



GW1NS-2C MCU IDE

软件参考手册

RN519-1.0,2018-08-21

## **版权所有©2018 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
2018/08/21	1.0	初始版本。

# 目录

目录 .....	i
图目录 .....	iii
<b>1 ARM KEIL 软件 .....</b>	<b>1</b>
1.1 ARM KEIL 下载与安装 .....	1
1.2 ARM KEIL 配置 .....	1
1.2.1 配置选项 .....	1
1.2.2 编译 .....	5
1.2.3 下载 .....	6
1.2.4 调试 .....	6
<b>2 GNU MCU Eclipse 软件 .....</b>	<b>8</b>
2.1 GNU MCU Eclipse 下载与安装 .....	8
2.1.1 软件与插件 .....	8
2.1.2 安装 Java 虚拟机 .....	8
2.1.3 安装 Eclipse 软件 .....	11
2.1.4 安装 Eclipse CDT 插件 .....	12
2.1.5 安装 Eclipse GNU ARM 插件 .....	13
2.1.6 安装交叉编译链 .....	13
2.1.7 安装 make .....	14
2.2 GNU MCU Eclipse 配置 .....	15
2.2.1 配置交叉编译链 .....	15
2.2.2 配置宏定义 .....	16
2.2.3 配置头文件路径 .....	17
2.2.4 配置 Flash 链接 .....	18
2.2.5 配置交叉编译链路径 .....	19
2.3 GNU MCU Eclipse 下载配置 .....	20
2.4 GNU MCU Eclipse 调试配置 .....	20
2.4.1 安装 J-LINK 驱动 .....	20

2.4.2 下载 Device Pack 包 .....	22
2.4.3 设置芯片类型 .....	23
2.4.4 配置 Debug .....	24
2.5 配置 JTAG 模式切换.....	24

# 图目录

图 1-1 配置器件 .....	2
图 1-2 配置 ROM 和 RAM.....	2
图 1-3 配置输出文件格式.....	3
图 1-4 配置宏定义和头文件路径.....	3
图 1-5 配置下载选项 .....	4
图 1-6 配置调试选项 .....	4
图 1-7 调试接口类型 .....	5
图 1-8 编译 .....	5
图 1-9 下载 .....	6
图 1-10 JTAG 模式切换命令 .....	6
图 1-11 启动调试 .....	7
图 2-1 环境变量设置 .....	9
图 2-2 classpath 环境变量设置.....	9
图 2-3 JAVA_HOME 环境变量设置 .....	10
图 2-4 Java JDK 验证 .....	11
图 2-5 Eclipse 插件安装界面 .....	12
图 2-6 CDT 插件安装.....	13
图 2-7 GNU ARM 插件安装 .....	13
图 2-8 GNU ARM-GCC 环境变量设置.....	14
图 2-9 Eclipse 编译环境变量配置 .....	14
图 2-10 make 环境变量设置 .....	15
图 2-11 配置交叉编译链.....	16
图 2-12 配置宏定义 .....	17
图 2-13 配置头文件路径 .....	18
图 2-14 配置 Flash 链接 .....	19
图 2-15 交叉编译配置 .....	19
图 2-16 配置下载工具 .....	20

---

图 2-17 配置 J-Link 路径.....	21
图 2-18 配置 Device Pack 路径 .....	22
图 2-19 下载 Device Pack .....	23
图 2-20 配置芯片类型.....	23
图 2-21 配置 Image 路径 .....	24
图 2-22 配置 Debugger.....	24
图 2-23 JTAT 模式切换.....	25
图 2-24 启动调试 .....	25

# 1 ARM KEIL 软件

## 1.1 ARM KEIL 下载与安装

请参考 ARM 提供的《*MDK Getting Started*》和《*uVision User's Guide*》  
(可通过以下路径获取该手册: GW1NS-2C\_package\release\rt...)

## 1.2 ARM KEIL 配置

使用 ARM KEIL 软件进行嵌入式微处理器软件开发，在创建工程和开发过程中需要进行工程选项配置，GW1NS-2C 选项配置、编译和下载方法如下描述。

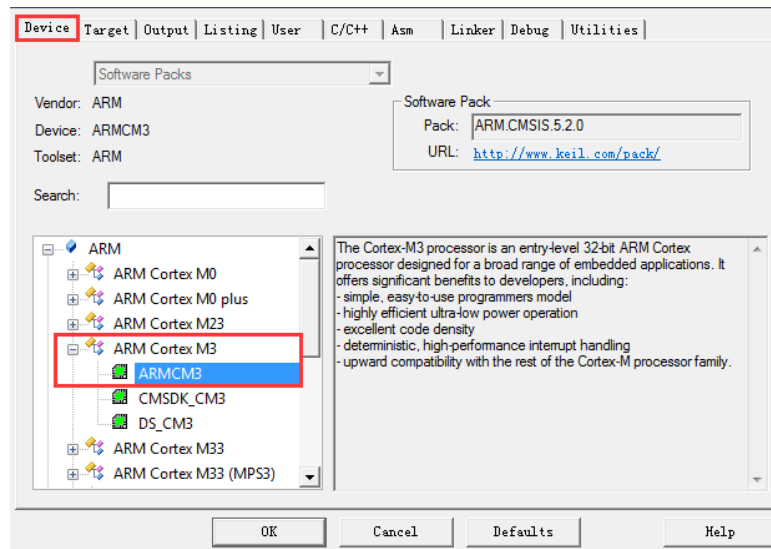
### 1.2.1 配置选项

#### 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 1-1 所示。



图 1-1 配置器件



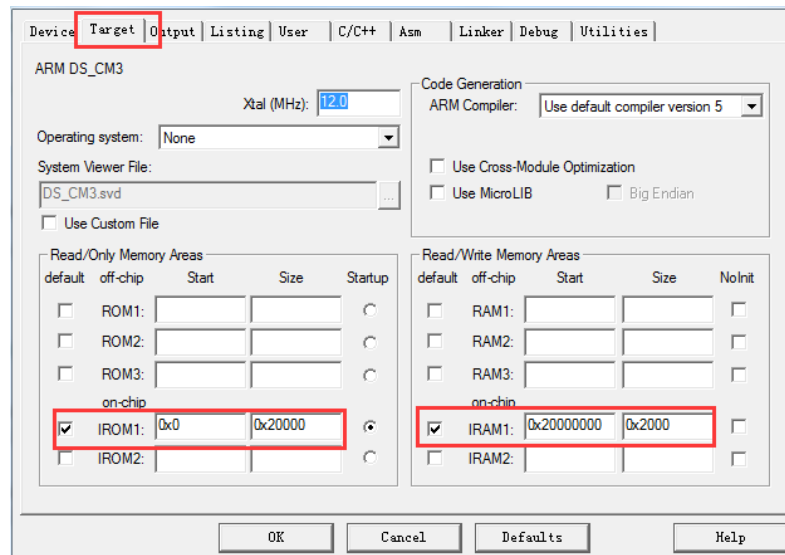
## 配置 ROM 和 RAM

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 8K Byte。配置如图 1-2 所示。

图 1-2 配置 ROM 和 RAM

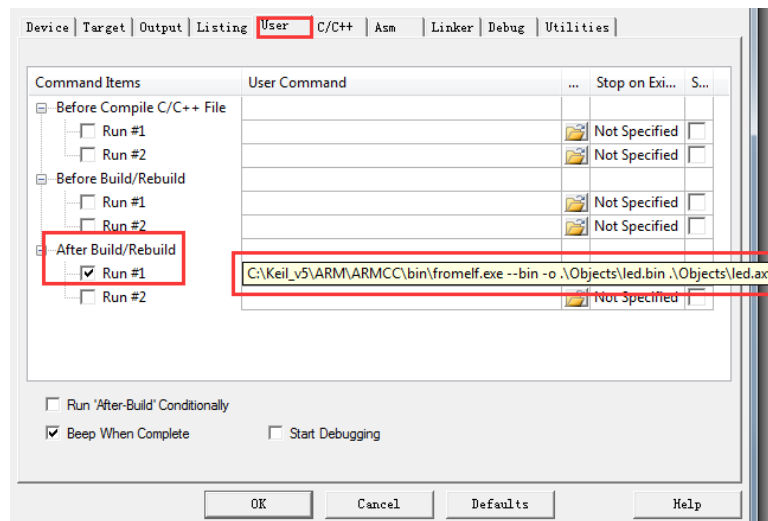


## 配置输出文件格式

Gowin Programmer 支持 BIN 二进制文件下载，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 1-3 所示。

图 1-3 配置输出文件格式



命令格式为:

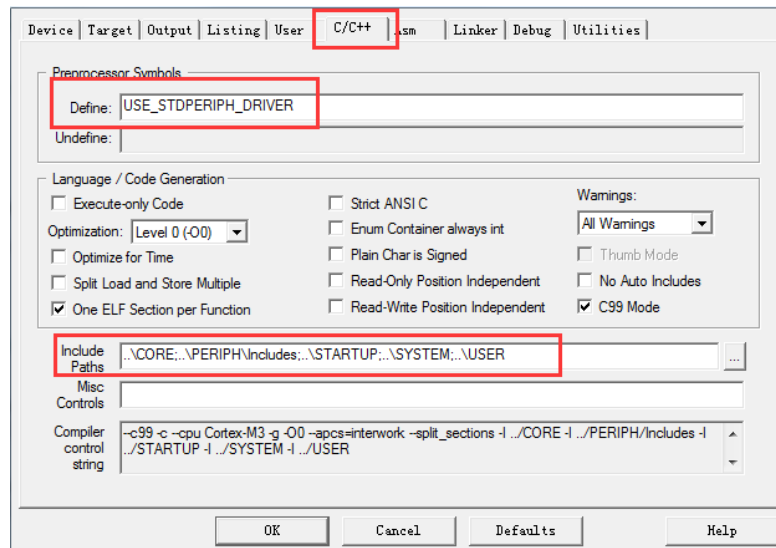
```
*:\Keil_v5\ARM\ARMCC\bin\fromelf.exe --bin
-o .\Objects/*.bin .\Objects/*.axf
```

### 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 1-4 所示。

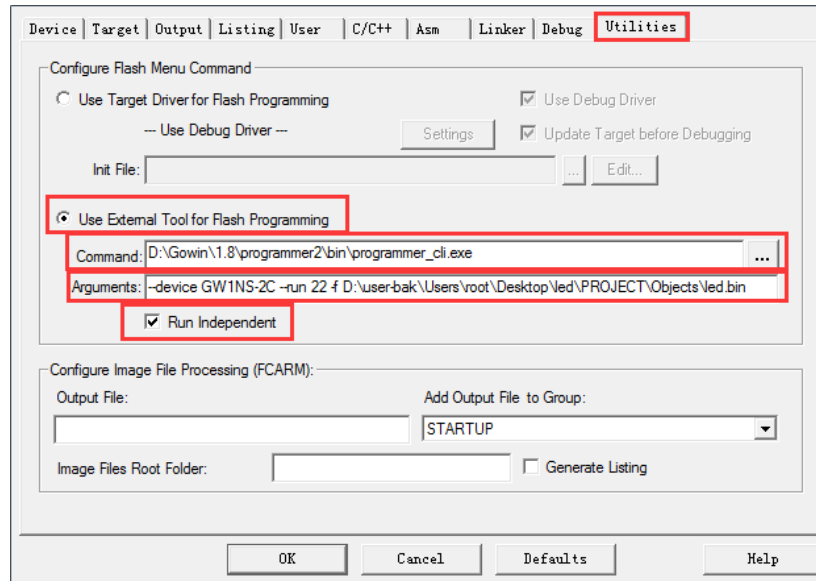
图 1-4 配置宏定义和头文件路径



### 配置下载选项

配置 Flash 下载工具为外部工具，使用 Gowin Programmer，如图 1-5 所示。

图 1-5 配置下载选项



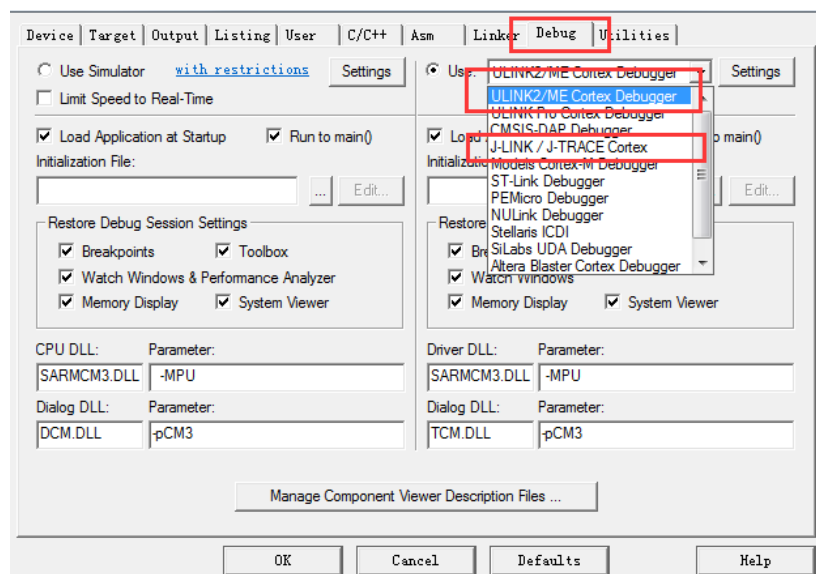
Command 为 Programmer 路径。

Arguments 为 Programmer 命令选项, 如--device GW1NS-2C --run 22 -f \*.bin。

### 配置调试选项

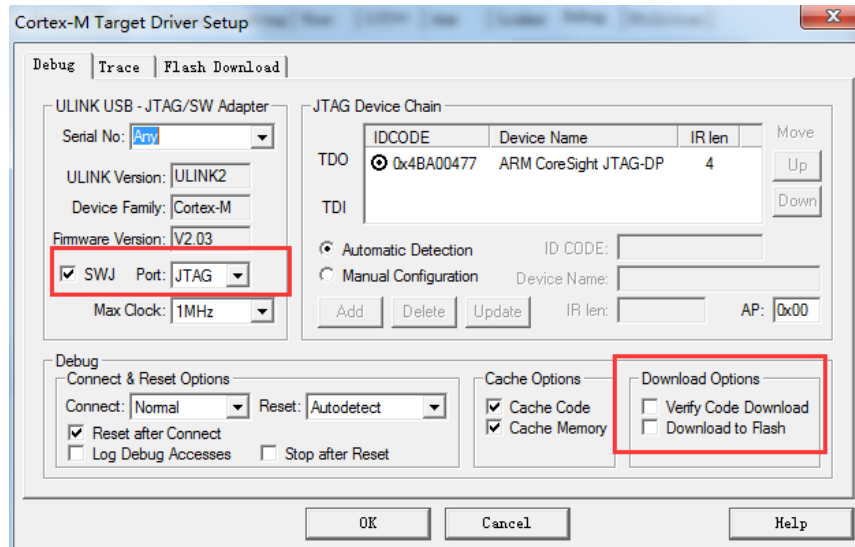
如果选择使用 U-LINK 仿真器, 则 Debug 选项配置为“ULINK2/ME Cortex Debugger”; 如果选择使用 J-LINK 仿真器, 则 Debug 选项配置为“J-LINK/J-TRACE Cortex”, 图 1-6 所示。

图 1-6 配置调试选项



调试接口类型配置为 JTAG, 如图 1-7 所示。

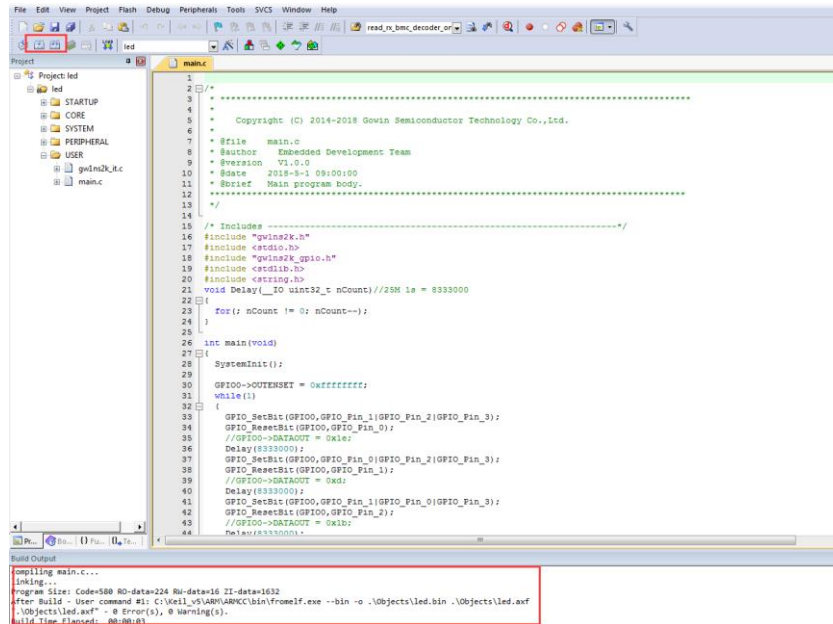
图 1-7 调试接口类型



## 1.2.2 编译

完成编码和配置后，编译生成 BIN 格式镜像文件，如图 1-8 所示。

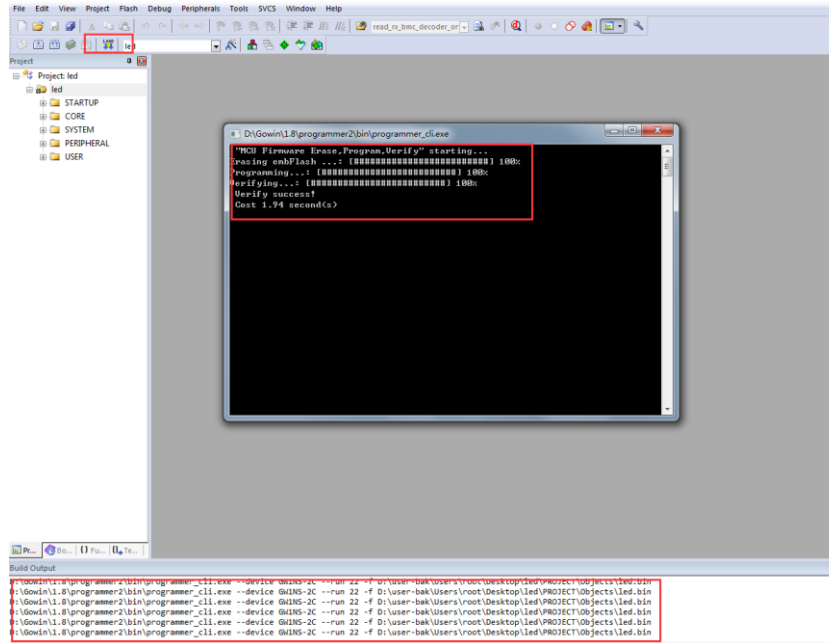
图 1-8 编译



## 1.2.3 下载

完成配置 Programmer 软件下载 BIN 文件到 Flash-Rom 后, 点击 “Download ” 完成下载, 如图 1-9 所示。

图 1-9 下载



## 1.2.4 调试

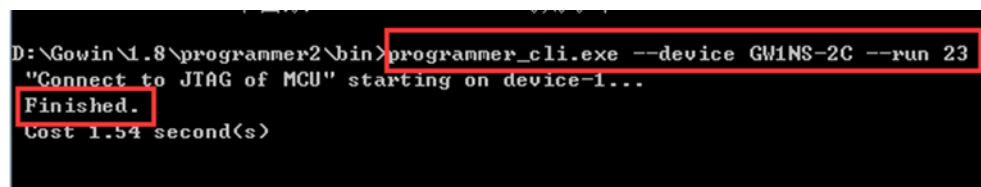
完成 ARM 镜像文件下载后, 如果用户设计出现问题, 可以连接 U-LINK 或 J-LINK 仿真器使用调试功能。

### JTAG 模式切换

如图 1-10 所示, 使用 Programmer 命令行手动将 JTAG 模式由下载模式切换到调试模式, 命令为:

```
programmer_cli.exe --device GW1NS-2C --run 23。
```

图 1-10 JTAG 模式切换命令



### JTAG 模式切换

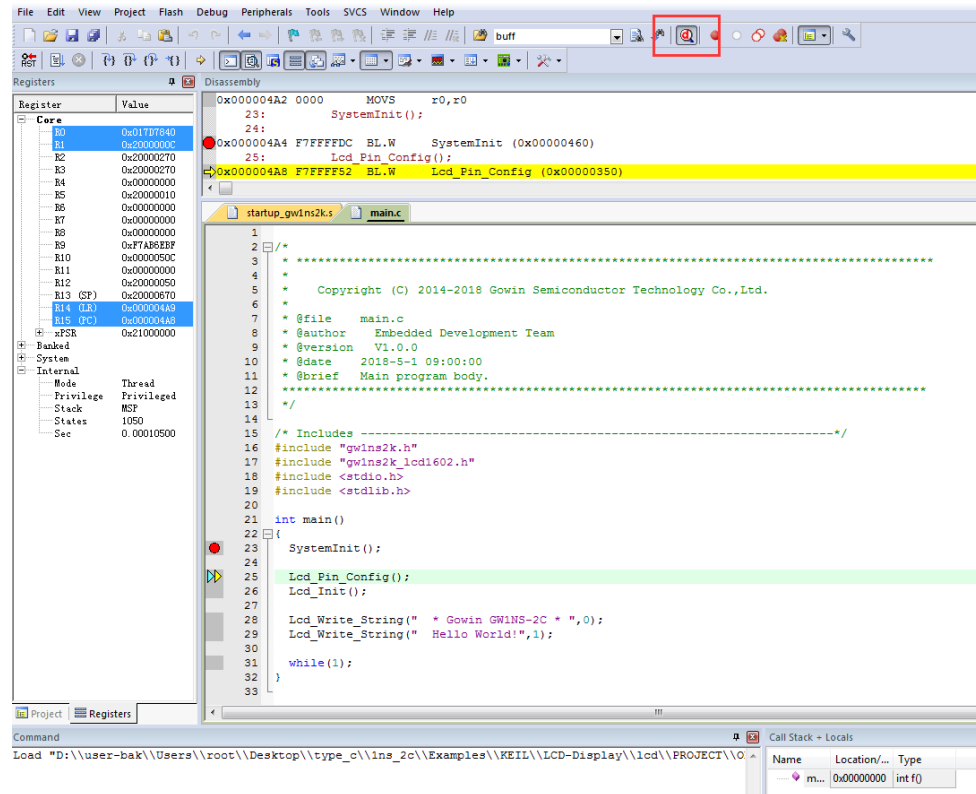
如果使用开发板 “DK-EVAL-GW1NS V1.1”, 需要手动将 JTAG 接口 TMS、

TCK、TDI 和 TDO 跳线帽由 FDTI 下载切换到 ARM 下载即，由” 3 和 1” 切换到” 3 和 4”。

## 启动调试

连接 U-LINK 或 J-LINK 仿真器，启动调试，如图 1-11 所示。

图 1-11 启动调试



# 2 GNU MCU Eclipse 软件

## 2.1 GNU MCU Eclipse 下载与安装

### 2.1.1 软件与插件

#### IDE 软件

开源 IDE 软件 Eclipse。

#### 插件

- Java JDK 虚拟机
- CDT
- GNU ARM

#### 编译器

开源交叉编译链 GNU ARM GCC 和 GNU Win32 make。

#### 调试器

开源调试器 GNU GDB。

### 2.1.2 安装 Java 虚拟机

#### 下载与安装

因 Eclipse 基于 Java，需下载 Java SE Development Kit，安装 Java 虚拟机。

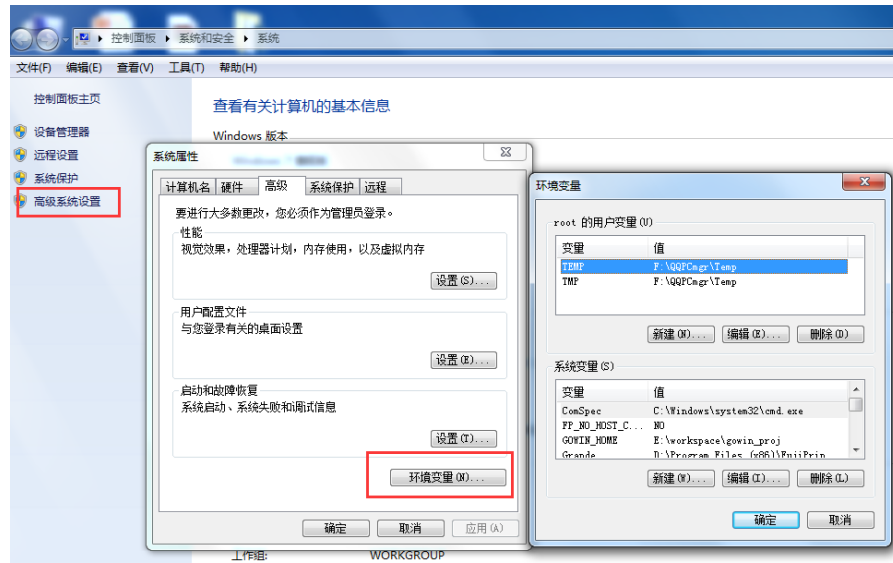
安装文件为” jdk-8u101-windows-x64.exe”。

请参考 Oracle [《Java Platform, Standard Edition Installation Guide》](#)。

#### 配置环境变量

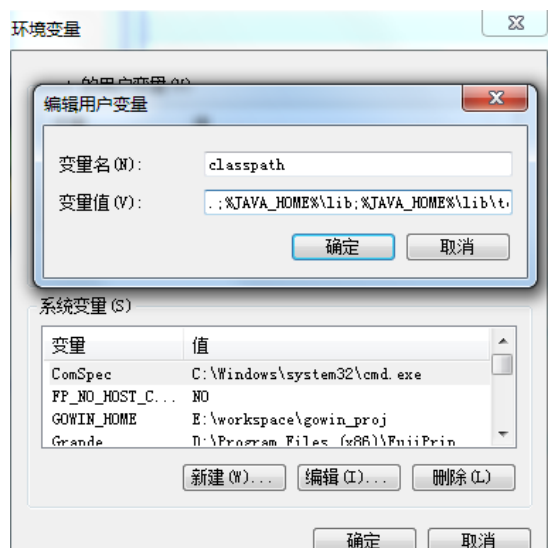
1. 右击我的电脑，选择属性，选择高级系统设置，选择环境变量，如图 2-1 所示。

图 2-1 环境变量设置



2. 新建变量名” classpath”，变量值为：  
” .;%JAVA\_HOME%\lib;%JAVE\_HOME%\lib\tools.jar”  
如图 2-2 所示。

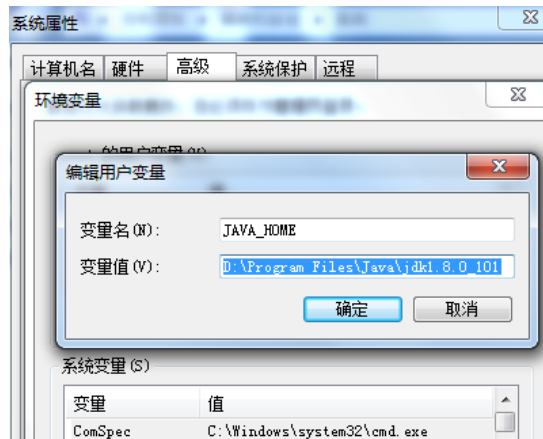
图 2-2 classpath 环境变量设置



3. 新建变量名”JAVE\_HOME”，变量值为安装的 Java JDK 的路径，如图 2-3 所示。



图 2-3 JAVA\_HOME 环境变量设置



4. 在已有的系统变量” path” 的变量值里添加：  
” .%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin”
5. 运行 cmd，检验是否配置成功，输入 java 命令，如果出现如下结果，则表明配置成功，如图 2-4 所示。

图 2-4 Java JDK 验证

```

C:\Users\root>java
用法: java [-options] class [args...]
           (执行类)
或 java [-options] -jar jarfile [args...]
           (执行 jar 文件)
其中选项包括:
  -d32          使用 32 位数据模型 (如果可用)
  -d64          使用 64 位数据模型 (如果可用)
  -server      选择 "server" VM
               默认 VM 是 server.

  -cp <目录和 zip/jar 文件的类搜索路径>
  -classpath <目录和 zip/jar 文件的类搜索路径>
              用 ; 分隔的目录, JAR 档案
              和 ZIP 档案列表, 用于搜索类文件。

  -D<名称>=<值>
              设置系统属性
  -verbose[:[class|gc|jni]]
              启用详细输出
  -version     输出产品版本并退出
  -version:<值>
              警告: 此功能已过时, 将在
              未来发行版中删除。
              需要指定的版本才能运行
  -showversion 输出产品版本并继续
  -jre-restrict-search | -no-jre-restrict-search
              警告: 此功能已过时, 将在
              未来发行版中删除。
              在版本搜索中包括/排除用户专用 JRE
  -? -help    输出此帮助消息
  -X          输出非标准选项的帮助
  -ea[:<packagename>...![:<classname>]]
  -enableassertions[:<packagename>...![:<classname>]]
              按指定的粒度启用断言
  -da[:<packagename>...![:<classname>]]
  -disableassertions[:<packagename>...![:<classname>]]
              禁用具有指定粒度的断言
  -esa | -enablesystemassertions
              启用系统断言
  -dsa | -disablesystemassertions
              禁用系统断言
  -agentlib:<libname>[=<选项>]
              加载本机代理库 (<libname>, 例如 -agentlib:hprof
              另请参阅 -agentlib:jdwp=help 和 -agentlib:hprof=help
  -agentpath:<pathname>[=<选项>]
              按完整路径名加载本机代理库
  -javaagent:<jarpath>[=<选项>]
              加载 Java 编程语言代理, 请参阅 java.lang.instrument
  -splash:<imagepath>
              使用指定的图像显示启动屏幕
有关详细信息, 请参阅 http://www.oracle.com/technetwork/java/javase/documentation/index.html。

```

## 2.1.3 安装 Eclipse 软件

Java 环境配置完成后, 下载和安装 C/C++ 的 Eclipse IDE。

选择下载 C/C++ 版本的 Eclipse, 解压后即可使用。

安装文件为” eclipse-cpp-neon-R-win32-x86\_64.zip”。

请参考 Eclipse [《Workbench User Guide》](#) 和 [《C/C++ Development User Guide》](#)。

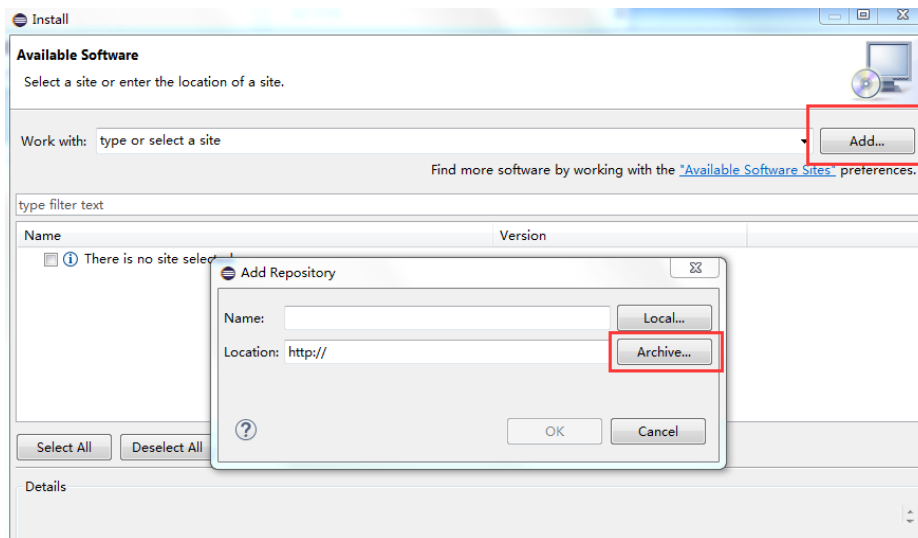
## 2.1.4 安装 Eclipse CDT 插件

Eclipse 安装完成后，下载 Eclipse CDT 插件。

安装文件为” cdt-9.0.1.zip”。

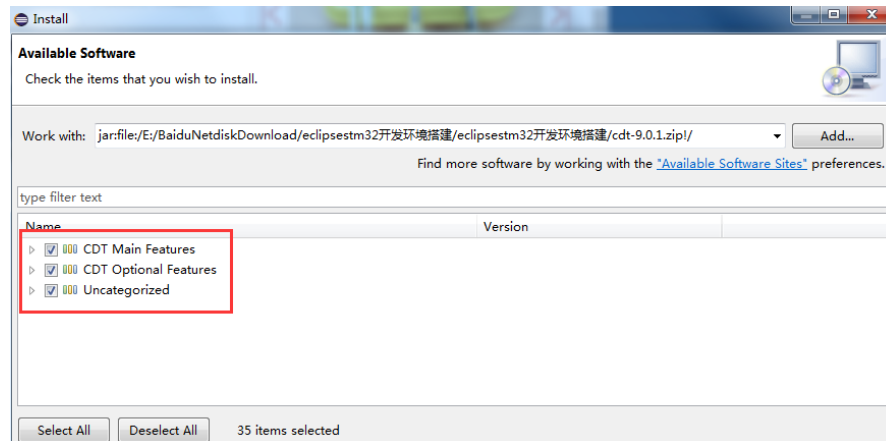
Eclipse 中插件安装界面如图 2-5 所示。

图 2-5 Eclipse 插件安装界面



Eclipse CDT 插件安装如图 2-6 所示。

图 2-6 CDT 插件安装



请参考 Eclipse [《CDT Plug-in Developer Guide》](#)。

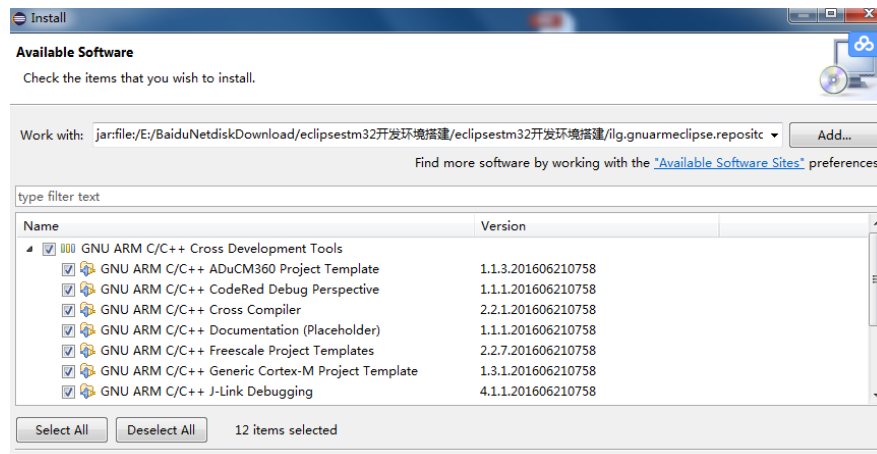
## 2.1.5 安装 Eclipse GNU ARM 插件

CDT 安装完成后，安装 GNU ARM Eclipse Plug-in。

安装文件为” ilg.gnuarmeclipse.repository-3.1.1-201606210758.zip”。

安装方式与 CDT 插件安装方式相同，如图 2-7 所示。

图 2-7 GNU ARM 插件安装



## 2.1.6 安装交叉编译链

### 下载与安装

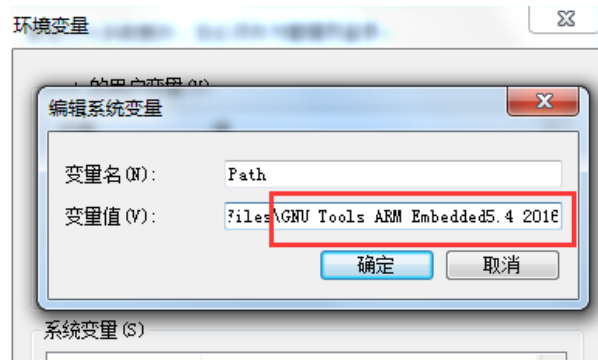
安装交叉编译工具链，针对 Cortex-M 采用 arm-none-eabi 版本。

安装文件为” gcc-arm-none-eabi-5\_4-2016q2-20160622-win32.exe”。

### 配置环境变量

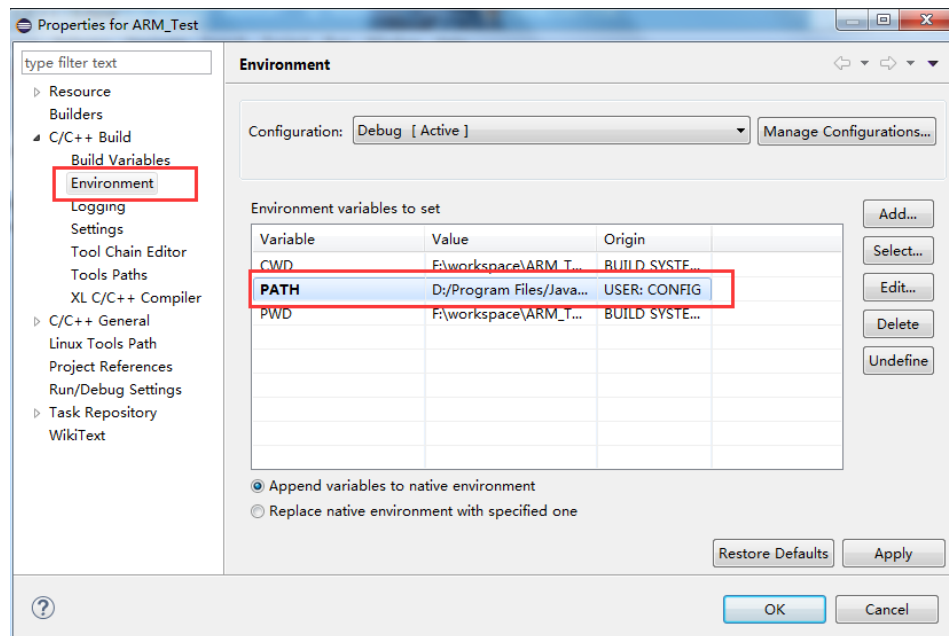
1. 在系统变量 Path 中，添加编译器的 bin 路径，如图 2-8 所示。

图 2-8 GNU ARM-GCC 环境变量设置



2. 在 Eclipse 中，添加编译器的路径，如图 2-9 所示。

图 2-9 Eclipse 编译环境变量配置



## 2.1.7 安装 make

### 下载与安装

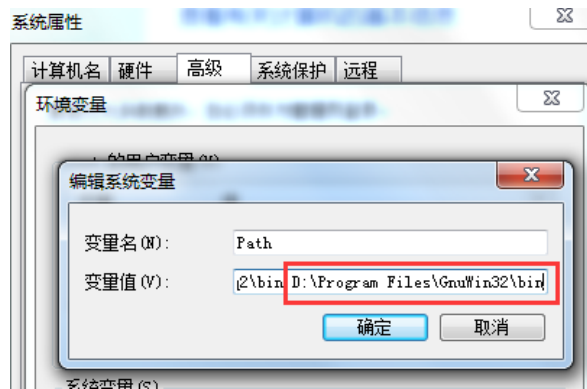
下载 GnuWin，文件为“sed-4.2.1-setup.exe”。

将 make 工具 make.exe 复制到 GnuWin 的 bin 路径下，并在系统环境变量 path 中添加 bin 路径。

### 配置环境变量

在系统变量 path 中，添加 make 的 bin 路径，如图 2-10 所示。

图 2-10 make 环境变量设置



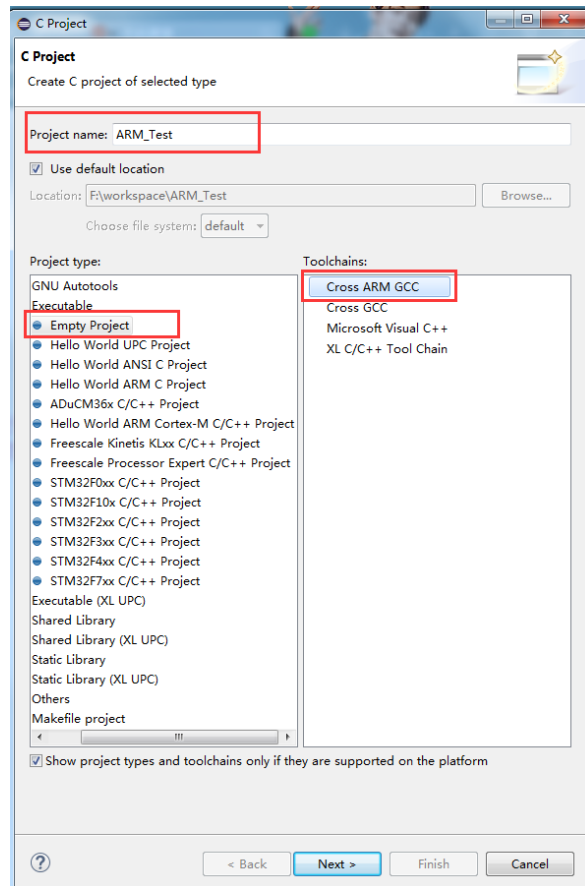
在 Eclipse 中，添加 make 的路径，如图 2-9 所示。

## 2.2 GNU MCU Eclipse 配置

### 2.2.1 配置交叉编译链

打开 Eclipse，选择 File->New->C Project，命名项目名称，选择工程类型，选择交叉编译工具” Cross ARM GCC”，如图 2-11 所示。

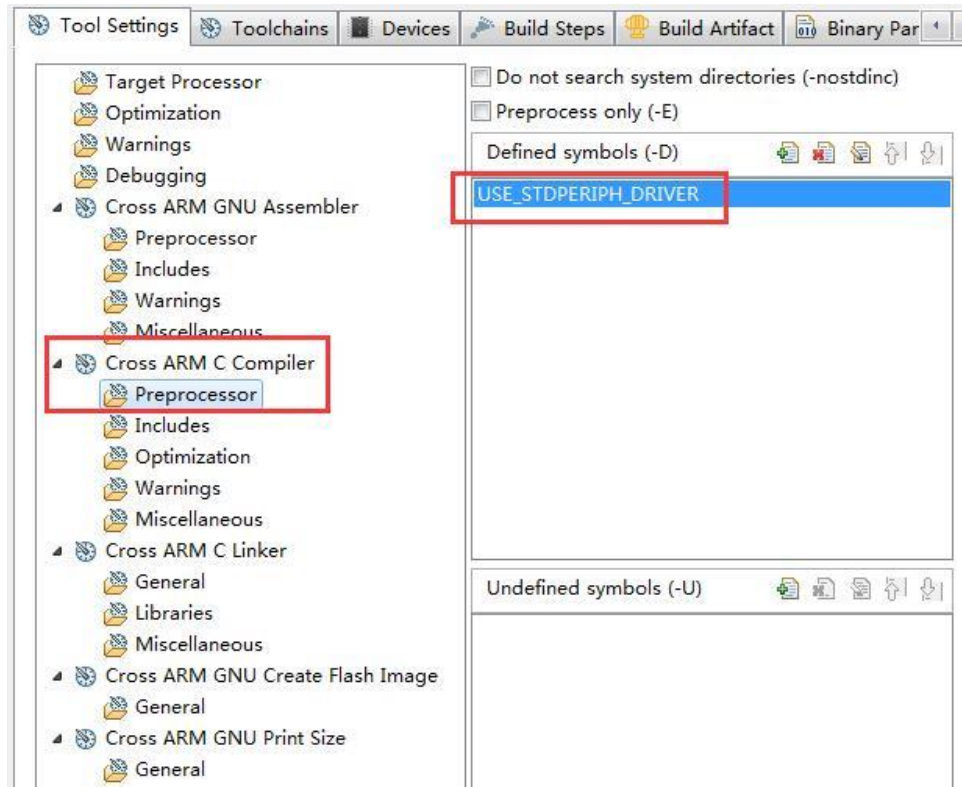
图 2-11 配置交叉编译链



## 2.2.2 配置宏定义

选择 Properties->C/C++Build->Setting，选择 Cross ARM GNU Assembler 和 Cross ARM C Compiler 的 Preprocessor，添加宏定义” USE\_STDPERIPH\_DRIVER”，如图 2-12 所示。

图 2-12 配置宏定义

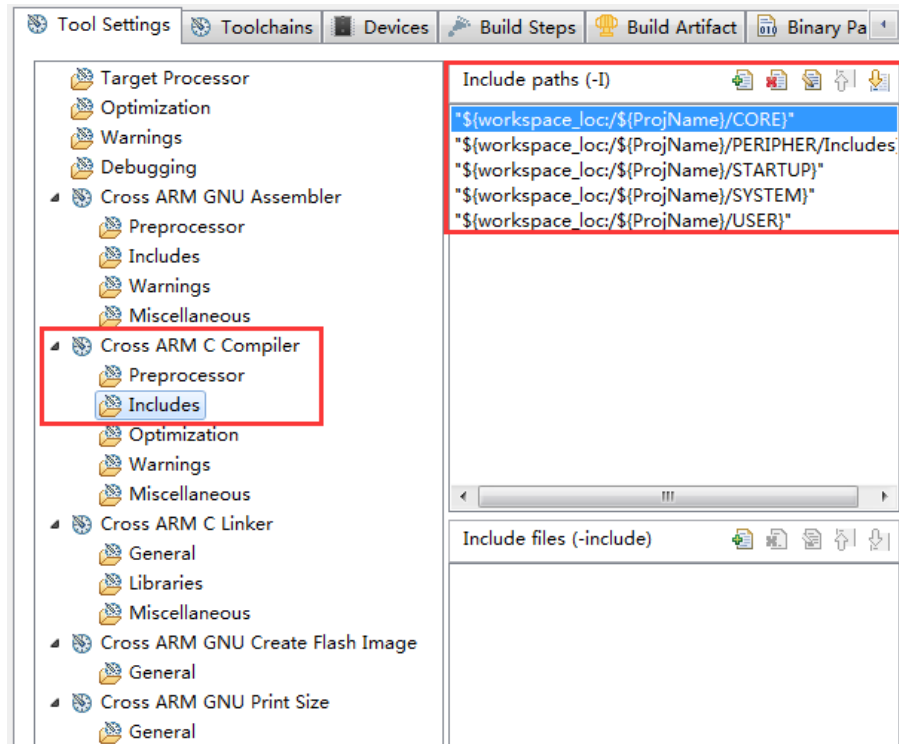


### 2.2.3 配置头文件路径

选择 Properties->C/C++Build->Setting, 选择 Cross ARM GNU Assembler 和 Cross ARM C Compiler 的 Includes, 添加头文件路径, 如图 2-13 所示。



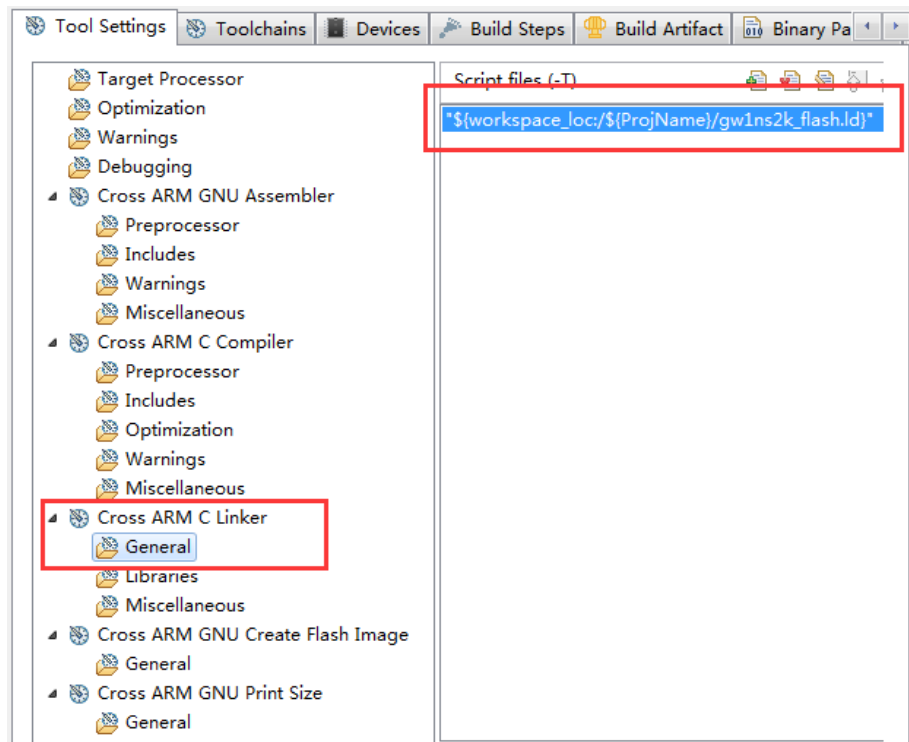
图 2-13 配置头文件路径



## 2.2.4 配置 Flash 链接

选择 Properties->C/C++Build->Setting, 选择 Cross ARM GNU Create Flash Image, 添加 Flash 链接文件, 如图 2-14 所示。

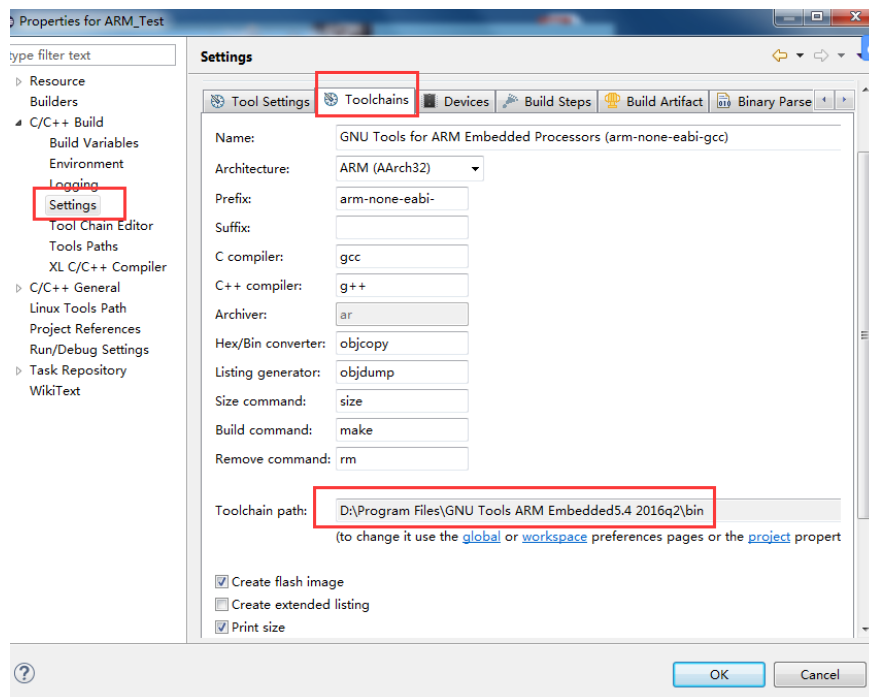
图 2-14 配置 Flash 链接



## 2.2.5 配置交叉编译链路径

选择 Properties->C/C++Build->Setting, 选择 Toolchains, 添加 Toolchain path, 如图 2-15 所示。

图 2-15 交叉编译配置

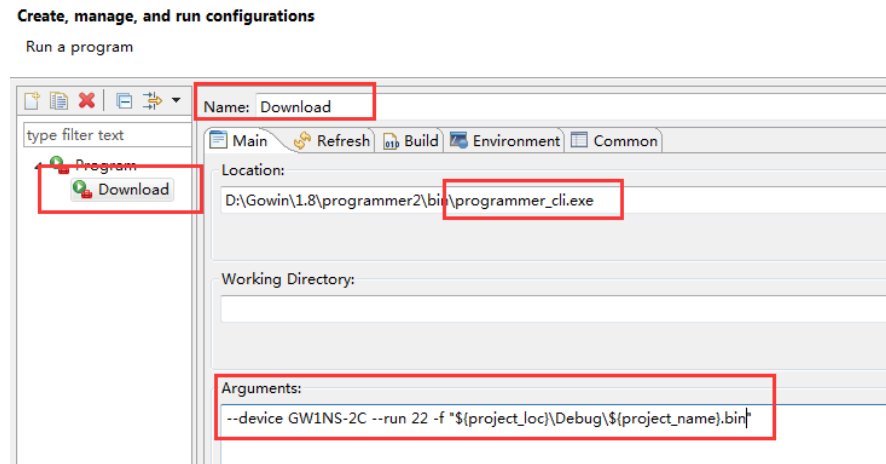


## 2.3 GNU MCU Eclipse 下载配置

GNU MCU Eclipse IDE 软件使用 Gowin Programmer 下载 Flash。

GNU MCU Eclipse IDE 软件中将 Gowin Programmer 添加外部工具，如图 2-16 所示，选择 Run->External Tools->External Tools Configurations...

图 2-16 配置下载工具



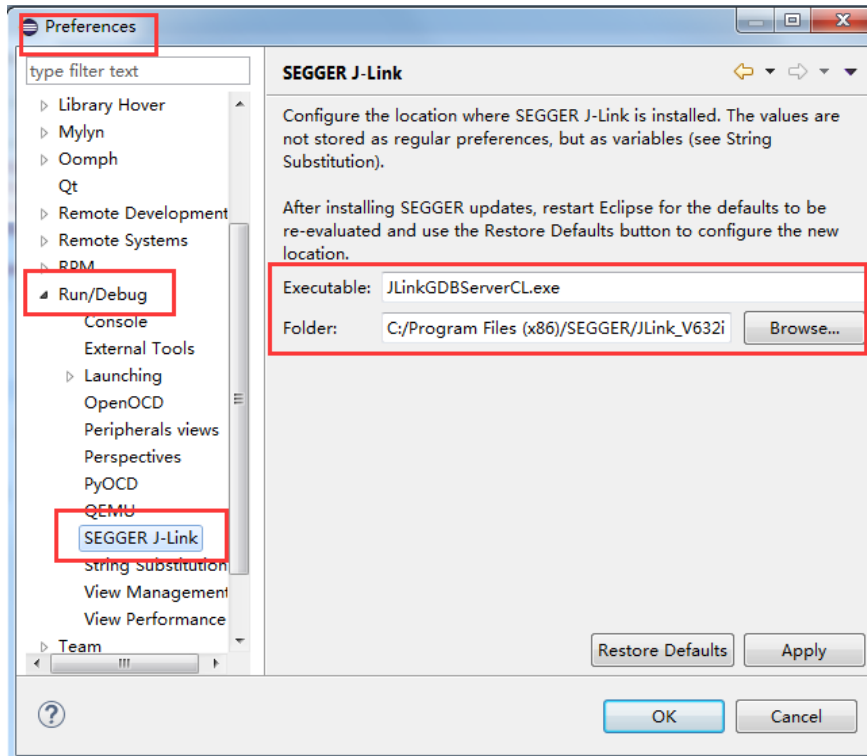
## 2.4 GNU MCU Eclipse 调试配置

安装 J-LINK 驱动程序包，完成后，“Window”菜单栏中“Preferences”->“Run/Debug”->“SEGGER J-Link”配置 J-Link 路径，如图 2.17 所示。

### 2.4.1 安装 J-LINK 驱动

安装 J-LINK 驱动程序包，完成后，“Window”菜单栏中“Preferences”->“Run/Debug”->“SEGGER J-Link”配置 J-Link 路径，如图 2-17 所示。

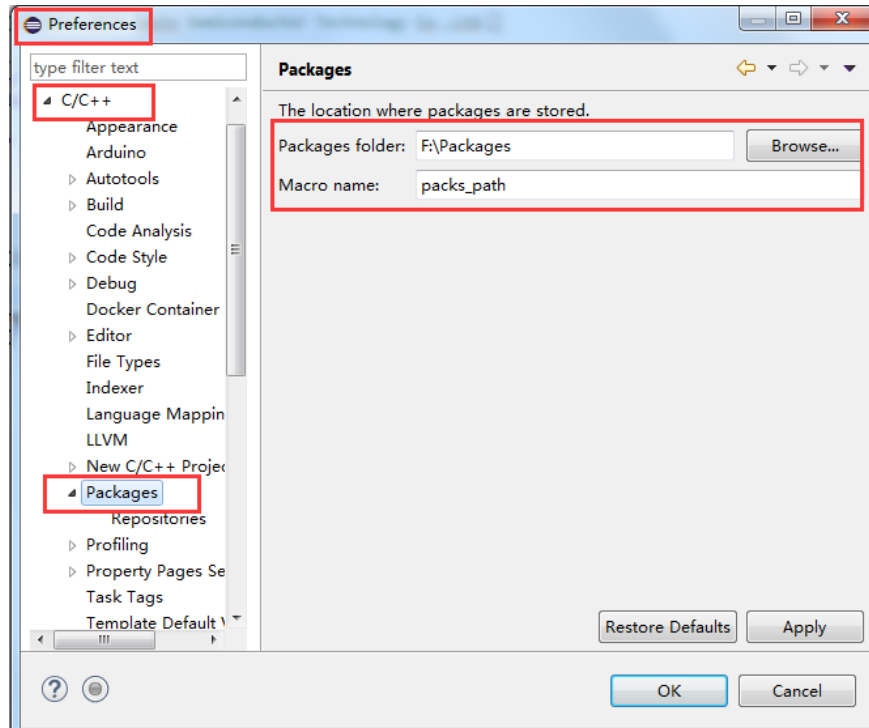
图 2-17 配置 J-Link 路径



## 2.4.2 下载 Device Pack 包

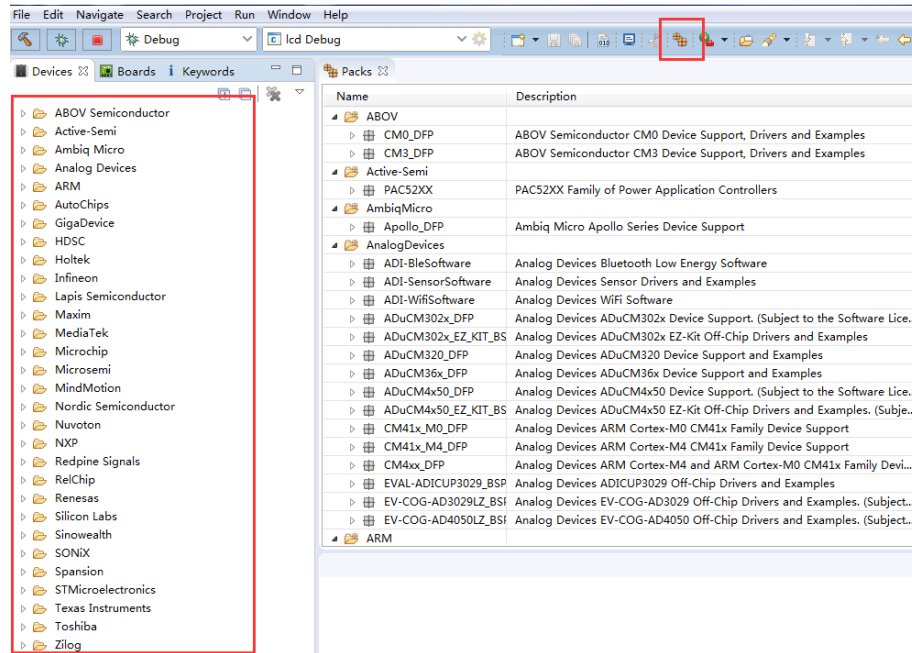
为支持所选器件的调试，需要在线下载器件数据包，首先设置器件数据包下载路径，“Window”菜单栏中“Preferences”->“C/C++”->“Packages”，如图 2-18 所示。

图 2-18 配置 Device Pack 路径



完成器件数据包路径配置后，开始在线选择下载所需器件数据包，如图 2-19 所示。

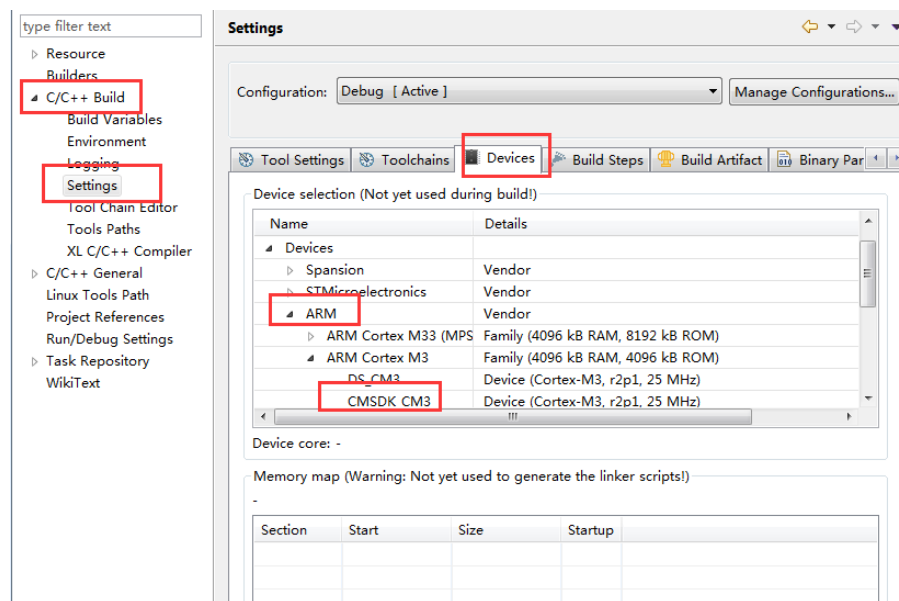
图 2-19 下载 Device Pack



### 2.4.3 设置芯片类型

用户设计编译完成后，如果需要调试，则设定芯片，”Properties” -> “C/C++ Build” -> “Setting” -> “Devices” 配置芯片类型，如图 2-20 所示。

图 2-20 配置芯片类型

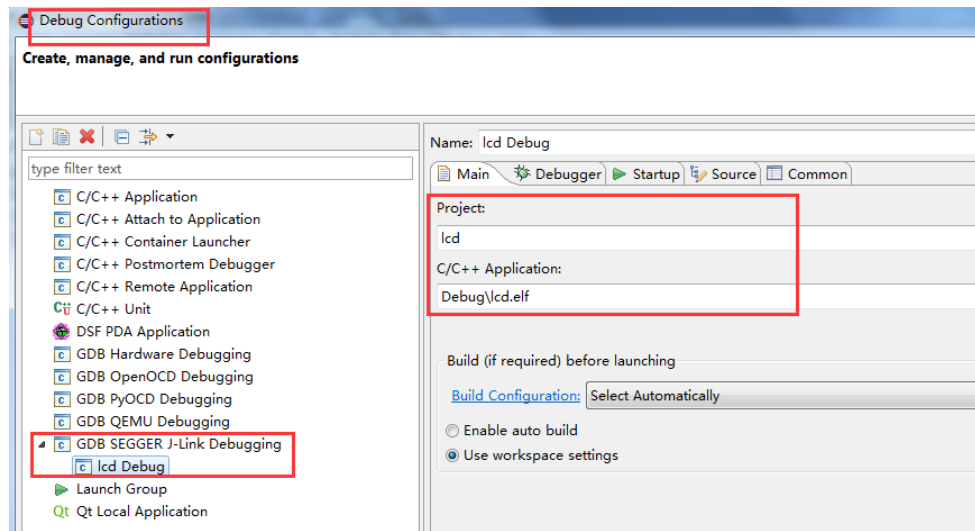


### 2.4.4 配置 Debug

“Run” -> “Debug Configurations” -> “GDB SEGGER J-Link Debugging”，配置所用工程的 Debug 选项。

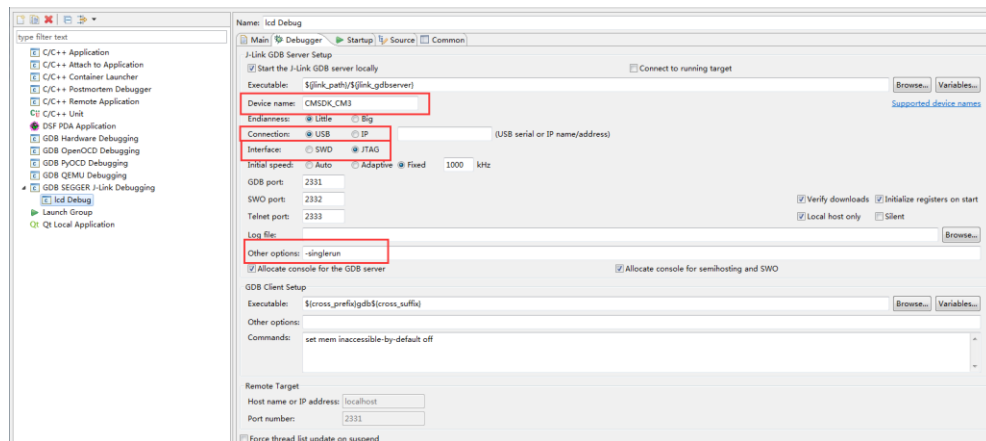
配置 ARM 镜像文件路径，如图 2-21 所示。

图 2-21 配置 Image 路径



配置 Debugger 器件名称和调试接口为 JTAG，如图 2-22 所示。

图 2-22 配置 Debugger

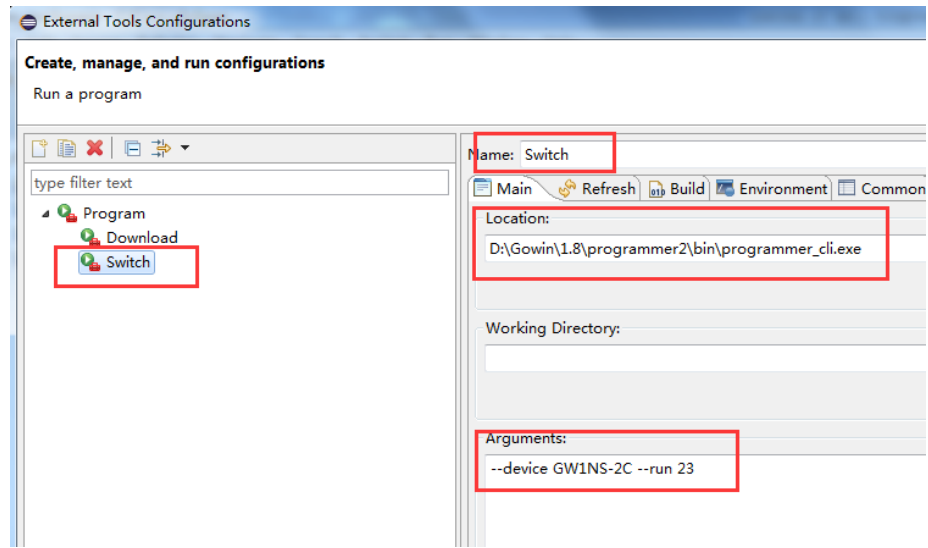


## 2.5 配置 JTAG 模式切换

完成 Image 下载后，如果用户需要调试，需要将 JTAG 模式由下载模式切换到调试模式，配置 JTAG 模式切换，” Run” -> “External Tools” -> “External Tools Configuration...”，配置工具 Switch，用于 JTAG 模式由下

载模式切换到调试模式，如图 2-23 所示。

图 2-23 JTAT 模式切换



完成 Debug 配置和 JTAG 模式切换后，连接 J-LINK 仿真器，即可开始启动调试，如图 2-24 所示。

图 2-24 启动调试

