




Gowin_EMPU_M1 IDE 软件 参考手册

IPUG536-2.2, 2024-07-12

版权所有 © 2024 广东高云半导体科技股份有限公司

GOWIN高云、、Gowin 以及高云均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其所有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

免责声明

本档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止反言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些档进行适时的更新。

版本信息

日期	版本	说明
2019/02/18	1.0	初始版本。
2019/07/18	1.1	<ul style="list-style-type: none">● MCU 硬件设计与软件编程设计支持扩展外部设备 CAN、Ethernet、SPI-Flash、RTC、TRNG、DualTimer、I2C、SPI、SD-Card;● MCU 支持片外 SPI-Flash 下载启动方式。
2019/08/18	1.2	<ul style="list-style-type: none">● MCU 硬件设计与软件编程设计支持扩展外部设备 DDR3 Memory;● 修复已知 ITCM、DTCM Size 和 IDE 问题。
2019/09/27	1.3	更新优化 MCU 编程软件 GOWIN MCU Designer 界面和功能。
2020/01/16	1.4	<ul style="list-style-type: none">● MCU 硬件设计与软件编程设计支持外部设备 PSRAM;● 更新 MCU 编译软件 GMD V1.0;● 更新 RTOS 参考设计;● 增加 AHB2 和 APB2 扩展总线接口硬件和软件参考设计。
2020/03/05	1.5	支持 FPGA 器件 GW2A-18C/GW2AR-18C/GW2A-55C。
2020/06/12	1.6	<ul style="list-style-type: none">● MCU 支持外部指令存储器;● MCU 支持外部数据存储器;● 扩展 6 个 AHB 总线接口;● 扩展 16 个 APB 总线接口;● GPIO 支持多种接口类型;● I²C 支持多种接口类型。
2021/07/16	1.7	更新 MCU 软件版本。
2021/10/12	1.8	GW2AN-9X/GW2AN-18X, 修改 ITCM 和 DTCM Size 范围。
2023/05/11	1.9	支持 Arora V FPGA 产品。
2023/07/21	2.0	更新已测试软件版本。
2024/03/07	2.1	<ul style="list-style-type: none">● 支持 GW5AT-60 A 版本产品;● 更新 IDE 软件的参考示例。
2024/07/12	2.2	<ul style="list-style-type: none">● 支持 GW5ART-15 A FPGA 产品;● 更新 IDE 软件的参考示例。

目录

目录	i
图目录	ii
1 ARM Keil 软件	1
1.1 软件安装	1
1.2 工程模板	1
1.2.1 创建工程	1
1.2.2 配置选项	2
1.2.3 编译	8
1.2.4 下载	9
1.2.5 软件在线调试	9
1.3 参考设计	9
2 GMD 软件	10
2.1 软件安装与配置	10
2.2 工程模板	10
2.2.1 创建工程	10
2.2.2 配置选项	12
2.2.3 编译	17
2.2.4 下载	18
2.2.5 软件在线调试	18
2.3 参考设计	22

图目录

图 1-1 创建工程	1
图 1-2 配置器件	2
图 1-3 配置 ROM 和 RAM.....	4
图 1-4 配置输出文件格式.....	5
图 1-5 配置头文件路径	5
图 1-6 配置 JTAG 调试接口类型.....	6
图 1-7 配置 SW 调试接口类型	7
图 1-8 配置 Flash 选项	7
图 1-9 配置调试初始化文件	8
图 1-10 工程编译	8
图 1-11 启动调试	9
图 2-1 创建工程	11
图 2-2 选择平台配置类型.....	11
图 2-3 选择编译工具链和路径	12
图 2-4 配置 Target Processor	13
图 2-5 配置 Cross ARM GNU Assembler > Preprocessor	14
图 2-6 配置“Cross ARM C Compiler > Includes”	15
图 2-7 配置 Cross ARM C Linker.....	16
图 2-8 配置 Cross ARM GNU Create Flash Image	16
图 2-9 配置 Devices.....	17
图 2-10 编译	17
图 2-11 建立软件调试配置选项.....	18
图 2-12 配置 Main 选项	19
图 2-13 配置 Debugger 选项	20
图 2-14 配置软件调试等级.....	21
图 2-15 启动软件在线调试.....	22

1 ARM Keil 软件

1.1 软件安装

请参考 ARM Keil MDK 网站提供的 [Getting Started with MDK](#).

注!

建议使用 ARM Keil MDK V5.26 及以上版本。

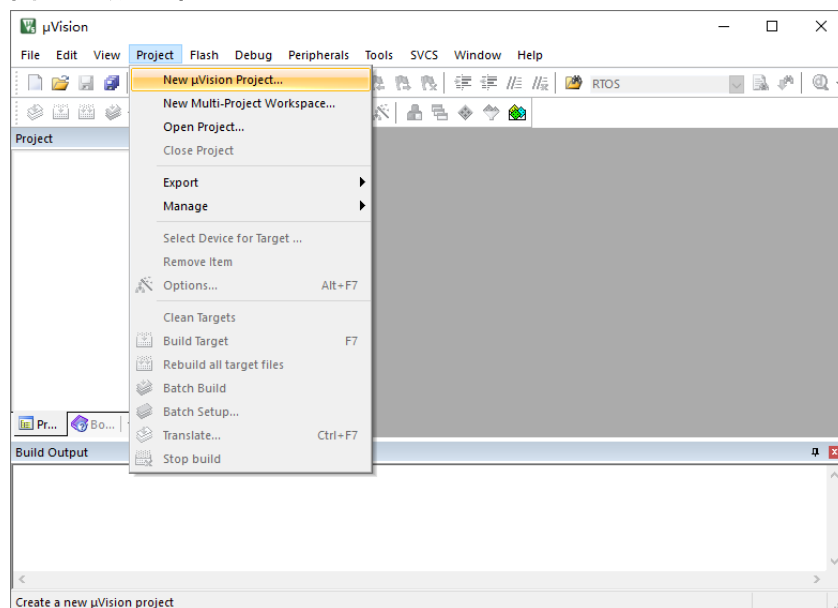
1.2 工程模板

使用 ARM Keil MDK 进行 Gowin_EMPU_M1 软件编程设计，需要创建工程、配置选项、编写代码、编译、下载和调试。

1.2.1 创建工程

双击打开 ARM Keil MDK，选择菜单栏“Project > New uVision Project...”，创建工程，如图 1-1 所示。

图 1-1 创建工程

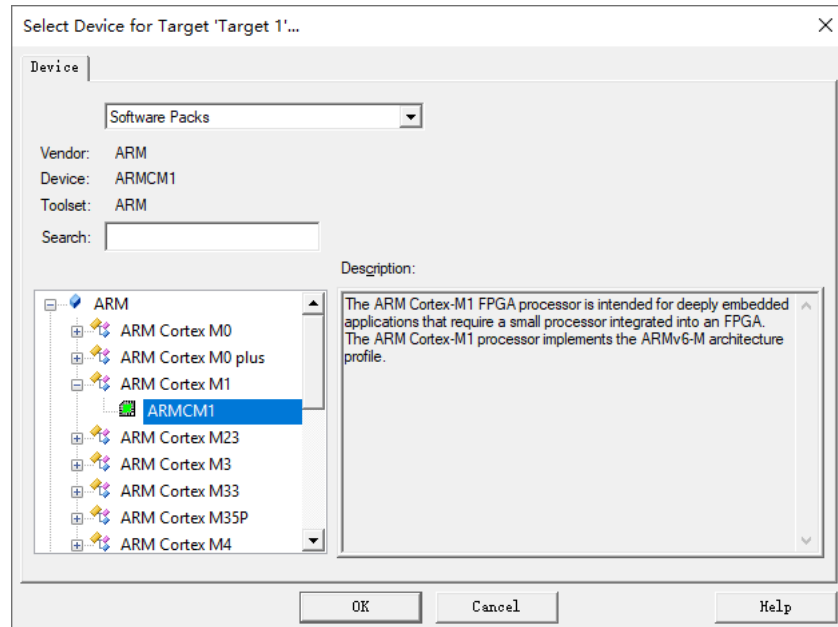


1.2.2 配置选项

配置器件

Gowin_EMPU_M1 内置 ARM Cortex-M1 内核，所以器件类型配置为“ARM Cortex M1 > ARMCM1”，如图 1-2 所示。

图 1-2 配置器件



配置 ROM 和 RAM

Gowin_EMPU_M1 的内部指令存储器或外部指令存储器作为 ROM。

Gowin_EMPU_M1 的内部数据存储器或外部数据存储器作为 RAM。

1. 配置 ROM (Internal Instruction Memory) 和 RAM (Internal Data Memory) 的起始地址和 Size。

ROM 起始地址和 Size 配置：

- 片外 SPI-Flash Memory 下载引导方式：
 - ROM 起始地址：0x400；
 - ROM Size：请根据硬件设计 ITCM Size 的实际配置来设置，例如 ITCM Size 32KB，则设置为 0x7C00。
- 片内 ITCM 初始值下载启动方式：
 - ROM 起始地址：0x00000000；
 - ROM Size：请根据硬件设计 ITCM Size 的实际配置来设置，例如 ITCM Size 32KB，则设置为 0x8000。

RAM 起始地址和 Size 配置：

- RAM 起始地址：0x20000000；
- RAM Size：请根据硬件设计 DTCM Size 的实际配置来设置，例如

DTCM Size 32KB，设置为 0x8000。

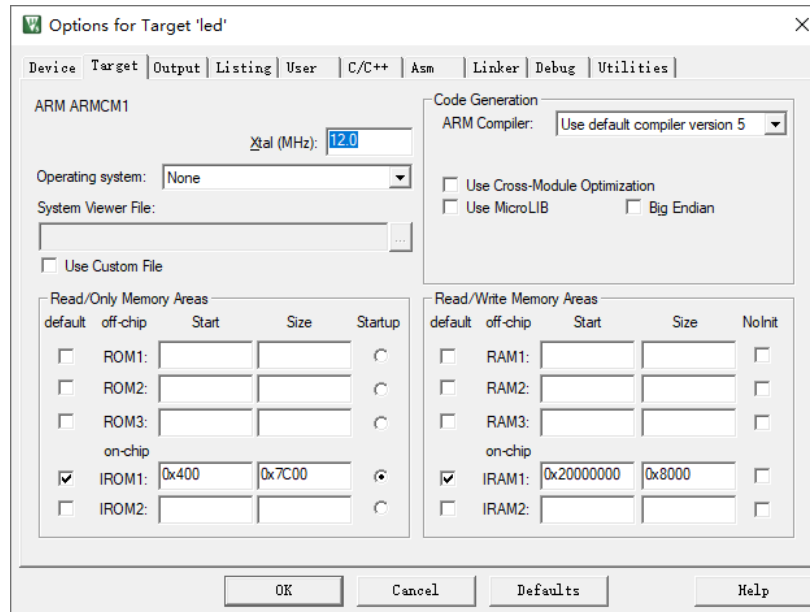
受限于片内 Block RAM 硬件存储资源，ITCM 和 DTCM 的 Size 配置不能超过片内 Block RAM 的最大存储资源：

- GW1N-9/GW1NR-9/GW1N-9 C 版/GW1NR-9 C 版，ITCM 或 DTCM 最大可配置为 32KB，如果 ITCM 或 DTCM 某个存储器已配置为 32KB，则另一个存储器最大只能配置为 16KB；
- GW2AN-9X/GW2AN-18X，ITCM 或 DTCM 最大可配置为 32KB，如果 ITCM 或 DTCM 某个存储器已配置为 32KB，则另一个存储器最大只能配置为 16KB；
- GW2A-18/GW2A-18 C 版/GW2AR-18/GW2AR-18 C 版/GW2ANR-18 C 版，ITCM 或 DTCM 最大可配置为 64KB，如果 ITCM 或 DTCM 某个存储器已配置为 64KB，则另一个存储器最大只能配置为 16KB；
- GW2A-55/GW2A-55 C 版/GW2AN-55 C 版，ITCM 或 DTCM 最大可配置为 256KB，如果 ITCM 或 DTCM 某个存储器已配置为 256KB，则另一个存储器最大只能配置为 16KB；
- GW5AT-138 B 版/GW5AST-138 B 版/GW5A-138 B 版/GW5AS-138 B 版，ITCM 或 DTCM 最大可配置为 512KB，如果 ITCM 或 DTCM 某个存储器已配置为 512KB，则另一个存储器最大只能配置为 128KB；
- GW5AT-75 B 版，ITCM 或 DTCM 最大可配置为 256KB，如果 ITCM 或 DTCM 某个存储器已配置为 256KB，则另一个存储器最大只能配置为 256KB；
- GW5A-25 A 版/GW5AR-25 A 版/GW5AS-25 A 版，ITCM 或 DTCM 最大可配置为 64KB，如果 ITCM 或 DTCM 某个存储器已配置为 64KB，则另一个存储器最大只能配置为 32KB；
- GW5AT-60 A 版，ITCM 或 DTCM 最大可配置为 128KB，如果 ITCM 或 DTCM 某个存储器已配置为 128KB，则另一个存储器最大只能配置为 64KB；
- GW5ART-15 A 版，ITCM 或 DTCM 最大可配置为 64KB，如果 ITCM 或 DTCM 某个存储器已配置为 64KB，则另一个存储器最大只能配置为 8KB。

ROM (Internal Instruction Memory) 和 RAM (Internal Data Memory) 的配置，如图 1-3 所示。

例如 DK-START-GW5A25 V2.0 开发板参考设计，“ROM”起始地址设置为“0x400”，“Size”设置为“0x7C00”，“RAM”起始地址设置为“0x20000000”，“Size”设置为“0x8000”。

图 1-3 配置 ROM 和 RAM



2. 配置 ROM（External Instruction Memory）和 RAM（External Data Memory）的起始地址和 Size。

ROM 起始地址和 Size 配置：

- ROM 起始地址：0x00000000；
- ROM Size：请根据硬件设计实际 Size 设置。

RAM 起始地址和 Size 配置：

- RAM 起始地址：0x20100000；
- RAM Size：请根据硬件设计实际 Size 设置。

配置输出文件格式

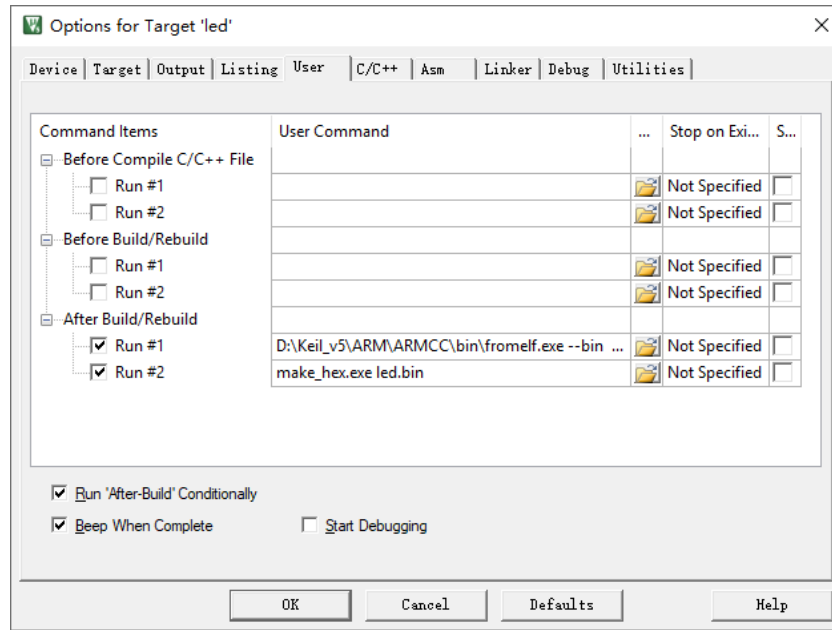
Gowin_EMPU_M1 软件编程设计，需要产生软件编程设计 Binary 文件，所以需要将 axf 文件格式转换为 Binary 文件格式。

如果使用软件编程设计可执行程序文件作为 ITCM 的初始值，需要使用 make_hex.exe 工具将软件编程设计 Binary 文件转换为四个十六进制文件 itcm0、itcm1、itcm2 和 itcm3。

User 命令行选项中文件格式转换工具调用方法，如图 1-4 所示。

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

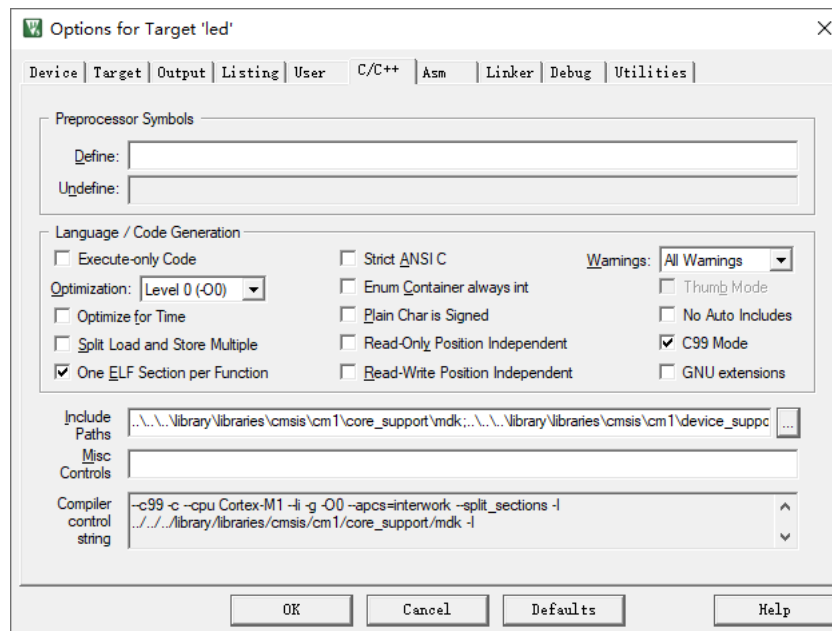
图 1-4 配置输出文件格式



配置头文件路径

配置 C 代码头文件路径，编译过程中用来调用 C 代码头文件，配置如图 1-5 所示。

图 1-5 配置头文件路径

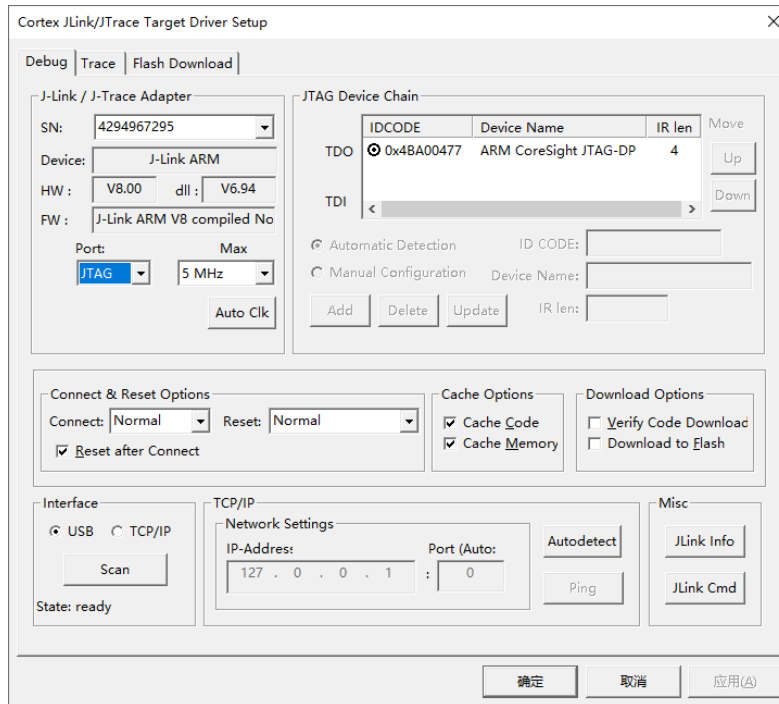


配置调试选项

- 配置仿真器类型
 - U-LINK 仿真器
 - 如果选择使用 U-LINK 仿真器，则 Debug 选项配置为 ULNK2/ME Cortex Debugger;

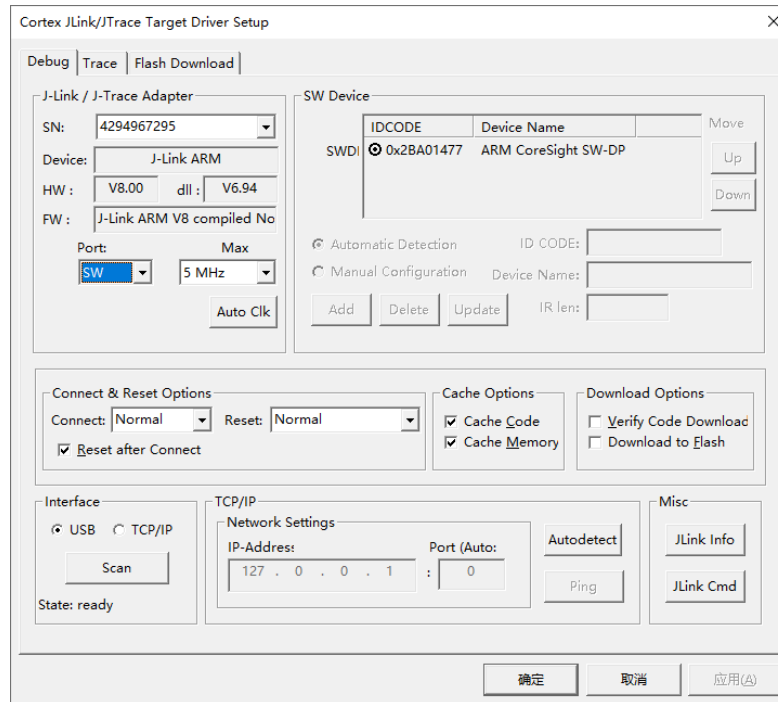
- J-LINK 仿真器
如果选择使用 J-LINK 仿真器，则 Debug 选项配置为 J-LINK/J-TRACE Cortex。
- 配置调试接口类型
 - JTAG 调试接口类型
如果配置为 JTAG 调试接口类型，配置方法如图 1-6 所示。

图 1-6 配置 JTAG 调试接口类型



- SW 调试接口类型
如果配置为 SW 调试接口类型，配置方法如图 1-7 所示。

图 1-7 配置 SW 调试接口类型



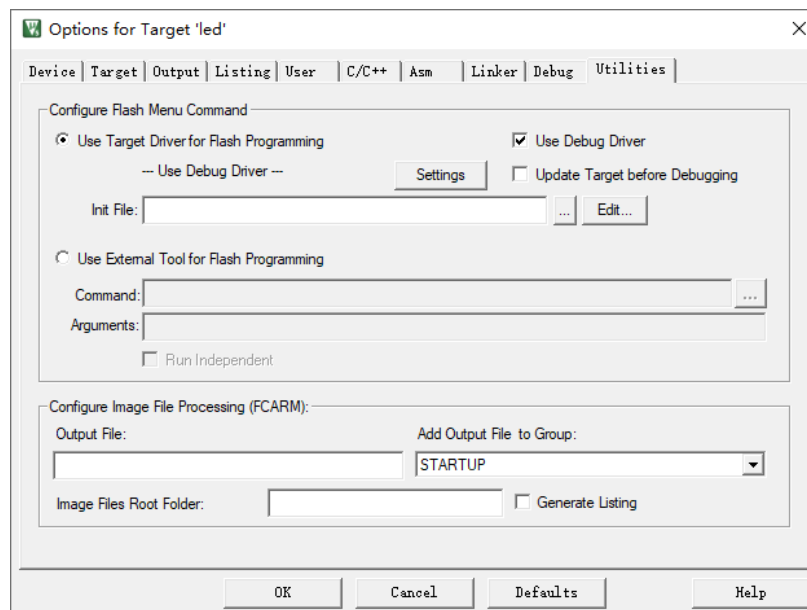
调试接口类型配置选项中：

- 请关闭“Download Options > Verify Code Download”选项。
- 请关闭“Download Options > Download to Flash”选项。

配置 Flash 选项

如果需要在线调试，请关闭“Update Target before Debugging”选项，如图 1-8 所示。

图 1-8 配置 Flash 选项

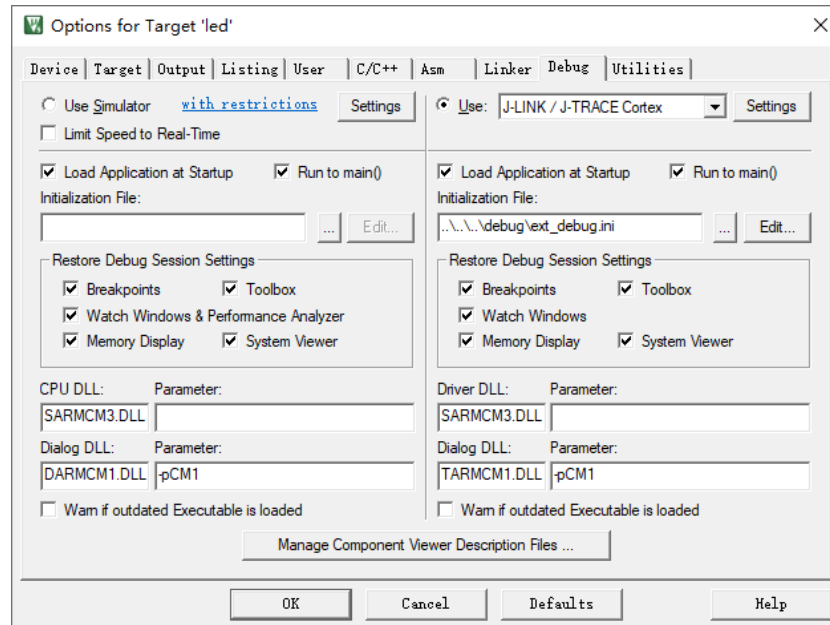


配置调试初始化文件

如果选择片外 SPI-Flash Memory 下载引导方式，软件在线调试时需要加载调试初始化文件，“Debug > Initialization File” 选项中，选择加载调试初始化文件 “ext_debug.ini”，如图 1-9 所示。

“ext_debug.ini” 文件位于 library\debug。

图 1-9 配置调试初始化文件



1.2.3 编译



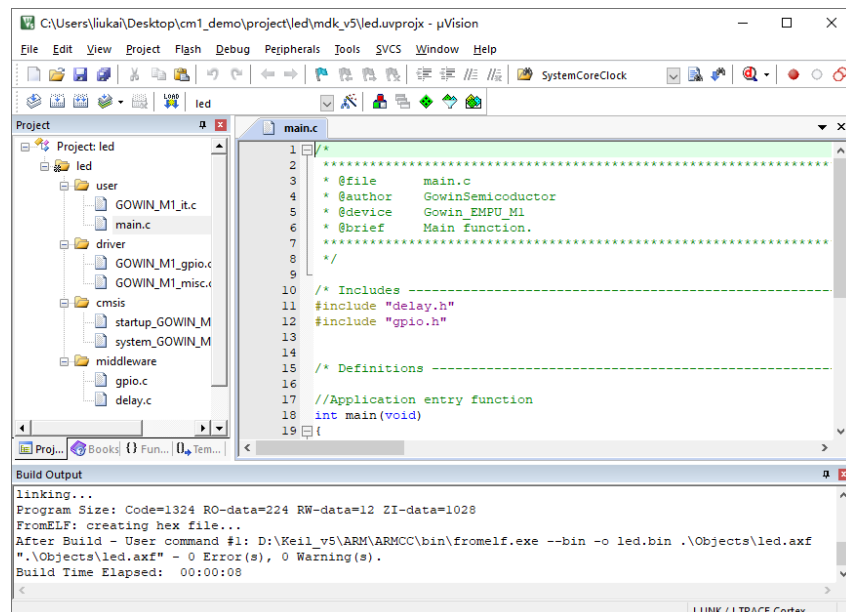
完成代码编写和选项配置后，单击工具栏 “Build” () 或 “Rebuild” ()，或菜单栏 “Project > Build Target” 或 “Project > Rebuild all target files”，编译产生软件编程设计 Binary 文件或四个十六进制文件 itcm0、itcm1、itcm2 和 itcm3，如图 1-10 所示。

图 1-10 工程编译



1.2.4 下载

完成 Gowin_EMPU_M1 软件编程设计编译后,软件编程设计下载方法,请参考: [IPUG532, Gowin_EMPU_M1 下载参考手册](#)。

1.2.5 软件在线调试

完成硬件设计产生的硬件设计码流文件和软件编程设计产生的软件编程设计 Binary 文件的下载后,如果用户软件设计出现问题,可以连接 U-LINK 或 J-LINK 仿真器在线调试。

用户可以下载、调试软件,无需重新编译硬件设计。

1. 连接仿真器

按照硬件设计中约束到 FPGA IO 的 Debug Access Port (JTAG: JTAG_3~JTAG_18、VCC 和 GND; 或 SWD: JTAG_7、JTAG_9、VCC 和 GND) 的位置,连接 J-LINK 或 U-LINK。

2. 启动调试


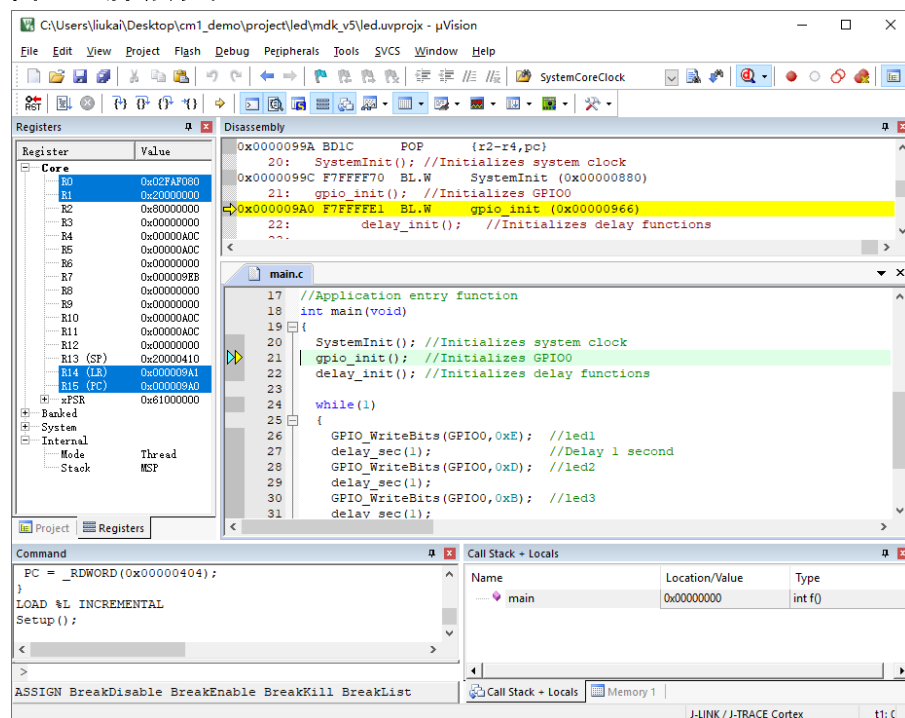
连接 U-LINK 或 J-LINK 仿真器,单击工具栏 “Debug” () ,或菜单栏 “Debug > Start/Stop Debug Session”,进入调试状态,可以执行断点设置、单步调试、复位和运行等操作,如图 1-11 所示。

图 1-11 启动调试



1.3 参考设计

Gowin_EMPU_M1 支持 ARM Keil MDK (已测试软件版本: V5.26) 软件环境的参考设计,通过链接获取如下[参考设计](#):

...\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo、
cm1_fatfs、cm1_freertos、cm1_rttthread_nano、cm1_tcpip、cm1_ucos_iii

2 GMD 软件

2.1 软件安装与配置

高云半导体网站提供 GMD 软件安装包下载
<http://www.gowinsemi.com.cn/prodshow.aspx>。

GMD 软件安装与配置，请参考 [SUG549, GOWIN MCU Designer 用户指南](#)。

注！


建议使用 GMD V1.2 及以上版本。

2.2 工程模板

使用 GMD 进行 Gowin_EMPU_M1 软件编程设计，需要创建工程、配置选项、编写代码、编译、下载和在线调试。

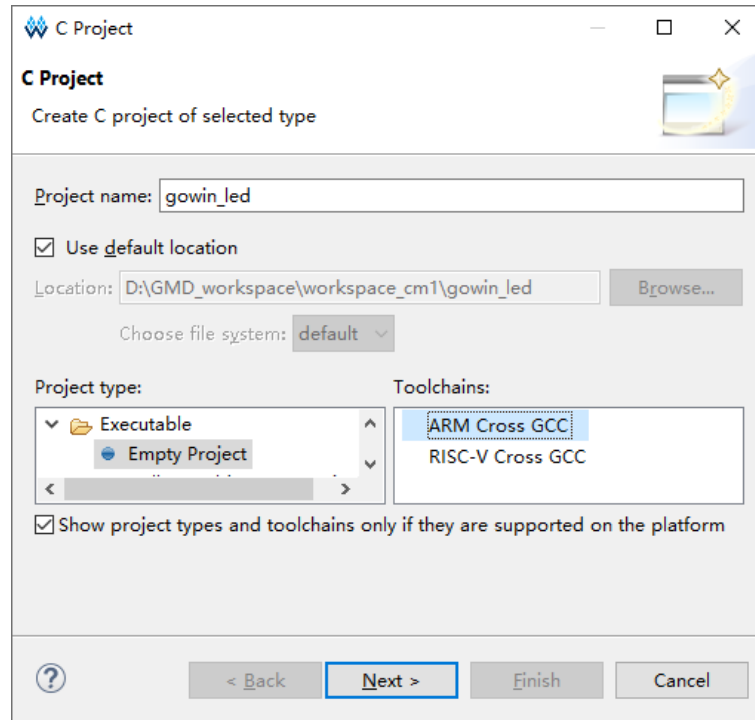
2.2.1 创建工程

创建工程

选择工具栏 “New” () 或菜单栏 “File > New > C Project”，如图 2-1 所示。

1. 建立项目名称和项目位置；
2. 选择项目类型 “Empty Project”；
3. 选择编译工具链 “ARM Cross GCC”。

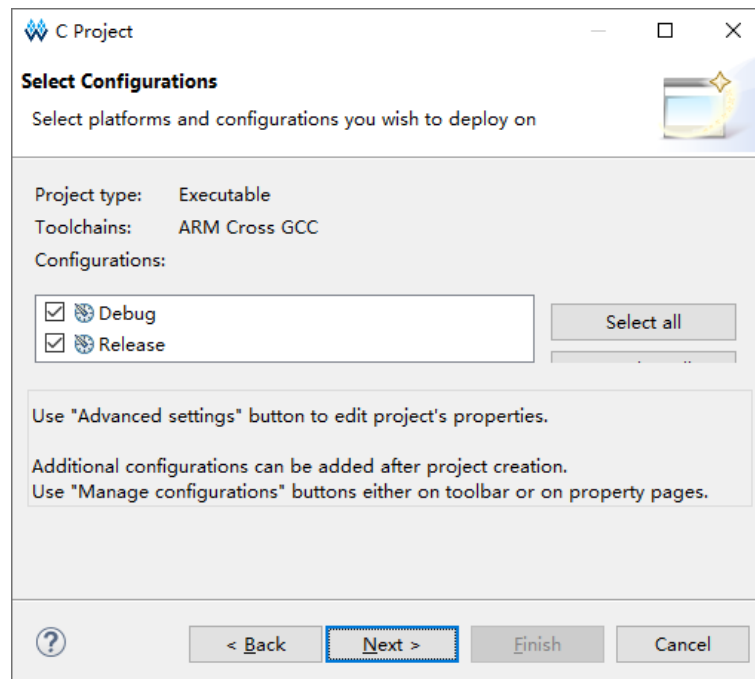
图 2-1 创建工程



选择平台配置类型

选择平台配置类型“Debug”和“Release”，如图 2-2 所示。

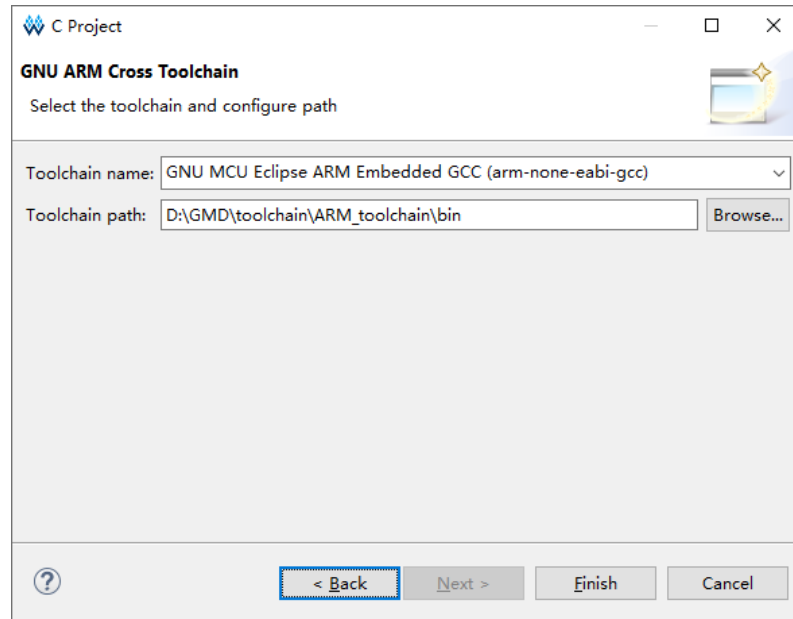
图 2-2 选择平台配置类型



选择编译工具链和路径

选择交叉编译工具链 `arm-none-eabi-gcc` 及其所在路径，建议默认配置 Toolchain name 和 Toolchain path，如图 2-3 所示。

图 2-3 选择编译工具链和路径



建立项目工程

完成工程创建后，在 **Project Explorer** 视图选择新建的项目工程，添加工程结构和代码，导入软件编程设计。

在 **Project Explorer** 视图选择当前工程，右键选择“**Refresh**”选项，可以自动更新当前项目工程的结构和代码。

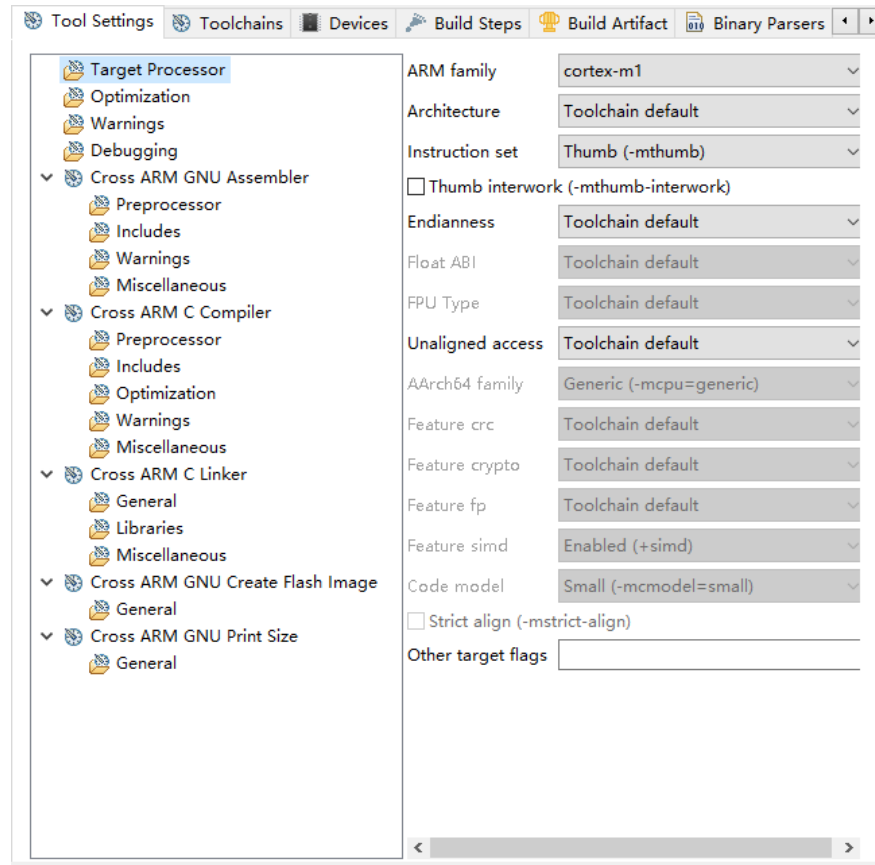
2.2.2 配置选项

Project Explorer 视图中，选择当前工程，右键选择“**Properties > C/C++ Build > Settings**”，配置当前工程的参数选项。

配置 Target Processor

配置“**Target Processor > ARM family**”选项，该选项配置为“**cortex-m1**”，如图 2-4 所示。

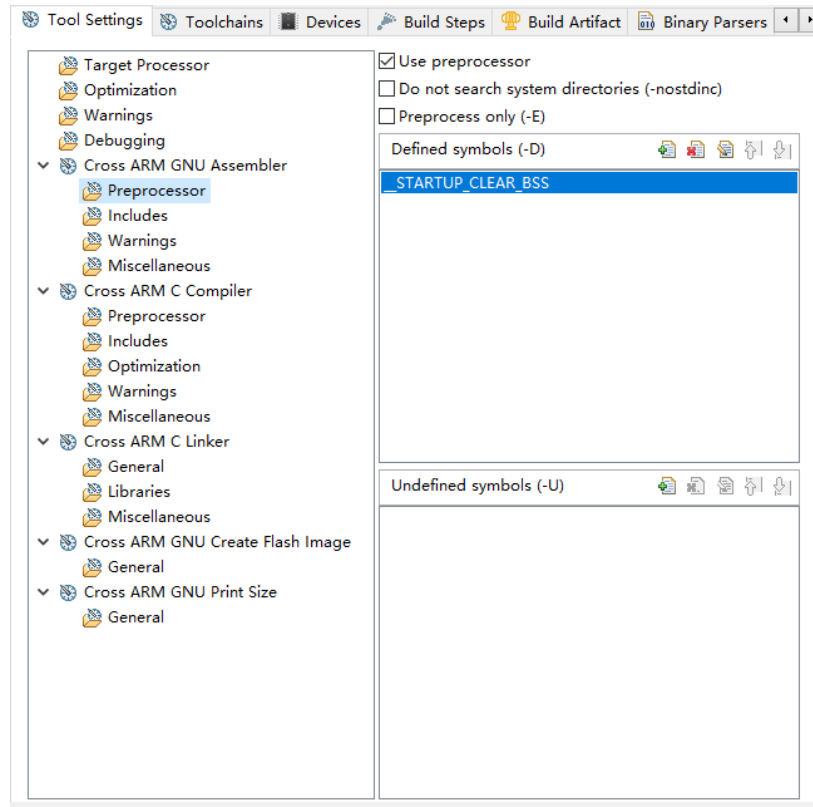
图 2-4 配置 Target Processor



配置 Cross ARM GNU Assembler > Preprocessor

配置 “Cross ARM GNU Assembler > Preprocessor > Defined symbols (-D)” 选项，该选项配置为 “`__STARTUP_CLEAR_BSS`”，如图 2-5 所示。

图 2-5 配置 Cross ARM GNU Assembler > Preprocessor



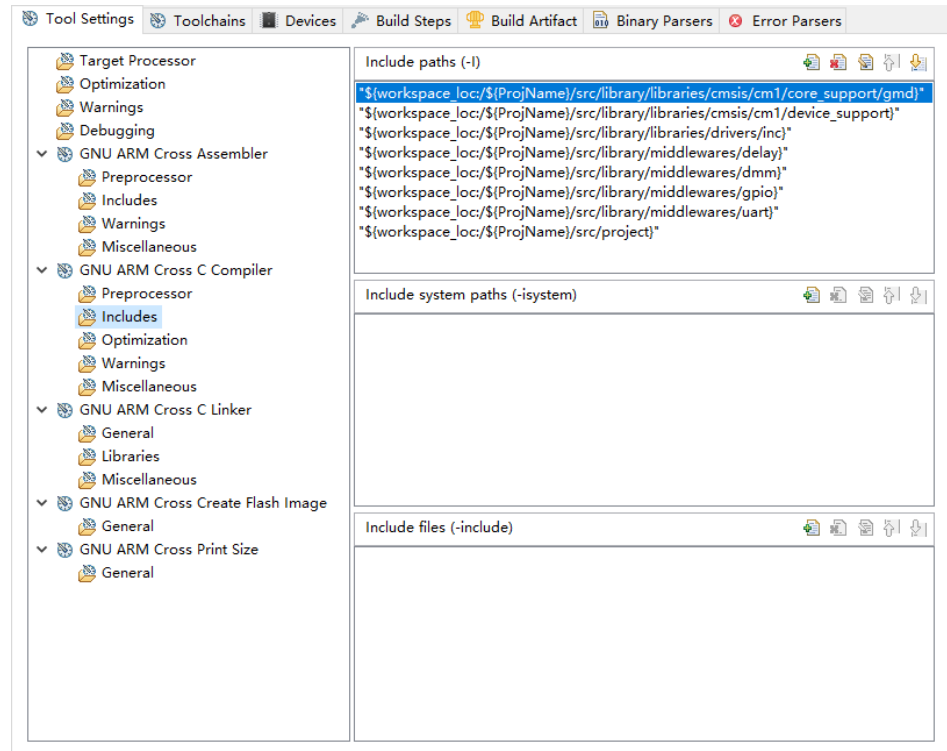
配置 Cross ARM C Compiler > Includes

配置 “Cross ARM C Compiler > Includes > Include paths (-I)” 选项，该选项配置 C 头文件引用路径，如图 2-6 所示。

例如软件编程参考设计 GMD_RefDesign\cm1_demo，C 头文件引用路径配置，如下所示：

- "\${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}"

图 2-6 配置 “Cross ARM C Compiler > Includes”



配置 Cross ARM C Linker

配置 “Cross ARM C Linker > General > Script files (-T)” 选项，该选项配置 “GOWIN_M1_flash_burn.ld” 或 “GOWIN_M1_flash_xip.ld” 作为 GMD Flash 链接器，如图 2-7 所示。

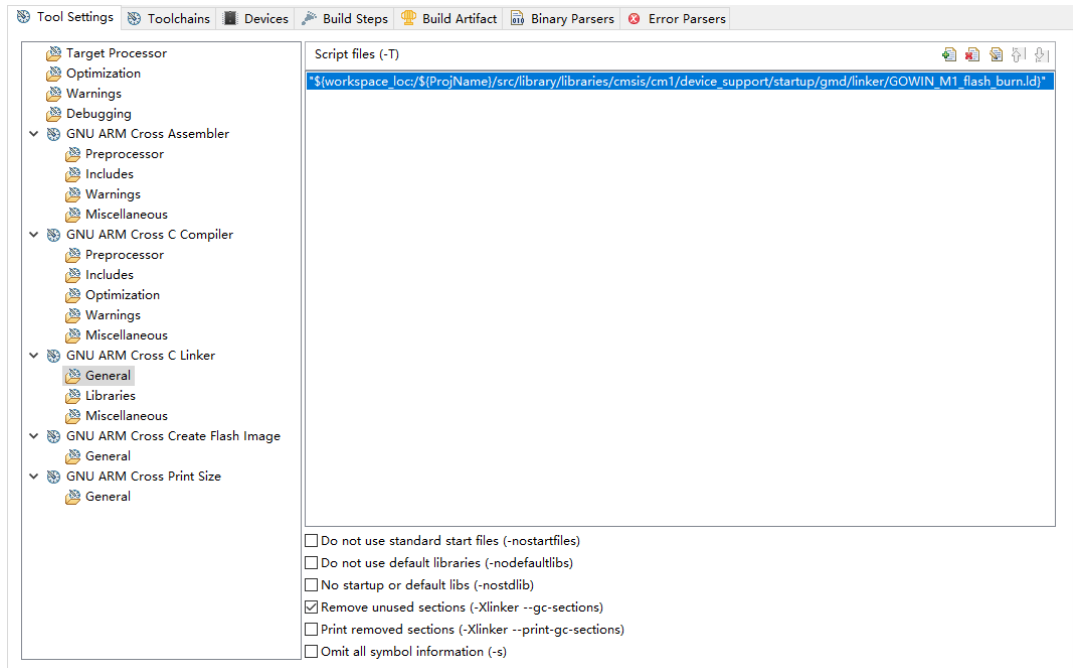
例如软件编程参考设计 GMD_RefDesign\cm1_demo，Flash 链接器配置，如下所示：

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

GMD Flash 链接器 Flash 起始地址 “FLASH ORIGIN” 设置，如下所示：

- Internal Instruction Memory
 - GOWIN_M1_flash_xip.ld: FLASH ORIGIN: 0x00000000, ITCM Initialization 下载运行方式；
 - GOWIN_M1_flash_burn.ld: FLASH ORIGIN: 0x00000400, 片外 SPI-Flash Memory 下载引导方式。
- External Instruction Memory
 - GOWIN_M1_flash_xip.ld: FLASH ORIGIN: 0x00000000。

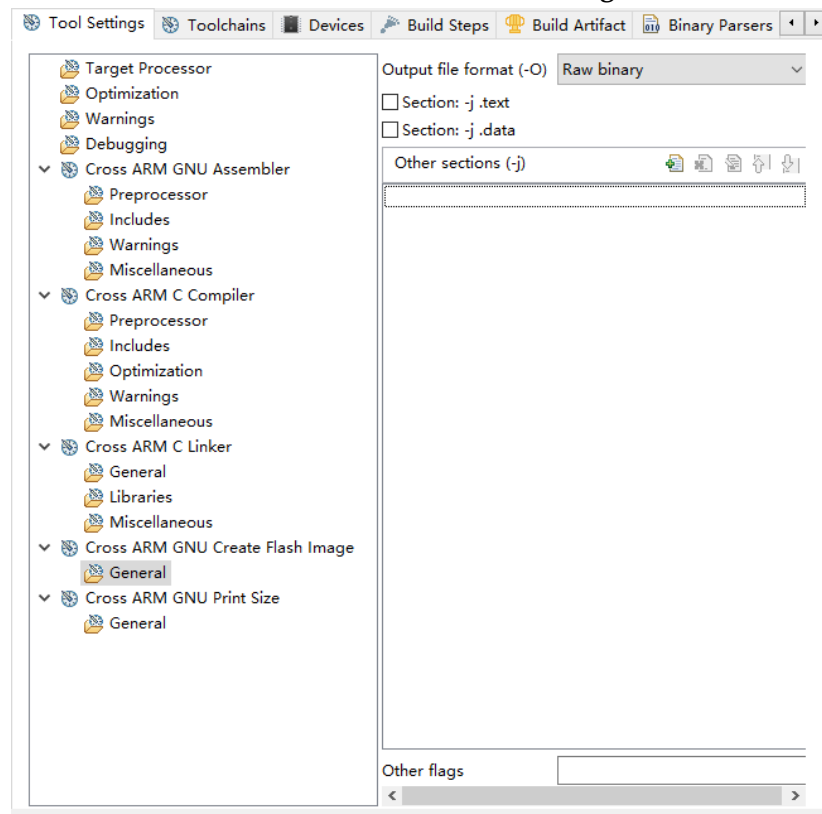
图 2-7 配置 Cross ARM C Linker



配置 Cross ARM GNU Create Flash Image

配置“Cross ARM GNU Create Flash Image > General > Output file format (-O)”选项，该选项配置为“Raw binary”，产生软件编程设计 Binary 文件，如图 2-8 所示。

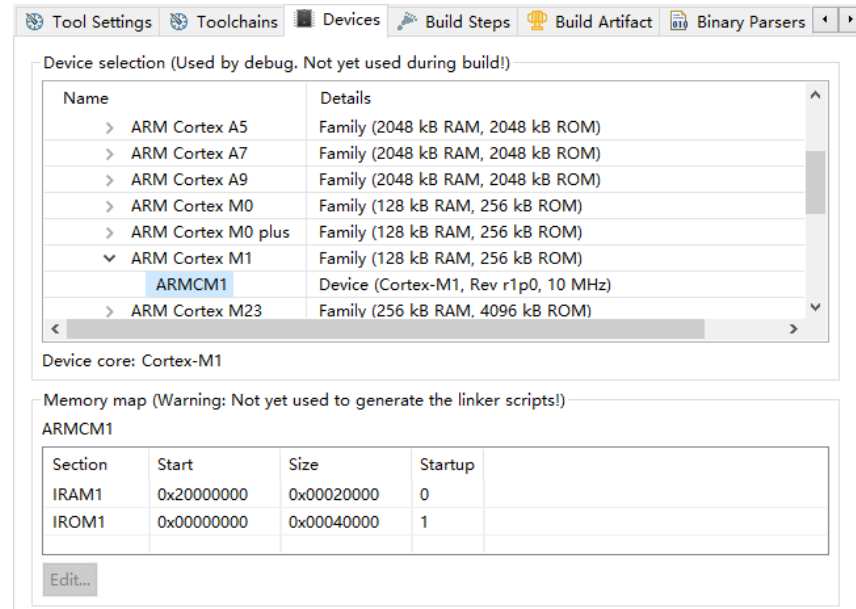
图 2-8 配置 Cross ARM GNU Create Flash Image



配置 Devices

配置“Devices > Devices”选项，该选项配置为“ARM Cortex M1 > ARMCM1”，如图 2-9 所示。

图 2-9 配置 Devices



2.2.3 编译



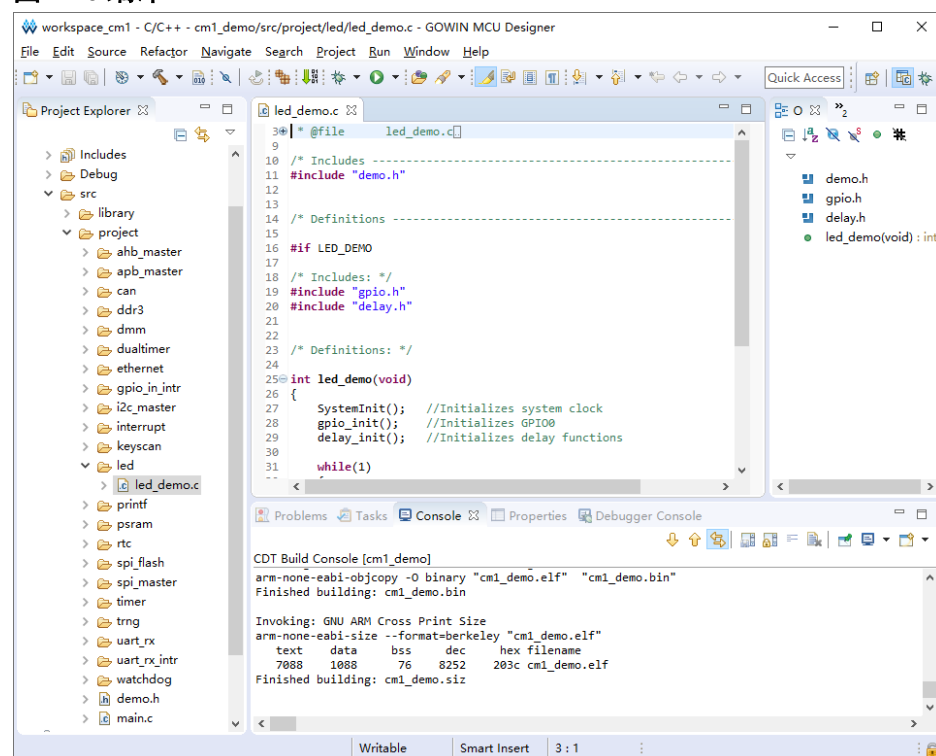
完成工程选项配置和代码编写后，编译当前工程，单击工具栏 Build“”或 Build All“”，或菜单栏“Project > Build Project”或“Project > Build All”，编译产生软件编程设计 Binary 文件，如图 2-10 所示。

图 2-10 编译



2.2.4 下载

完成 Gowin_EMPU_M1 软件编程设计编译后,软件编程设计下载方法,请参考: [IPUG532, Gowin_EMPU_M1 下载参考手册](#)。

2.2.5 软件在线调试

完成 Gowin_EMPU_M1 软件编程设计 Binary 文件下载后, 如果用户软件设计出现问题, 可以连接开发板与 J-LINK 仿真器, 在线调试当前软件设计 (在线调试的软件设计必须与下载到芯片中的软件设计保持一致)。

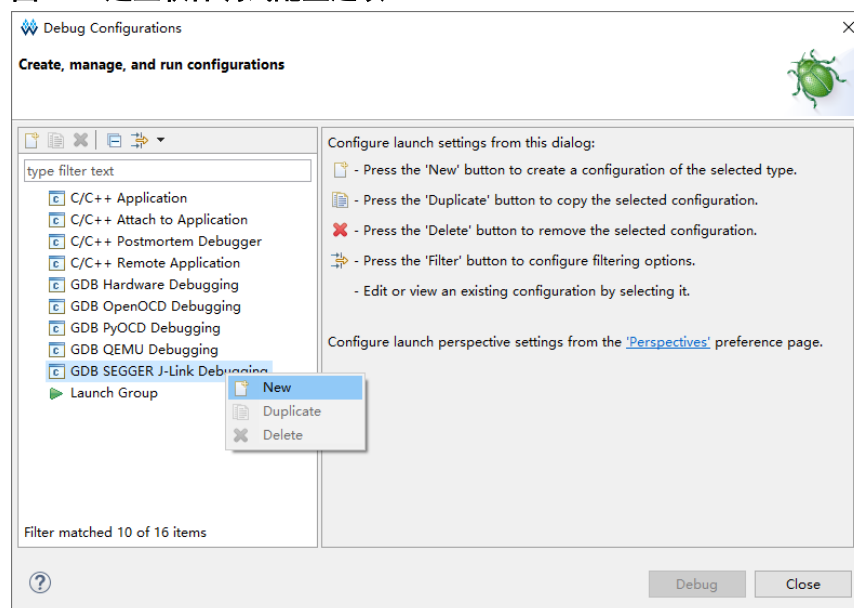
Gowin_EMPU_M1 软件在线调试流程, 包括:

- 配置软件调试选项
- 配置软件调试等级
- 连接调试仿真器
- 启动软件在线调试

配置软件调试选项

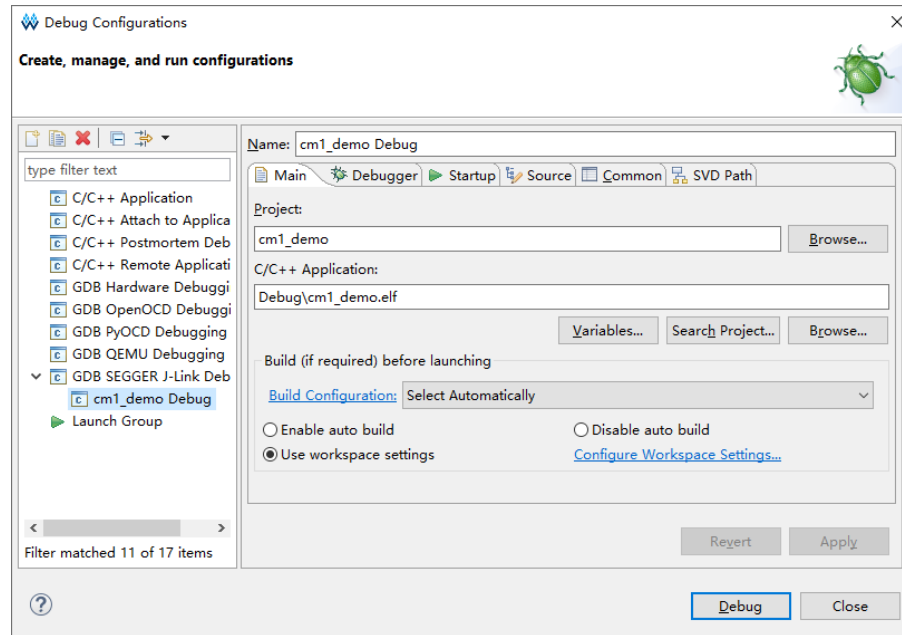
1. 选择菜单栏 “Run > Debug Configurations > GDB SEGGER J-Link Debugging”, 右键选择 “New”, 建立当前工程的调试配置选项, 如图 2-11 所示。

图 2-11 建立软件调试配置选项



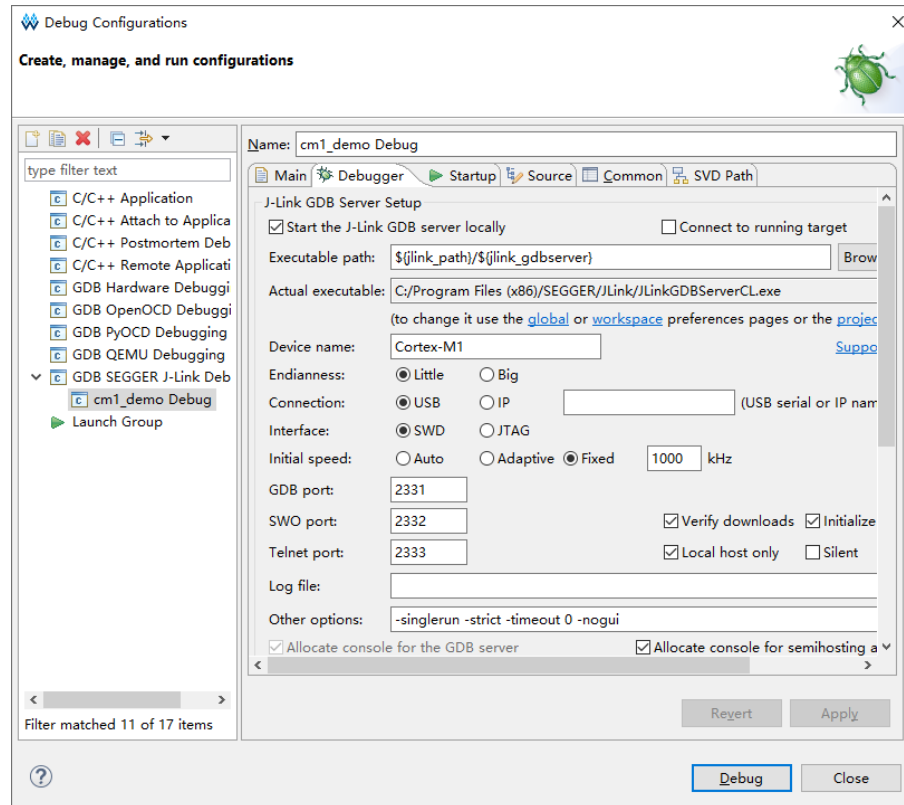
2. 选择已建立的软件调试选项的“Main”选项, 配置当前调试工程的 Project 和 C/C++ Application 等选项, 如图 2-12 所示。

图 2-12 配置 Main 选项



3. 选择已建立的软件调试选项的“Debugger”选项，配置当前调试工程的 J-Link 和 GDB 等选项，如图 2-13 所示。
 - Device Name: Cortex-M1
 - Interface: JTAG 或 SWD
 - Endianness: Little
 - Connection: USB

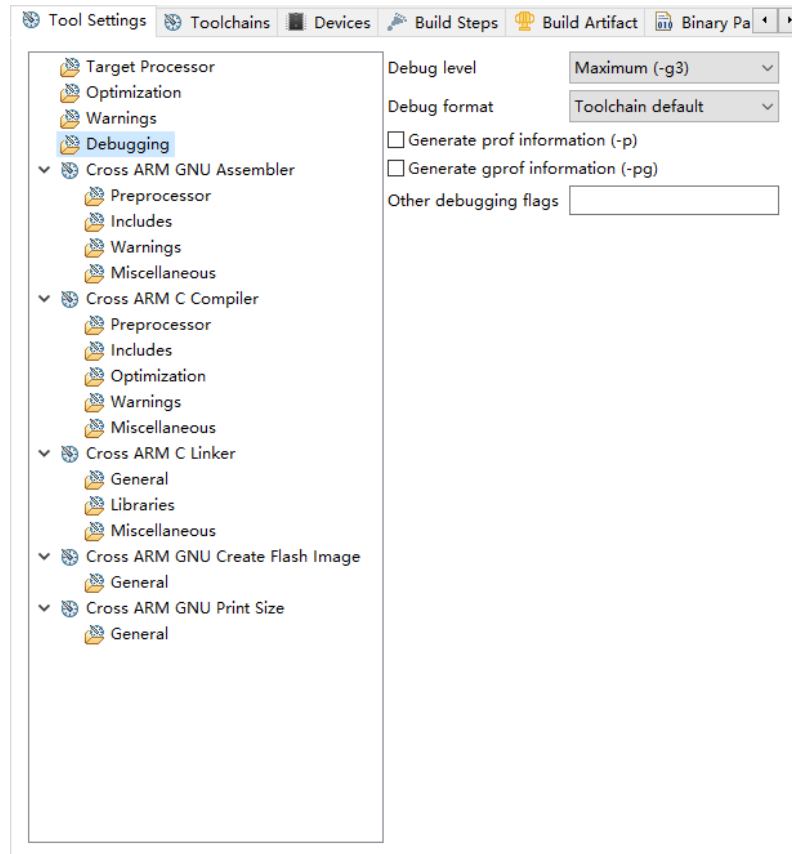
图 2-13 配置 Debugger 选项



配置软件调试等级

Project Explorer 视图中, 选择当前调试项目工程的“Properties > C/C++ Build > Settings > Debugging > Debug level”选项, 建议配置调试等级为 Default(-g)或 Maximum(-g3), 如图 2-15 所示。

图 2-14 配置软件调试等级



启动软件在线调试

按照硬件设计 JTAG 调试接口（JTAG: JTAG_3~JTAG_18、VCC 和 GND；或 SWD: JTAG_7、JTAG_9、VCC 和 GND）的物理约束位置，连接 J-LINK 仿真器与开发板。


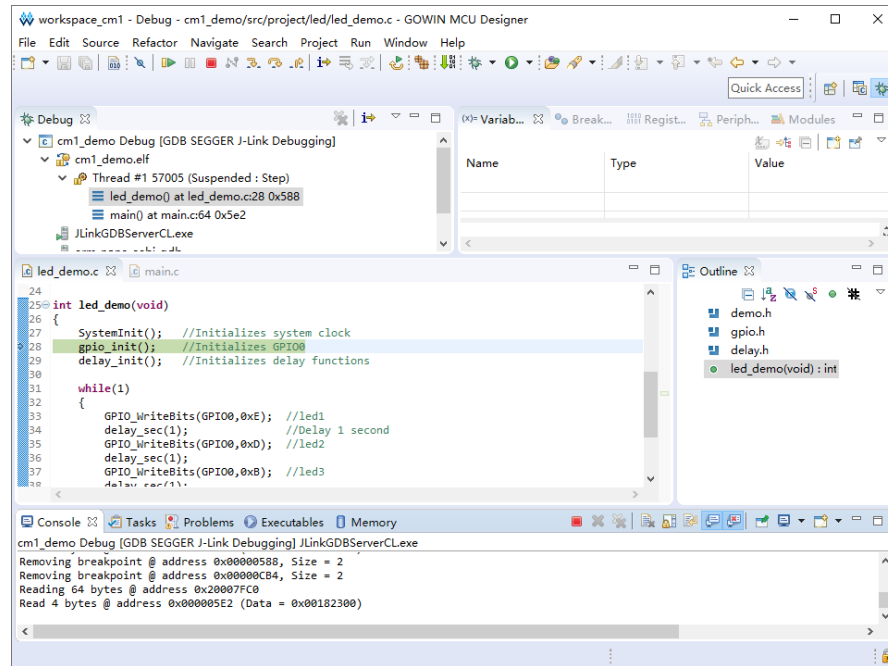
单击工具栏 Debug 按钮下拉列表“”，选择当前项目的 Debug 配置，单击进入调试状态，执行断点设置、单步调试、复位和运行等操作，如图 2-15 所示。

图 2-15 启动软件在线调试



2.3 参考设计

Gowin_EMPU_M1 支持 GMD（已测试软件版本：V1.2）软件环境的参考设计，通过链接获取如下[参考设计](#)：

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

