



GW1NS-2C MCU IDE

软件参考手册

RN519-1.2,2019-04-12

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

目录

目录	i
图目录	iii
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 GNU MCU Eclipse 软件	10
2.1 软件安装	10
2.1.1 安装 JDK	11
2.1.2 安装 J-LINK	15
2.1.3 安装 MCU 器件包	18
2.2 软件配置	21
2.2.1 导入模板工程	21
2.2.2 选择 Properties	22
2.2.3 配置环境变量	23
2.2.4 配置工具链	24
2.2.5 配置下载工具	25
2.2.6 配置 JTAG 模式切换工具	25
2.3 工程模板	26
2.3.1 工程创建	26
2.3.2 工程创建	30
2.3.3 工程编译	38

2.3.4 工程下载.....	39
2.3.5 工程调试.....	40
2.4 参考设计.....	44

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	7
图 1-12 启动调试	9
图 2-1 GoWin GNU MCU Eclipse 安装包目录结构	10
图 2-2 JDK 和 J-LINK 安装目录结构.....	11
图 2-3 JDK 安装向导	12
图 2-4 选择安装 JDK 可选功能	13
图 2-5 选择 JRE 安装路径	13
图 2-6 完成 JDK 安装.....	14
图 2-7 测试 JDK 安装.....	14
图 2-8 J-LINK License Agreement.....	15
图 2-9 选择安装路径	16
图 2-10 选择功能组件	16
图 2-11 开始安装向导	17
图 2-12 完成安装	18
图 2-13 选择 Preferences.....	19
图 2-14 指定 Packages 路径.....	20
图 2-15 更新器件包列表	21

图 2-16 导入模板工程.....	22
图 2-17 选择 Properties.....	23
图 2-18 配置环境变量.....	24
图 2-19 配置工具链.....	24
图 2-20 配置下载工具.....	25
图 2-21 配置 JTAG 模式切换工具.....	26
图 2-22 交叉编译配置.....	26
图 2-23 新建工程.....	27
图 2-24 选择工程类型配置.....	28
图 2-25 选择工具链.....	29
图 2-26 工程结构.....	30
图 2-27 选择 Properties.....	31
图 2-28 配置 Cross ARM GNU Assembler Preprocessor.....	32
图 2-29 配置 Cross ARM GNU Assembler Includes.....	33
图 2-30 配置 Cross ARM C Compiler Preprocessor.....	34
图 2-31 配置 Cross ARM C Compiler Includes.....	35
图 2-32 配置 Cross ARM C Linker.....	36
图 2-33 配置 Cross ARM GNU Create Flash Image.....	37
图 2-34 配置 Devices.....	38
图 2-35 编译工程.....	39
图 2-36 工程下载.....	39
图 2-37 建立调试配置选项.....	41
图 2-38 配置 Image 路径.....	41
图 2-39 配置 Debugger.....	42
图 2-40 JTAG 接口切换.....	43
图 2-41 启动调试.....	44

1 ARM Keil MDK 软件

1.1 软件安装

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

1.2 工程模板

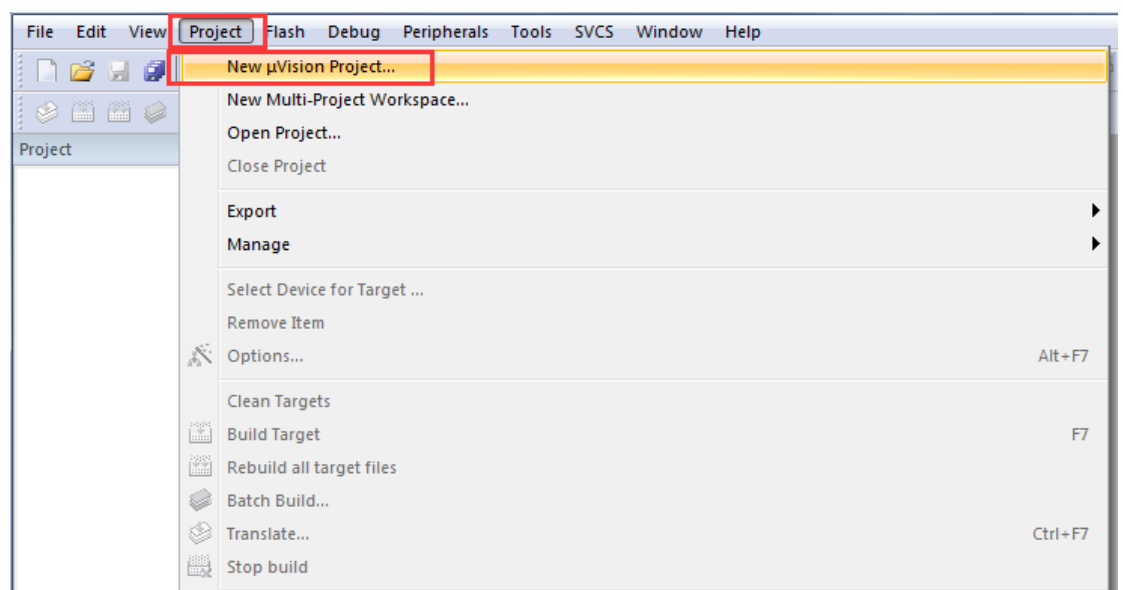
使用 ARM Keil MDK 软件进行嵌入式软件开发，需要创建工程、配置工程选项、编码、编译、下载和调试。

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

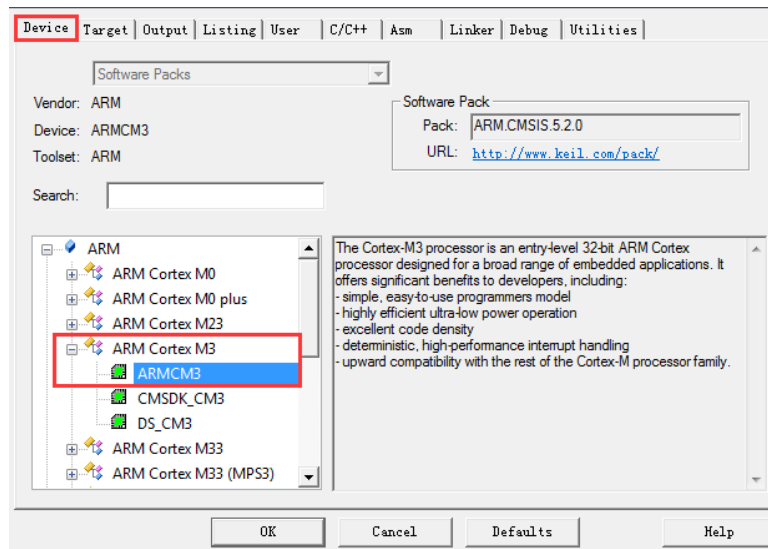


1.2.2 工程配置

配置器件

GW1NS-2C MCU 为 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



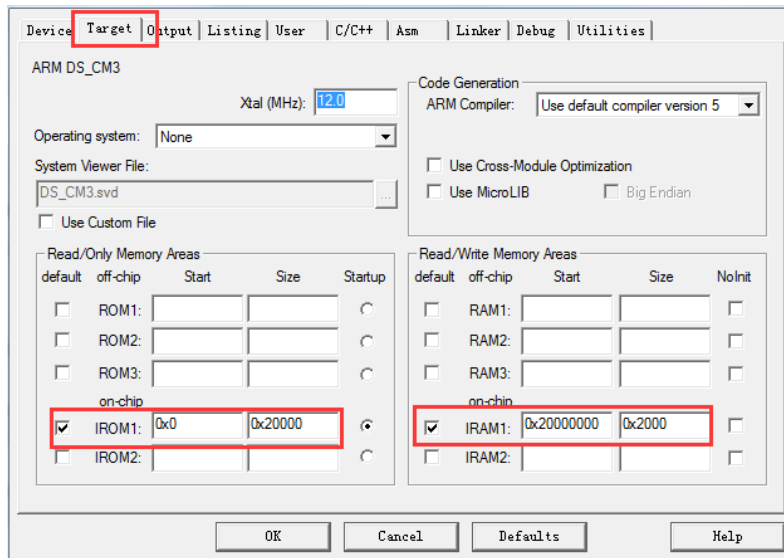
配置 ROM 和 RAM

配置 ROM 和 RAM 的起始地址和容量大小。

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

GW1NS-2C Sram 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

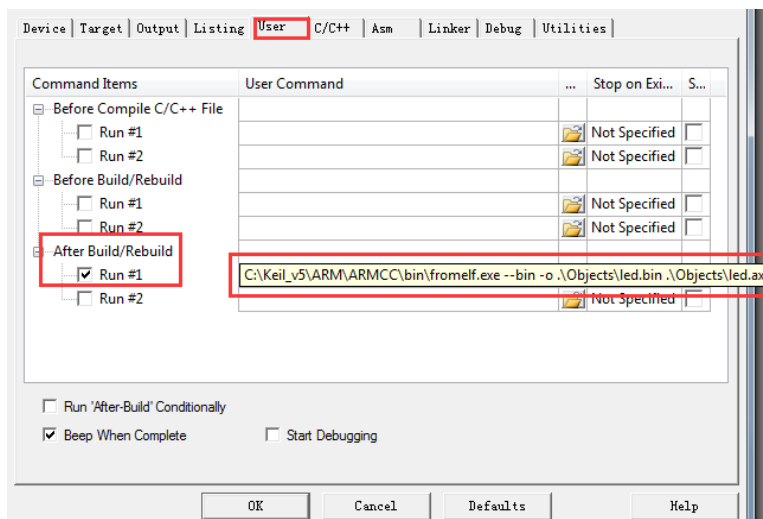


配置输出文件格式

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：

```
*:\Keil_v5\ARM\ARMCC\bin\fromelf.exe --bin
```

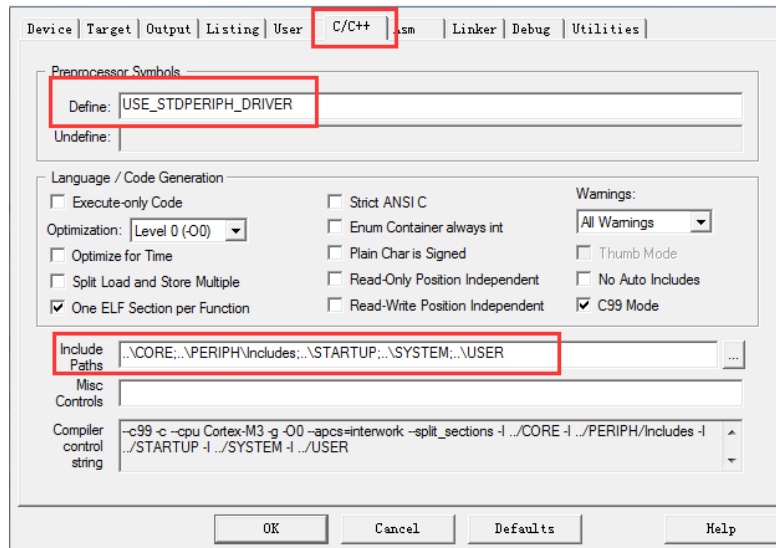
-o ./Objects/*.bin ./Objects/*.axf

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。

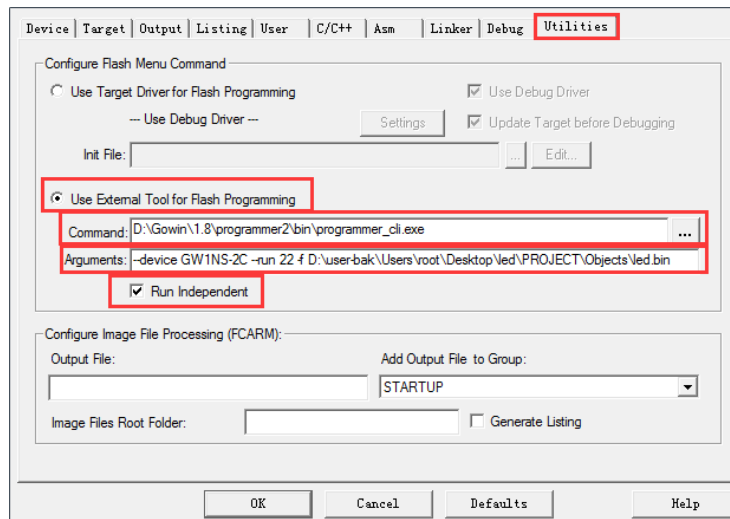
图 1-5 配置宏定义和头文件路径



配置下载工具

配置 Flash 下载工具为外部工具，使用下载工具 Programmer，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

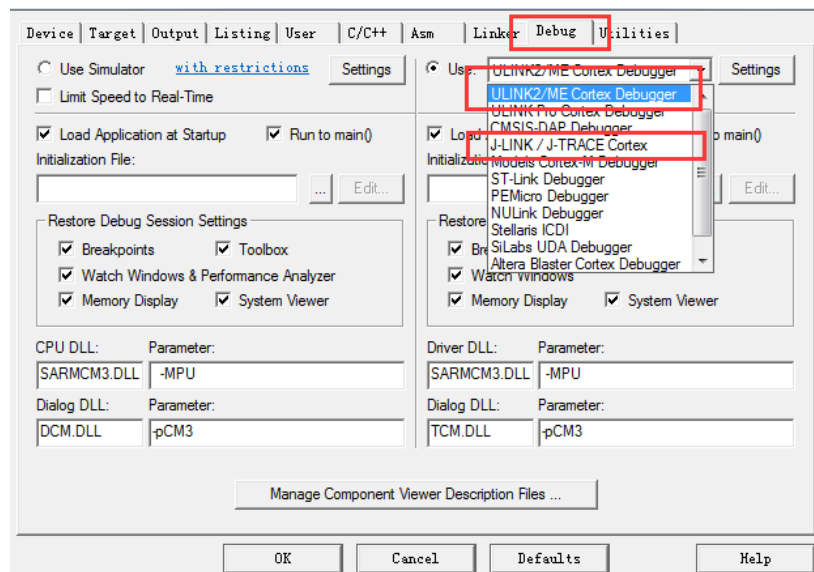
*.bin。

配置调试选项

如果选择使用 U-LINK 仿真器,则 Debug 选项配置为“ULINK2/ME Cortex Debugger”。

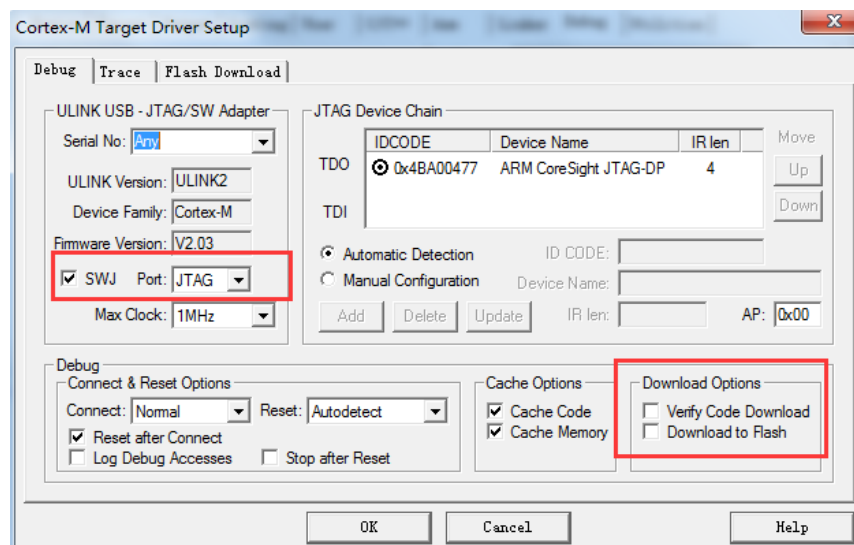
如果选择使用 J-LINK 仿真器,则 Debug 选项配置为“J-LINK/J-TRACE Cortex”。如图 1-7 所示。

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。

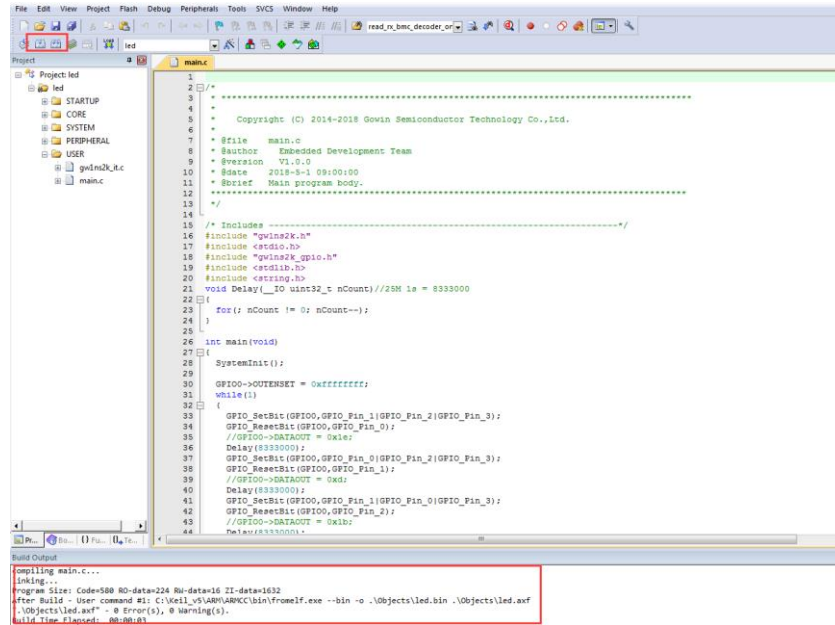
图 1-8 调试接口类型



1.2.3 工程编译

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

图 1-9 工程编译



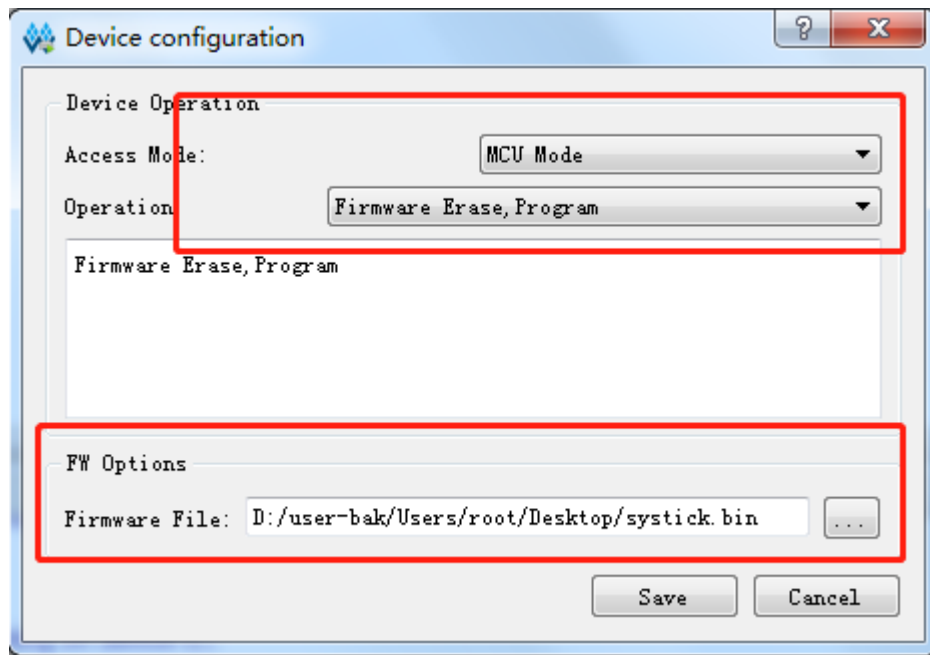
1.2.4 工程下载

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工具 Programmer，下载 MCU 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 MCU 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

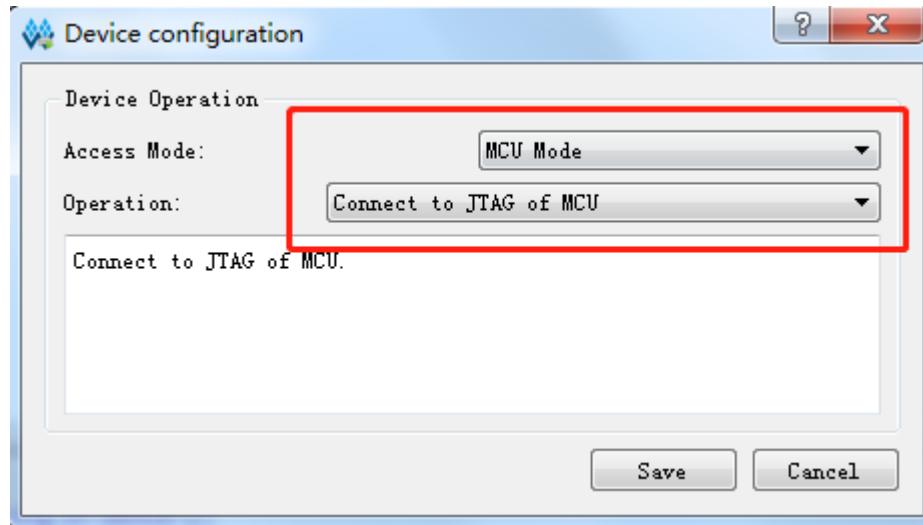
完成 MCU 二进制 BIN 文件下载后，如果用户设计出现问题，可以连接 U-LINK 或 J-LINK 仿真器使用调试功能。

JTAG 模式切换

如图 1-11 所示，使用 Programmer 工具将 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

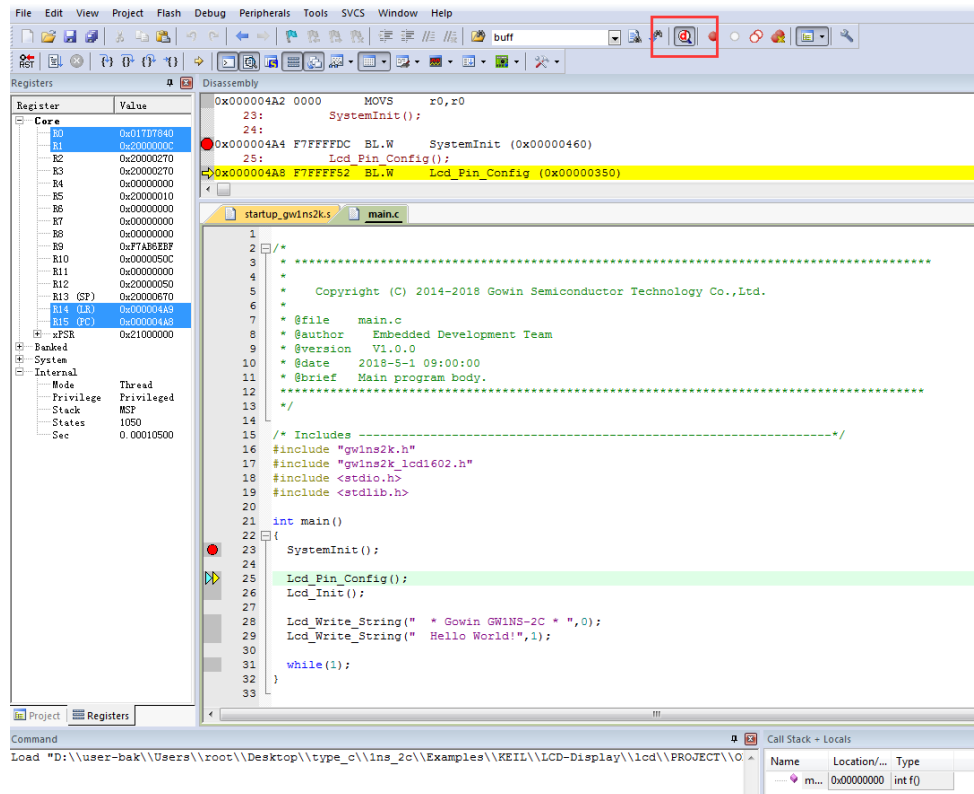
如果使用开发板“DK-EVAL-GW1NS2 V1.1”，需要手动将 JTAG 接口 TMS、TCK、TDI 和 TDO 跳线帽由 FDTI 下载切换到 ARM 下载，即由(1,3)端口切换到(3,4)端口。

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 拨码开关切换由 FPGA 下载到 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

GW1NS-2C MCU 提供 ARM Keil MDK 软件环境的参考设计：
Gowin_EMPU_RefDesign\MCU_RefDesign\Keil_RefDesign

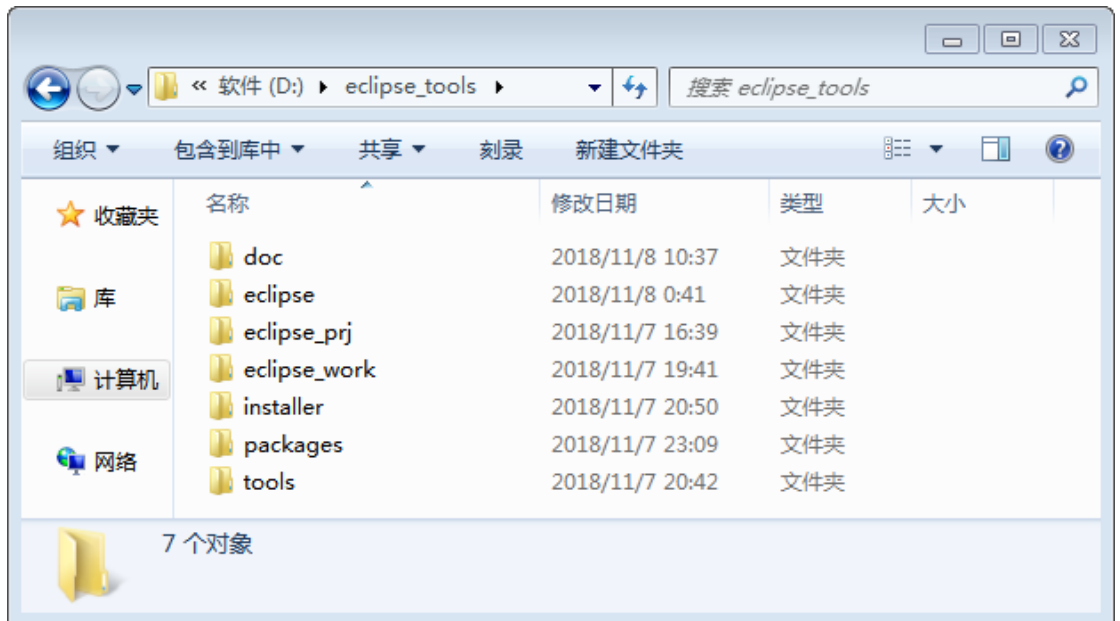
2 GoWin GNU MCU Eclipse 软件

2.1 软件安装

高云提供 GoWin GNU MCU Eclipse 安装包。

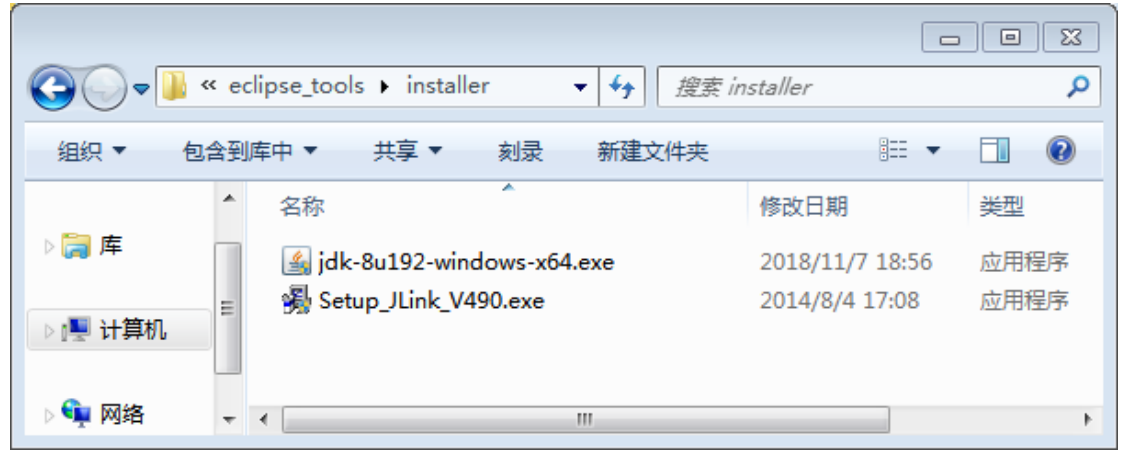
获取 GoWin GNU MCU Eclipse 安装包，解压到本地 D 盘根目录下，如图 2-1 所示。

图 2-1 GoWin GNU MCU Eclipse 安装包目录结构



选择 `installer` 目录, 获取 JDK 安装文件和 J-LINK 仿真器驱动安装文件, 如图 2-2 所示。

图 2-2 JDK 和 J-LINK 安装目录结构



2.1.1 安装 JDK

选择 JDK 安装文件

选择安装文件 `jdk-8u192-windows-x64.exe`, 双击打开, 如图 2-3 所示。

图 2-3 JDK 安装向导



选择要安装的可选功能和安装目录

选择要安装的可选功能和安装目录，如图 2-4 所示，默认安装路径为 C:\Program Files\Java\jdk1.8.0_192。

图 2-4 选择安装 JDK 可选功能



选择 JRE 的安装目录

选择 JRE 的安装目录，如图 2-5 所示，默认路径为 C:\Program Files\Java\jre1.8.0_192。

图 2-5 选择 JRE 安装路径



完成 JDK 安装

如图 2-6 所示，JDK 安装成功，选择关闭即可。

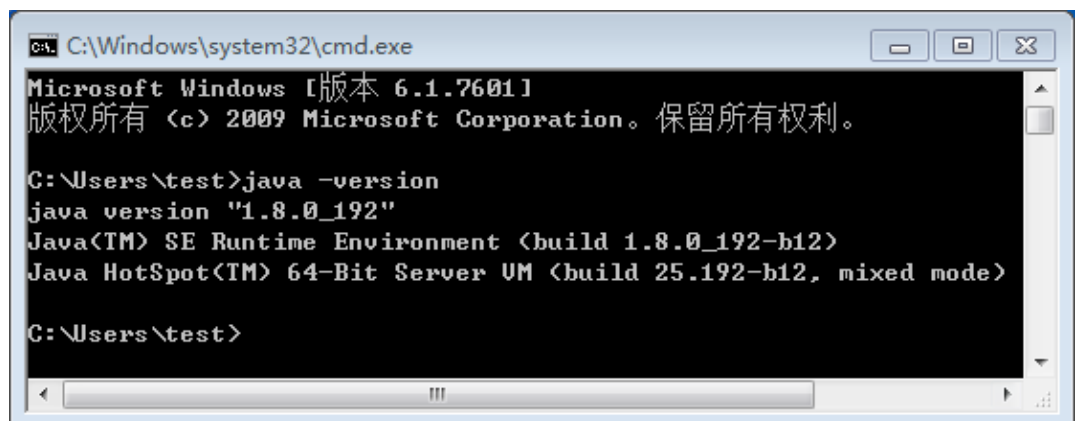
图 2-6 完成 JDK 安装



测试 JDK 安装

本地选择开始菜单，选择打开 cmd 命令行，输入命令“java-version”，如果可以输出 Java 版本信息，则安装成功，如图 2-7 所示。

图 2-7 测试 JDK 安装



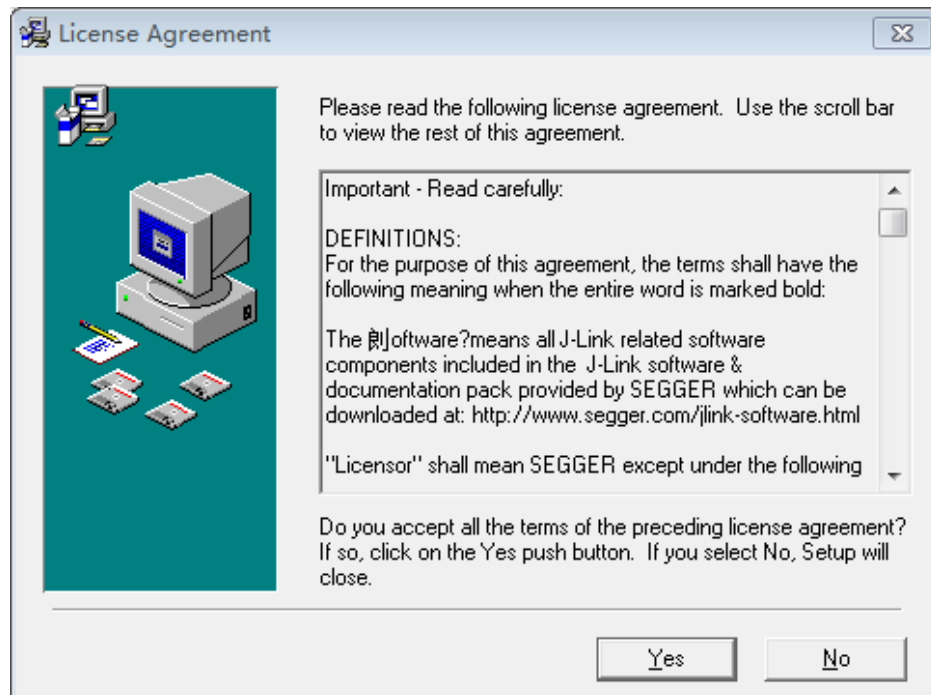
2.1.2 安装 J-LINK

选择安装文件

选择安装文件 `Setup_JLink_V490.exe`，双击打开，如图 2-8 所示，确认 License Agreement。

如果本地已安装 J-LINK，不需要再次安装 J-LINK，只需在 Eclipse 中设置本地已安装的 J-LINK 的路径。

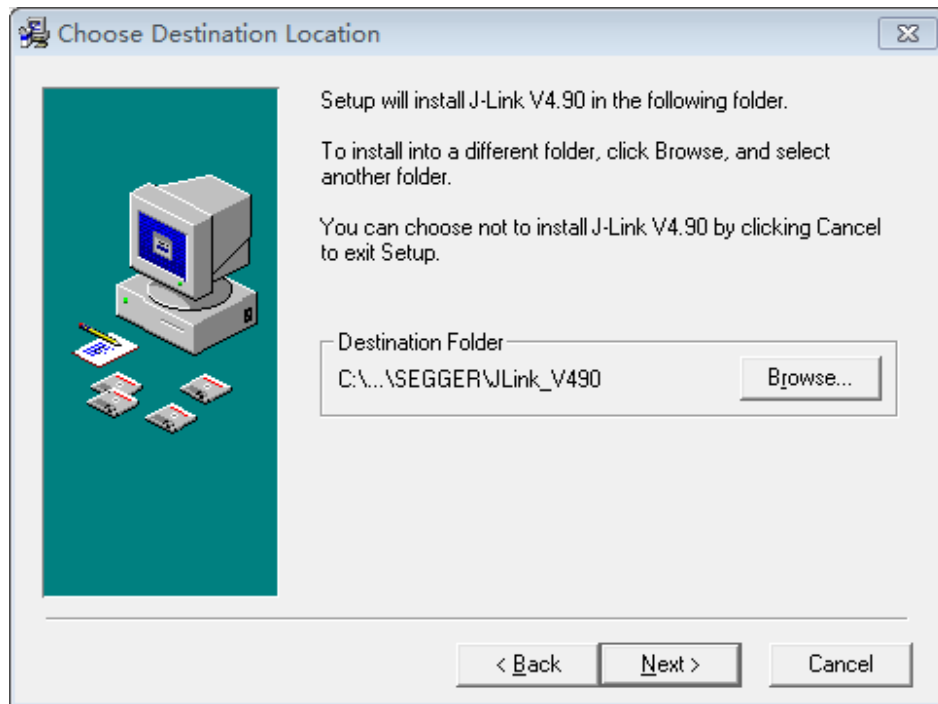
图 2-8 J-LINK License Agreement



选择要安装目录

选择要安装的目录，如图 2-9 所示，默认路径为 `C:\Program Files (x86)\SEGGER\JLink_V490`。

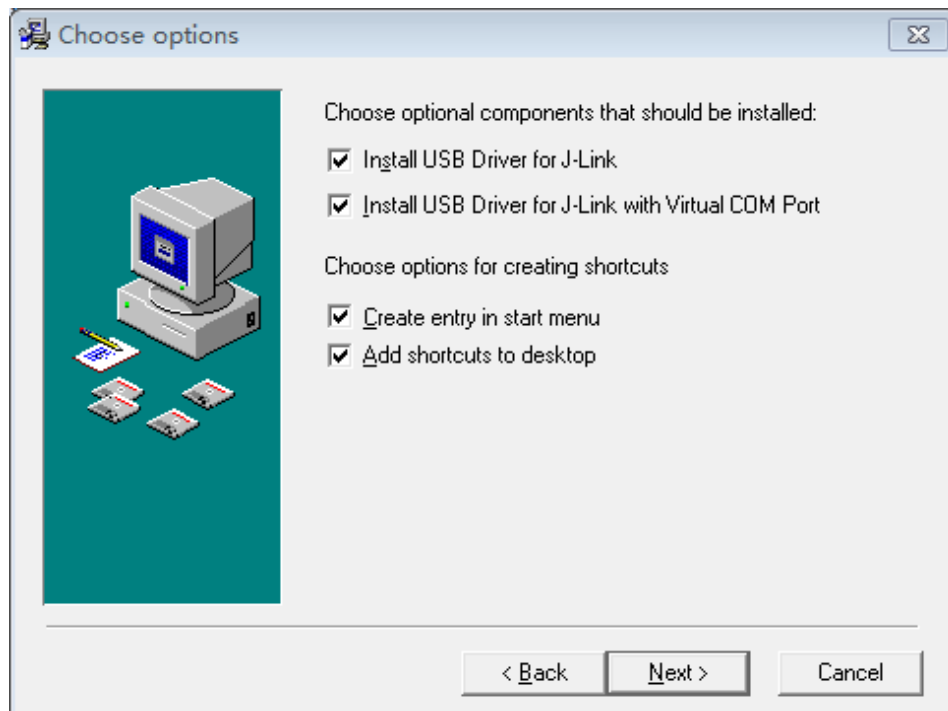
图 2-9 选择安装路径



选择要安装的功能组件

选择要安装的功能组件，如图 2-10 所示。

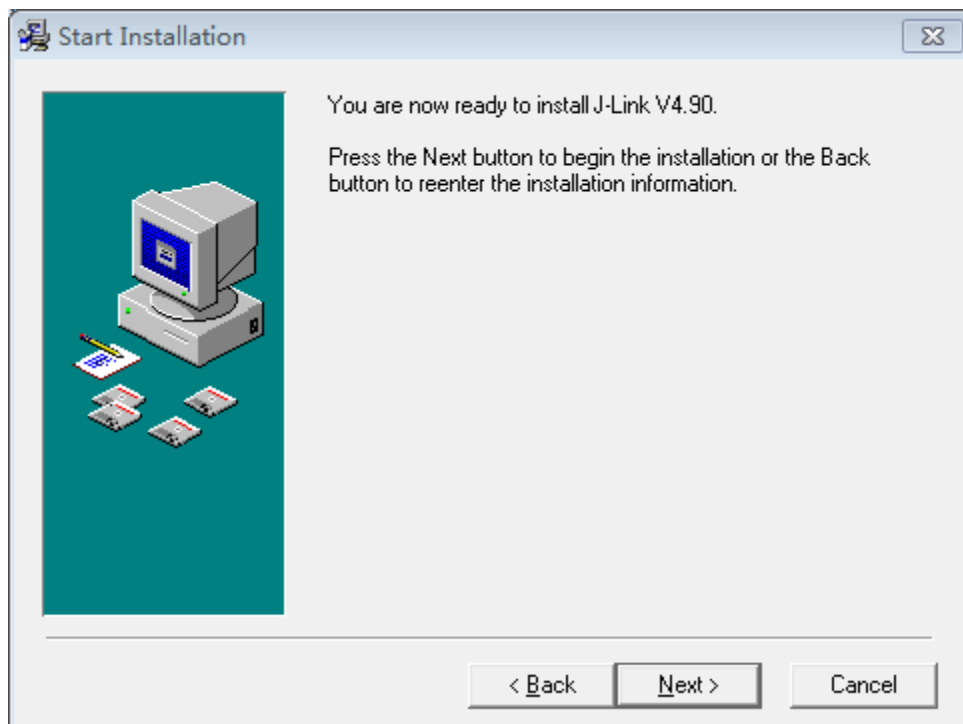
图 2-10 选择功能组件



选择开始安装

开始安装，如图 2-11 所示。

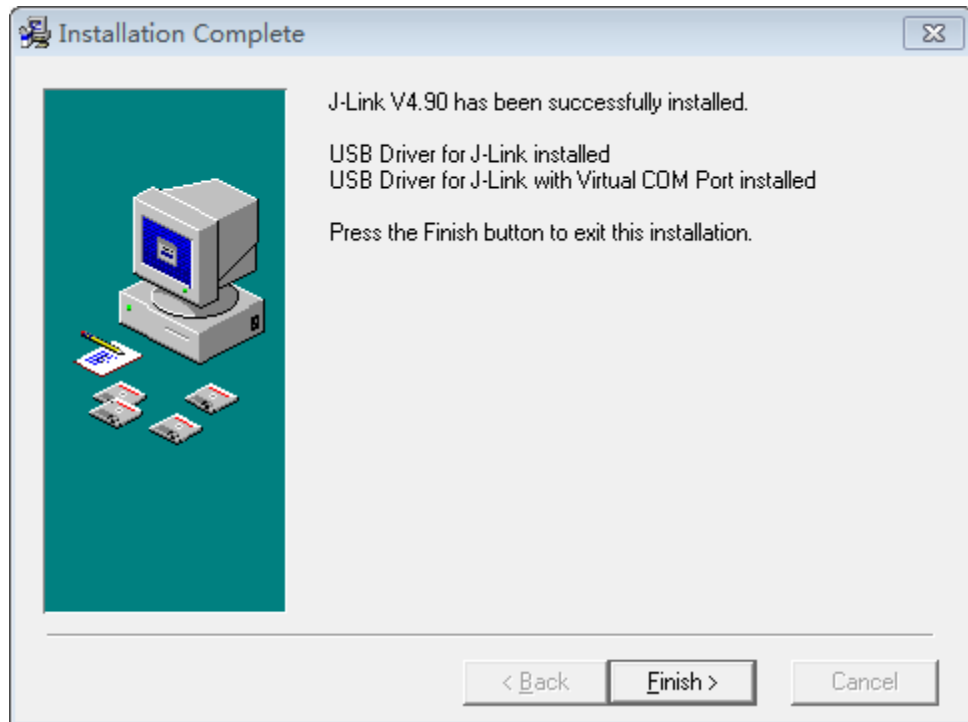
图 2-11 开始安装向导



完成安装

完成安装，如图 2-12 所示。

图 2-12 完成安装



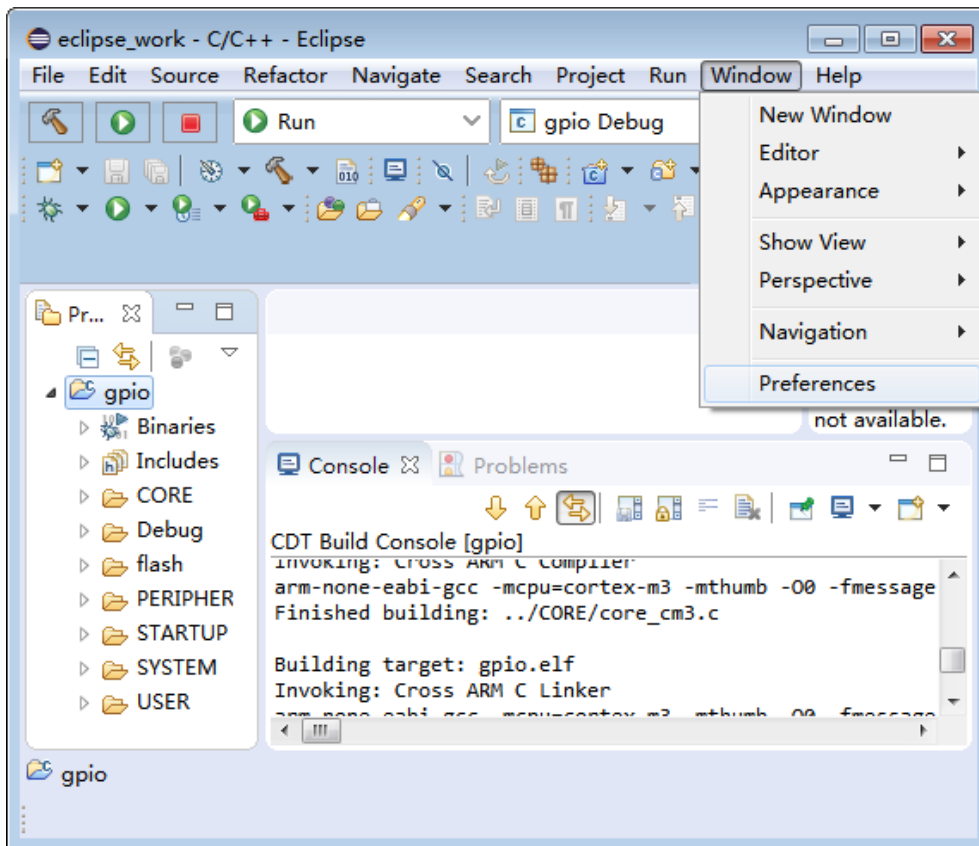
2.1.3 安装 MCU 器件包

Gowin GNU MCU Eclipse 安装包中提供 MCU 器件包 packages，如图 2-1 所示，packages 为 MCU 器件包。

指定 MCU 器件包路径

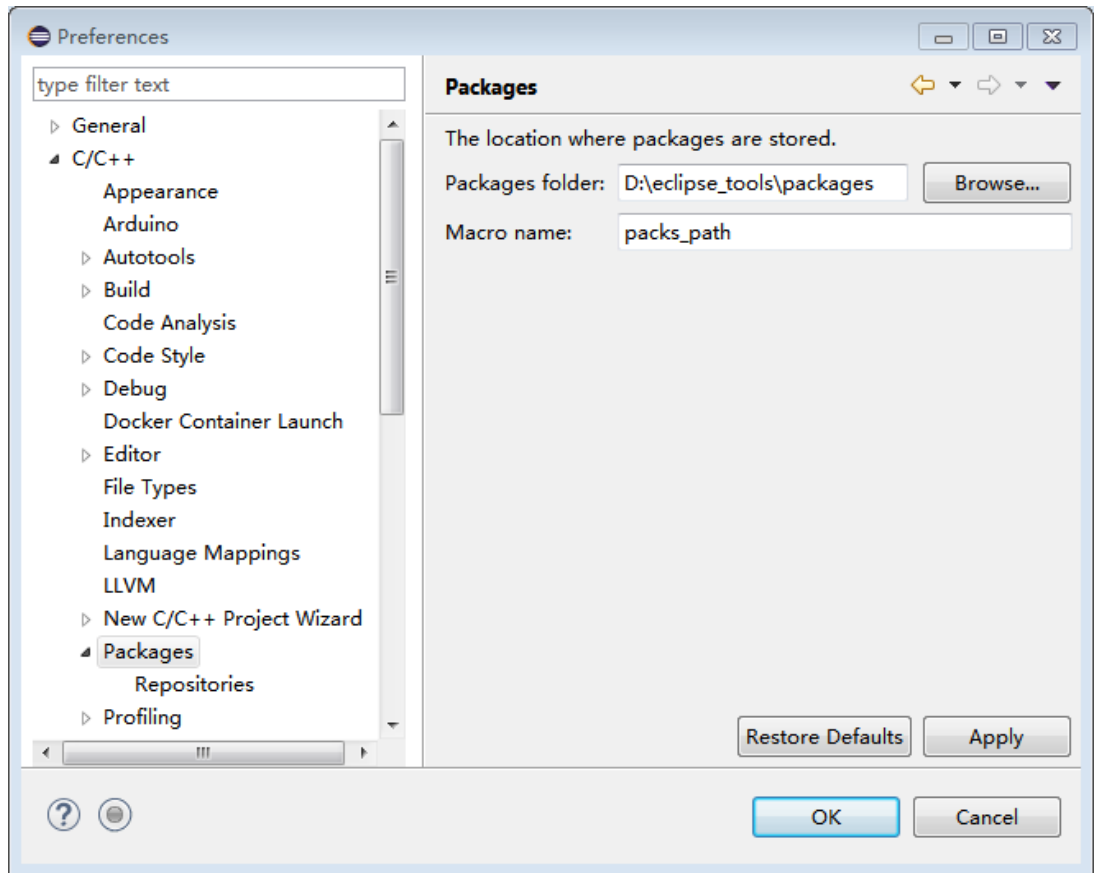
选择菜单栏 Window 中的 Preferences，如图 2-13 所示。

图 2-13 选择 Preferences



选择 Preferences 中的 C/C++选项卡下的 Packages 选项，指定 Packages folder 和 Macro name，如图 2-14 所示。

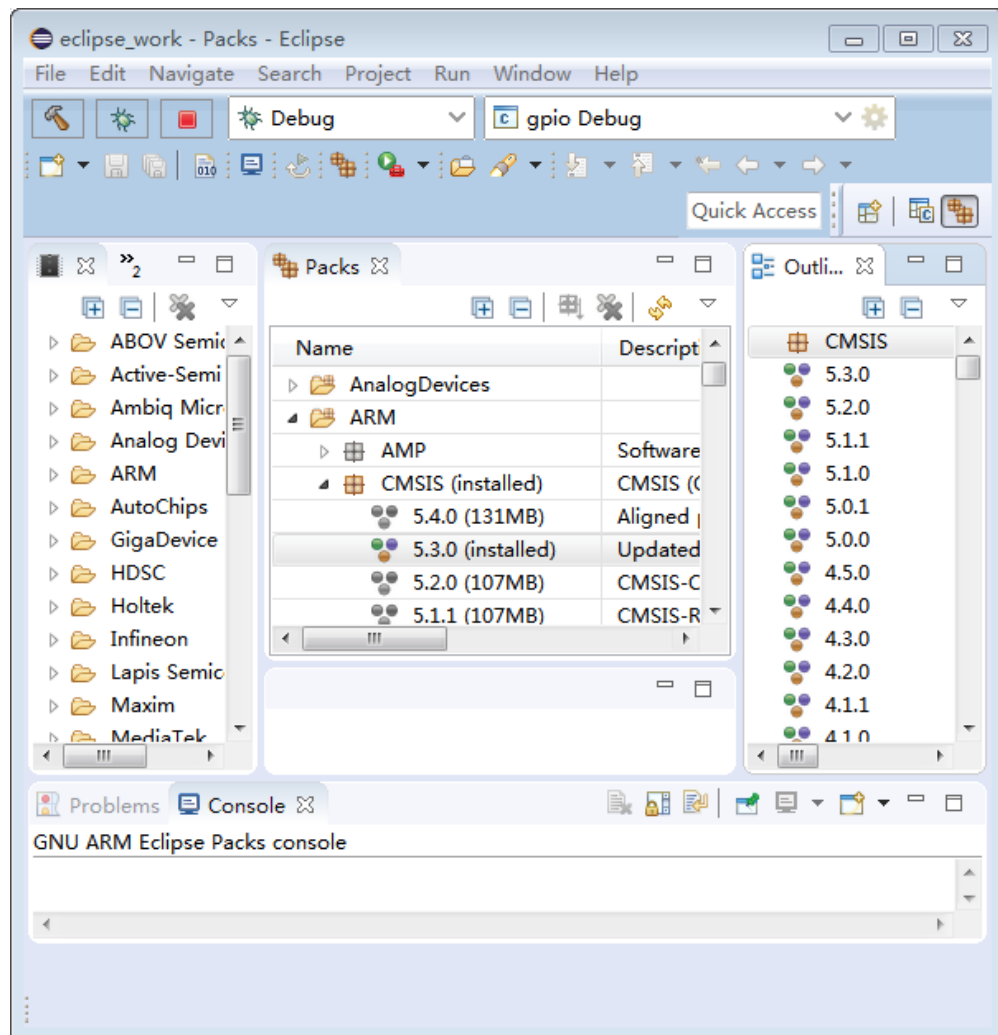
图 2-14 指定 Packages 路径



在线更新 MCU 器件列表

选择工具栏中的 Packs 按钮，在线更新 MCU 器件包列表，如图 2-14 所示。因为 GNU MCU Eclipse 安装包已经提供 ARM.CMSIS.5.3.0 MCU 器件包，所以图 2-15 中显示该器件包已安装。

图 2-15 更新器件包列表



2.2 软件配置

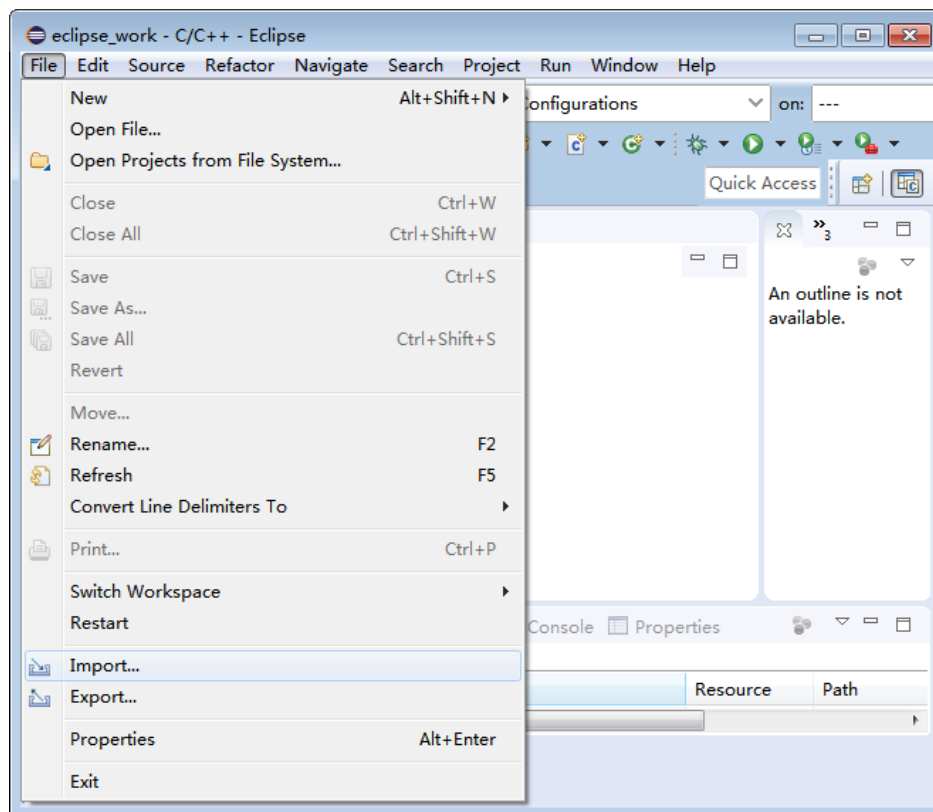
GNU MCU Eclipse 软件安装包中提供已配置的模板工程，导入软件安装包中 `eclipse_proj\demo\gpio` 模板工程来配置 Eclipse 软件。

如果 GNU MCU Eclipse 安装包路径、JDK 安装路径、J-LINK 安装路径和高云云源软件安装路径与如上所述一致，则不需要配置 Eclipse 即可使用 Eclipse 软件进行编译、链接、烧录和调试功能。

2.2.1 导入模板工程

如图 2-16 所示，选择菜单栏 File 中的 Open Projects from File System，导入模板工程 `eclipse_proj\demo\gpio`。

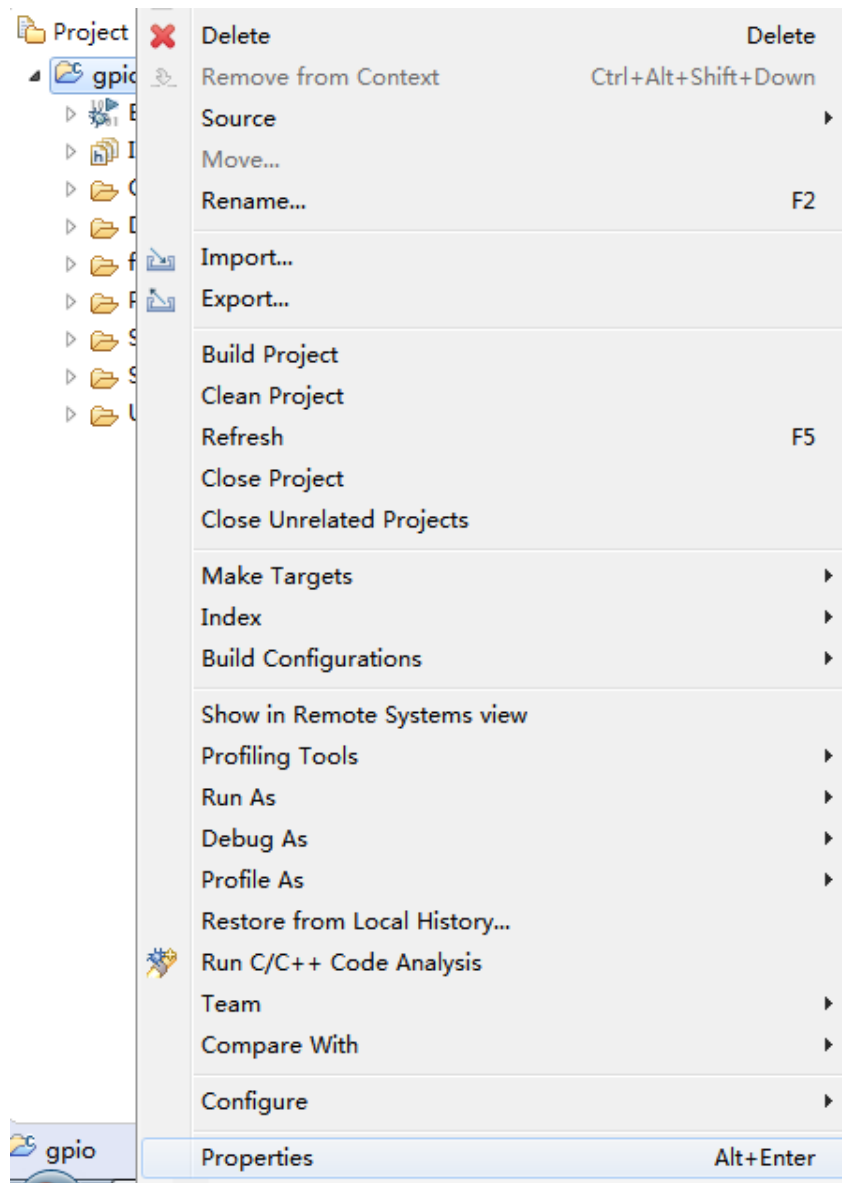
图 2-16 导入模板工程



2.2.2 选择 Properties

导入模板工程后，选择模板工程 gpio，右键选择 Properties，如图 2-17 所示。

图 2-17 选择 Properties



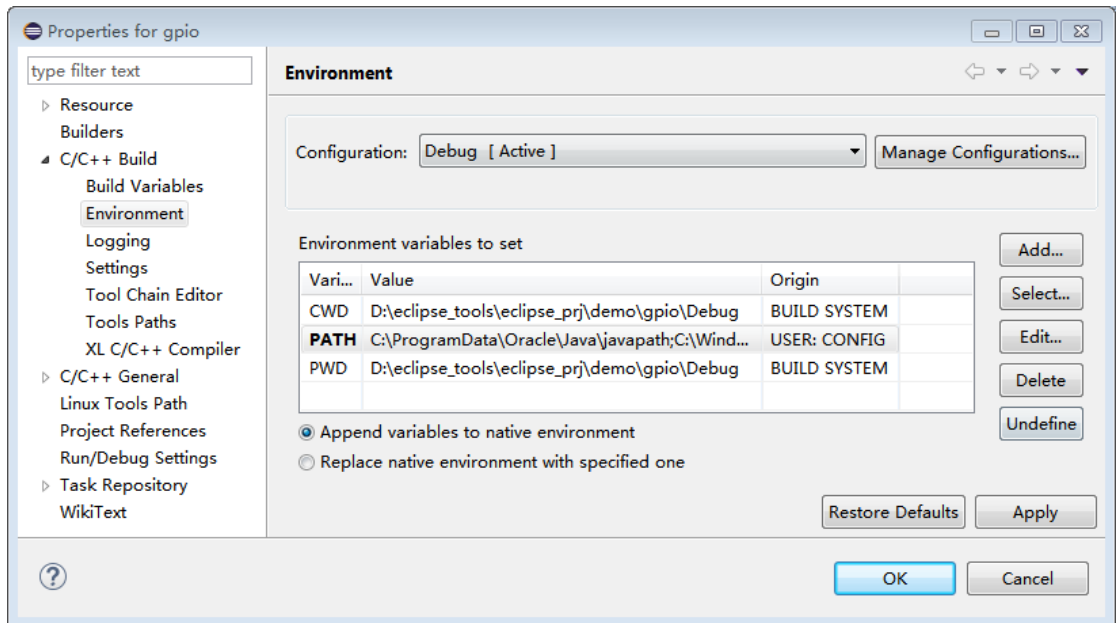
2.2.3 配置环境变量

选择 C/C++ Build 选项卡中 Environment 选项，如图 2-18 所示，配置如下环境变量：

- C:\ProgramData\Oracle\Java\javapath;
- C:\Windows\system32;C:\Windows;
- C:\Windows\System32\Wbem;
- C:\Windows\System32\WindowsPowerShell\v1.0\;
- D:\eclipse_tools\tools\GNU Tools ARM Embedded\5.4 2016q2\bin;

- D:\eclipse_tools\tools\GNU MCU Eclipse\Build Tools\2.11-20180428-1604\bin

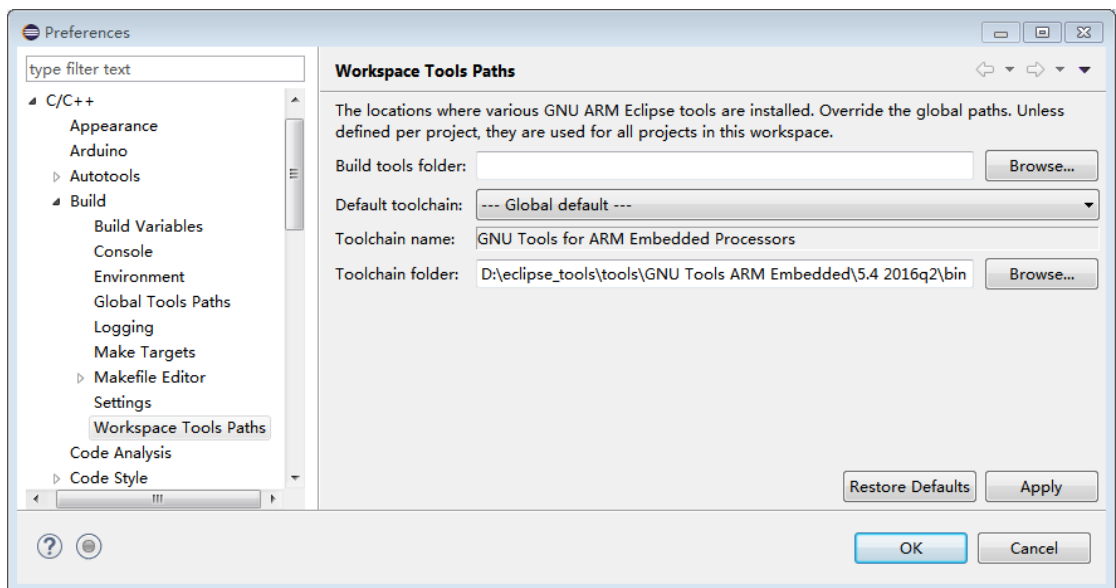
图 2-18 配置环境变量



2.2.4 配置工具链

选择菜单栏 Window 中的 Preferences, 选择选项卡 C/C++ 中的 Build 下的 Workspace Tools Paths, 配置工具链路径, 如图 2-19 所示。

图 2-19 配置工具链

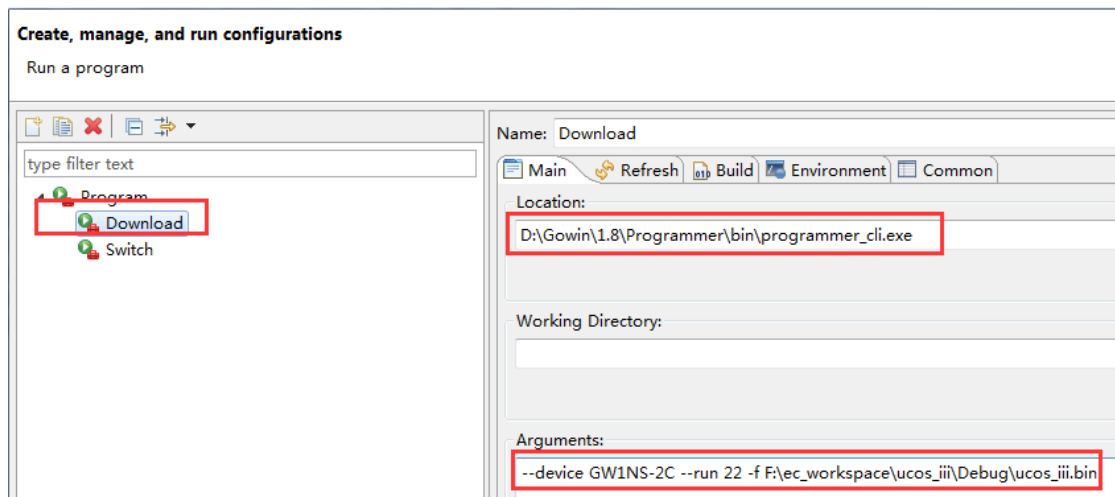


2.2.5 配置下载工具

GoWin GNU MCU Eclipse 软件使用 Programmer 工具下载 MCU 二进制 BIN 文件。

在 GoWin GNU MCU Eclipse 软件中配置 Programmer 工具为 MCU 二进制 BIN 文件下载工具，选择菜单栏 Run 中 External Tools 下的 External Tools Configurations...，配置下载工具，如图 2-20 所示。

图 2-20 配置烧录工具



2.2.6 配置 JTAG 模式切换工具

如果用户需要调试项目工程，需要将 JTAG 模式由下载模式切换为调试模式。

在 GoWin GNU MCU Eclipse 软件中配置 Programmer 命令行工具为 JTAG 模式切换工具，选择菜单栏 Run 中 External Tools 下的 External Tools Configurations...，配置切换工具，如图 2-21 所示。

图 2-21 配置 JTAG 模式切换工具

Create, manage, and run configurations

External tool location does not exist

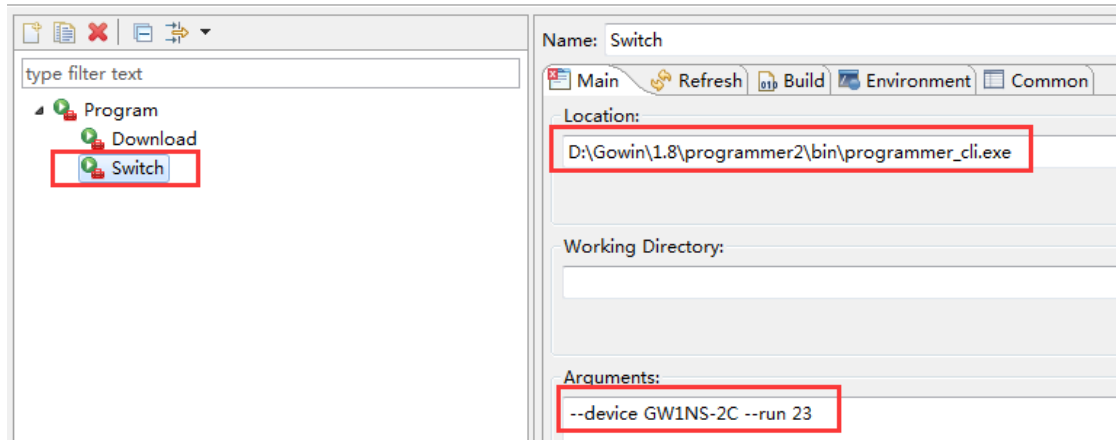
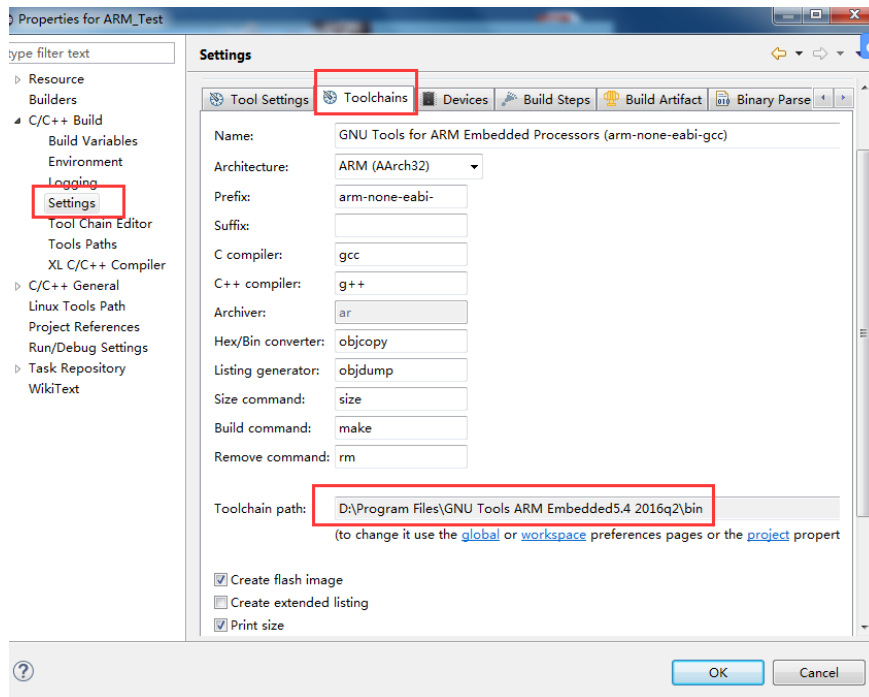


图 2-22 交叉编译配置



2.3 工程模板

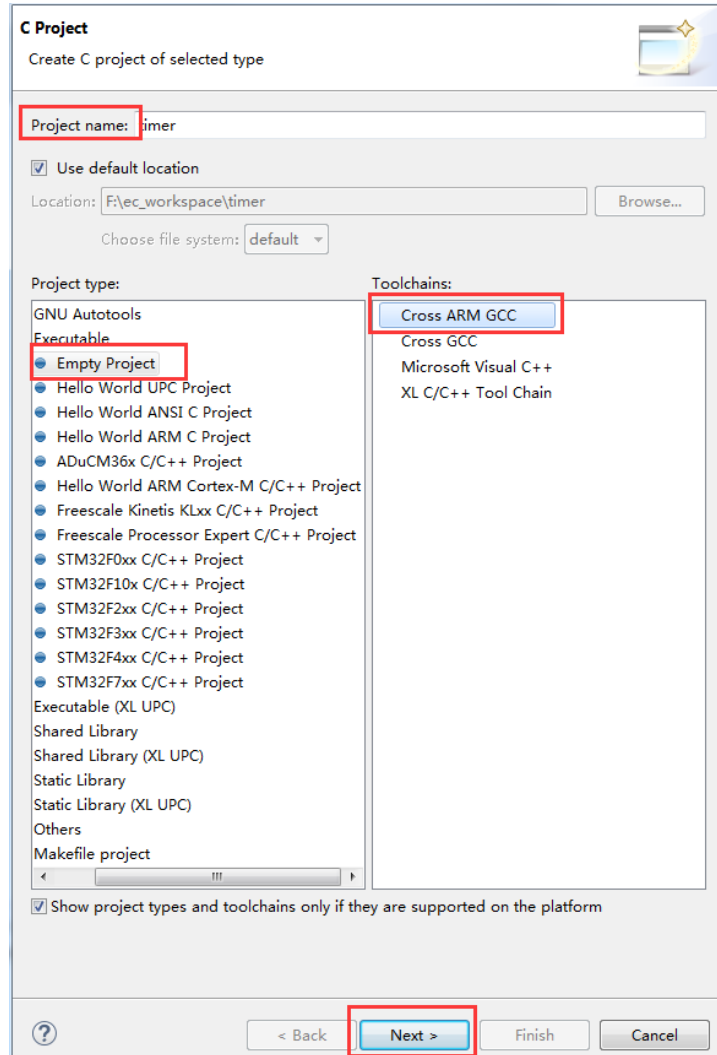
2.3.1 工程创建

新建工程

选择菜单栏 File 中的 New 下的 C Project，如图 2-23 所示。

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

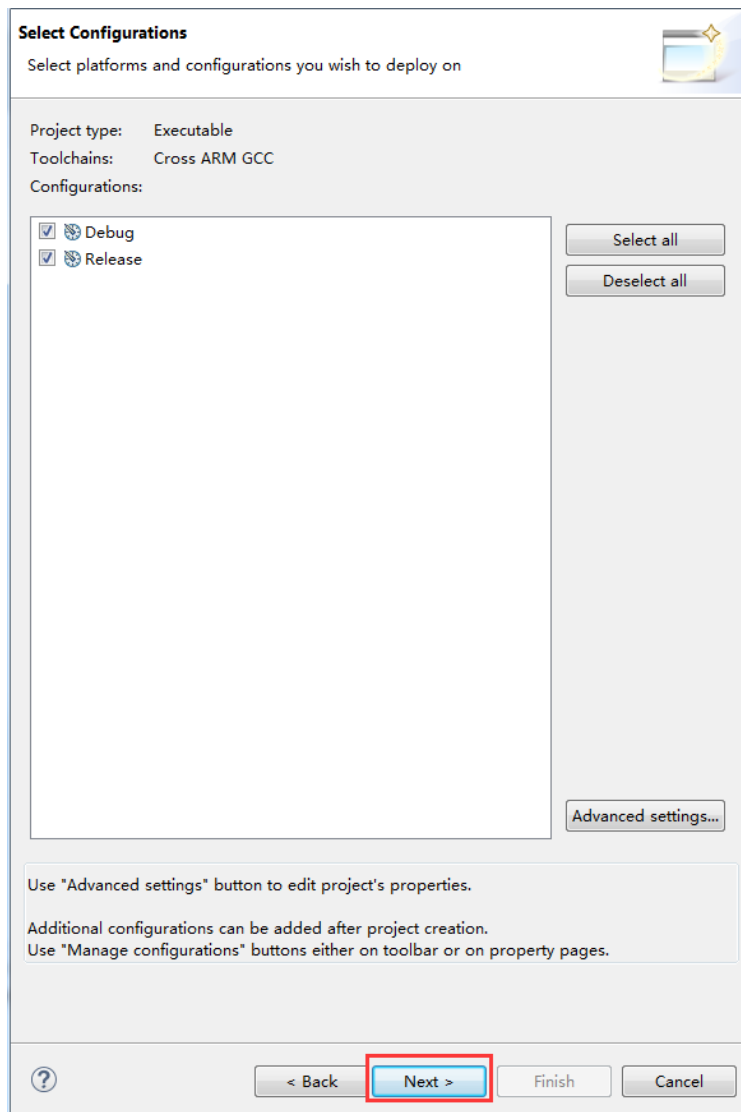
图 2-23 新建工程



选择配置

选择工程配置类型 Debug 和 Release，如图 2-24 所示。

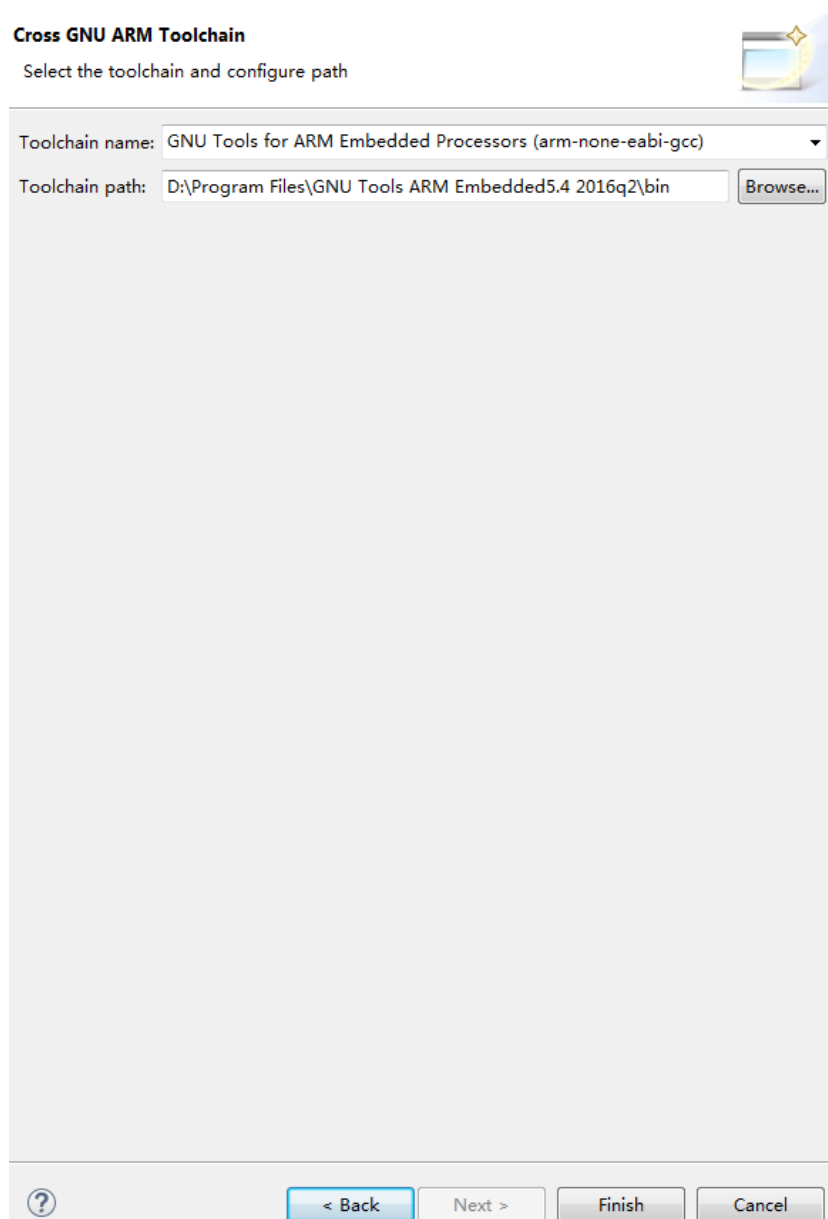
图 2-24 选择工程类型配置



选择工具链和路径

选择交叉编译工具链 `arm-none-eabi-gcc`，如图 2-25 所示。

图 2-25 选择工具链



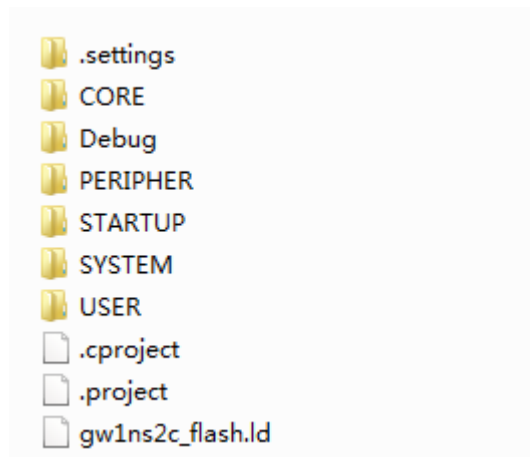
建立工程结构

完成新建工程后，选择 Eclipse 工作空间下新建的项目工程，添加工程结构和代码，如图 2-26 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 2-26 工程结构



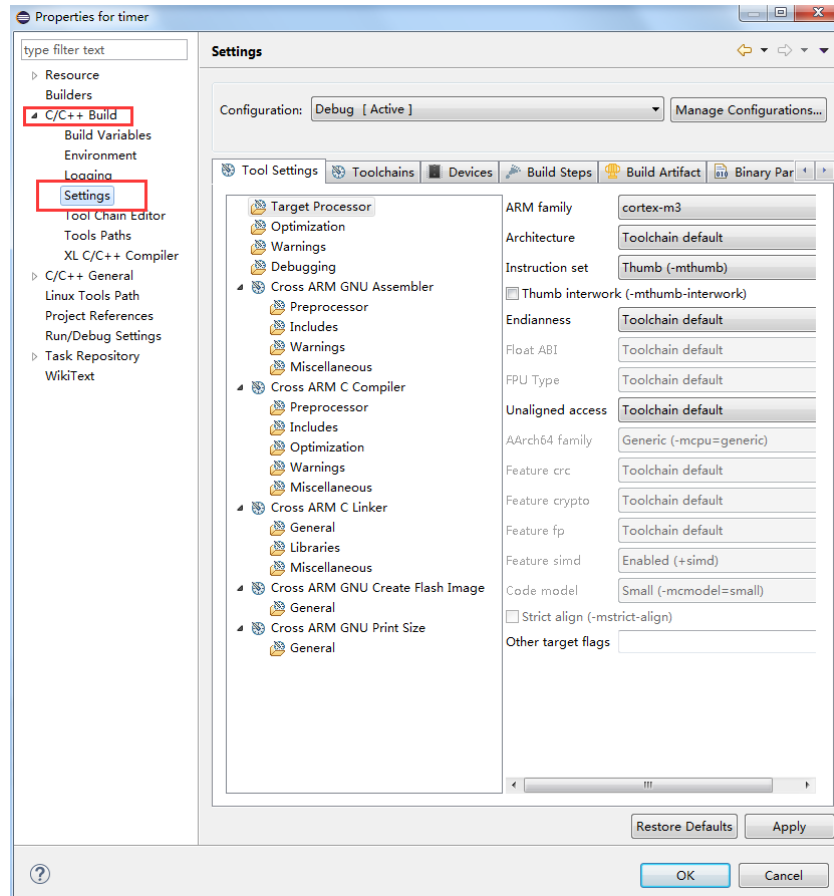
导入文件

Eclipse 软件中, 选择当前工程, 右键选择 Refresh, 自动导入如图 2-24 所示的工程结构和代码。

2.3.2 工程创建

Eclipse 软件中, 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 2-27 所示。

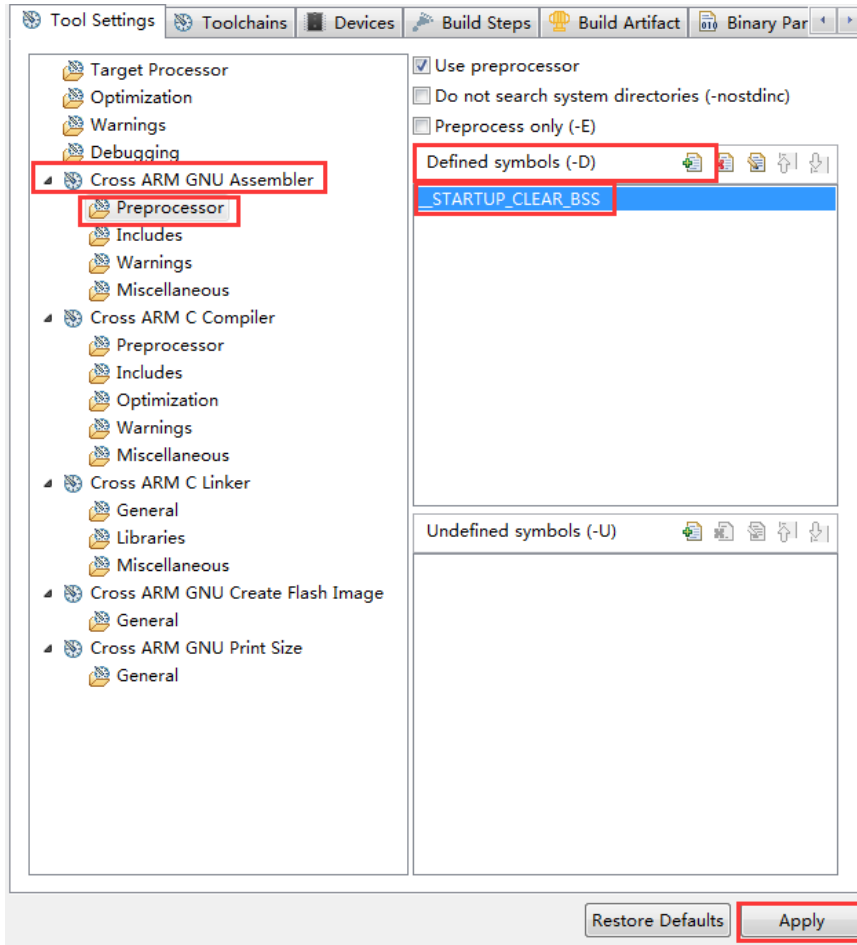
图 2-27 选择 Properties



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 2-28 所示。

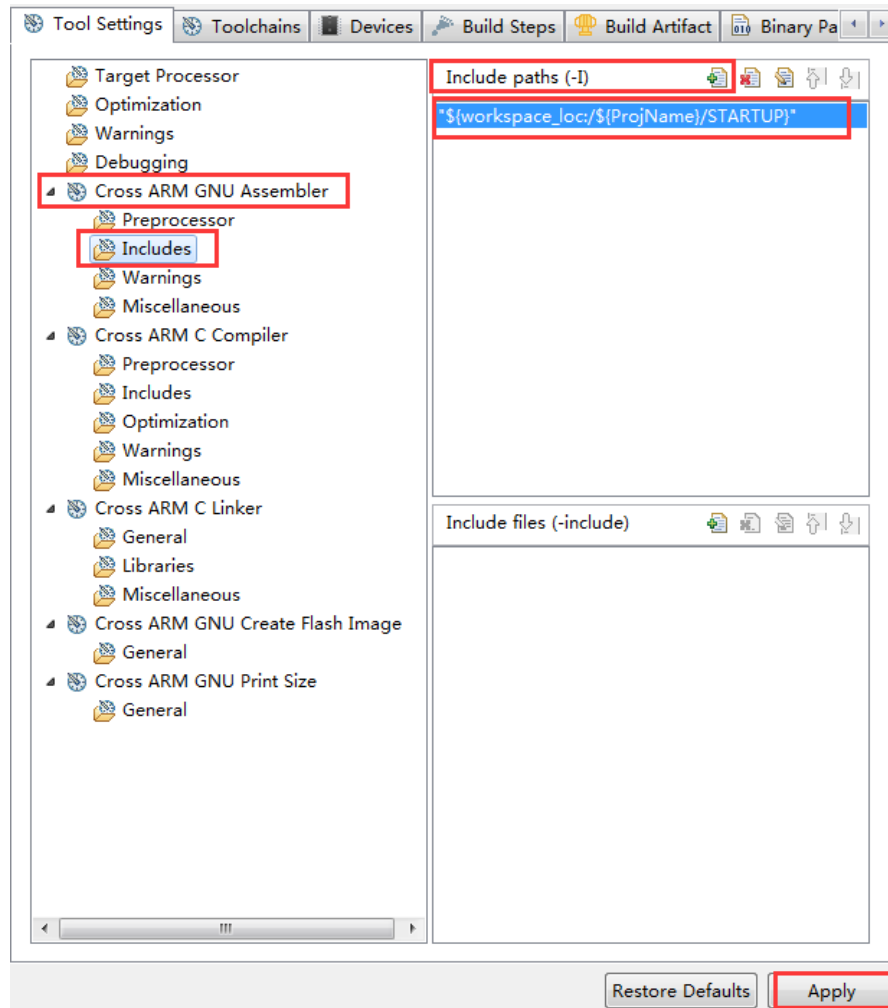
图 2-28 配置 Cross ARM GNU Assembler Preprocessor



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 2-29 所示。

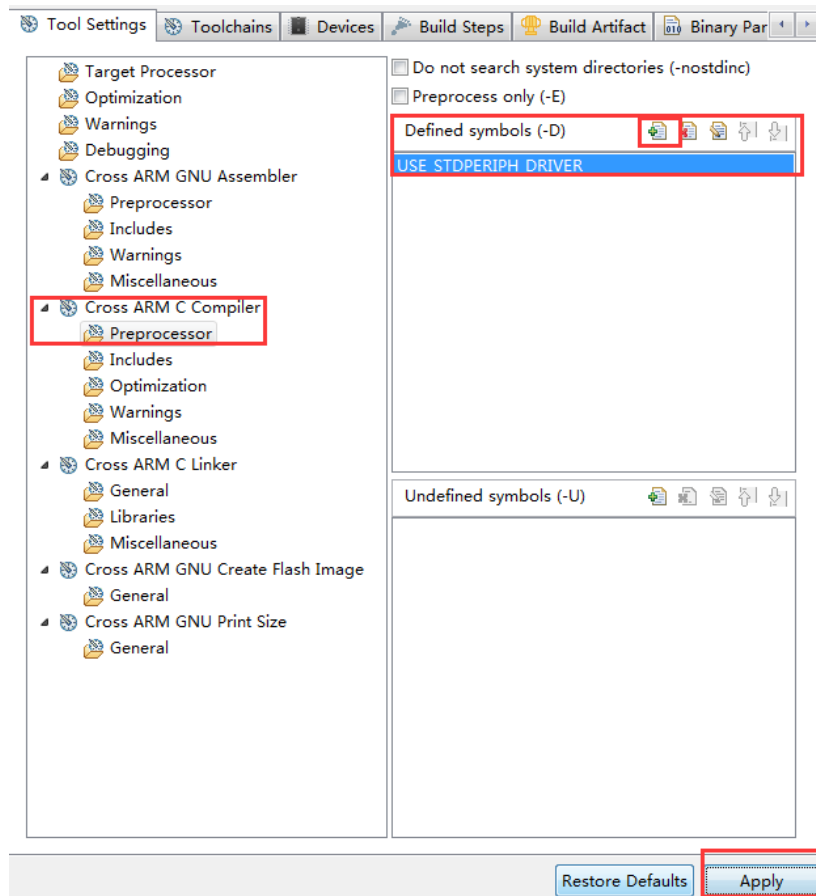
图 2-29 配置 Cross ARM GNU Assembler Includes



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 2-30 所示。

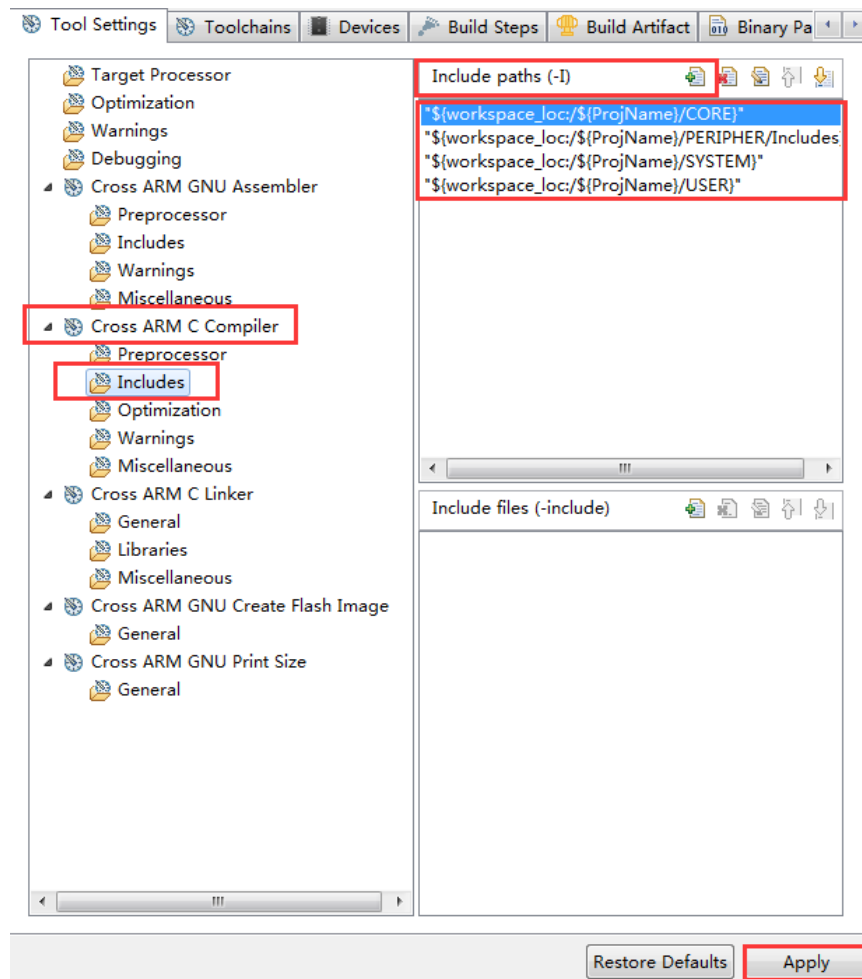
图 2-30 配置 Cross ARM C Compiler Preprocessor



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 2-31 所示。

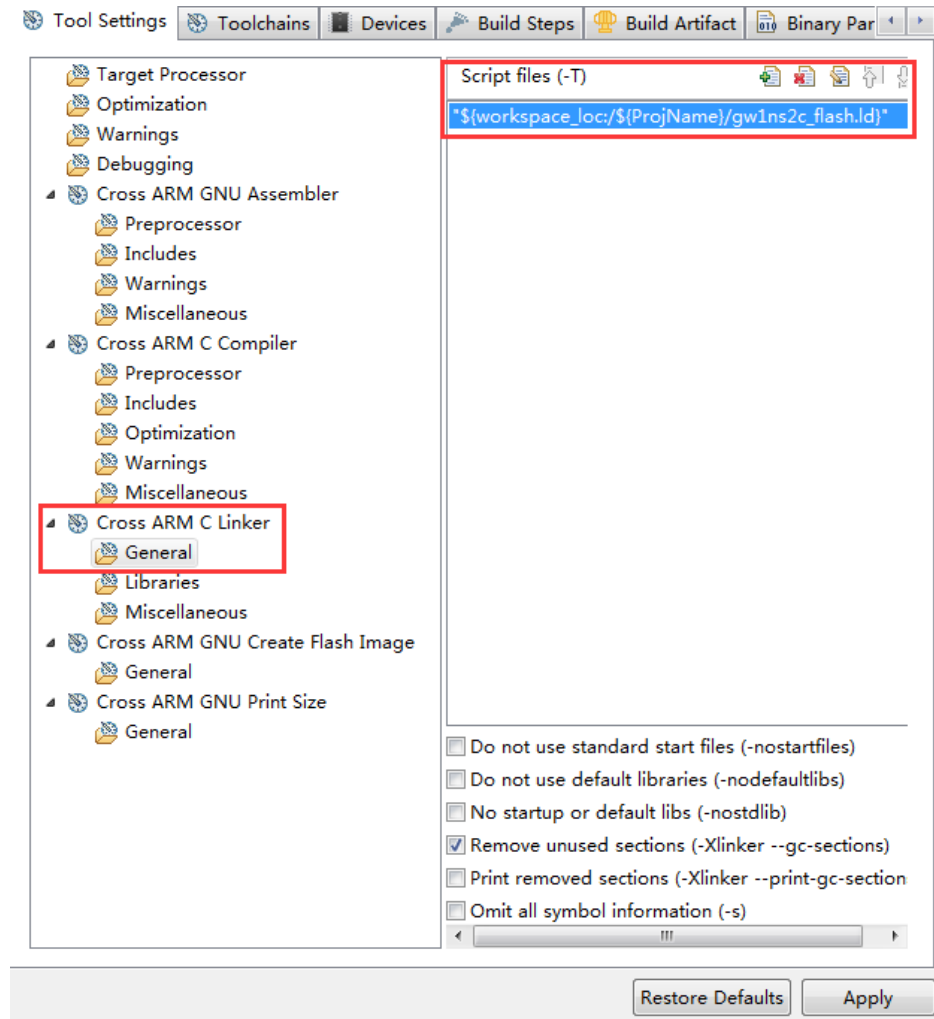
图 2-31 配置 Cross ARM C Compiler Includes



配置 Cross ARM C Linker

选择 Cross ARM C Linker 中的 General，配置 Flash 链接脚本，如图 2-32 所示。

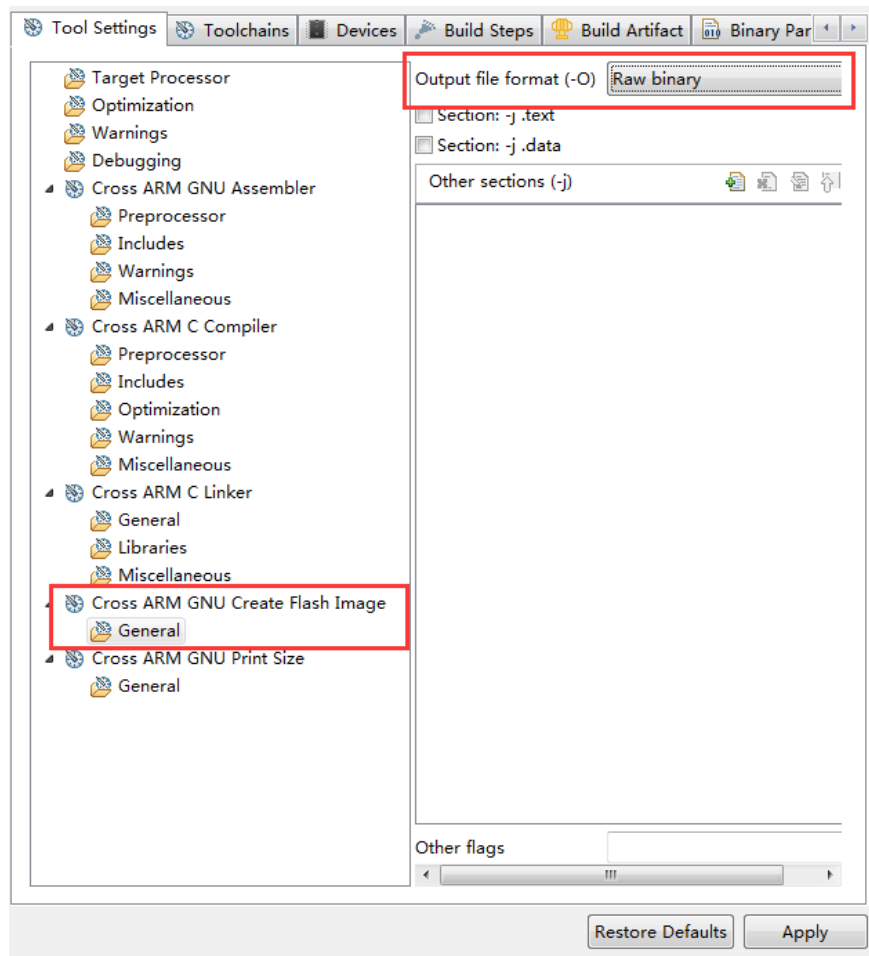
图 2-32 配置 Cross ARM C Linker



配置 Cross ARM GNU Create Flash Image

选择 Cross ARM GNU Create Flash Image 中的 General，配置 MCU 镜像文件格式为 RAW binary，如图 2-33 所示。

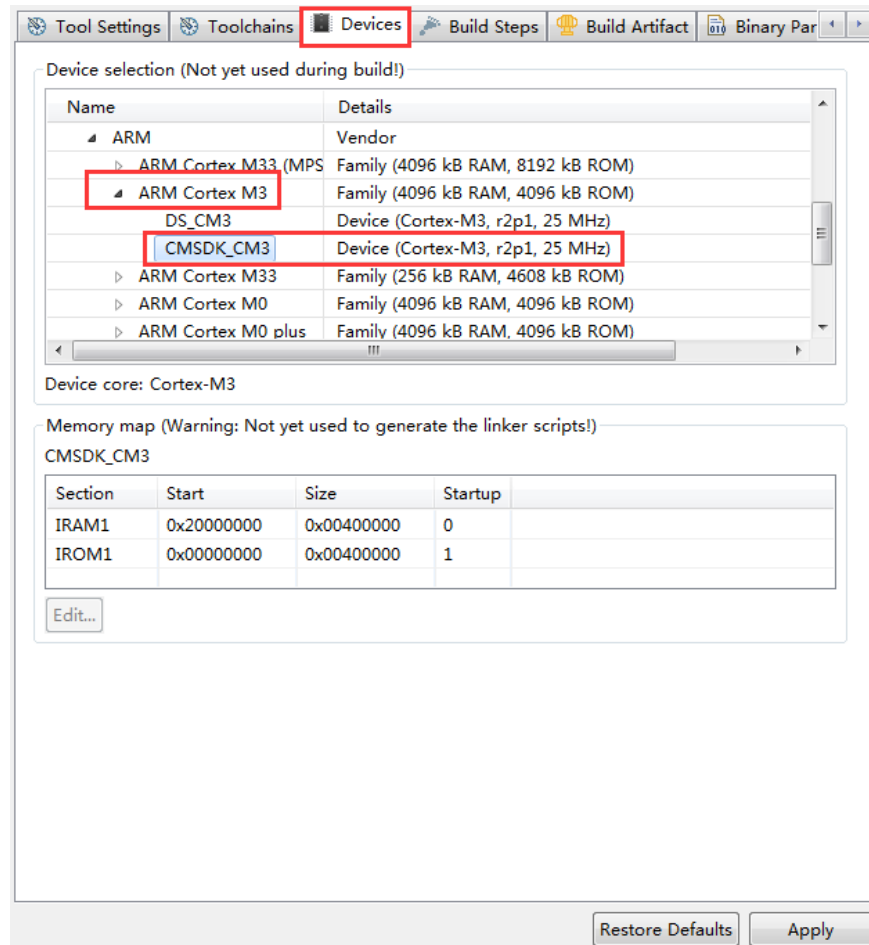
图 2-33 配置 Cross ARM GNU Create Flash Image



配置 Devices

选择 Devices 选项卡，选择器件 ARM Cortex-M3 “CMSDK_CM3”，如图 2-34 所示。

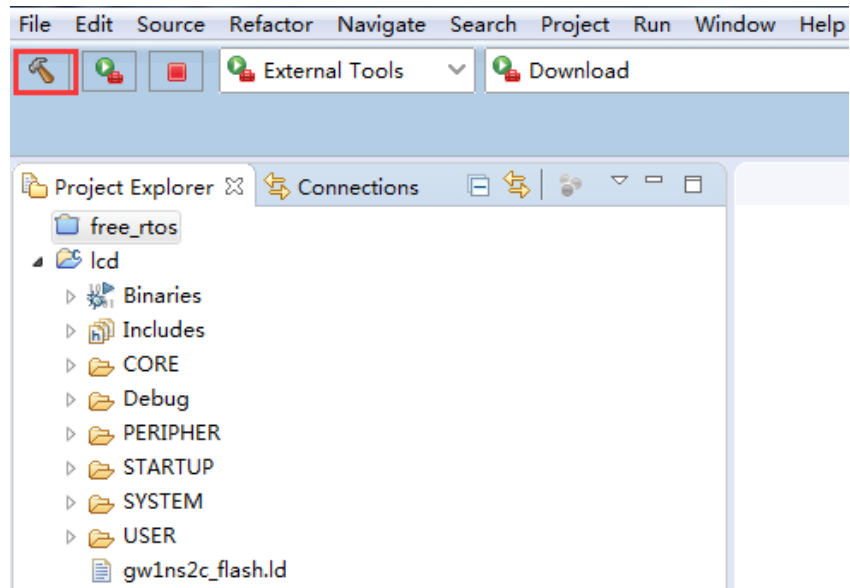
图 2-34 配置 Devices



2.3.3 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 2-35 所示。

图 2-35 编译工程



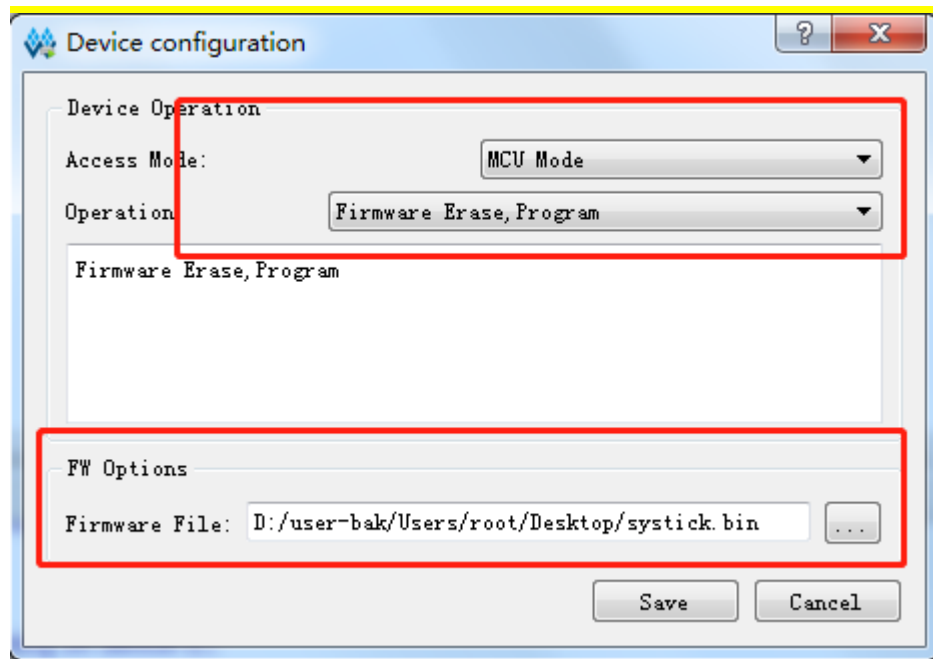
2.3.4 工程下载

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 GNU MCU Eclipse 工具栏下载按钮，调用已经配置的下下载工具 Download，下载 MCU 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 MCU 二进制 BIN 文件，下载 MCU 二进制 BIN 文件，如图 2-36 所示。

图 2-36 工程下载

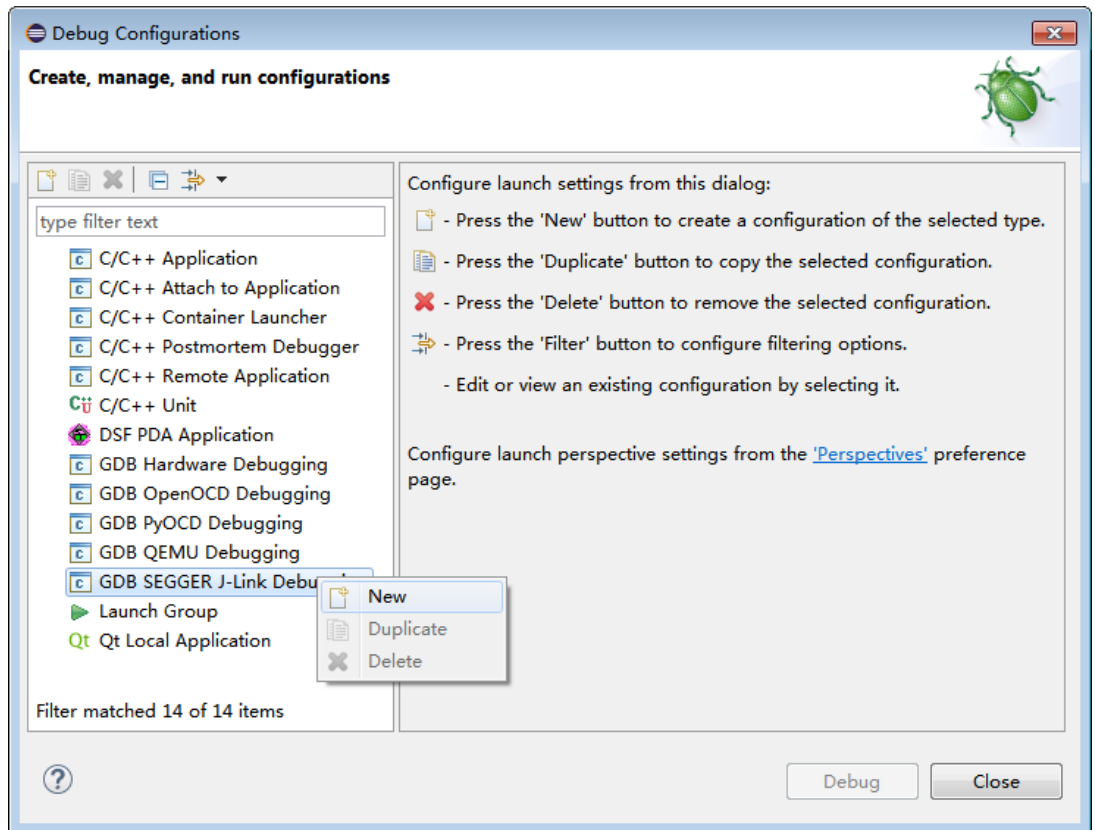


2.3.5 工程调试

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 2-37 所示。

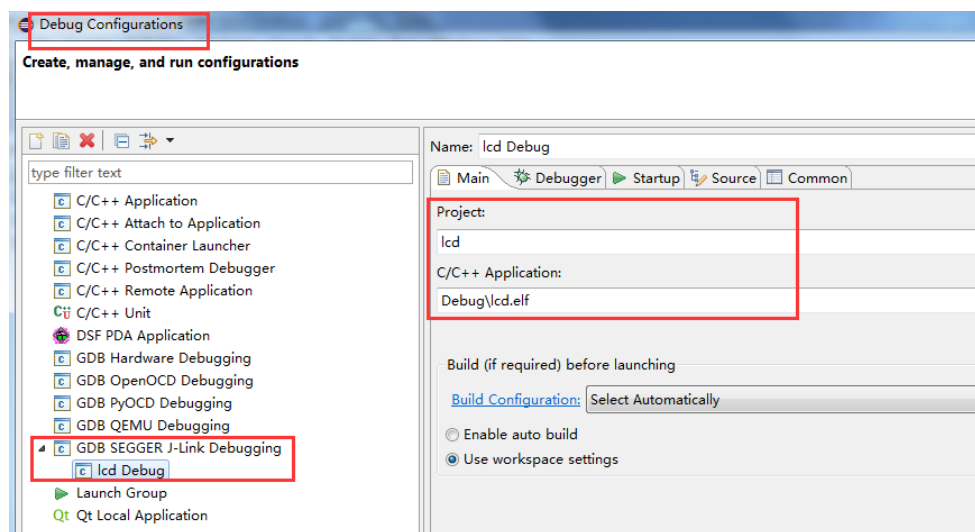
图 2-37 建立调试配置选项



选择已建立的调试配置选项。

1. 选择 Main 选项卡，配置镜像文件路径，如图 2-38 所示。

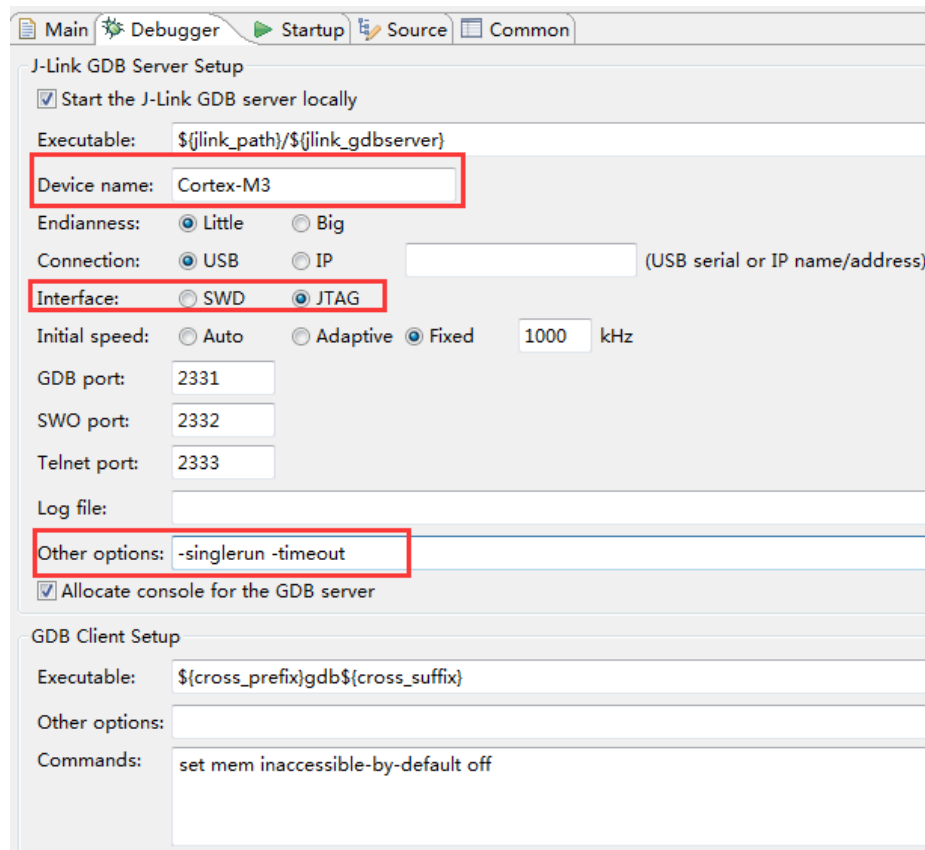
图 2-38 配置 Image 路径



2. 选择 Debugger 选项卡, 配置 Debugger 器件名称和调试接口为 JTAG, 如图 2-39 所示。

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

图 2-39 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 Eclipse 软件配置中已配置的 JTAG 模式切换工具 Switch, 完成 JTAG 软件模式由下载模式切换到调试模式。

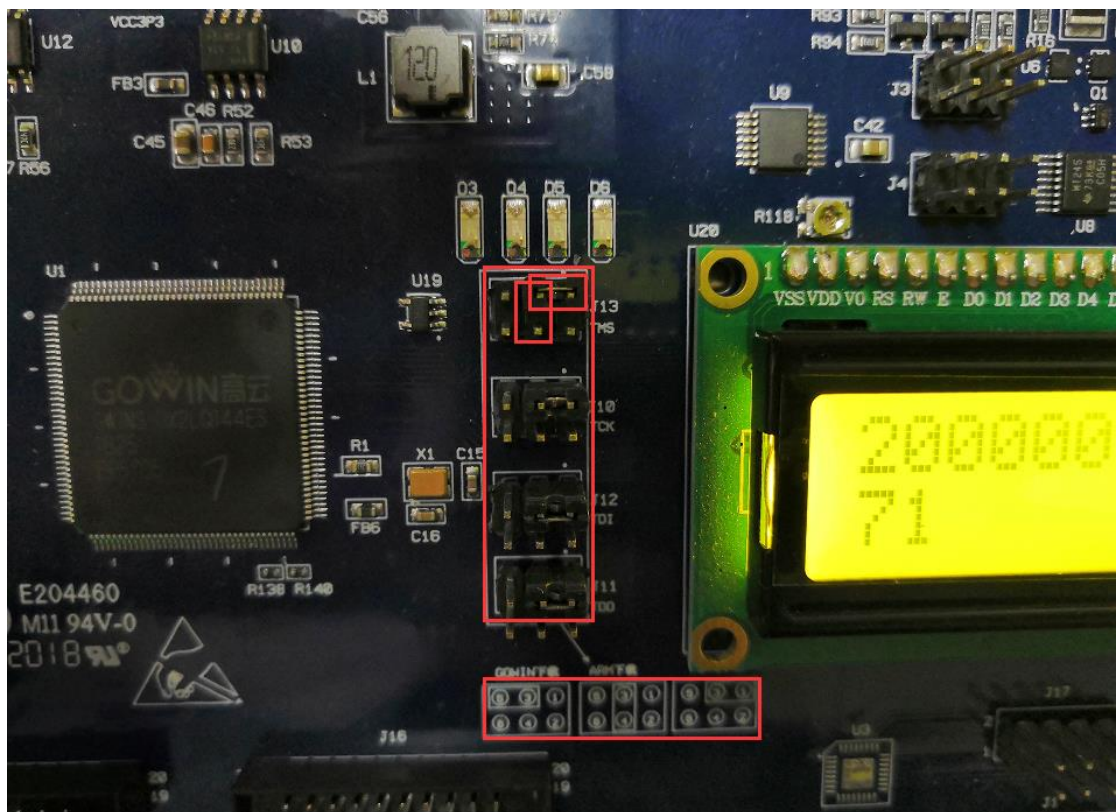
第二种方法: 打开 Programmer, 配置器件访问模式为 MCU Mode, 操作模式为 Connect to JTAG of MCU, 完成 JTAG 软件模式由下载模式切换到调试模式。

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 2-40 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。

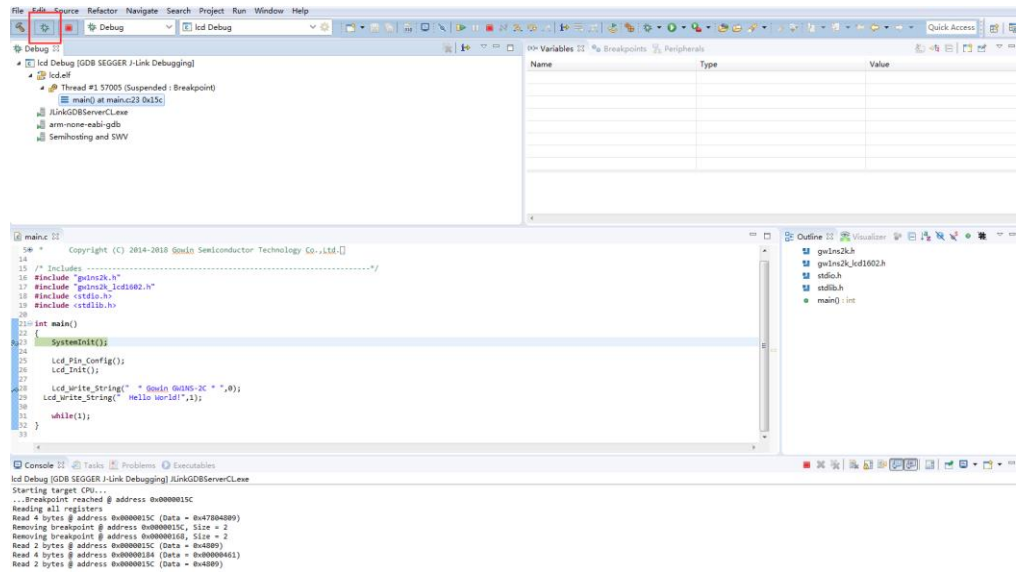
图 2-40 JTAG 接口切换



调试

连接 J-LINK 仿真器，选择工具栏 Debug 按钮，开始调试，如图 2-41 所示。

图 2-41 启动调试



2.4 参考设计

GW1NS-2C MCU 提供 GoWin GNU MCU Eclipse 软件环境的参考设计：

Gowin_EMPU_RefDesign\ MCU_RefDesign\GNU_RefDesign

