Gowin_EMPU_M1 IDE Software

Reference Manual

IPUG536-2.3.1E, 07/18/2025

Disclaimer

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

Revision History

Date	Version	Description
02/18/2019	1.0E	Initial version published.
07/18/2019	1.1E	● MCU hardware design and software programming design support extended peripherals: CAN, Ethernet, SPI-Flash, RTC, DualTimer, TRNG, I²C, SPI, SD-Card. ● MCU supports off-chip SPI-Flash downloading startup.
08/18/2019	1.2E	● MCU hardware design and software programming design support extended peripheral: DDR3 Memory. ● Fixed known issues of ITCM, DTCM Size and IDE.
09/27/2019	1.3E	Updated and optimized MCU programming software and the interface and functions of Gowin MCU Designer.
01/16/2020	1.4E	● MCU hardware design and software programming design supports PSRAM. ● MCU compiling software GMD V1.0 updated. ● RTOS reference design updated. ● Hardware and software reference design of AHB2 and APB2 extension bus interface added.
03/10/2020	1.5E	GW2A-18C/GW2AR-18C/GW2A-55C devices added.
06/12/2020	1.6E	● MCU supports external instruction memory. ● MCU supports external data memory. ● Extension of 6 AHB bus interfaces. ● Extension of 16 APB bus interfaces. ● GPIO supports multiple interface types. ● I²C supports multiple interface types.
07/16/2021	1.7E	MCU version updated.
10/12/2021	1.8E	ITCM and DTCM Size of GW2AN-9X/GW2AN-18X modified.
05/11/2023	1.9E	Arora V FPGA products supported.
07/21/2023	2.0E	Tested software version updated.
03/07/2024	2.1E	● GW5AT-60 Version A products supported. ● Reference example for IDE software updated.
07/12/2024	2.2E	● GW5ART-15 Version A products supported. ● Reference example for IDE software updated.
01/17/2025	2.3E	The note on online software debugging added.
07/18/2025	2.3.1E	The link of GMD software installation package in “2.1 Software Installation” updated.

Contents

List of Figures

1 ARM Keil

1.1 Software Installation

For the detailed, please refer to Getting Started with MDK provided by ARM Keil MDK website.

Note!

ARM Keil MDK (V5.26 and above) is recommended.

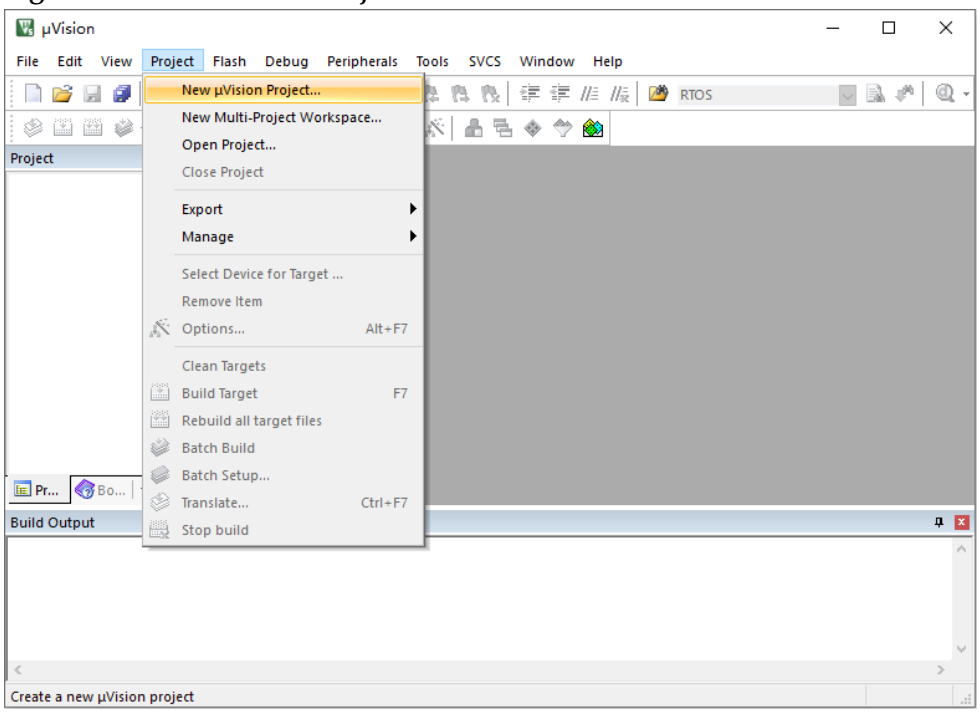
1.2 Project Template

ARM Keil MDK can be used for Gowin_EMPU_M1 software programming. The steps include project creation, configuration, coding, compilation, downloading and debugging.

1.2.1 Create a New Project

Double click to open ARM Keil MDK and select "Project > New uVision Project…" to create a new project, as shown in Figure 1-1.

Figure 1-1 Create a New Project

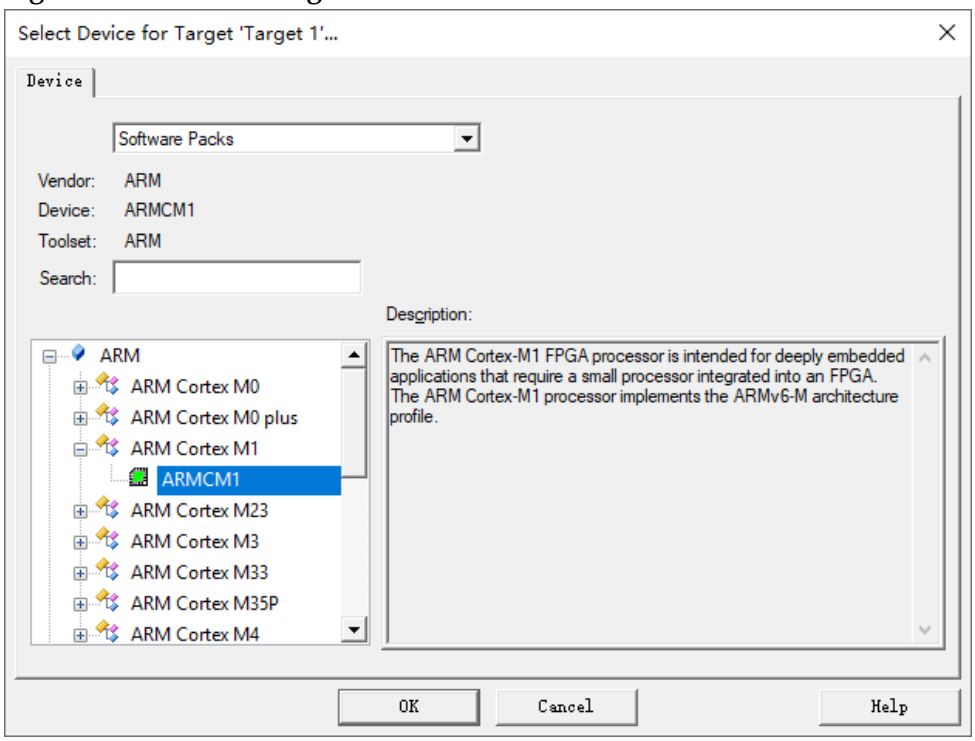


1.2.2 Configuration Option

Device Configuration

ARM Cortex-M1 is embedded in Gowin_EMPU_M1, so the device type is configured as " ARM Cortex M1 > ARMCM1", as shown in Figure 1-2.

Figure 1-2 Device Configuration



ROM and RAM Configuration

Gowin_EMPU_M1 internal instruction memory or external instruction memory is the ROM.

Gowin_EMPU_M1 internal data memory or external data memory is the RAM.

1. Configure the initial address and the size of ROM (Internal Instruction Memory) and RAM (Internal Data Memory).

   ROM initial address and size configuration:

   - Off-chip SPI-Flash memory download startup
     - ROM initial address: 0x400
     - ROM Size: Set according to the actural configuration of the hardware design ITCM Size. For example, if ITCM Size 32KB, ROM size is set to 0x7C00.
   - On-chip ITCM initialization value download startup
     - ROM initial address: 0x00000000.
     - ROM Size: Please set according to the actural configuration of the hardware design ITCM Size. For example, if ITCM Size

32KB, ROM size is set to 0x8000.

RAM initial address and Size configuration:

- RAM initial address: 0x20000000
- RAM Size: Please set according to the actual configuration of the hardware design DTCM Size. For example, if DTCM Size 32KB, RAM size is set to 0x8000.

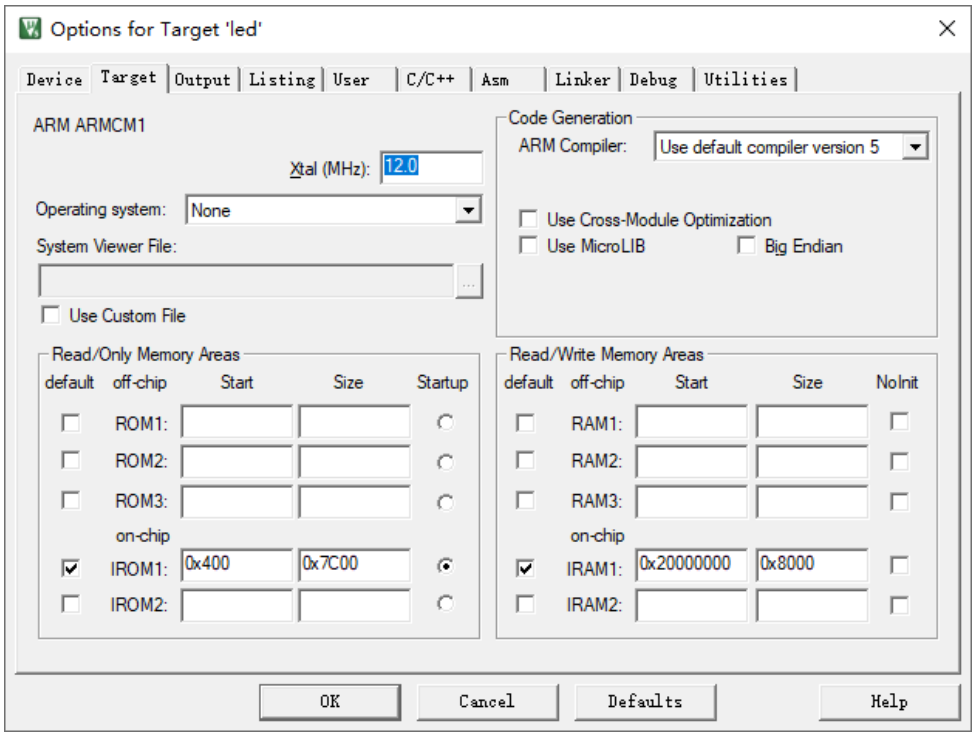
Limited by the on-chip memory resource, the size configuration of ITCM and DTCM cannot exceed the max. on-chip memory size.

- For GW1N/R-9, ITCM or DTCM can be configured up to 32KB. If ITCM or DTCM has been configured to 32KB, the other can only be configured up to 16KB.
- For GW2AN-9X/18X, ITCM or DTCM can be configured up to 32KB. If ITCM or DTCM has been configured to 32KB, the other can only be configured up to 16KB.
- For GW2A/R/NR-18, ITCM or DTCM can be configured up to 64KB. If ITCM or DTCM has been configured to 64KB, the other can only be configured up to 16KB.
- For GW2A/N-55, ITCM or DTCM can be configured up to 256KB. If ITCM or DTCM has been configured to 256KB, the other can only be configured up to 16KB.
- For GW5A/T/ST/S-138, ITCM or DTCM can be configured up to 512KB. If ITCM or DTCM has been configured to 512KB, the other can only be configured up to 218KB.
- For GW5AT-75, ITCM or DTCM can be configured up to 256KB. If ITCM or DTCM has been configured to 256KB, the other can only be configured up to 256KB.
- For GW5A/R/S-25, ITCM or DTCM can be configured up to 64KB. If ITCM or DTCM has been configured to 64KB, the other can only be configured up to 32KB.
- For GW5A/T-60, ITCM or DTCM can be configured up to 128KB. If ITCM or DTCM has been configured to 128KB, the other can only be configured up to 64KB.
- For GW5AT/RT-15, ITCM or DTCM can be configured up to 64KB. If ITCM or DTCM has been configured to 64KB, the other can only be configured up to 8KB.

The configuration of ROM (Internal Instruction Memory) and RAM (Internal Data Memory) is as shown in Figure 1-3.

Using DK_START_GW5A-LV25UG324 V2.0 development board reference design an instance, the initial address of ROM is 0x400 and the "Size" is 0x7C00. The initial address of RAM is 0x20000000 and the Size is 0x8000.

Figure 1-3 ROM and RAM Configuration



2. Configure the initial address and the size of ROM (External Instruction Memory) and RAM (External Data Memory).

   ROM initial address and Size configuration:

   - ROM initial address: 0x00000000.
   - ROM Size: Please set according to the actual Size of the hardware design.

   RAM initial address and Size configuration:

   - RAM initial address: 0x20100000.
   - RAM Size: Please set according to the actual Size of the hardware design.

Output File Format Configuration

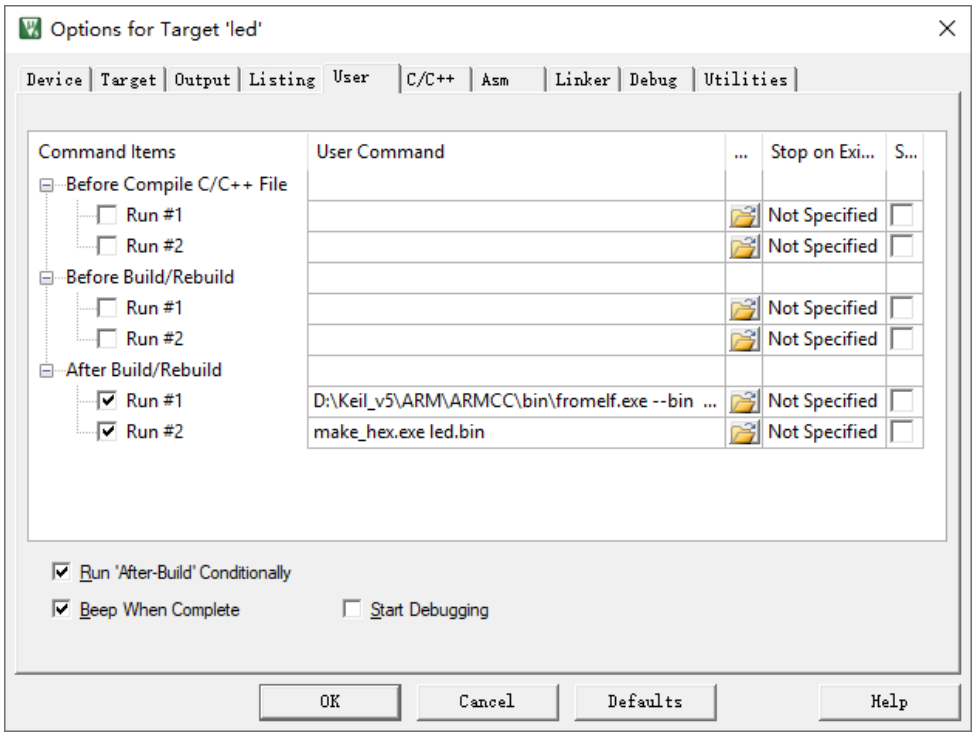
Gowin_EMPU_M1 outputs BIN file, so axf file format should be converted to BIN file format.

If executable program file is used as the initial value of ITCM, the BIN file should be converted to four hex files, itcm0, itcm1, itcm2, and itcm3 using Gowin script make_hex.exe.

The usage of calling the file format tool with user command line is as shown in Figure 1-4.

- Run #1: fromelf.exe --bin -o bin-file axf-file
- Run #2: make_hex.exe bin-file

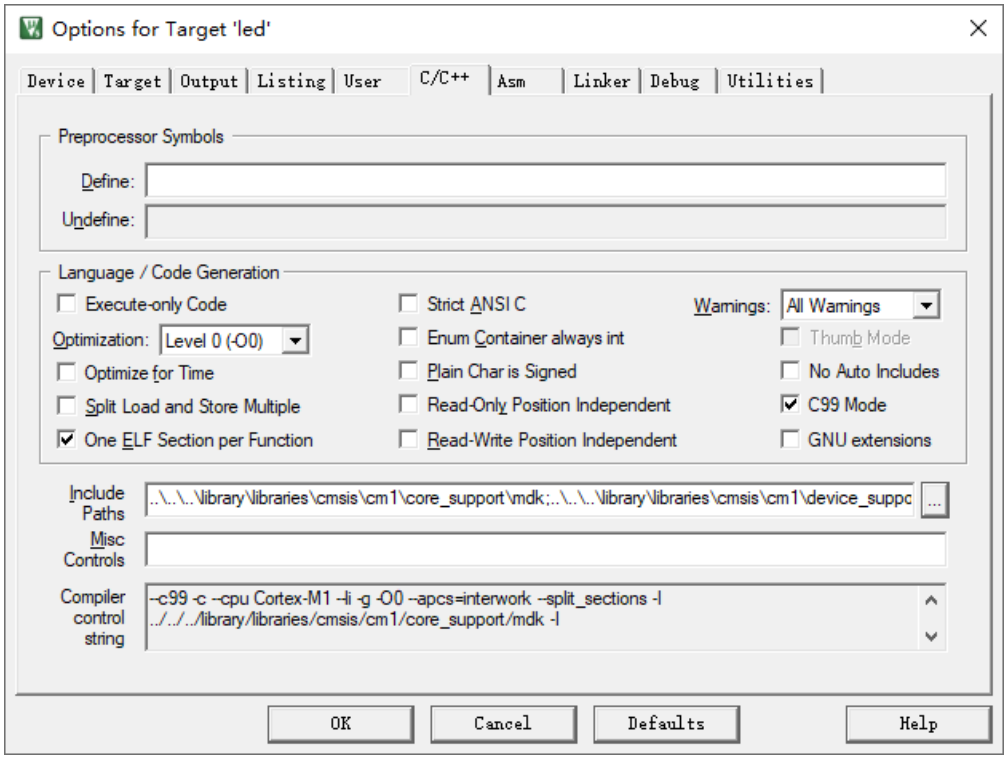
Figure 1-4 Output File Format Configuration



Header File Path Configuration

The C code header file configuration is used to call the C code header file during building. The configuration is as shown in Figure 1-5.

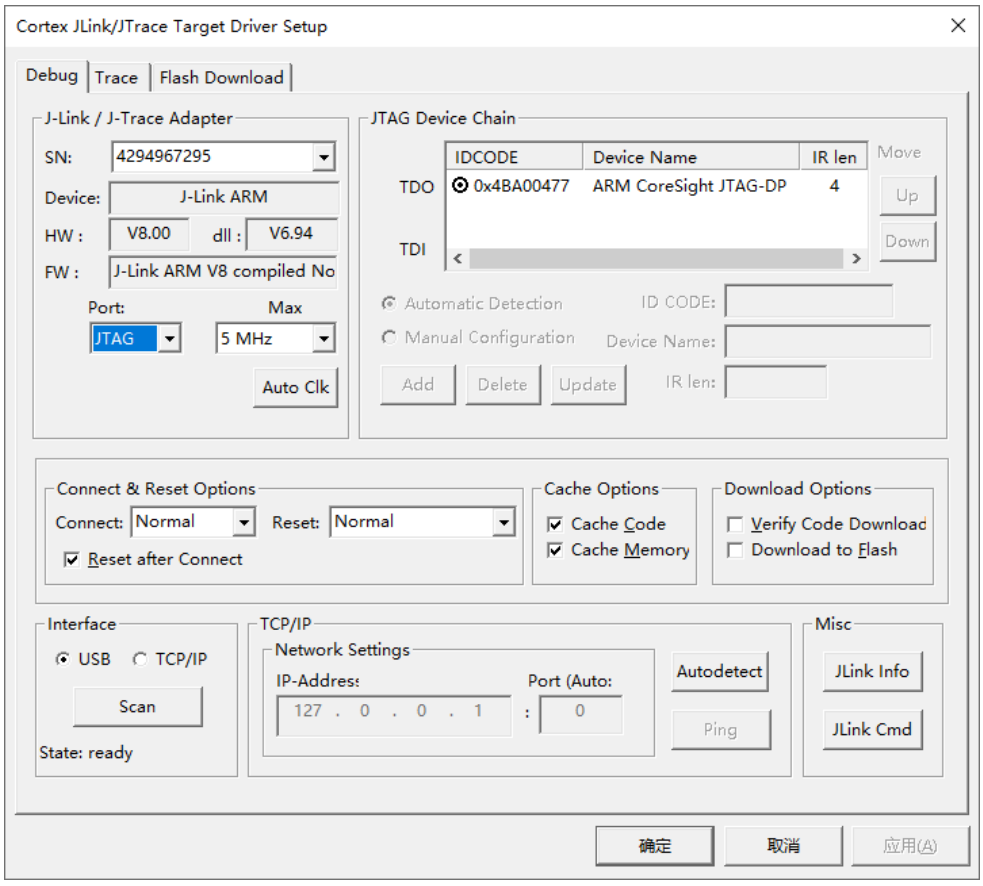
Figure 1-5 Header File Path Configuration



Debug Configuration

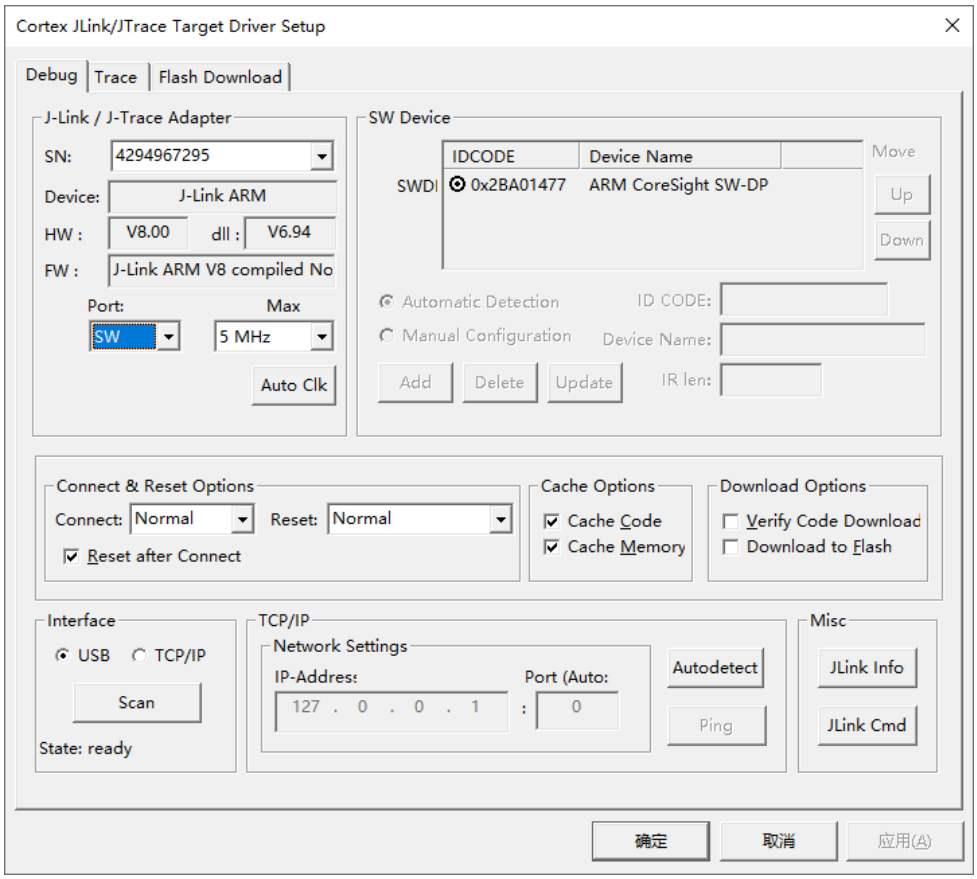
- Configure the Emulator
  - U-LINK Emulator
    If the U-LINK emulator is selected, use "ULNK2/ME Cortex Debugger".
  - J-LINK Emulator
    If the J-LINK emulator is selected, use "J-LINK/J-TRACE Cortex".
- Configure the Debug Interface
  - JTAG Debug Interface
    If it is configured as the JTAG debug interface, the configuration method is as shown in Figure 1-6.

Figure 1-6 JTAG Debug Interface Configuration



  - SW Interface
    If it is configured as the SW debug interface, the configuration is as shown in Figure 1-7.

Figure 1-7 SW Debug Interface Configuration



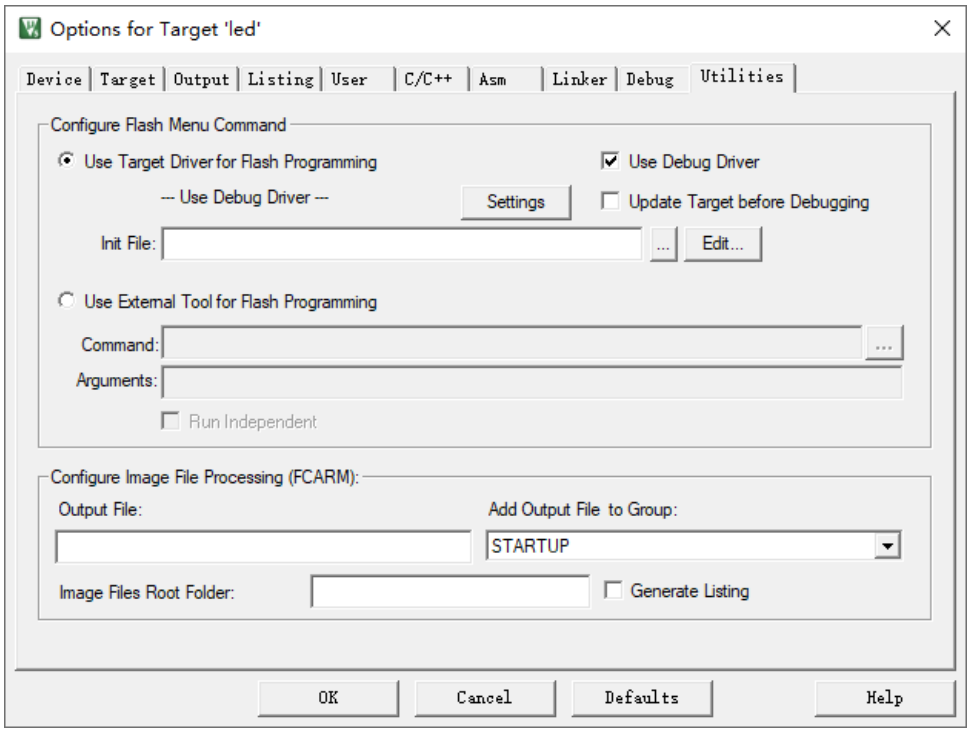
In the Debug Interface Type Configuration option:

- Please do not select the "Download Options > Verify Code Download" option.
- Please do not select the "Download Options > Download to Flash" option.

Flash Configuration

If online debugging is required, "Update Target before Debugging" cannot be selected, as shown in Figure 1-8.

Figure 1-8 Flash Configuration

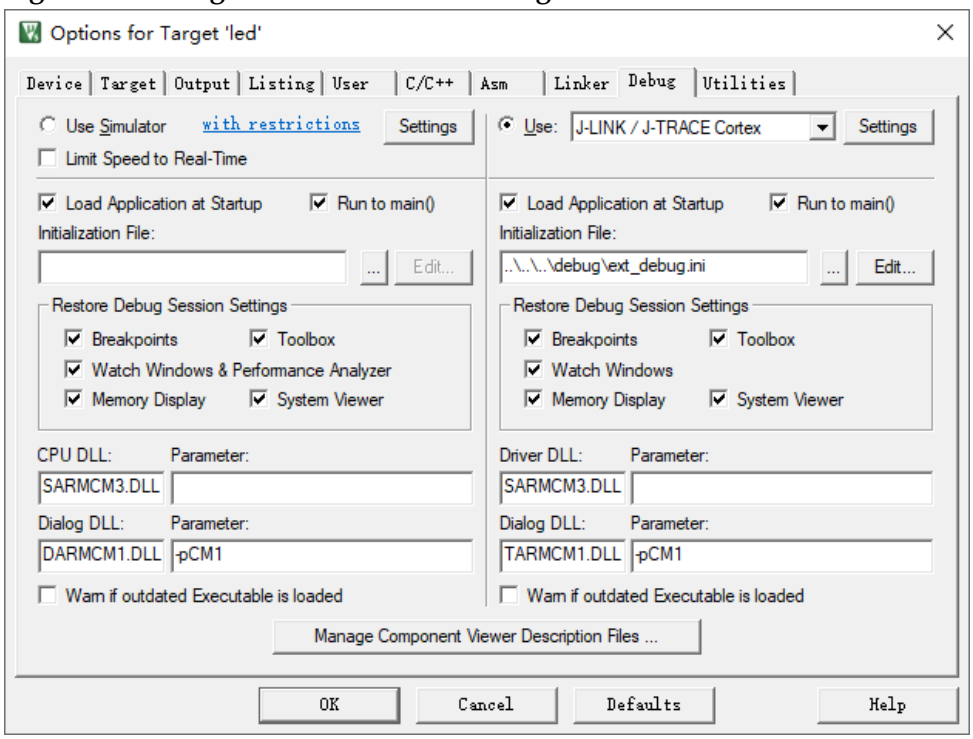


Debug Initialization File Configuration

If selecting off-chip SPI-Flash memory download startup, it needs to load debug initialization file when debugging online. Select ext_debug.ini in "Initialization File" option as shown in Figure 1-9.

"ext_debug.ini" file is located in the "library\debug".

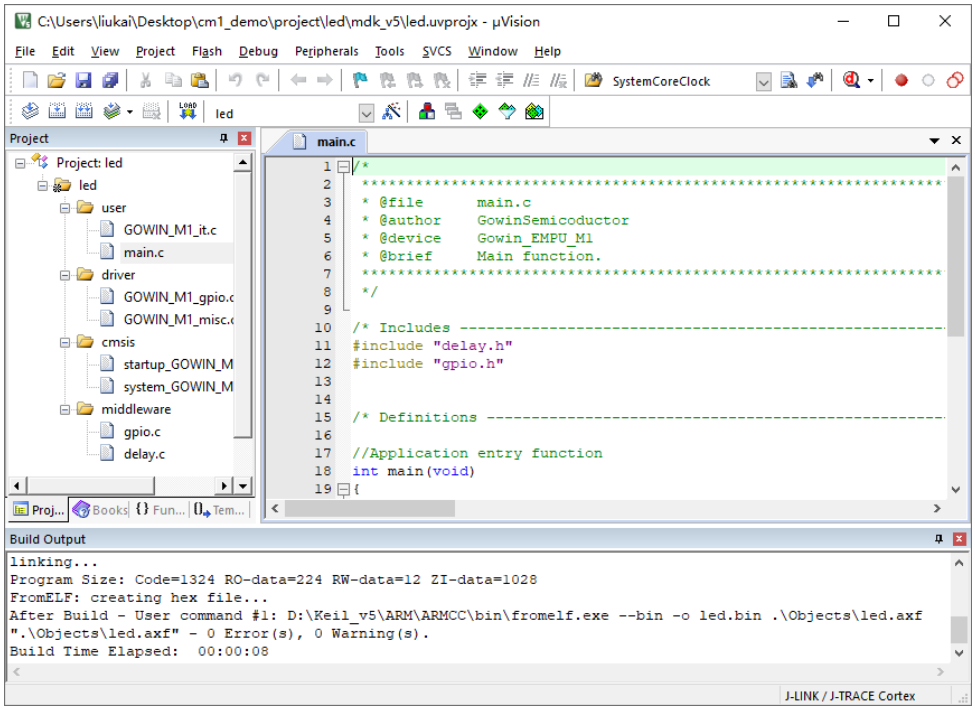
Figure 1-9 Debug Initialization File Configuration



1.2.3 Build

After encoding and configuration, click Build " " or Rebuild " " or click "Project > Build Target" or "Project > Rebuild all target files" on the menu bar to build the project to generate software design BIN file and four hex files of itcm0, itcm1, itcm2, and itcm3, as shown in Figure 1-10.

Figure 1-10 Project Compiling



1.2.4 Download

After compiling Gowin_EMPU_M1software programming design, for the downloading, please refer to *IPUG532, Gowin_EMPU_M1Download Reference Manual.*

1.2.5 Software Online Debug

After completing the download of the hardware design bitstream files generated by the hardware design and the software design BIN files generated by the software programming design, if there are any issues, you can use the U-LINK and J-LINK to debug online.

You can download and debug the software, no recompilation required.

Note!

Gowin_EMPU_M1 does not support automatic downloading during debugging with ARM Keil software. Prior to each debugging session, please use the Programmer tool to download the Binary file of the software design you intend to debug. Then, start debugging to ensure that the software project being debugged is the current one.

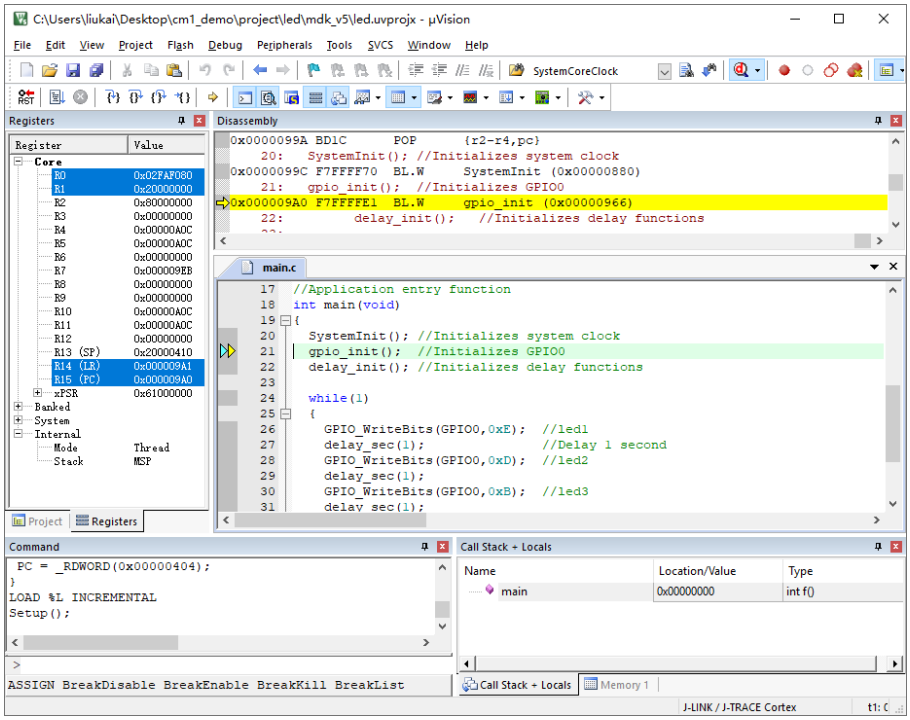
1. Connect Emulator

Connect J-LINK or U-LINK according to the Debug Access Port (JTAG: JTAG_3~JTAG_18, VCC and GND; or SWD: JTAG_7, JTAG_9, VCC and GND) location constrained to FPGA IO in the hardware design.

2. Start Debug

Connect the U-LINK or J-LINK Emulator. Click the Debug button " " on the tool bar, or click "Debug > Start/Stop Debug Session" on the menu bar to start debug. You can perform operations of breakpoint setting, single-step debug, reset and run, as shown in Figure 1-11.

Figure 1-11 Start Debug



1.3 Reference Design

Gowin provides reference design in ARM Keil MDK (tested software version V5.26) software environment. Click this link to get following reference design:

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo、cm1_fatfs、cm1_freertos、cm1_rtthread_nano、cm1_tcpip、cm1_ucos_iii

2 GMD Software

2.1 Software Installation

GMD software installation package is available at Gowinsemi website.

For the software installation and configuration of GMD, please refer to *SUG549, GOWIN MCU Designer User Guide*.

Note!

GOWIN MCU Designer (V1.2 and above) is recommended.

2.2 Project Template

Using GMD for Gowin_EMPU_M1 software programming design, it involves projects creation, option configuration, code writing, building, downloading, and online debug.

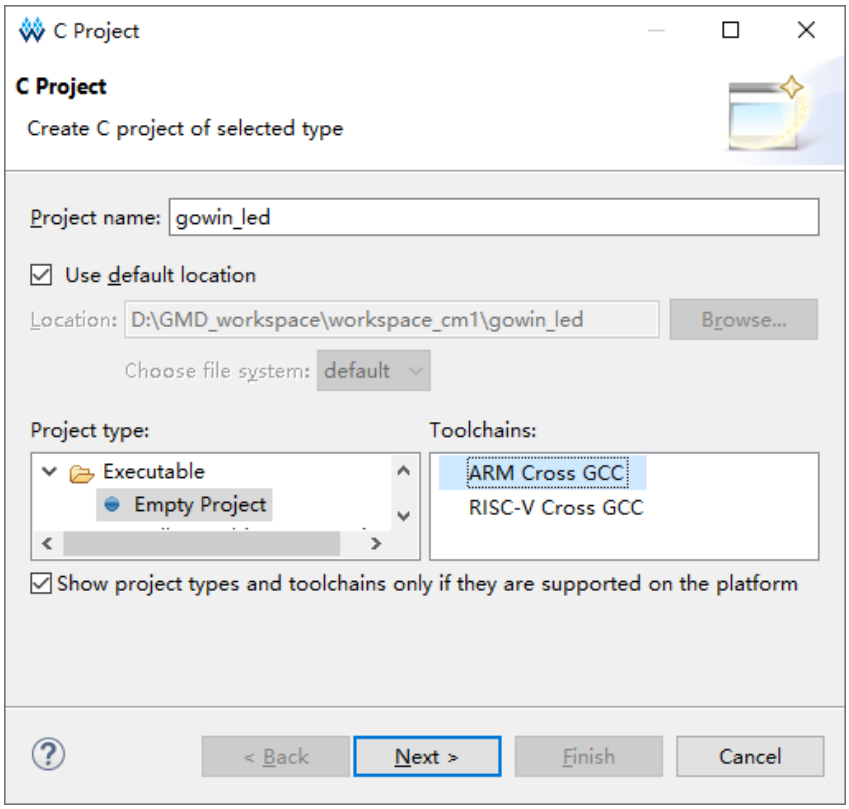
2.2.1 Create a Project

Create a New Project

Click "New" ( ) on the tool bar or select "File > New > C Project" on the menu bar, as shown in Figure 2-1.

1. Create a project name and location.
2. Select "Empty Project" type.
3. Select "ARM Cross GCC" building tool chain.

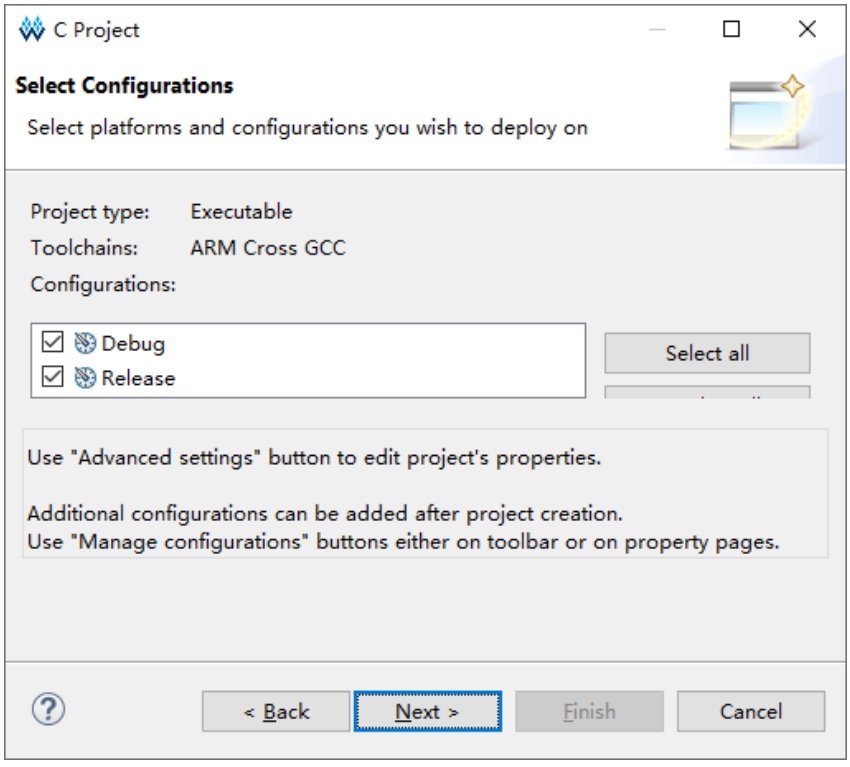
Figure 2-1 Creat a New Project



Select Platforms and Configurations

Select "Debug" and "Release" in configuration interface, as shown in Figure 2-2.

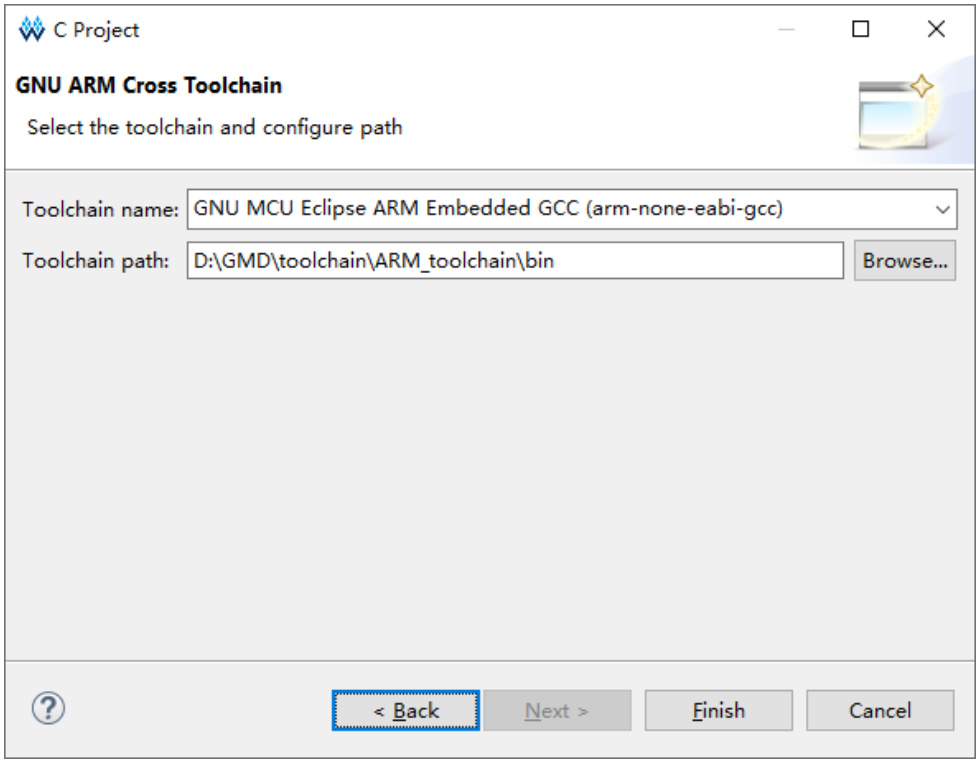
Figure 2-2 Select Platforms and Configurations



Select Configuration Toolchain and Path

Select "arm-none-eabi-gcc" as the cross compiling toolchain and import its path. It is recommended that Toolchain name and Toolchain path be configured by default, as shown in Figure 2-3.

Figure 2-3 Select Configuration Toolchain and Path



Create a Project

After project creation, select the created project in Project Explorer view, add engineering structure and import the software programming design.

Using GMD_RefDesign for an instance, the software programming design projects and codes are listed as follows.

Select the current project in Project Explorer view, and right-click "Refresh" option to automatically update the structure and code of the current project.

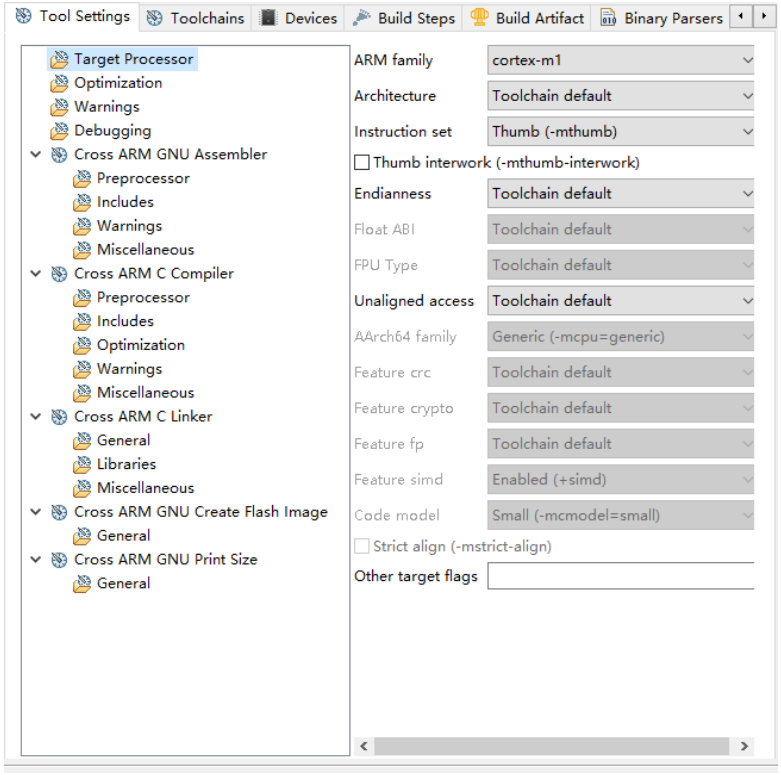
2.2.2 Configuration Option

In Project Explorer view, select the current project, right-click "Properties > C/C++ Build > Settings" to configure the parameters of current project.

Target Processor Configuration

Select " Target Processor > ARM family " and configure the option to "cortex-m1", as shown in Figure 2-4.

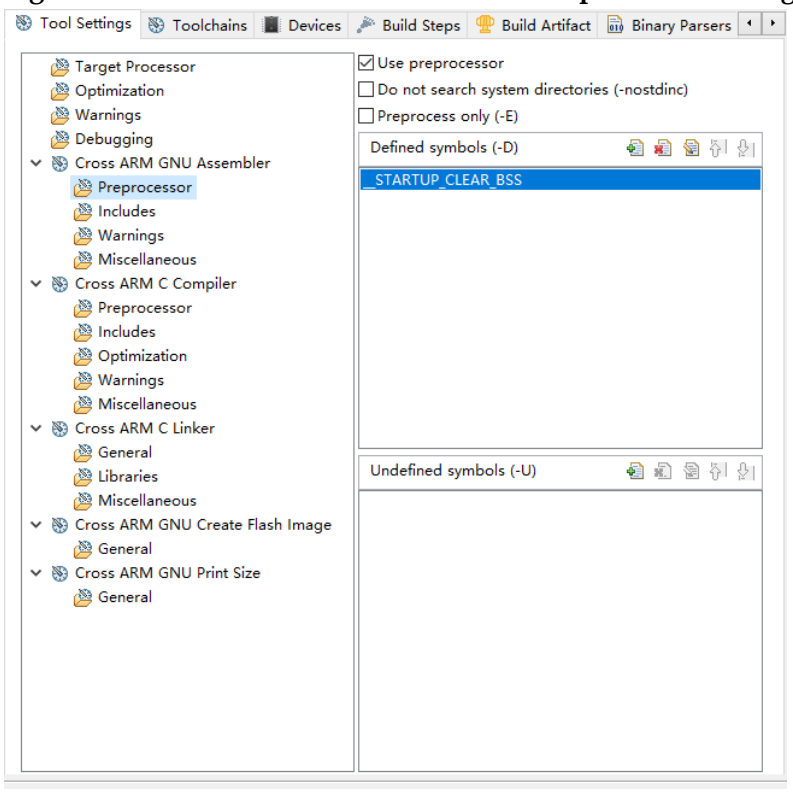
Figure 2-4 Target Processor Configuration



Cross ARM GNU Assembler > Preprocessor Configuration

Select " Cross ARM GNU Assembler > Preprocessor > Defined symbols (-D)" to configure the option to "__STARTUP_CLEAR_BSS" as shown in Figure 2-5.

Figure 2-5 Cross ARM GNU Assembler > Preprocessor Configuration



Cross ARM C Compiler > Includes Configuration

Select "Cross ARM C Compiler > Includes > Include paths (-I)" to configure the C header file path, as shown in "${workspace_loc:/${ProjName}/src/library/libraries/cmsis/cm1/core_support/gmd}"

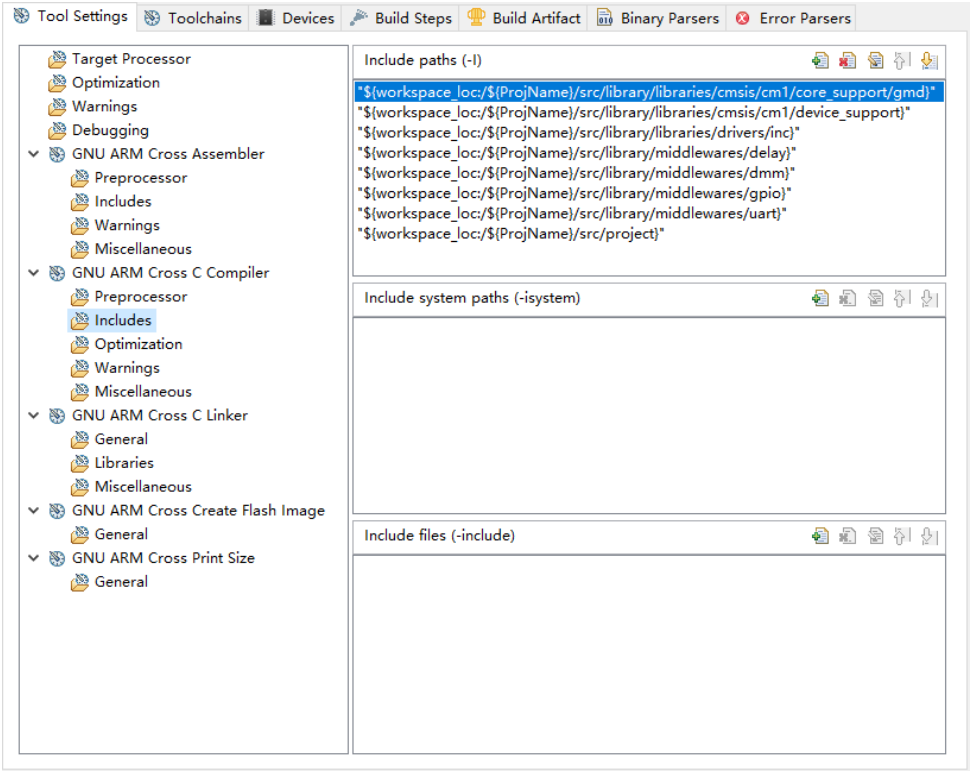
- "${workspace_loc:/${ProjName}/src/library/libraries/cmsis/cm1/device_support}"
- "${workspace_loc:/${ProjName}/src/library/libraries/drivers/inc}"
- "${workspace_loc:/${ProjName}/src/library/middlewares/delay}"
- "${workspace_loc:/${ProjName}/src/library/middlewares/dmm}"
- "${workspace_loc:/${ProjName}/src/library/middlewares/gpio}"
- "${workspace_loc:/${ProjName}/src/library/middlewares/uart}"
- "${workspace_loc:/${ProjName}/src/project}"

Figure 2-6.

For example, in the software programming reference design "GMD_RefDesign\cm1_demo", the configuration of the C header file reference path is described as below.

- "${workspace_loc:/${ProjName}/src/library/libraries/cmsis/cm1/core_support/gmd}"
- "${workspace_loc:/${ProjName}/src/library/libraries/cmsis/cm1/device_support}"
- "${workspace_loc:/${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/uart}"
- "${workspace_loc:/${ProjName}/src/project}"

Figure 2-6 Cross ARM C Compiler > Includes Configuration



Cross ARM C Linker Configuration

Select "Cross ARM C Linker > General > Script files (-T)" to configure "GOWIN_M1_flash_burn.ld" or "GOWIN_M1_flash_xip.ld" as GMD Flash linker, as shown in Figure 2-7.

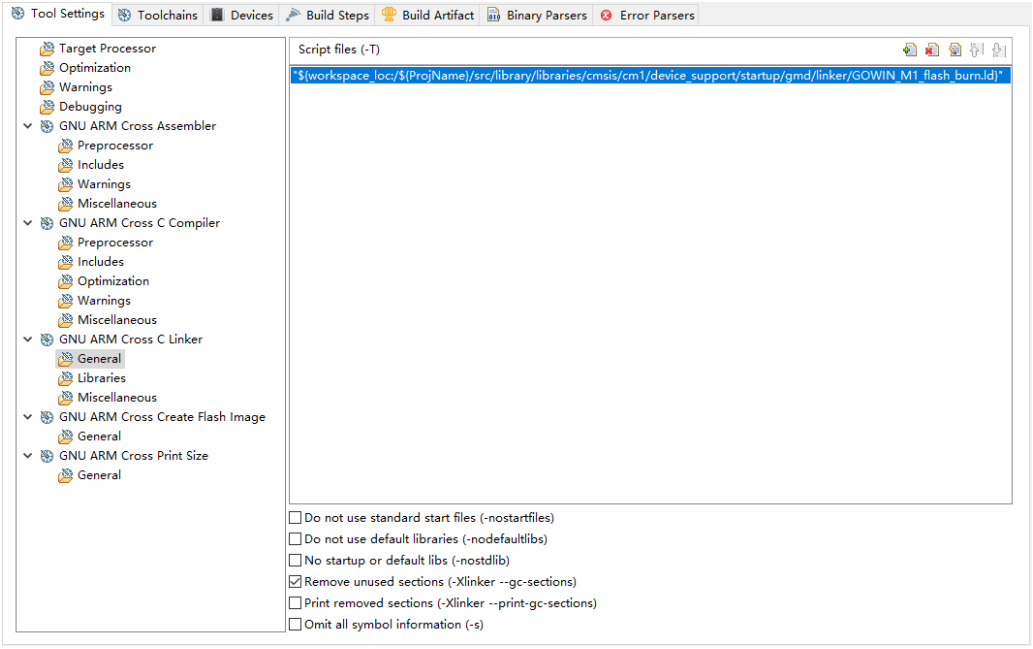
Using software programming reference design GMD_RefDesign\cm1_demo for an instance, the Flash link is configured as below.

"${workspace_loc:/${ProjName}/src/library/libraries/cmsis/cm1/device_support/startup/gmd/linker/GOWIN_M1_flash_burn.ld}"

The GMD Flash linker Flash initial address "FLASH ORIGIN" setting is shown below:

- Internal Instruction Memory:
  - GOWIN_M1_flash_xip.ld: FLASH ORIGIN: 0x00000000, ITCM Initialization download running
  - GOWIN_M1_flash_burn.ld: FLASH ORIGIN: 0x00000400, off-chip SPI-Flash memory download boot
- External Instruction Memory:
  - GOWIN_M1_flash_xip.ld: FLASH ORIGIN: 0x00000000.

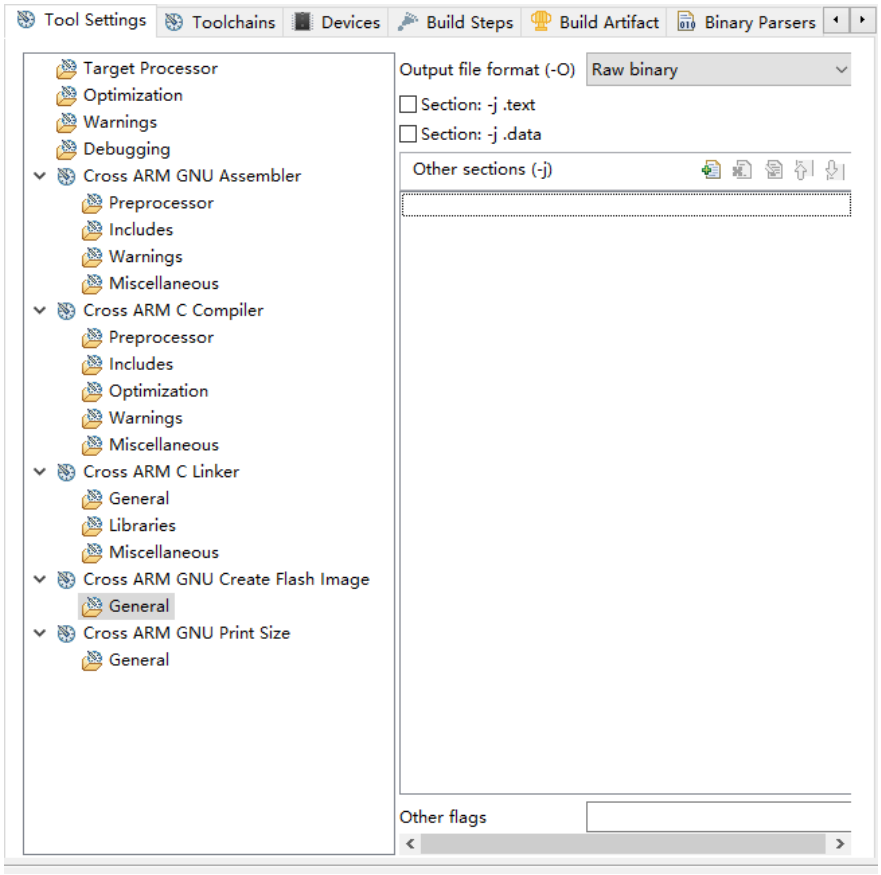
Figure 2-7 Cross ARM C Linker Configuration



Cross ARM GNU Create Flash Image Configuration

Select " Cross ARM GNU Create Flash Image > General > Output file format (-O)" to configure the option as "Raw binary" and generate software programming design BIN file, as shown in Figure 2-8.

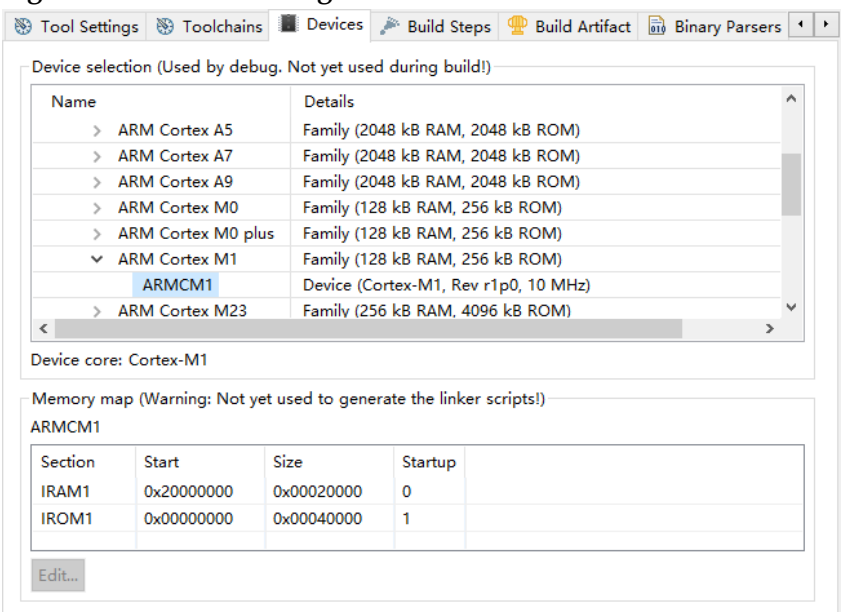
Figure 2-8 Cross ARM GNU Create Flash Image Configuration



Devices Configuration

Select " Devices > Devices" and configure the option as "ARM Cortex M1 > ARMCM1", as shown in Figure 2-9.

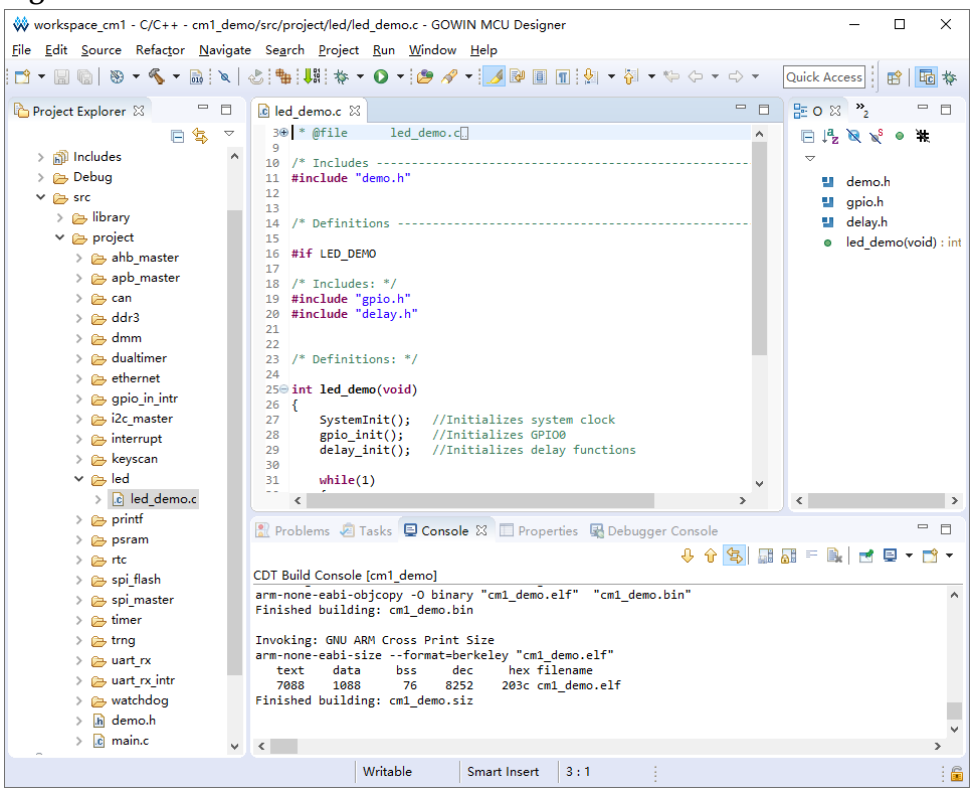
Figure 2-9 Devices Configuration



2.2.3 Build

After project option configuration and coding, click "Build" ( ) or "Build All" ( ) on the tool bar, or select "Project > Build Project" or "Project > Build All" on the menu bar to generate software design BIN file, as shown in Figure 2-10.

Figure 2-10 Build



2.2.4 Download

After building Gowin_EMPU_M1software programming design, for the downloading, see *IPUG532, Gowin_EMPU_M1Download Reference Manual*.

2.2.5 Software Online Debug

After downloading the Gowin_EMPU_M1 software programming design BIN file, if there is a problem with your software design, you can connect the development board to the J-LINK emulator and debug the current software design online (the online debugging software design must be consistent with the software design downloaded to the chip).

Note!

Gowin_EMPU_M1 does not support automatic downloading during debugging with ARM Keil software. Prior to each debugging session, please use the Programmer tool to download the Binary file of the software design you intend to debug. Then, start debugging to ensure that the software project being debugged is the current one.

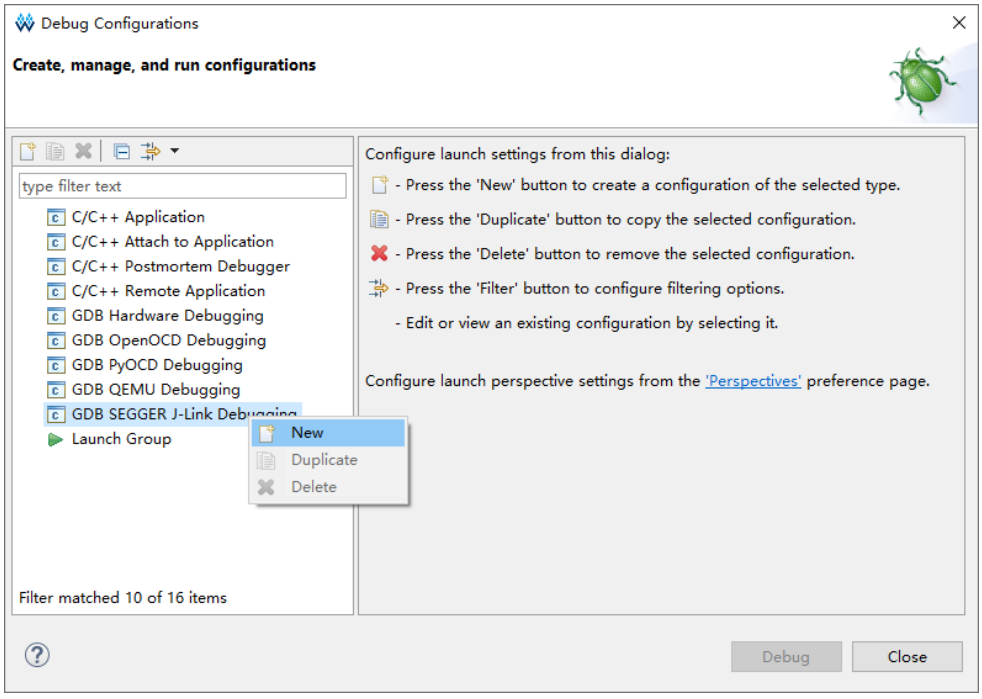
Gowin_EMPU_M1 software online debugging process includes:

- Configure software debugging options
- Configure software debugging levels
- Connect debugging emulators
- Start software online debugging

Software Debugging Configurations

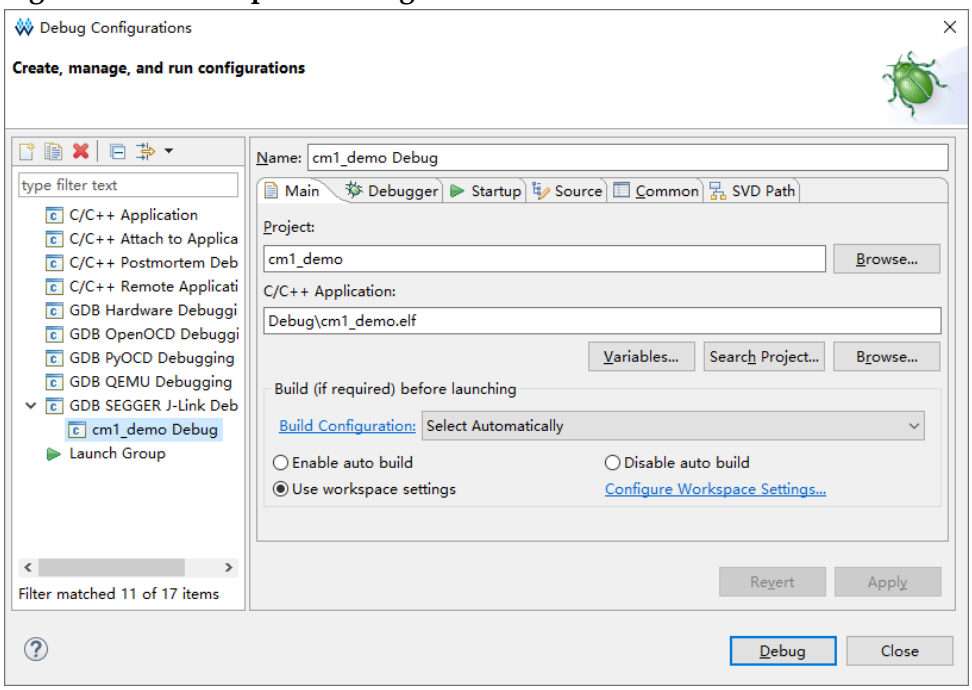
1. As shown in Figure 2-11, select "Run > Debug Configurations > GDB SEGGER J-Link Debugging > New" to create the debug configuration option of current project.

Figure 2-11 Create Software Debugging Configurations Option



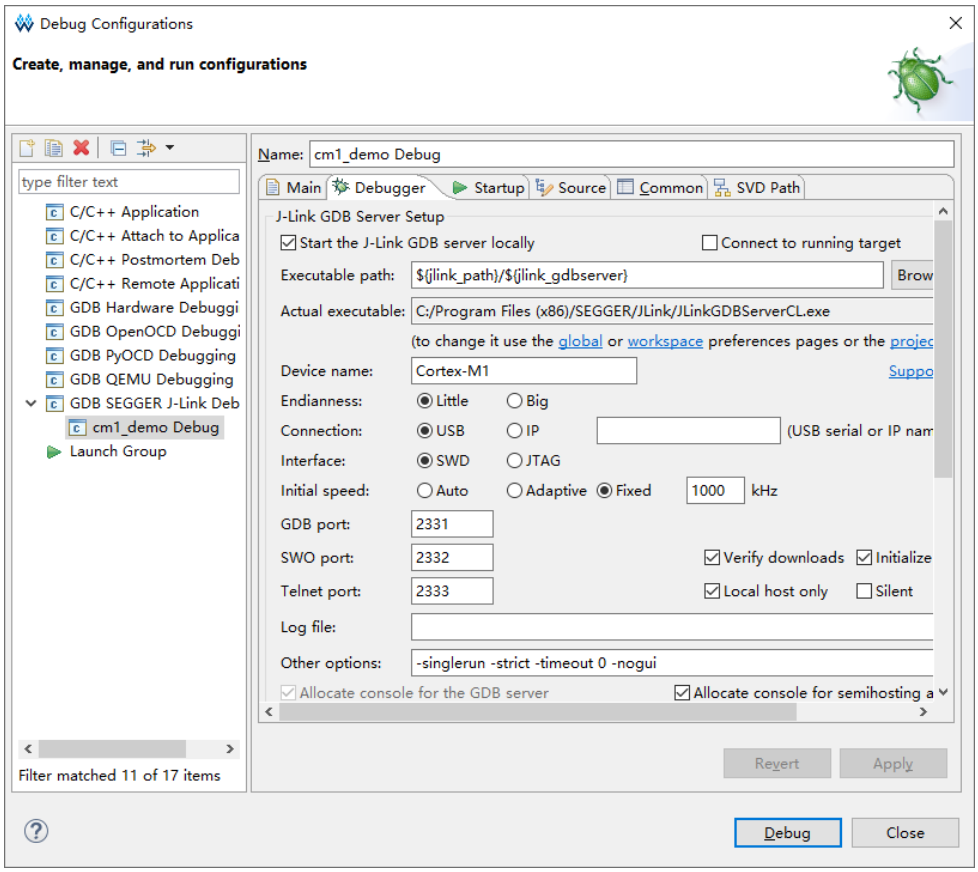
2. Select "Main" option in the created software debugging options to configure "Project" and "C/C++ Application" options of current debugging project, as shown in Figure 2-12.

Figure 2-12 Main Option Configuration



3. Select the "Debugger" option of the created software debugging options to configure the J-Link and GDB options of the current debugging project, as shown in Figure 2-13.
   - Device Name: Cortex-M1
   - Interface: JTAG or SWD
   - Endianness: Little
   - Connection: USB

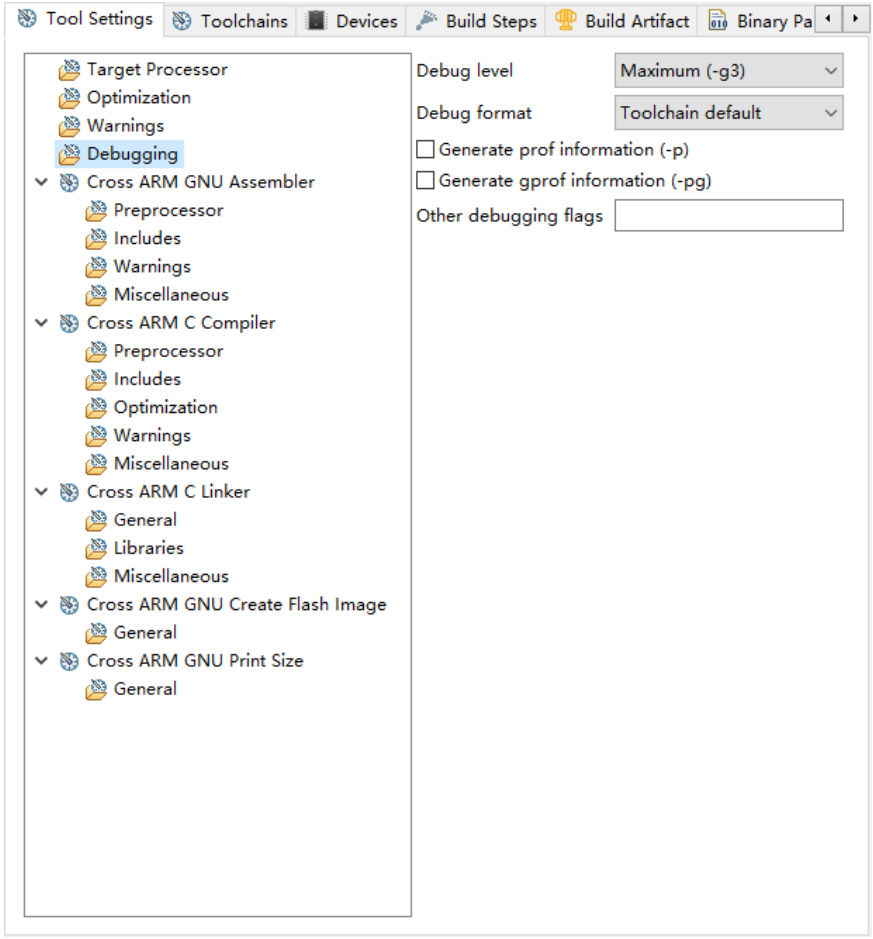
Figure 2-13 Debugger Option Configuration



Software Debugging Level Configuration

In the Project Explorer view, select "Properties > C/C++ Build > Settings > Debugging > Debug level" option of the current debugging project, and recommend configuring the debugging level as Default(-g) or Maximum(-g3), as shown in Figure 2-14.

Figure 2-14 Software Debugging Level Configuration

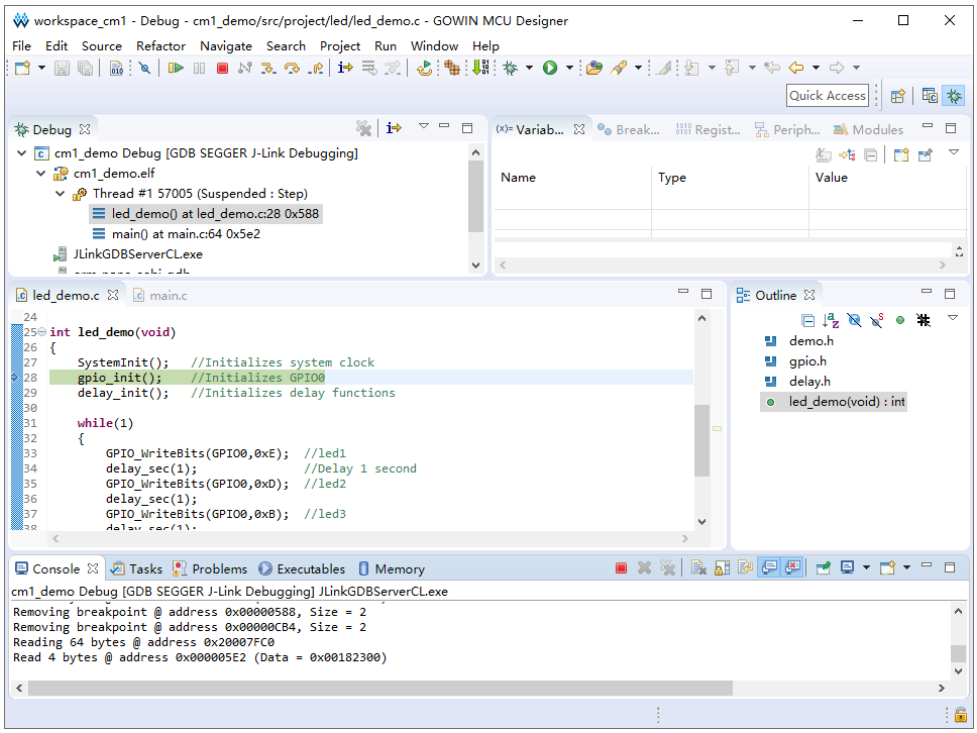


Software Online Debugging Start-up

According to the physical constraints location of JTAG debugging interface (JTAG: JTAG_3~JTAG_18, VCC and GND; or SWD: JTAG_7, JTAG_9, VCC and GND) in the hardware design, connect the J-LINK emulator and the development board.

Click "Debug" button in the tool bar to drop the list "", select the current project Debug configuration, click to enter the debug state, perform breakpoint settings, single-step debugging, reset and run, etc., as shown in Figure 2-15.

Figure 2-15 Software Online Debugging Start-up



2.3 Reference Design

Gowin_EMPU_M1 provides reference design in GMD (tested software version V1.2) software environment. Click this link to get following reference design:

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo、cm1_fatfs、cm1_freertos、cm1_rtthread_nano、cm1_ucos_iii


GOWIN
PROGRAMMING FOR THE FUTURE