



# Gowin\_EMPU\_M1 IDE 软件 参考手册

IPUG536-1.4, 2020-01-16

## **版权所有©2020 广东高云半导体科技股份有限公司**

未经本公司书面许可，任何单位和个人不得擅自摘抄、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，或以禁止发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"><li>● 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;</li><li>● MCU 支持片外 SPI-Flash 下载启动方式。</li></ul>
2019/08/18	1.2	<ul style="list-style-type: none"><li>● MCU 硬件设计与软件编程设计支持扩展外部设备 DDR3 Memory;</li><li>● 修复已知 ITCM、DTCM Size 和 IDE 问题。</li></ul>
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"><li>● MCU 硬件设计与软件编程设计支持外部设备 PSRAM;</li><li>● 更新 MCU 编译软件 GMD V1.0;</li><li>● 更新 RTOS 参考设计;</li><li>● 增加 AHB2 和 APB2 扩展总线接口硬件和软件参考设计。</li></ul>

# 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 ..... 2

        1.2.3 编译 ..... 8

        1.2.4 下载 ..... 9

        1.2.5 软件调试 ..... 9

    1.3 参考设计 ..... 11

**2 GOWIN MCU Designer ..... 12**

    2.1 软件安装与配置 ..... 12

    2.2 工程模板 ..... 12

        2.2.1 工程创建 ..... 12

        2.2.2 工程配置 ..... 15

        2.2.3 编译 ..... 20

        2.2.4 下载 ..... 20

        2.2.5 调试 ..... 20

    2.3 参考设计 ..... 23

# 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.....	3
图 1-4 配置输出文件格式.....	4
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 .....	6
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 .....	9
图 1-11 仿真器连接 .....	10
图 1-12 启动调试 .....	10
图 2-1 新建工程 .....	13
图 2-2 选择平台配置类型.....	14
图 2-3 选择编译工具链和路径.....	14
图 2-4 配置 Target Processor .....	15
图 2-5 配置 Cross ARM GNU Assembler Preprocessor.....	16
图 2-6 配置 Cross ARM C Compiler Includes.....	17
图 2-7 配置 Cross ARM C Linker.....	18
图 2-8 配置 Cross ARM GNU Create Flash Image .....	19
图 2-9 配置 Devices.....	19
图 2-10 编译 .....	20
图 2-11 建立调试配置选项 .....	21
图 2-12 配置 Image 路径 .....	21
图 2-13 配置 Debugger.....	22
图 2-14 启动调试 .....	22

# 1 ARM Keil 软件

## 1.1 软件安装

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

注!

建议使用 ARM Keil MDK V5.24.2.0 及以上版本。

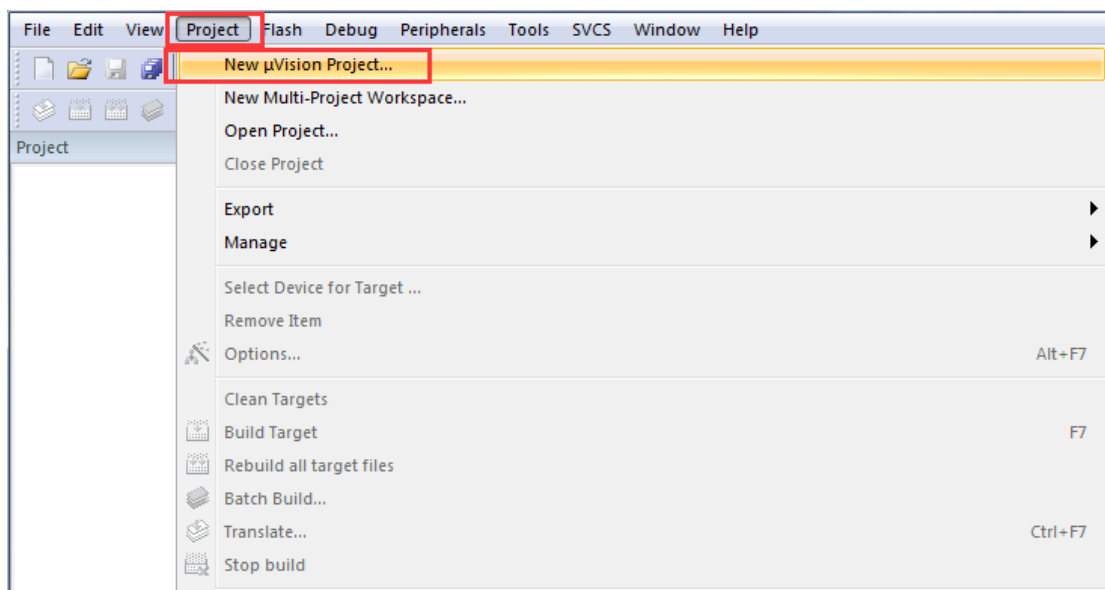
## 1.2 工程模板

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、编译和调试。

### 1.2.1 创建工程

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

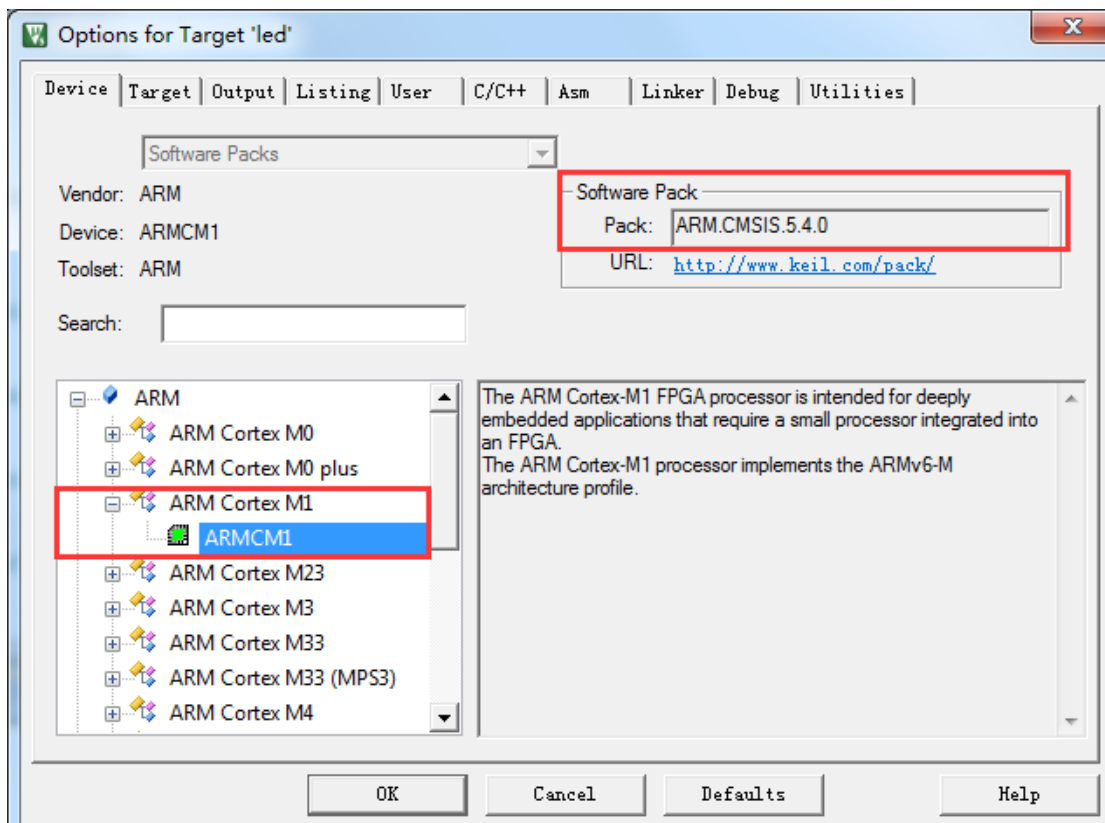


## 1.2.2 配置

### 配置器件

Gowin\_EMPU\_M1 内置 ARM Cortex-M1 内核，所以器件选择 ARM Cortex M1 的“ARMCM1”，如图 1-2 所示。

图 1-2 配置器件



### 配置 ROM 和 RAM

Gowin\_EMPU\_M1 的 ITCM 作为 ROM。

Gowin\_EMPU\_M1 的 DTCM 作为 RAM。

配置 ROM (ITCM) 和 RAM (DTCM) 的起始地址和容量大小。

如果选择片外 SPI-Flash 下载启动方式，则 Gowin\_EMPU\_M1 的 ITCM 的起始地址为 0x400，容量根据硬件实际配置设定，软件开发工具包的参考设计中配置为 0x7C00。

如果选择片内 ITCM 初始值下载启动方式，则 Gowin\_EMPU\_M1 的 ITCM 起始地址为 0x00000000，容量可以配置为 1KB 或 2KB 或 4KB 或 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB，以 DK-START-GW2A18 V2.0 参考设计 gowin\_empu\_m1 为例，ITCM Size 默认 32KB。请参考 IPUG531，Gowin\_EMPU\_M1 硬件设计参考手册 ITCM Size 设定的值。

Gowin\_EMPU\_M1 的 DTCM 起始地址为 0x20000000，容量可以配置为 1KB 或 2KB 或 4KB 或 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB，

以 DK-START-GW2A18 V2.0 参考设计 gowin\_empu\_m1 为例，DTCM Size 默认 32KB。请参考 IPUG531，Gowin\_EMPU\_M1 硬件设计参考手册中 DTCM Size 设定的值。

受片内存储资源限制，ITCM 和 DTCM 的容量配置不能超过片内最大存储容量。

GW1N-9 系列,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;

GW1NR-9 系列, 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;

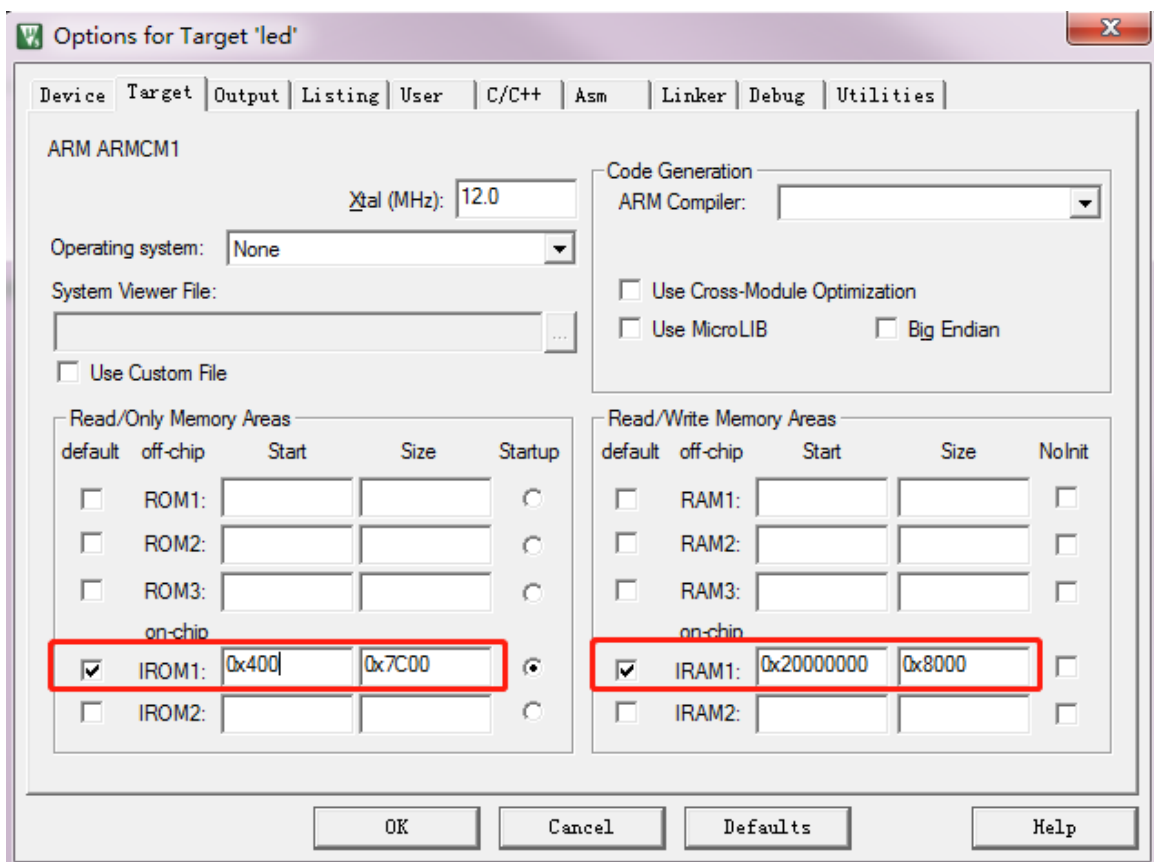
GW2A-18 系列, ITCM 或 DTCM 最大可配置为 64KB, 如果 ITCM 或 DTCM 某个存储器已配置为 64KB, 则另一个存储器最大只能配置为 16KB;

GW2AR-18 系列, 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 系列, 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。

ROM 和 RAM 的配置如图 1-3 所示, 以 DK-START-GW2A18 V2.0 参考设计 gowin\_empu\_m1 为例, 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

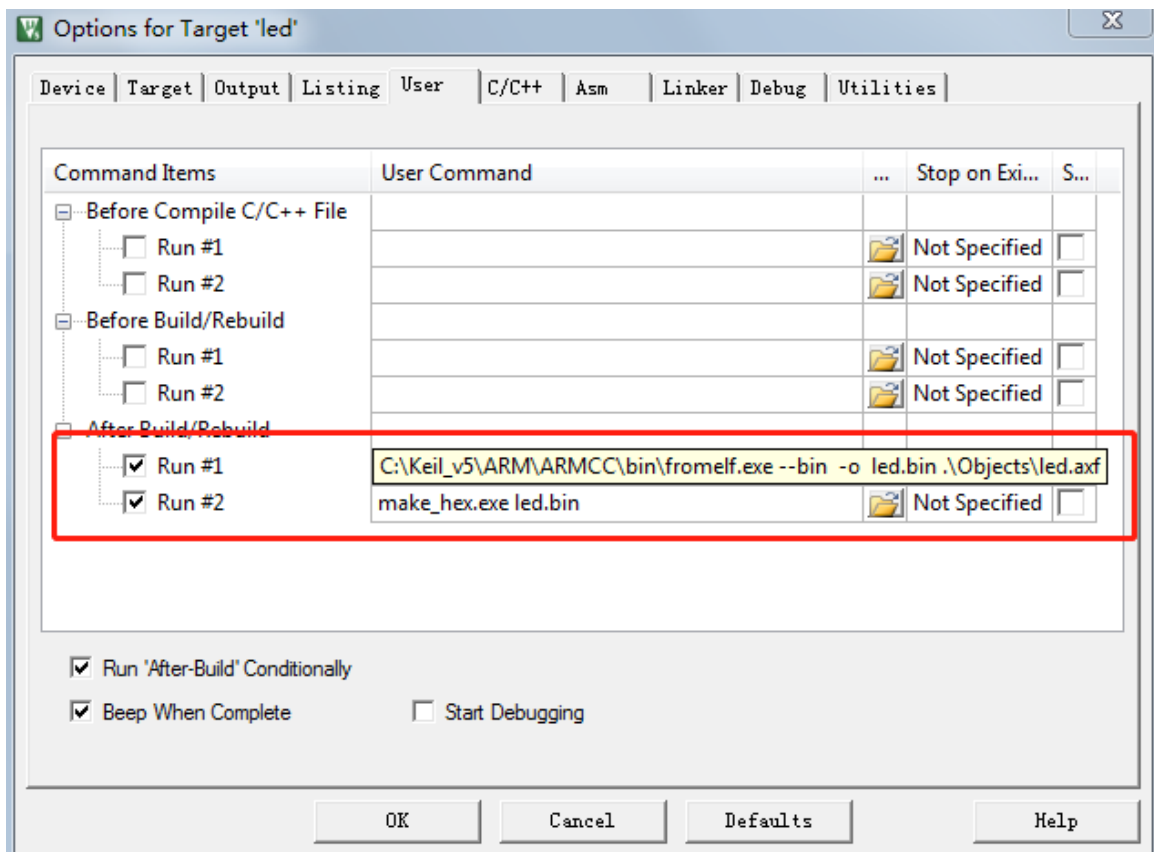




## 配置输出文件格式

- Gowin\_EMPU\_M1 软件编程设计输出文件格式使用二进制 BIN 格式, 所以需要将 axf 文件格式转换为二进制 BIN 文件格式
- 如果使用二进制 BIN 文件作为 ITCM 的初始值, 需要使用 make\_hex.exe 工具将二进制 BIN 文件转换为四个十六进制映像文件 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

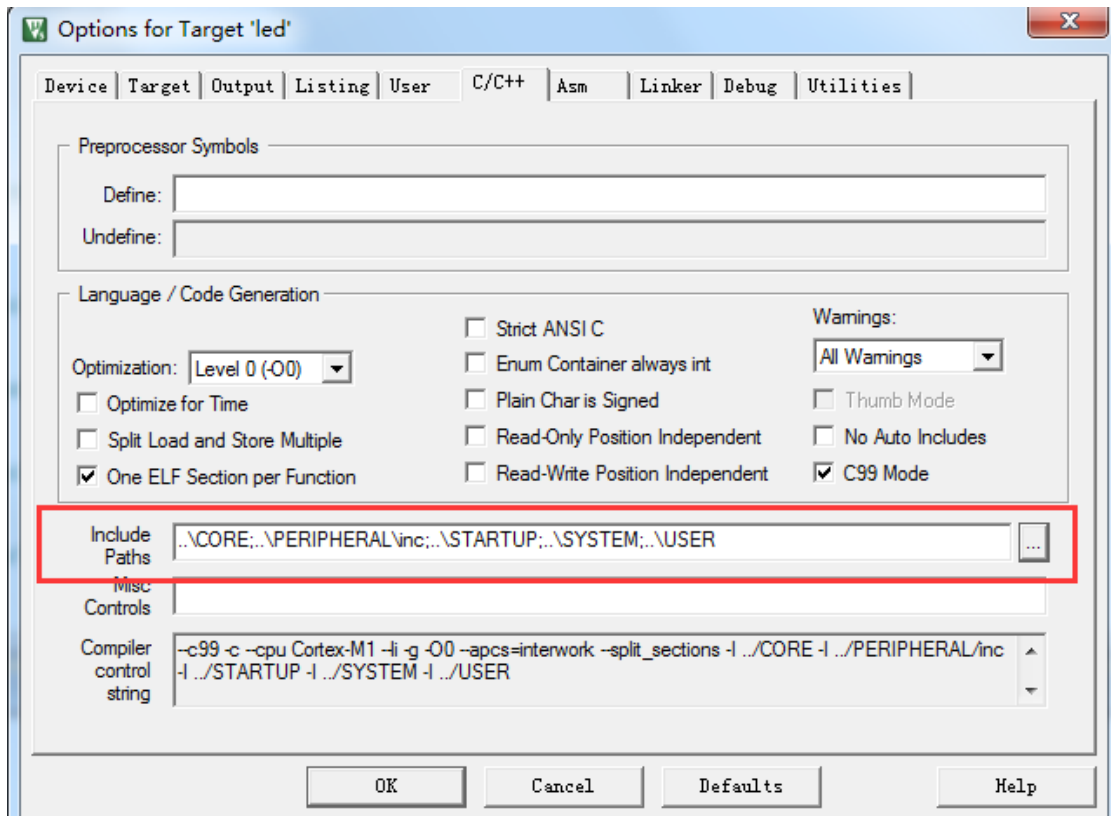
图 1-4 配置输出文件格式



## 配置头文件路径

配置头文件路径, 编译过程中用来调用头文件, 配置如图 1-5 所示。

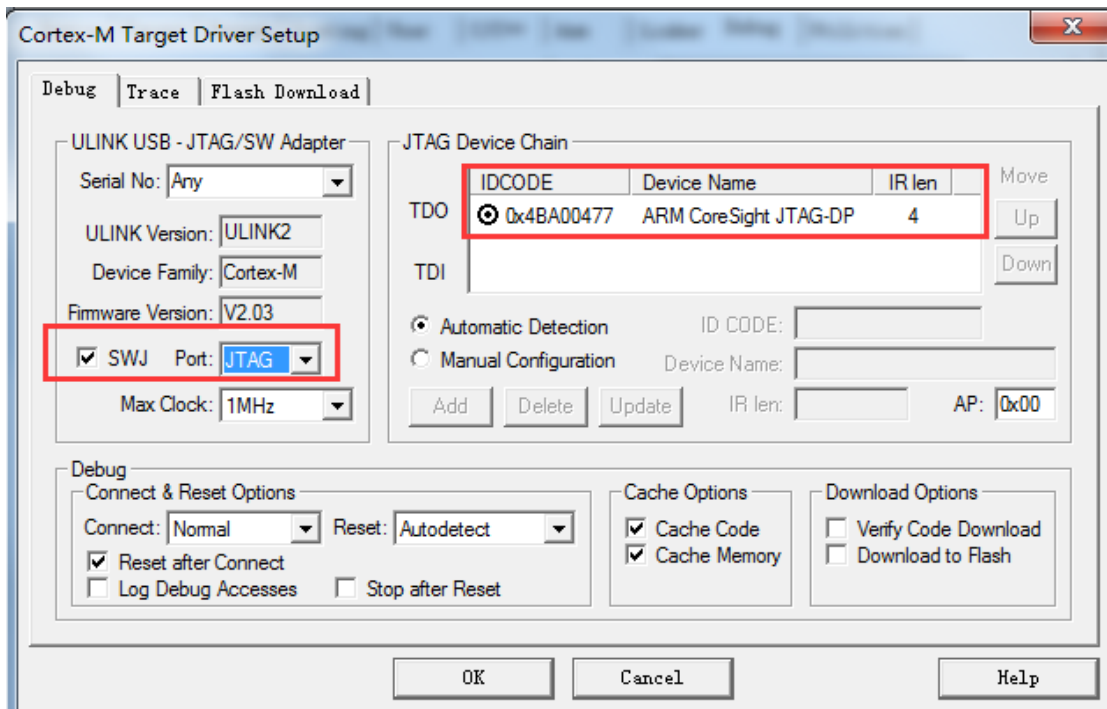
图 1-5 配置头文件路径



### 配置调试选项

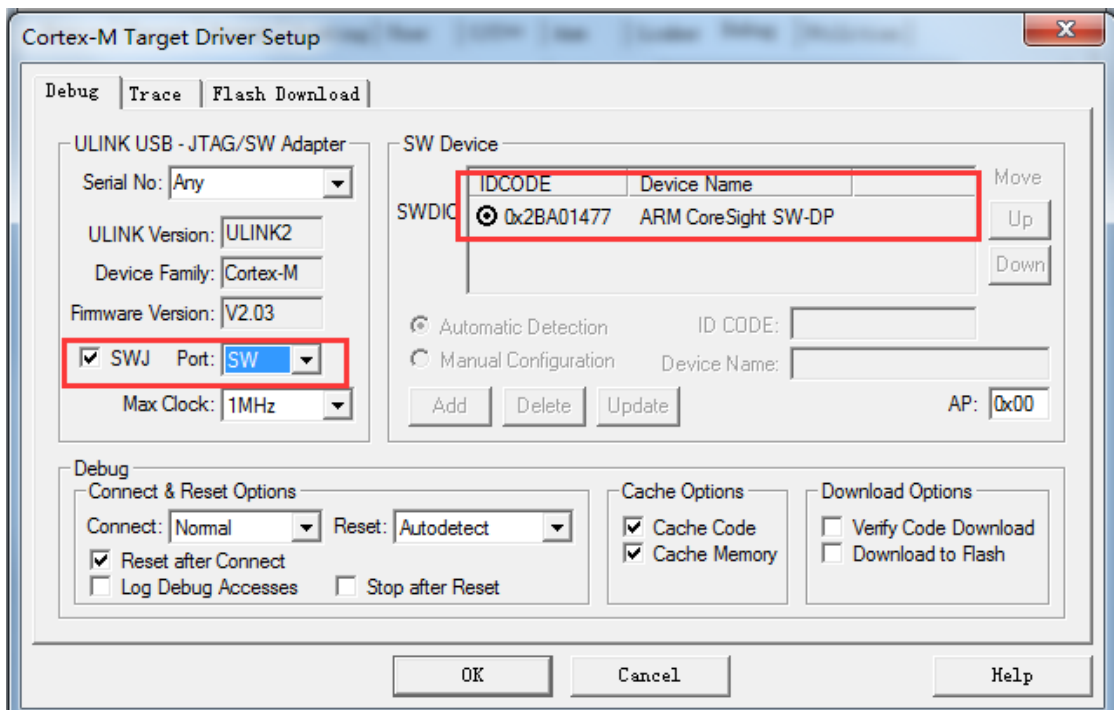
- 配置仿真器类型
  - U-LINK 仿真器  
如果选择使用 U-LINK 仿真器，则 Debug 选项配置为 ULNK2/ME Cortex Debugger。
  - J-LINK 仿真器  
如果选择使用 J-LINK 仿真器，则 Debug 选项配置为 J-LINK/J-TRACE Cortex。
- 配置调试接口类型
  - JTAG 接口  
如果配置为 JTAG 接口，如图 1-6 所示。

图 1-6 配置 JTAG 调试接口类型



- SW 调试接口  
如果配置为 SW 调试接口，如图 1-7 所示。

图 1-7 配置 SW 调试接口类型

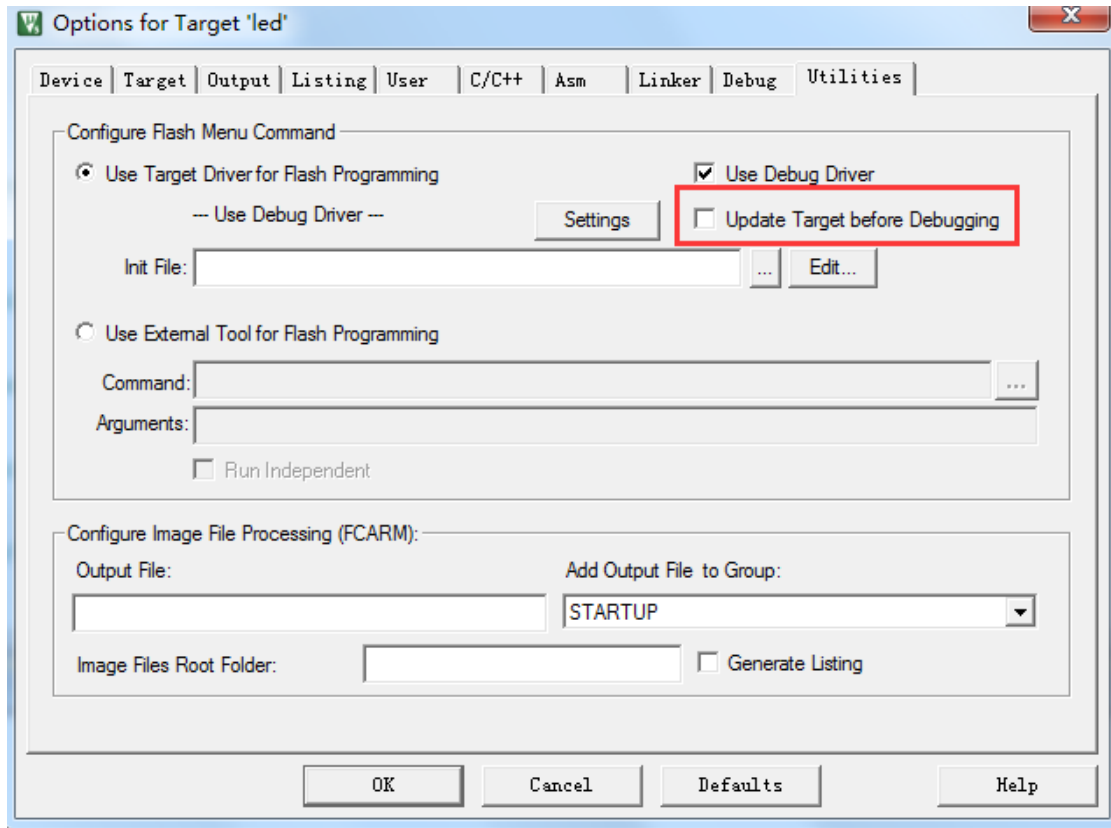


### 配置 Flash 选项

如果需要在线调试，请不要选择“Update Target before Debugging”选

项，如图 1-8 所示。

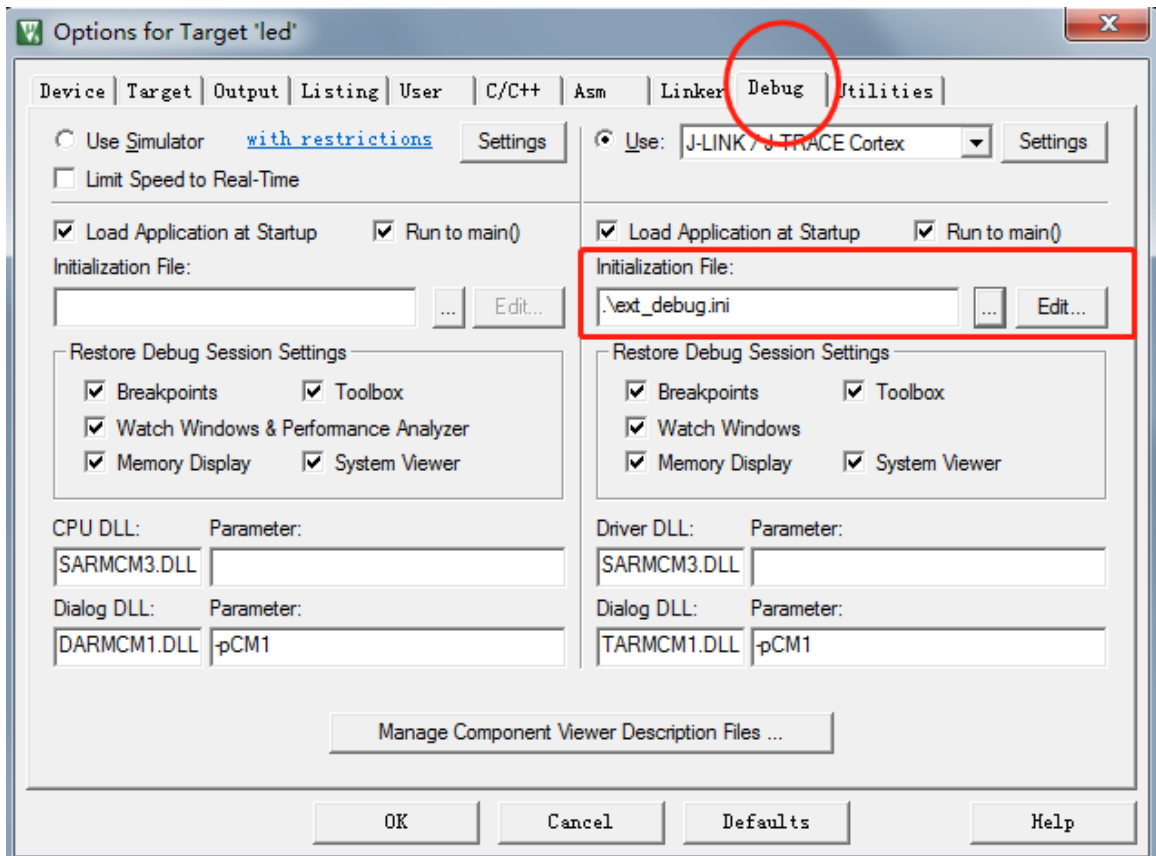
图 1-8 配置 Flash 选项



### 配置调试初始化文件

如果选择片外 SPI-Flash 下载方式，在线调试时需要加载调试初始化文件，“Initialization File” 选项中选择加载初始化文件 “ext\_debug.ini”，如图 1-9 所示。

图 1-9 配置调试初始化文件



### 1.2.3 编译



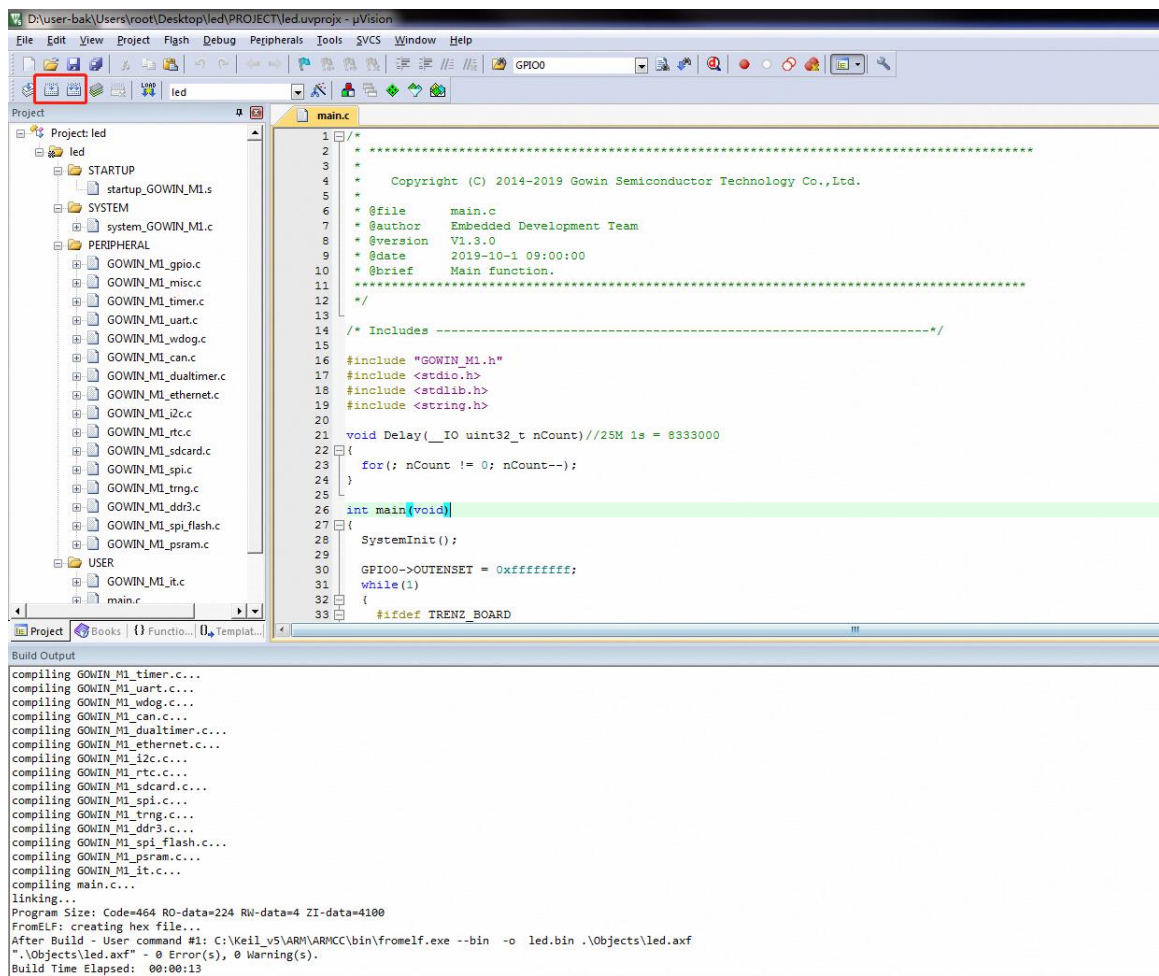
完成代码编写和工程配置后，单击工具栏 Build“”或 Rebuild“”，编译输出二进制 BIN 文件和四个十六进制映像文件 itcm0、itcm1、itcm2 和 itcm3，如图 1-10 所示。

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 软件调试

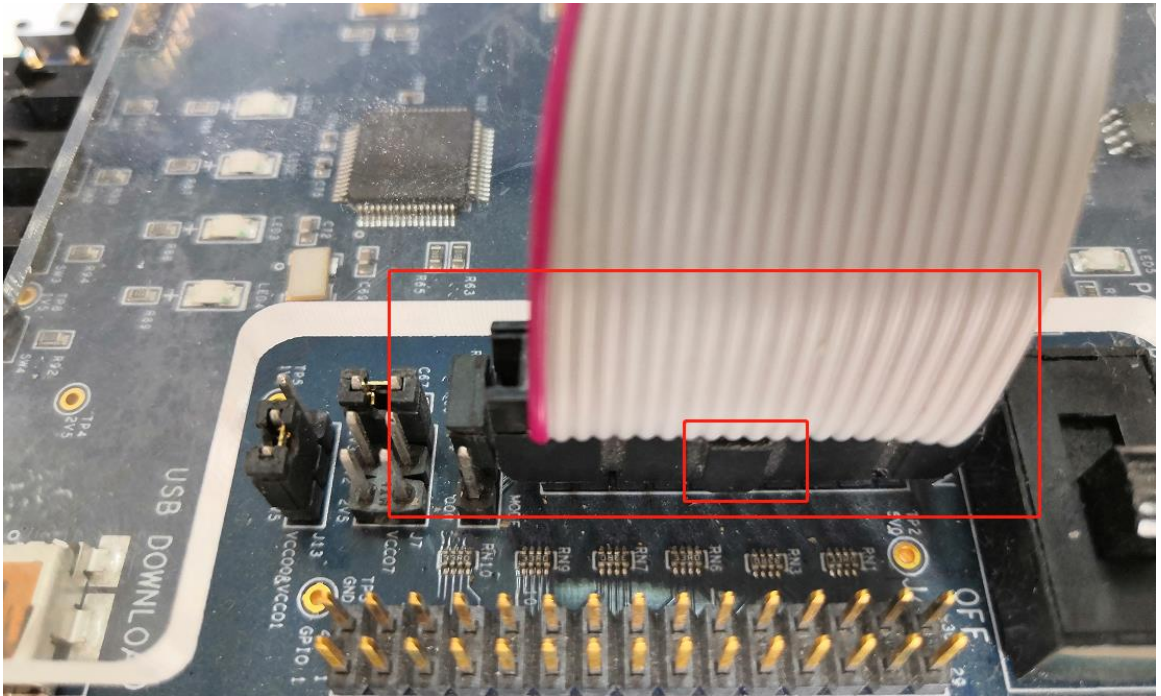
完成硬件设计输出的码流文件和软件编程设计输出的二进制 BIN 文件或四个十六进制映像文件下载后，如果用户软件设计出现问题，可以连接 U-LINK 或 J-LINK 仿真器在线调试。用户可以下载、调试软件，无需重新编译硬件设计。

### 1. 连接仿真器

按照硬件设计中约束到 FPGA IO 的 Debug Access Port 的位置，连接 J-LINK 或 U-LINK。

以 DK-START-GW2A18 V2.0 参考设计 gowin\_empu\_m1 为例，连接方式如图 1-11 所示。

图 1-11 仿真器连接



## 2. 启动调试


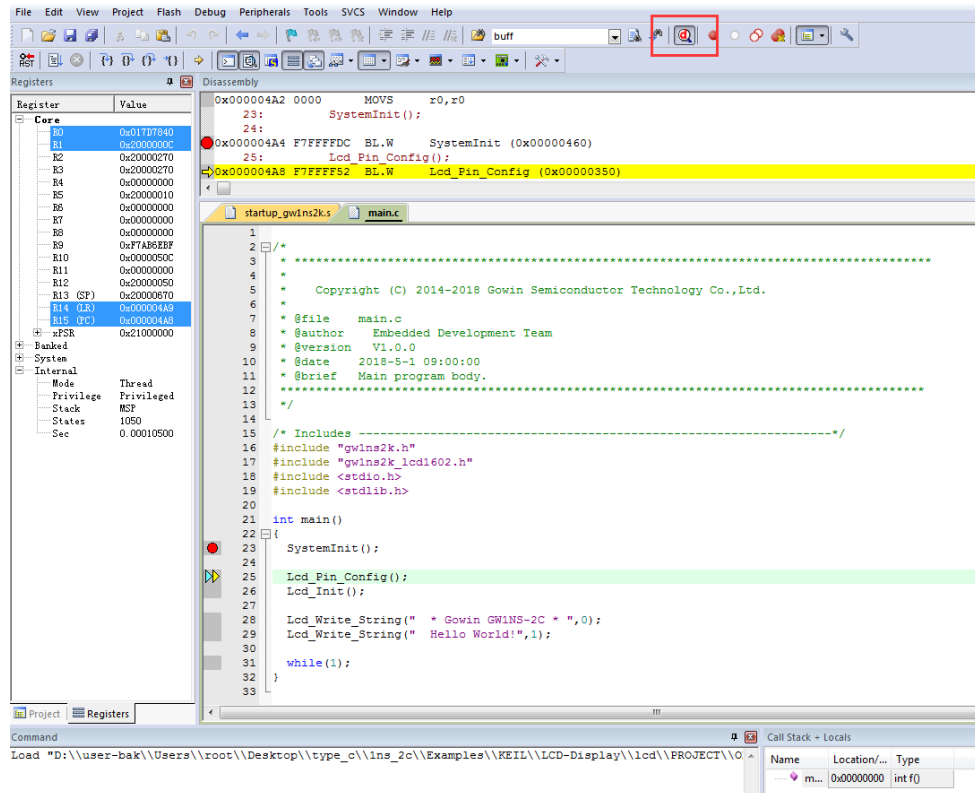
连接 U-LINK 或 J-LINK 仿真器，单击工具栏 Debug “”，进入调试状态，可以进行断点设置、单步调试、复位和运行等操作，如图 1-12 所示。

图 1-12 启动调试



## 1.3 参考设计

Gowin\_EMPU\_M1 支持 ARM Keil MDK 软件环境的参考设计:

Gowin\_EMPU\_M1\ref\_design\MCU\_RefDesign\Keil\_RefDesign



# 2 GOWIN MCU Designer

## 2.1 软件安装与配置

高云半导体官网提供 GOWIN MCU Designer 软件安装包下载  
<http://www.gowinsemi.com.cn/prodshow.aspx>。

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

## 2.2 工程模板

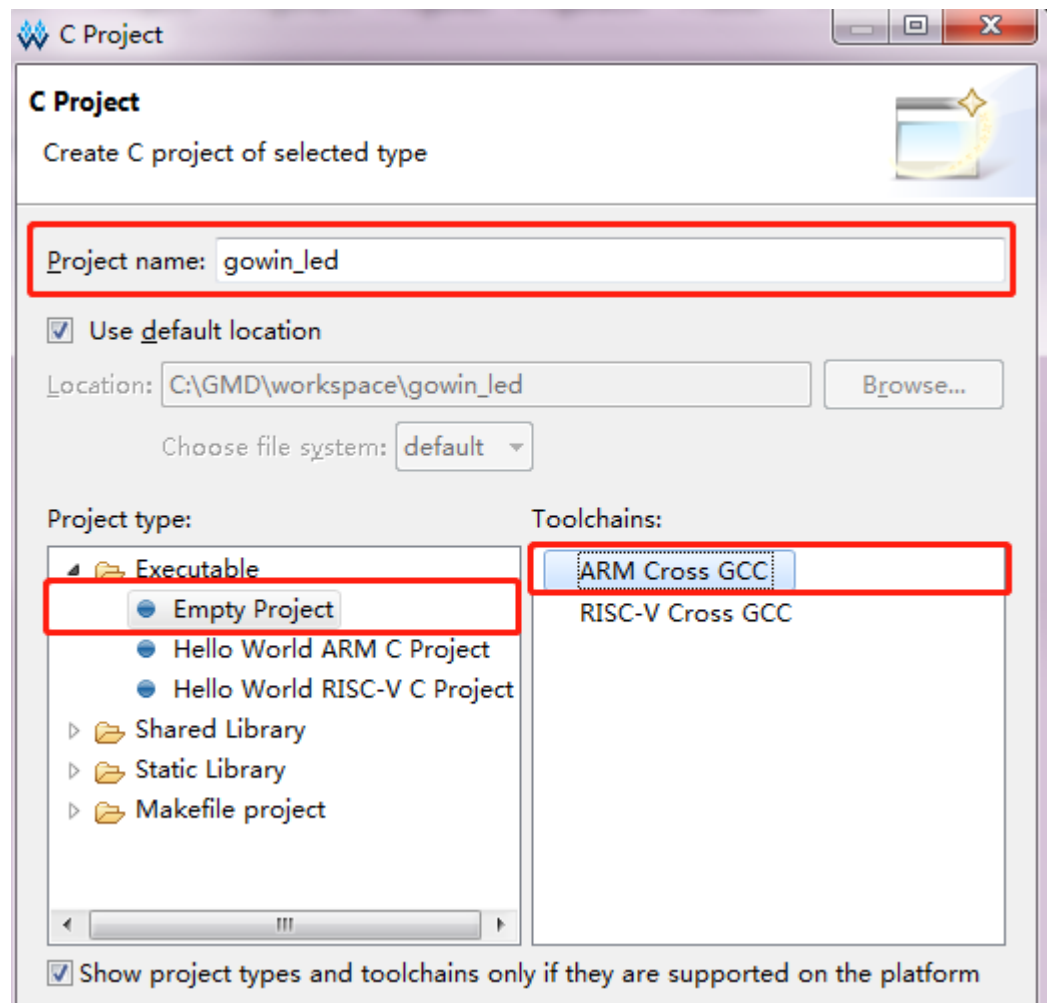
### 2.2.1 工程创建

#### 创建工程

选择菜单栏“File > New > C Project”，如图 2-1 所示。

1. 建立项目名称和位置
2. 选择项目类型 Empty Project
3. 选择工具链 ARM Cross GCC

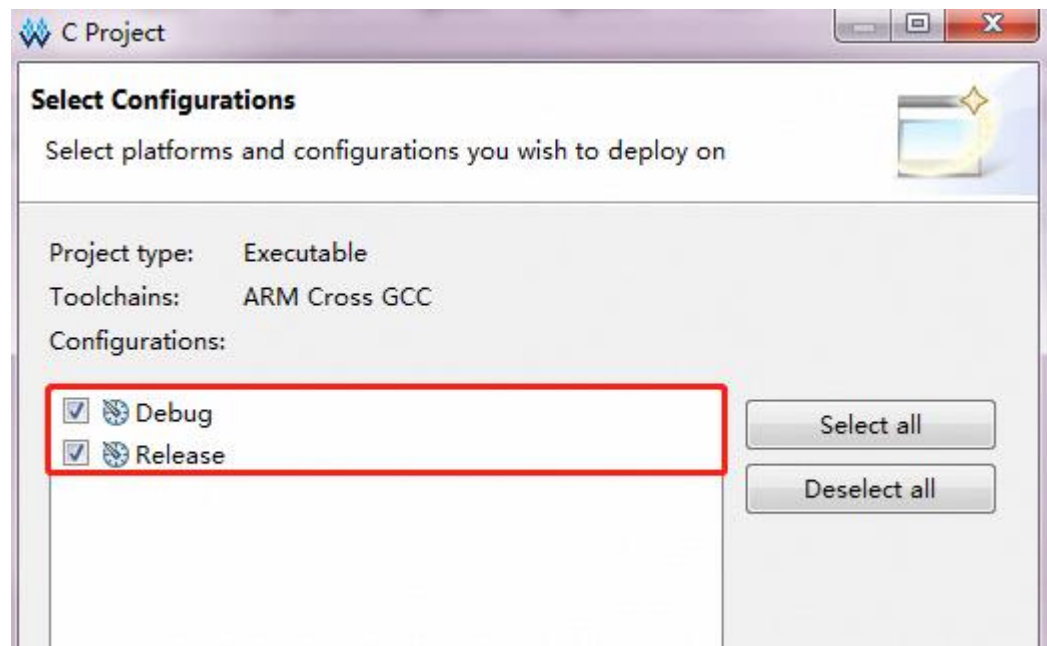
图 2-1 新建工程



### 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。

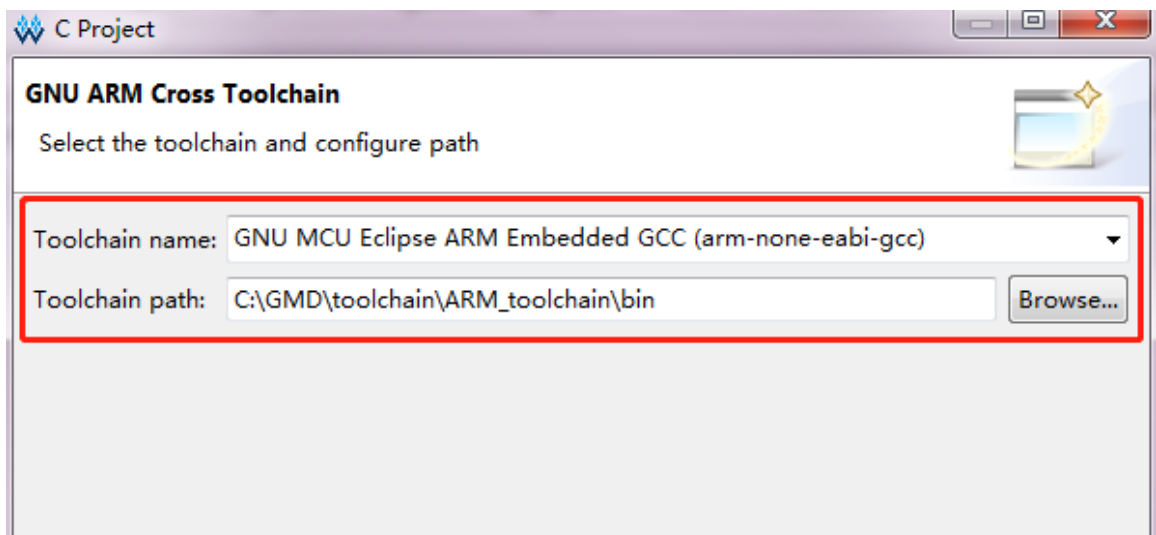
图 2-2 选择平台配置类型



### 选择编译工具链和路径

选择交叉编译工具链 `arm-none-eabi-gcc` 和其所在路径，默认配置，如图 2-3 所示。

图 2-3 选择编译工具链和路径



### 导入软件设计

完成新建工程后，选择 GOWIN MCU Designer 工作空间 `workspace` 下新建的项目工程，导入软件编程设计。

以 `GMD_RefDesign` 参考设计为例，软件编程设计目录及代码如下所示。

- CORE: ARM Cortex-M1 内核定义

- PERIPHERAL: 外设驱动库
- STARTUP: 引导启动文件
- SYSTEM: 寄存器定义、系统初始化和系统时钟定义
- USER: 用户设计
- GOWIN\_M1\_flash.Id: Flash 链接脚本

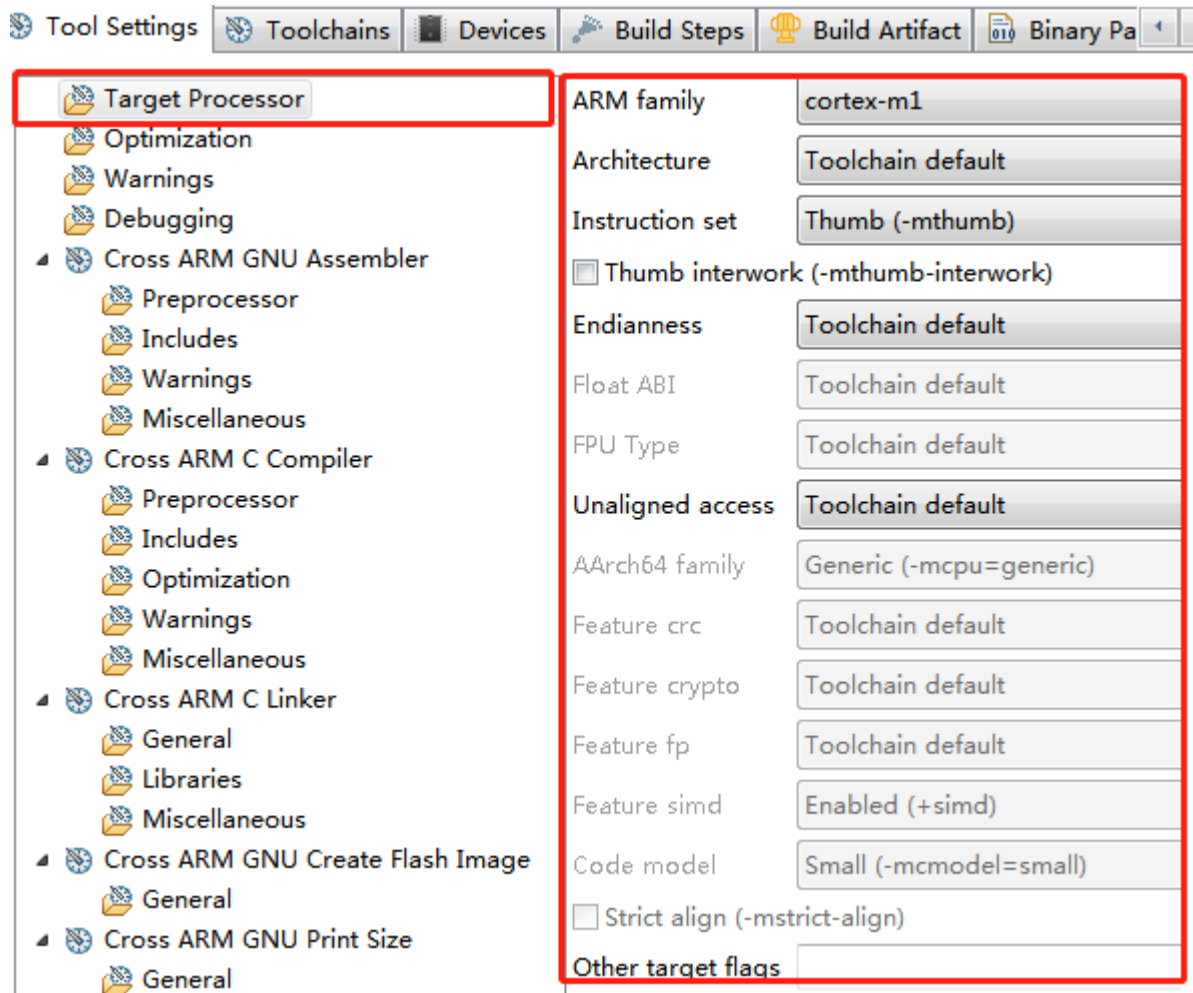
## 2.2.2 工程配置

GOWIN MCU Designer 中，选择当前工程，右键选择“Properties > C/C++ Build > Setting”，配置工程参数选项。

### 配置 Target Processor

选择“Target Processor”，配置“ARM family”为“cortex-m1”，如图 2-4 所示。

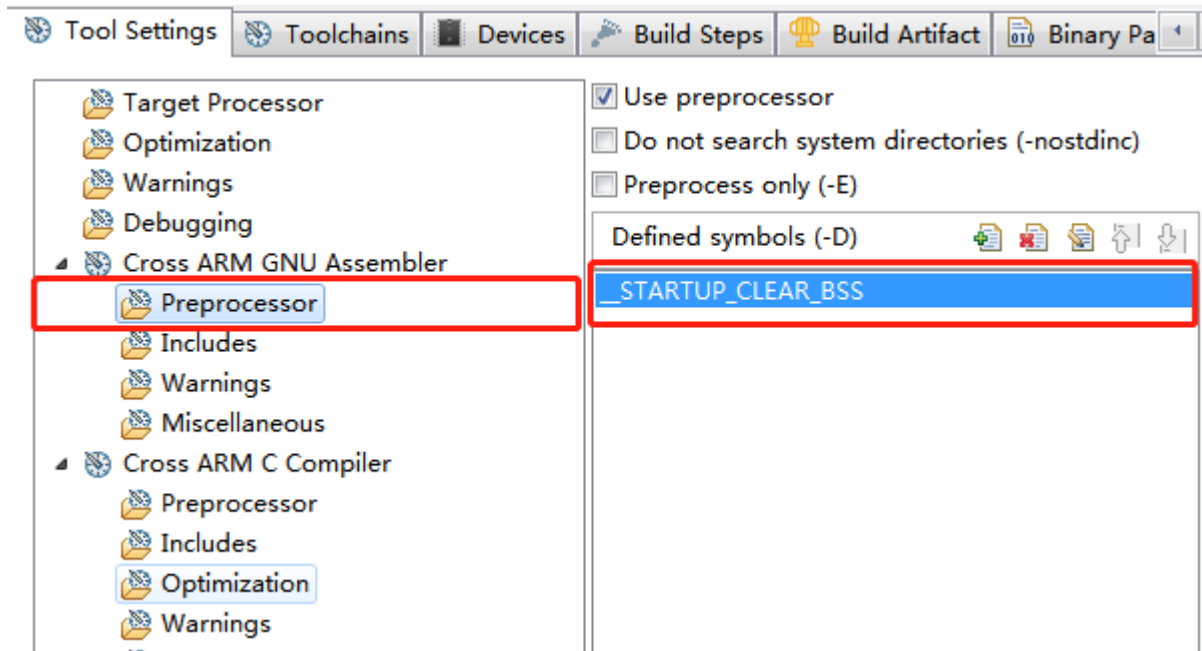
图 2-4 配置 Target Processor



### 配置 Cross ARM GNU Assembler Preprocessor

选择“Cross ARM GNU Assembler”下的“Preprocessor”选项，配置汇编宏定义“\_\_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

选择“Cross ARM C Compiler”下的“Includes”选项，配置 C 头文件路径，如图 2-6 所示。

以 GMD\_RefDesign 参考设计为例，头文件路径配置如下所示。

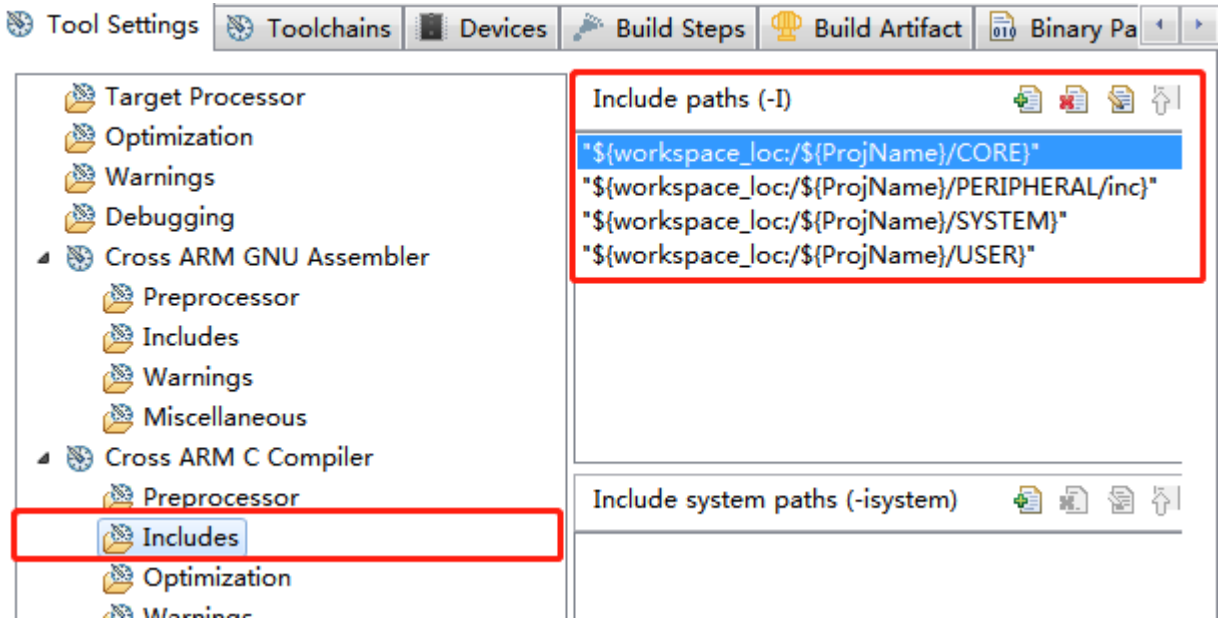
```
"${workspace_loc:${ProjName}/CORE}"
```

```
"${workspace_loc:${ProjName}/PERIPHERAL/inc}"
```

```
"${workspace_loc:${ProjName}/SYSTEM}"
```

```
"${workspace_loc:${ProjName}/USER}"
```

图 2-6 配置 Cross ARM C Compiler Includes



### 配置 Cross ARM C Linker

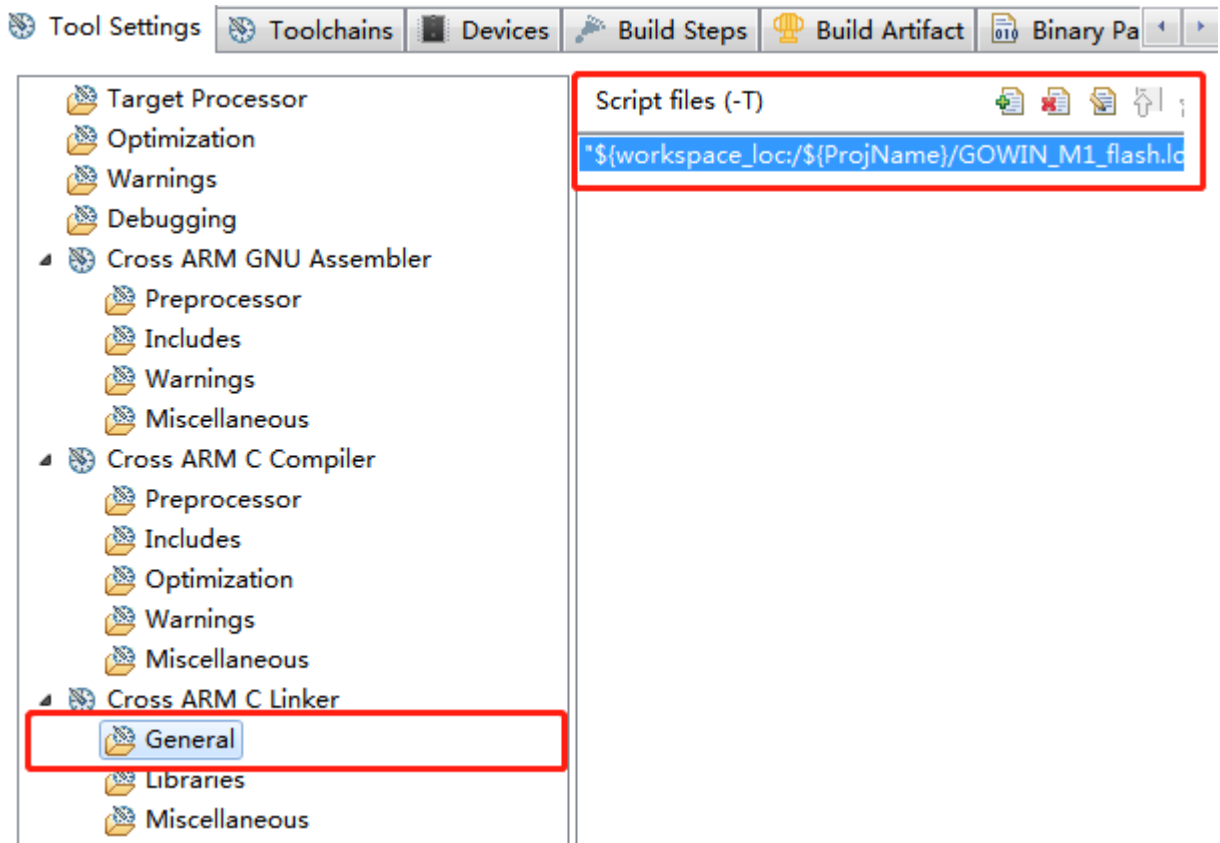
选择“Cross ARM C Linker”下的“General”选项，配置 Flash 链接脚本“GOWIN\_M1\_flash.Id”，如图 2-7 所示。

以 GMD\_RefDesign 参考设计为例，Flash 链接脚本如下所示。

```
"${workspace_loc}/${ProjName}/GOWIN_M1_flash.Id"
```

如果 Gowin\_EMPU\_M1 软件编程设计使用片外 SPI-Flash 下载启动方式，请在 GOWIN\_M1\_flash.Id 中修改 Flash 起始地址“FLASH ORIGIN”为 0x00000400。

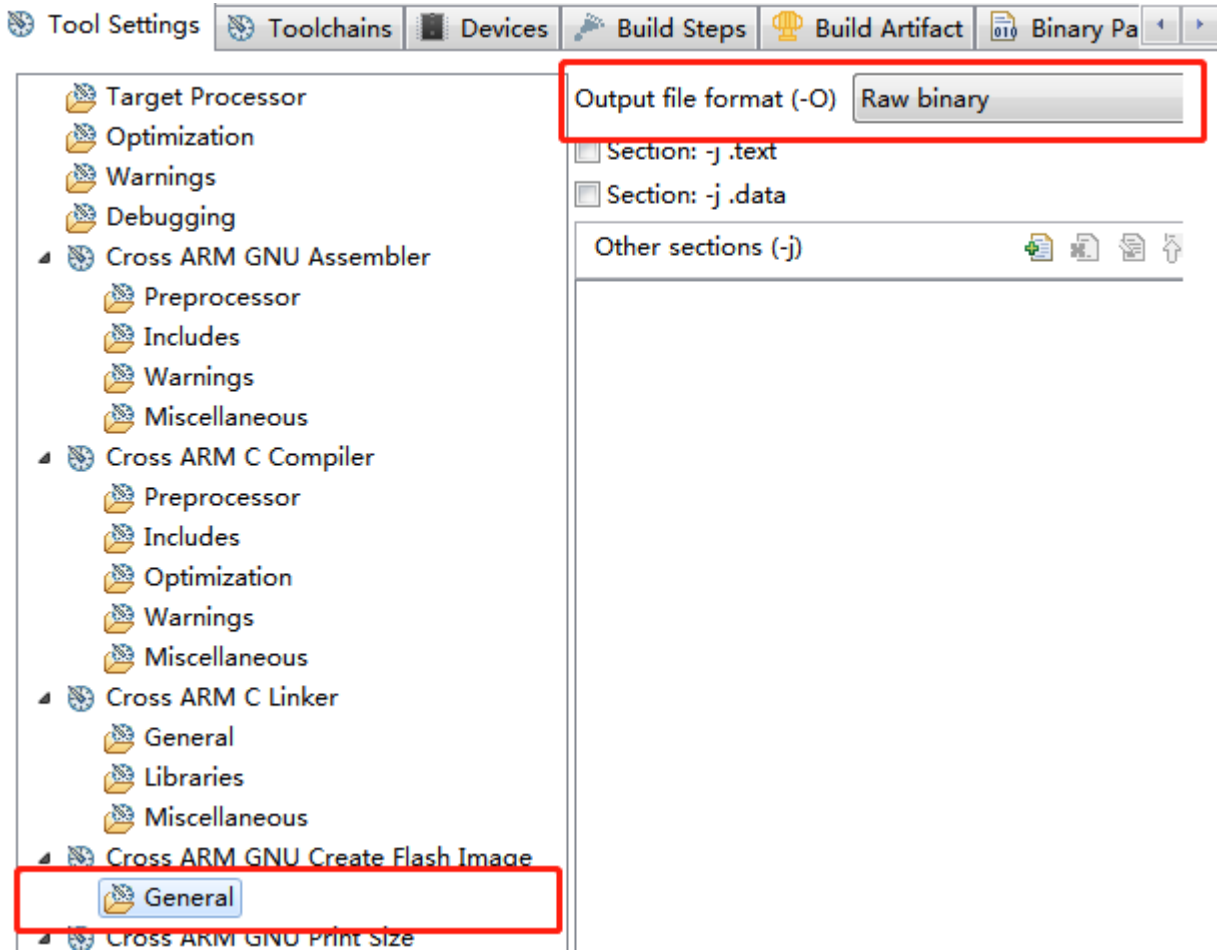
图 2-7 配置 Cross ARM C Linker



### 配置 Cross ARM GNU Create Flash Image

选择“Cross ARM GNU Create Flash Image”中的“General”选项，配置 Gowin\_EMPU\_M1 输出文件格式为“Raw binary”，如图 2-8 所示。

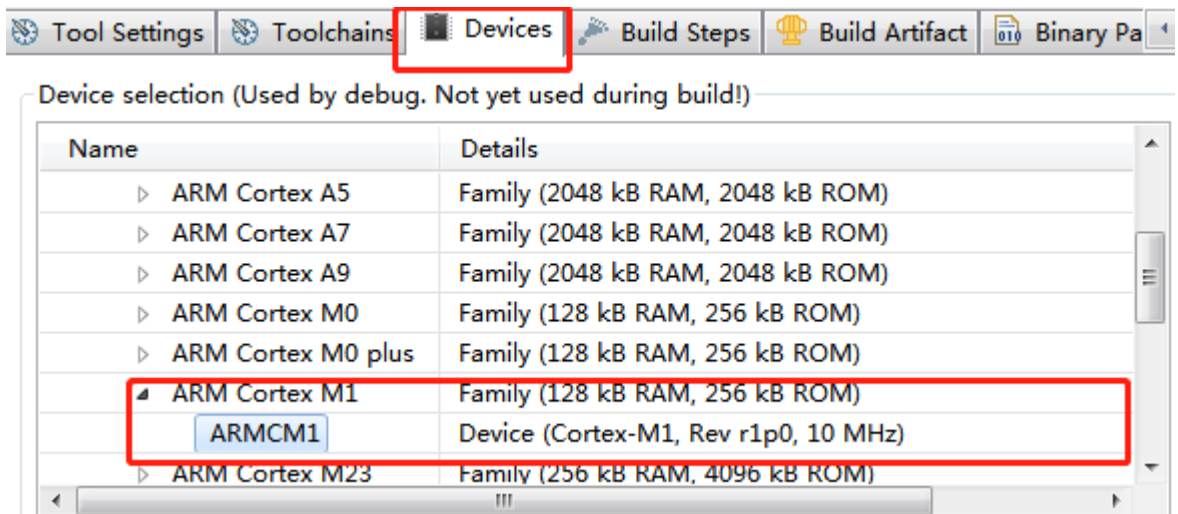
图 2-8 配置 Cross ARM GNU Create Flash Image



### 配置 Devices

选择“Devices”选项卡，选择器件“ARM Cortex M1 ARMCM1”，如图 2-9 所示。

图 2-9 配置 Devices

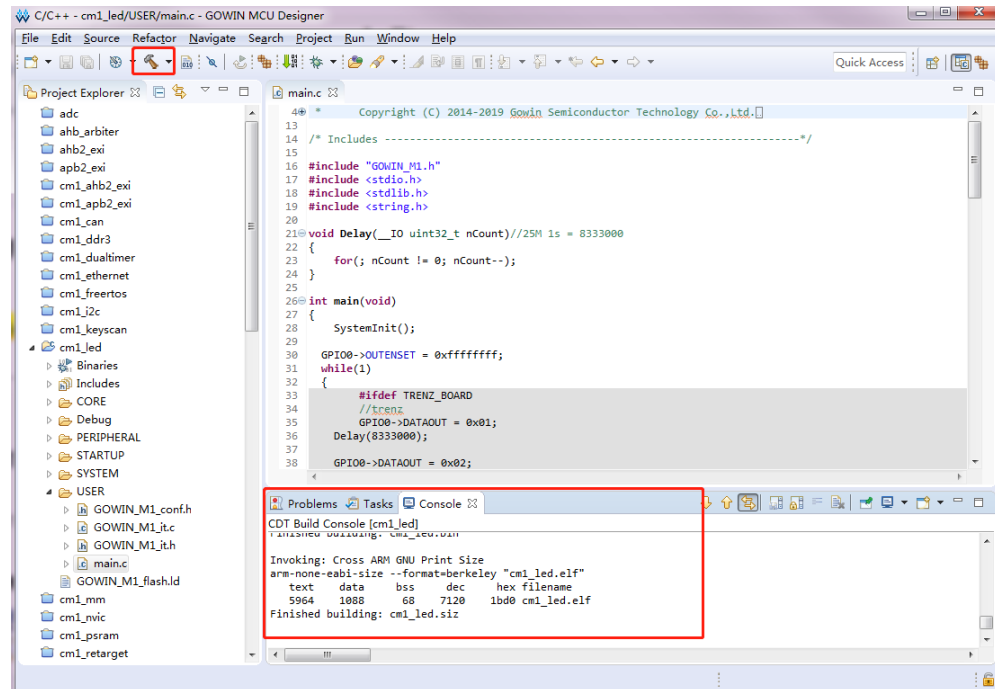




## 2.2.3 编译

完成工程配置和代码编写后，编译工程，单击工具栏编译按钮“”，编译生成 Gowin\_EMPU\_M1 二进制 BIN 文件，如图 2-11 所示。

图 2-10 编译



## 2.2.4 下载

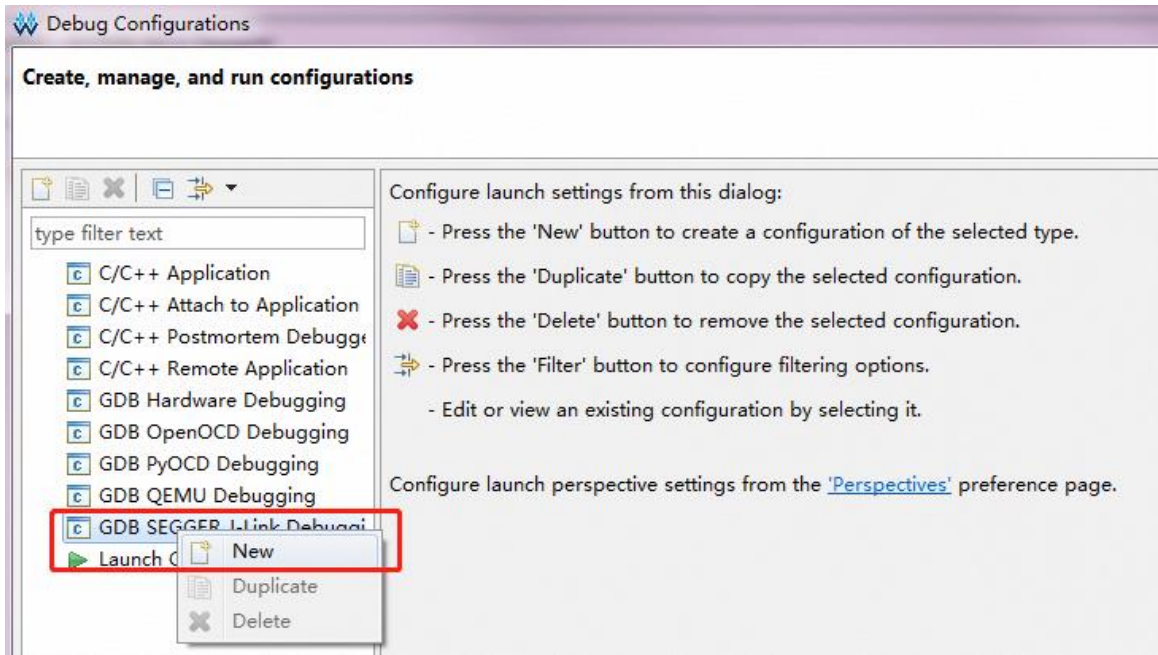
完成 Gowin\_EMPU\_M1 软件编程设计编译后，软件编程设计下载方法请参考：[IPUG532](#), Gowin\_EMPU\_M1 下载参考手册

## 2.2.5 调试

### 配置调试选项

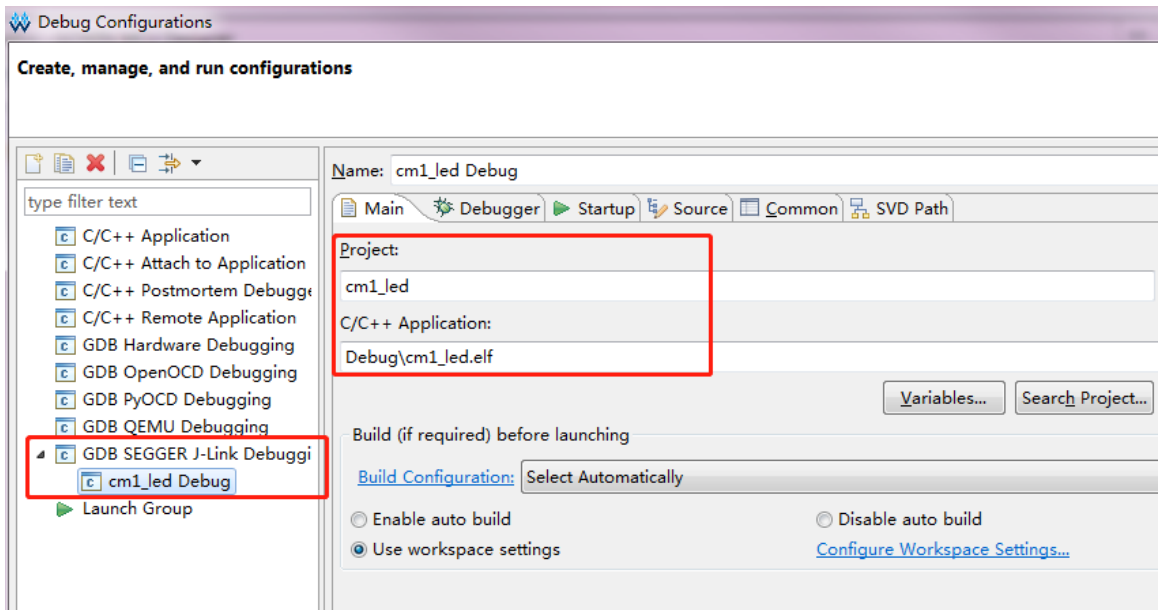
1. 选择菜单栏“Run > Debug Configurations > GDB SEGGER J-Link Debugging”，右键选择“New”，建立工程的调试配置选项，如图 2-11 所示。

图 2-11 建立调试配置选项



2. 选择“Main”选项卡，配置输出文件路径，如图 2-12 所示。

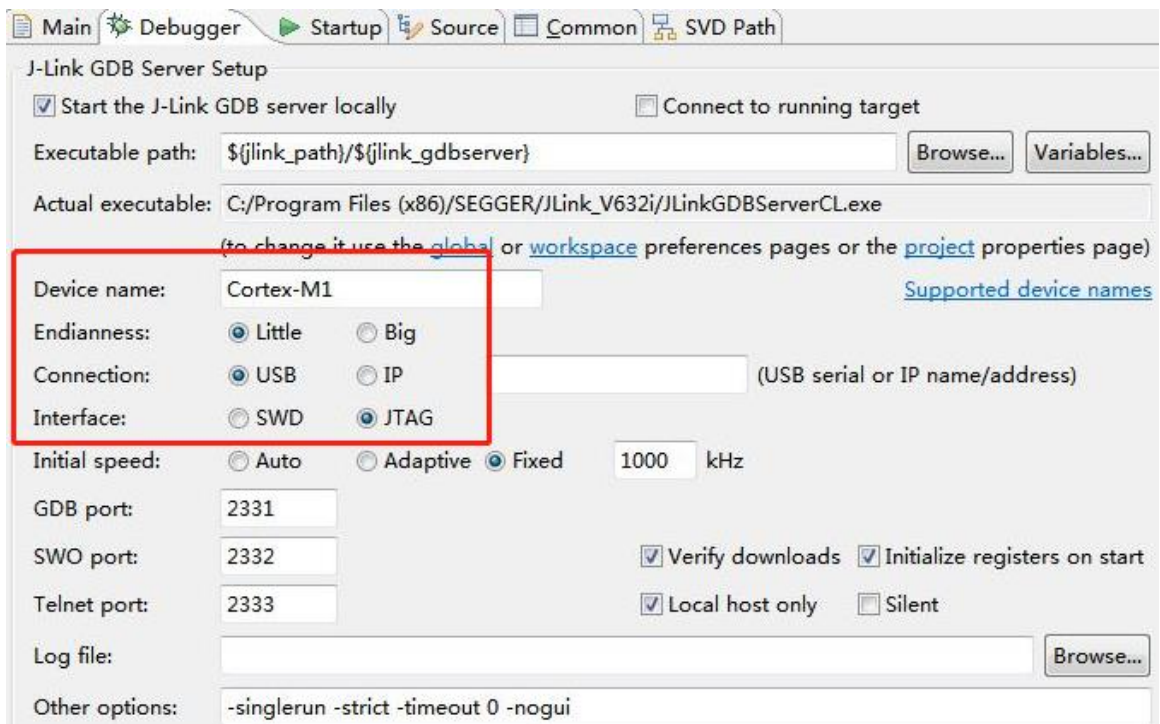
图 2-12 配置 Image 路径



3. 选择“Debugger”选项卡，配置 Debugger 器件名称为“Cortex-M1”，调试接口为“JTAG”或“Serial Wire”，如图 2-13 所示。

- Device Name, Cortex-M1
- Interface, JTAG 或 SWD
- Endianness, Little
- Connection, USB

图 2-13 配置 Debugger



## 调试


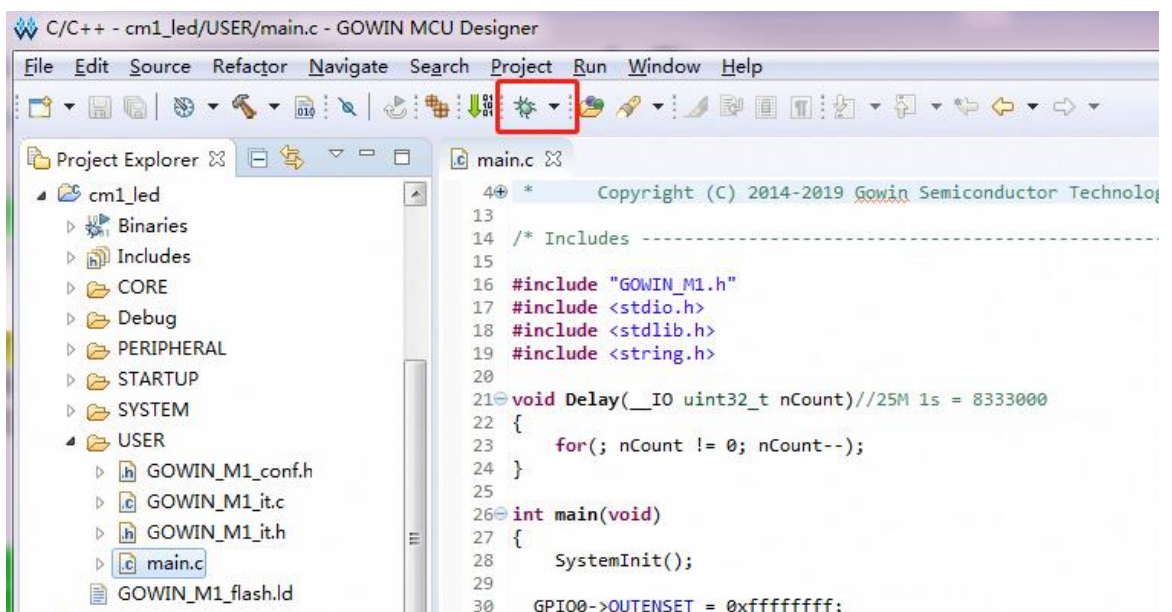
连接 J-LINK 仿真器与开发板,单击工具栏 Debug 按钮下拉列表“”,选择当前项目的 Debug 配置,单击进入调试状态,可以进行断点设置、单步调试、复位和运行等操作,如图 2-14 所示。

图 2-14 启动调试



## 2.3 参考设计

Gowin\_EMPU\_M1 支持 GOWIN MCU Designer 软件环境的参考设计：  
Gowin\_MCU\_M1\ref\_design\MCU\_RefDesign\GMD\_RefDesign

