



Gowin_EMPU(GW1NS-4C) IDE リリースノート

IPUG928-2.1.1J, 2026-08-28

著作権について(2026)

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** 及び **Gowin** 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
2020/04/20	1.0J	初版。
2021/02/08	1.1J	● 周辺機器 AHB PSRAM Memory Interface のサポートを追加。 ● 周辺機器 AHB HyperRAM Memory Interface のサポートを追加。 ● 周辺機器 APB Nor Flash のサポートを追加。 ● GPIO が複数のポートタイプをサポートするように更新。 ● I²C が複数のポートタイプをサポートするように更新。 ● ソフトウェアバージョンと SDK をアップグレード。
2022/12/16	1.2J	SDK を更新。
2024/03/14	2.0J	ソフトウェア・リファレンス・デザインを更新。
2025/08/5	2.1J	● 関連ソフトウェアのバージョン情報を更新。 ● GMD の V2025.01 バージョンに適合するようにリファレンス・デザインを更新。 ● Code Coverage 機能と Profiling 機能のリファレンス・デザインを追加。
2026/08/28	2.1.1J	__STARTUP_CLEAR_BSS の説明を追加。

目次

目次	i
図一覧	ii
1 ARM Keil MDK	1
1.1 ソフトウェアのインストール	1
1.2 プロジェクトテンプレート	1
1.2.1 プロジェクトの作成	1
1.2.2 コンパイルオプションの構成	2
1.2.3 コンパイル	6
1.2.4 ダウンロード	6
1.2.5 オンラインデバッグ	8
1.3 リファレンス・デザイン	13
2 GMD ソフトウェア	14
2.1 ソフトウェアのインストール	14
2.2 プロジェクトテンプレート	14
2.2.1 プロジェクトの新規作成	14
2.2.2 オプションの構成	17
2.2.3 コンパイル	22
2.2.4 ダウンロード	23
2.2.5 オンラインデバッグ	26
2.3 リファレンス・デザイン	33

図一覧

図 1-1 プロジェクトの新規作成	2
図 1-2 デバイスの構成	2
図 1-3 ROM と RAM の構成	3
図 1-4 出力されるファイルの形式の構成	4
図 1-5 ヘッダーファイルのパスの構成	5
図 1-6 Flash オプションの構成	5
図 1-7 コンパイル	6
図 1-8 GW1NS-4C/GW1NSR-4C のダウンロード・オプションの構成	7
図 1-9 GW1NSER-4C のダウンロード・オプションの構成	8
図 1-10 GW1NS-4C/GW1NSR-4C の場合の構成	9
図 1-11 GW1NSER-4C の場合の構成	10
図 1-12 エミュレーターの構成	11
図 1-13 デバッグの構成	12
図 1-14 デバッグの開始	13
図 2-1 プロジェクトの新規作成	15
図 2-2 構成のタイプの選択	16
図 2-3 ツールチェーンおよびそのパスの選択	17
図 2-4 Target Processor の構成	18
図 2-5 GNU ARM Cross Assembler > Preprocessor の構成	19
図 2-6 GNU ARM Cross C Compiler > Includes の構成	20
図 2-7 GNU ARM Cross C Linker の構成	21
図 2-8 GNU ARM Cross Create Flash Image の構成	22
図 2-9 Programmer	24
図 2-10 GW1NS-4C/GW1NSR-4C のダウンロード・オプションの構成	25

図 2-11 GW1NSER-4C のダウンロード・オプションの構成.....	26
図 2-12 デバッグレベルの構成	27
図 2-13 デバッグ・ビューの作成.....	28
図 2-14 Main ビューの構成	29
図 2-15 Debugger ビューの構成	30
図 2-16 Startup ビューの構成	31
図 2-17 GW1NS-4C/GW1NSR-4C の場合の構成	32
図 2-18 GW1NSER-4C の場合の構成.....	32
図 2-19 デバッグの開始	33

1 ARM Keil MDK

1.1 ソフトウェアのインストール

ARM Keil MDK(V5.26 以降)のホームページ上の [Getting Started with MDK](#) を参照してください。

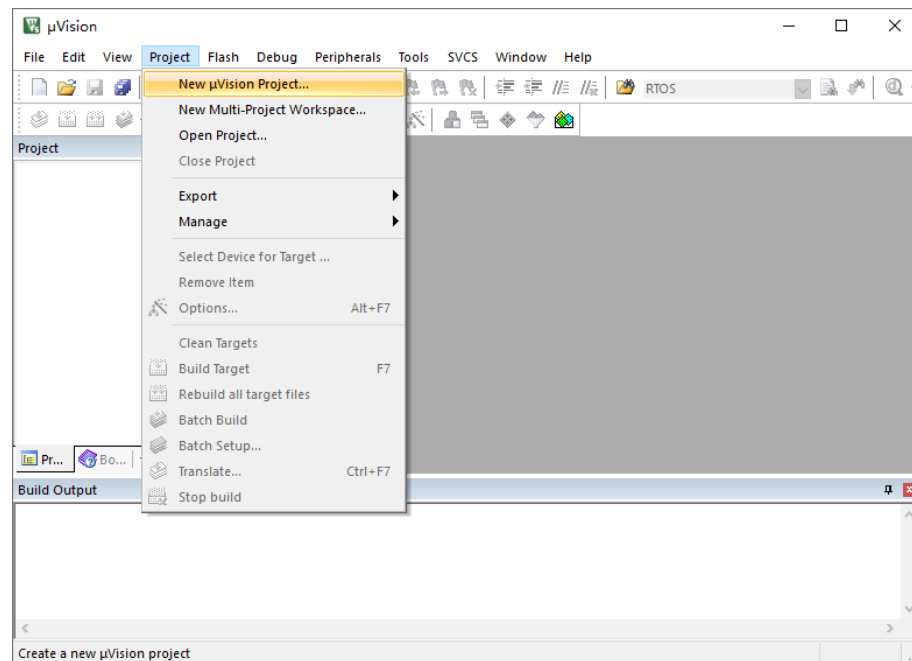
1.2 プロジェクトテンプレート

ARM Keil MDK を利用した Gowin_EMPU(GW1NS-4C)ソフトウェアプログラミングには、プロジェクトの作成、コンパイルオプションの構成、コーディング、コンパイル、ダウンロード、およびオンラインデバッグが含まれます。

1.2.1 プロジェクトの作成

ARM Keil MDK を開いたあと、メニューバーで「Project > New uVision Project…」を選択し、新しいプロジェクトを作成します(図 1-1)。

図 1-1 プロジェクトの新規作成

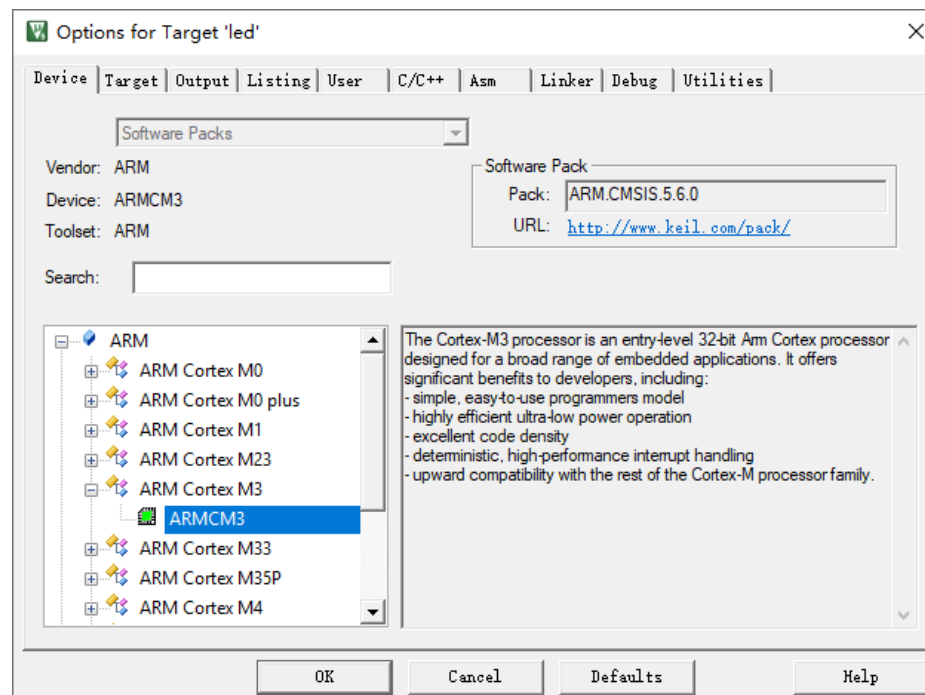


1.2.2 コンパイルオプションの構成

デバイスの構成

Gowin_EMPU(GW1NS-4C)には ARM Cortex-M3 MCU が組み込まれているため、「ARM Cortex-M3 > ARMCM3」を選択します(図 1-2)。

図 1-2 デバイスの構成



ROM と RAM の構成

ROM(命令メモリ、Flash)および RAM(データメモリ、SRAM)の開始アドレスとサイズを構成します(図 1-3)。

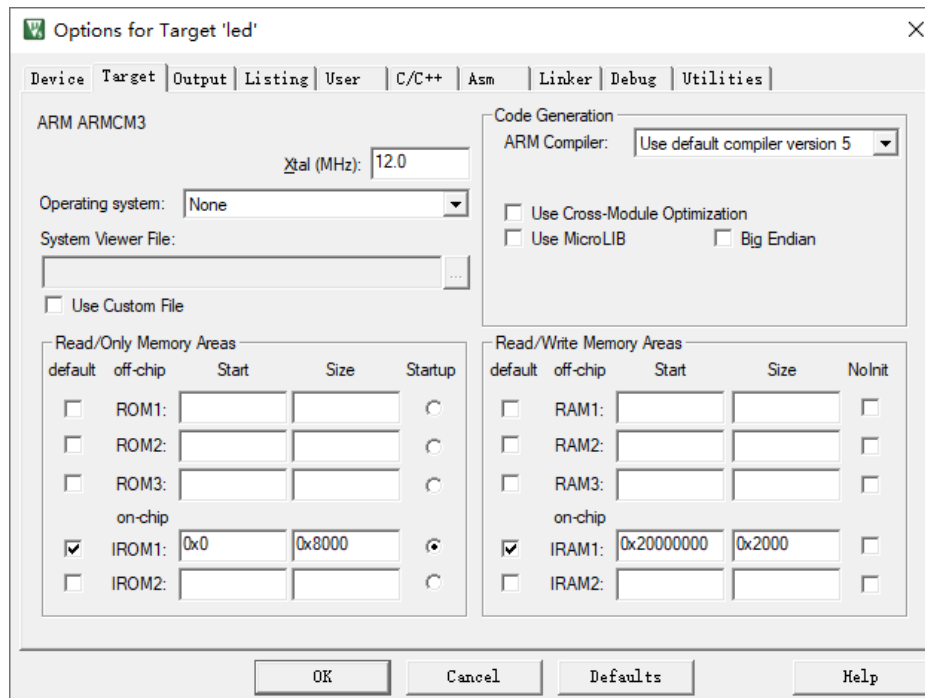
ROM の構成 :

- 開始アドレス : 0x00000000
- Size : 0x8000 (32KB)

RAM の構成 :

- 開始アドレス : 0x20000000
- Size : Gowin ソフトウェア > IP Core Generator > Gowin_EMPU(GW1NS-4C)IP コアの SRAM Size の構成と一致する必要があります。2KB、4KB、8KB、または 16KB がサポートされます。

図 1-3 ROM と RAM の構成



出力されるファイルの形式の構成

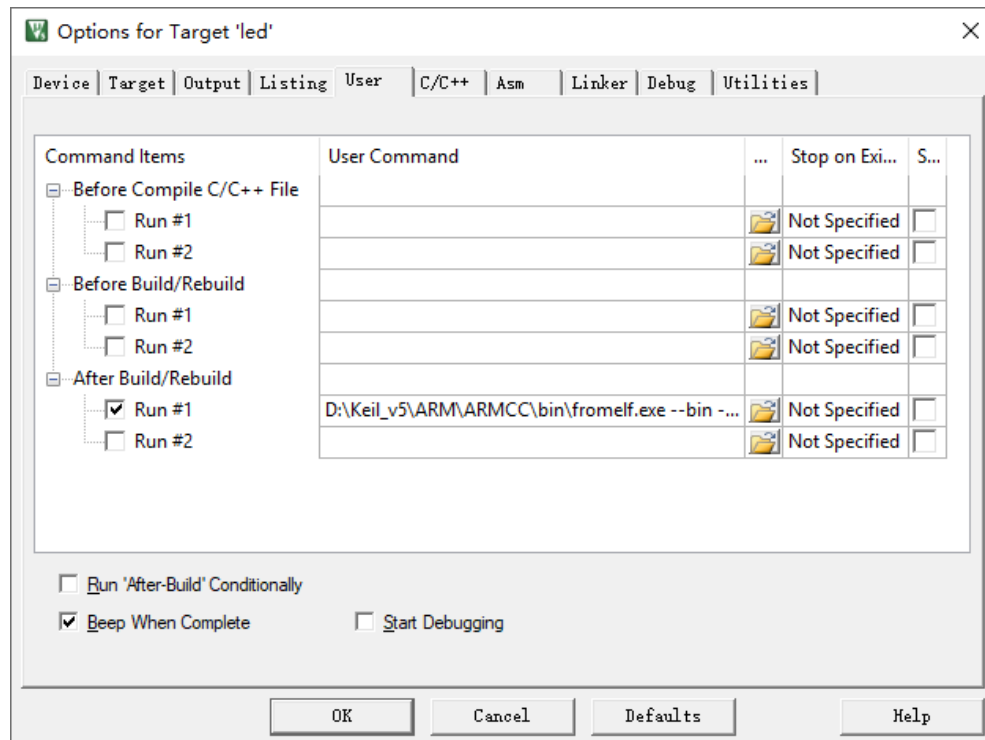
Gowin_EMPU(GW1NS-4C)ソフトウェア設計では実行ファイルを生
成する必要があるため、.axf ファイルを .bin ファイルに変換する必要があ
ります。

User タブで、形式変換ツールの fromelf を使用して axf ファイルを bin
ファイルに変換することができます(図 1-4)。

fromelf の形式変換コマンド : fromelf.exe --bin -o bin-file axf-file。例えば、C:\Keil_v5\ARM\ARMCC\bin\fromelf.exe --bin -o led.bin .\Objects\led.axf。

fromelf のインストールパスは、ARM Keil MDK のインストールパスによって異なります。

図 14 出力されるファイルの形式の構成



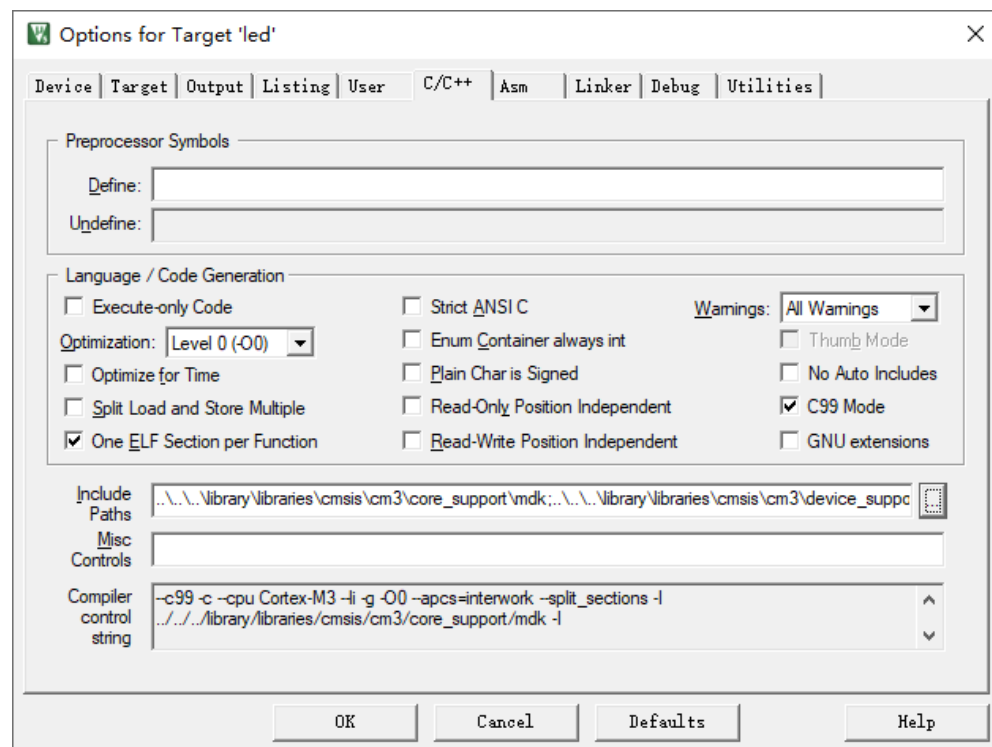
ヘッダーファイルのパスの構成

コンパイル中に C ヘッダーファイルを呼び出すために使用される C ヘッダーファイルのパスを構成します(図 1-5)。

例 :

- "..\..\..\library\libraries\cmsis\cm3\core_support\mdk"
- "..\..\..\library\libraries\cmsis\cm3\device_support"
- "..\..\..\library\libraries\drivers\inc"
- "..\..\..\library\middlewares\delay"
- "..\..\..\library\middlewares\gpio"
- "..\inc"

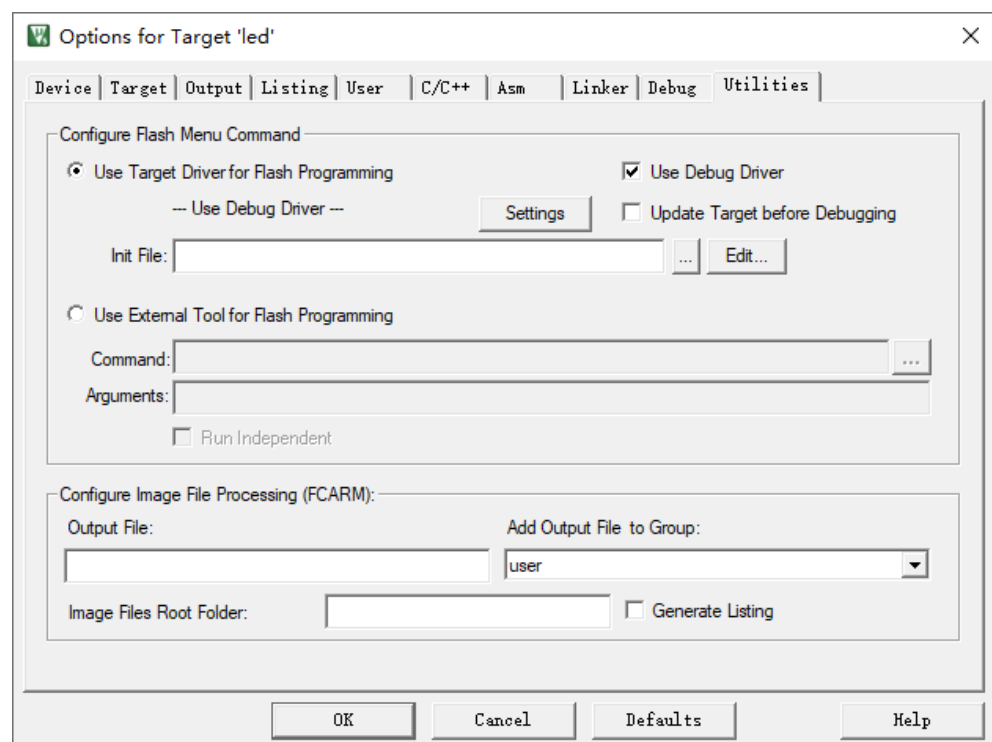
図 1-5 ヘッダーファイルのパスの構成



Flash オプションの構成

オンラインでデバッグする必要がある場合は、「Utilities > Update Target before Debugging」オプションをオフにしてください(図 1-6)。

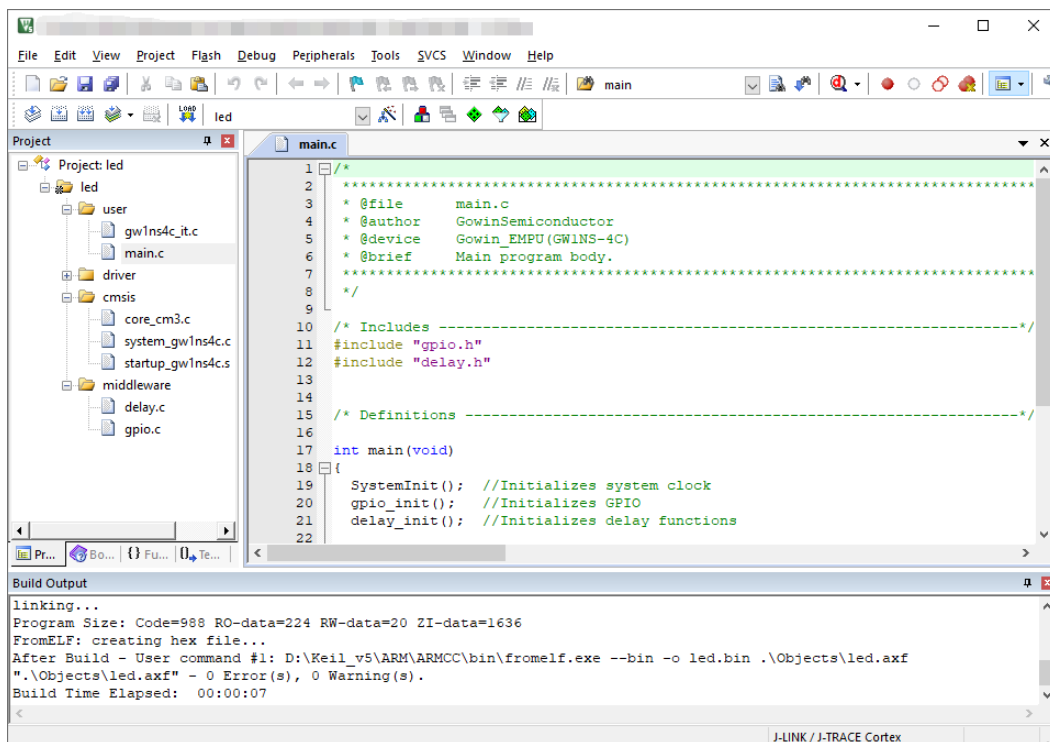
図 1-6 Flash オプションの構成



1.2.3 コンパイル

コーディングとプロジェクトの構成が完了したら、ツールバーの **Build**  もしくは **Rebuild** 、またはメニューバーの「Project > Build Target」もしくは「Project > Rebuild all target files」をクリックし、コンパイルして実行ファイルを生成します(図 1-7)。

図 1-7 コンパイル



1.2.4 ダウンロード

Programmer を使用してソフトウェア設計の実行ファイルをダウンロードします。

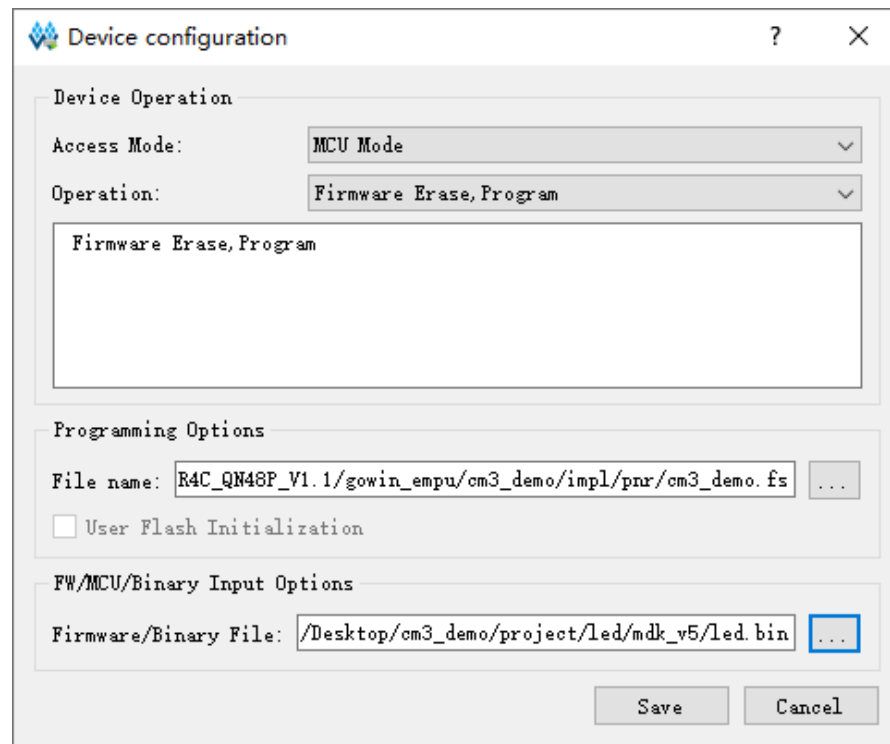
Gowin ソフトウェアまたはインストールパスから **Programmer** を開きます。

「Programmer > Edit > Configure Device」またはツールバーの「Configure Device 」をクリックして、「Device configuration」ダイアログボックスを開きます。

GW1NS-4C または GW1NSR-4C の場合、ダウンロード・オプションの構成は図 1-8 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「MCU Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Firmware Erase, Program」または「Firmware Erase, Program, Verify」を選択します。

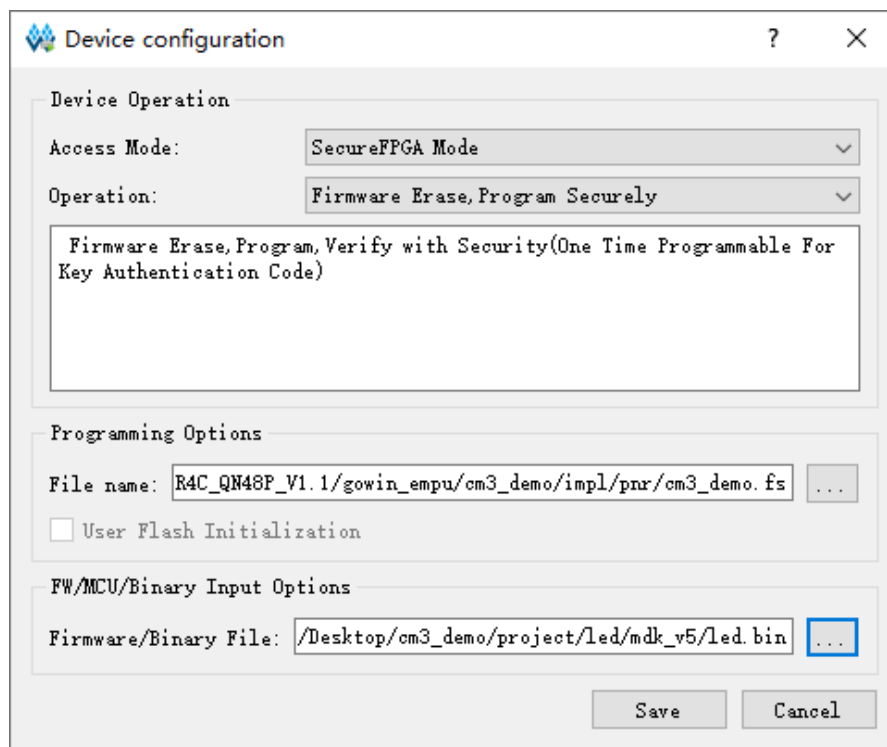
図 1-8 GW1NS-4C/GW1NSR-4C のダウンロード・オプションの構成



GW1NSR-4C の場合、ダウンロード・オプションの構成は図 1-9 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「SecureFPGA Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Firmware Erase, Program Securely」を選択します。

図 1-9 GW1NSER-4C のダウンロード・オプションの構成



- 「FW/MCU/Binary Input Options > Firmware/Binary File」でソフトウェア設計の実行ファイルを選択します。
- 最後に「Save」をクリックします。

注記：

「Programming Options > File name」でビットストリームファイルをインポートします。
『Gowin_EMPU(GW1NS-4C)ハードウェア設計ユーザーガイド([IPUG932](#))』を参照してください。

Device configuration が完了したら、Programmer のツールバー上の「Program/Configure」()をクリックしてソフトウェア設計の実行ファイルとハードウェア設計のビットストリームファイルをダウンロードします。

1.2.5 オンラインデバッグ

ソフトウェア設計の実行ファイルとハードウェア設計のビットストリームファイルのダウンロードが完了した後、ユーザーのソフトウェア設計に問題がある場合は、U-LINK または J-LINK などエミュレーターに接続してオンラインでデバッグすることができます。

デバッグのプロセスは次のとおりです：

- MCU JTAG モードの切り替え
- MCU JTAG インターフェースの切り替え

- デバッグオプションの構成
- エミュレーターへの接続
- デバッグの開始

MCU JTAG モードの切り替え

Programmer において MCU JTAG モードをダウンロードモードからデバッグモードに切り替えます。

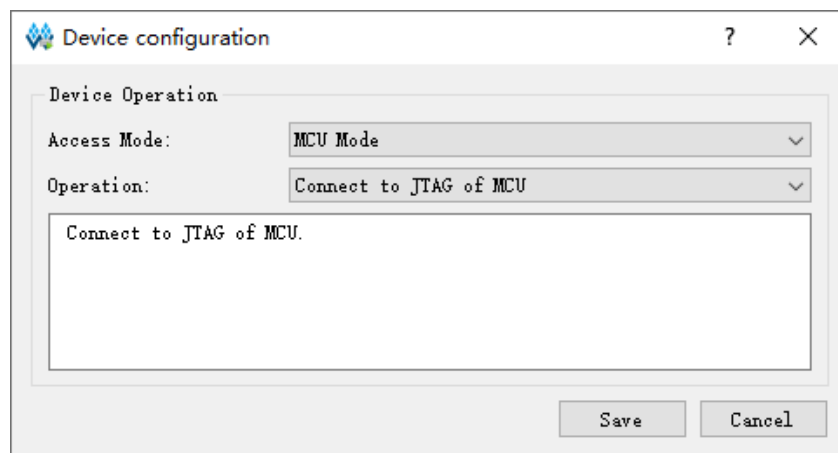
Gowin ソフトウェアまたはインストールパスから Programmer を開きます。

Programmer のメニューバーの Edit > Configure Device またはツールバーのをクリックして、Device configuration ダイアログボックスを開きます。

GW1NS-4C または GW1NSR-4C の場合の構成を図 1-10 に示します。

- 「Access Mode」のドロップダウン・リストから「MCU Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Connect to JTAG of MCU」を選択します。

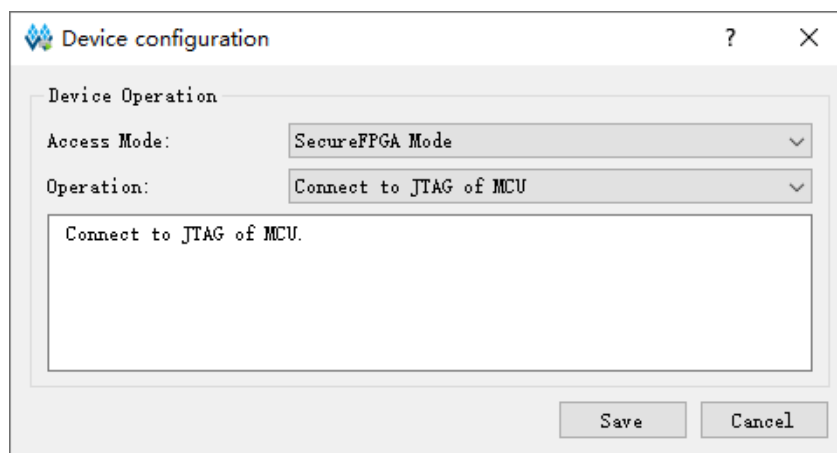
図 1-10 GW1NS-4C/GW1NSR-4C の場合の構成



GW1NSR-4C の場合の構成を図 1-11 に示します。

- 「Access Mode」のドロップダウン・リストから「SecureFPGA Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Connect to JTAG of MCU」を選択します。

図 1-11 GW1NSER-4C の場合の構成



- 「Save」をクリックして MCU JTAG モードの切り替えを完了します。

Device configuration が完了したら、Programmer のツールバーの Program/Configure 「」をクリックして JTAG モードの切り替えを完了します。

MCU JTAG インターフェースの切り替え

例えば、DK-START-GW1NSR4C-QN48P V1.1 開発ボードの場合、開発ボード上の SW3、SW4、SW5、SW6 DIP スイッチ(JTAG インターフェースの TCK、TDO、TDI、TMS)を手動でダウンロード側からデバッグ側に切り替えます。

注記：

- MCU JTAG モードの切り替えおよび MCU JTAG インターフェースの切り替え中は、開発ボードの電源をオフにしてはなりません。
- 開発ボードが途中でパワーオフした場合、電源を再投入すると自動的に MCU JTAG ダウンロードモードに戻ります

デバッグオプションの構成

ARM Keil MDK でツールバー > Options for Target… 「」をクリックし、「Debug」タブを構成します。

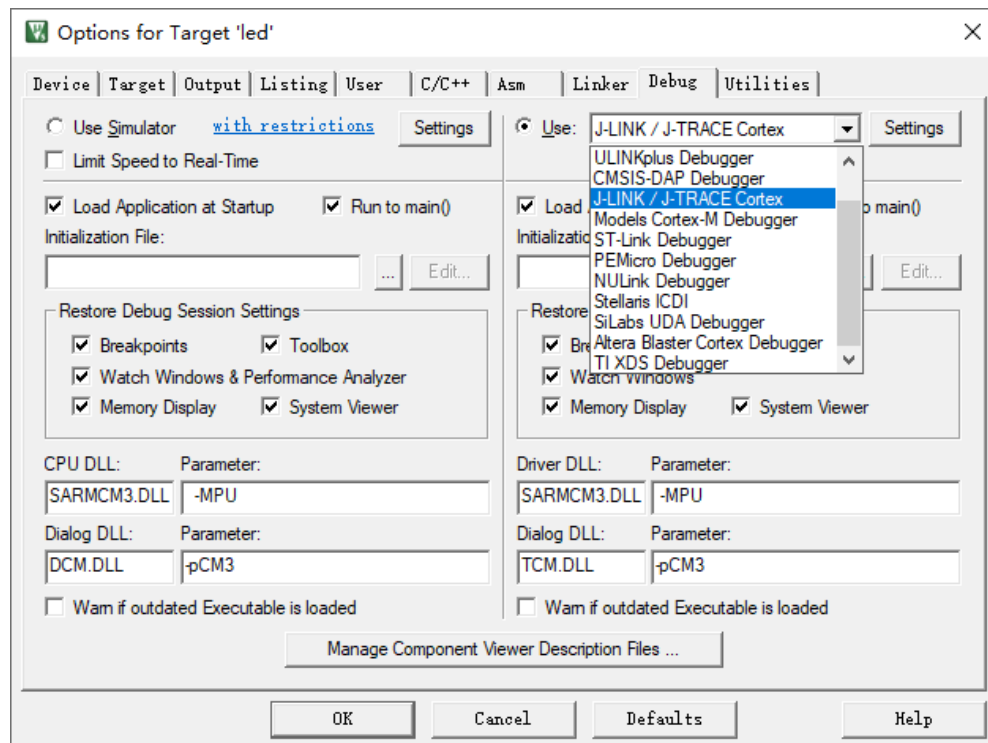
- エミュレーターの選択

図 1-12 に示すようにエミュレーターを選択します。

- U-LINK エミュレーター
U-LINK エミュレーターを選択した場合、「ULNK2/ME Cortex Debugger」が使用されます。
- J-LINK エミュレーター
J-LINK エミュレーターを選択した場合、「J-LINK/J-TRACE Cortex」

が使用されます。

図 1-12 エミュレーターの構成

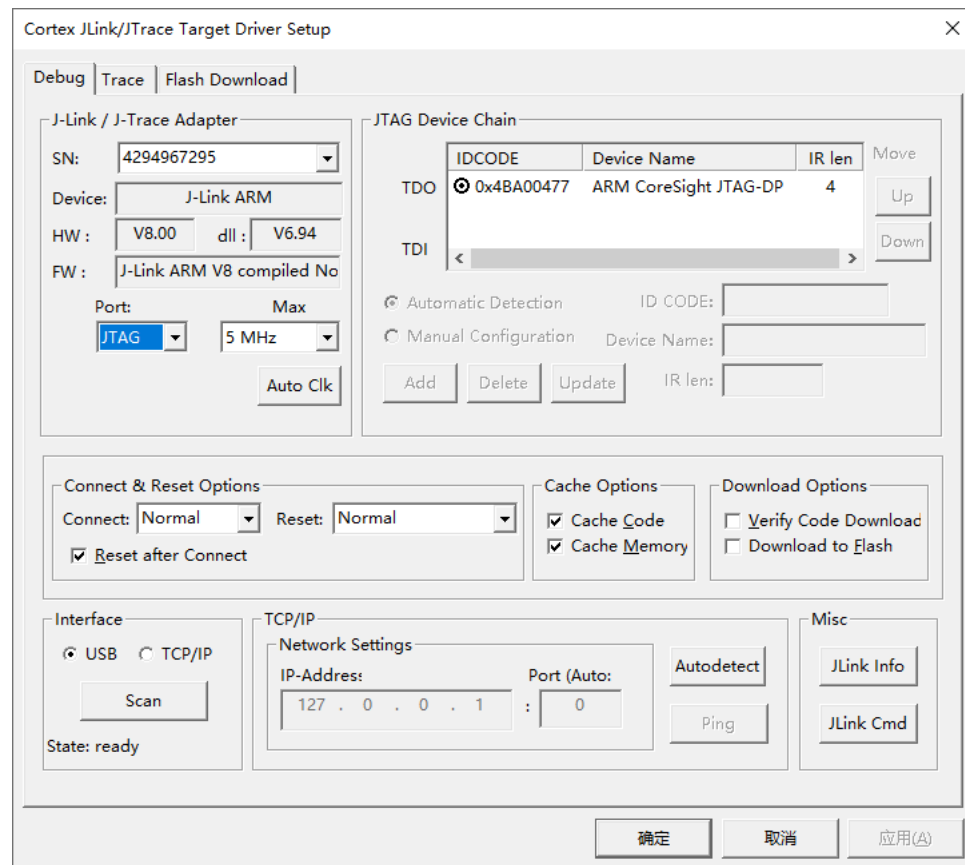


- デバッグポートのタイプの構成
Settings をクリックすると、J-LINK が選択された場合、Cortex JLink/JTrace Target Driver Setup が開き、U-LINK が選択された場合、ULINK2/ME Cortex-M Target Driver Setup が開きます。

J-LINK エミュレーターのデバッグの構成画面を図 1-13 に示します。

- JTAG デバッグポート
サポートします。
- SW デバッグポート
サポートしません。

図 1-13 デバッグの構成



Download Options で、Verify Code Download と Download to Flash の 2 つのオプションをオフにします。

上記の手順が完了したあと、エミュレーターに接続すると、Gowin_EMPU(GW1NS-4C) の IDCODE、Device Name などの情報が JTAG Device Chain に表示されます。

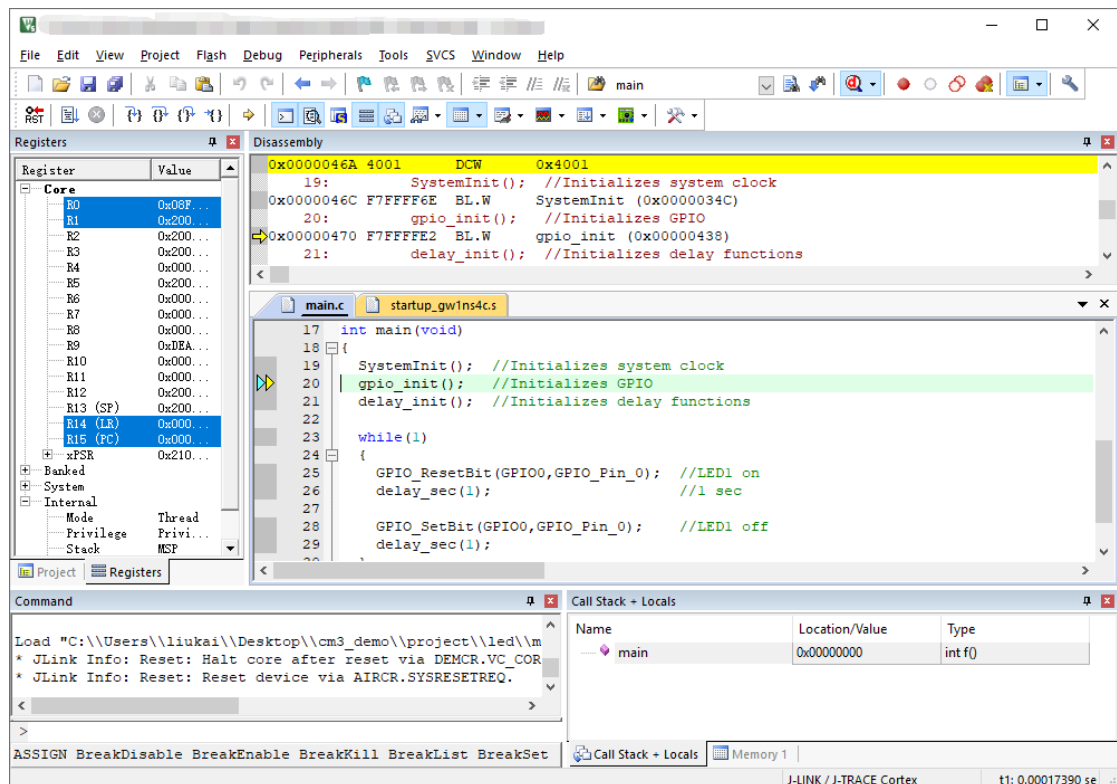
デバッグの開始

デバッグオプションの構成が完了したら、U-LINK または J-LINK エミュレーターを接続します。

ツールバー上の Debug 「」、またはメニューバー上の「Debug > Start/Stop Debug Session」をクリックしてデバッグモードに入ります。

ブレークポイントの設定、シングルステップ・デバッグ、リセット、および実行などの操作が可能です(図 1-14)。

図 1-14 デバッグの開始



1.3 リファレンス・デザイン

ARM Keil MDK(テスト済みバージョン : V5.26)を使用したリファレンス・デザイン:[cdn.gowinsemi.com.cn/Gowin EMPU\(GW1NS-4C\) V2.1.zip](http://cdn.gowinsemi.com.cn/Gowin_EMPU(GW1NS-4C)_V2.1.zip)

…¥ref_design¥MCU_RefDesign¥MDK_RefDesign¥cm3_demo、cm3_freertos、cm3_rttthread_nano、cm3_ucos_iii

2 GMD ソフトウェア

2.1 ソフトウェアのインストール

GMD のインストールパッケージは、GOWIN のホームページからダウンロードできます：<https://www.gowinsemi.com/ja/support/database/747/>。

注記：

GMD のインストールと構成については、『GOWIN MCU Designer ユーザーガイド (SUG549)』を参照してください。

2.2 プロジェクトテンプレート

GMD を利用した Gowin_EMPU(GW1NS-4C)ソフトウェアプログラミングには、プロジェクトの作成、コンパイルオプションの構成、コーディング、コンパイル、ダウンロード、およびオンラインデバッグが含まれます。

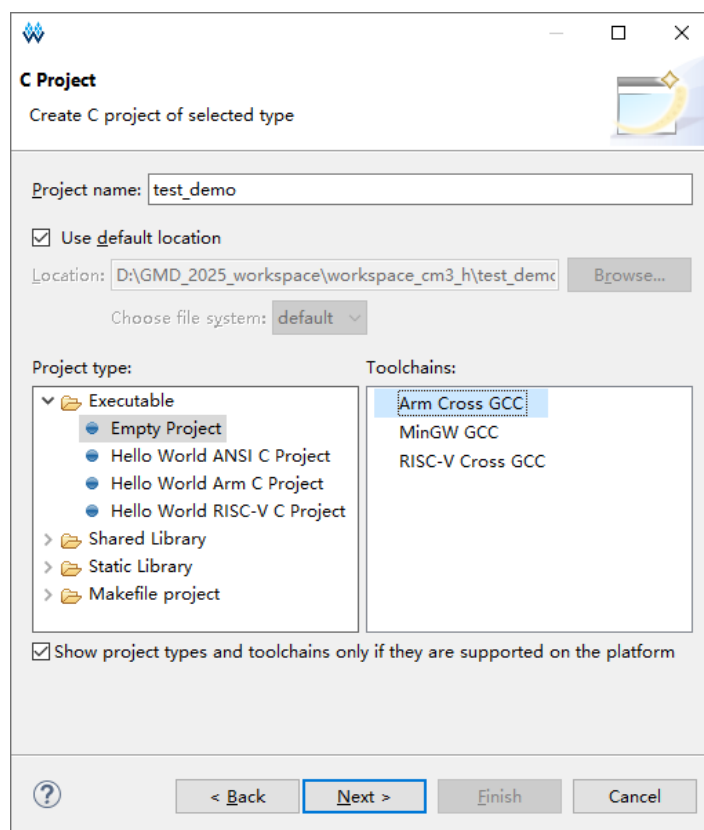
2.2.1 プロジェクトの新規作成

プロジェクトの作成

ツールバー上の「New」() またはメニューバー上の「File > New > C Project」を選択します(図 2-1)。

- プロジェクトの名前とパスを構成します。
- プロジェクトのタイプ(Empty Project)を選択します。
- コンパイルツールチェーン(Arm Cross GCC)を選択します。

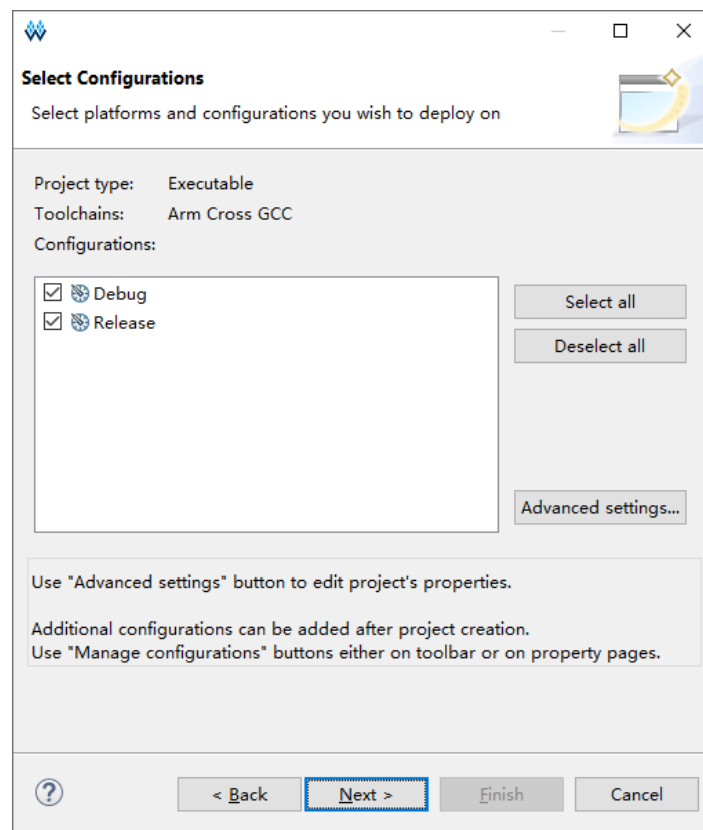
図 2-1 プロジェクトの新規作成



構成のタイプの選択

構成のタイプとして **Debug** と **Release** を選択します(図 2-2)。

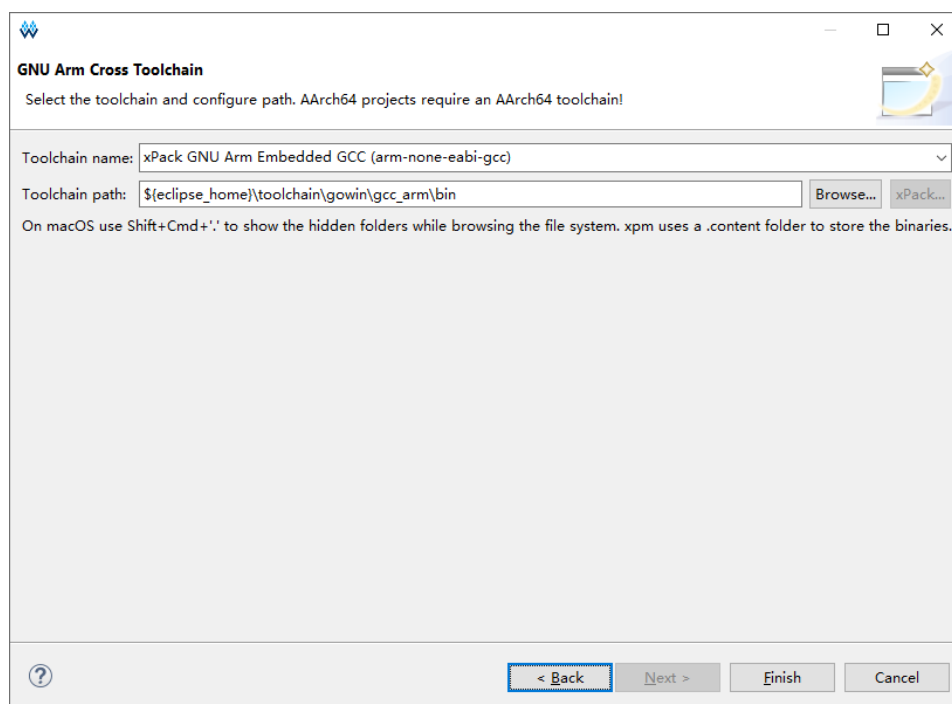
図 2-2 構成のタイプの選択



ツールチェーンおよびそのパスの選択

クロスコンパイル・ツールチェーンとそのパスを選択します。GMD ソフトウェアはツールチェーンを統合し、パスを構成しています。したがって、通常はデフォルトの選択(`arm-none-eabi-gcc`)で問題ありません(図 2-3)。

図 2-3 ツールチェーンおよびそのパスの選択



プロジェクトの作成

ソフトウェア・プロジェクトを新規作成した後、**Project Explorer** ビューで新規プロジェクトを選択してプロジェクトの構造とコードを追加し、ソフトウェア設計をインポートします。

Project Explorer ビューで現在のプロジェクトを選択し、右クリックして「**Refresh**」オプションを選択すると、現在のプロジェクトの構造とコードが自動的に更新されます。

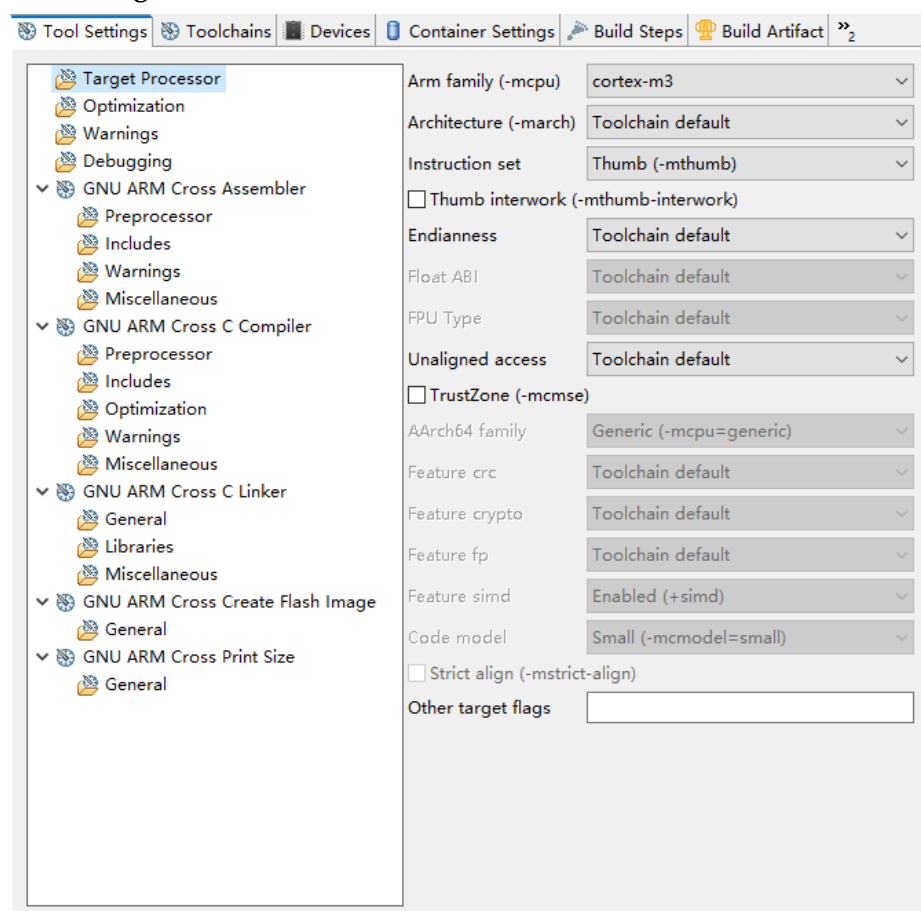
2.2.2 オプションの構成

Project Explorer ビューで、現在のプロジェクトを選択し、右クリックして「**Properties > C/C++ Build > Settings**」を選択してプロジェクトのパラメーターを構成します。

Target Processor の構成

「**Target Processor > Arm family**」では「**cortex-m3**」を選択します(図 2-4)。

図 2-4 Target Processor の構成

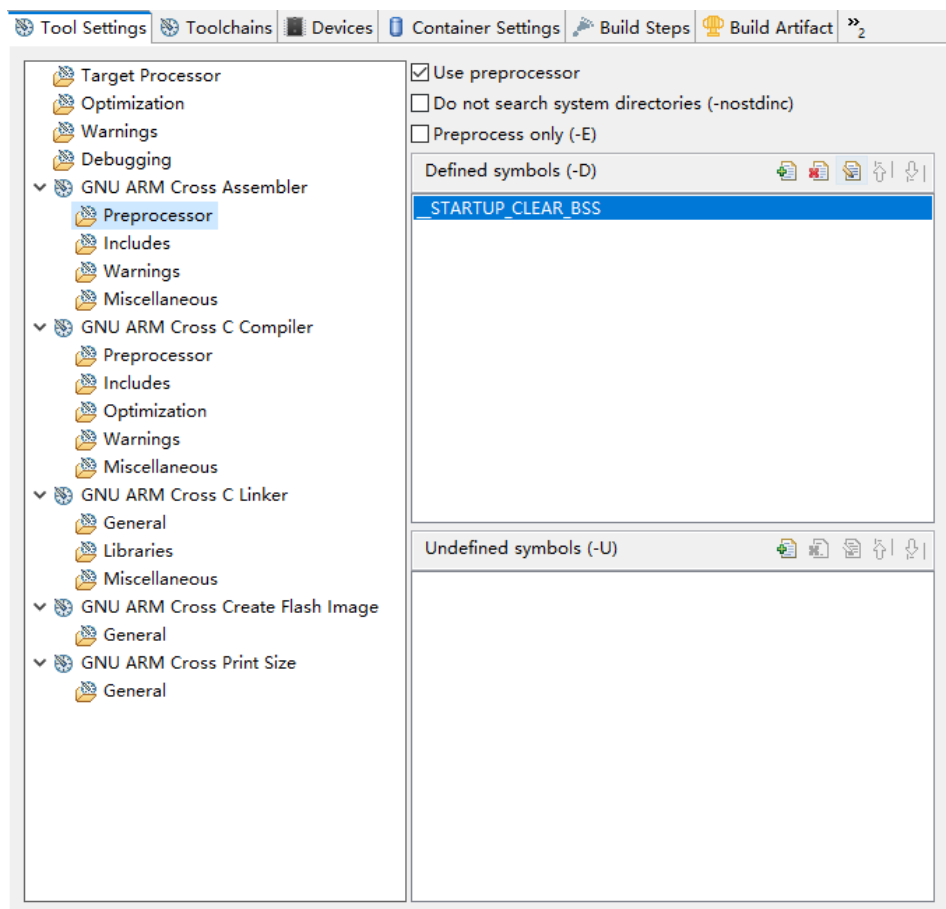


GNU ARM Cross Assembler > Preprocessor の構成

「GNU ARM Cross Assembler > Preprocessor > Defined symbols (-D)」では「`__STARTUP_CLEAR_BSS`」を選択します(図 2-5)。

「`__STARTUP_CLEAR_BSS`」は、MCU の起動時に、bss セクションをクリアする処理を有効にするために使用します。これにより、明示的に初期化されていないグローバル変数および静的変数が、MCU のリセット後に正しく 0 に初期化されることを保証します。リセット後にアプリケーションを正常に再起動する必要がある場合は、このマクロを定義することを推奨します。

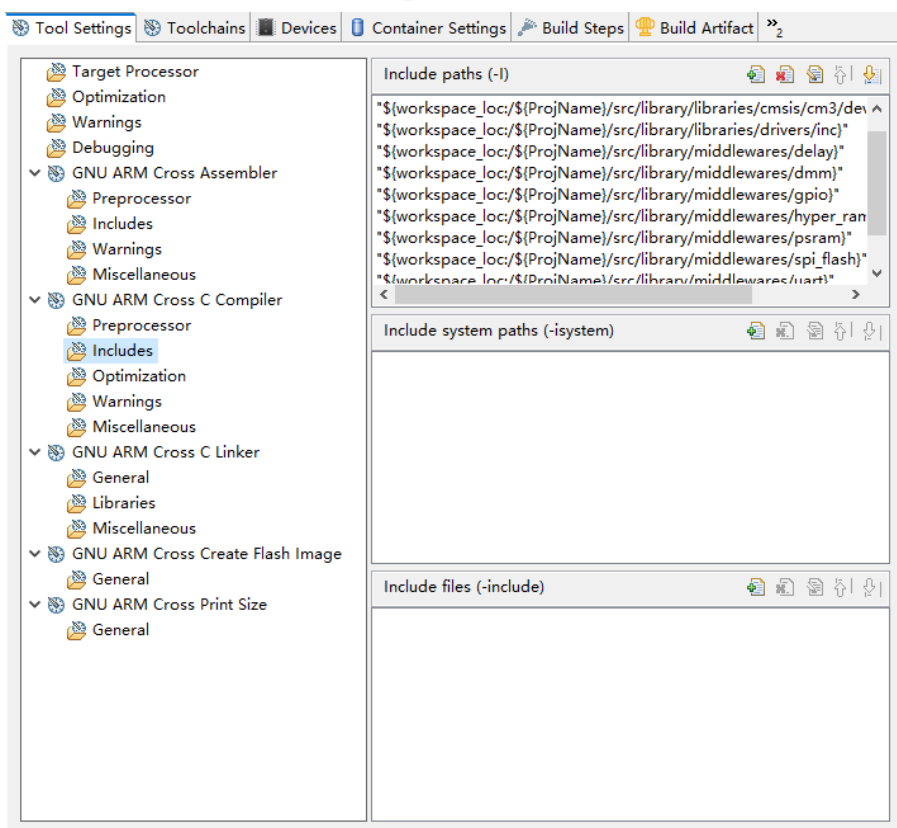
図 2-5 GNU ARM Cross Assembler > Preprocessor の構成



GNU ARM Cross C Compiler > Includes の構成

「GNU ARM Cross C Compiler > Includes > Include paths (-I)」オプションを現在のプロジェクトの C ヘッダーファイルのインクルードパスとして構成します(図 2-6)。

図 2-6 GNU ARM Cross C Compiler > Includes の構成



例えば、ソフトウェア・リファレンス・デザイン

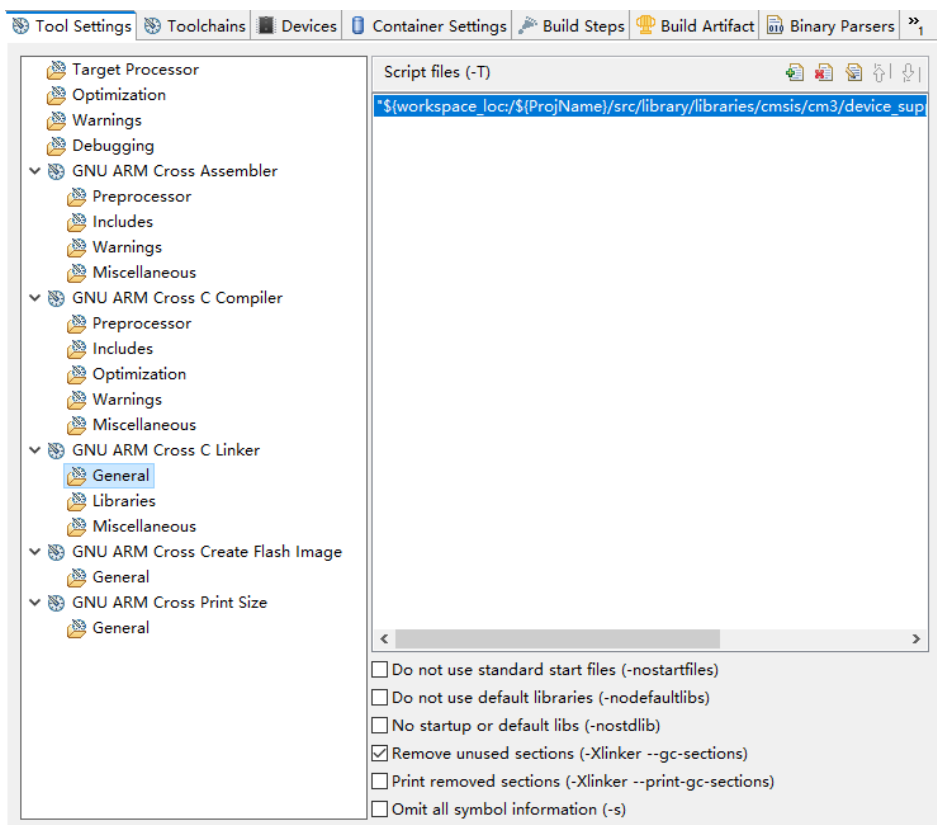
GMD_RefDesign¥cm3_demo の場合、C ヘッダーファイルのパスの構成は以下のようになります。

- "\${workspace_loc}/\${ProjName}/src/library/libraries/cmsis/cm3/device_support}"
- "\${workspace_loc}/\${ProjName}/src/library/libraries/cmsis/cm3/core_support/gmd}"
- "\${workspace_loc}/\${ProjName}/src/library/libraries/drivers/inc}"
- "\${workspace_loc}/\${ProjName}/src/library/middlewares/delay}"
- "\${workspace_loc}/\${ProjName}/src/library/middlewares/dmm}"
- "\${workspace_loc}/\${ProjName}/src/library/middlewares/gpio}"
- "\${workspace_loc}/\${ProjName}/src/library/middlewares/hyper_ram}"
- "\${workspace_loc}/\${ProjName}/src/library/middlewares/psram}"
- "\${workspace_loc}/\${ProjName}/src/library/middlewares/spi_flash}"
- "\${workspace_loc}/\${ProjName}/src/library/middlewares/uart}"
- "\${workspace_loc}/\${ProjName}/src/project}"

GNU ARM Cross C Linker の構成

「GNU ARM Cross C Linker > General > Script files (-T)」では GMD Flash リンカーとして「gw1ns4c_flash.ld」を構成します(図 2-7)。

図 2-7 GNU ARM Cross C Linker の構成



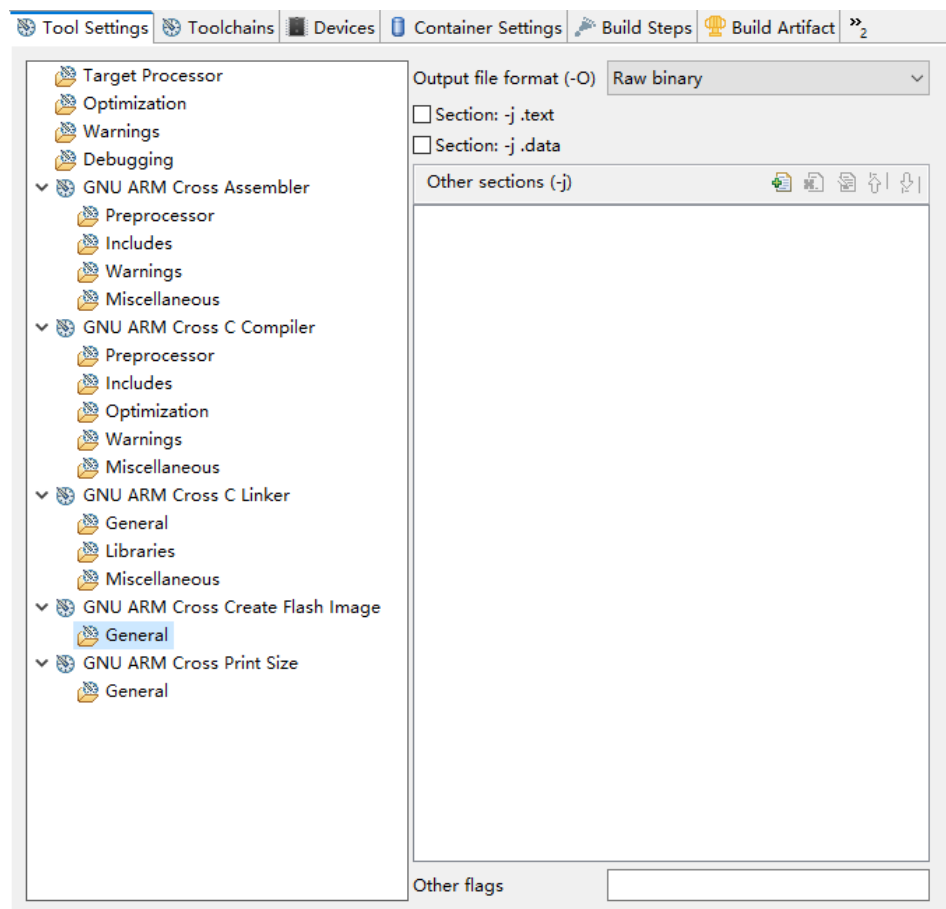
例えば、ソフトウェア・リファレンス・デザイン GMD_RefDesign¥cm3_demo の場合、Flash リンカーの構成は次のとおりです。

```
"${workspace_loc:${ProjName}/src/library/libraries/cmsis/cm3/device_support/startup/gmd/linker/gw1ns4c_flash.ld}"
```

GNU ARM Cross Create Flash Image の構成

「Cross ARM GNU Create Flash Image > General > Output file format (-O)」では「Raw binary」を選択します(図 2-8)。

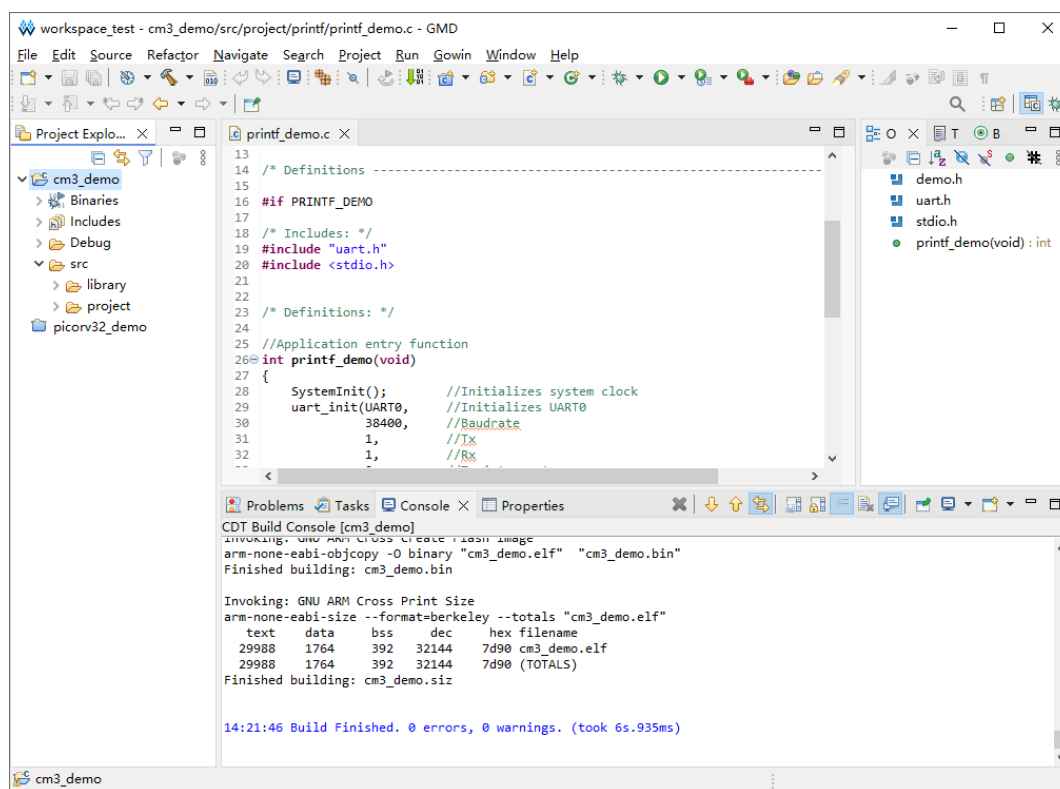
図 2-8 GNU ARM Cross Create Flash Image の構成



2.2.3 コンパイル

プロジェクトの構成とコーディングが完了した後、プロジェクトをコンパイルします。ツールバー上の **Build** 「」もしくは **Build All** 「」、またはメニューバー上の「**Project > Build Project**」もしくは「**Project > Build All**」をクリックしてコンパイルして実行ファイルを生成します(図 2-)。

図 2-9 コンパイル

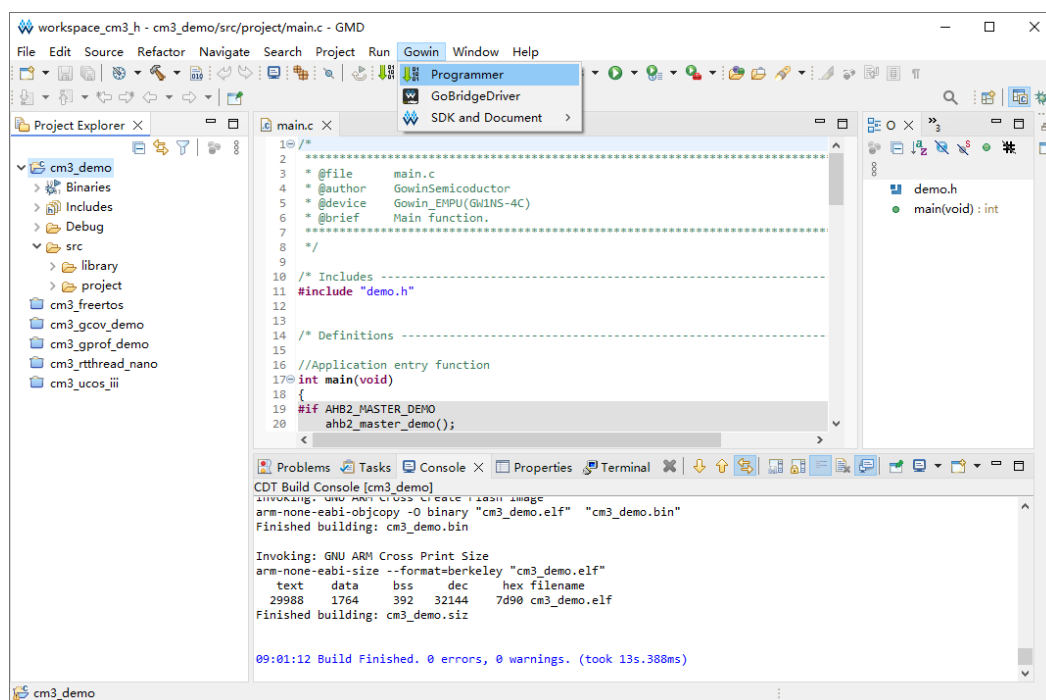


2.2.4 ダウンロード

Programmer を使用してソフトウェア設計の実行ファイルをダウンロードします。

GMD ソフトウェアのメニューバー上の「Gowin > Programmer」またはツールバー上の「Programmer」()をクリックして Programmer を開きます。

図 2-9 Programmer

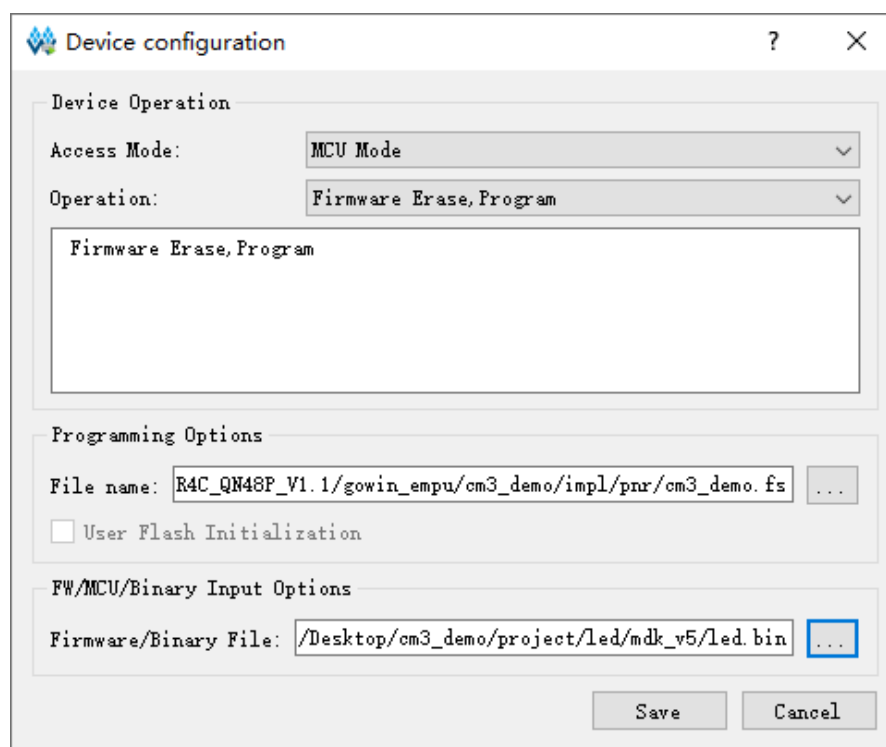


Programmer のメニューバー上の「Edit > Configure Device」またはツールバー上の「Configure Device 」をクリックして、「Device configuration」ダイアログボックスを開きます。

GW1NS-4C または GW1NSR-4C の場合、ダウンロード・オプションの構成は図 2-10 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「MCU Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Firmware Erase, Program」または「Firmware Erase, Program, Verify」を選択します。

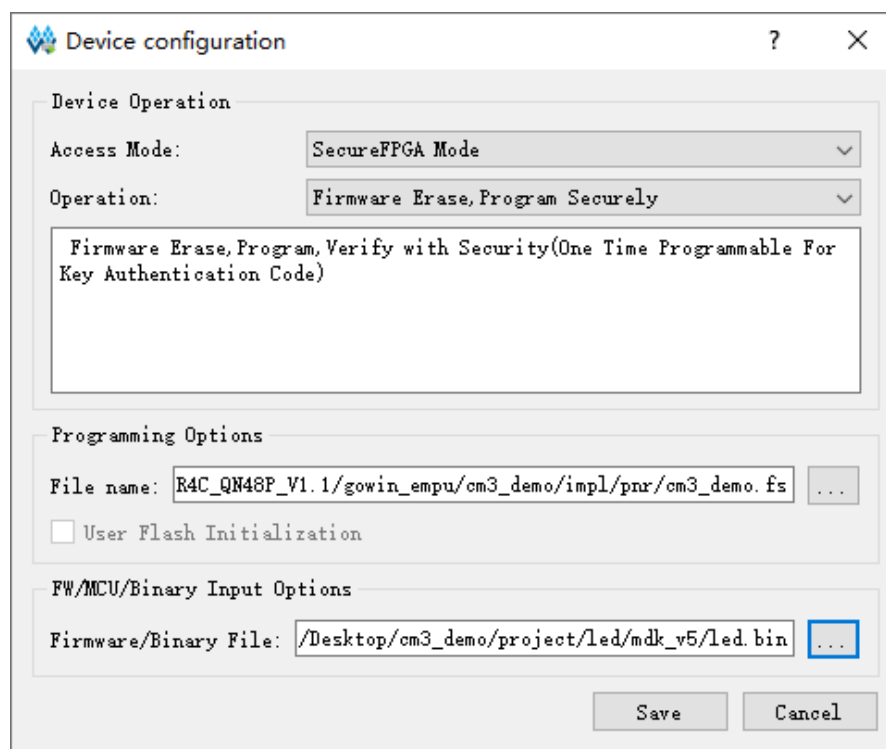
図 2-10 GW1NS-4C/GW1NSR-4C のダウンロード・オプションの構成



GW1NSR-4C の場合、ダウンロード・オプションの構成は図 2-11 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「SecureFPGA Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Firmware Erase, Program Securely」を選択します。

図 2-11 GW1NSER-4C のダウンロード・オプションの構成



- 「FW/MCU/Binary Input Options > Firmware/Binary File」でソフトウェア設計の実行ファイルを選択します。
- 最後に「Save」をクリックします。

注記：

「Programming Options > File name」でビットストリームファイルをインポートします。『Gowin_EMPU(GW1NS-4C)ハードウェア設計ユーザーガイド([IPUG932](#))』を参照してください。

Device configuration が完了したら、Programmer のツールバー上の「Program/Configure」()をクリックしてソフトウェア設計の実行ファイルとハードウェア設計のビットストリームファイルをダウンロードします。

2.2.5 オンラインデバッグ

ソフトウェア設計の実行ファイルとハードウェア設計のビットストリームファイルのダウンロードが完了した後、当該ソフトウェア設計に問題がある場合は、GWU2X、GWUSB、J-LINK などのエミュレーターに接続してオンラインで当該 MCU ソフトウェア設計をデバッグすることができます(オンラインデバッグ用の MCU ソフトウェア設計は、チップにダウンロードされた MCU ソフトウェア設計と一致している必要があります)。

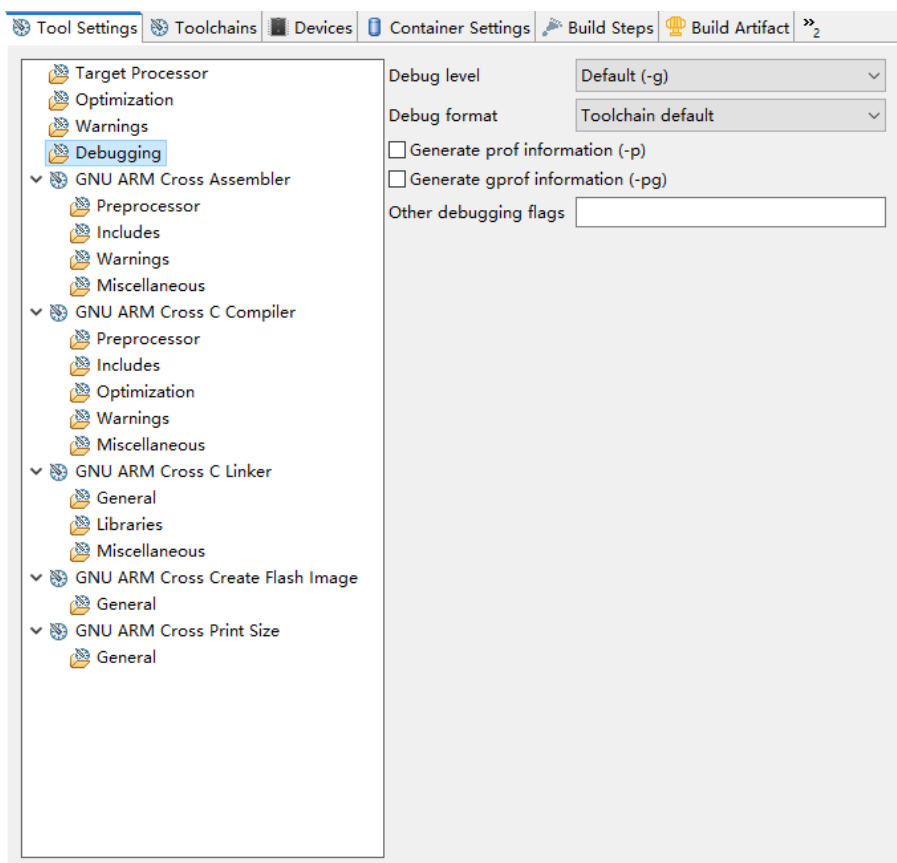
デバッグのプロセスは次のとおりです：

- デバッグレベルの構成
- デバッグモードの構成
- MCU JTAG モードの切り替え
- MCU JTAG インターフェースの切り替え
- エミュレーターへの接続
- デバッグの開始

デバッグレベルの構成

Project Explorer ビューで、現在のプロジェクトの「Properties > C/C++ Build > Settings > Tool Settings > Debugging > Debug level」でデバッグレベルを選択します。「Default(-g)」または「Maximum(-g3)」に設定することをお勧めします。

図 2-12 デバッグレベルの構成



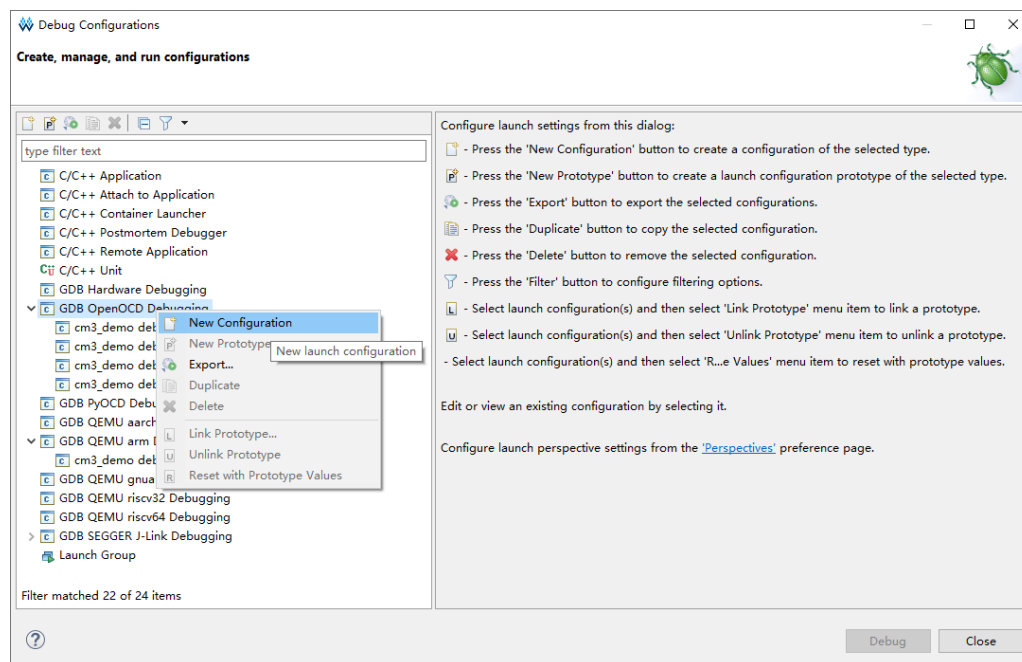
デバッグモードの構成

GMD ソフトウェアは GDB OpenOCD Debugging、GDB QEMU arm Debugging、GDB SEGGER J-Link Debugging などのデバッグモードをサポートしています。

例えば、GDB OpenOCD Debugging デバッグモードと GWU2X エミュレーターを選択します。

メニューバーの「Run > Debug Configurations… > GDB OpenOCD Debugging」を右クリックして「New」を選択し、このプロジェクトのデバッグモードを作成します。

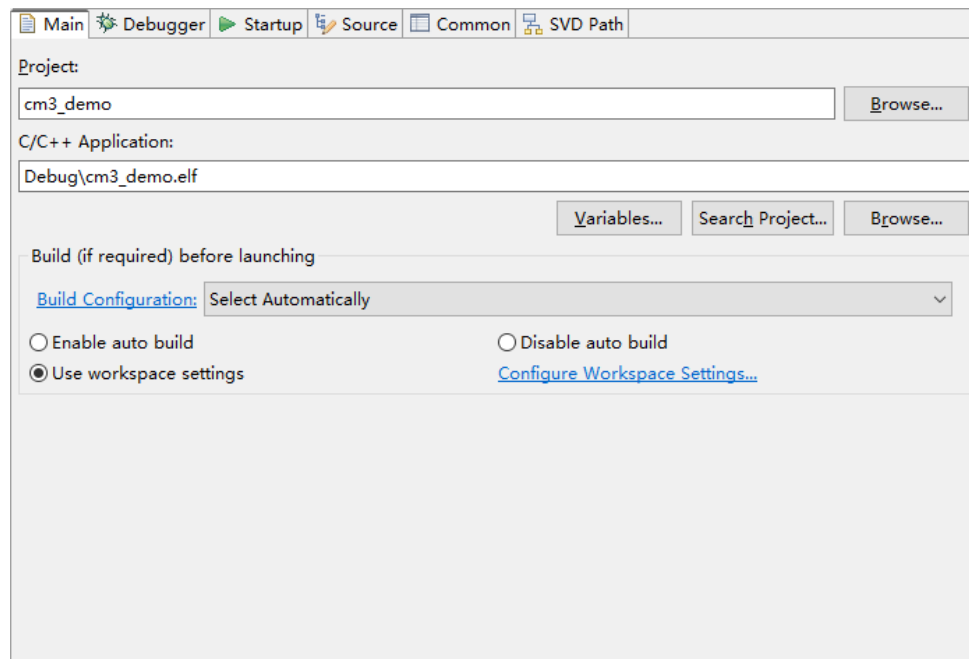
図 2-13 デバッグ・ビューの作成



デバッグ・ビューには、Main、Debugger、Startup などのビューがあります。その構成方法は以下に示すとおりです。

1. 「Main」ビューを図 2-に示します。

図 2-14 Main ビューの構成



2. 「Debugger」ビューを下図に示します。

例：

- Start OpenOCD locally
- GDB port: 3333
- Telnet port: 4444
- Tcl port: 6666
- Config options: -f board/gowin_u2x_m3.cfg
- Start GDB session
- Command : set mem inaccessible-by-default off

Gowin_EMPU(GW1NS-4C)は JTAG デバッグポートのみをサポートします。

図 2-15 Debugger ビューの構成

The screenshot shows the 'Debugger' configuration window with the following settings:

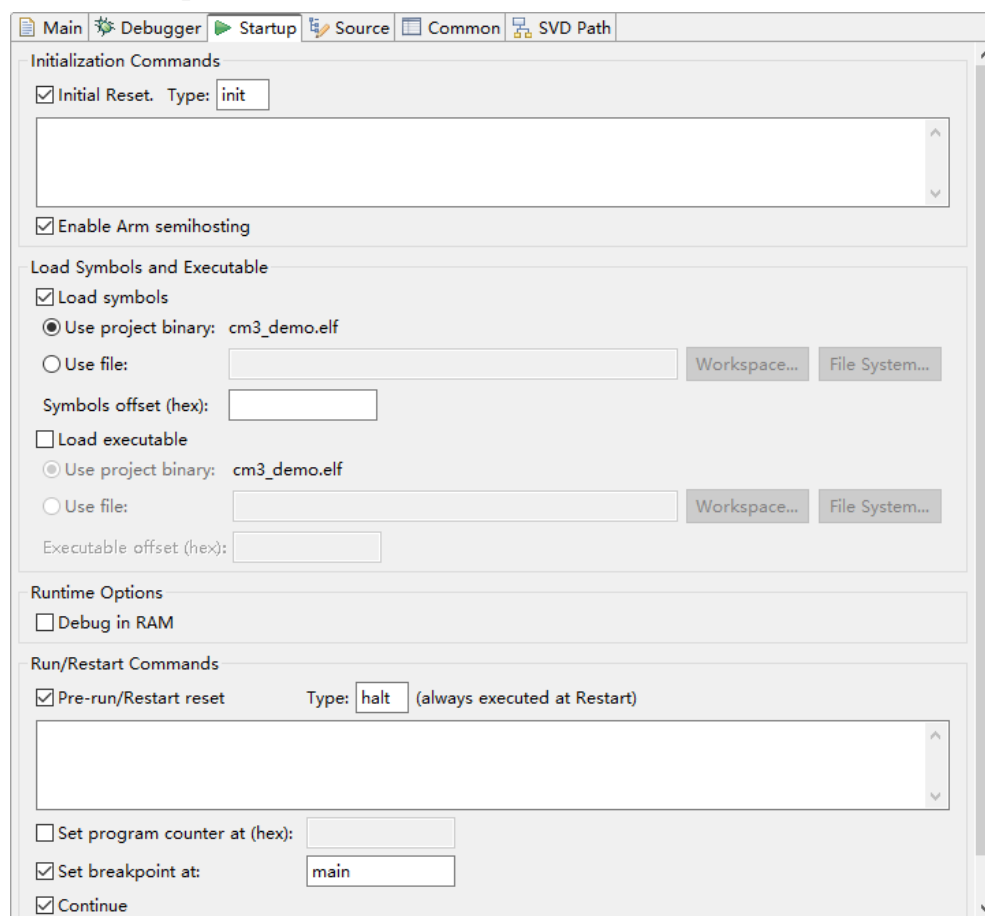
- OpenOCD Setup**
 - ☒ Start OpenOCD locally
 - Executable path: `${openocd_path}/${openocd_executable}` (with Browse... and Variables... buttons)
 - Actual executable: `E:\GMD\toolchain\gowin\openocd\bin\openocd.exe`
 - (to change it use the [global](#) or [workspace](#) preferences pages or the [project](#) properties page)
 - GDB port: `3333`
 - Telnet port: `4444`
 - Tcl port: `6666`
 - Config options: `-f board/gowin_u2x_m3.cfg`
 - ☒ Allocate console for OpenOCD
 - ☐ Allocate console for the telnet connection
- GDB Client Setup**
 - ☒ Start GDB session
 - Executable name: `${cross_prefix}gdb${cross_suffix}` (with Browse... and Variables... buttons)
 - Actual executable: `arm-none-eabi-gdb`
 - Other options: (empty field)
 - Commands: `set mem inaccessible-by-default off`
- Remote Target**
 - Host name or IP address: `localhost`
 - Port number: `3333`
 - ☐ Force thread list update on suspend

3. 「Startup」ビューを下図に示します。

例：

- Initial Reset. Type: init
- Enable Arm semihosting
- Load symbols
- Pre-run/Restart reset
- Type: halt
- Set breakpoint at: main
- Continue

図 2-16 Startup ビューの構成



MCU JTAG モードの切り替え

Programmer において Gowin_EMPU(GW1NS-4C)の MCU JTAG モードをダウンロードモードからデバッグモードに切り替えます。

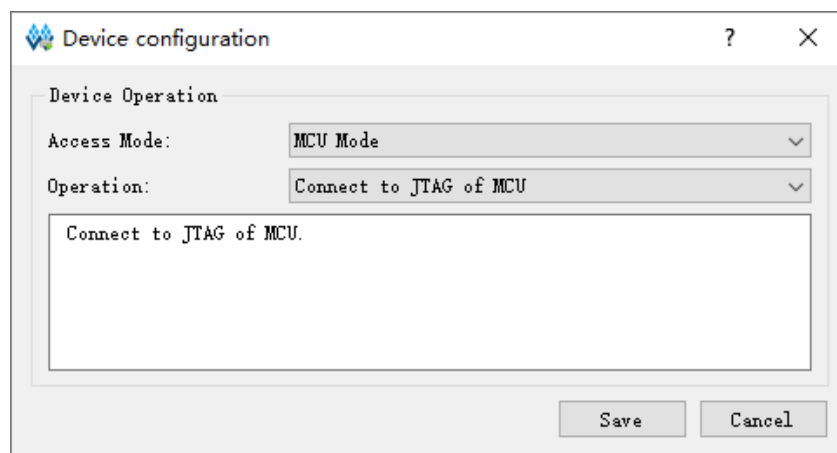
GMD ソフトウェアのメニューバー上の「Run > Programmer」またはツールバー上の「Programmer」(🔧)をクリックして Programmer を開きます(図 2-10)。

Programmer のメニューバーの Edit > Configure Device またはツールバーの🔧をクリックして、Device configuration ダイアログボックスを開きます。

GW1NS-4C または GW1NSR-4C の場合の構成を図 2-17 に示します。

- 「Access Mode」のドロップダウン・リストから「MCU Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Connect to JTAG of MCU」を選択します。

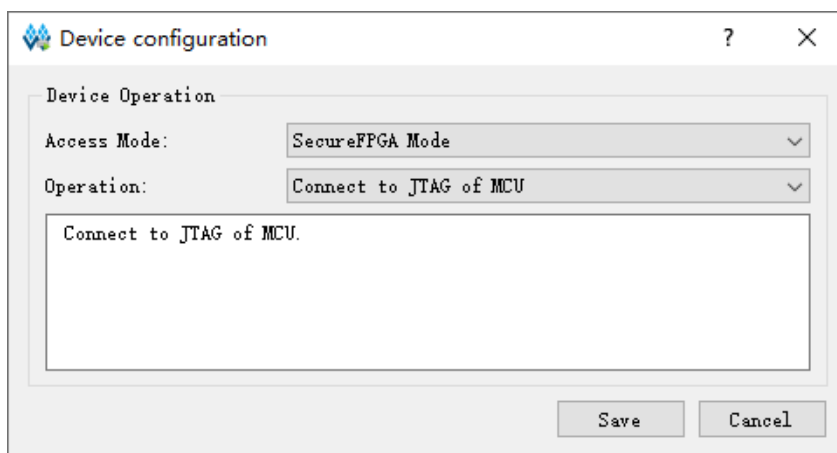
図 2-17 GW1NS-4C/GW1NSR-4C の場合の構成



GW1NSR-4C の場合の構成を図 2-18 に示します。

- 「Access Mode」のドロップダウン・リストから「SecureFPGA Mode」を選択します。
- 「Operation」のドロップダウン・リストから「Connect to JTAG of MCU」を選択します。

図 2-18 GW1NSR-4C の場合の構成



- 「Save」をクリックして MCU JTAG モードの切り替えを完了します。

Device configuration が完了したら、Programmer のツールバーの Program/Configure 「」をクリックして JTAG モードの切り替えを完了します。

MCU JTAG インターフェースの切り替え

例えば、DK-START-GW1NSR4C-QN48P V1.1 開発ボードの場合、開発ボード上の SW3、SW4、SW5、SW6 DIP スイッチ(JTAG インターフェ

ースの TCK、TDO、TDI、TMS)を手動でダウンロード側からデバッグ側に切り替えます。

注記：

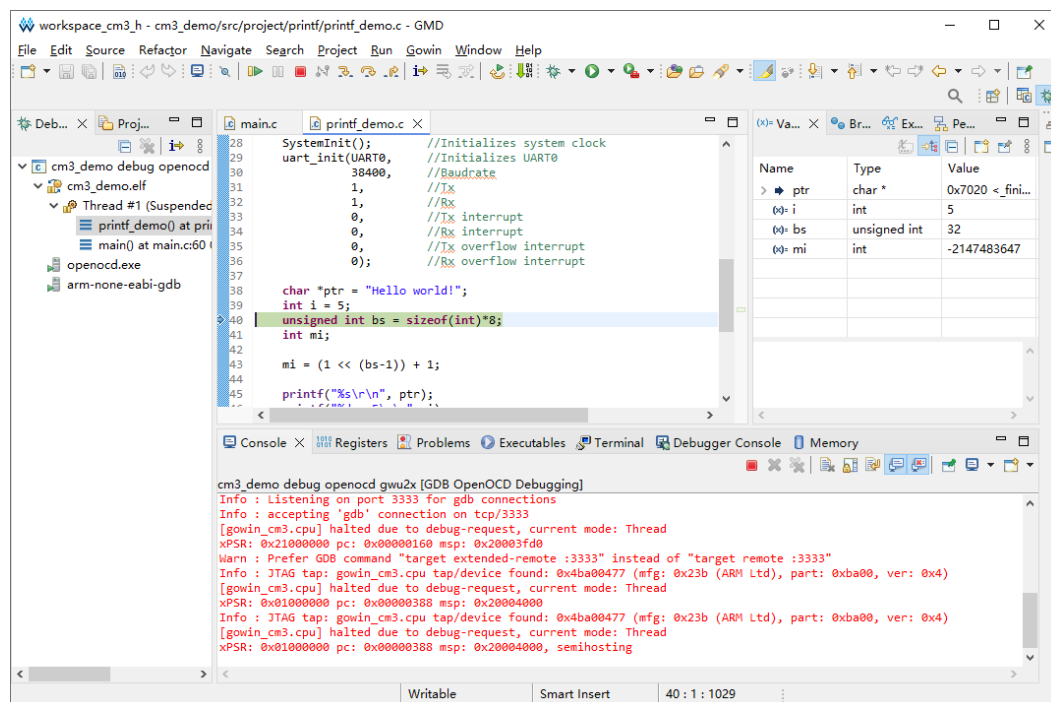
- MCU JTAG モードの切り替えおよび MCU JTAG インターフェースの切り替え中は、開発ボードの電源をオフにはなりません。
- 開発ボードが途中でパワーオフした場合、電源を再投入すると自動的に MCU JTAG ダウンロードモードに戻ります

デバッグの開始

GWU2X シミュレーターに接続します。

ツールバー上の **Debug** ボタン「」をクリックしてデバッグモードに入ります。ブレークポイントの設定、シングルステップ・デバッグ、リセット、実行などの操作を実行できます(図 2-19)。

図 2-19 デバッグの開始



2.3 リファレンス・デザイン

GMD(テスト済みバージョン: V2025.01)を使用したリファレンス・デザイン：[cdn.gowinsemi.com.cn/Gowin EMPU\(GW1NS-4C\) V2.1.zip](https://cdn.gowinsemi.com.cn/Gowin_EMPU(GW1NS-4C)_V2.1.zip)

…¥ref_design¥MCU_RefDesign¥GMD_RefDesign¥cm3_demo、
cm3_freertos、cm3_rtthread_nano、cm3_ucos_iii、cm3_gcov_demo、
cm3_gprof_demo

