Gowin Programmer

User Guide

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.

Contents

List of Figures

List of Tables

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. 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

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

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 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 and jtagserver_lpt.exe\ jtagserver_u2x.exe belong to GAO tool chain.

2.1.5 Cable5.uid.up.exe

Gowin USB Cable Version 5.0 UID configuration tool

2.1.6 Gowin_USB_Cable_Installer.sh and Makefile

Used to modify the privileges of a cable in Linux system

2.2 Install and Start 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
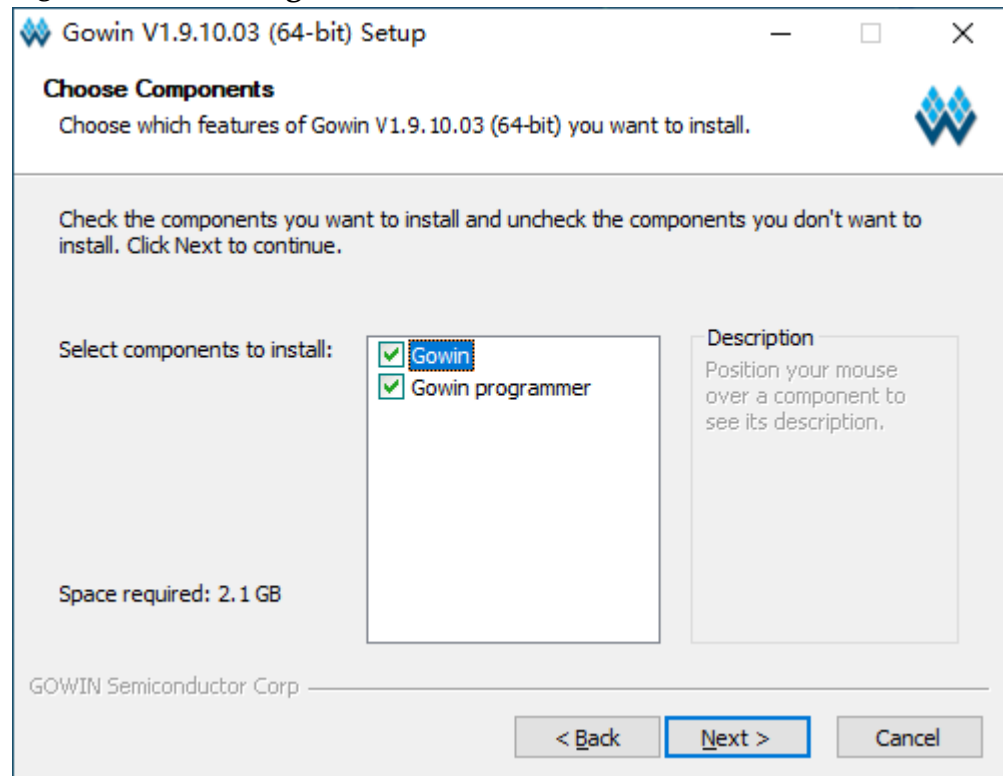
2.2.2 The Second Method to Install

Download Gowin Programmer installation package at official website: https://www.gowinsemi.com/en/support/download_eda/. After unzipping the installation package, open directory of programmer2\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 Start Programmer Tool Chain

- After installation, the .exe file is under \x.x\Programmer\bin\; Double-click on programmer.exe to start the software, as shown in Figure 2-3.

Figure 2-3 Start programmer.exe
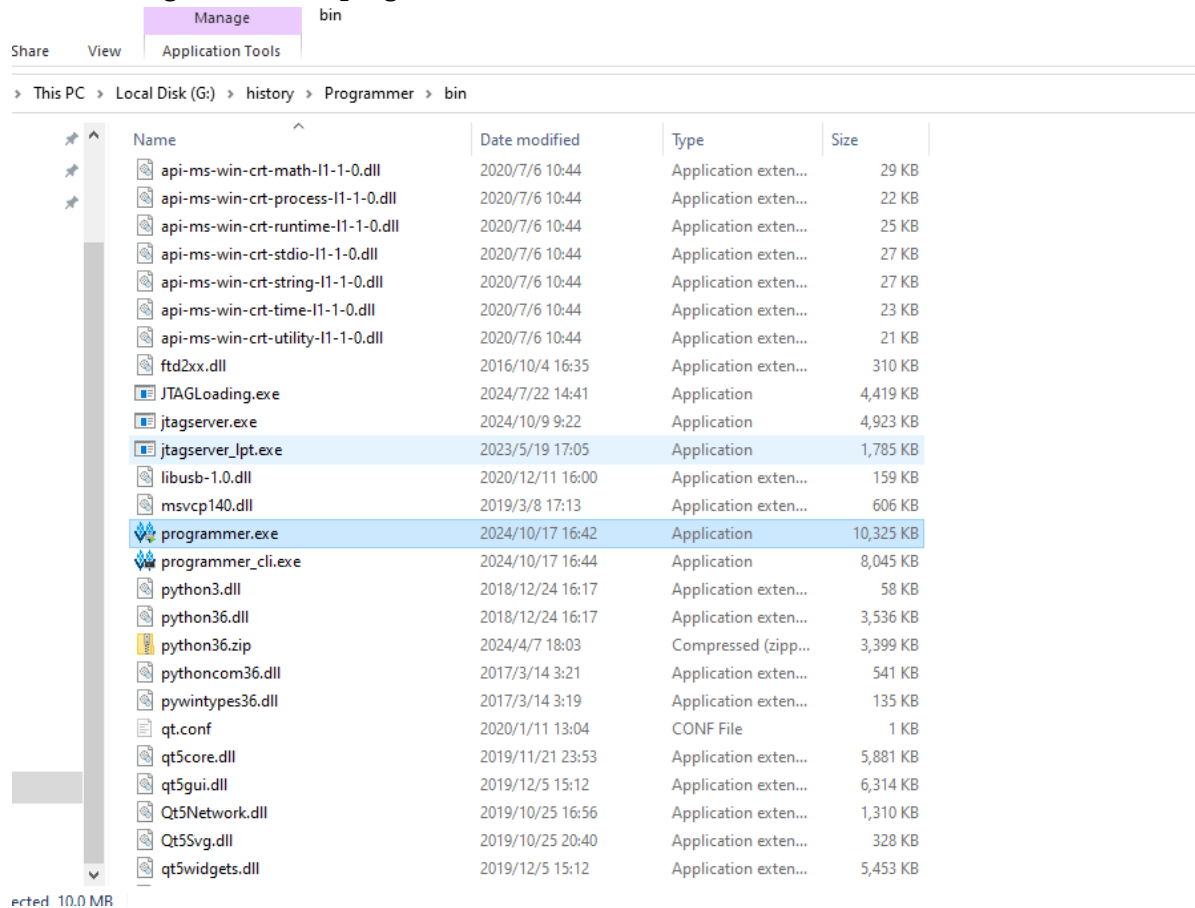
- For command line software, please open it in CMD; for example, open programmer_cli.exe.

Figure 2-4 Start programmer_cli.exe

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

Figure 2-5 Start Programmer Using Shortcut Key
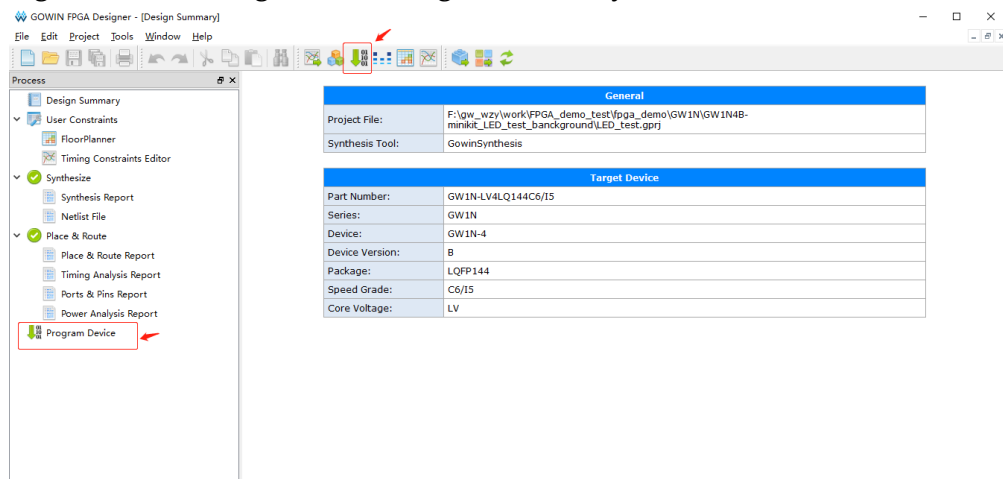
2.3 Software Interface

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

Figure 2-6 Programmer Interface
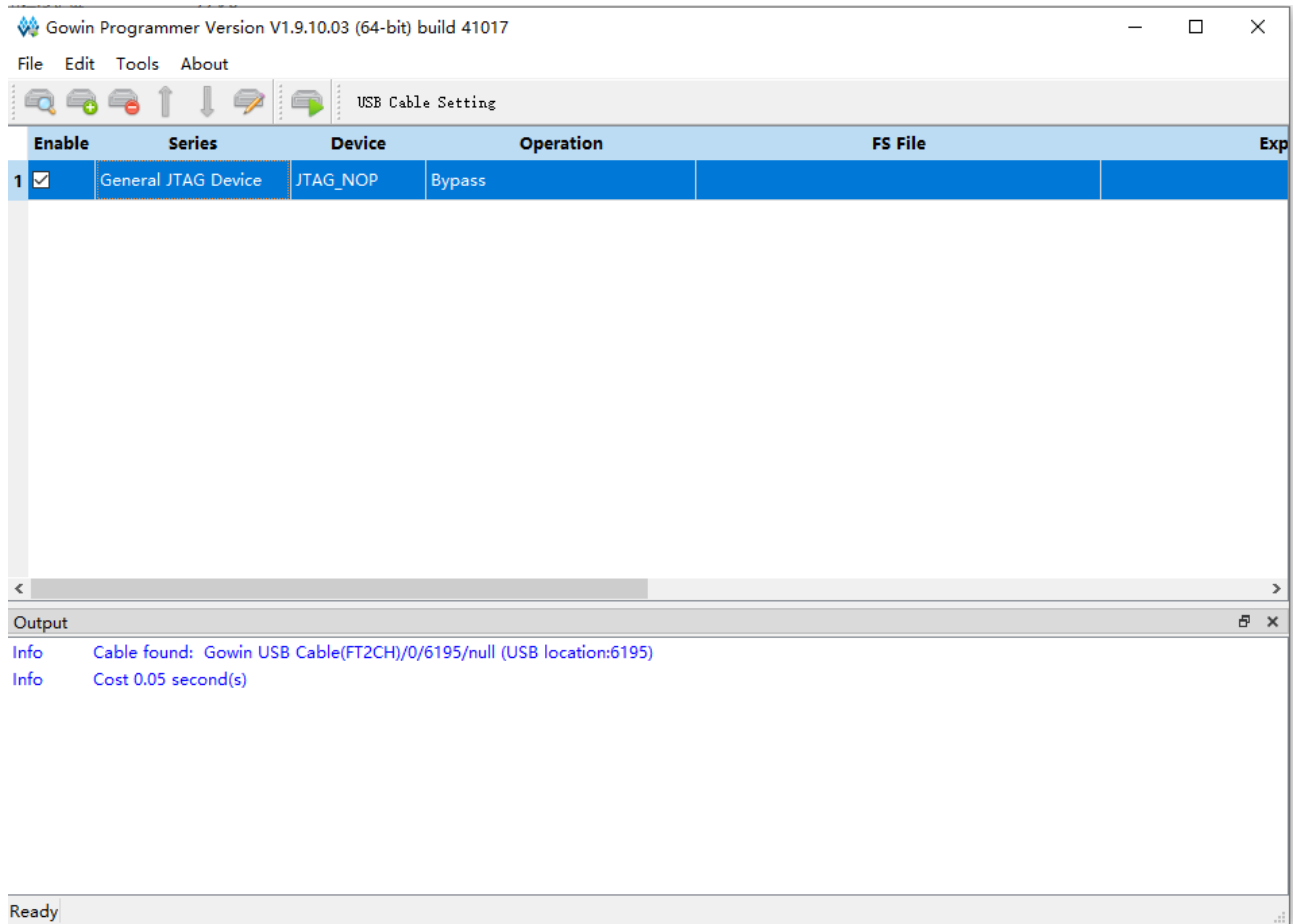
In the device table, all the devices that will be programmed in daisy chain are displayed through automatically scanning or manual configuration. Each row of the table represents a device, which can be programmed or not by selecting Enable column.

The device table includes Enable, Family, Device, Operation, FS File, Checksum, 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 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, and the row where the device is located will not be editable.

2.4 Software Version

Gowin Programmer and Gowin IDE 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

```
 ⊠   fzq@localhost:~/software/application/cable_linux_privileges_20230417  _  □

 File  Edit  View  Search  Terminal  Help
[fzq@localhost cable_linux_privileges_20230417]$ sudo make

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

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

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

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 privilege, and give the privilege 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 2.5MHz.

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 data 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". Three types of cables are supported currently: Gowin USB Cable (FTDI), Gowin USB Cable (GWU2X) and LPT.

1. Gowin USB Cable (FT2CH) is as shown in Figure 3-1.

    ● Cable: Gowin USB Cable (FT2CH)

    ● Port: The first available port will be selected by default. The last character A represents the channel number of programmer. There are three channels: S, A and B.

    ● Frequency: JTAG with 2MHz, 2.5MHz,15MHz,10MHz, and the default is 2.5 MHz.

Figure 3-1 Gowin USB Cable (FT2CH)
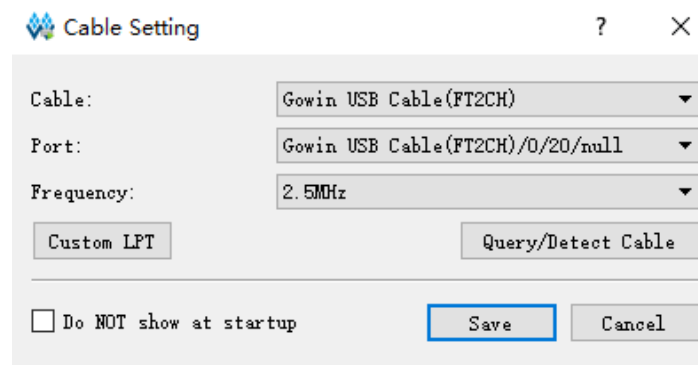
2. LPT is as shown in Figure 3-2.

   ● Cable: 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
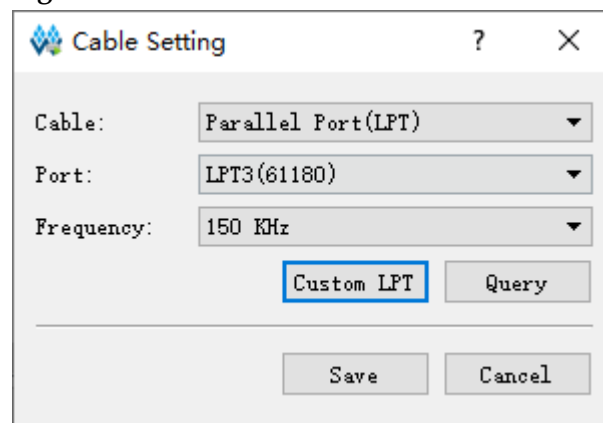
3. Gowin USB Cable (GWU2X) is as shown in

   ● Cable: Gowin USB Cable(GWU2X)

   ● Port: The first available port will be selected by default. The last character A represents the channel number of programmer. There are three channels: S, A and B

   ● Frequency: 1.33MHz by default

Figure 3-3 Gowin USB Cable (GWU2X)
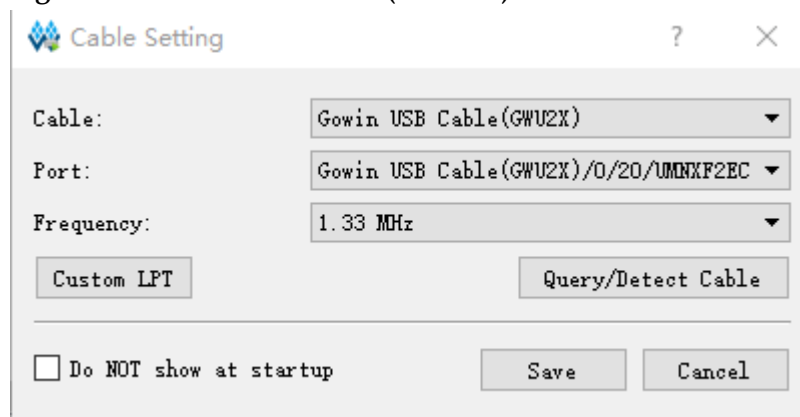
3.2 Scan Daisy Chain

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 3-4.

Figure 3-4 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 Device

1. Select "Edit > Add Device" or click " " in menu to add a new device.

2. Click "Family" to select the device family 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 Remove Device

Select the device row and remove the device by clicking "Edit >

Remove Device" or the " ⊝ ".

3.3.3 Modify Device Position in Chain

Select the device row and modify the device position in the chain by

clicking "Edit > 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 3-5.

Figure 3-5 Device Configuration Interface
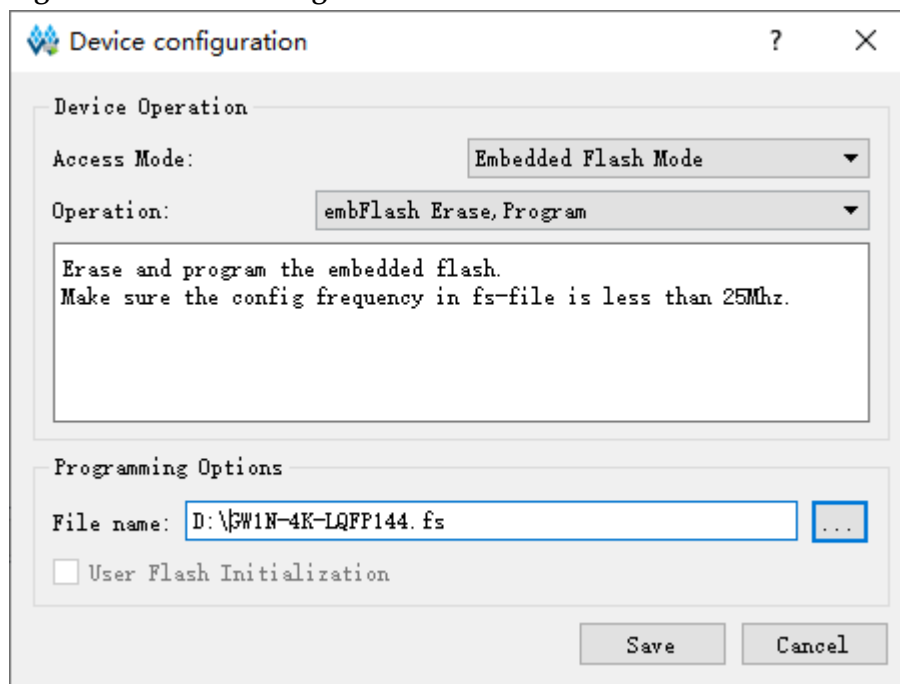
- Access Mode: Select programming mode.
- Operation: Select programming operation, see Table 3-1 for details.
- Instruction Register Length: When the device is selected as JTAG-NOP, select instruction register length.
- Programming File: Select programming data file.
- Device: When the programming mode 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, User Code, Status Code.
	Read User Code	Read Device User Code
	Read Status Register	Read Device Status
	Reprogram	–
	SRAM Erase	Erase SRAM
	SRAM Program	Configure the bitstream file to FPGA SRAM
	SRAM program and Verify	Write data to SRAM and verify
Embedded Flash Mode	embFlash Erase, Program	Erase embFlash, then write data.
	embFlash Erase, Program, Verify	Erase embFlash, then write data and verify.
	embFlash Erase Only	Erase embFlash only
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.
	exFlash Program Without Erasure	Write data to external Flash without erasure
	exFlash Bulk Erase	Erase external Flash
	exFlash Verify	Verify external Flash
	exFlash Erase, Program in bscan	Erase external Flash and write data to external Flash in bscan
	exFlash Erase, Program, Verify in bscan	Erase external Flash, write data to external flash and verify in bscan.
	exFlash Verify in bscan	–
	exFlash Program in bscan without erasure.	Write data to external Flash without erasure in bscan.
	exFlash Bulk Erase in bscan	Verify external Flash in bscan
	exFlash C Bin Erase, Program	Erase external Flash, then write RISC-V bin files to external Flash.
	exFlash C Bin Erase, Program, Verify	Erase external Flash, then write RISC-V bin files to external Flash and verify.
	exFlash C Bin Program	Write RISC-V bin files to external Flash
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

Note!

GW2A/GW2AR series of chips do not have embedded Flash and do not support this mode.

3.4.1 SRAM Configuration

1. Select the device row and open the Device Configuration dialog by clicking "Edit > 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 as required.

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 device (JTAG-NOP) only supports Bypass.

3.4.2 Embedded Flash Configuration of LittleBee Series of FPGA Products

The GW1N/GW1NZ series of 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 operation in "Operation" drop-down list as required;

4. Programming File: Select programming bitstream file.

5. Click "Save" to complete the configuration.

3.4.3 External Flash 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" in "Access Mode" drop-down list.

3. Select operation in "Operation" as required.

4. If "exFlash Program" selected in "Operation", the corresponding programming bitstream file needs to be selected in "Programming File".

5. For external Flash part number, LittleBee and Arora family support SPI Flash with read commands of 0x03 or 0x0B.

6. If there is no flash selected in the menu, please select Generic Flash to try to program automatically.

7. Select the initial address of the external Flash. Currently, the default is 0x000000.

8. Click "Save" to complete the configuration.

3.4.4 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 operation in "Operation" drop-down list as required.

4. If "Slave SPI Program SRAM" selected in "Operation", the corresponding programming bitstream file needs to be selected in "Programming File".

5. 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 3-6 I/O State Editor
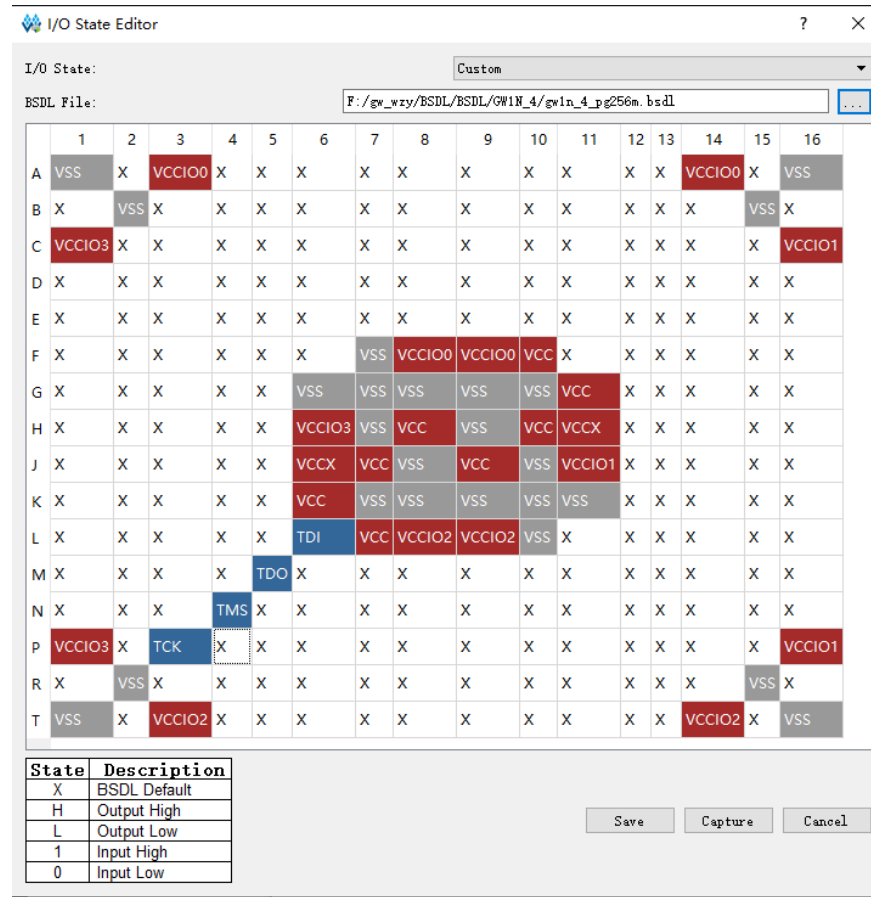
3.6 Device Security

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 3-7.

Figure 3-7 Security Configuration
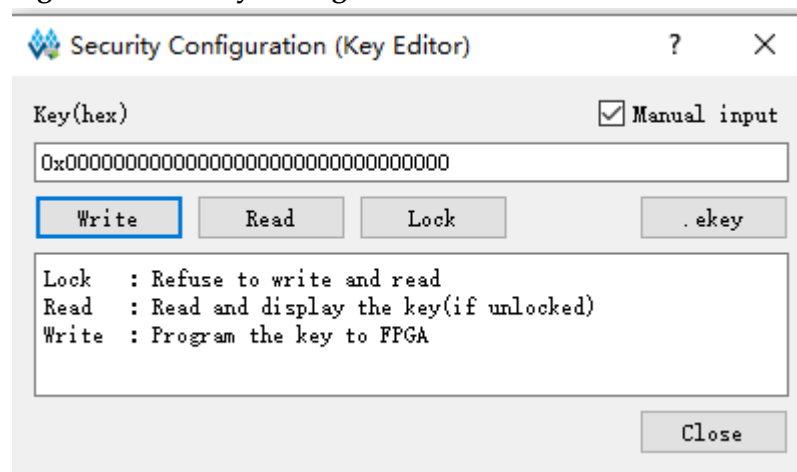
- Manual input: Input the key in plaintext or not.
- .ekey: Open the key file.

- 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 verification is successful, the key will be locked inside the FPGA.

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 "Design > Run" 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. Currently, only GW1N-4 SVF file creation 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 "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 3-8.

4. Click "OK" to complete the creation of the SVF file.

Figure 3-8 Create SVF File
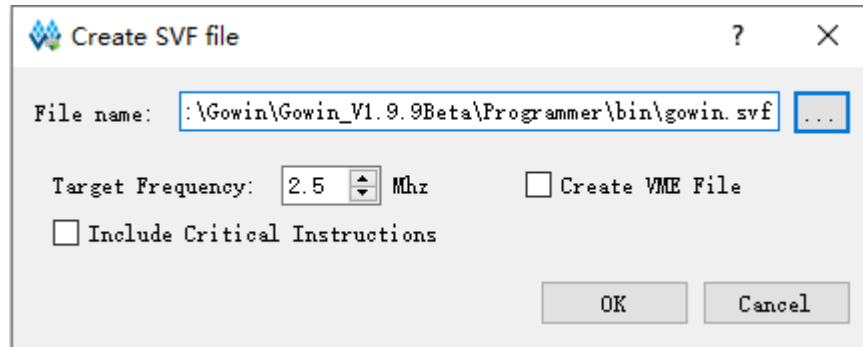
3.9 I2C Slave Address Operation

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 3-9 I2C Slave Address Operation
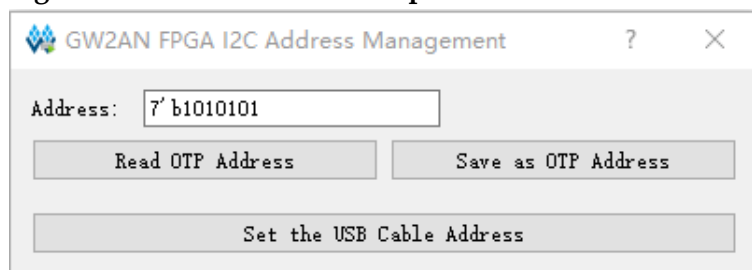
- Read OTP Address: Read I2C slave address.

- 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, ranging from 0x000000 to 0xF00000. Currently, this operation is only supported on GW5A-25 A devices.
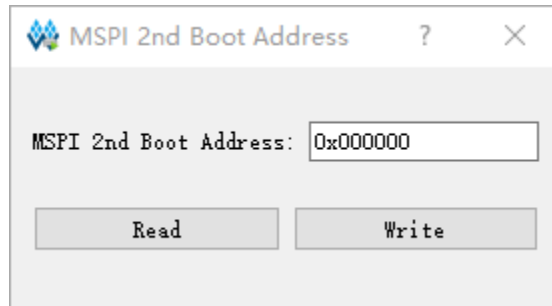
- Read: Read the MSPI 2nd boot address
- Write: Write the specified MSPI 2nd boot address to the FPGA

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 Viewer" or finding the status code display line, right-click "Analyze Status Code," and you can open the "Status Code Analyzer" dialog box, as shown in Figure 3-11 below.

Figure 3-11 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
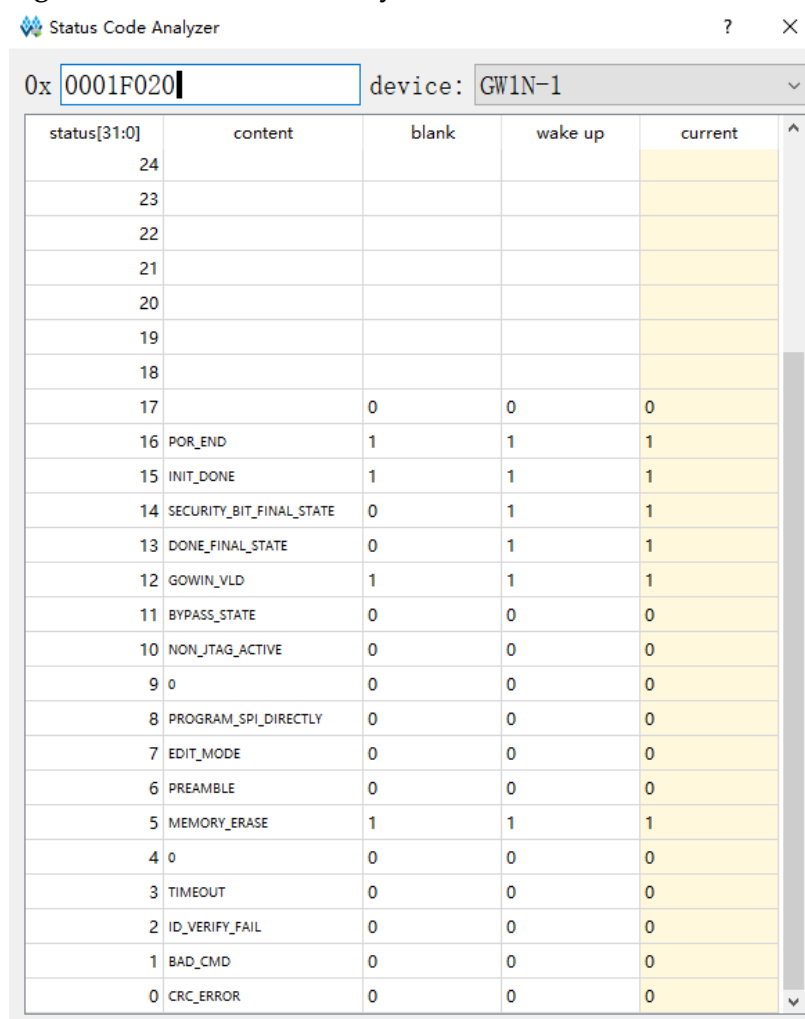
First, select the appropriate development board (you can select the board in the device list), then enter the status code. The results for each bit of the status code will be output in the list below.

3.12 User Flash Initialization

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 3-12 .

Figure 3-12 User Flash Initialization
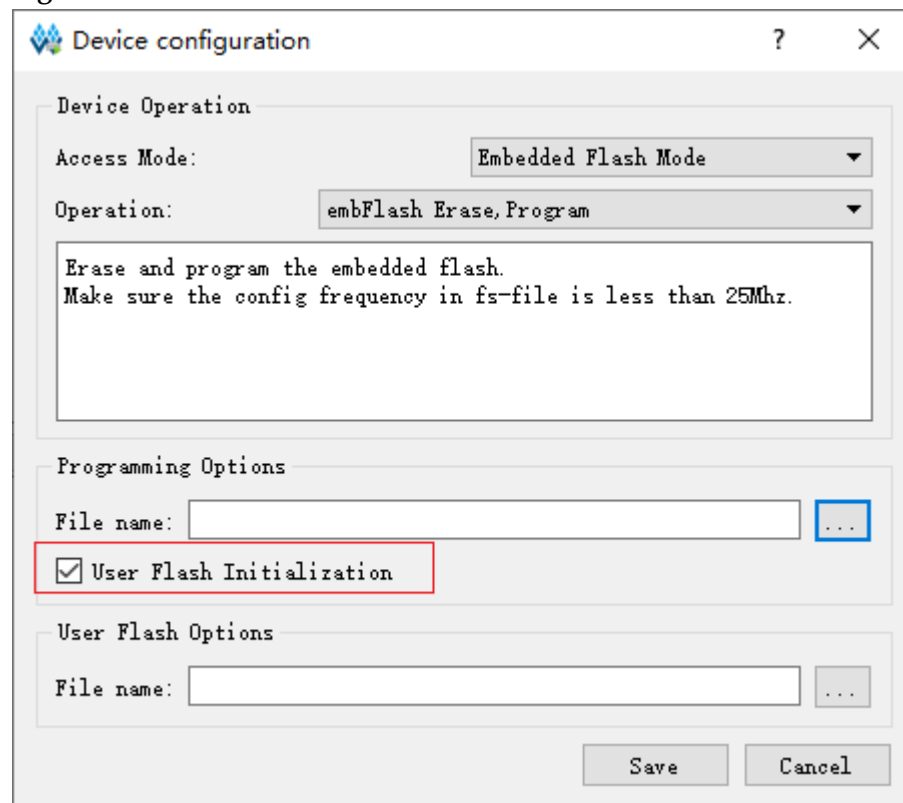
Table 3-2 User Flash Reference

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

Series	Device	Flash Type	Address	Data Width
	GW1NR-4B			
	GW1NR-9	FLASH608K	304* 64	
GW1NS	GW1NS-2	FLASH128K	32786	
	GW1NS-2C			
GW1NSR	GW1NSR-2	FLASH128K	32786	
	GW1NSR-2C			
GW1NZ	GW1NZ-1	FLASH64KZ	32* 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 3-13 Adjustment of JTAG State Machine Shift-IR Width
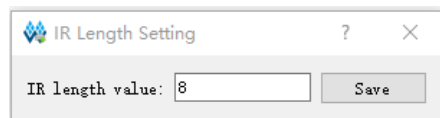
- IR length value: The width value of TAG state machine shift-ir.
- Save: Writes the selected value into the FPGA.

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 3-14.

Figure 3-14 jtagserver Location
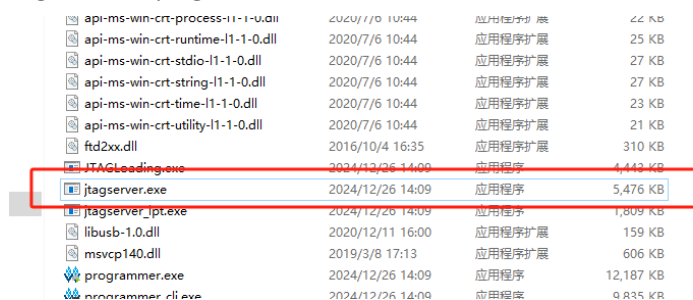
Figure 3-15 jtagserver in Running State
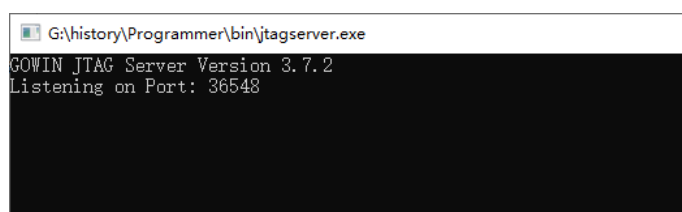
jtagserver listens on a specified port to support client connections. The default port is 36548. If you need to use a different port, navigate to the data folder within the same directory as the programmer, locate the config folder, and modify the port in the gvc.ini file.

Figure 3-16 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 3-18.

Figure 3-17Add GVC Server Option
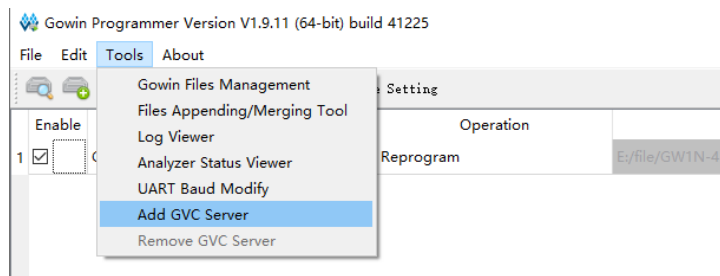
Figure 3-18 GVC
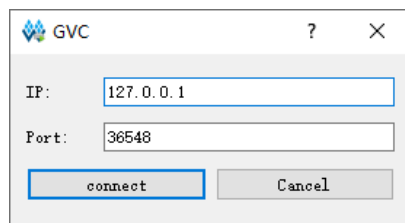
- IP: IP address of the remote server.
- Port: Port on which the remote server is listening.
- Connect: Connect to the remote server.
- Cancel: Cancel the operation and exit the page.

After clicking "Connect," if the connection is successful, a success screen as shown in Figure 3-19 will appear.

Figure 3-19 Connection Succeed
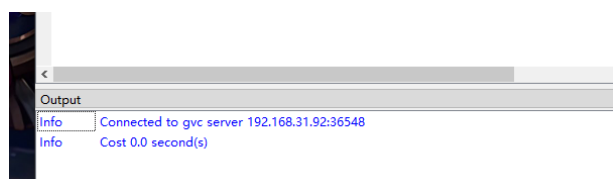
If the connection fails, an error screen as shown in Figure 3-20 will appear.

Figure 3-20 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.

 If the connection fails, an error screen as shown in Figure 3-21will appear.

Figure 3-21 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 3-22 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 GW1AN-1C	'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'

Device	Function
	'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 GW1N-4D GW1N-4B GW1N-9 GW1N-9C GW1NZ-1 GW1NZ-1C GW1NZ-2B GW1NZ-2C GW1NS-4 GW1NS-4C GW1NSR-4 GW1NSR-4C GW1NRF-4B GW1NR-2 GW1NR-2B GW1NR-2C GW1NR-4D GW1NR-4B GW1NR-9 GW1NR-9C	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" 'SRAM Program' 'SRAM Program JTAG 1149' 'embFlash Background Erase Only ' 'embFlash Background Erase,Program' 'embFlash Background Erase,Program,Verify' 'embFlash Erase Only'、'embFlash Erase,Program' 'embFlash Erase,Program,Verify' 'exFlash Erase,Program thru GAO-Bridge' 'exFlash Erase,Program,Verify thru GAO-Bridge' 'exFlash C Bin Erase,Program thru GAO-Bridge' 'exFlash C Bin Erase,Program,Verify thru GAO-Bridge' 'exFlash Bulk Erase thru GAO-Bridge' 'exFlash Verify thru GAO-Bridge' 'exFlash Erase,Program' 'exFlash Erase,Program,Verify' 'exFlash C Bin Erase,Program' 'exFlash C Bin Erase,Program,Verify' 'exFlash Background Erase,Program' 'exFlash Background Erase,Program,Verify' 'exFlash Bulk Erase' 'exFlash Verify'
GW1AN-9A GW2AN-9X GW2AN-18X	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" 'SRAM Program' 'SRAM Program JTAG 1149' 'sFlash Erase,Program,Verify thru GAO-Bridge' 'sFlash Background Erase,Program,Verify thru GAO-Bridge' 'sFlash Erase,Program' ' sFlash Erase,Program,Verify' 'sFlash Background Erase,Program' 'sFlash Background Erase,Program,Verify'
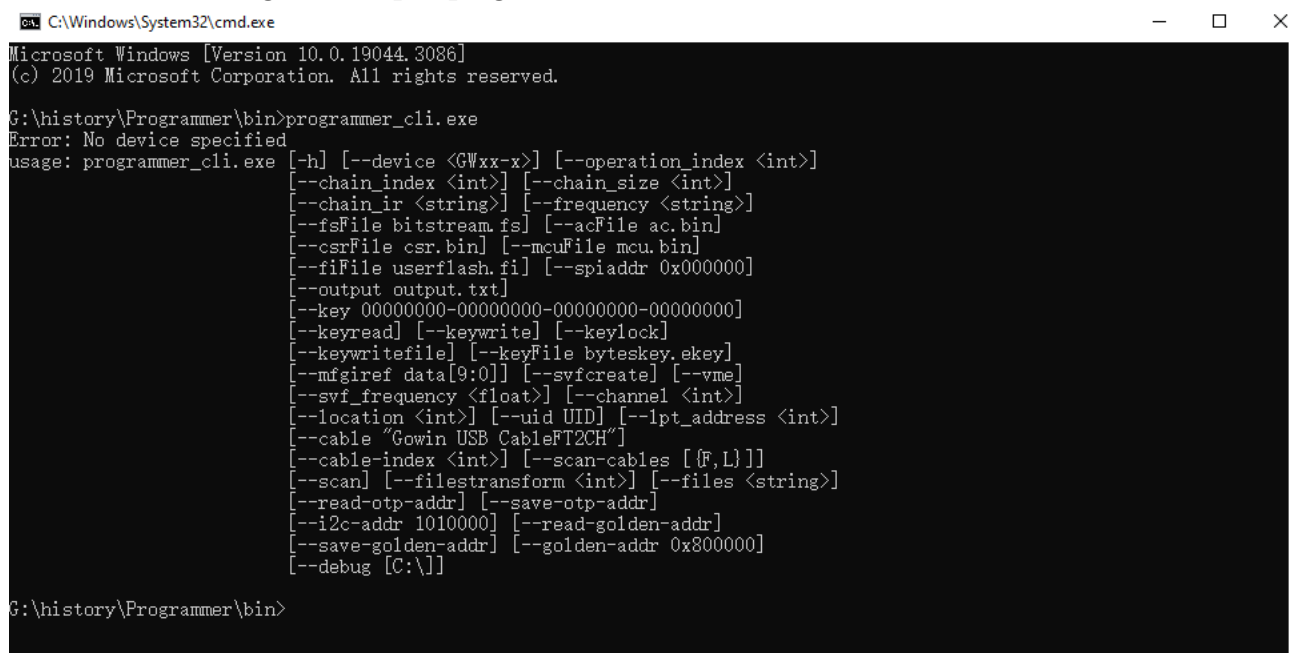
Device	Function
	'sFlash Bulk Erase'、'sFlash Verify Only'
GW2A-55 GW2A-55C GW2A-18 GW2A-18C GW2AN-55C GW2AR-18 GW2AR-18C GW2ANR-18C	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase"、'SRAM Program' 'SRAM Program JTAG 1149' 'exFlash Erase,Program thru GAO-Bridge' 'exFlash 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' 'exFlash C Bin Erase,Program' 'exFlash C Bin Erase,Program,Verify' 'exFlash Background Erase,Program' 'exFlash Background Erase,Program,Verify' 'exFlash Bulk Erase' 'exFlash Verify'
GW5AT-15A GW5AT-15B GW5AT-60B GW5AT-60ES GW5AT-75B GW5AT-75C GW5AT-138 GW5AT-138B GW5AT-138C GW5AST-138B GW5AST-138C GW5AS-25A GW5AS-25B GW5AS-138B GW5AS-138C GW5A-60B GW5A-25A GW5A-25B GW5A-138B GW5A-138C GW5AR-25A GW5AR-25B GW5ART-15A GW5ART-15B GW5ANT-15A GW5ANRT-15A	'Read Device Codes' 'Read User Code' 'Read Status Register' "Reprogram" "SRAM Erase" 'SRAM Program' 'SRAM Program JTAG 1149' 'exFlash Erase,Program thru GAO-Bridge 5A' 'exFlash Erase,Program,Verify thru GAO-Bridge 5A' 'exFlash Verify thru GAO-Bridge 5A' 'exFlash Erase,Program 5A' 'exFlash Erase,Program,Verify 5A' 'exFlash C Bin Erase,Program 5A' 'exFlash C Bin Erase,Program,Verify 5A' 'exFlash Bulk Erase 5A' 'exFlash Verify 5A'

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/GW20LCRZ (USB location:99091) (SN: GW20LCRZ)
 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 0 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)
 --cable-index <int>  Select a number for USB cable:
                      0: Gowin USB Cable(GWU2X);
                      1: Gowin USB Cable(FT2CH);
                      2: Parallel Port(LPT);
                      3: Digilent USB Device;
                      4: Unverified FT Device;
                      Higher priority than --cable, default cable-index is 0
```

4.4 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

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 USB Cable type.

- --fsFile <bitstream.fs> is used to specify bitstream file path.

- --run <int> and --operation_index are used to specify execution mode.

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

4.7 Configure SRAM

Specify a bitstream file, corresponding device and SRAM configuration mode to configure FPGA SRAM, 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 Example

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 USB Cable as "Gowin USB Cable (FT2CH)".

4.8 Configure Embedded Flash of LittleBee Family

4.8.1 Configure Flash

The --operation_index label for programming Flash is as follows:

5: embFlash Erase,Program;

6: embFlash Erase,Program,Verify;

7: embFlash Erase Only;

An example is as follows:

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

Figure 4-10 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)
```

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

The --operation_index label for programming external SPI Flash is as follows, and it is recommended to use "thru GAO-Bridge" type.

8 exFlash Erase,Program;

9: exFlash Erase,Program,Verify;

10: exFlash Bulk Erase;

11: exFlash Verify;

12: exFlash Erase,Program in bscan;

13: exFlash Erase,Program,Verify in bscan;

14: exFlash Bulk Erase in bscan;

15: exFlash Verify in bscan;

36: exFlash Erase,Program thru GAO-Bridge;

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;

An example is as follows:

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

Note!

For more information, see programmer_cli –help.