




# Gowin Programmer ユーザーガイド

SUG502-2.3J, 2025-08-29

## 著作権について(2025)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

**GOWIN**高云、、Gowin、及びLittleBeeは、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

## 免責事項

当社は、GOWINSEMI Terms and Conditions of Sale(GOWINSEMI取引条件)に規定されている内容を除き、(明示的か又は黙示的に拘わらず)いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

## バージョン履歴

日付	バージョン	説明
2017/04/06	1.0J	初版。
2017/08/06	1.1J	デバイスのプログラミングに関する内容を変更。
2019/10/28	1.2J	<ul style="list-style-type: none"> <li>● Slave SPIモードを追加。</li> <li>● SVFファイルの作成の情報を追加。</li> <li>● User Flashの初期化の情報を追加。</li> </ul>
2020/02/17	1.3J	Programmerのインストールと起動の情報を追加。
2022/05/30	1.4J	<ul style="list-style-type: none"> <li>● セクション「2.1 Programmerのツールチェーンの概要」を追加。</li> <li>● チャプター「4 Programmer_cliでのダウンロード手順」を追加。</li> </ul>
2023/06/08	1.5J	<ul style="list-style-type: none"> <li>● Linux OSにおけるケーブルの権限の構成に関する説明を追加。</li> <li>● Gowin USB Cable (GWU2X)の構成に関する説明を追加。</li> <li>● 表3-1 Deviceに対する操作の説明からSRAM Program JTAG 1149を削除。</li> <li>● ソフトウェアのスクリーンショットを更新。</li> </ul>
2024/05/09	1.6J	<ul style="list-style-type: none"> <li>● 「3.6セキュリティ」における説明を更新。</li> <li>● SRAMまたはFlashのコンフィギュレーション・プログラミングのコマンド形式の説明を更新。</li> <li>● 「3.9 I<sup>2</sup>C の Slave アドレスに対する操作」を追加。</li> </ul>
2024/06/28	1.7J	ステータスコード分析の説明を追加。
2024/10/25	1.8J	<ul style="list-style-type: none"> <li>● 「3.10 MSPI 2nd Bootアドレスに対する操作」を追加。</li> <li>● 一部のスクリーンショットを更新。</li> </ul>
2024/12/31	1.9J	「3.13 JTAG状態機械のShift-IR幅の調整」および「3.14 リモートのコンフィギュレーションおよびプログラミング」を追加。
2025/02/28	2.0J	「3.1 ダウンロードケーブルの設定」にピンの接続の注記を追加。
2025/03/28	2.1J	<ul style="list-style-type: none"> <li>● Gowin USB Cable(WINUSB)とUSB Debugger Aの説明を追加。</li> <li>● 「3.4.7 MCUモード」を追加。</li> <li>● 「3.4.8 セキュアFPGAモードの構成」を追加。</li> <li>● 「3.4.10 デバッグモード」を追加。</li> <li>● 「3.4.11 goConfigIPモードの構成」を追加。</li> <li>● 「3.4.12 バックグラウンド・プログラミング・モード」を追加。</li> <li>● 「4.9.3 Arora VデバイスのオフチップFlashのプログラミング」を追加。</li> </ul>
2025/04/30	2.2J	<ul style="list-style-type: none"> <li>● 「3.15 ファイル変換機能」を追加。</li> <li>● 「3.16 ビットストリーム・ファイル・マージ機能」を追加。</li> <li>● 「3.17 ログ表示機能」を追加。</li> <li>● 「3.18 ボーレート変更機能」を追加。</li> </ul>
2025/06/27	2.2.1J	「3.1 ダウンロードケーブルの設定」における説明を更新。
2025/08/29	2.3J	<ul style="list-style-type: none"> <li>● 「表3-1 Deviceに対する操作の説明」を更新。</li> <li>● 「4.11 MSPI 2nd Bootアドレスに対する操作」を追加。</li> <li>● 「4.12 リモートのコンフィギュレーションおよびプログラミング」を追加。</li> </ul>

日付	バージョン	説明
		<ul style="list-style-type: none"> <li>● 「4.13セキュリティ」を追加。</li> <li>● 「4.14 SVFファイルの作成」を追加。</li> <li>● 「4.15ファイル変換機能」を追加。</li> <li>● 「4.16 I<sup>2</sup>C Slaveアドレスに対する操作」を追加。</li> </ul>

# 目次

目次 .....	i
図一覧 .....	iv
表一覧 .....	vi
<b>1 本マニュアルについて .....</b>	<b>1</b>
1.1 マニュアルの内容 .....	1
1.2 関連ドキュメント .....	1
1.3 用語、略語 .....	1
1.4 テクニカル・サポートとフィードバック .....	3
<b>2 概要 .....</b>	<b>4</b>
2.1 Programmer のツールチェーンの概要 .....	4
2.1.1 programmer.exe .....	4
2.1.2 programmer_cli.exe .....	4
2.1.3 JTAGLoading.exe .....	4
2.1.4 jtagserver.exe .....	4
2.1.5 Gowin_USB_Cable_Installer.sh、Makefile .....	4
2.2 Programmer のツールチェーンのインストールと起動 .....	4
2.2.1 インストール方法 1 .....	4
2.2.2 インストール方法 2 .....	5
2.2.3 Programmer のツールチェーンの起動 .....	6
2.3 ソフトウェアの画面 .....	7
2.4 ソフトウェアバージョンの確認 .....	9
2.5 Linux OS におけるケーブルの権限の構成 .....	9
2.5.1 Makefile .....	9
2.5.2 Gowin_USB_Cable_Installer.sh .....	10
<b>3 GUI 版 Programmer でのダウンロード手順 .....</b>	<b>11</b>
3.1 ダウンロードケーブルの設定 .....	11
3.2 デバイスのデジタイゼーションのスキャン .....	14
3.3 デバイスのデジタイゼーションの構成 .....	15
3.3.1 デバイスの追加 .....	15

3.3.2 デバイスの取り外し.....	15
3.3.3 チェーン内のデバイス位置の変更.....	15
3.4 プログラミングの構成.....	15
3.4.1 SRAM モードの構成.....	23
3.4.2 LittleBee ファミリーFPGA での組み込み Flash モードの構成.....	23
3.4.3 オフチップ Flash モードの構成.....	23
3.4.4 GW2AN-9X/18X デバイスの内部 Flash モードの構成.....	24
3.4.5 Arora V デバイスのオフチップ Flash モードの構成.....	25
3.4.6 Slave SPI モード.....	25
3.4.7 MCU モード.....	26
3.4.8 セキュア FPGA モードの構成.....	26
3.4.9 I2C モード.....	27
3.4.10 デバッグモード.....	27
3.4.11 goConfigIP モードの構成.....	27
3.4.12 バックグラウンド・プログラミング・モード.....	28
3.5 ピンの状態の編集.....	28
3.6 セキュリティ.....	29
3.7 ダウンロード.....	30
3.8 SVF ファイルの作成.....	31
3.9 I2C Slave アドレスに対する操作.....	31
3.10 MSPI 2nd Boot アドレスに対する操作.....	32
3.11 ステータスコードの分析.....	32
3.12 User Flash の初期化とデータ保持.....	33
3.13 JTAG 状態機械の Shift-IR 幅の調整.....	36
3.14 リモートのコンフィギュレーションおよびプログラミング.....	36
3.15 ファイル変換機能.....	41
3.16 ビットストリーム・ファイル・マージ機能.....	41
3.17 ログ表示機能.....	42
3.18 ボーレート変更機能.....	43
<b>4 Programmer_cli でのダウンロード手順.....</b>	<b>44</b>
4.1 プレビューとヘルプ.....	44
4.2 USB Cable デバイスのスキャン.....	47
4.3 USB Cable の種類とポートの指定.....	48
4.4 USB Cable の場所または UID の指定.....	48
4.5 FPGA デバイスのスキャン.....	48
4.6 Programmer 実行モードの指定.....	49

4.7 SRAM のコンフィギュレーション .....	51
4.8 LittleBee ファミリーFPGA の組み込み Flash のプログラミング .....	52
4.8.1 Flash のみのプログラミング .....	52
4.8.2 Flash および UserFlash 初期化ファイルのプログラミング .....	53
4.9 オフチップ SPI Flash のプログラミング .....	54
4.9.1 LittleBee ファミリーと Arora ファミリー(Arora V を除く)のオフチップ SPI Flash のプログラ ミング .....	54
4.9.2 GW2AN-9X/18X デバイスの内部 Flash のプログラミング .....	55
4.9.3 Arora V デバイスのオフチップ Flash のプログラミング .....	56
4.10 I2C コンフィギュレーション .....	58
4.11MSPI 2nd Boot アドレスに対する操作.....	58
4.12 リモートのコンフィギュレーションおよびプログラミング .....	58
4.13 セキュリティ .....	59
4.14 SVF ファイルの作成.....	59
4.15 ファイル変換機能 .....	60
4.16 I <sup>2</sup> C Slave アドレスに対する操作 .....	60

## 図一覧

図 2-1 Gowin ソフトウェアのインストール際のコンポーネントの選択.....	5
図 2-2 Programmer のドライバーのインストール .....	5
図 2-3 programmer.exe の起動 .....	6
図 2-4 programmer_cli.exe の起動.....	7
図 2-5 Gowin ソフトウェア上の Programmer ショートカットキー.....	7
図 2-6 Programmer Main Window.....	8
図 2-7 バージョンの確認.....	9
図 2-8 ファイル一覧.....	9
図 2-9 Makefile のインストール例(一般ユーザー).....	10
図 2-10 Makefile のインストール例(root ユーザー).....	10
図 2-11 スクリプトのインストール例.....	10
図 3-1 Gowin USB Cable (FT2CH) .....	12
図 3-2 LPT .....	13
図 3-3 Gowin USB Cable (GWU2X).....	13
図 3-4 Gowin USB Cable (WINUSB) .....	14
図 3-5 USB Debugger A.....	14
図 3-6 Device Table .....	15
図 3-7 Device Configuration Interface.....	16
図 3-8 I/O State Editor.....	29
図 3-9 Security Configuration .....	29
図 3-10 Create SVF File.....	31
図 3-11 I <sup>2</sup> C Slave アドレスに対する操作.....	31
図 3-12 MSPI 2nd Boot アドレスに対する操作 .....	32
図 3-13 ステータスコード分析の画面 .....	33
図 3-14 User Flash Initialization .....	34
図 3-15 Retained User Flash Data.....	35
図 3-16 JTAG 状態機械の Shift-IR 幅の調整 .....	36
図 3-17 jtagserver.exe の位置 .....	36
図 3-18 jtagserver の起動状態.....	36



図 3-19 jtagserver の起動状態.....	37
図 3-20 Add GVC Server オプション.....	37
図 3-21 GVC 接続画面.....	37
図 3-22 接続成功画面.....	37
図 3-23 接続失敗画面.....	38
図 3-24 リモート操作際の出力.....	38
図 3-25 リモート・コンフィギュレーションの終了.....	38
図 3-26 ファイル変換画面.....	41
図 3-27 ビットストリーム・ファイル・マージ画面.....	42
図 3-28 ログ表示機能.....	43
図 3-29 ボーレートの変更.....	43
図 4-1 CMD で programmer_cli を開く.....	44
図 4-2 ヘルプ情報.....	45
図 4-3 USB Cable デバイスのスキャン.....	47
図 4-4 ヘルプ情報.....	48
図 4-5 ヘルプ情報.....	48
図 4-6 FPGA デバイスのスキャン.....	49
図 4-7 ヘルプ情報.....	49
図 4-8 ヘルプ情報.....	50
図 4-9 SRAM コンフィギュレーションの例.....	51
図 4-10 FlashFlash のみのプログラミングの例.....	52
図 4-11 オフチップ Flash のプログラミングの例.....	54
図 4-12 GW2AN-9X/18X デバイスの内部 Flash のプログラミングの例.....	55
図 4-13 Arora V デバイスのオフチップ Flash のプログラミングの例.....	56
図 4-14 I2C コンフィギュレーションの例.....	58
図 4-15 MSPI 2nd Boot アドレスに対する操作の例.....	58
図 4-16 リモートのコンフィギュレーションおよびプログラミングの例.....	59
図 4-17 セキュリティ.....	59
図 4-18 SVF ファイル作成の例.....	60
図 4-19 ファイル変換の例.....	60
図 4-20 I2C Slave アドレスに対する操作の例.....	60

## 表一覧

表 1-1 用語、略語 .....	2
表 3-1 Device に対する操作の説明 .....	16
表 3-2 User Flash の情報一覧 .....	35
表 3-3 各デバイスでサポートされるリモート・プログラミング/コンフィギュレーション機能 .....	38
表 4-1 SRAM の場合の機能構成インデックス .....	52
表 4-2 LittleBee ファミリーの場合の機能構成インデックス .....	52
表 4-3 LittleBee ファミリーと Arora ファミリーの場合の機能構成インデックス .....	54
表 4-4 GW2AN-9X/18X デバイスの場合の機能構成インデックス .....	55
表 4-5 Arora V デバイスの場合の機能構成インデックス .....	57
表 4-6 I2C の場合の機能構成インデックス .....	58

# 1 本マニュアルについて

## 1.1 マニュアルの内容

本マニュアルでは GOWIN セミコンダクターのプログラミングツールである Gowin Programmer の使用方法について説明します。本マニュアルに記載のスクリーンショットとサポートされる製品リストは、1.9.12 バージョンの場合のものです。ソフトウェアのアップデートにより、一部の内容が変更される場合があります。

## 1.2 関連ドキュメント

GOWIN セミコンダクターの[ホームページ](#)から、以下の関連ドキュメントをダウンロード及び閲覧できます。

- Gowin ソフトウェア ユーザーガイド([SUG100](#))
- Gowin FPGA 製品 JTAG コンフィギュレーション ユーザーガイド([TN653](#))
- Gowin FPGA 製品プログラミング・コンフィギュレーション ユーザーガイド([UG290](#))
- Arora V 138K FPGA 製品プログラミング・コンフィギュレーション ユーザーガイド([UG704](#))
- Arora V 25K FPGA 製品プログラミング・コンフィギュレーション ユーザーガイド([UG714](#))
- Arora V 60K FPGA 製品プログラミング・コンフィギュレーション ユーザーガイド([UG718](#))
- Arora V 15K FPGA 製品プログラミング・コンフィギュレーション ユーザーガイド([UG720](#))

## 1.3 用語、略語

本マニュアルで使用される用語、略語、及びその意味を表 1-1 に示します。

表 1-1 用語、略語

用語、略語	正式名称	意味
BSDL	Boundary Scan Description Language	バウンダリスキャン記述言語
FPGA	Field Programmable Gate Array	フィールド・プログラマブル・ゲート・アレイ
GAO	Gowin Analyzer Oscilloscope	Gowinアナライザ オシロスコープ
I/O	Input/Output	入力/出力
SRAM	Static Random Access Memory	スタティックRAM

## 1.4 テクニカル・サポートとフィードバック

GOWIN セミコンダクターは、包括的な技術サポートをご提供しています。使用に関するご質問、ご意見については、直接弊社までお問い合わせください。

ホームページ : [www.gowinsemi.com/ja](http://www.gowinsemi.com/ja)

E-mail : [support@gowinsemi.com](mailto:support@gowinsemi.com)

# 2 概要

## 2.1 Programmer のツールチェーンの概要

### 2.1.1 programmer.exe

グラフィカルツールの programmer.exe は、Gowin FPGA ダウンロード用ソフトウェアであり、グラフィカル操作画面を持ち、ビットストリームのコンフィギュレーションまたはダウンロード機能を便利かつ直感的に提供します。

### 2.1.2 programmer\_cli.exe

programmer\_cli.exe は、Programmer のコマンドライン版です。

### 2.1.3 JTAGLoading.exe

Gowin SVF コマンドラインソフトウェアであり、現在は Windows 版のみあります。現在のバージョンは USB Cable Version 3.0 および 4.0 のみをサポートします。

### 2.1.4 jtagserver.exe

jtagserver.exe は GAO ツールチェーンの一部です。

### 2.1.5 Gowin\_USB\_Cable\_Installer.sh、Makefile

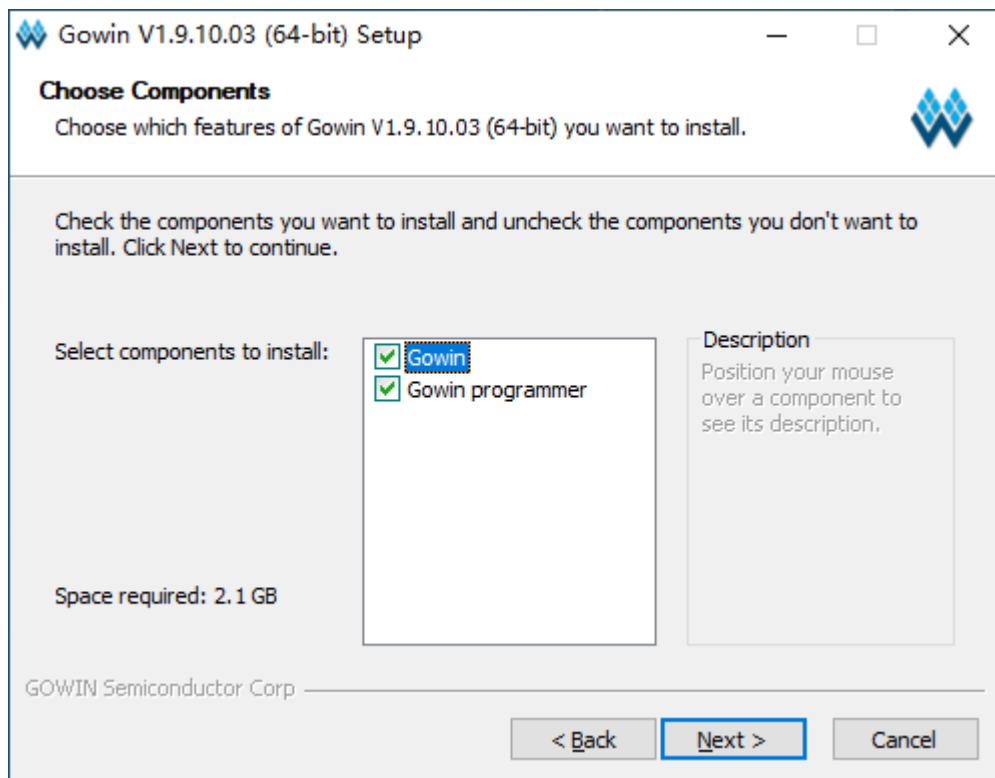
Linux OS におけるケーブルの権限の変更に使用されます。

## 2.2 Programmer のツールチェーンのインストールと起動

### 2.2.1 インストール方法 1

Gowin ソフトウェアをインストールする際にコンポーネントとして Gowin Programmer を選択します(図 2-1)。Gowin ソフトウェアのインストールについては、『Gowin ソフトウェア ユーザーガイド([SUG100](#))』を参照してください。

図 2-1 Gowin ソフトウェアのインストール際のコンポーネントの選択



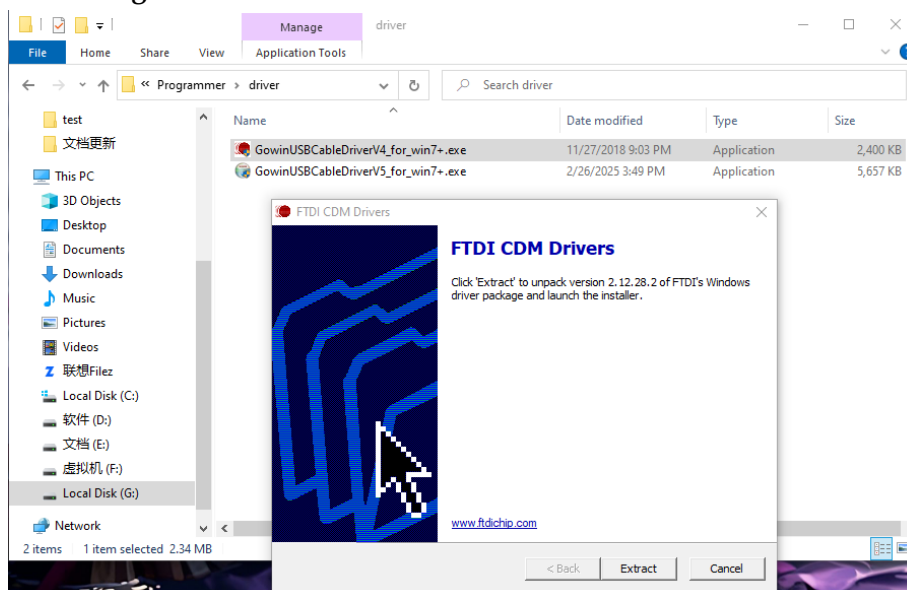
## 2.2.2 インストール方法 2

[ホームページ](#)から Gowin Programmer のインストールパッケージをダウンロードしてインストールします。また、**programmer¥driver** ディレクトリで対応するドライバーを選択してインストールする必要があります(図 2-2)。

注記：

Windows XP システムに GWU2X ドライバーをインストールする場合は、まず対応する USB デバイスを挿入する必要があります。

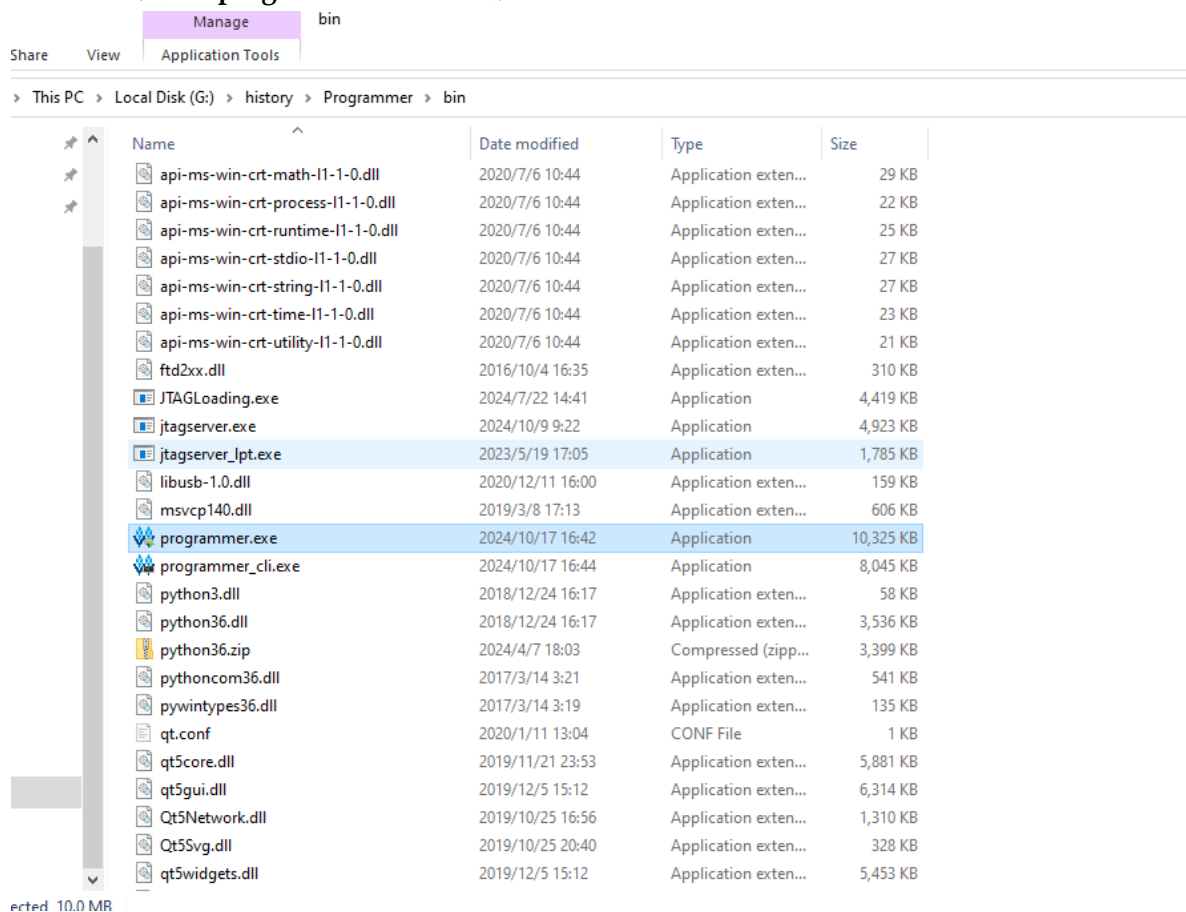
図 2-2 Programmer のドライバーのインストール



## 2.2.3 Programmer のツールチェーンの起動

- Programmer のインストールが完了したら、  
¥x.x¥Programmer¥bin¥programmer.exe をダブルクリックして  
Programmer を起動します(図 2-3)。

図 2-3 programmer.exe の起動



- コマンドライン版の場合、CMD 内で起動できます。例えば、  
programmer\_cli.exe を起動します。



図 2-4 programmer\_cli.exe の起動

```

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

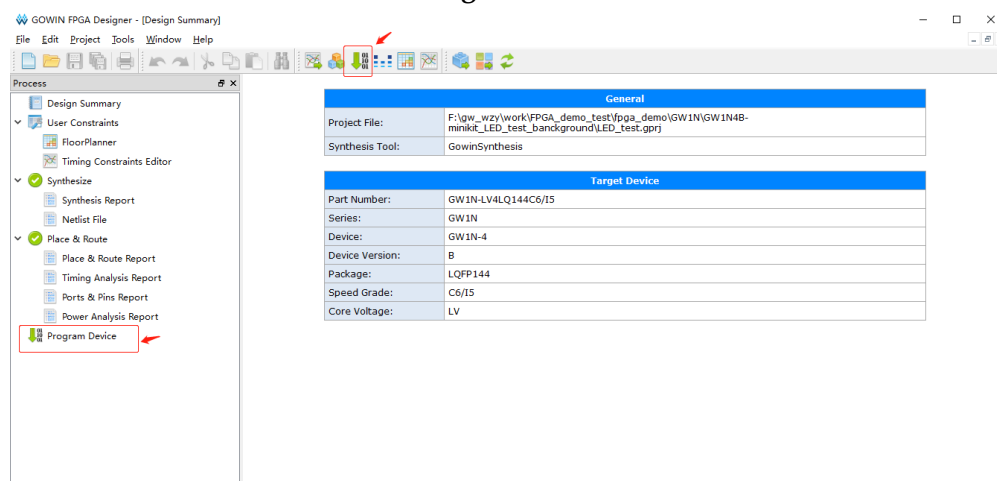
G:\history\Programmer\bin>programmer_cli.exe
Error: No device specified
usage: programmer_cli.exe [-h] [--device <GWxx-x>] [--operation_index <int>]
                        [--chain_index <int>] [--chain_size <int>]
                        [--chain_ir <string>] [--frequency <string>]
                        [--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] [--lpt_address <int>]
                        [--cable "Gowin USB CableFT2CH"]
                        [--cable-index <int>] [--scan-cables [{F,L}]]
                        [--scan] [--filestransform <int>] [--files <string>]
                        [--read-otp-addr] [--save-otp-addr]
                        [--i2c-addr 1010000] [--read-golden-addr]
                        [--save-golden-addr] [--golden-addr 0x800000]
                        [--debug [C:\]]

G:\history\Programmer\bin>

```

- さらに、図 2-5 に示すように、Gowin ソフトウェアのショートカットキーを使用してソフトウェアを起動できます。

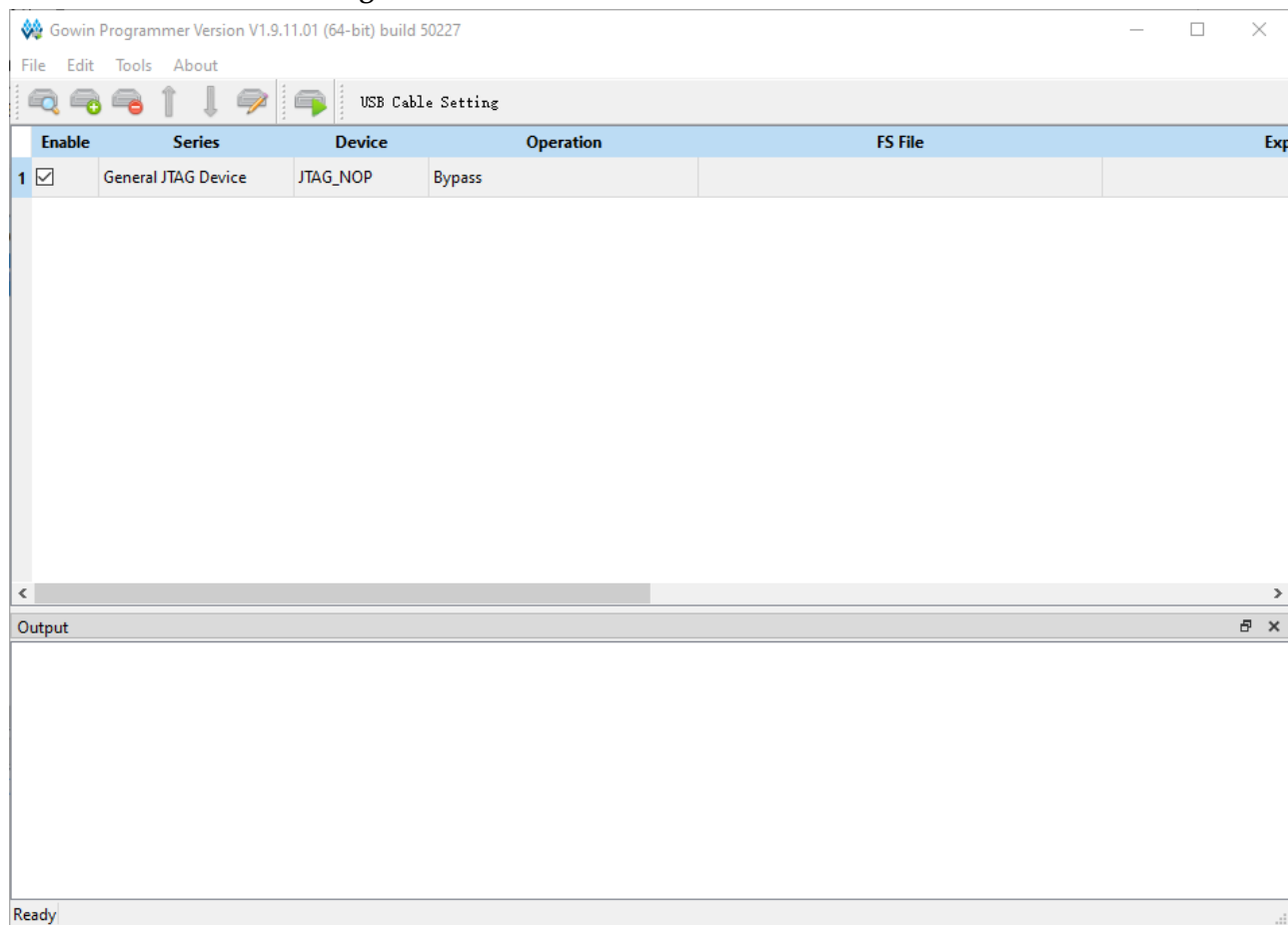
図 2-5 Gowin ソフトウェア上の Programmer ショートカットキー



## 2.3 ソフトウェアの画面

Gowin Programmer の画面には、メニューバー、ツールバー、デバイスステابل、および出力パネルがあります(図 2-6)。

図 2-6 Programmer Main Window



デバイステーブルには、プログラムされるデジタイゼーションのデバイスがすべて表示されます。これらのデバイスは自動スキャンにより検出されるか、手動で追加できます。デバイスはテーブルで行として配列されています。**Enable** をチェックすると、プログラミングすることになります。

テーブルには、**Enable**、**Series**、**Device**、**Operation**、**FS File**、**Export File**、**User Code**、および **IDCODE** が含まれています。そのうち **Enable**、**Family**、**Device**、**Operation**、および **FS File** はクリックして編集できます。ほかの編集不可オプションをダブルクリックすると、デバイスコンフィギュレーション(**Device Configuration**)ダイアログが表示され、このダイアログでプログラミングを構成できます。詳細は、[3.4 プログラミングの構成](#)を参照してください。

出力パネルには、**Output**、**Error**、**Warning**、および **Info** などの情報が含まれます。

#### 注記：

デバイスの「**Enable**」列のチェックが外れている場合、**Programmer** はそのデバイスをチェーン内にはないものとして扱います。

## 2.4 ソフトウェアバージョンの確認

Gowin Programmer と Gowin ソフトウェアには別々のソフトウェアバージョン番号があります。図 2-7 に示すように、ソフトウェア画面で [About] オプションをクリックするとバージョン番号が表示されます。

図 2-7 バージョンの確認



## 2.5 Linux OS におけるケーブルの権限の構成

図 2-8 に示すように、Gowin\_USB\_Cable\_Installer.sh はスクリプトファイル、Makefile はテキストファイルです。どちらもケーブルの権限の変更に使用できます。

図 2-8 ファイル一覧

- 50-programmer\_usb.rules
- Gowin\_USB\_Cable\_Installer.sh
- Makefile
- readme.txt

### 2.5.1 Makefile

ターミナルを開き、`sudo make` コマンドを入力するか、`root` 権限に切り替えて `make` コマンドを入力し、「File 50-programmer\_usb.rules has been copied to /etc/udev/rules/d/」と表示されればインストール成功です（一部の CentOS 6 では再起動が必要）。図 2-9 と図 2-10 に示すとおりです。

図 2-9 Makefile のインストール例(一般ユーザー)

```

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]$ █

```

図 2-10 Makefile のインストール例(root ユーザー)

```

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

```

## 2.5.2 Gowin\_USB\_Cable\_Installer.sh

まず `programmer` があるフォルダを開き、図 2-8 の 4 つのファイルが全て存在するかを確認します。その後、ターミナルを開き、`root` 権限に切り替えて `Gowin_USB_Cable_Installer.sh` に実行の権限を与えます。次に `Gowin_USB_Cable_Installer.sh` を実行し、`complete` と表示されればインストール成功です(図 2-11)。

図 2-11 スクリプトのインストール例

```

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

```

# 3 GUI 版 Programmer でのダウンロード 手順

ダウンロードとは、ダウンロードケーブルを介してビットストリーム・ファイルを **FPGA** デバイスの **SRAM**、オンチップ **Flash**、またはオフチップ **Flash** に伝送するプロセスです。具体的な手順は次のとおりです。

1. ダウンロードケーブルの設定(オプション)：ダウンロード用のケーブルの種類、ポート、周波数などを設定します。

注記：

デフォルトでは、**Programmer** は最初に表示されるポートを使用します。デフォルトの周波数は **2MHz** です。

2. デイジーチェーンとプログラミング操作の構成：デイジーチェーンの実際の物理接続と一致するようにデイジーチェーンを構成し、各デバイスに必要なプログラミング操作とビットストリーム・ファイルを選択します。最上層は **Programmer** 側にあります。
3. ダウンロード：構成済みデイジーチェーンにダウンロードし、最終結果が出力パネルに表示されます。

## 3.1 ダウンロードケーブルの設定

ダウンロードケーブルの設定とは、利用可能なダウンロードケーブルの種類、ポート、周波数などを選択することを意味します。メニューバーから「**Edit > Setting > Cable Setting**」ダイアログを開きます。現在、**Gowin USB Cable (FTDI)**、**Gowin USB Cable (GWU2X)**、**LPT**、**Gowin USB Cable (WINUSB)**、**USB Debugger A** の 5 種類のダウンロードケーブルがサポートされています。

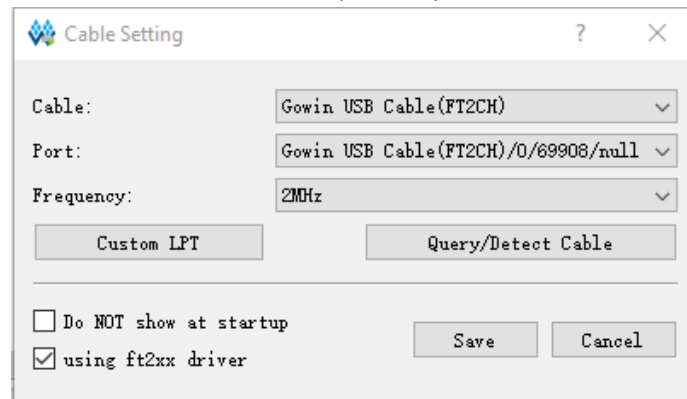
1. **Gowin USB Cable (FT2CH)**を図 3-1 に示します。

- **Cable** : **Gowin USB Cable** を選択します。
- **Port** : デフォルトでは最初の使用可能なポートを使用します。
- **Frequency** : **JTAG** の周波数であり、**2MHz**、**2.5MHz**、**15MHz**、また

は 10MHz を選択できます。デフォルトでは 2MHz です。

- **Do not show at startup** : チェックすると、ソフトウェアを開いたときにこのページはポップアップ表示されません。
- **using ft2xx driver** : チェックすると、ftd2xx.xx を使用して Gowin USB Cable (FTDI) を操作することになります。

図 3-1 Gowin USB Cable (FT2CH)



注記：

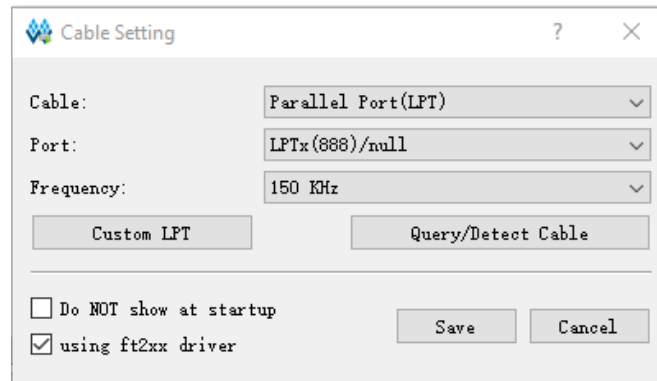
I2C プロトコルを使用する場合、TDI ピンと TDO ピンの接続方法はダウンロードのバージョンによって異なります。次の 2 つのバージョンのダウンロードでは、TCK ピンを SCL ピンに接続する必要があります。

- 4.0 バージョンのダウンロードでは、TDI ピンと TDO ピンを外部で SDA ピンに接続する必要があります。
- 4.1 バージョンのダウンロードでは、TDI ピンは SDA ピンに直接接続され、FTDI チップのピン 27 が Low にプルダウンされるようにします。
- 4.1 バージョンのダウンロードでは、UART プロトコルがさらにサポートされています。その接続要件は次のとおりです。
  - TCK ピンは TX として使用されます。
  - TDI ピンは RX として使用されます。
  - FTDI チップのピン 17 が内部で Low にプルダウンされます。

2. パラレルポート(LPT)を図 3-2 に示します。

- **Cable** : Parallel Port (LPT)を選択します。
- **Port** : 使用可能なポート。PC のデバイス・マネージャーの PCI プロパティに従って選択します。
- **Frequency** : 周波数、150KHz。

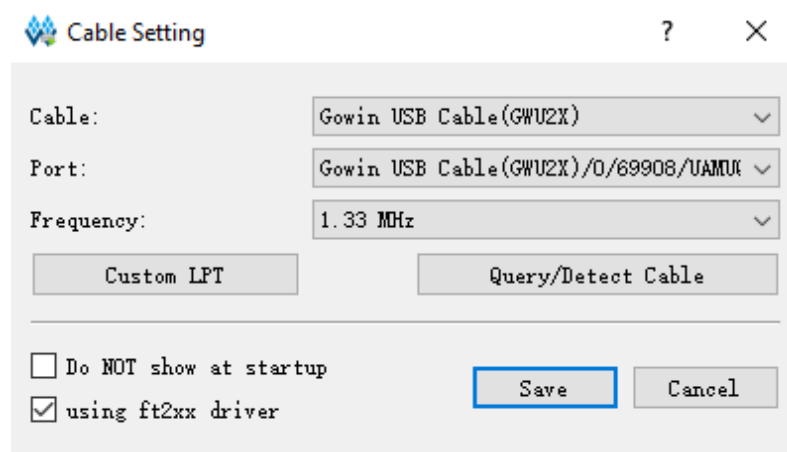
図 3-2 LPT



3. Gowin USB Cable (GWU2X)を図 3-3 に示します。

- Cable : Gowin USB Cable(GWU2X)を選択します。
- Port : デフォルトでは最初の使用可能なポートを使用します。
- Frequency : 周波数、デフォルトでは 1.33MHz。

図 3-3 Gowin USB Cable (GWU2X)



注記：

I2C プロトコルを使用する場合、TDI ピンと TDO ピンの接続方法はダウンローダのバージョンによって異なります。次の 2 つのバージョンのダウンローダでは、TCK ピンを SCL ピンに接続する必要があります。

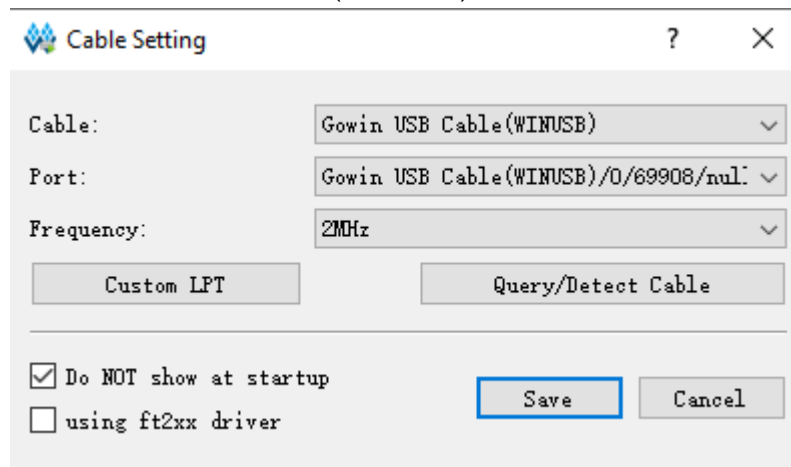
- 5.0 バージョンのダウンローダでは、TDI ピンと TDO ピンを外部で SDA ピンに接続する必要があります。
- 5.1 バージョンのダウンローダでは、TDI ピンは SDA ピンに直接接続され、U2X チップのピン P14 が Low にプルダウンされるようにします。

4. Gowin USB Cable (WINUSB)を図 3-4 に示します。

- Cable : Gowin USB Cable (WINUSB)を選択します。
- Port : デフォルトでは最初の使用可能なポートを使用します。
- Frequency : JTAG の周波数であり、2MHz, 2.5MHz, 15MHz, または 10MHz を選択できます。デフォルトでは 2MHz です。

- using ft2xx driver : チェックされていない場合は、libusbxx.xx を使用して Gowin USB Cable (WINUSB) を操作します。

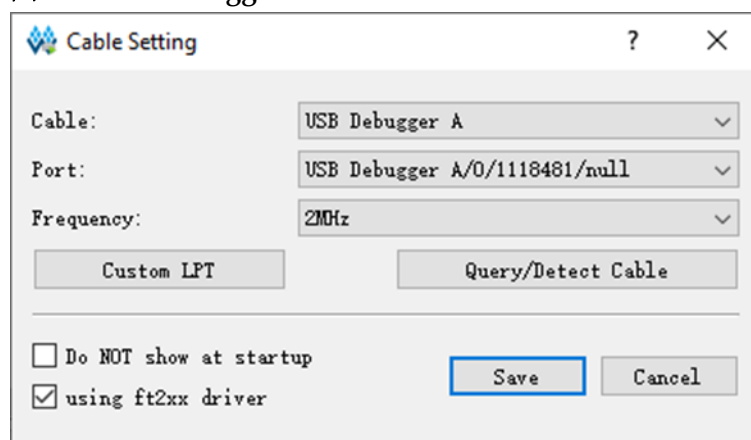
図 3-4 Gowin USB Cable (WINUSB)



5. USB Debugger A を図 3-5 に示します。

- Cable : USB Debugger A を選択します。
- Port : デフォルトでは最初の使用可能なポートを使用します。
- Frequency : 周波数、デフォルトでは 2MHz。

図 3-5 USB Debugger A



## 3.2 デバイスのデイジーチェーンのスキャン


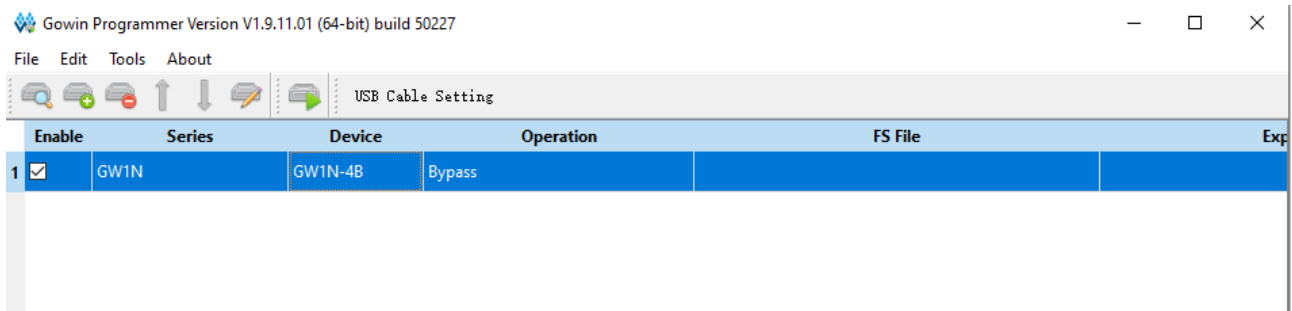
Programmer は、コンピュータに接続されたデバイスのデイジーチェーンを自動的にスキャンすることをサポートしています。「」をクリックしてスキャンします。スキャン完了後、すべてのデバイスはチェーン内の順序で Gowin Programmer のデバイステーブルにリストされます(図 3-6)。



図 3-6 Device Table



	Enable	Series	Device	Operation	FS File	Exp
1	<input checked="" type="checkbox"/>	GW1N	GW1N-4B	Bypass		


注記：

一部のデバイスが同じ ID を有しているため(例：GW2A-18/GW2AR-18)、スキャン後、プロンプトに従って対応するデバイスを選択する必要があります。

Programmer では、デバイスのデイジーチェーンを手動で構成できます。これにはデバイスの追加と取り外し、チェーン内の位置の変更などの操作が含まれます。

## 3.3 デバイスのデイジーチェーンの構成


### 3.3.1 デバイスの追加

1. メニューバーで「Edit>Add Device」またはツールバーで「」をクリックして新しいデバイスを追加します。
2. 「Series」列のセルをクリックし、プルダウンメニューからシリーズを選択します。
3. 「Device」列のセルをクリックし、プルダウンメニューからデバイスを選択します。



注記：

デバイスが選択されている場合、新しいデバイスは選択された位置に追加されます。それ以外の場合、デイジーチェーンの最後に追加されます。


### 3.3.2 デバイスの取り外し

デバイスの行を選択し、メニューバーで「Edit> Delete Device」、またはツールバーで「」をクリックしてデバイスを削除します。

### 3.3.3 チェーン内のデバイス位置の変更

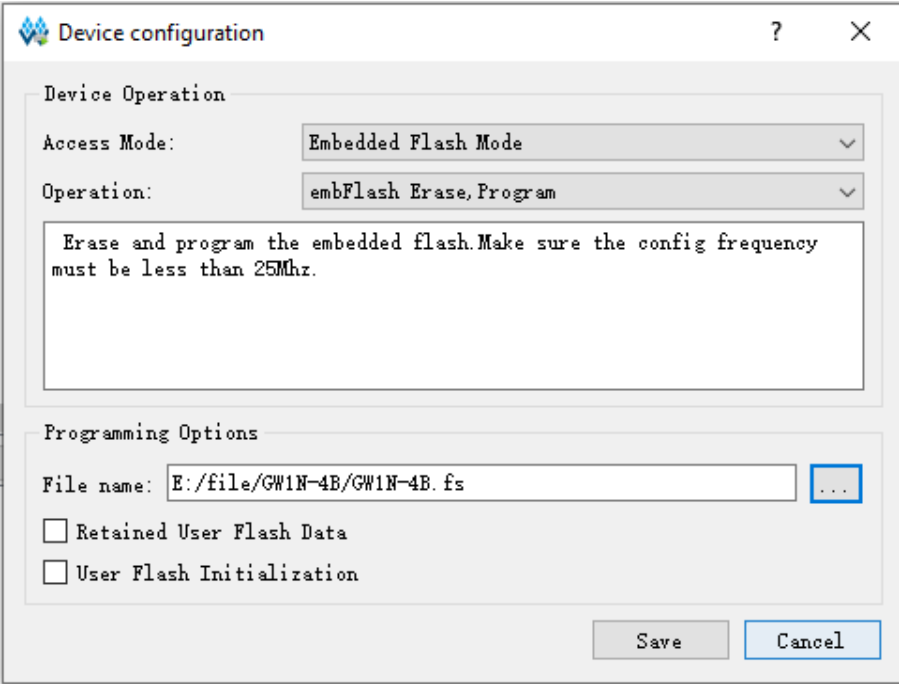
デバイスの行を選択し、メニューバーで「Edit > Move Device Up(Down)」、またはツールバーで「 ()」をクリックしてデバイスのチェーン内の位置を調整します。

## 3.4 プログラミングの構成

デバイスの行を選択し、メニューバーで「Edit > Configure Device」またはツールバーで「」をクリックするか、「Operation」列のセルを

ダブルクリックして「Device Configuration」ダイアログを開きます(図 3-7)。

図 3-7 Device Configuration Interface



- Access Mode : デバイスのプログラミングモードを選択します。
- Operation : デバイスのプログラミング操作を選択します。詳細は表 3-1 を参照してください。
- File name : ビットストリーム・ファイルを選択します。
- Device : プログラミングモードとして External Flash Mode が選択された場合、オフチップ Flash の型番を選択する必要があります。
- Start Address : プログラミングモードとして External Flash Mode が選択された場合、オフチップ Flash の開始アドレスを指定する必要があります。

表 3-1 Device に対する操作の説明

Access Mode	Operation	説明
SRAM Mode	Bypass	Bypass
	Read Device Code	デバイスのID Code、User Code、Status Codeを読み出します
	Read User Code	デバイスのUser Codeを読み出します
	Read Status Register	デバイスの状態を読み出します
	Reprogram	リロード。Flash内のビットストリーム・ファイルをSRAMにリロードするために使用されます。
	SRAM Erase	SRAM内のデータを消去します
	SRAM Program	FPGA SRAMにビットストリーム・ファイ

Access Mode	Operation	説明
		ルをダウンロードします
	SRAM program JTAG 1149	JTAGを介するピュアデータ書き込みモード。 <b>CRC</b> チェックはサポートされません。暗号化または圧縮されたビットストリーム・ファイルはサポートされません
Embedded Flash Mode	embFlash Erase, Program	オンチップFlashを消去した後、データを書き込みます。
	embFlash Erase, Program, Verify	オンチップFlashを消去した後、データを書き込み、検証を行います。
	emFlash Erase Only	オンチップFlashのみを消去します。
Embedded Flash Background Mode	embFlash Background Erase, Program	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込みます。
	embFlash Background Erase, Program, Verify	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込み、検証を行います。
	emFlash Background Erase Only	バックグラウンド・プログラミング。オンチップFlashを消去します。
	embFlash Background Erase, Program without Reprogram	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込みます。その後、 <b>reprogram</b> 命令を送信しません。
	embFlash Background Erase, Program, Verify without Reprogram	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込み、検証を行います。その後、 <b>reprogram</b> 命令を送信しません。
External Flash Mode	exFlash Erase, Program	オフチップFlashを消去した後、データを書き込みます。
	exFlash Erase, Program, Verify	オフチップFlashを消去した後、データを書き込み、検証を行います。
	exFlash Bulk Erase	オフチップFlashを消去します。
	exFlash Verify	オフチップFlashのデータを検証します。
	exFlash Background Erase, Program	バックグラウンド・プログラミング。オフチップFlashを消去した後、データを書き込みます。
	exFlash Background Erase, Program, Verify	バックグラウンド・プログラミング。オフチップFlashを消去した後、データを書き込み、検証を行います。
	exFlash Erase, Program in bscan	<b>bscan</b> モードでオフチップFlashを消去した後にデータを書き込みます。
	exFlash Erase, Program, Verify in bscan	<b>bscan</b> モードでオフチップFlashを消去した後にデータを書き込んで検証します。
	exFlash Verify in bscan	<b>bscan</b> モードでオフチップFlash内のデータを読み出してビットストリーム・ファイル

Access Mode	Operation	説明
		と比較して検証します。
	exFlash Bulk Erase in bscan	bscanモードでオフチップFlashを消去します。
	exFlash C Bin Erase, Program	オフチップFlashを消去した後、RISC-VのbinファイルをオフチップFlashに書き込みます。
	exFlash C Bin Erase, Program, Verify	オフチップFlashを消去した後、RISC-VのbinファイルをオフチップFlashに書き込んで検証します。
	exFlash Erase, Program thru GAO-Bridge	GAO-BridgeでオフチップFlashを消去した後、データを書き込みます。
	exFlash Erase, Program, Verify thru GAO-Bridge	GAO-BridgeでオフチップFlashを消去した後、データを書き込み、検証を行います。
	exFlash C Bin Erase, Program thru GAO-Bridge	GAO-BridgeでオフチップFlashを消去した後、RISC-Vのbinファイルを書き込みます。
	exFlash C Bin Erase, Program, Verify thru GAO-Bridge	GAO-BridgeでオフチップFlashを消去した後、RISC-VのbinファイルをオフチップFlashに書き込んで検証します。
	exFlash Bulk Erase thru GAO-Bridge	GAO-BridgeでオフチップFlashを消去します。
	exFlash Verify thru GAO-Bridge	GAO-BridgeモードでオフチップFlash内のデータを読み出してビットストリーム・ファイルと比較して検証します。
	exFlash Export thru GAO-Bridge	GAO-BridgeモードでオフチップFlash内のデータを読み出してbinファイルとして保存します。
	exFlash Import thru GAO-Bridge	GAO-BridgeモードでオフチップFlash内のデータを読み出してbinファイルとして保存します。
	exFlash Erase Sectors About File	指定された外部Flash容量のセクターをファイル・サイズに応じて消去します。
	exFlash Erase Sectors About File thru GAO-Bridge	GAO-Bridgeで、指定された外部Flash容量のセクターをファイル・サイズに応じて消去します。
Slave SPI Mode	Slave SPI Read ID Code	SSPIモードでデバイスIDを読み出します。
	Slave SPI Scan exFlash	SSPIモードでオフチップFlashをスキャンします。
	Slave SPI Program SRAM	SSPIモードでデータをSRAMに書き込みます。
GoConfigure IP Mode	embFlash Background Ese,Program,thru IP-JTAG	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込みます。

Access Mode	Operation	説明
	embFlash Background Erase,Program,Verify thru IP-JTAG	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込み、検証します。
	embFlash Background Erase Only thru IP-JTAG	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去します。
	embFlash Background Program,Hotboot thru JTAG-JTAG	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込み、検証し、ホットブートします。
	embFlash Background Erase Only thru IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去します。
	embFlash Background Erase,Program,thru IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込みます。
	embFlash Background Erase,Program,Verify thru IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込み、検証します。
	Read ID Register thru IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にID Codeを読み出します。
	Read User Code thru IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にUser Codeを読み出します。
	Read Status Code thru IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にStatus Codeを読み出します。
	exFlash Erase,Program,thru UART-IP-SPI	GoConfigモードでのバックグラウンド・プ

Access Mode	Operation	説明
		ログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込みます。
	exFlash Erase,Program,Verify thru UART-IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込み、検証します。
	SRAM Reprogram thru UART-IP-SPI	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にデバイスを再起動します。
	embFlash Background Erase,Program,thru IP-I2C	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込みます。
	embFlash Background Erase,Program,Verify thru IP-I2C	GoConfigモードでのバックグラウンド・プログラミング。まず、対応するIPを含むビットストリーム・ファイルをSRAMにダウンロードし、次にオンチップFlashを消去し、データを書き込み、検証します。
External Flash Mode Arora V	exFlash Erase,Program Arora V	Arora VデバイスのオフチップFlashを消去した後、データを書き込みます。
	exFlash Erase,Program,Verify Arora V	Arora VデバイスのオフチップFlashを消去した後、データを書き込み、検証を行います。
	exFlash Verify Arora V	Arora VデバイスのオフチップFlashのデータを検証します。
	exFlash Bulk Erase Arora V	Arora VデバイスのオフチップFlashのデータを消去します。
	exFlash C Bin Erase,Program Arora V	Arora VデバイスのオフチップFlashを消去した後、RISC-VのbinファイルをオフチップFlashに書き込みます。
	exFlash C Bin Erase,Program,Verify Arora V	Arora VデバイスのオフチップFlashを消去した後、RISC-VのbinファイルをオフチップFlashに書き込んで検証します。
	exFlash Background Erase,Program Arora V	バックグラウンド・プログラミング。Arora VデバイスのオフチップFlashを消去した後、データを書き込みます。
	exFlash Background Erase,Program,Verify Arora V	バックグラウンド・プログラミング。Arora VデバイスのオフチップFlashを消去した後、データを書き込み、検証を行います。

Access Mode	Operation	説明
	exFlash Erase,Program thru GAO-Bridge Arora V	GAO-BridgeでArora VデバイスのオフチップFlashを消去した後、データを書き込みます。
	exFlash Erase,Program,Verify thru GAO-Bridge Arora V	GAO-BridgeでArora VデバイスのオフチップFlashを消去した後、データを書き込み、検証を行います。
	exFlash Verify thru GAO-Bridge Arora V	GAO-BridgeでArora VデバイスのオフチップFlashのデータを検証します。
	exFlash Export thru GAO-Bridge Arora V	まず、JTAG to SPIのブリッジを構築し、Arora VデバイスのオフチップFlashのデータを読み出し、binファイルとして保存します。
	exFlash Erase Sectors About File Arora V	Arora Vデバイスの指定された外部Flash容量のセクターをファイル・サイズに応じて消去します。
	exFlash Erase Sectors About File thru GAO-Bridge Arora V	GAO-BridgeでArora Vデバイスの指定された外部Flash容量のセクターをファイル・サイズに応じて消去します。
Internal Flash Mode	sFlash Erase,Program	オンチップFlashを消去した後、データを書き込みます。
	sFlash Erase,Program,Verify	オンチップFlashを消去した後、データを書き込み、検証を行います。
	sFlash Verify Only	オンチップFlashのデータを読み出します
	sFlash Bulk Erase	オンチップFlashのデータを消去します
	sFlash Background Erase,Program	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込みます。
	sFlash Background Erase,Program,Verify	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込み、検証を行います。
	sFlash Erase,Program in bscan	bscanモードでオンチップFlashを消去した後にデータを書き込みます。
	sFlash Bulk Erase in bscan	bscanモードでオンチップFlashを消去します。
	sFlash Erase,Program,Verify thru GAO-Bridge	GAO-BridgeでオンチップFlashを消去した後、データを書き込みます。
	sFlash Background Erase,Program,Verify thru GAO-Bridge	GAO-Bridgeでのバックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込み、検証を行います。
	sFlash Export thru GAO-Bridge	GAO-BridgeモードでオンチップFlash内のデータを読み出してbinファイルとして保存します。
	sFlash Erase Sectors About	指定された外部Flash容量のセクターをファ



Access Mode	Operation	説明
	File	イル・サイズに応じて消去します。
	sFlash Erase Sectors About File thru GAO-Bridge	GAO-Bridgeで、指定された外部Flash容量のセクターをファイル・サイズに応じて消去します。
MCU Mode(L)	Firmware Erase Only	ファームウェアを消去します
	Firmware Erase,Program	ファームウェアを消去した後、書き込みます。
	Firmware Erase,Program,Verify	ファームウェアを消去した後、書き込んで検証します。
SecureFPGA Mode (L)	Connect to JTAG of MCU	デバイスのJTAG制御をMCUに転送します
	Firmware Erase,Program Securely	まずファームウェアを消去し、次にセキュアモードでファームウェアを書き込んで検証します。
	Authentication Code Erase,Program,Verify	-
	Authentication Code Read	-
I2C Mode	I2C Program SRAM	I2Cモードでビットストリーム・ファイルをSRAMにダウンロードします。
	I2C Program Flash	I2Cモードでビットストリーム・ファイルをオンチップFlashにダウンロードします。
	I2C Erase Flash Only	I2CモードでオンチップFlashのデータを消去します
Debugging/Temporary Mode	Set Flash QE For 9X/18X	GW2AN-9X/18Xデバイスの内部FlashのQE bitをセットします。
	Set Flash QE For Arora V	Arora VデバイスのオフチップFlashのQE bitをセットします。
	exFlash Detect ID	オフチップFlashのIDを読み出します
	Progam Arora V DNA Data	Arora VデバイスのDNAデータをダウンロードします
	efuse read Arora V	Arora Vデバイスのuser_fuse領域から128ビットのデータを読み出します。
	SRAM Program thru Serial	Serialモードでビットストリーム・ファイルをSRAMにダウンロードします。
	Set Boot Address Arora V	Arora Vデバイスの起動アドレスを設定します。つまり、デバイスの起動アドレスをSRAMにダウンロードします。


## 注記：

- GW2A/GW2AR シリーズ FPGA には embedded Flash がありません。
- バックグラウンド・プログラミング：現在実行中の機能に影響を与えず、SRAM を消去せずに Flash をプログラムします。



- **GAO-Bridg** モード：通常、オフチップ Flash または内部 Flash をコンフィギュレーションするために使用されます。まず、JTAG to SPI のブリッジを構築します (Gao-Bridge 機能を SRAM にコンフィギュレーションします)。次に、JTAG を介して Flash をプログラムします。
- **goConfigIP** モード：その実装原理は Gao-Bridge と同じです。ただし、一般的には、ユーザープロジェクトに goconfigIP を追加することをお勧めします。

### 3.4.1 SRAM モードの構成


1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」  
またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。
2. 「Access Mode」のドロップダウン・リストから「SRAM Mode」を選択します。
3. 必要に応じて「Operation」のドロップダウン・リストから対応する操作を選択します。
4. デバイスが GOWIN デバイスでない場合、手動で命令レジスタの長さを指定するか、デバイスの BSDL ファイルを指定して Programmer に命令レジスタの長さを読み出させる必要があります。
5. 「Save」をクリックして構成を完了します。

注記：

他社デバイス(JTAG-NOP)の場合、Bypass 操作のみがサポートされます。

### 3.4.2 LittleBee ファミリーFPGA での組み込み Flash モードの構成


LittleBee ファミリー製品は、組み込み Flash を備えているため、組み込み Flash モードが使用できます。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」  
またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。
2. 「Access Mode」のドロップダウン・リストから「Embedded Flash Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「File name」で対応するビットストリーム・ファイルを選択します。
5. 「Save」をクリックして構成を完了します。

### 3.4.3 オフチップ Flash モードの構成


GOWIN プログラマは、オフチップ Flash を使用したプログラミング

をサポートします。オフチップ Flash モードの構成プロセスは次のとおりです。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」  
またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。
2. 「Access Mode」のドロップダウン・リストから「External Flash Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」で「exFlash Program」または「exFlash Verify」を選択する場合、「File name」で対応するビットストリーム・ファイルを選択する必要があります。
5. LittleBee と Arora ファミリーは、読み出しコマンドが 0x03 または 0x0B の SPI Flash をサポートしています。これに加えて、Arora V デバイスは読み出しコマンドが 0x6B の SPI Flash をさらにサポートしています。
6. Flash のタイプはデフォルトで「Generic Flash」です。
7. オフチップ Flash の開始アドレスを選択します。現在のデフォルト値は 0x000000 です。
8. 「Save」をクリックして構成を完了します。

### 3.4.4 GW2AN-9X/18X デバイスの内部 Flash モードの構成

Gowin Programmer は、GW2AN-9X/18X デバイスの内部 Flash のプログラミングをサポートします。その構成プロセスは次のとおりです。


1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」  
またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。
2. 「Access Mode」のドロップダウン・リストから「Internal Flash Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」で「sFlash Program」または「sFlash Verify」を選択する場合、「File name」で対応するビットストリーム・ファイルを選択する必要があります。
5. Flash のタイプはデフォルトで Generic Flash です。
6. Flash の開始アドレスを選択します。現在のデフォルト値は

0x000000 です。

7. 「Save」をクリックして構成を完了します。


### 3.4.5 Arora V デバイスのオフチップ Flash モードの構成

GOWIN プログラマは、オフチップ Flash を使用したプログラミングをサポートします。オフチップ Flash モードの構成プロセスは次のとおりです。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」  
またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。
2. 「Access Mode」のドロップダウン・リストから「External Flash Mode Arora V」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」で「exFlash Program」または「exFlash Verify」を選択する場合、「File name」で対応するビットストリーム・ファイルを選択する必要があります。
5. 読み出しコマンドが 0x03、0x0B、0x6B のオフチップ SPI Flash をサポートします。
6. Flash のタイプはデフォルトで Generic Flash です。
7. Flash の開始アドレスを選択します。現在のデフォルト値は 0x000000 です。
8. 「Save」をクリックして構成を完了します。

### 3.4.6 Slave SPI モード

Slave SPI モードでは、ダウンロードケーブルを SSPI 専用ピンに接続する必要があります。『Gowin FPGA 製品プログラミング・コンフィギュレーション ユーザーガイド([UG290](#))』を参照してください。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」  
またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。
2. 「Access Mode」のドロップダウン・リストから「Slave SPI Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」で「Slave SPI Program SRAM」を選択する場合、「File

name」で対応するビットストリーム・ファイルを選択する必要があります。

5. 「Save」をクリックして構成を完了します。

### 3.4.7 MCU モード

Gowin Programmer は、デバイスに内蔵されたハードコア・プロセッサ M3 のファームウェアの更新をサポートします。その構成プロセスは次のとおりです。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」

またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。

2. 「Access Mode」のドロップダウン・リストから「MCU Mode」または「MCU Mode L」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」で「Firmware Program」を選択する場合、「File name」で対応するビットストリーム・ファイルとファームウェアを選択する必要があります。
5. 「Save」をクリックして構成を完了します。

### 3.4.8 セキュア FPGA モードの構成

Gowin Programmer は、GW1NSE/GW1NSER シリーズのセキュア FPGA モードの構成をサポートします。その構成プロセスは次のとおりです。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」

またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。

2. 「Access Mode」のドロップダウン・リストから「SecureFPGA Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」で「Authentication Code Erase, Program, Verify」を選択する場合、「File name」で対応するファイルを選択する必要があります。
5. 「Save」をクリックして構成を完了します。

### 3.4.9 I2C モード

I2C モードでは、ダウンロードラインを専用の I2C ピンに接続する必要があります。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」

またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。

2. 「Access Mode」のドロップダウン・リストから「I2C Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」で「I2C Program SRAM」または「I2C Program Flash」を選択する場合、「File name」で対応するビットストリーム・ファイルを選択する必要があります。
5. 「Save」をクリックして構成を完了します。

### 3.4.10 デバッグモード

Gowin Programmer はデバッグモードをサポートしており、その構成プロセスは次のとおりです。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」

またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。

2. 「Access Mode」のドロップダウン・リストから「Debugging/Temporary Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「SRAM Program thru Serial」を選択する場合、「File name」で対応するビットストリーム・ファイルを選択する必要があります。
5. 「Save」をクリックして構成を完了します。

### 3.4.11 goConfigIP モードの構成

Gowin Programmer は、goConfigIP によるプログラミングをサポートしています (このモードは、goConfigIP の機能が正常かどうかを確認するためのデバッグ・ステージにのみ適用されます)。その構成プロセスは次のとおりです。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」

またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。

2. 「Access Mode」のドロップダウン・リストから「GoConfig IP Mode」を選択します。
3. 「Operation」のドロップダウン・リストから必要に応じて対応する操作を選択します。
4. 「Operation」でビットストリーム・ファイルのダウンロードが必要な操作を選択する場合、「File name」で対応するビットストリーム・ファイルを選択する必要があります。
5. 「Save」をクリックして構成を完了します。

### 3.4.12 バックグラウンド・プログラミング・モード

Gowin Programmer は、バックグラウンド・プログラミング (プログラミング中にデバイスの現在の実行状態に影響を与えません)をサポートします。その構成プロセスは次のとおりです。

1. デバイスの行を選択し、メニューバーで「Edit > Configure Device」

またはツールバーで「」をクリックするか、「Operation」列のセルをダブルクリックして「Device Configuration」ダイアログを開きます。

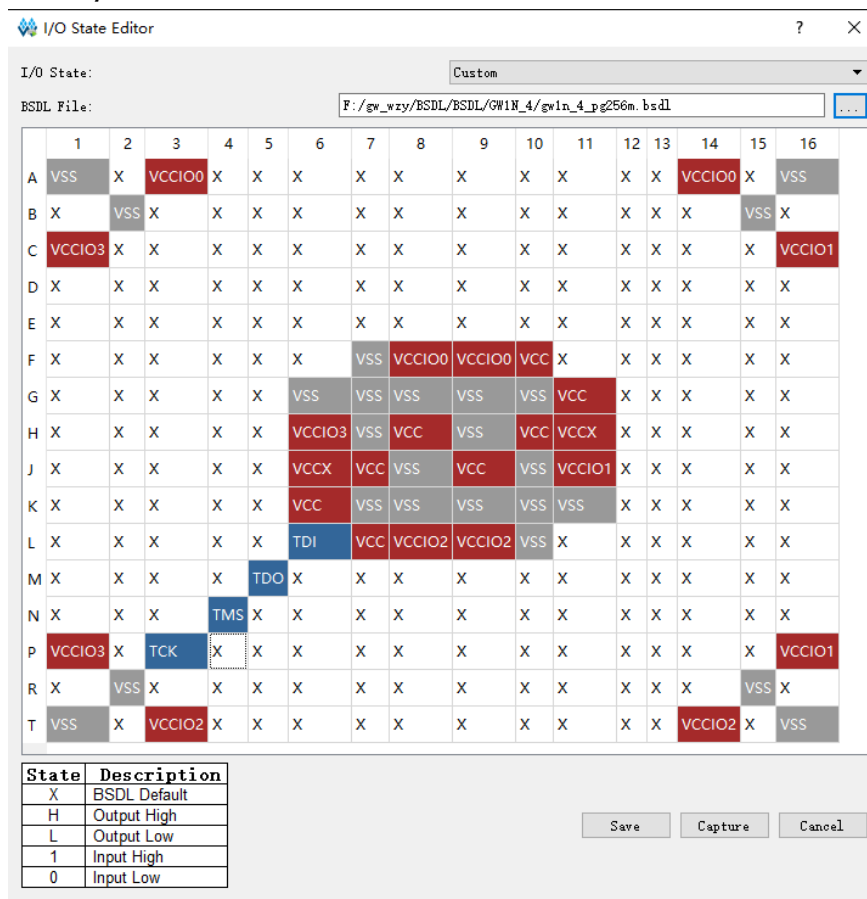
2. 「Operation」のドロップダウン・リストから Background という単語が含まれる操作を選択します。
3. 「Operation」でビットストリーム・ファイルのダウンロードが必要な操作を選択する場合、「File name」で対応するビットストリーム・ファイルを選択する必要があります。
4. 「Save」をクリックして構成を完了します。

## 3.5 ピンの状態の編集

Programmer は I/O State Editor ツールを使用して入力及び出力ピンの値を編集し、ダウンロード前のピンの状態を設定できます。

1. デバイスの行を選択してから、メニューバーで「Edit > I/O State」をクリックするか、「I/O State」を右クリックして I/O State Editor を開きます。
2. デバイス型番とパッケージに一致する BSM ファイルを選択します。
3. セルをクリックしてピンの状態を変更するか、右クリックしてすべてのピンを同じ状態に設定することができます。

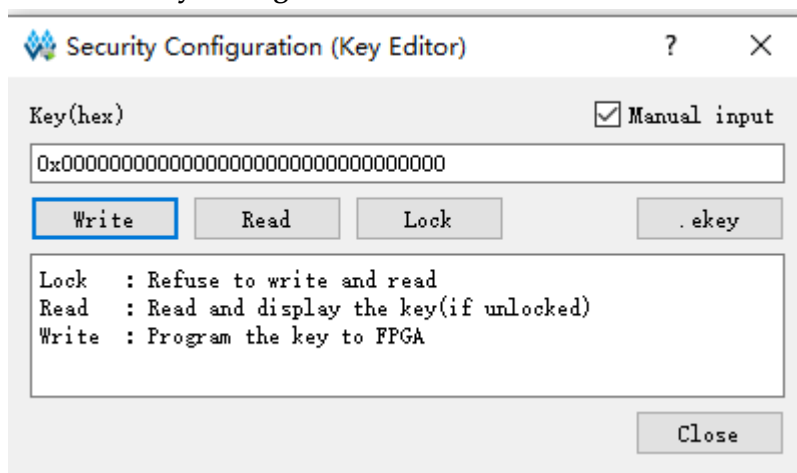
図 3-8 I/O State Editor



## 3.6 セキュリティ

暗号化されたビットストリーム・ファイルを使用する場合、ビットストリーム・ファイルの復号化キー(key)を FPGA に書き込む必要があります。デバイスの行を選択し、メニューバーで「Edit > Security Key Setting」をクリックするか、右クリックメニューから「Security Key Setting」をクリックして「Security Configuration」ダイアログを開きます(図 3-9)。

図 3-9 Security Configuration





- **Manual input** : キーを平文で入力するかどうか選択します。
- **.ekey** : キーファイルを開きます。
- **Write** : 指定された **key** 値を **FPGA** に書き込みます。
- **Read** : ロックしていない状況で、**FPGA** 内の **key** 値を読み出し、表示します。
- **Lock** : **FPGA** 内の **key** をロックすると、読み出しと書き込みができなくなります。

復号化キーを書き込む方法は 2 つあります。

#### 平文でのキー書き込み

復号化キーが書き込まれた後、検証のために画面上の読み出し(read) 命令を選択して書き込まれたキーをリードバックすることができます。

キーが書き込まれた後、ユーザーは **lock** 命令を使用してキーを **FPGA** 内にロックすることができます。これにより、キーの読み出しと書き込みはすべて無効になります：キーの値は変更できず、読み出されたビットは全部「1」となります。

#### キーファイルによる書き込み


キーファイルを開き、「**Write**」ボタンをクリックすると、キーは開発ボードに書き込まれます。次に検証のためにキーを読み出します。検証に成功すると、キーは **FPGA** 内にロックされます。キーファイルを開かずに直接「**Write**」ボタンをクリックすると、デフォルトのキー値が **FPGA** に書き込まれます。検証に成功すると、デフォルトのキー値が **FPGA** 内にロックされます。

復号化キーが設定された後、暗号化されたビットストリームデータは、復号化キーとの照合に成功した後にのみ利用可能です。暗号化されていないビットストリームデータのコンフィギュレーションは、キーの影響を受けません。

#### 注記：

**GOWIN** セミコンダクター **FPGA** のキーの初期値のすべてのビットは **0** です。キー値のビットが **1** に変更されると、**0** に戻すことはできません。たとえば、書き込まれたキー値が **00000000-00000000-00000000-00000001** の場合、このデバイスのキーの最下位ビットは常に **1** でなければなりません。詳細は、『**GW2A(R)** シリーズ **FPGA** 製品 **AES** 暗号化プログラミング ユーザーガイド([TN654](#))』を参照してください。

## 3.7 ダウンロード

ダウンロードケーブルとデジチェーンの構成が完了した後、メニューバーで「上の「**Program/Configure**」を選択するか、ツールバーで「」をクリックし、デバイスをコンフィギュレーションします。最終結果は出力パネルに表示されます。

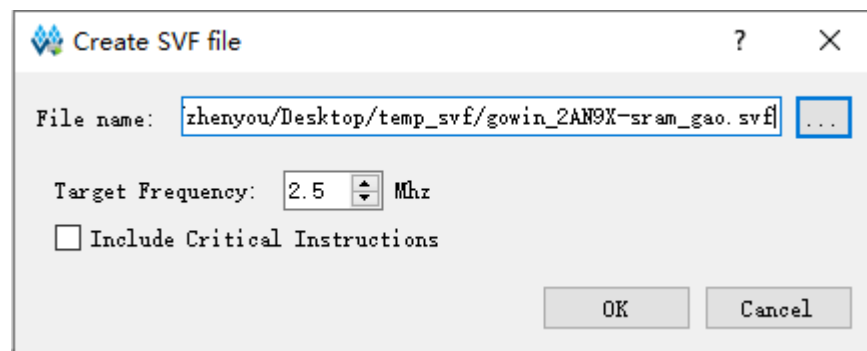


## 3.8 SVF ファイルの作成

fs ファイルで SVF ファイルを作成することがサポートされています。

1. 「LittleBee ファミリーFPGA での組み込み Flash モードの構成」を参照しながら構成します(GW1N4 を選択)。
2. デバイスチェーンを選択し、メニューバーで「Edit > SVF File Create」をクリックするか、右クリックして「SVF File Create」を選択して Create SVF file ダイアログを開きます。
3. File name フィールドで SVF ファイルのファイル名と保存パスを編集できます(図 3-10)。
4. 「OK」ボタンをクリックして SVF ファイルの作成を完了します。

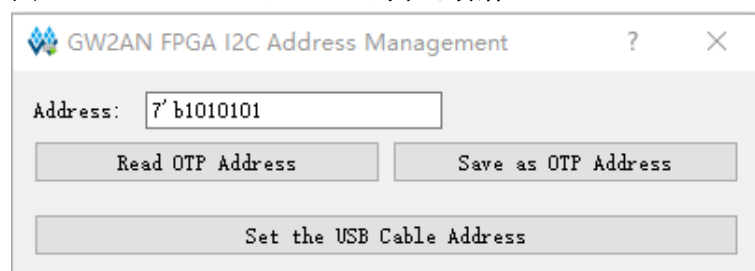
図 3-10 Create SVF File



## 3.9 I<sup>2</sup>C Slave アドレスに対する操作

I<sup>2</sup>C Slave アドレスの読み出し、書き込み、設定などの操作がサポートされます。現在、この操作は GW2AN-18X と GW2AN-9X でのみサポートされています。

図 3-11 I<sup>2</sup>C Slave アドレスに対する操作



- Read OTP Address : I<sup>2</sup>C Slave アドレスを読み出します。
- Save as OTP Address : 指定された I<sup>2</sup>C Slave アドレスを FPGA に書き込みます。
- Set the USB Cable Address : FPGA を I<sup>2</sup>C モードで操作する場合、I<sup>2</sup>C Slave アドレスを設定する必要があります。

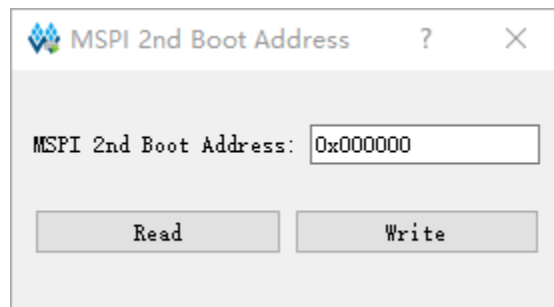
注記 :

I<sup>2</sup>C Slave アドレスに対しては、2 ビットののみが変更可能です。デフォルトのアドレスは 7'b1010**00**0 です。黄色のビットを 0 から 1 に変更でき、ただし、1 に変更した場合、0 に戻すことはできません。

## 3.10 MSPI 2nd Boot アドレスに対する操作

Arora V デバイスの Golden Image アドレスの読み出しと書き込みがサポートされます。アドレスの範囲は 0x000000-0xF00000 です。

図 3-12 MSPI 2nd Boot アドレスに対する操作



- Read : MSPI 2nd boot アドレスを読み出します。
- Write : 指定された MSPI 2nd boot アドレスを FPGA に書き込みます。

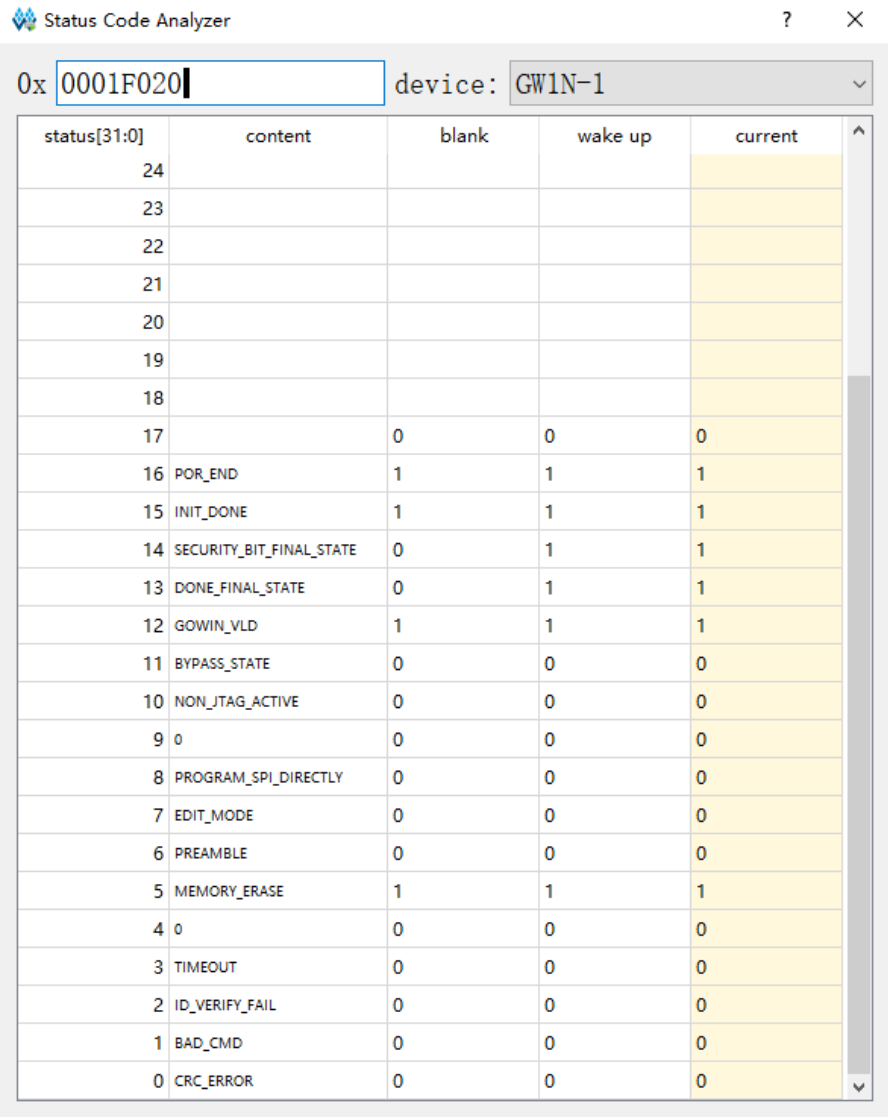
注記 :

GOWIN FPGA の 2nd boot address の初期値のすべてのビットは 0 です。アドレスのビットが 1 に変更されると、0 に戻すことはできません。

## 3.11 ステータスコードの分析

開発ボードの現在のステータスを確認する必要がある場合、ステータスコードを分析することで開発ボードのステータスを取得できます。メニューバー上の「Tools > Analyzer Status Viewer」オプションをクリックするか、ステータスコード表示行を右クリックして「Analyze Status Code」をクリックして、「Status Code Analyzer」ダイアログボックスを開きます(図 3-13)。ステータス・レジスタの詳細については、『GOWIN FPGA ステータス・レジスタの説明([TN711](#))を参照してください。

図 3-13 ステータスコード分析の画面

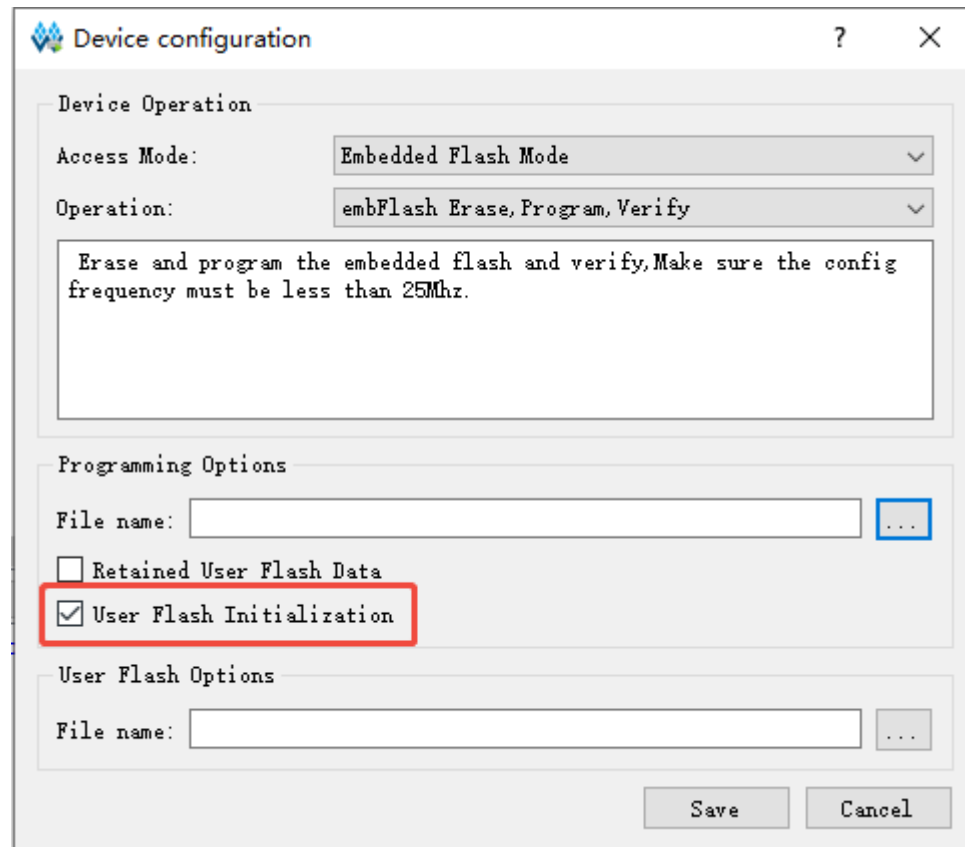


まず開発ボードを選択し(device リストから選択)、次にステータスコードを入力します。こうすると、ステータスコードの各ビットの結果は以下のリストに出力されます。

### 3.12 User Flash の初期化とデータ保持

LittleBee ファミリーの製品は、ユーザーに User Flash を提供しています。Programmer は、オンチップ Flash への書き込みと同時に User Flash データを User Flash に書き込むことができます。しかし、セキュリティのために、Programmer では User Flash プログラミングのみがサポートされ、リードバックはサポートされていません。プログラミングの際に、拡張子が.fi のファイルを User Flash の初期化ファイルとして使用できます(図 3-14)。

図 3-14 User Flash Initialization



User Flash のデータを保持する機能とは、オンチップ Flash をプログラムするときに User Flash 領域のデータが変更されないことを意味します。「Retained User Flash Data」と「User Flash Initialization」は相互に排他的であり、下図に示すとおりです。

図 3-15 Retained User Flash Data

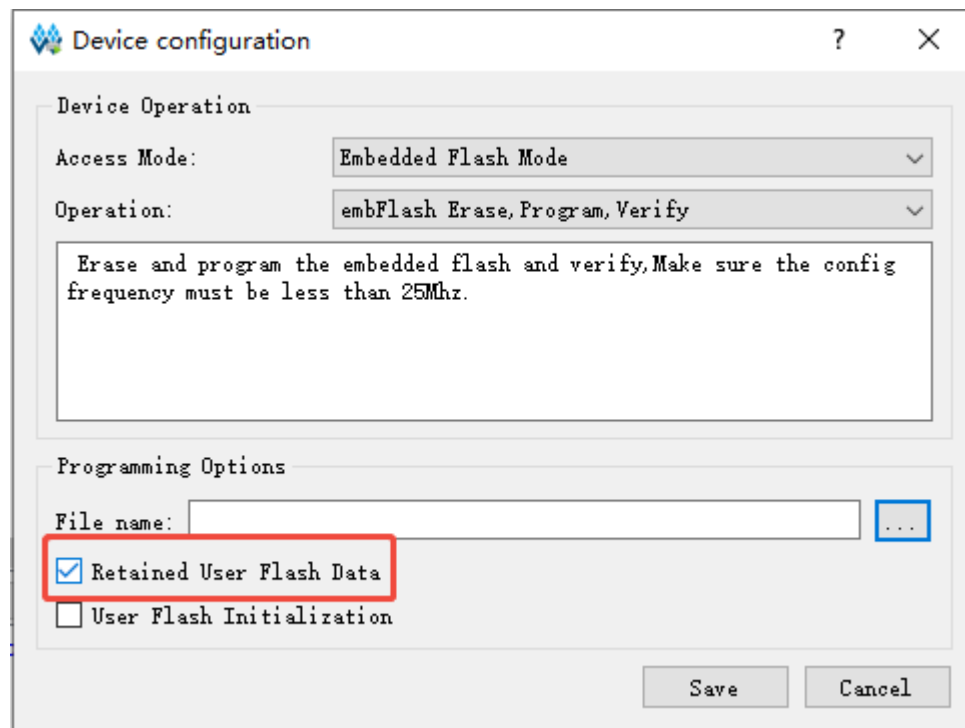


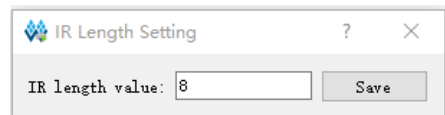
表 3-2 User Flash の情報一覧

プリミティブ	サポートされるデバイス	アドレス	データ幅
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 JTAG 状態機械の Shift-IR 幅の調整

JTAG 状態機械の Shift-IR 幅の調整がサポートされています。この機能はデイジーチェーン・コンフィギュレーションでのみ使用されます。

図 3-16 JTAG 状態機械の Shift-IR 幅の調整



- IR length value : JTAG 状態機械の Shift-IR の幅。
- Save : 選択した値を FPGA に書き込みます。

## 3.14 リモートのコンフィギュレーションおよびプログラミング

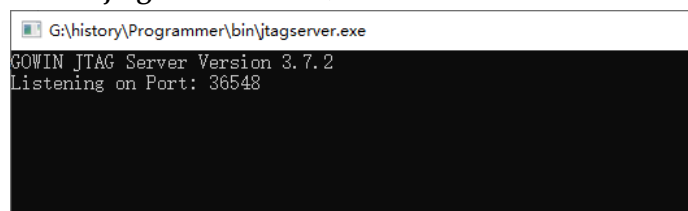
リモートのプログラミングおよびコンフィギュレーションがサポートされています。この機能を使用するには、2 台の PC 間で正常に TCP/IP 通信できることを確認する必要があります。

1. リモート PC 上で jtagserver.exe を起動する必要があります。プログラマと同じディレクトリで、jtagserver.exe ファイルをダブルクリックして開きます(図 3-17)。

図 3-17 jtagserver.exe の位置



図 3-18 jtagserver の起動状態



jtagserver は、クライアントとの接続をサポートするために指定ポートをリスンします。デフォルトのポートは 36548 です。他のポートを使用したい場合は、Programmer/bin/data/config/gvi.ini ファイル内でポート番号を変更してください。

図 3-19 jtagserver の起動状態

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

- host : リモートの IP アドレス。
  - port: リッスンされるポート。
2. ローカルの PC 上で programmer.exe を起動し、そのメニューバーの「Tools > Add GVC Server」をクリックして開きます。GVC の画面は図 3-21 に示す通りです。

図 3-20 Add GVC Server オプション

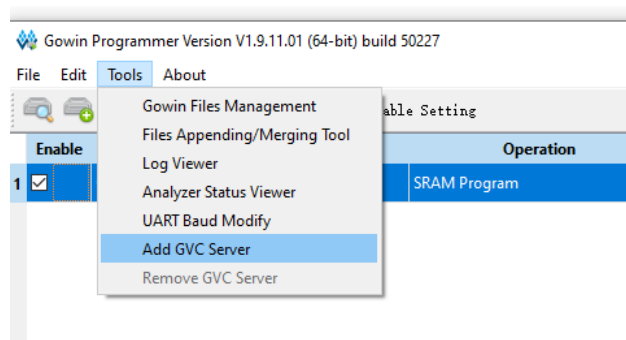
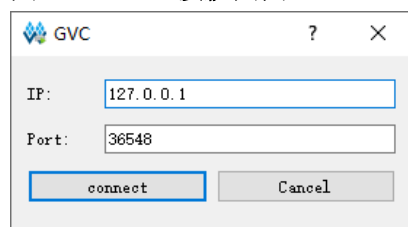


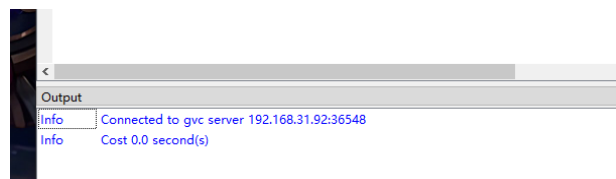
図 3-21 GVC 接続画面



- IP: リモート・サーバーの IP アドレス。
- Port : リモート・サーバーがリッスンするポート。
- Connect : リモート・サーバーに接続します。
- Cancel : 操作をキャンセルし、このページを終了します。

「Connect」をクリックした後、リモート・サーバーへの接続に成功すると、以下の画面がポップアップ表示されます(図 3-22)。

図 3-22 接続成功画面



接続に失敗した場合は、以下の画面がポップアップ表示されます(図 3-23)。

図 3-23 接続失敗画面

```
Info      Job quitting.
Error     Failed to connected to remote server 192.168.31.94:36548
Info      Cost 3.0 second(s)
```

3. 接続に成功した後は、すべてのリモート操作はローカル操作と同じです。ただし、リモート操作中に下図に示すリモート・デバッグ情報が出力されます。

図 3-24 リモート操作際の出力

```
Info      Utilizing GVC for remote debugging
Info      Target Cable: Gowin USB Cable(FI2CH)/0/6162/null@2MHz
Info      Target Device: GW1N-4B(0x1100381B)
Info      Operation "Reprogram" for device#1...
Info      Finished.
Info      Cost 0.22 second(s)
```

4. 操作が完了したら、「Tools > Remove GVC Server」オプションをクリックして、リモート・コンフィギュレーション・モードを終了できます。

図 3-25 リモート・コンフィギュレーションの終了

```
Info      Disconnected from remote server
Info      Cost 0.0 second(s)
```

表 3-3 各デバイスでサポートされるリモート・プログラミング/コンフィギュレーション機能

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



デバイス	機能
	'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 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'

デバイス	機能
	'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' '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	'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'

デバイス	機能
GW5ANT-15A GW5ANRT-15A	

## 3.15 ファイル変換機能

ファイル変換機能は、さまざまなファイル形式を変換するために使用されます。メニューバーの「Tools > Gowin Files Management」から「Gowin Files Management」ダイアログボックスを開きます(図 3-26)。

図 3-26 ファイル変換画面

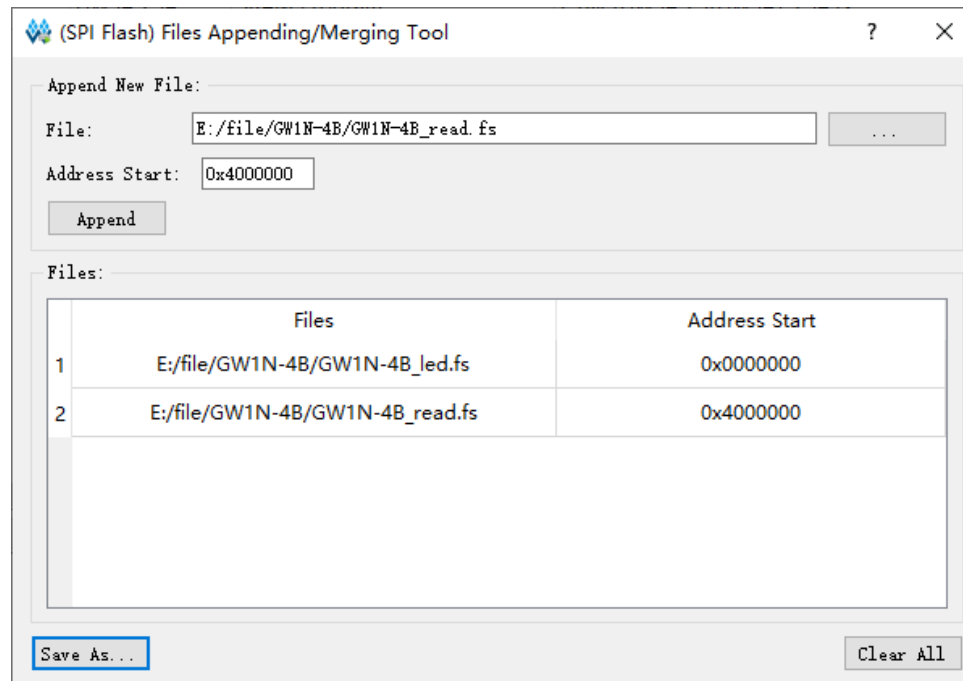


- Operation : 入力ファイルのタイプと変換後の対象ファイルのタイプを選択します。
- Clear : ファイル・ボックス内のすべてのファイルをクリアします。
- Add File(s) : 入力ファイルのタイプに一致するファイルを選択します。
- Convert/Merge : ファイル変換機能を実行します。

## 3.16 ビットストリーム・ファイル・マージ機能

指定された開始アドレスに従って、複数のビットストリーム・ファイル(.fs または .bin)を 1 つの bin ファイルにマージすることをサポートします。図 3-27 に示すように、メニューバーの「Tools > Files Appending/Merging Tool」から「(SPI Flash) Files Appending/Merging Tool」ダイアログボックスを開きます。

図 3-27 ビットストリーム・ファイル・マージ画面

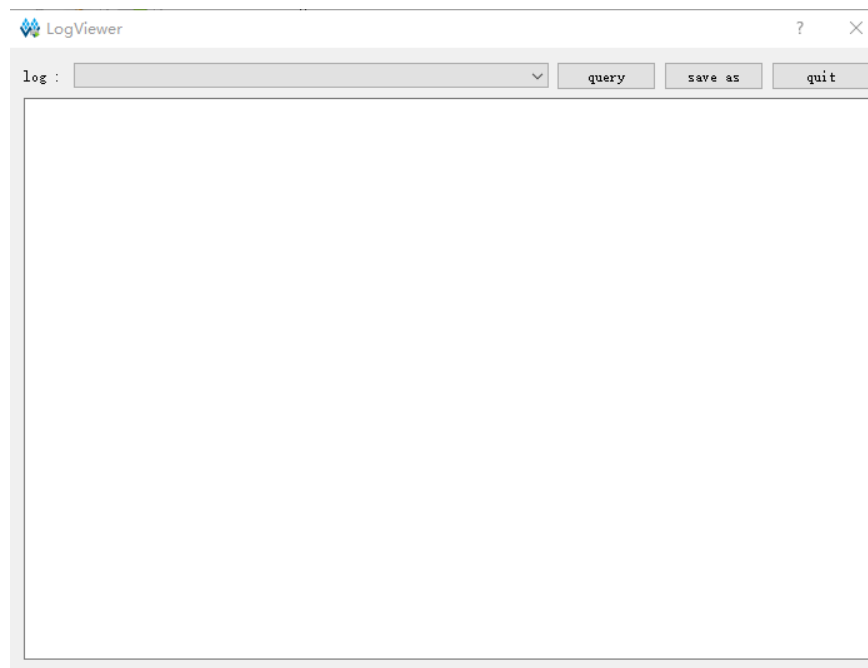


- **File(…)** : .fs ファイル(または.bin ファイル)を選択します。
- **Address Start** : アドレスを指定します。
- **Append** : 指定された fs ファイル(または.bin ファイル)と指定されたアドレスを下のリストに追加します。
- **Save As…** : リスト内の.fs ファイルを 1 つずつ.bin ファイルに変換し、指定されたアドレスに従って順番に新しい.bin ファイルに書き込みます。
- **Clear All** : リスト内のすべてのデータをクリアします。

## 3.17 ログ表示機能

ソフトウェアの実行中に情報バーに「**Error Found**」というプロンプトが表示されることは、ソフトウェアに実行エラーが発生したことを意味します。ログ表示機能により、トラブルシューティングのための詳細なエラー情報を取得できます。メニューバーの「**Tools > Log Viewer**」から「**LogViewer**」ダイアログボックスを開きます(図 3-28)。

図 3-28 ログ表示機能

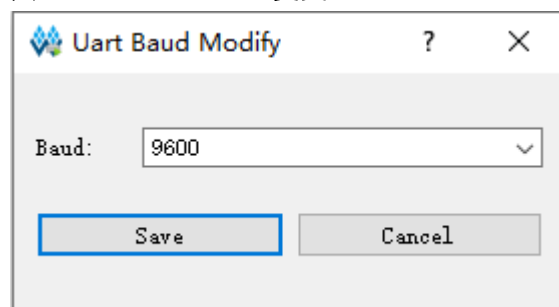


- **query** : xx.log ファイルを検索し、検索されたリストを左側のドロップダウン ボックスに表示します。
- **save as** : 以下のすべてのログ情報を新しいファイルとして保存します。
- **quit** : この画面を終了します。

## 3.18 ボーレート変更機能

ボーレートの変更をサポートします。図 3-29 に示すように、メニューバーの「Tools > Uart Baud Modify」から「Uart Baud Modify」を開きます。

図 3-29 ボーレートの変更



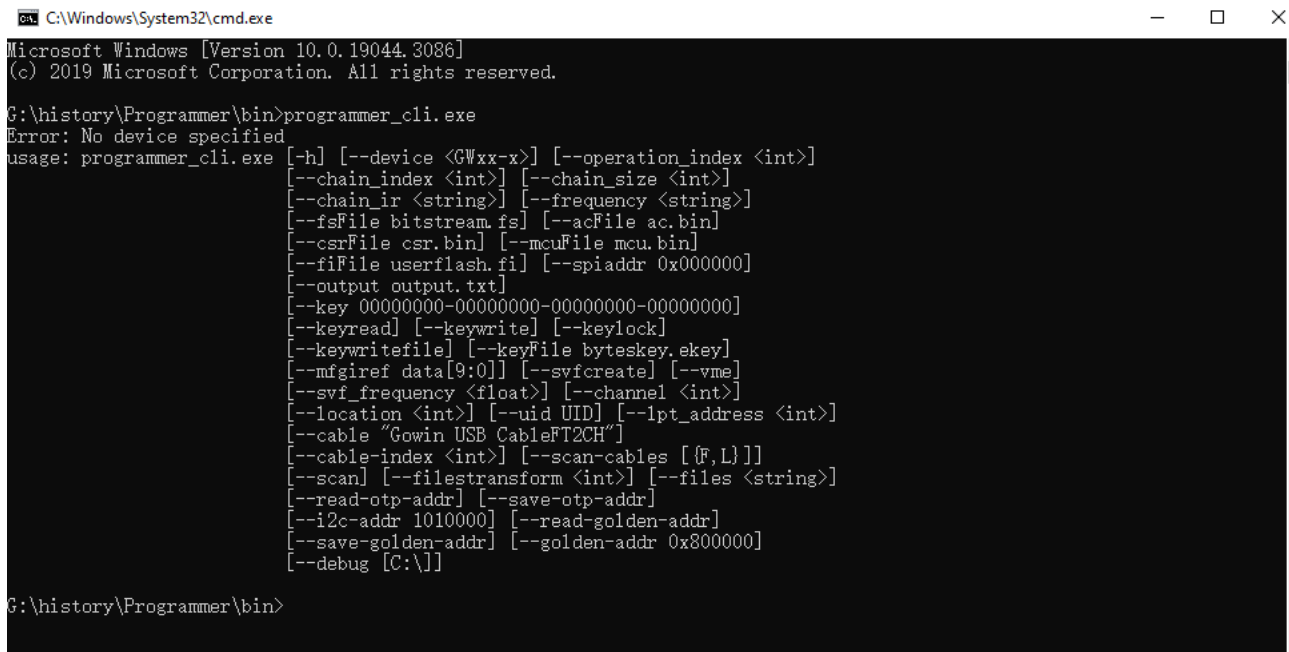
- **Save** : 変更したボーレートを保存します。
- **Cancel** : 今回の操作をキャンセルし、この画面を終了します。

# 4 Programmer\_cli のダウンロード手順

## 4.1 プレビューとヘルプ

CMD で `programmer_cli` ツールを開いた後、パラメーターが使用されていない場合、デバイスが指定されていないというメッセージと簡単なヘルプの説明が表示されます。

図 4-1 CMD で `programmer_cli` を開く



```
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] [--lpt_address <int>]
                        [--cable "Gowin USB CableFT2CH"]
                        [--cable-index <int>] [--scan-cables [{F,L}]]
                        [--scan] [--filestransform <int>] [--files <string>]
                        [--read-otp-addr] [--save-otp-addr]
                        [--i2c-addr 1010000] [--read-golden-addr]
                        [--save-golden-addr] [--golden-addr 0x800000]
                        [--debug [C:\]]

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

パラメータ `--help` を使用して詳細なヘルプ情報を取得することができます。

図 4-2 ヘルプ情報

```

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

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

```

```

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

```



```

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

```

## 4.2 USB Cable デバイスのスキャン

スキャンして、接続されている USB Cable の情報を表示します。

利用方法 : programmer\_cli.exe --scan-cables

図 4-3 USB Cable デバイスのスキャン

```

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)はケーブルタイプです。
- /0/は channel 番号です。
- 99091 は USB location です。

- /GW20LCRZ は USB Cable UID です。
- (USB location:99091) (SN: GW20LCRZ)は説明情報です。

## 4.3 USB Cable の種類とポートの指定

FPGA をスキャンまたはコンフィギュレーションする場合、パラメータ「`--cable-index`」を使用して USB Cable のタイプを指定できます。`-cable-index` を 0 として指定した場合、Cable 種類は Gowin USB Cable (GWU2X) です。`-cable-index` を 1 として指定した場合、Cable 種類は Gowin USB Cable (FT2CH) などです。デフォルトでは、`--cable-index` の値は 1 です。

FTDI タイプのケーブルなどのマルチポートケーブルを使用する場合は、複数のポートから選択できます。パラメータ「`--channel`」を使用してポートを指定できます。`--channel` のデフォルト値は 0 です。詳細なヘルプ情報を次の図に示します

図 4-4 ヘルプ情報

```
--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 USB Cable の場所または UID の指定

複数の USB Cable を使用する場合は、USB ポートまたは Cable UID を指定することで特定の USB Cable デバイスを指定できます。ヘルプ情報を次の図に示します。

図 4-5 ヘルプ情報

```
--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 FPGA デバイスのスキャン

次のコマンドでデバイスをスキャンします。

```
programmer_cli.exe - scan
```

図 4-6 FPGA デバイスのスキャン

```
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 Programmer 実行モードの指定

パラメータ「--operation\_index」、「--run」、または「-r」を使用して、SRAM のコンフィギュレーション、Flash のプログラミングなどの実行モードを指定します。ヘルプ情報を次の図に示します。

図 4-7 ヘルプ情報

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

FPGA をコンフィギュレーションするときは、パラメータ「**--device**」を使用して FPGA Device タイプを指定します。ヘルプ情報を次の図に示します。

図 4-8 ヘルプ情報

```
--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
```

SRAM または Flash をコンフィギュレーション・プログラミングするためには、通常次のコマンド形式が使用されます。

```
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** は JTAG 周波数を指定するために使用されます。現在、FTDI タイプのケーブルにのみ適用されます。U2X タイプのケーブルの周波数は 1.33MHz に固定されています。
- **--chain\_index <int>**、**--chain\_size <int>**、および **--chain\_ir <string>** を併用することにより、デイジーチェーン内のターゲットデバイスの位置を指定できます。
- **--chain\_index <int>** は、デイジーチェーン内のターゲットデバイスの位置を指定するために使用されます。例： **--chain\_index n** は、**n+1** 番目のデバイスを指定します。
- **--chain\_size <int>** は、デイジーチェーン内のデバイスの合計数を示すために使用されます。例： **--chain\_size n** は、デイジーチェーン内のデバイスの合計数が **n** であることを示します。
- **--chain\_ir <string>** は、デイジーチェーン内の JTAG 状態機械の IR の長さを指定するために使用されます。例： **--chain\_ir 8,8** は、デイジーチェーン内の両方のデバイスの IR の長さが **8** であることを意味します(デフォルトでは **8**)。
- **--location** は、ターゲットデバイスが配置されている USB ポートを指定するために使用され、優先度は **UID** よりも高くなります。
- **--uid** は、ターゲットデバイスが使用する USB Cable を指定するため

に使用されます。

- `--cable-index <int>` は、USB Cable の種類を指定するために使用されます。
- `--fsFile <bitstream.fs>` は、ビットストリーム・ファイルのパスを指定するために使用されます。
- `--run <int>` は `--operation_index` と同じで、実行モードを指定するために使用されます。
- `--device<GWxx-x>` はターゲットデバイス名を指定するために使用されます(大文字と小文字を区別)。

## 4.7 SRAM のコンフィギュレーション

ビットストリーム・ファイル、対応するデバイス、および SRAM コンフィギュレーション・モードを指定します。例えば、

SRAM Program を構成します。`--operation_index` パラメータの「SRAM Program」に対応する値は 2 であるため、この操作のコマンドは次のようになります。

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

`--cable-index` および `--location` が指定されていない場合、デフォルト値が使用されるため、省略できます。

例：

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

図 4-9 SRAM コンフィギュレーションの例

```
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)
```

パラメータの意味は次のとおりです：

- `--device GW1N-4B` はターゲット FPGA デバイスを指定します。デバイス名はスキャンの時に出力されます。
- `--fsFile d:\¥bitstream.fs` は、ビットストリーム・ファイル(`d:\¥bitstream.fs`)を指定するために使用されます。
- `--cable-index 1` は、USB Cable を「Gowin USB Cable(FT2CH)」として指定するために使用されます。

表 4-1 SRAM の場合の機能構成インデックス

インデックス	操作	説明
0	Read Device Codes	IDCode、UserCode、StatusCodeを含むFPGAコードを読み出します。
1	Reprogram	デバイスを再起動します
2	SRAM Program	SRAMのコンフィギュレーション
3	SRAM Program and Verify	SRAMをコンフィギュレーションして検証します
16	SRAM Program JTAG 1149	JTAGを介するピュアデータ書き込みモード。CRCチェックはサポートされません。暗号化または圧縮されたビットストリーム・ファイルはサポートされません
17	SRAM Program, Verify JTAG 1149	JTAGを介するピュアデータ書き込みモードでSRAMをコンフィギュレーションして検証します
24	SRAM Erase	SRAMのデータを消去します
73	Read User Code	FPGAデバイスのUserCodeを読み出します
74	Read Status Register	FPGAデバイスのStatus Codeを読み出します

## 4.8 LittleBee ファミリーFPGA の組み込み Flash のプログラミング

### 4.8.1 Flash のみのプログラミング

Flash をプログラムする例：

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

図 4-10 FlashFlash のみのプログラミングの例

```
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%
Programming...: [#####] 100%
Status Code is: 0x0001F020
User Code: 0x000054C9
Program Finished!
Finished.
Cost 3.18 second(s)
```

表 4-2 LittleBee ファミリーの場合の機能構成インデックス

インデックス	操作	説明
5	embFlash Erase,Program	オンチップFlashを消去した後、デー

インデックス	操作	説明
		タを書き込みます。
6	embFlash Erase,Program,Verify	オンチップFlashを消去した後、データを書き込み、検証を行います。
7	embFlash Erase Only	オンチップFlashのみを消去します
70	embFlash Background Erase,Program	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込みます。
71	embFlash Background Erase,Program,Verify	バックグラウンド・プログラミング。オンチップFlashを消去した後、データを書き込み、検証を行います。
72	embFlash Background Erase Only	バックグラウンド・プログラミング。オンチップFlashのみを消去します。

#### 4.8.2 Flash および UserFlash 初期化ファイルのプログラミング

パラメータ「--fiFile userflash.fi」を使用して UserFlash 初期化ファイルを指定し、Flash をプログラムすると同時に UserFlash をプログラムします。例えば、

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

## 4.9 オフチップ SPI Flash のプログラミング

### 4.9.1 LittleBee ファミリーと Arora ファミリー(Arora V を除く)のオフチップ SPI Flash のプログラミング

LittleBee ファミリーと Arora ファミリー(Arora V を除く)のオフチップ SPI Flash のプログラミングを構成する例：

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

図 4-11 オフチップ Flash のプログラミングの例

```
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/mul1@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)
```

表 4-3 LittleBee ファミリーと Arora ファミリーの場合の機能構成インデックス

インデックス	操作	説明
8	exFlash Erase,Program	オフチップ Flash を消去した後、データを書き込みます。
9	exFlash Erase,Program,Verify	オフチップ Flash を消去した後、データを書き込み、検証を行います。
10	exFlash Bulk Erase	オフチップ Flash を消去します。
11	exFlash Verify	オフチップ Flash のデータを検証します。
12	exFlash Erase,Program in bscan	bscan モードでオフチップ Flash を消去した後にデータを書き込みます。
13	exFlash Erase,Program,Verify in bscan	bscan モードでオフチップ Flash を消去した後にデータを書き込んで検証します。
14	exFlash Bulk Erase in bscan	bscan モードでオフチップ Flash を消去します。
15	exFlash Verify in bscan	bscan モードを使用して、オフチップ Flash のデータを検証します。
31	exFlash C Bin Erase,Program	オフチップ Flash を消去した後、RISC-V の bin ファイルをオフチップ Flash に書き込みます。
32	exFlash C Bin	オフチップ Flash を消去した後、



インデックス	操作	説明
	Erase,Program,Verify	RISC-V の bin ファイルをオフチップ Flash に書き込んで検証します。
36	exFlash Erase,Program thru GAO-Bridge	GAO-Bridge でオフチップ Flash を消去した後、データを書き込みます。
37	exFlash Erase,Program,Verify thru GAO-Bridge	GAO-Bridge でオフチップ Flash を消去した後、データを書き込み、検証を行います。
38	exFlash C Bin Erase,Program thru GAO-Bridge	GAO-Bridge でオフチップ Flash を消去した後、RISC-V の bin ファイルを書き込みます。
39	exFlash C Bin Erase,Program,Verify thru GAO-Bridge	GAO-Bridge でオフチップ Flash を消去した後、RISC-V の bin ファイルを書き込んで検証します。

## 4.9.2 GW2AN-9X/18X デバイスの内部 Flash のプログラミング

内部 Flash のプログラミングを構成する例：

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

図 4-12 GW2AN-9X/18X デバイスの内部 Flash のプログラミングの例

```
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)
```

表 4-4 GW2AN-9X/18X デバイスの場合の機能構成インデックス

インデックス	操作	説明
44	sFlash Erase,Program	内部 Flash を消去した後、データを書き込みます。
45	sFlash Erase,Program,Verify	内部 Flash を消去した後、データを書き込んで検証します。
46	sFlash Verify Only	内部 Flash のデータを検証します。
47	sFlash Bulk Erase	内部 Flash のデータを消去します。
48	sFlash Background Erase,Program	バックグラウンド・プログラミング。内部 Flash を消去した後、データを書き込みます。

インデックス	操作	説明
49	sFlash Background Erase,Program,Verify	バックグラウンド・プログラミング。内部 Flash を消去した後、データを書き込んで検証します。
50	sFlash Erase,Program,Verify thru GAO-Bridge	GAO-Bridge で内部 Flash を消去した後、データを書き込みます。
63	sFlash Background Erase,Program,Verify thru GAO-Bridge	GAO-Bridge(JTAG to SPI のブリッジ) によるバックグラウンド・プログラミング。内部 Flash を消去した後、データを書き込んで検証します。
64	sFlash Bulk Erase in bscan	bscan モードで内部 Flash を消去します
65	sFlash Erase,Program in bscan	bscan モードで内部 Flash を消去して書き込みます。

### 4.9.3 Arora V デバイスのオフチップ Flash のプログラミング

Arora V デバイスのオフチップ Flash のプログラミングを構成する例：

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

図 4-13 Arora V デバイスのオフチップ Flash のプログラミングの例

```
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/nul1@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%
Programming Flash starts from 0x000000.
Programming...: [#####] 100%
Programming Flash ends from 0x0B5300.
Program sFlash finished.
The device is reloading, please check later...
Finished.
Cost 17.62 second(s)
```

表 4-5 Arora V デバイスの場合の機能構成インデックス

インデックス	操作	説明
52	exFlash Bulk Erase Arora V	オフチップ Flash を消去します。
53	exFlash Erase,Program Arora V	オフチップ Flash を消去した後、データを書き込みます。
54	exFlash Erase,Program,Verify Arora V	オフチップ Flash を消去した後、データを書き込み、検証を行います。
55	exFlash C Bin Erase,Program Arora V	オフチップ Flash を消去した後、RISC-V の bin ファイルを書き込みます。
56	exFlash C Bin Erase,Program,Verify Arora V	オフチップ Flash を消去した後、RISC-V の bin ファイルを書き込んで検証します。
66	exFlash Verify Arora V	オフチップ Flash のデータを検証します。
67	exFlash Verify thru GAO-Bridge Arora V	GAO-Bridge でオフチップ Flash を検証します。
68	exFlash Erase,Program thru GAO-Bridge Arora V	GAO-Bridge でオフチップ Flash を消去した後、データを書き込みます。
69	exFlash Erase,Program,Verify thru GAO-Bridge Arora V	GAO-Bridge でオフチップ Flash を消去した後、データを書き込んで検証します。

## 4.10 I2C コンフィギュレーション

I2C コンフィギュレーションを構成する例：

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

図 4-14 I2C コンフィギュレーションの例

```
G:\history\window\x64\1.9.11.01 Alpha\20250310\programmer1.9.11.01 Alpha.Win64\Programmer\bin>programmer_cli.exe -d GW2AN-9X -r 57 --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 "I2C Program SRAM" for device#1...
Programming: [#####] 100%
Use fixed frequency 0.4MHz for I2C interface
Finished.
Cost 28.45 second(s)
```

表 4-6 I2C の場合の機能構成インデックス

インデックス	操作	説明
57	I2C Program SRAM	SRAM のコンフィギュレーション
58	I2C Program Flash	Flash のプログラミング
59	I2C Erase Flash Only	Flash の消去

## 4.11 MSPI 2nd Boot アドレスに対する操作

パラメータを構成することで、Arora V デバイスの Golden Image アドレス(範囲は 0x000000 ~ 0xF00000)の読み出しと書き込みを実現できます。

--read-golden-addr : Golden Image アドレスを読み出します

--save-golden-addr : Golden Image アドレスを書き込みます

--golden-addr : Golden Image アドレス

例：

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

図 4-15 MSPI 2nd Boot アドレスに対する操作の例

```
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)
```

## 4.12 リモートのコンフィギュレーションおよびプログラミング

パラメータを構成することで、リモートのコンフィギュレーションおよびプログラミングが可能になります。事前の準備については、[3.14 リモートのコンフィギュレーションおよびプログラミング](#)を参照してください



例：

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

図 4-18 SVF ファイル作成の例

```
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 ファイル変換機能

パラメータを構成することで、異なるファイル形式間で変換できます。

--filestransform：ファイル変換を実行します。

--files：変換するファイル。複数のファイルは「,」で区切ります。

例：

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

図 4-19 ファイル変換の例

```
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-25A.hex.
Cost 1.41 second(s)
```

## 4.16 I<sup>2</sup>C Slave アドレスに対する操作

パラメータを構成することで、I<sup>2</sup>C Slave アドレスの読み出しと書き込みが可能になります。現在、この操作は GW2AN-18X および GW2AN-9X デバイスのみでサポートされています。

--read-otp-addr：I<sup>2</sup>C Slave アドレスを読み出します

--save-otp-addr：I<sup>2</sup>C Slave アドレスを書き込みます

--i2c-addr：I<sup>2</sup>C Slave アドレス

例：

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

図 4-20 I<sup>2</sup>C Slave アドレスに対する操作の例

```
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)
```

