



GW1NS-2C MCU IDE

软件参考手册

IPUG519-1.2.1,2019-08-06

版权所有©2019 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
2018/08/21	1.0	初始版本。
2018/11/21	1.1	<ul style="list-style-type: none">● 优化 Eclipse 安装过程● 增加工程实例操作
2019/04/12	1.2	<ul style="list-style-type: none">● 更新软件编程库● 更新软件编程参考设计
2019/08/06	1.2.1	修复已知 SPI 和 ADC 问题。

目录

目录	i
图目录	ii
1 ARM Keil MDK 软件	1
1.1 软件安装	1
1.2 工程模板	1
1.2.1 配置选项	1
1.2.2 工程配置	2
1.2.3 工程编译	6
1.2.4 工程下载	7
1.2.5 工程调试	7
1.3 参考设计	9
2 GOWIN MCU Designer	10
2.1 软件安装	10
2.2 工程模板	10
2.2.1 工程创建	10
2.2.2 工程配置	14
2.2.3 工程编译	21
2.2.4 工程下载	21
2.2.5 工程调试	22
2.3 参考设计	26

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.....	3
图 1-5 配置宏定义和头文件路径.....	4
图 1-6 配置下载选项	4
图 1-7 配置调试选项	5
图 1-8 调试接口类型	5
图 1-9 工程编译	6
图 1-10 MCU 下载	7
图 1-11 JTAG 模式切换命令	8
图 1-12 启动调试	9
图 2-1 图新建工程.....	11
图 2-2 选择平台类型配置.....	12
图 2-3 选择工具链和路径.....	13
图 2-4 工程结构	13
图 2-5 选择 Properties	14
图 2-6 配置 Cross ARM GNU Assembler Preprocessor.....	15
图 2-7 配置 Cross ARM GNU Assembler Includes.....	16
图 2-8 配置 Cross ARM C Compiler Preprocessor	17
图 2-9 配置 Cross ARM C Compiler Includes.....	18
图 2-10 配置 Cross ARM C Linker.....	19
图 2-11 配置 Cross ARM GNU Create Flash Image.....	20
图 2-12 配置 Devices.....	21
图 2-13 编译工程	21
图 2-14 工程下载	22
图 2-15 建立调试配置选项.....	23
图 2-16 配置 Image 路径	23

图 2-17 配置 Debugger.....	24
图 2-18 JTAG 接口切换.....	25
图 2-19 启动调试	25

1 ARM Keil MDK 软件

1.1 软件安装

请参考 ARM Keil MDK 官网提供的《[MDK Getting Started](#)》和《[uVision User's Guide](#)》。

1.2 工程模板

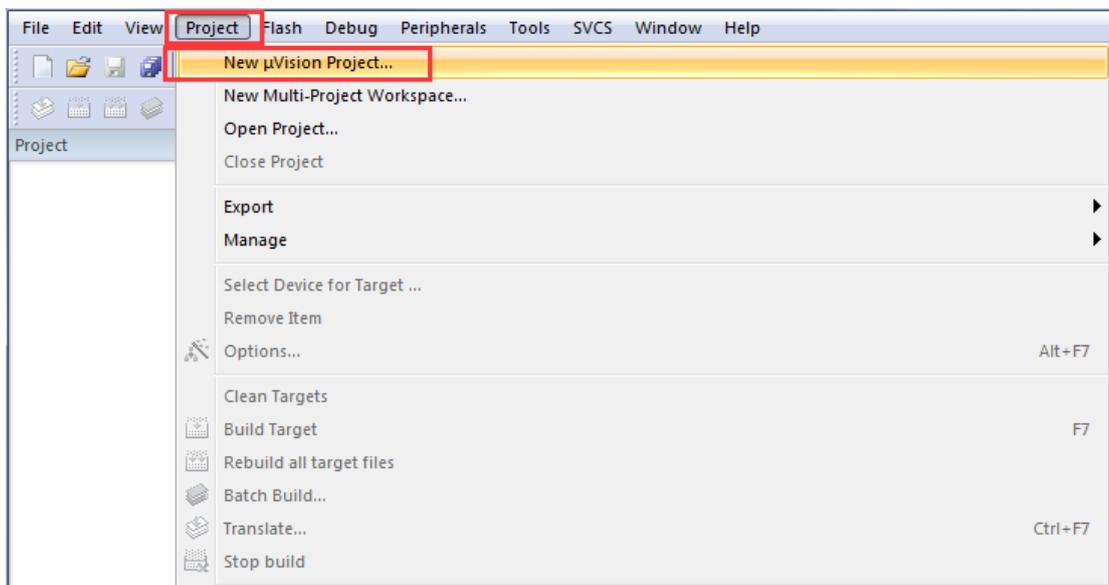
使用 ARM Keil MDK 软件进行 Gowin_EMPU 软件编程设计，需要创建工程、配置工程选项、编码、编译、下载和调试。

1.2.1 配置选项

工程创建

打开 ARM Keil MDK 软件，选择菜单栏 Project 中 New uVision Project...，创建工程，如图 1-1 所示。

图 1-1 创建工程

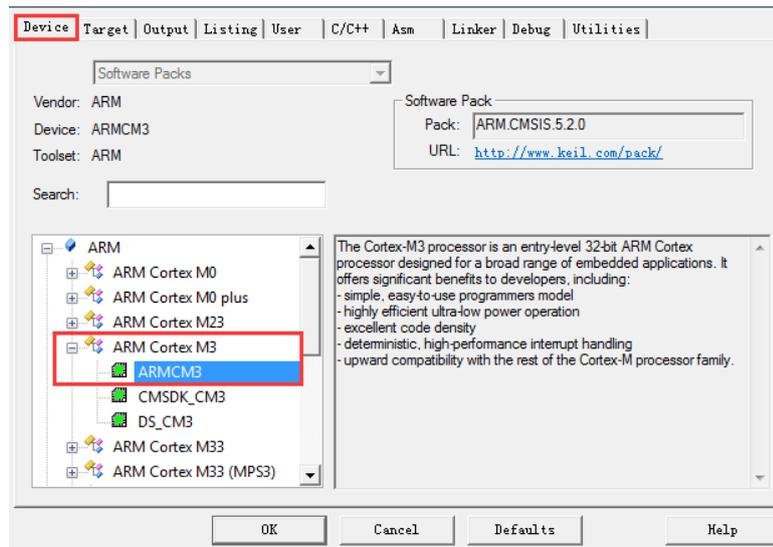


1.2.2 工程配置

配置器件

Gowin_EMPU for GW1NS-2C 内置 ARM Cortex-M3 内核，所以器件选择 ARM Cortex-M3 的“ARMCM3”，如图 1-2 所示。

图 1-2 配置器件



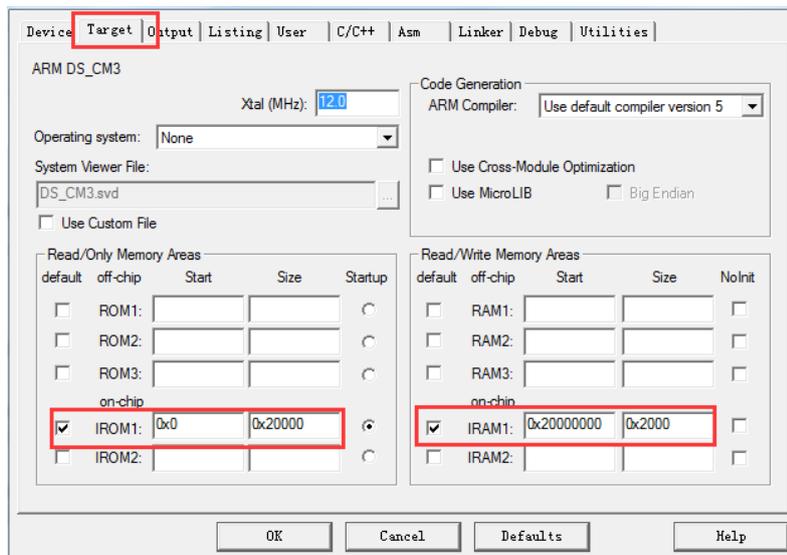
配置 ROM 和 RAM

配置 ROM 和 RAM 的起始地址和容量大小。

GW1NS-2C ROM 起始地址为 0x00000000，容量为 128K Byte。

GW1NS-2C RAM 起始地址为 0x20000000，容量为 2KB、4KB 或 8KB，如图 1-3 所示。

图 1-3 配置 ROM 和 RAM

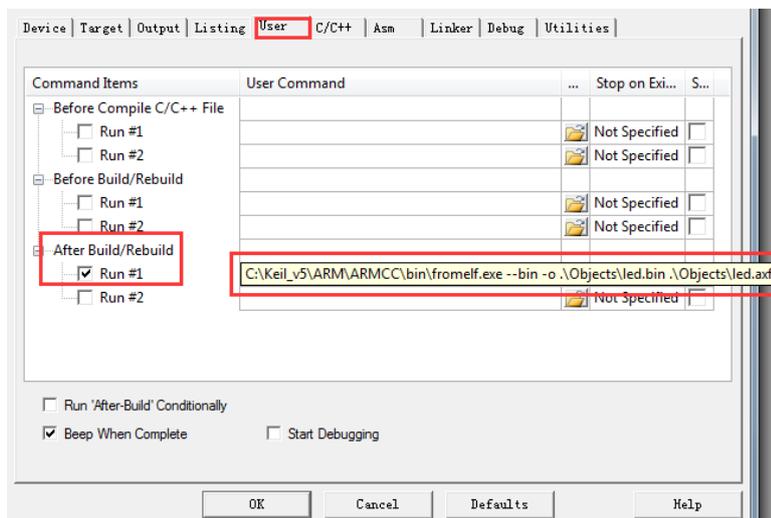


配置输出文件格式

下载工具 Programmer 支持 BIN 二进制文件下载格式，所以配置输出文件格式为 BIN 格式。

User 命令行选项中转换*.axf 文件为*.bin 文件，如图 1-4 所示。

图 1-4 配置输出文件格式



命令格式为：

- Run #1

- fromelf.exe --bin -o bin-file axf-file

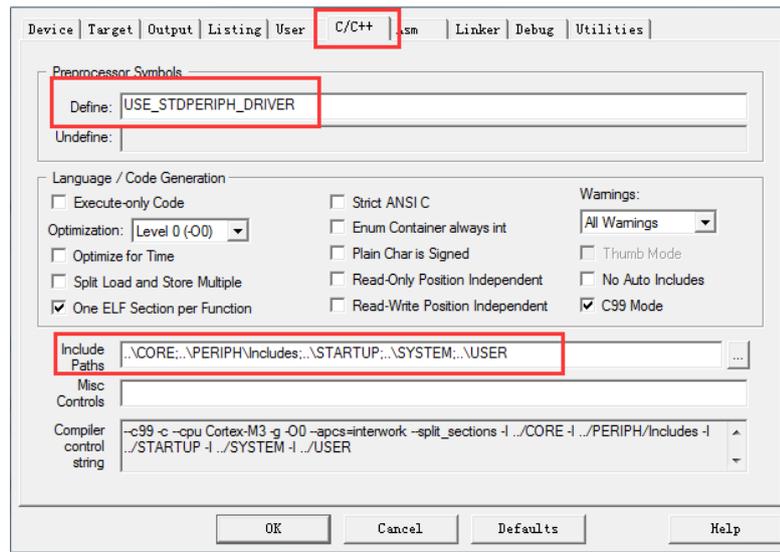
配置宏定义和头文件路径

配置宏定义“USE_STDPERIPH_DRIVER”，用来调用标准外设。

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。

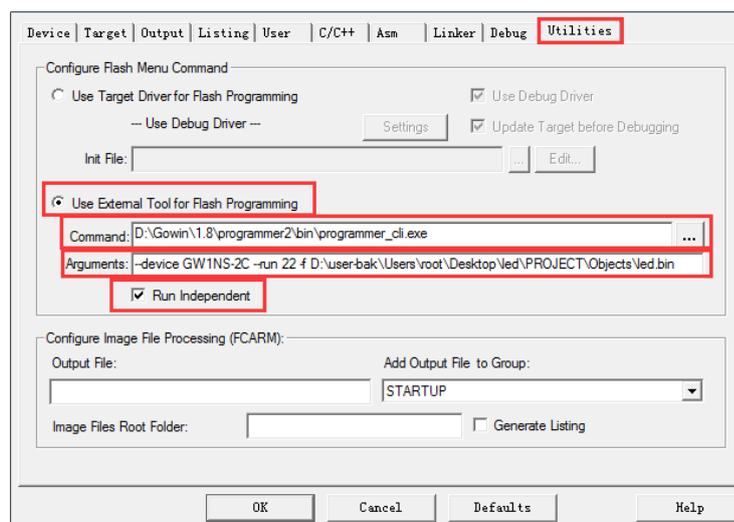
图 1-5 配置宏定义和头文件路径



配置下载工具

如果使用 Keil 软件内部下载方法，则配置 Flash 下载工具为外部工具 programmer_cli，如图 1-6 所示。

图 1-6 配置下载选项



Command 为 Programmer 路径。

Arguments 为 Programmer 命令选项，如--device GW1NS-2C --run 22

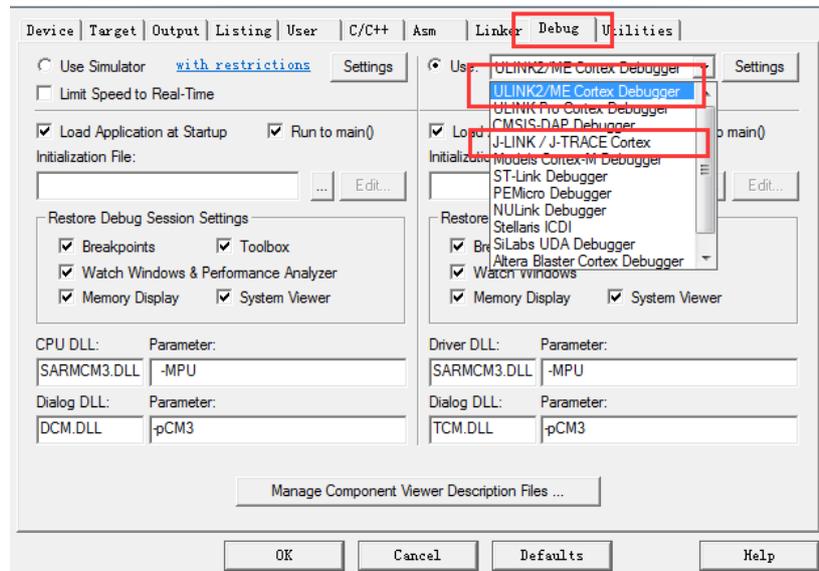
-f bin-file。

配置调试选项

如图 1-7 所示：

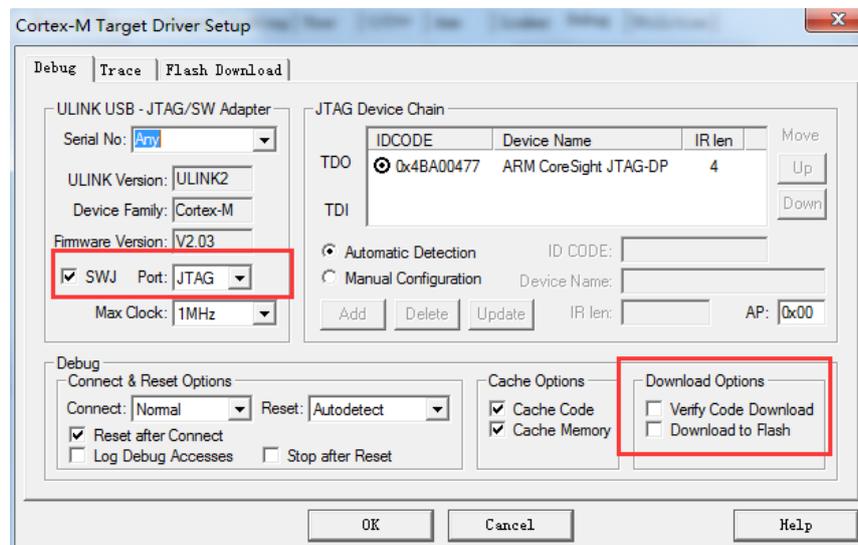
- 如果选择使用 U-LINK 仿真器，则 Debug 选项配置为“ULNK2/ME Cortex Debugger”。
- 如果选择使用 J-LINK 仿真器，则 Debug 选项配置为“J-LINK/J-TRACE Cortex”。

图 1-7 配置调试选项



调试接口类型配置为 JTAG，如图 1-8 所示。

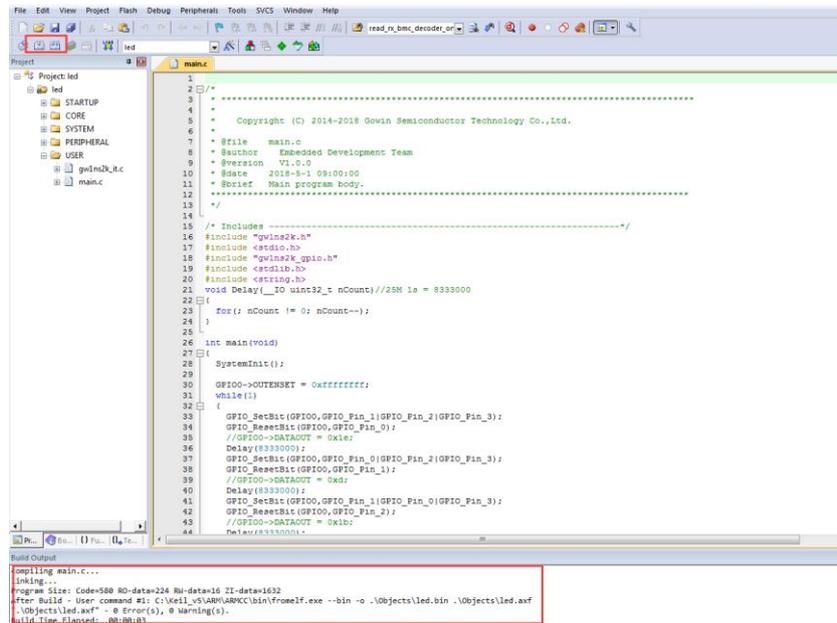
图 1-8 调试接口类型



1.2.3 工程编译

完成编码和工程配置后，编译生成 Gowin_EMPU 二进制 BIN 文件，如图 1-9 所示。

图 1-9 工程编译

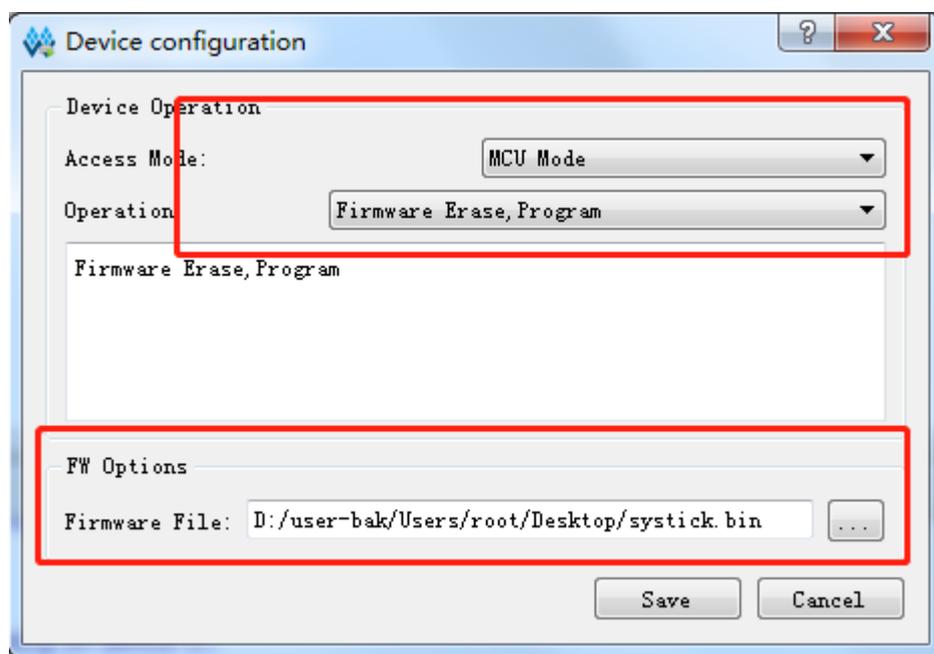


1.2.4 工程下载

完成工程配置和编译后，可以使用两种下载方式：

- 第一种方法：点击 ARM Keil MDK 工具栏下载按钮，调用上述已经配置的外部命令行工具 programmer_cli，下载 Gowin_EMPU 二进制 BIN 文件。
- 第二种方法：打开 Programmer 工具，配置器件访问模式为 MCU Mode，操作选项选择 Firmware Erase, Program，Firmware File 导入 Gowin_EMPU 二进制 BIN 文件，进行下载，如图 1-10 所示。

图 1-10 MCU 下载



1.2.5 工程调试

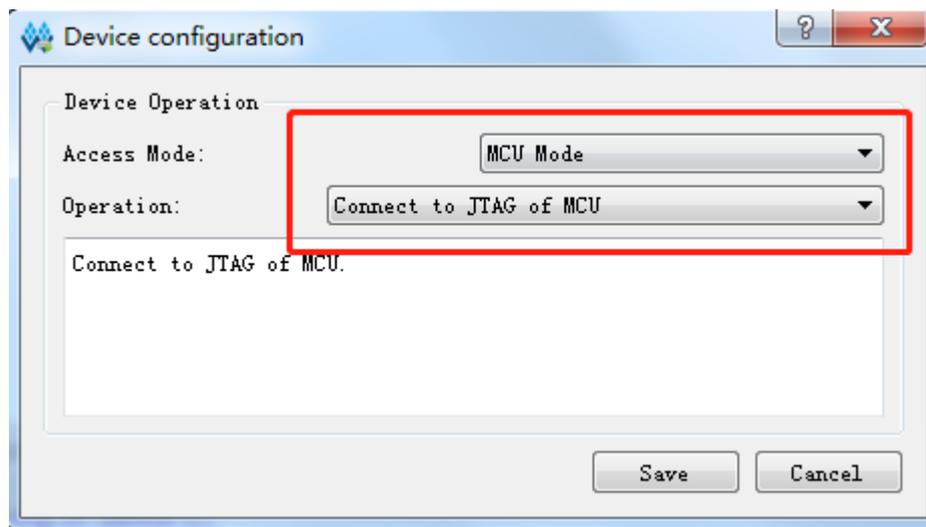
完成 Gowin_EMPU 二进制 BIN 文件下载后，如果用户设计出现问题，可以连接 U-LINK 或 J-LINK 仿真器使用调试功能。

JTAG 模式切换

如图 1-11 所示，使用 Programmer 下载软件将 JTAG 模式由下载模式切换到调试模式。

打开 Programmer，器件访问配置为 MCU Mode，操作模式配置为 Connect to JTAG of MCU，完成 JTAG 模式切换。

图 1-11 JTAG 模式切换命令



JTAG 接口切换

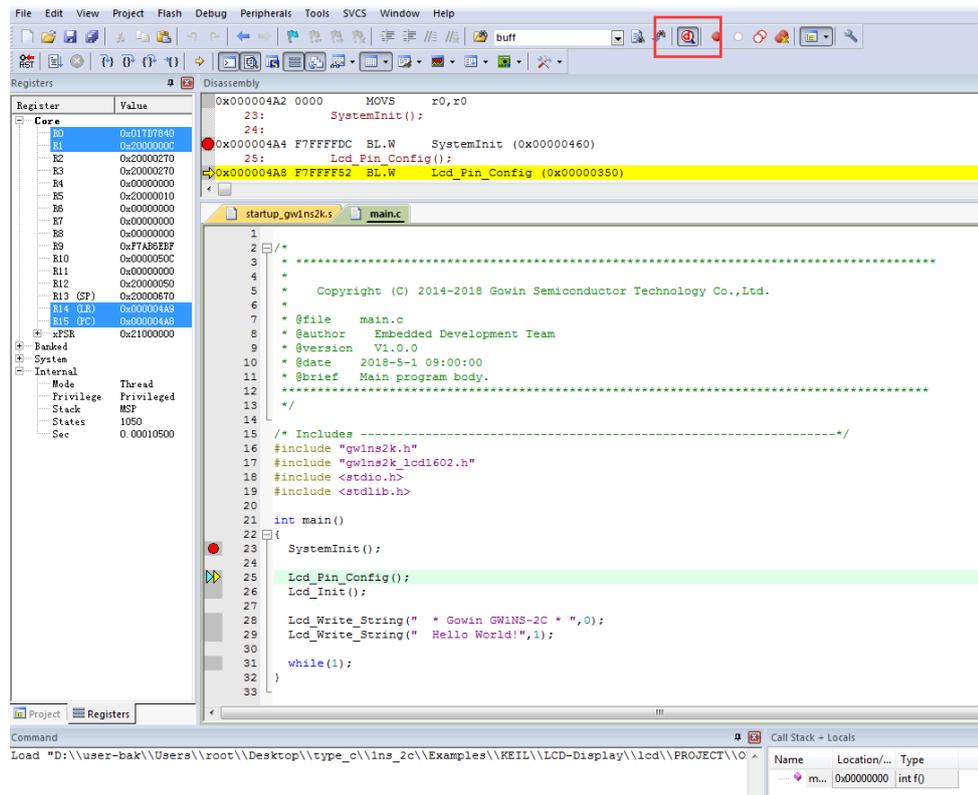
如果使用开发板“DK-EVAL-GW1NS2 V1.1”，需要手动将 JTAG 接口 TMS、TCK、TDI 和 TDO 跳线帽由 FDTI 下载切换到 ARM 下载，即由(1,3)端口切换到(3,4)端口。

如果使用开发板“DK-START-GW1NS2 V1.1”，需要手动将 JTAG 接口 TMS、TCK、TDI 和 TDO 拨码开关由 FPGA 下载切换到 ARM 下载。

启动调试

连接 U-LINK 或 J-LINK 仿真器，选择工具栏 Debug 按钮，开始调试，如图 1-12 所示。

图 1-12 启动调试



1.3 参考设计

Gowin_EMPU for GW1NS-2C 支持 ARM Keil MDK 软件环境的参考设计:

Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign

2 GOWIN MCU Designer

2.1 软件安装

高云半导体官网提供 GOWIN MCU Designer 软件安装包下载。

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

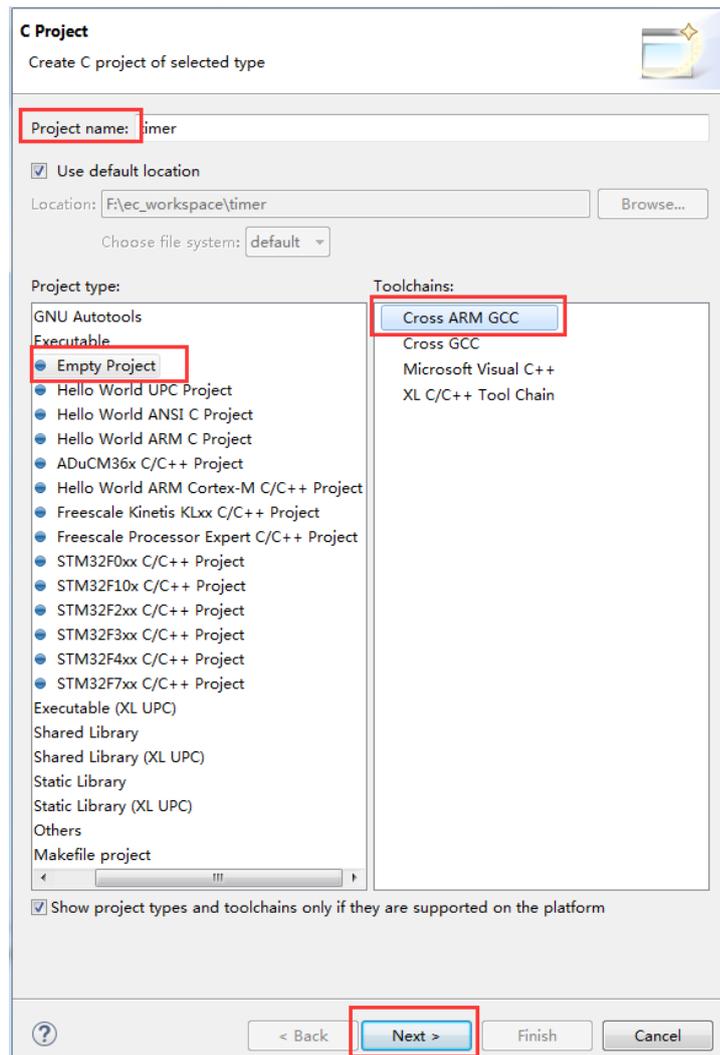
2.2.1 工程创建

新建工程

选择菜单栏 File 中的 New 下的 C Project, 如图 2-1 所示。

- 建立项目名称
- 选择项目类型 Empty Project
- 选择工具链 Cross ARM GCC

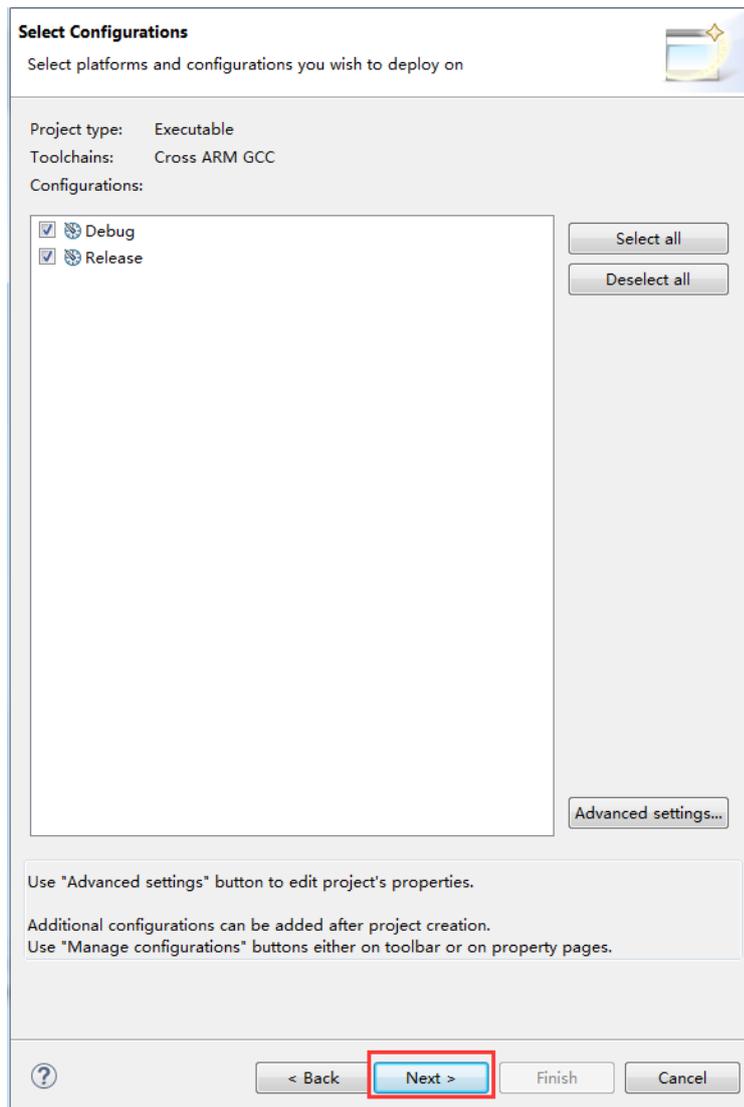
图 2-1 图新建工程



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。

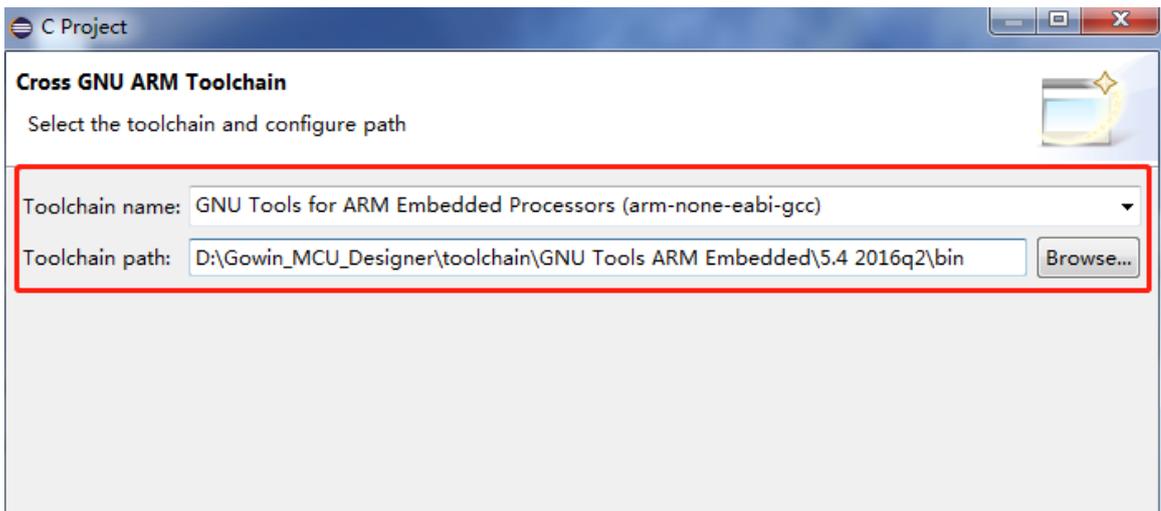
图 2-2 选择平台类型配置



选择工具链和路径

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

图 2-3 选择工具链和路径

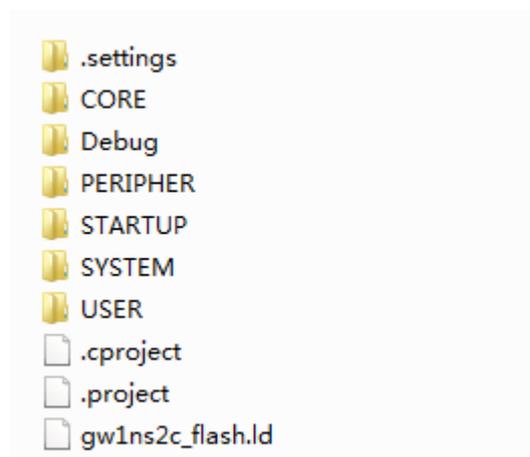


建立工程结构

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，添加工程结构和代码，如图 2-4 所示。

- CORE: ARM Cortex-M3 内核定义
- PERIPHER: 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
- gw1ns2c_flash.ld: Flash 链接脚本

图 2-4 工程结构



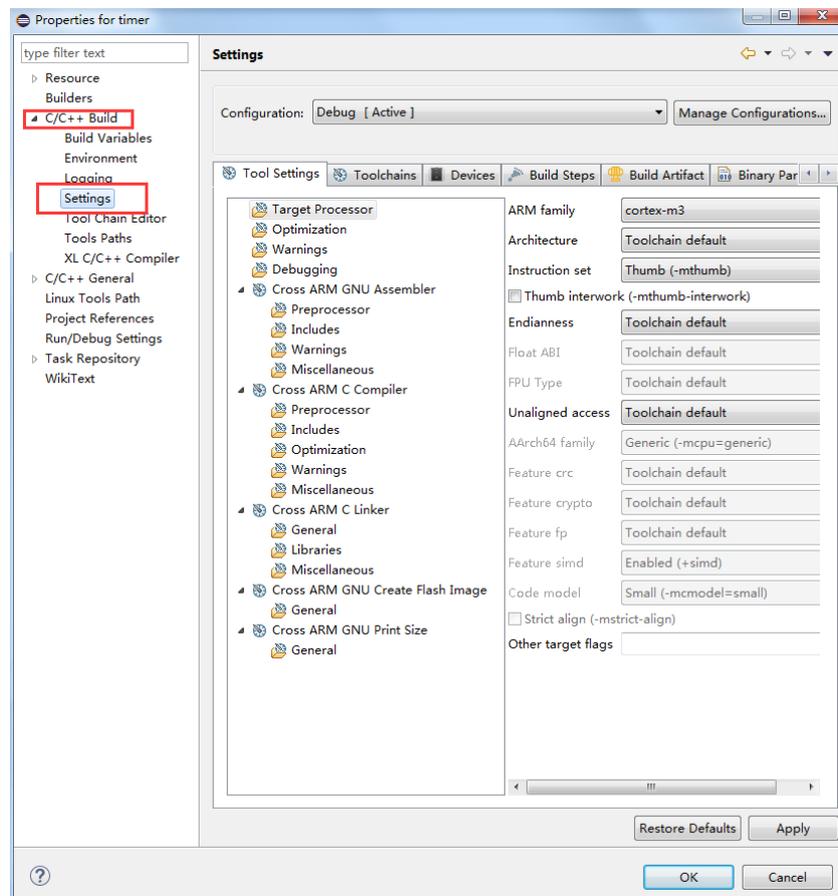
导入工程

选择当前工程，右键选择 Refresh，自动导入如图 2-4 所示的工程结构和代码。

2.2.2 工程配置

选择当前工程，右键选择 Properties，选择选项卡 C/C++ Build，选择 Setting 选项，如图 2-5 所示。

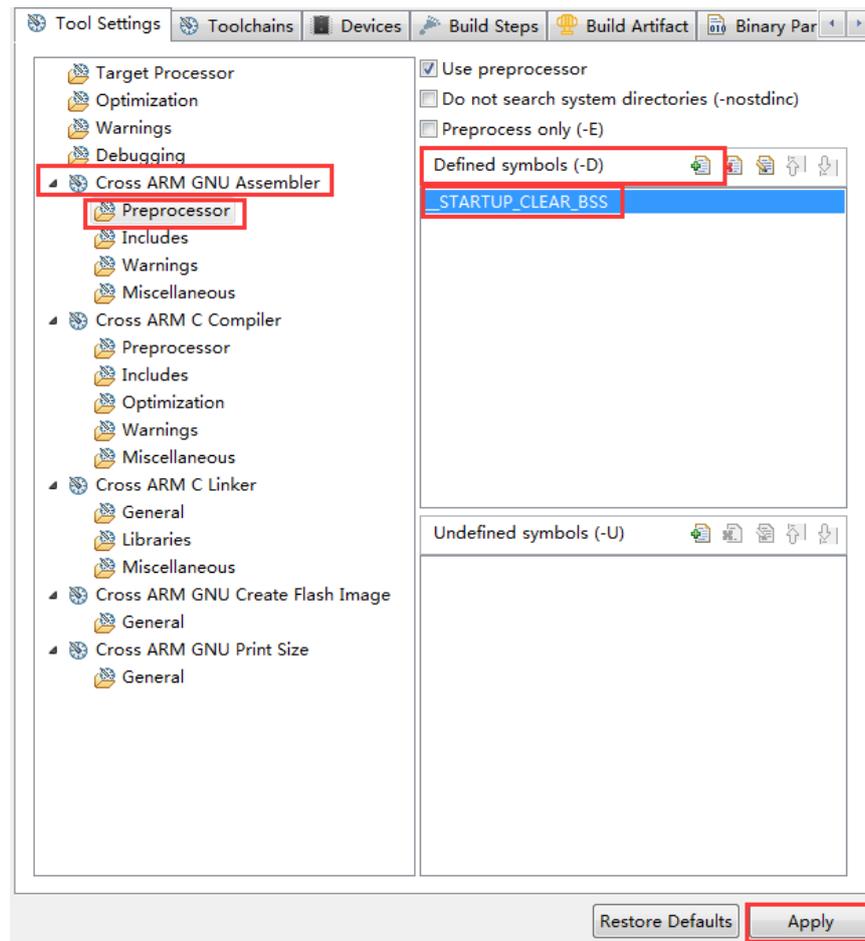
图 2-5 选择 Properties



配置 Cross ARM GNU Assembler Preprocessor

选择 Cross ARM GNU Assembler 中的 Preprocessor，配置汇编宏定义 `__STARTUP_CLEAR_BSS`，如图 2-6 所示。

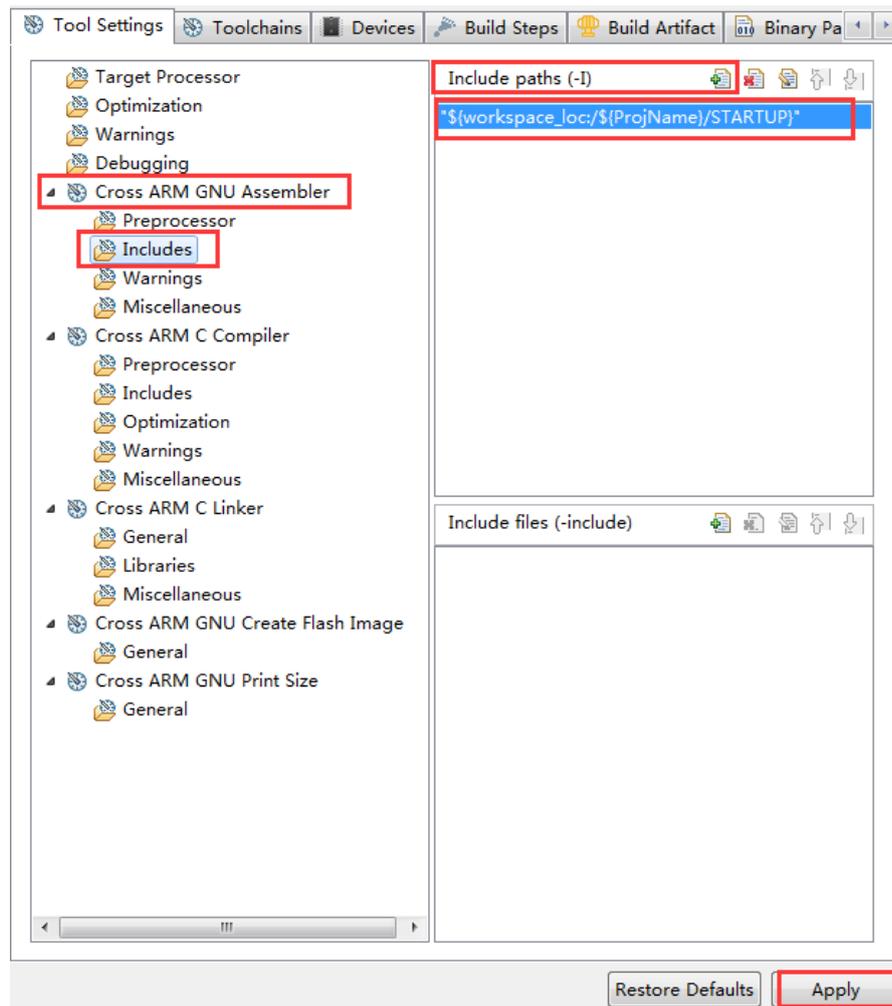
图 2-6 配置 Cross ARM GNU Assembler Preprocessor



配置 Cross ARM GNU Assembler Includes

选择 Cross ARM GNU Assembler 中的 Includes, 配置汇编引用文件路径, 如图 2-7 所示。

图 2-7 配置 Cross ARM GNU Assembler Includes



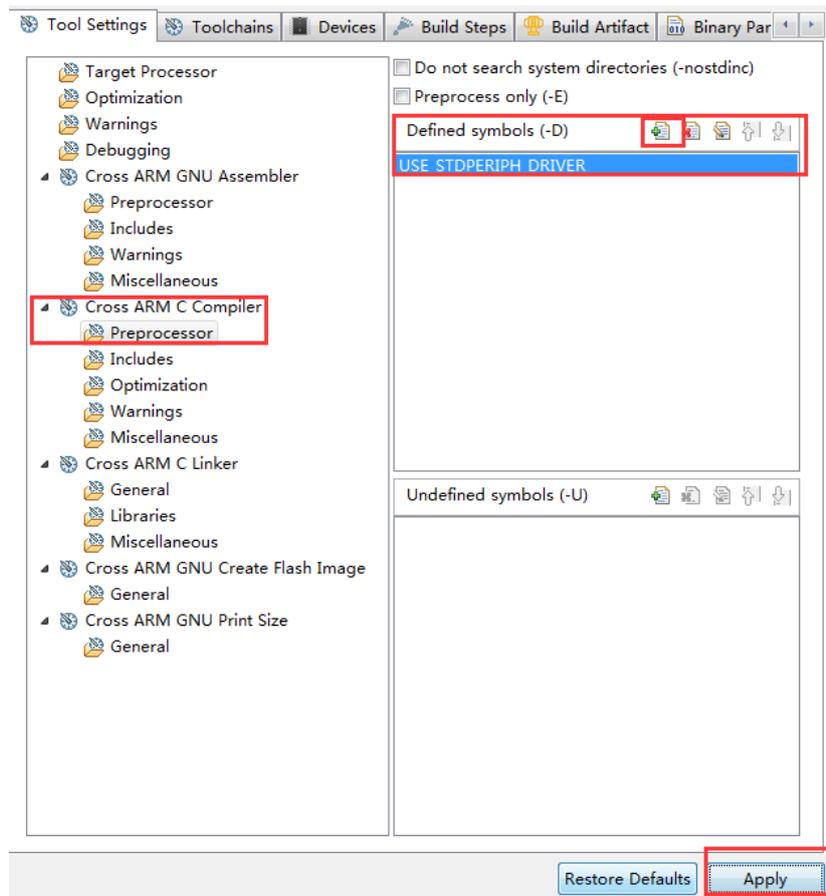
以 GNU_RefDesign 参考设计为例，Assembler 头文件路径配置如下所示。

```
"${workspace_loc}/${ProjName}/STARTUP"
```

配置 Cross ARM C Compiler Preprocessor

选择 Cross ARM C Compiler 中的 Preprocessor，配置 C 宏定义 USE_STDPERIPH_DRIVER，如图 2-8 所示。

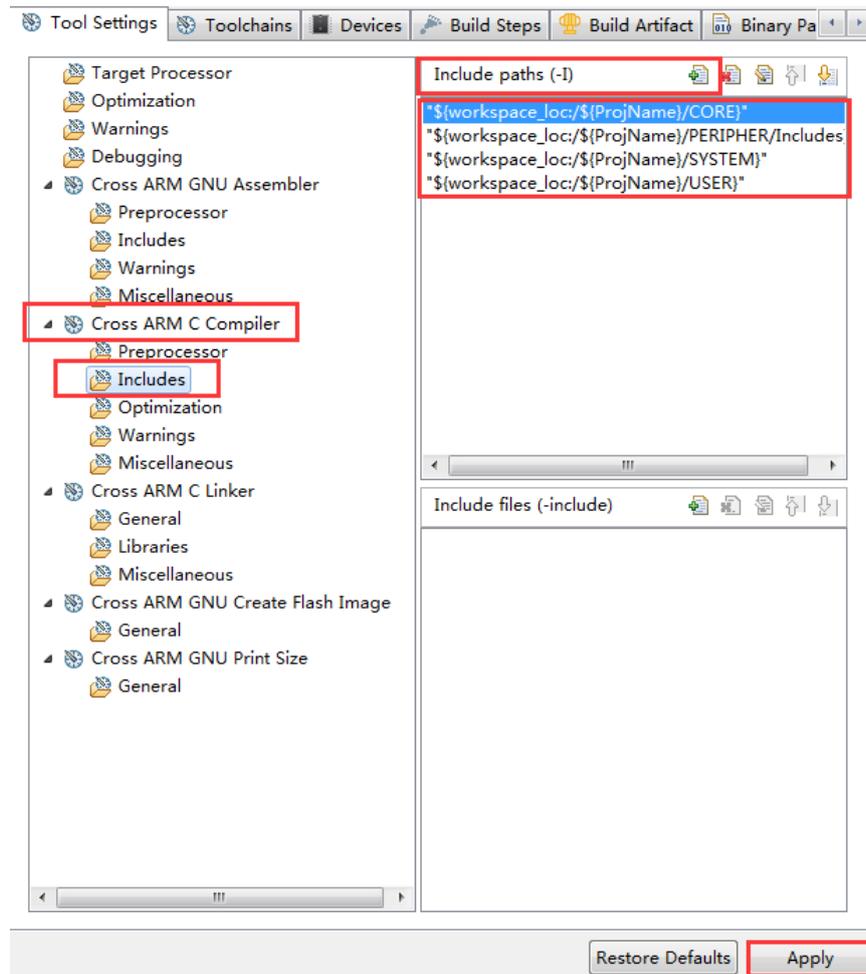
图 2-8 配置 Cross ARM C Compiler Preprocessor



配置 Cross ARM C Compiler Includes

选择 Cross ARM C Compiler 中的 Includes，配置 C 引用文件路径，如图 2-9 所示。

图 2-9 配置 Cross ARM C Compiler Includes



以 GNU_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"
```

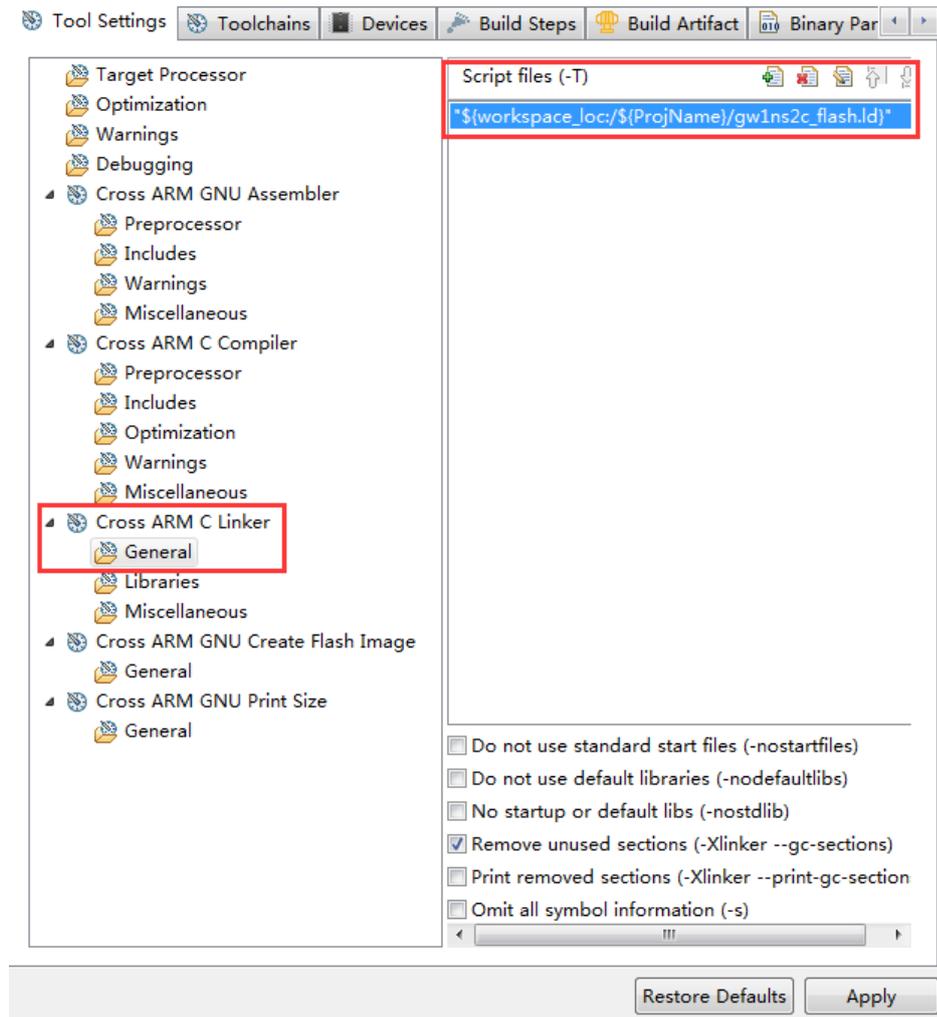
```
"${workspace_loc}/${ProjName}/SYSTEM"
```

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

配置 Cross ARM C Linker

选择 Cross ARM C Linker 中的 General，配置 Flash 链接脚本 gw1ns2c_flash.Id，如图 2-10 所示。

图 2-10 配置 Cross ARM C Linker



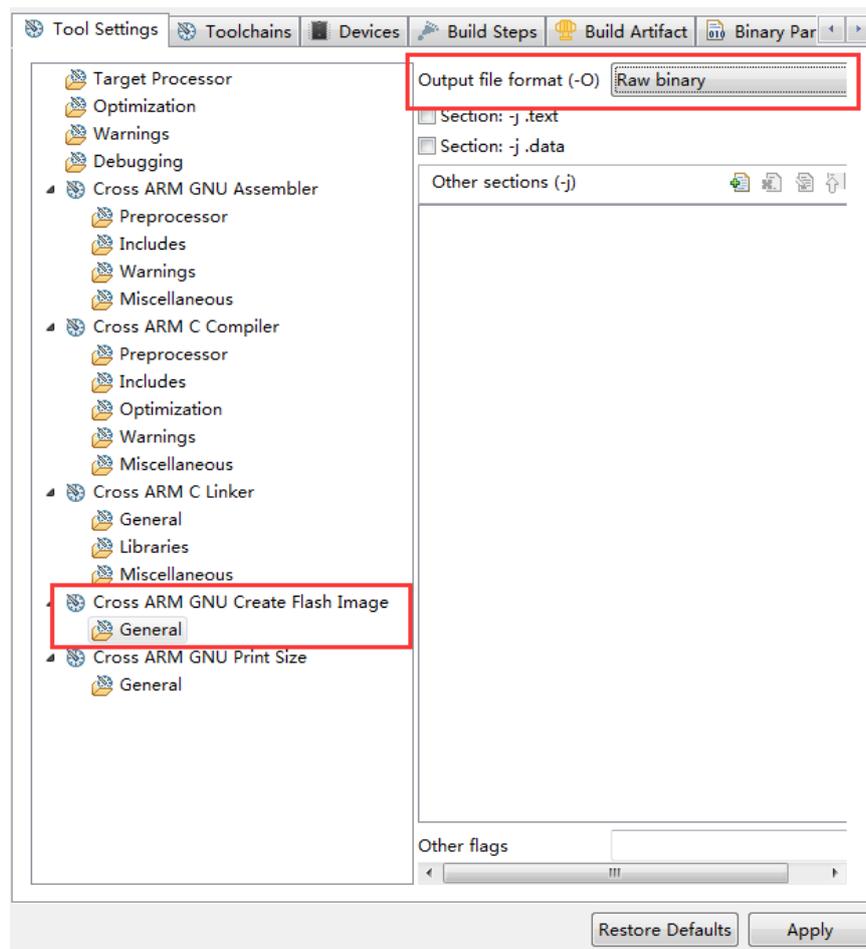
以 GNU_RefDesign 参考设计为例，Flash 链接脚本如下所示。

```
"${workspace_loc}/${ProjName}/gw1ns2c_flash.ld"
```

配置 Cross ARM GNU Create Flash Image

选择 Cross ARM GNU Create Flash Image 中的 General，配置 Gowin_EMPU 映像文件格式为 RAW binary，如图 2-11 所示。

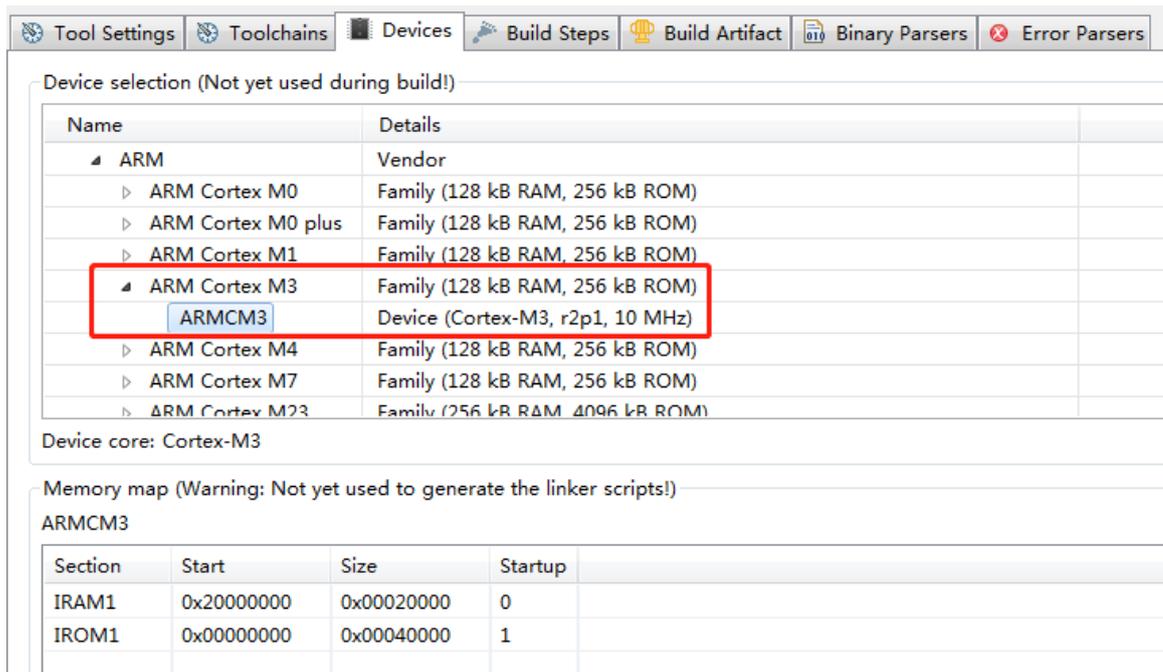
图 2-11 配置 Cross ARM GNU Create Flash Image



配置 Devices

选择 Devices 选项卡, 选择器件 ARM Cortex-M3 “ARMCM3”, 如图 2-12 所示。

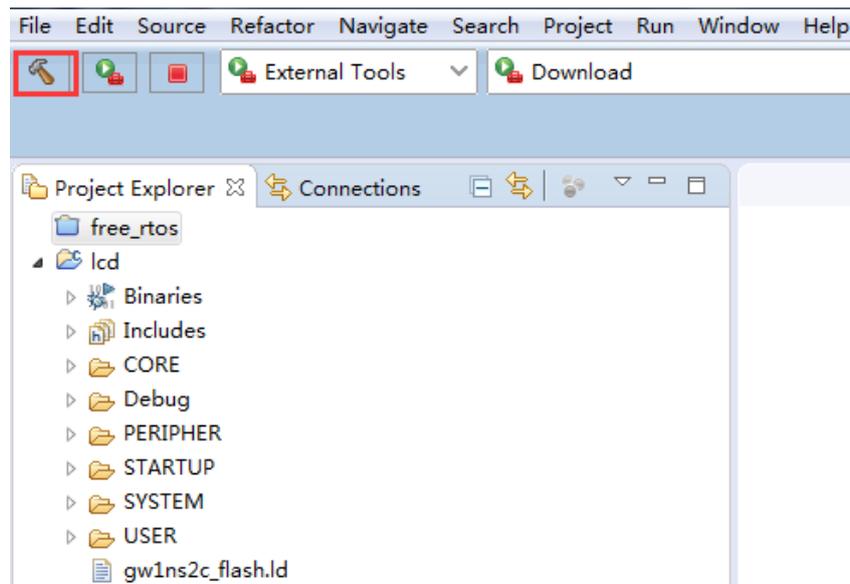
图 2-12 配置 Devices



2.2.3 工程编译

完成工程配置和编码后，编译工程，选择工具栏编译按钮，开始编译，如图 2-13 所示。

图 2-13 编译工程

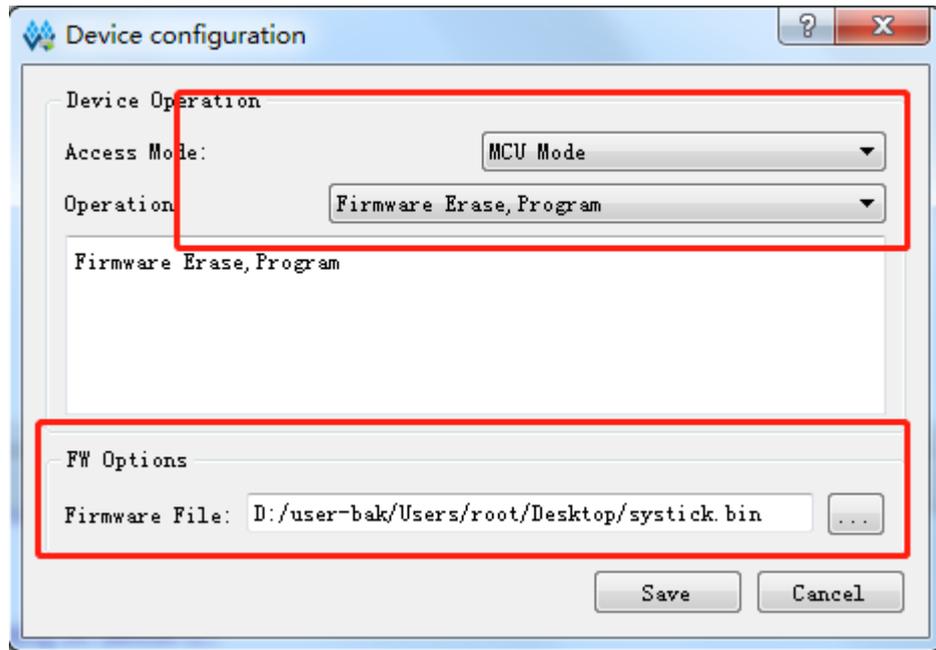


2.2.4 工程下载

完成工程配置和编译后，可以使用两种下载方法：

- 第一种方法：点击 GOWIN MCU Designer 工具栏下载按钮，调用已经配置的下下载工具 download，下载 Gowin_EMPU 二进制 BIN 文件。
- 第二种方法：打开 Programmer 工具，配置器件访问模式为 MCU Mode，操作选项选择 Firmware Erase, Program，Firmware File 导入 Gowin_EMPU 二进制 BIN 文件，下载 Gowin_EMPU 二进制 BIN 文件，如图 2-14 所示。

图 2-14 工程下载

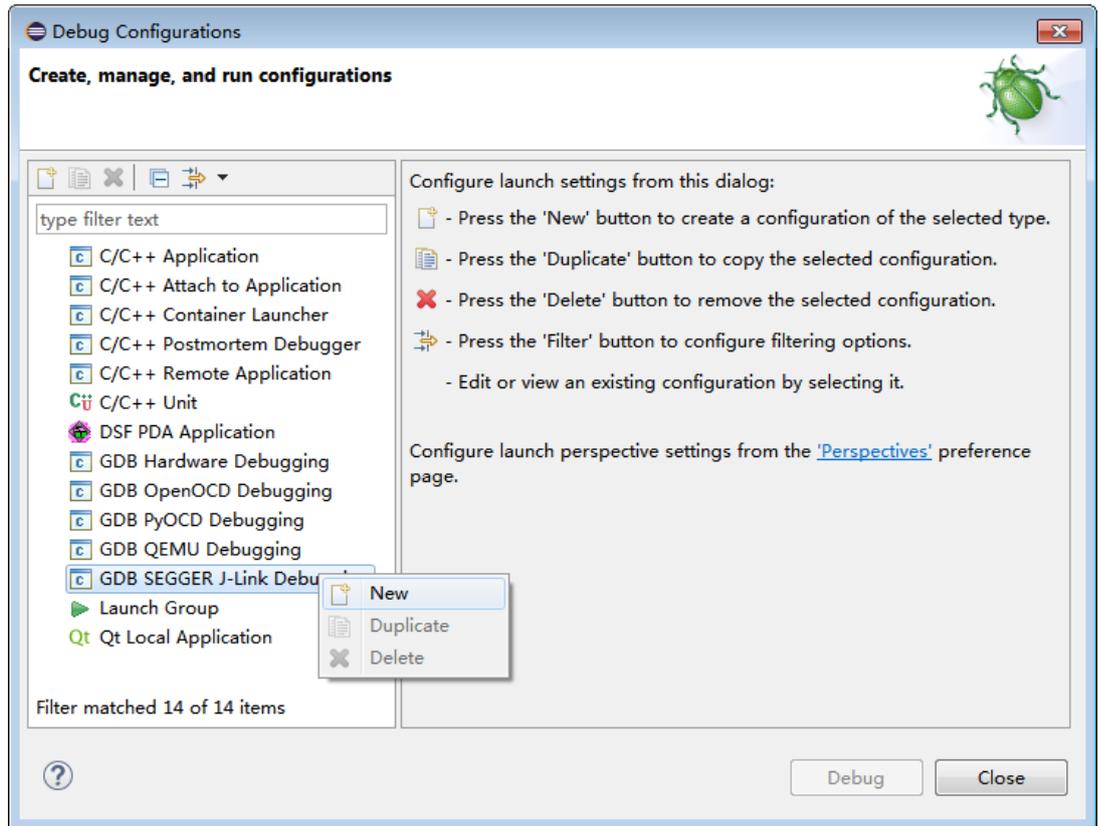


2.2.5 工程调试

配置调试选项

选择菜单栏 Run 中的 Debug Configurations 下的 GDB SEGGER J-Link Debugging 选项卡，右键选择 New，建立工程的调试配置选项，如图 2-15 所示。

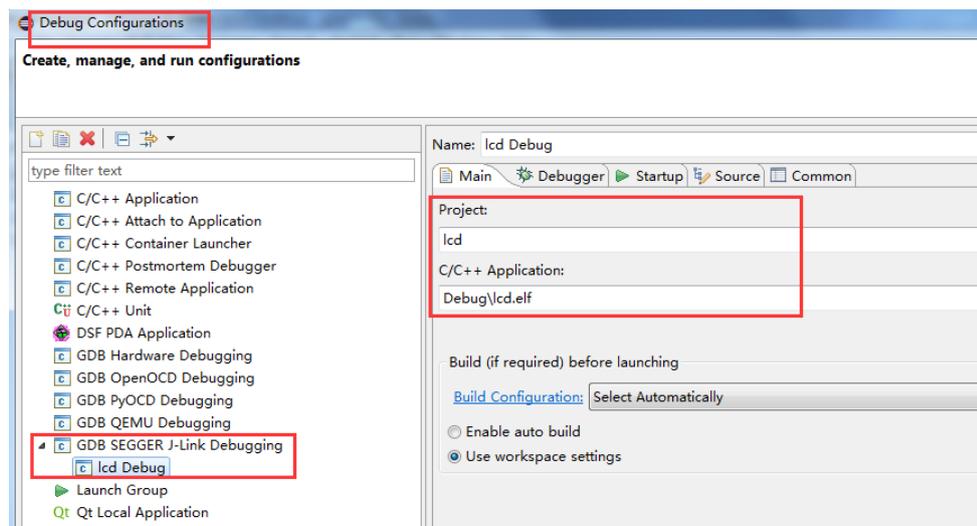
图 2-15 建立调试配置选项



选择已建立的调试配置选项。

1. 选择 Main 选项卡，配置映像文件路径，如图 2-16 所示。

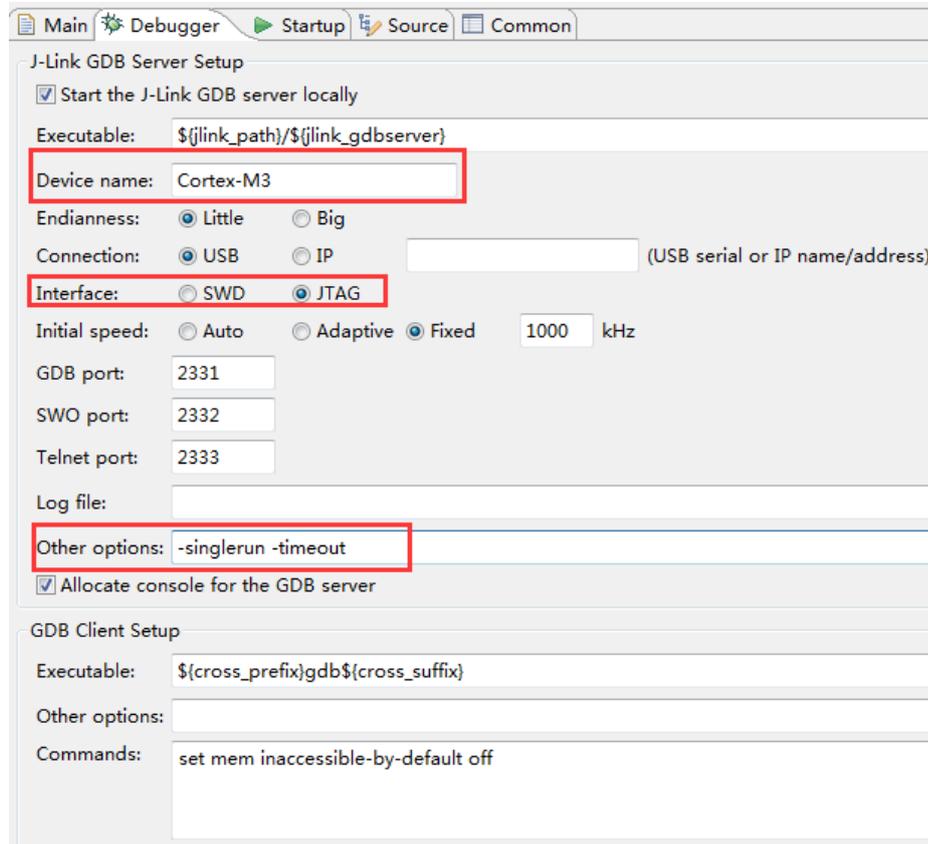
图 2-16 配置 Image 路径



2. 选择 Debugger 选项卡，配置 Debugger 器件名称和调试接口为 JTAG，如图 2-17 所示。

- Device Name: Cortex-M3
- Interface: JTAG
- Other options: -singlerun -timeout

图 2-17 配置 Debugger



JTAG 模式切换

完成调试模式配置后，可以使用两种方法完成 JTAG 模式切换：

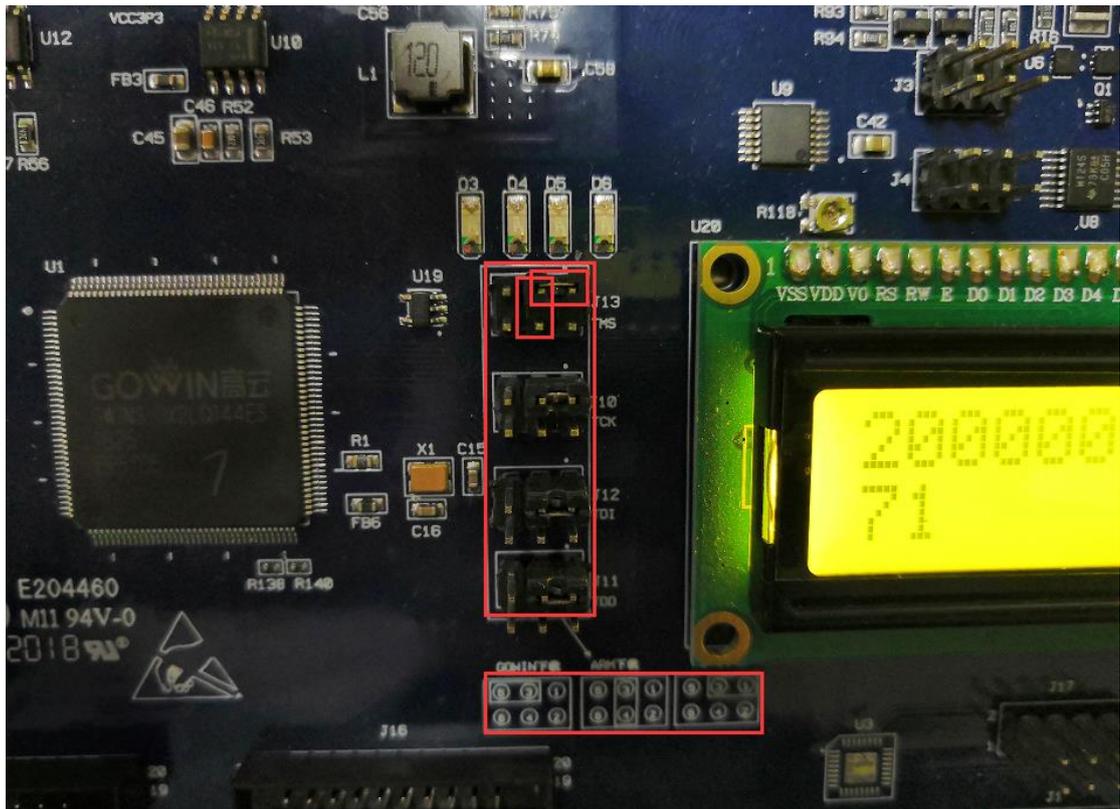
- 第一种方法：调用上述软件配置中已配置的 JTAG 模式切换工具 `switchmode`，完成 JTAG 软件模式由下载模式切换到调试模式。
- 第二种方法：打开 Programmer，配置器件访问模式为 MCU Mode，操作模式为 Connect to JTAG of MCU，完成 JTAG 软件模式由下载模式切换到调试模式。

JTAG 接口切换

如果使用开发板“DK-EVAL-GW1NS2 V1.1”，需要将开发板上的 JTAG 接口 TMS、TCK、TDI、TDO 四个跳线帽由(1,3)端切换到(3,4)端口，完成 JTAG 接口切换，如图 2-18 所示。

如果使用开发板“DK-START-GW1NS2 V1.1”，需要将开发板上的 JTAG 接口 TMS、TCK、TDI、TDO 四个拨码开关由 FPGA 下载切换到 ARM 下载。

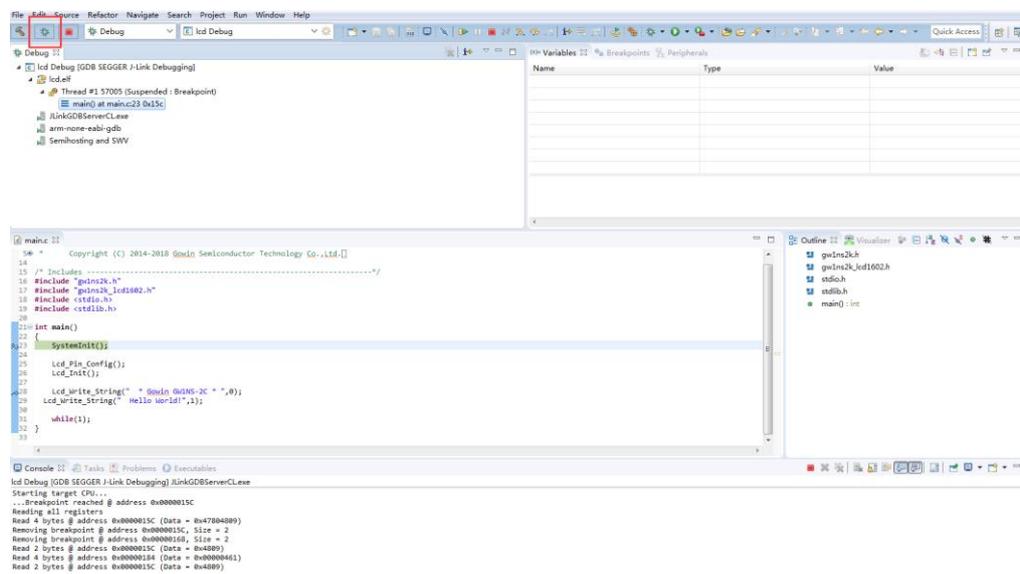
图 2-18 JTAG 接口切换



调试

连接 J-LINK 仿真器，选择工具栏 Debug 按钮，开始调试，如图 2-19 所示。

图 2-19 启动调试



2.3 参考设计

Gowin_EMPU for GW1NS-2C 支持 GOWIN MCU Designer 软件环境的参考设计:

Gowin_EMPU\ref_design\MCU_RefDesign\GNU_RefDesign

