erase the external Flash sectors corresponding to the file size for Arora V devices via GAO-Bridge.
Internal Flash Mode	sFlash Erase,Program	First erase the sipped Flash, then write the bitstream into the Flash.
	sFlash Erase,Program,Verify	First erase the sipped Flash, write the bitstream into the Flash, and then read back the data for comparison with the bitstream file.
	sFlash Verify Only	Read data from the sipped Flash and compare it with the corresponding bitstream file.
	sFlash Bulk Erase	Perform bulk erase of the sipped Flash.
	sFlash Background Erase,Program	Background programming; first erase the sipped Flash, then write the bitstream into the Flash .
	sFlash Background Erase,Program,Verify	Background programming; first erase the sipped Flash, write the bitstream into the Flash, and then read back the data for comparison with the bitstream file.
	sFlash Erase,Program in bscan	Use bscan mode to first erase the sipped Flash, then write the bitstream into the Flash.
	sFlash Bulk Erase in bscan	Use bscan mode to perform bulk erase of the sipped Flash.
	sFlash Erase,Program,Verify thru GAO-Bridge	Use GAO-Bridge mode to first erase the sipped Flash, then write the bitstream into the Flash.
	sFlash Background Erase,Program,Verify thru GAO-Bridge	Use GAO-Bridge mode for background programming; the SRAM is not erased; instead directly erase the sipped Flash, write the bitstream into the Flash, and then read back the data for comparison with the bitstream file.
	sFlash Export thru	Use GAO-Bridge mode to read the sipped

Access Mode	Operation	Description
	GAO-Bridge	Flash data and save it as a binary file.
	sFlash Erase Sectors About File	Erase the sipped Flash sectors determined by the size of the bitstream file.
	sFlash Erase Sectors About File thru GAO-Bridge	Use GAO-Bridge mode to erase the sipped Flash sectors determined by the size of the bitstream file.
MCU Mode(L)	Firmware Erase Only	Erase firmware data.
	Firmware Erase,Program	First erase firmware data, then write data.
	Firmware Erase,Program,Verify	First erase firmware data, then write data and verify it.
SecureFPGA Mode (L)	Connect to JTAG of MCU	Transfer JTAG control of the device to MCU.
	Firmware Erase,Program Securely	Erase firmware data, then securely write and verify firmware.
	Authentication Code Erase,Program,Verify	–
	Authentication Code Read	–
I2C Mode	I2C Program SRAM	Download the bitstream file into SRAM using I2C configuration mode.
	I2C Program Flash	Download the bitstream file into the embedded Flash using I2C configuration mode.
	I2C Erase Flash Only	Erase data in the embedded Flash using I2C configuration mode.
Debugging/Temporary Mode	Set Flash QE For 9X/18X	Enable the QE bit for the sipped Flash in 9X/18X devices
	Set Flash QE For Arora V	Set the QE mode for the external Flash of Arora V devices; enable the QE bit for the external SPI-Flash of Arora V devices.
	exFlash Detect ID	Read the ID of the external Flash.
	Program Arora V DNA Data	Download the DNA data of Arora V devices
	efuse read Arora V	Read 128-bit data from the user_fuse area of Arora V devices.
	SRAM Program thru Serial	Download the bitstream file into SRAM using serial configuration mode.
	Set Boot Address Arora V	Configure the boot address of Arora V devices. This operation programs the boot address into SRAM, which is volatile and will be lost after power-off. Before performing this configuration, the ready signal must be held low; once the configuration is complete, the ready signal should be driven high. This function is intended for debugging purposes only.

Note!

- LittleBee family chips include embedded Flash.
- Background Programming: Performs programming without erasing SRAM or interrupting current operations.

- GAO-Bridge Method: Typically used to configure external Flash or sipped Flash. First, establish a JTAG-to-SPI bridge by loading the GAO-Bridge function into SRAM; then configure the Flash via JTAG.
- GoConfigIP Method: Operates on the same principle as GAO-Bridge. However, it is recommended to add the GoConfigIP functions into the user's design project.
- sFlash is an abbreviation for "sipped Flash".
- If the user design multiplexes JTAG pins as general purpose I/Os and the JTAGSEL_N pin is not bonded out in the current package, the JTAG function cannot be restored by directly pulling this signal low via hardware. In this case, users must use internal logic to control JTAGSEL_N. Under this condition, certain FPGA programming operations in non-empty device states (involving SRAM erase operations) will be restricted, as follows:
  1. Bypass, Read Device Code, Read User Code, Read Status Register, and Reprogram are not restricted.
  2. Operation modes with background programming support are not restricted.
  3. Other operation modes except those listed in 1 and 2 are not supported.

3.4.1 SRAM Configuration

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Program/Configure Device" or " " or double-clicking "Operation".
2. Select SRAM Mode in "Access Mode" drop-down list.
3. Select operation in "Operation" drop-down list.
4. For non-Gowin devices, you need to manually specify the length of the instruction register or instruct the programmer to read the length of the instruction register of the BSDL file.
5. Click "Save" to complete the configuration.

Note!

- Non-Gowin devices (JTAG-NOP) only support the Bypass operation.
- Descriptions for 138K/75K devices: When the bitstream file is configured in Quad mode in Gowin Software (including Multi-Boot and MSPI JUMP), and "Print BSRAM Initial Value" is set to "BSRAM Partial Print Mode" to print the BSRAM initial values, direct SRAM configuration is not supported. If direct SRAM configuration is required, the user must set "Print BSRAM Initial Value" to "Default Mode" to print the BSRAM initial values and regenerate the bitstream file.

3.4.2 Embedded Flash Configuration of LittleBee Family FPGA Products

The LittleBee family FPGA products includes embedded Flash; you can select Embedded Flash Mode.

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit> Configure Device" or " " or double-clicking "Operation".
2. Select "Embedded Flash Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. Select the programming bitstream file from "File name".
5. Click "Save" to complete the configuration.

3.4.3 External Flash Configuration

Note!

This applies to LittleBee family, GW2A, GW2AR and GW3A series devices.

Gowin Programmer supports external Flash programming. The external flash configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select "External Flash Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "exFlash Program" or "exFlash Verify" is selected in the "Operation", the corresponding programming bitstream file needs to be selected in "File name".
5. For external Flash part number, SPI Flash with read commands of 0x03 or 0x0B are supported.
6. The default Flash type is "Generic Flash".
7. Select the initial address of the external Flash. Currently, the default is 0x000000.
8. Click "Save" to complete the configuration.

3.4.4 Sipped Flash Configuration

Configuration of the sipped Flash can be classified into two categories according to the device characteristics:

- GW1AN-3A(B)/GW1AN-5A(B)
- GW2AN-9X/GW2AN-18X

Gowin Programmer supports sipped Flash programming of GW1AN-3A(B)/GW1AN-5A(B)/GW2AN-9X/GW2AN-18X devices. The configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " ", or double-clicking "Operation".
2. Select "Internal Flash Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "sFlash Program" or "sFlash Verify" is selected in the "Operation", the corresponding programming bitstream file needs to be selected in "File name".
5. The default Flash type is "Generic Flash".
6. Select the initial address of the Flash. Currently, the default is 0x000000.

7. Click "Save" to complete the configuration.

GW1AN-1SA/GW1AN-1C/GW2ANR-18C/GW2AN-55C

Gowin Programmer supports sipped Flash programming of GW1AN-1SA/GW1AN-1C/GW2ANR-18C/GW2AN-55C devices. The configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " ", or double-clicking "Operation".
2. Select "External Flash Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "exFlash Program" or "exFlash Verify" is selected in the "Operation", the corresponding programming bitstream file needs to be selected in "File name".
5. For external Flash part number, SPI Flash with read commands of 0x03 or 0x0B are supported.
6. The default Flash type is "Generic Flash".
7. Select the initial address of the external Flash. Currently, the default is 0x000000.
8. Click "Save" to complete the configuration.

3.4.5 External Flash of Arora V Devices Configuration

Gowin Programmer supports external Flash programming. The external flash configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select "External Flash Mode Arora V" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "exFlash Program" or "exFlash Verify" is selected in the "Operation", the corresponding programming bitstream file needs to be selected in "File name".
5. The external Flash supports SPI Flash with read commands 0x03, 0x0B, and 0x6B.
6. The default Flash type is "Generic Flash".
7. Select the initial address of the Flash. Currently, the default is 0x000000.
8. Click "Save" to complete the configuration.

3.4.6 Slave SPI Mode

In Slave SPI Mode, the download cable shall be connected to the dedicated SSPI pin, see *UG290, Gowin FPGA Products Programming and Configuration Guide*.

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit> Configure Device" or " " or double-clicking "Operation".
2. Select "Slave SPI Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "Slave SPI Program SRAM" is selected in "Operation", the corresponding programming bitstream file needs to be selected in "File Name".
5. Click "Save" to complete the configuration.

3.4.7 MCU Mode Configuration

Gowin Programmer supports firmware updates for embedded M3 hard-core processor in the device. The configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select "MCU Mode" or "MCU Mode L" in "Access Mode" drop-down list.
3. Select an operation in "Operation" as required.
4. If "Firmware Program" is selected in the "Operation", the corresponding programming bitstream file and firmware need to be selected in "File name".
5. Click "Save" to complete the configuration.

3.4.8 SecureFPGA Mode Configuration

Gowin Programmer supports GW1NSE/GW1NSER secure FPGA configuration. The configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select "SecureFPGA Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "Authentication Code Erase, Program, Verify" is selected in the "Operation", the corresponding bitstream file needs to be selected in "File name".
5. Click "Save" to complete the configuration.

3.4.9 I2C Mode Configuration

In I2C Mode, the download cable shall be connected to the dedicated I2C pin. The configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select "I2C Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "I2C Program SRAM" or "I2C Program Flash" is selected in the "Operation", the corresponding bitstream file needs to be selected in "File name".
5. Click "Save" to complete the configuration.

3.4.10 Debugging Mode Configuration

Gowin Programmer supports debugging mode. The configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select "Debugging/Temporary Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. If "SRAM Program thru Serial" is selected in the "Operation", the corresponding programming bitstream file needs to be selected in "File name".
5. Click "Save" to complete the configuration.

3.4.11 GoConfigIP Mode Configuration

Gowin Programmer supports programming using goConfig IP, and this mode is intended for user debugging phases to verify the functions of GoConfigIP. The configuration process is as follows:

1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select "GoConfig IP Mode" in "Access Mode" drop-down list.
3. Select an operation in "Operation" drop-down list.
4. When the operation in the "Operation" field involves downloading a bitstream file, the corresponding programming bitstream stream file needs to be selected in "File name".
5. Click "Save" to complete the configuration.

3.4.12 Background Programming Mode Configuration

Gowin Programmer supports background programming, which allows programming without affecting the current operating of the device. The configuration process is as follows:

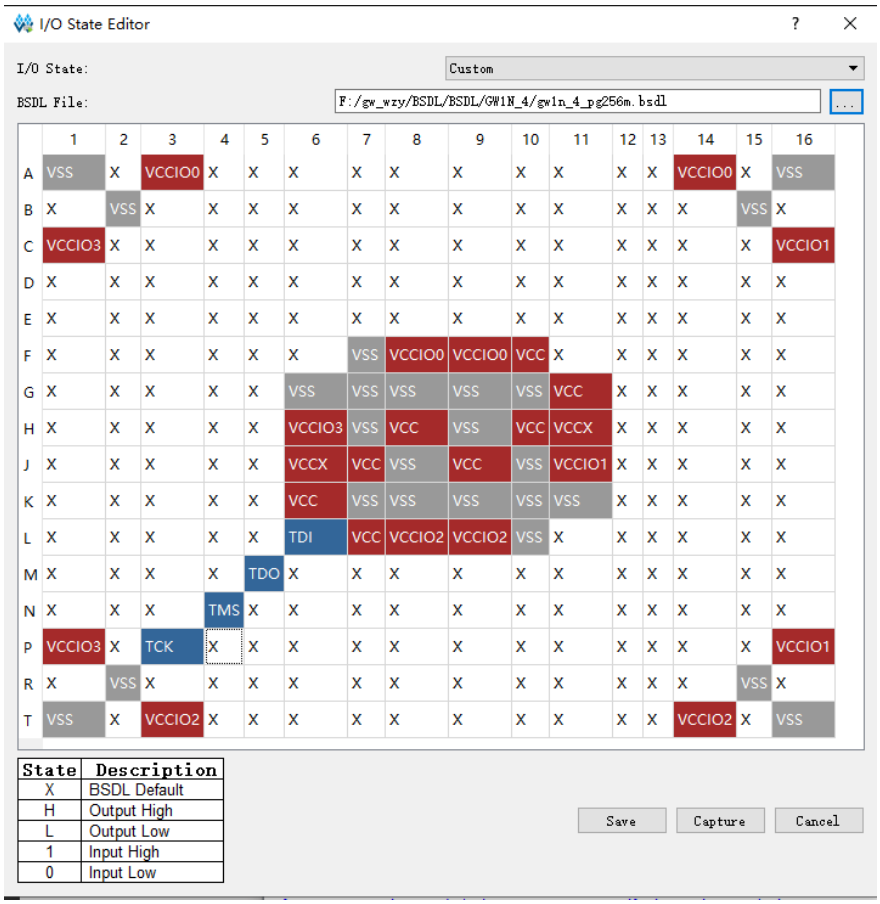
1. Select the device row and open the "Device Configuration" dialog by clicking "Edit > Configure Device" or " " or double-clicking "Operation".
2. Select an option that includes Background programming in "Operation" drop-down list.
3. When the operation in the "Operation" field involves downloading a bitstream file, the corresponding programming bitstream stream file needs to be selected in "File name".
4. Click "Save" to complete the configuration.

3.5 Edit Pin State

Programmer uses I/O State Editor to edit the I/O pin value, which allows you to set the state of the pins prior to programming download.

1. Select the device row and open I/O State Editor by clicking "Edit > I/O State" or right clicking "I/O State".
2. Select the BSM file that conforms with the device part number and package.
3. Modify the pin state by clicking on the cell location or set the same state for all pins by right-clicking on the menu.

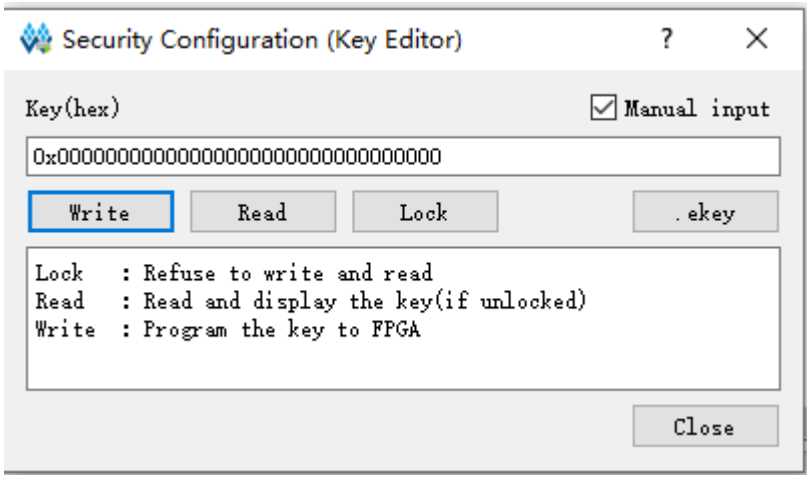
Figure 3-8 I/O State Editor



3.6 Device Security

When programming with encrypted bitstream files, you need to write the key of the bitstream file to FPGA. Select the device row and open Security Configuration dialog by clicking "Edit > Configure Security" from the menu or right-clicking "Security Key Setting", as shown in Figure 3-9.

Figure 3-9 Security Configuration



- Manual input: Input the key in plaintext or not.
- .ekey: Open the key file.

- Write: Write the specified key value to the FPGA.
- Read: Read and display the key value in the FPGA in the case of unlocking.
- Lock: Lock the key in the FPGA, and the key is unable to be read or written after being locked.

There are following two methods to write the decryption key.

Through Plaintext

After the decryption key is written successfully, readback the written value by clicking the "Read" button on the interface to verify.

After the key is written successfully, lock it in the FPGA via the Lock command. Once you have performed this operation, any read and write key operations will be invalid; the key value cannot be modified, and all the read bits are 1.

Through File

Open the key file, then click the "Write" button to first write the key to the development board. After reading the key for verification, if the verification succeeds, the key will be locked inside the FPGA. If you click “Write” without opening a key file, a failure message will be displayed.

After the decryption key is set, the encrypted bitstream data will only work when the data matches the decryption key. The key does not affect the non-encrypted bitstream data.

Note!

All bits of the initial value of Gowin FPGA keys are 0. If a certain bit of the key value is changed to 1, it cannot be changed back to 0. For example, the key value written during an operation is 00000000-00000000-00000000-00000001, and the last bit of the modified key must be 1. For more detailed information, see *TN654, GW2A(R) series of FPGA Products AES Programming Guide*.

3.7 Programming Download

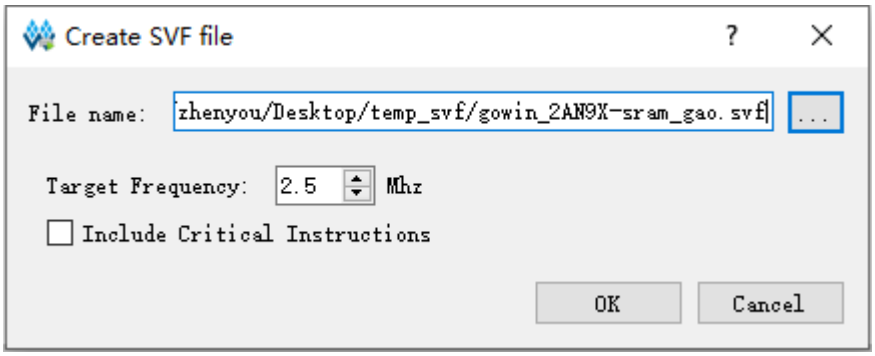
After configuring the download cable and daisy chain, select "Program/Configure" from the menu or click "▶" to download. The final result will be displayed in the output panel.

3.8 SVF File Creation

Using fs file to create SVF file is supported.

1. Configure embedded Flash mode according to the GW1N/GW1NZ and select GW1N4 device.
2. Select the device chain and open the Create SVF File dialog by clicking, "Edit> SVF File Create" or right-clicking to select "SVF File Create".
3. The generated SVF file can be named in File Name and it can choose SVF save path, as shown in Figure 3-10.
4. Click "OK" to complete the creation of the SVF file.

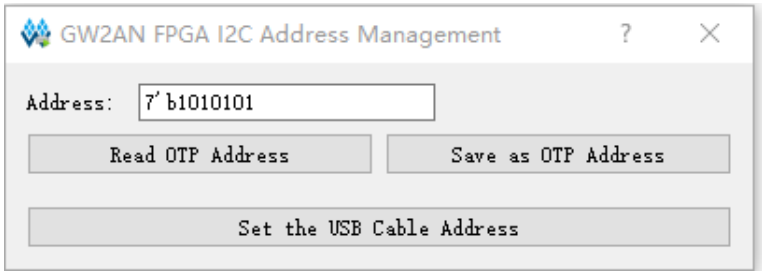
Figure 3-10 Create SVF File



3.9 I2C Slave Address Operation

Read, write, and configuration operations of I2C slave address are supported. Currently these operations are available only for GW2AN-18X and GW2AN-9X devices.

Figure 3-11 I2C Slave Address Operation



- Read OTP Address: Read I2C slave address.
- Save as OTP Address: Write the specified I2C slave address to the FPGA.
- Set the USB Cable Address: When operating the FPGA in I2C mode, you need to set the I2C slave address.

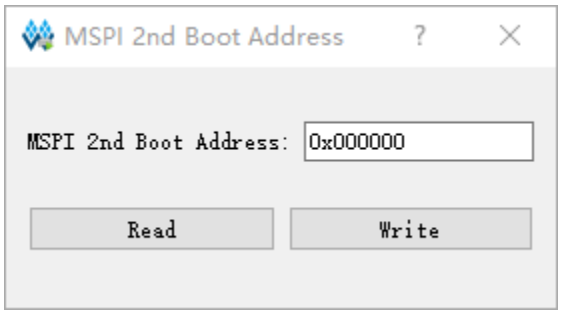
Note!

The I2C slave address only supports 2-bit configuration. The default address is 7'b1010000, and only the yellow character 0 can be modified to 1, but it cannot be changed back to 0.

3.10 MSPI 2nd Boot Address Operation

Support read and write operations for the Golden Image address for Arora V devices, ranging from 0x000000 to 0xF00000.

Figure 3-12 MSPI 2nd Boot Address Operation



- Read: Read the MSPI 2nd boot address
- Write: Write the specified MSPI 2nd boot address to the FPGA

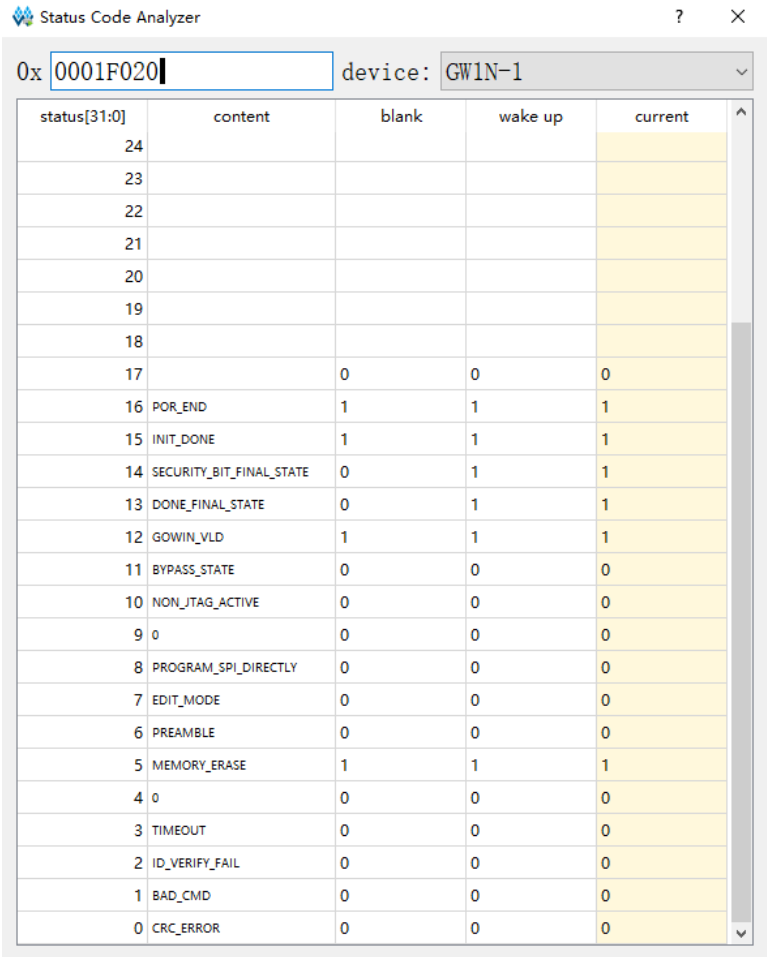
Note!

- The configurable bits correspond to bit[23:20] of the address bus (a total of 4 bits), which are used to set the upper 4 bits of the boot address.
- By default, all four bits are 0 at factory shipment (that is, the default boot address is 0x800000).
- These bits can only be programmed from 0 to 1, and cannot be reverted from 1 back to 0.
- Ensure that the V_{EFUSE} input voltage meets the specification (1.62 V to 1.98 V).
- For example, if bit20 is programmed to 1 (e.g., setting the offset to 0x100000), this bit will be permanently locked at 1, and only the remaining bits that are still 0 (bit21–bit23) can be programmed thereafter.

3.11 Status Code Analyzer

To check the status of the development board, you can analyze the status code to determine its status. By clicking on the menu bar "Tools > Analyzer Status Viewer" or finding the status code display row, right-click the row where the device is located and select "Analyze Status Code" to open the "Status Code Analyzer" dialog box, as shown in Figure 3-13 below. For the descriptions of status register codes, you can see *TN711, GOWIN FPGA Status Register Codes*.

Figure 3-13 Status Code Analyzer



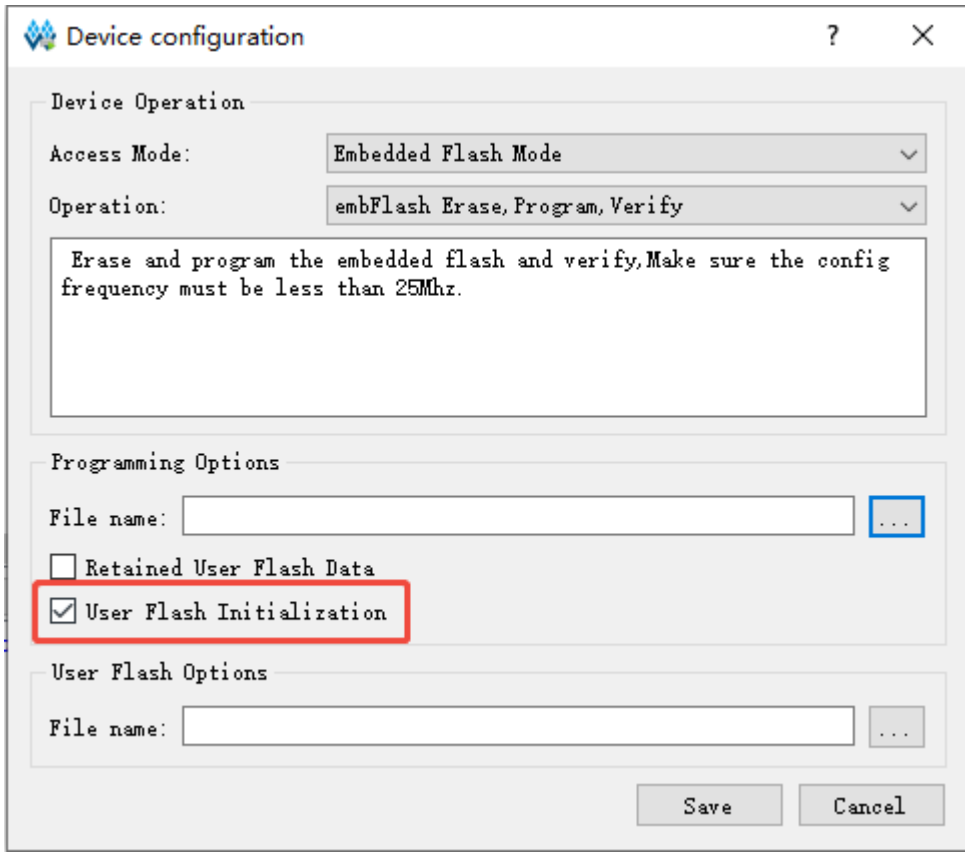
status[31:0]	content	blank	wake up	current
24				
23				
22				
21				
20				
19				
18				
17		0	0	0
16	POR_END	1	1	1
15	INIT_DONE	1	1	1
14	SECURITY_BIT_FINAL_STATE	0	1	1
13	DONE_FINAL_STATE	0	1	1
12	GOWIN_VLD	1	1	1
11	BYPASS_STATE	0	0	0
10	NON_JTAG_ACTIVE	0	0	0
9	0	0	0	0
8	PROGRAM_SPI_DIRECTLY	0	0	0
7	EDIT_MODE	0	0	0
6	PREAMBLE	0	0	0
5	MEMORY_ERASE	1	1	1
4	0	0	0	0
3	TIMEOUT	0	0	0
2	ID_VERIFY_FAIL	0	0	0
1	BAD_CMD	0	0	0
0	CRC_ERROR	0	0	0

First, select the appropriate development board (you can select the board in the device list), then enter the status code. The results for each bit of the status code will be output in the list below.

3.12 User Flash Initialization and Data Retention

LittleBee family provides users with User Flash space. User Flash data can be used to program embedded Flash and User Flash space at the same time. For the security of the design, this operation only supports user flash programming on the Programmer and readback is not supported. You can choose user flash initialization file with .fi suffix when programming, as shown in Figure 3-14 .

Figure 3-14 User Flash Initialization



The user flash data retention means that the data in the User Flash area will not be altered when programming the embedded Flash. "Retained User Flash Data" and "User Flash Initialization" are mutually exclusive, and only one operation can be selected, as shown in the figure below.

Figure 3-15 Retained User Flash Data

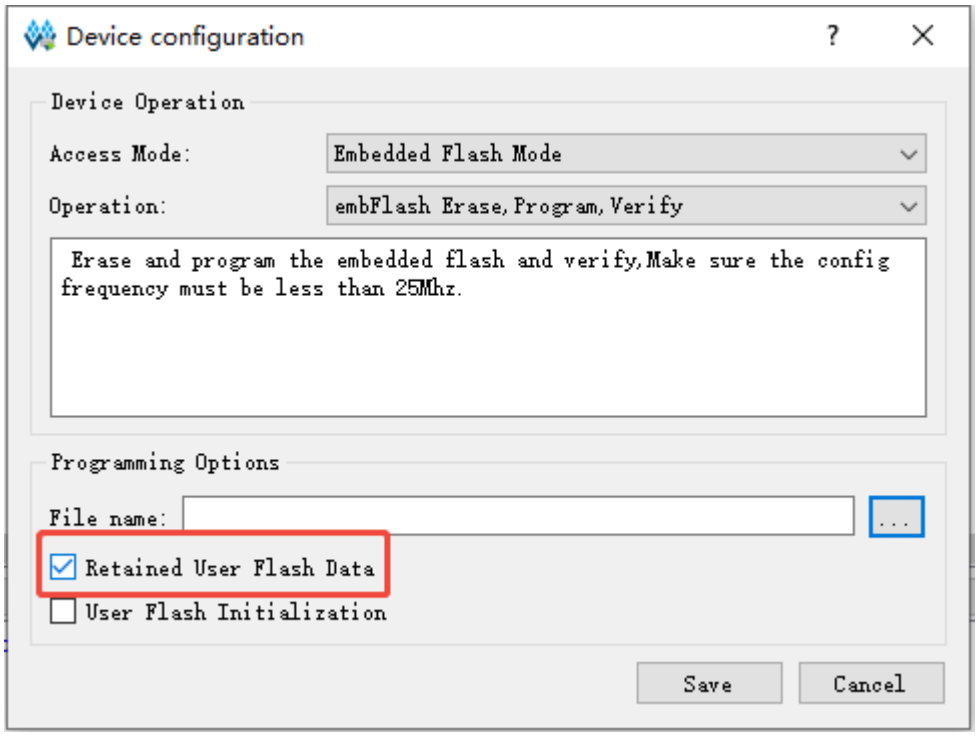


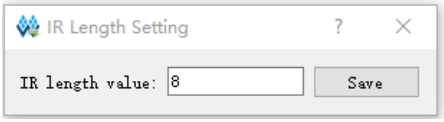
Table 3-2 User Flash Reference

Primitive	Supported Devices	Address	Data Width
FLASH96K	GW1N-1, GW1N-1S, GW1NR-1	48*64	32Bits
FLASH64KZ	GW1NZ-LV1	32*64	
FLASH64K	GW1NZ-ZV1, GW1NZ-1C		
FLASH256K	GW1N-4, GW1N-4B, GW1N-4D, GW1NR-4, GW1NR-4B, GW1NR-4D, GW1NRF-4B, GW1NS-4, GW1NS-4C, GW1NSR-4, GW1NSR-4C, GW1NSER-4C	128*64	
FLASH96KA	GW1N-2, GW1N-2B, GW1N-2C, GW1N-1P5, GW1N-1P5B, GW1N-1P5C, GW1NR-2, GW1NR-2B, GW1NR-2C, GW1NZ-2B, GW1NZ-2C	48*64	
FLASH608K	GW1N-9, GW1N-9C, GW1NR-9, GW1NR-9C	304*64	

3.13 Adjustment of JTAG State Machine Shift-IR Width

Adjustment of JTAG state machine shift-ir width is supported, and it is only used during daisy-chain configuration.

Figure 3-16 Adjustment of JTAG State Machine Shift-IR Width



- IR length value: The width value of TAG state machine shift-ir.
- Save: Writes the selected value into the FPGA.

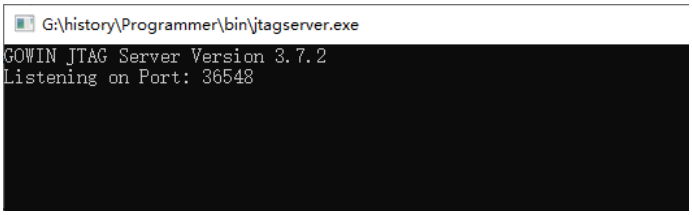
3.14 Remote Configuration and Programming

The remote programming and configuration is supported. When using this function, you need to ensure that TCP/IP communication between the two PCs is functioning correctly.

1. On the remote PC, start jtagserver.exe. Locate the jtagserver file in the same directory as the programmer, double-click to open it, as shown in Figure 3-17.

Figure 3-17 jtagserver Location

Figure 3-18 jtagserver in Running State



jtagserver listens on a specified port to support client connections. The default port is 36548. If you need to use a different port, you can configure in /Programmer/bin/data/config/gvi.ini file.

Figure 3-19 jtagserver in Running State

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

- Host: IP address of the remote.
- Port: Port listening on.

2. On the local PC, start programmer.exe. In the "Tools" menu, locate the "Add GVC Server" option. Click to open the GVC interface, as shown in Figure 3-21.

Figure 3-20 Add GVC Server Option

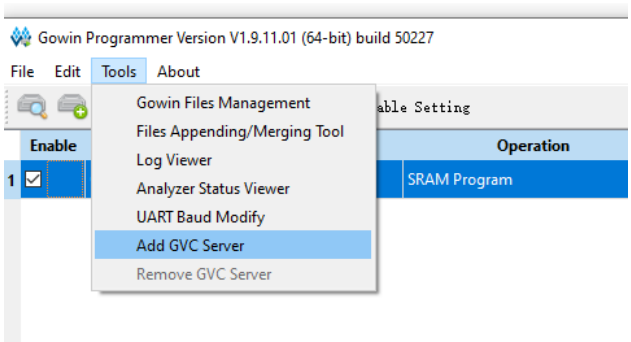
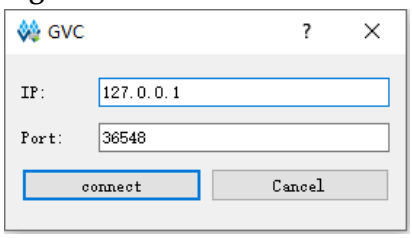


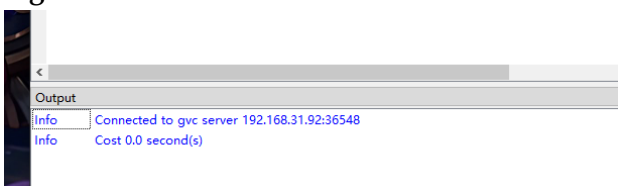
Figure 3-21 GVC



- IP: IP address of the remote server.
- Port: Port on which the remote server is listening.
- Connect: Connect to the remote server.
- Cancel: Cancel the operation and exit the page.

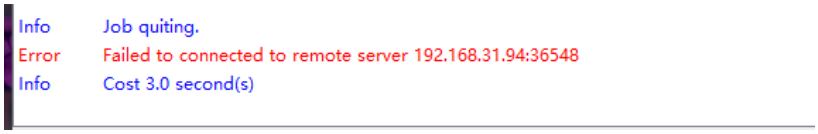
After clicking "Connect," if the connection is successful, a success screen as shown in Figure 3-22 will appear.

Figure 3-22 Connection Succeed



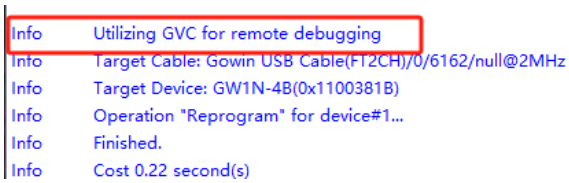
If the connection fails, an error screen as shown in Figure 3-23 will appear.

Figure 3-23 Connection Failed



3. After a successful connection, all remote operations are identical to local operations. The only difference is that remote debugging information will be printed for distinction.

Figure 3-24 Remote Operation Output



4. After completing the operation, navigate to the "Tools" menu and select the "Remove GVC Server" option. Click it to exit the remote configuration mode.

Figure 3-25 Exit Remote Configuration Mode

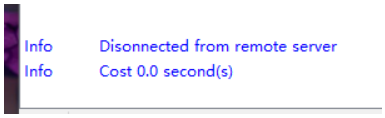


Table 3-3 Reference Table for Device and Remote Programming Function

Device	Function
GW1N-1 GW1N-1S GW1N-4 GW1NSER-4C GW1NR-4	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" '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' '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	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" 'SRAM Program'

Device	Function
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	'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 GW2A-18C GW2AN-55C GW2AR-18 GW2AR-18C GW2ANR-18C	'Read Device Codes' 'Read User Code' 'Read Status Register' "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'

Device	Function
	'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-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 Arora V' 'exFlash Erase,Program,Verify thru GAO-Bridge Arora V' 'exFlash Verify thru GAO-Bridge Arora V' 'exFlash Erase,Program Arora V' 'exFlash Erase,Program,Verify Arora V' 'exFlash C Bin Erase,Program Arora V' 'exFlash C Bin Erase,Program,Verify Arora V' 'exFlash Bulk Erase Arora V' 'exFlash Verify Arora V'

Note!

When using GAO-Bridge to operate the external Flash, there is no need to specifically set the mode pin.

3.15 File Conversion Function

The file conversion function is designed to help users easily convert between different file formats. Open “Gowin Files Management” dialog box via the menu bar by selecting "Tools > Gowin Files Management", as shown in Figure 3-26.

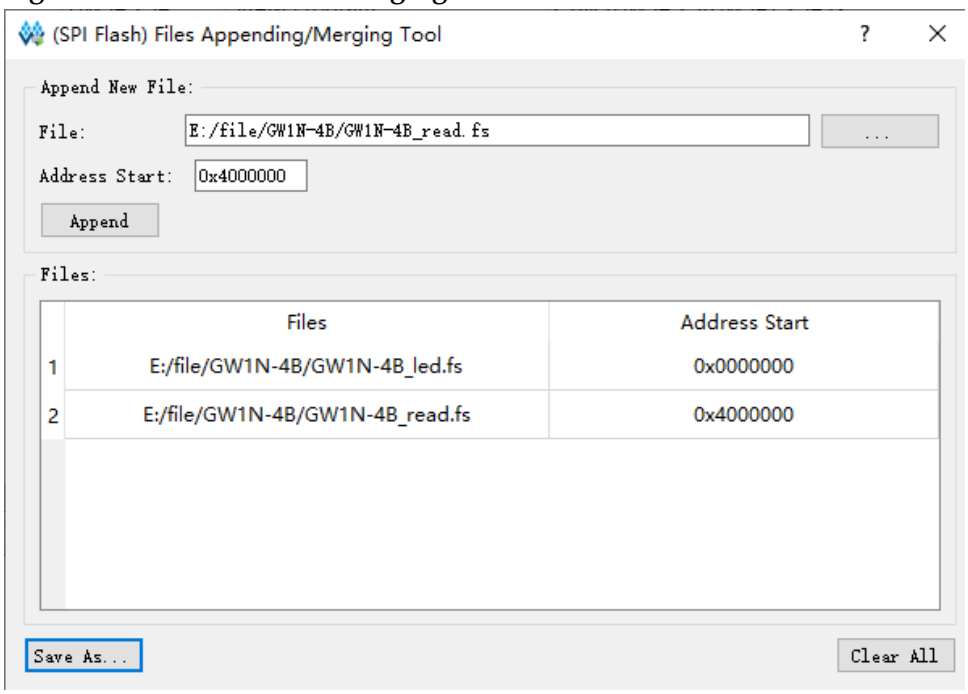
Figure 3-26 File Conversion Interface

- Operation: Used to select the type of input file to be operated on, as well as the target file type for conversion.
- Clear: Used to clear all files listed in the file box.
- Add File(s): Used to select the input file type.
- Convert/Merge: Used to execute the file conversion operation.

3.16 Bitstream File Merging Function

It supports merging multiple bitstream files (.fs or .bin) into a single .bin file based on specified starting addresses. Open the "(SPI Flash) Files Appending/Merging Tool" dialog box by selecting "Tools > Files Appending/Merging Tool" from the menu bar, as shown in Figure 3-27.

Figure 3-27 Bitstream File Merging Interface



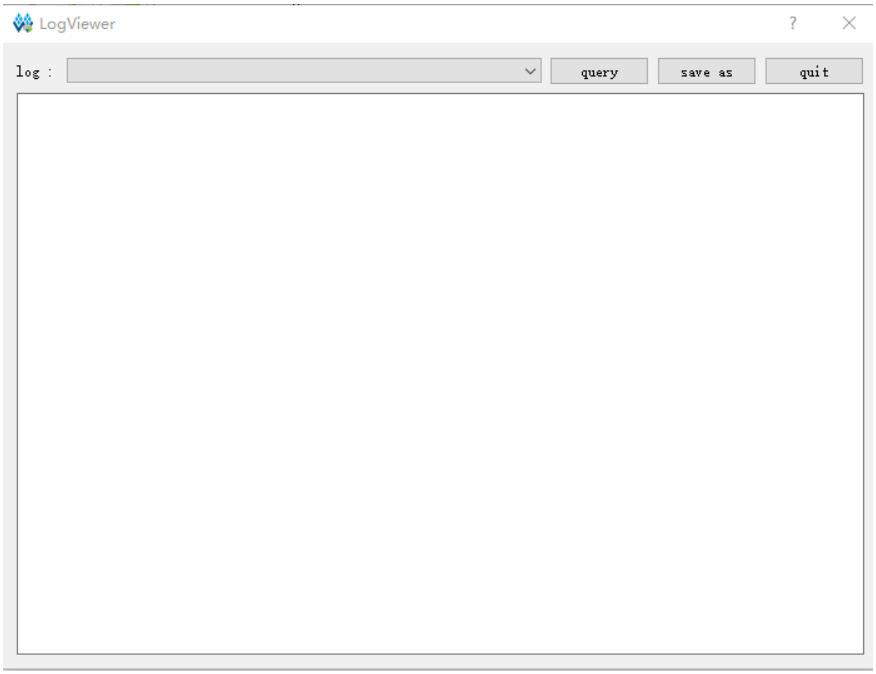
- File(...): Used to select an .fs or a .bin file. Its address will be displayed in the text box.
- Address Start: Used to specify the start address.

- Append: Used to add the specified .fs or .bin file and address to the list below.
- Save As...: Used to convert each .fs file in the list to .bin format and write them sequentially into a new .bin file based on the specified addresses.
- Clear All: Used to clear all data from the list.

3.17 Log Viewer Function

During software operation, if the message bar displays “Error Found,” it indicates a runtime error has occurred. Use the log viewer to obtain detailed error information for troubleshooting. Open the “Log Viewer” dialog box via the menu bar by selecting "Tools > Log Viewer", as shown in Figure 3-28.

Figure 3-28 Log Viewer Interface

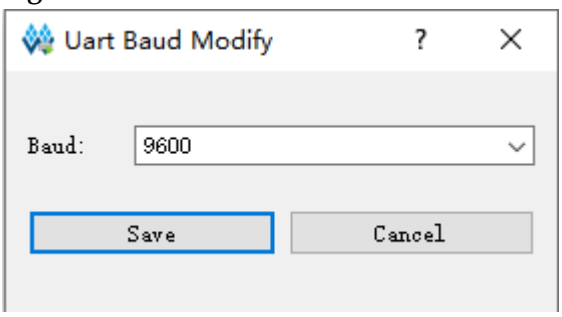


- query: Used to search for xx.log files and display the results in the dropdown list on the left for quick selection.
- save as: Used to save all log information below into a new file.
- quit: Exit the page.

3.18 Baud Rate Modification Function

It supports modifying the UART baud rate. Open the "Uart Baud Modify" dialog box via the menu bar by selecting "Tools > Uart Baud Modify", as shown in Figure 3-29.

Figure 3-29 Baud Rate Modification Interface



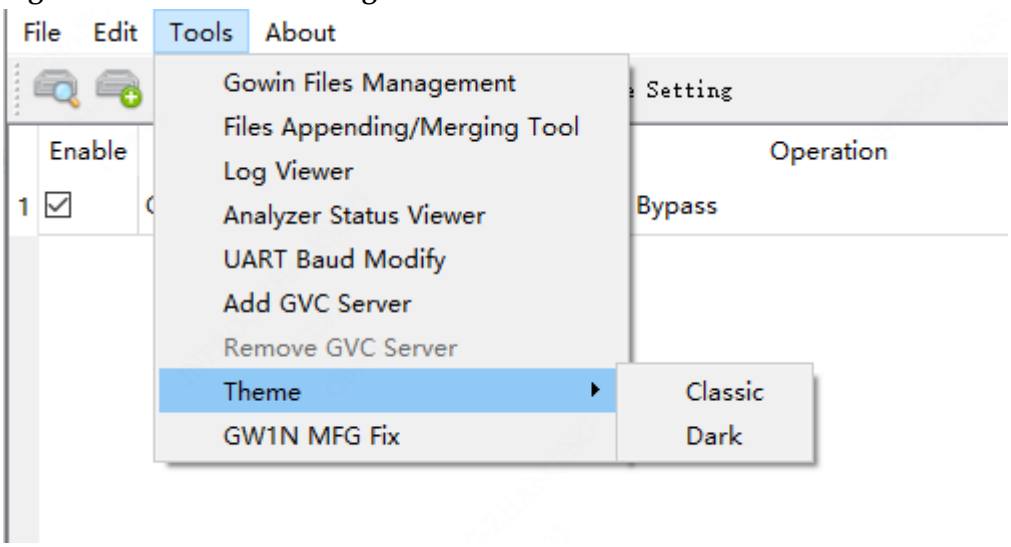
- Save: Save the modified baud rate.
- Cancel: Cancel the current operation and exit the page.

3.19 Interface Theme Switching

The software supports switching the interface theme between Classic and Dark modes. Select "Tools > Theme" from the menu bar to switch to the desired background, as shown in Figure 3-30.

- Classic: Standard mode
- Dark: Dark mode

Figure 3-30 Theme Switching



Note!

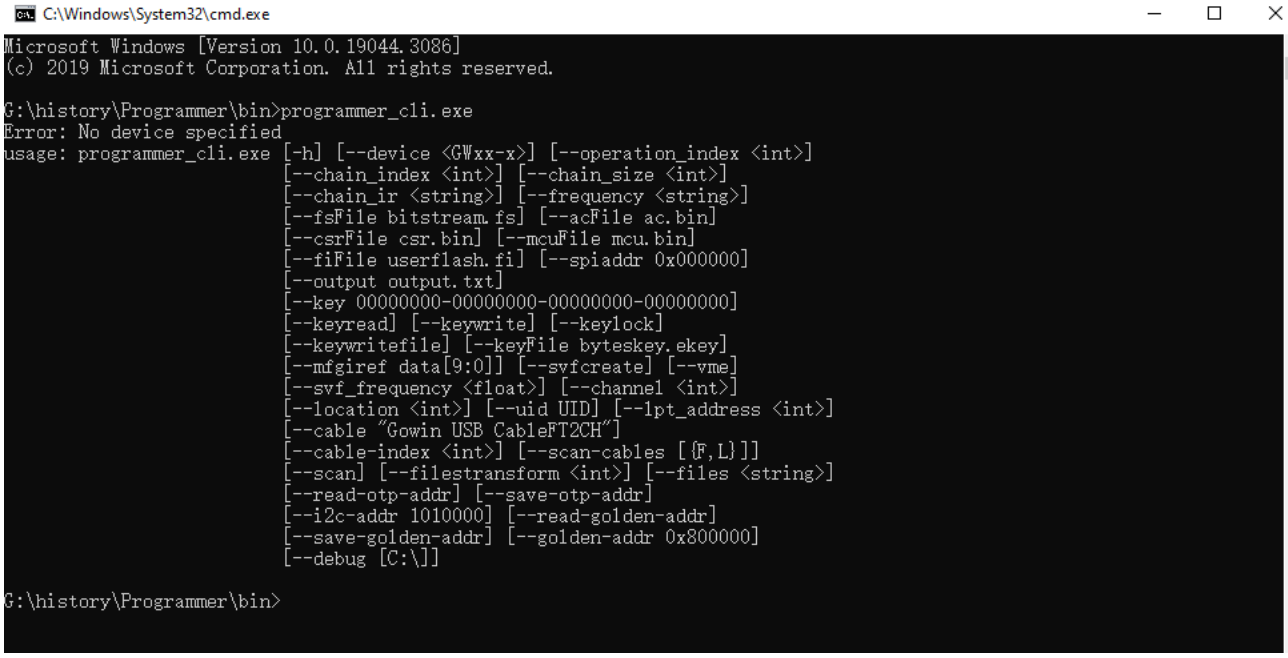
The software must be restarted for the changes to take effect.

4 Programmer_cli Programming Download Flow

4.1 Preview and Help

Open the programmer_cli tool in CMD. When no parameters are used, a brief help description will be prompted, and no device specified will also be prompted.

Figure 4-1 Open programmer_cli in CMD



Use the parameter –help to get more help information.

Figure 4-2 Help Information

```
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 GW1NSER-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: bsdl 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)"
                        "Digilent 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: Digilent 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 Scan USB Cable Device

Scan and display the connected USB Cable information.

Use programmer_cli.exe --scan-cables

Figure 4-3 Scan USB Cable Device

```
G:\history\Programmer\bin>programmer_cli.exe --scan-cables
 Cable found:  Gowin USB Cable(FT2CH)/0/99091/GW2OLCRZ (USB location:99091) (SN: GW2OLCRZ)
 Cost 0.05 second(s)
```

- Gowin USB Cable (FT2CH) is the cable type.
- /0/ is the channel number.
- 99091 is the USB location information.

- /GW20LCRZ is the USB Cable UID.
- (USB location:99091) (SN: GW20LCRZ) is the description information.

4.3 Specify USB Cable Type and Port

When scanning or configuring FPGA, you can specify the USB Cable type using the parameter "--cable-index". When --cable-index is 0, the cable type is Gowin USB Cable (GWU2X); when --cable-index is 1, the cable type is Gowin USB Cable (FT2CH), and so on; and --cable-index is 1 by default.

When using a multi-port cable, such as an FTDI cable, there will be multiple ports to choose. You can use the parameter "--channel" to specify the port, and the value of -channel is 0 by default; the help information is as follows:

Figure 4-4 Help Information

```
--channel <int>       Define download cable channel. Default is 0. Only works for Gowin USB Cable(FT2CH)
                          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: Digilent USB Device;
                       4: USB Debugger A;
                       5: Gowin USB Cable(WINUSB);
                       Higher priority than --cable, default cable-index is 0
```

4.4 Specify USB Cable Location or UID

When using multiple USB Cables, you can specify the USB port location or Cable UID to confirm a USB Cable device, and the help information is as follows:

Figure 4-5 Help Information

```
 --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 Scan FPGA Device

Scan the device with the following instruction.

programmer_cli.exe –scan

Figure 4-6 Scan FPGA Device

```
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 Specify Programmer Operation Mode

Use the parameter "--operation_index" or "--run" or "-r" to specify the operation mode, such as SRAM, Flash configuration, etc. The help information is as follows:

Figure 4-7 Help Information

```
 --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: bsdl 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;
```

When configuring the FPGA, use the parameter "--device" to specify FPGA Device, and the help information is as follows:

Figure 4-8 Help Information

```
 --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
```

You can use the following command formats to configure SRAM or Flash:

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 is used to specify the JTAG frequency, currently only applicable to FTDI Cable; U2X Cable frequency is fixed at 1.33MHz.
- --chain_index <int> --chain_size <int> --chain_ir <string> can be used in conjunction to specify the location of the target device in the daisy chain.
  - --chain_index <int> is used to specify the position of the target device in the daisy chain. For example: --chain_index n indicates selecting the nth device, where n refers to the (n+1)th device.
  - --chain_size <int> is used to indicate the total number of devices in the daisy chain. For example: --chain_size n indicates the total number of devices in the daisy chain is n.
  - --chain_ir <string> is used to specify the IR length in the JTAG state machine in this daisy chain. For example: --chain_ir 8,8 indicates that the IR length for both devices in this daisy chain is 8 (defaults to 8 if not specified).
- --location is used to specify the USB port where the target device is located, and the priority is higher than the UID.
- --uid is used to specify the USB Cable used by the target device.
- --cable-index <int> is used to specify the USB Cable type.
- --fsFile <bitstream.fs> is used to specify the bitstream file path.
- --run <int> and --operation_index are used to specify the execution mode.

- --device <GWxx-x> is used to specify the target device name, case-sensitive.

4.7 Configure SRAM

FPGA SRAM can be configured by specifying a bitstream file, the target device, and the SRAM configuration mode. For example:

Configure SRAM Program; the "SRAM Program" value in the --operation_index parameter is 2, so the operation command is as follows:

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

Among them, --cable-index and --location have default values when they are specified, and they can be omitted.

An example is as follows:

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

--cable-index 1

Figure 4-9 SRAM Configuration Example

```
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)
```

The parameter meanings are as follows:

- --device GW1N-4B is used to specify the target FPGA device, and the device name will be printed when scanning.
- --fsFile d:\bitstream.fs is used to specify the bitstream file d:\bitstream.fs.
- --cable-index 1 is used to specify the USB Cable as "Gowin USB Cable (FT2CH)".

Table 4-1 SRAM Function Configuration Index

Index	Operation	Description
0	Read Device Codes	Read the FPGA device codes, including IDCode, UserCode, and StatusCode.
1	Reprogram	Reboot the device.
2	SRAM Program	Configure the SRAM.
4	SRAM Program and Verify	Configure the SRAM and verify it.
16	SRAM Program JTAG 1149	JTAG raw data programming mode. CRC check, nor encrypted or compressed bitstream files are not supported.

Index	Operation	Description
17	SRAM Program,Verify JTAG 1149	Configure the SRAM in raw data mode and verify it.
24	SRAM Erase	Erase SRAM data.
73	Read User Code	Read the device User Code.
74	Read Status Register	Read the device Status Code.

4.8 Configure Embedded Flash of LittleBee Family

4.8.1 Configure Flash Only

An example of programming Flash is as follows:

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

Figure 4-10 Embedded Flash Programming of LittleBee Family Example

```
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)
```

Table 4-2 Function Configuration Index for LittleBee Family

Index	Operation	Description
5	embFlash Erase,Program	Erase the embedded Flash first, and then write the data.
6	embFlash Erase,Program,Verify	Erase the embedded Flash first, then write the data and verify it.
7	embFlash Erase Only	Erase the embedded Flash only
70	embFlash Background Erase,Program	Background programming; erase the embedded Flash first, and then write the data.
71	embFlash Background Erase,Program,Verify	Background programming; erase the embedded Flash first, then write the data and verify it.
72	embFlash Background Erase Only	Background programming; erase the embedded Flash only.

4.8.2 Configure Flash and UserFlash Initialization File

Use the parameter "--fiFile userflash.fi” to specify UserFlash initialization file, and configure UserFlash at the same time when programming Flash, for example:

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

4.9 Configure External SPI Flash

4.9.1 External Flash Programming of LittleBee Family, GW2A(R) and GW3A Devices

By configuring parameters, it is possible to program the external Flash of LittleBee family, GW2A(R), and GW3A devices. An example is shown below.

programmer_cli --run 36 --fsFile d:\bitstream.fs --device GW1N-4B --cable-index 1

Figure 4-11 External Flash Programming of LittleBee Family Example

```
G:\history\Programmer\bin>programmer_cli.exe -r 36 --fsFile E:\file\GW1N-4B\GW1N-4B.fs -d GW1N-4B --cable-index 1
 Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
 Target Device: GW1N-4B(0x1100381B)
 Operation "exFlash Erase,Program thru GAO-Bridge" for device#1...
Preparing data...: [##########################] 100%
 Loading GAO-Bridge...
Programming...: [##########################] 100%
 Gao-Bridge User Code is: 0x0000BCAA
 Status Code is: 0x0001F020
 Try to program spi-flash: 0xC84017!
 Erasing 36 sectors from address 0x000000 to 0x024000...
Programming Flash...: [##########################] 100%
 Program flash finished.
 Finished.
 Cost 5.09 second(s)
```

Table 4-3 Function Configuration Index for LittleBee Family and GW2A(R) Devices

Index	Operation	Description
8	exFlash Erase,Program	Erase the external Flash first, then write the data.
9	exFlash Erase,Program,Verify	Erase the external Flash first, then write the data and verify it.
10	exFlash Bulk Erase	Erase the external Flash data
11	exFlash Verify	Verify the external Flash data
12	exFlash Erase,Program in bscan	Use bscan mode to erase the external Flash first, then write the data.
13	exFlash Erase,Program,Verify in bscan	Use bscan mode to erase the external Flash first, then write the data and verify it.
14	exFlash Bulk Erase in bscan	Use bscan mode to erase the external Flash data
15	exFlash Verify in bscan	Use bscan mode to verify the external Flash data
31	exFlash C Bin Erase,Program	Erase the external Flash first, then write the RISC-V bin file to the external Flash.
32	exFlash C Bin Erase,Program,Verify	Erase the external Flash first, then write the RISC-V bin file to the external Flash and verify it.
36	exFlash Erase,Program thru	Use the GAO-Bridge mode to erase the

Index	Operation	Description
	GAO-Bridge	external Flash first, then write the data.
37	exFlash Erase,Program,Verify thru GAO-Bridge	Use the GAO-Bridge mode to erase the external Flash first, then write the data and verify it.
38	exFlash C Bin Erase,Program thru GAO-Bridge	Use the GAO-Bridge mode to erase the external Flash first, then write the RISC-V bin file.
39	exFlash C Bin Erase,Program,Verify thru GAO-Bridge	Use the GAO-Bridge mode to erase the external Flash first, then write the RISC-V bin file and verify it.

4.9.2 Sipped Flash Programming

Configuration of the sipped Flash can be classified into two categories according to the device characteristics:

- GW1AN-3A(B)/GW1AN-5A(B)
- GW2AN-9X/GW2AN-18X

By configuring parameters, it is possible to program the sipped Flash. An example is shown below.

programmer_cli --run 44 --fsFile d:\bitstream.fs --device GW2AN-9X --

Figure 4-12 Example for Sipped Flash Programming of GW2AN-9X Devices

```
G:\history\Programmer\bin>programmer_cli.exe -r 44 -d GW2AN-9X --fsFile E:\file\GW2AN-9X\GW2AN-9X.fs
 Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
 Target Device: GW2AN-9X(0x0000581B)
 Operation "sFlash Erase,Program" for device#1...
 Status Code is: 0x00026020
 Read Status after reinit: 0x00020230
 Read sFlash ID: 0xC86015
Programming...: [##########################] 100%
 Program sFlash finished.
 Reloading
 User Code is: 0x0000023F
 Status Code is: 0x00026020
 Finished.
 Cost 10.0 second(s)
```

Table 4-4 Device Function Configuration Index

Index	Operation	Description
44	sFlash Erase,Program	Erase the sipped Flash, then write the data.
45	sFlash Erase,Program,Verify	Erase the sipped Flash, then write the data and verify it.
46	sFlash Verify Only	Verify the sipped Flash data
47	sFlash Bulk Erase	Erase the sipped Flash data
48	sFlash Background Erase,Program	Background programming; erase the sipped Flash, then write the data.
49	sFlash Background Erase,Program,Verify	Background programming; erase the sipped Flash, then write the data and verify it.
50	sFlash Erase,Program,Verify	Use the GAO-Bridge mode to erase the

Index	Operation	Description
	thru GAO-Bridge	sipped Flash, then write the data and verify it.
63	sFlash Background Erase,Program,Verify thru GAO-Bridge	Establish a bridge from JTAG to SPI, and in the background programming mode, erase the sipped Flash, then write the data and verify it.
64	sFlash Bulk Erase in bscan	Use bscan mode to erase the sipped Flash data
65	sFlash Erase,Program in bscan	Use bscan mode to erase the sipped Flash data, then write the data.

GW1AN-1SA/GW1AN-1C/GW2ANR-18C/GW2AN-55C

By configuring parameters, it is possible to program the sipped Flash. An example is shown below.

programmer_cli --run 36 --fsFile d:\bitstream.fs --device GW2ANR-18C --cable-index 1

Figure 4-13 Example for Sipped Flash Programming of GW2AN-18C Devices

```
E:\release\1.9.12\programmer1.9.12(2558).Win64\Programmer\bin>programmer_cli --run 36 --fsFile d:\bitstream.fs --device
GW2ANR-18C --cable-index 1
 Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
 Target Device: GW2ANR-18C(0x0000081B)
 Operation "exFlash Erase,Program thru GAO-Bridge" for device#1...
Preparing data...: [##########################] 100%
 Loading GAO-Bridge...
Programming...: [##########################] 100%
 Gao-Bridge User Code is: 0x0000B8DF
 Status Code is: 0x00006020
 Try to program spi-flash: 0xF83217!
 Erasing 141 sectors from address 0x000000 to 0x08C000...
Programming Flash...: [##########################] 100%
 Program flash finished.
 Finished.
 Cost 16.63 second(s)
```

Table 4-5 Device Function Configuration Index

Index	Operation	Description
8	exFlash Erase,Program	Erase the sipped Flash, then write the data
9	exFlash Erase,Program,Verify	Erase the sipped Flash, then write the data and verify it.
10	exFlash Bulk Erase	Erase the sipped Flash data
11	exFlash Verify	Verify the sipped Flash data
12	exFlash Erase,Program in bscan	Use bscan mode to first erase the sipped Flash, then write data.
13	exFlash Erase,Program,Verify in bscan	Use bscan mode to first erase the sipped Flash, then write data and verify it.
14	exFlash Bulk Erase in bscan	Use bscan mode to erase the sipped Flash
15	exFlash Verify in bscan	Use bscan mode to verify the sipped Flash

Index	Operation	Description
31	exFlash C Bin Erase,Program	Erase the sipped Flash first, then write the RISC-V binary file into the sipped Flash.
32	exFlash C Bin Erase,Program,Verify	Erase the sipped Flash first, then write the RISC-V binary file into the sipped Flash and verify it.
36	exFlash Erase,Program thru GAO-Bridge	Use GAO-Bridge mode to first erase the sipped Flash, then write the data.
37	exFlash Erase,Program,Verify thru GAO-Bridge	Use GAO-Bridge mode to first erase the sipped Flash, then write the data and verify it.
38	exFlash C Bin Erase,Program thru GAO-Bridge	Use GAO-Bridge mode to first erase the sipped Flash, then write the RISC-V binary file.
39	exFlash C Bin Erase,Program,Verify thru GAO-Bridge	Use GAO-Bridge mode to first erase the sipped Flash, then write the RISC-V binary file and verify it.

4.9.3 External Flash Programming of Arora V Devices

By configuring parameters, it is possible to program the external Flash of Aora V devices. An example is shown below.

programmer_cli --run 53 --fsFile d:\bitstream.fs --device GW5A-25A

Figure 4-14 External Flash Programming of Arora V Devices Example

```
G:\history\window\x64\1.9.11.01 Alpha\20250310\programmer1.9.11.01 Alpha.Win64\Programmer\bin>programmer_cli.exe -d GW5A
-25A -r 53 --fsFile E:\file\GW5A-25\GW5AT-25A.fs
 Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
 Target Device: GW5A-25A(0x0001281B)
 Operation "exFlash Erase,Program 5A" for device#1...
 Status Code is: 0x36020238
 Try to program spi flash: 0xC84017!
Reading BitStream File...: [##########################] 100%
 Programmning Flash starts from 0x000000.
Programming...: [##########################] 100%
 Programmning Flash ends from 0x0B5300.
 Program sFlash finished.
 The device is reloading, please check later...
 Finished.
 Cost 17.62 second(s)
```

Table 4-6 Function Configuration Index for Arora V Devices

Index	Operation	Description
52	exFlash Bulk Erase Arora V	Erase the external Flash
53	exFlash Erase,Program Arora V	Erase the external Flash first, then write the data.
54	exFlash Erase,Program,Verify Arora V	Erase the external Flash first, then write the data and verify.
55	exFlash C Bin Erase,Program Arora V	Erase the external Flash first, then write the RISC-V bin file.
56	exFlash C Bin Erase,Program,Verify Arora V	Erase the external l Flash first, then write the RISC-V bin file and verify.
66	exFlash Verify Arora V	Verify the external Flash data

Index	Operation	Description
67	exFlash Verify thru GAO-Bridge Arora V	Use the GAO-Bridge mode to verify the external Flash data.
68	exFlash Erase,Program thru GAO-Bridge Arora V	Use the GAO-Bridge mode to erase the external Flash and write the data.
69	exFlash Erase,Program,Verify thru GAO-Bridge Arora V	Use the GAO-Bridge mode to erase the external Flash, write the data and verify.

4.10 Configure I2C

Configure the FPGA using the I2C configuration mode. An example is shown below:

programmer_cli --run 57 --fsFile d:\bitstream.fs --device GW2AN-9X

Figure 4-15 I2C Configuration Example

```
G:\history\window\x64\1.9.11.01 Alpha\20250310\programmer1.9.11.01 Alpha.Win64\Programmer\bin>programmer_cli.exe -d GW2A
N-9X -r 57 --fsFile E:\file\GW2AN-9X\GW2AN-9X.fs
 Target Cable: Gowin USB Cable(FT2CH)/0/None/nul1@2.5MHz
 Target Device: GW2AN-9X(0x0000581B)
 Operation "I2C Program SRAM" for device#1...
Programming: [##########################] 100%
 Use fixed frequency 0.4MHz for I2C interface
 Finished.
 Cost 28.45 second(s)
```

Table 4-7 Function Configuration Index for I2C

Index	Operation	Description
57	I2C Program SRAM	Configure SRAM
58	I2C Program Flash	Configure Flash
59	I2C Erase Flash Only	Erase Flash data

4.11 MSPI 2nd Boot Address Operation

By configuring the parameters, users can perform read and write operations on the Golden Image address of Arora V devices. The address range is from 0x000000 to 0xF00000.

--read-golden-addr: Read the Golden Image address

--save-golden-addr: Write the Golden Image address

--golden-addr: The Golden Image address

A read example is shown below:

programmer_cli --read-golden-addr -d GW5A-25A

Figure 4-16 MSPI 2nd Boot Address Operation Example

```
C:\Users\zhanqing\Desktop\programmer1.9.11.02.Win64\Programmer\bin>programmer_cli.exe --read-golden-addr -d GW5A-25A
 Mspi 2nd address is 0xB00000
 Finished
 Cost 0.55 second(s)
```

A write example is shown below:

programmer_cli --save-golden-addr –golden-addr 0xF00000 -d GW5A-25A

Figure 4-17 MSPI 2nd Boot Address Operation Example

```
E:\release\1.9.12\programmer1.9.12(2558).Win64\Programmer\bin>programmer_cli.exe -d GW5A-25A --save-golden-addr --golden
-addr 0xF00000
 Write Success
 Finished
 Cost 2.69 second(s)
```

4.12 Remote Configuration and Programming

By configuring the parameters, remote configuration and programming can be performed. For the required setup, refer to Section 3.14 Remote Configuration and Programming.

--gvc: Set the remote IP address and port (format: xxx.xxx.xxx.xxx:xxxx)

For example:

programmer_cli --gvc 192.168.31.92:36548 -d GW5A-25A

Figure 4-18 Remote Configuration and Programming Example

```
C:\Users\zhanqing\Desktop\programmer1.9.11.02.Win64\Programmer\bin>programmer_cli.exe --gvc 192.168.31.92:36548 -d GW5A-
25A
 Connected to gvc server 192.168.31.92:36548
 Utilizing GVC for remote debugging
 Target Cable: Gowin USB Cable(FT2CH)/0/None/null@2.5MHz
 Target Device: GW5A-25A(0x0001281B)
 Operation "Read Device Codes" for device#1...
 ID Code is: 0x0001281B
 User Code is: 0x00000000
 Status Code is: 0x16020238
 Finished.
 Disonnected from remote server
 Cost 1.12 second(s)
```

4.13 Device Security

By configuring the parameters, the decryption key for the bitstream file can be written into the FPGA. This function supports both plaintext key writing and encrypted key file writing.

--key: Plaintext product key

--keyread: Read the plaintext product key

--keywrite: Write the plaintext product key

--keywritefile: Write the product key from an encrypted file

--keyFile: Encrypted key file

For example:

programmer_cli --keyread -d GW5A-25A

Figure 4-19 Device Security Example

```
C:\Users\zhanqing\Desktop\programmer1.9.11.02.Win64\Programmer\bin>programmer_cli.exe --keyread -d GW5A-25A
 Key1 Sel.
 Key Reading.
 Key=0x00000000000000000000000000003FFFF
Value: 0
 Device Identifier (DNA) is 0x000000000000000F.
 Cost 1.39 second(s)
```

4.14 Create an SVF File

By configuring the parameters, the tool supports generating an SVF file from an FS file.

--svfcreate: Create an SVF file

--svf_frequency: Specify the SVF file frequency

For example:

programmer_cli.exe --svfcreate --svf_frequency 2.5 -r 2 --fsFile E:\file\GW5A-25\GW5AT-25A.fs -d GW5A-25A

Figure 4-20 Creating an SVF File Example

```
C:\Users\zhanqing\Desktop\programmer1.9.11.02.Win64\Programmer\bin>programmer_cli.exe --svfcreate --svf_frequency 2.5 -r
 2 --fsFile E:\file\GW5A-25\GW5AT-25A.fs -d GW5A-25A
Creating : [##########################] 100%
 SVF File Created: "C:\Users\zhanqing\Desktop\programmer1.9.11.02.Win64\Programmer\bin\output.svf"
 Cost 0.45 second(s)
```

4.15 File Conversion Function

By configuring the parameters, mutual conversion between different file formats can be performed.

--filestransform: Execute the file conversion operation

--files: Specify the files to be converted; multiple files are separated by commas

For example:

programmer_cli.exe --filestransform 2 --files E:\file\GW5A-25\GW5AT-25A.fs

Figure 4-21 File Conversion Function Example

```
C:\Users\zhanqing\Desktop\programmer1.9.11.02.Win64\Programmer\bin>programmer_cli.exe --filestransform 2 --files E:\file
\GW5A-25\GW5AT-25A.fs
 HEX file saved to E:\file\GW5A-25\GW5AT-25.hex.
 Cost 1.41 second(s)
```

4.16 I2C Slave Address Operation

By configuring the parameters, read and write operations on the I2C Slave address can be performed. Currently, this operation only supports devices GW2AN-18X and GW2AN-9X.

--read-otp-addr: Read the I2C Slave address

--save-otp-addr: Write the I2C Slave address

--i2c-addr: Specify the I2C Slave address

For example:

programmer_cli.exe –read-otp-addr –d GW2AN-9X

Figure 4-22 I2C Slave Address Operation Example

```
C:\Users\zhanqing\Desktop\programmer1.9.11.02.Win64\Programmer\bin>programmer_cli.exe --read-otp-addr -d GW2AN-9X
 Current device status code is: 0x00020638
 Reading I2C OTP Address.
 I2C OTP Address is 7'b1010001.
 Cost 0.63 second(s)
```

GOWIN

PROGRAMMING FOR THE FUTURE