

MounRiver Studio 帮助手册

V1. 40

目录

一、概述.....	2
二、计算机配置要求.....	2
2.1 最低配置.....	2
2.2 推荐配置.....	2
三、MounRiver Studio 安装	2
四、MounRiver Studio 环境.....	4
4.1 菜单栏.....	4
4.2 快捷工具栏.....	9
4.3 工程目录窗口.....	10
4.4 其他显示窗口.....	11
五、MounRiver Studio 工程.....	11
5.1 新建工程.....	11
5.2 打开工程.....	12
5.3 导入 KEIL 工程.....	12
5.4 编译.....	14
5.5 调试.....	23
5.6 下载	25
5.7 工程导出为模板	28
5.8 工程模板管理.....	28
六、快速问答.....	29
七、更新.....	32
7.1 在线更新	32
7.2 离线更新	32
八、联系我们.....	33

注:WCH-LINK使用方法详见 [WCH-Link使用说明-V1.1.Pdf](#)

(文件位置: MRS安装路径\ MounRiver\ MounRiver_Studio\ ExTool)

一、概述

MounRiver Studio 是一款针对嵌入式开发的集成开发环境。MounRiver Studio 提供了包括 C 编译器、宏汇编、链接器、库管理、强大的调试器和下载器等在内的完整开发资源，通过一个集成开发环境将这些部分组合在一起。

二、计算机配置要求

2.1 最低配置

CPU 主频:	1 GHz
内存:	2G
硬盘剩余空间:	1GB
操作系统:	Windows7

2.2 推荐配置

CPU 主频:	2 GHz 及以上
内存:	2GB 及以上
硬盘剩余空间:	2GB 及以上
操作系统:	Windows 7 及以上

三、MounRiver Studio 安装

下载网址: <http://www.mounriver.com/download>。双击安装包进行安装，出现如图 3.1 所示界面：



图 3.1

按照提示点击“下一步”进行安装，出现如图 3.2 所示界面：



图 3.2

点击“我接受”，出现如图 3.3 所示界面：

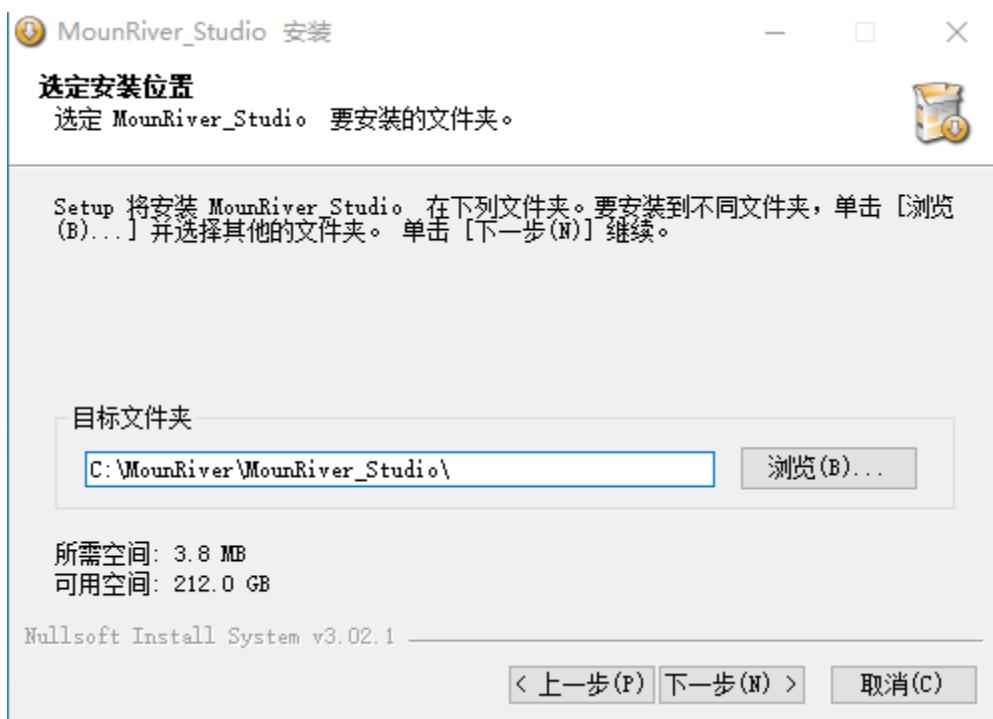


图 3.3

选择安装位置，安装路径不能包含空格，点击下一步，出现如图 3.4 所示界面，点击安装：

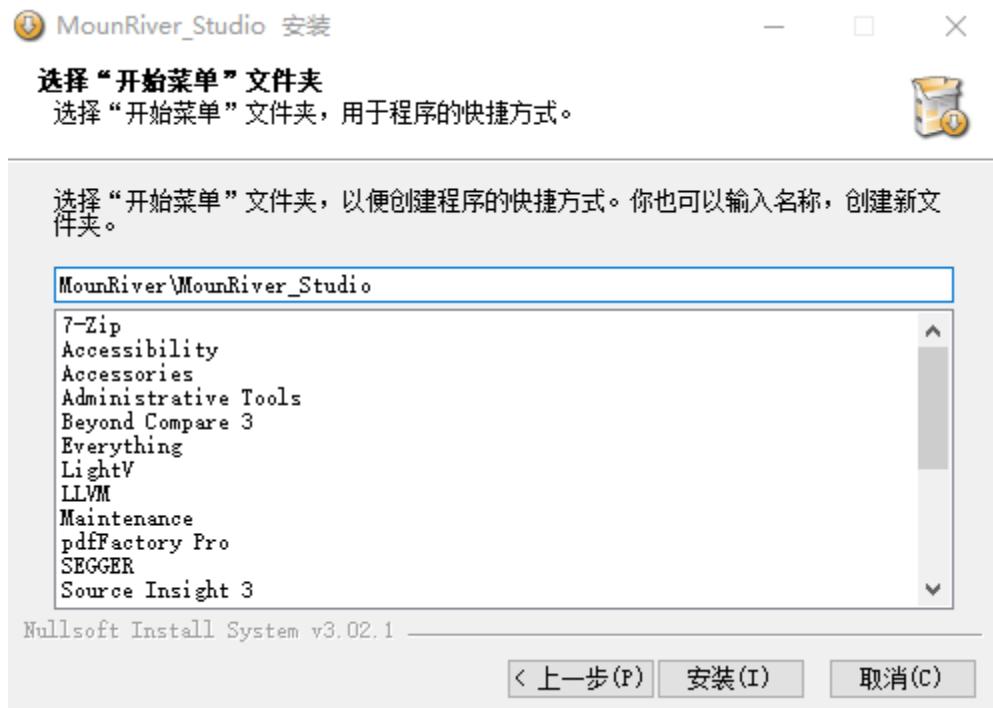


图 3.4

四、MounRiver Studio 环境

4.1 菜单栏

主菜单栏如图4.1所示：



图 4.1

1. File:

New

Alt+Shift+N >

新建项目

 Import Keil Project

导入待转换的 Keil 工程

Load Project

加载 MounRiver 工程

Close

Ctrl+W

关闭资源管理器中选中的工程

Close All

Ctrl+Shift+W

关闭资源管理器中所有的工程



Save

Ctrl+S

保存

 Save As...

另存为

 Save All

Ctrl+Shift+S

全部保存

 Move...

移动

 Rename...

F2

重命名

 Refresh

F5

刷新 IDE

 Import...

导入

 Export...

导出

 Properties

Alt+Enter

属性

 Restart

重新启动 IDE

 Exit

关闭 IDE

2. Edit:

 Undo Typing

Ctrl+Z

撤销

 Redo

Ctrl+Y

反撤销

 Cut

Ctrl+X

剪切

 Copy

Ctrl+C

复制

 Paste

Ctrl+V

粘贴

 Remove

删除

Select All Ctrl+A

全选

Expand Selection To >

将选择范围扩展到

Toggle Block Selection Alt+Shift+A

打开块选择

Find/Replace... Ctrl+H

查找/替换

Find Word

查找单词

Find Next Ctrl+K

查找下一个

Find Previous Ctrl+Shift+K

查找上一个

Incremental Find Next Ctrl+J

增量式查找下一个

Incremental Find Previous Ctrl+Shift+J

增量式查找上一个

Add Bookmark...

添加书签

Smart Insert Mode Ctrl+Shift+Insert

智能插入模式

Show Tooltip Description F2

显示工具提示描述

Word Completion Alt+ /

文字补全

Quick Fix Ctrl+1

快速修正

Content Assist Tab

内容辅助

Parameter Hints Ctrl+Shift+Space

参数提示

Set Encoding...

设置编码

3. Project:

 Open Project

打开工程

 Close Project

关闭工程

 Build All

Ctrl+B

编译全部工程

 Build Project

F7

增量编译选中的工程

 Clean...

清理工程

 Build Automatically

自动编译

 Concise Build Output Mode

精简编译输出模式

 Target Link Type

目标调试器类型

 Template Management

Ctrl+Shift+T

工程模板管理

 Save As Project Template

Ctrl+Shift+X

导出工程为模板

 Properties

工程属性

4. Run:

 Run

Ctrl+F11

运行

 Debug

F11

调试

 Run History

>

运行历史记录

 Run As

>

运行方式

 Run Configurations...

运行配置

 Debug History

>

调试历史记录

 Debug As >

调试方式

 Debug Configurations...

调试配置

5. Flash:

 Download for RISC-V

RISC-V 内核芯片代码烧录

 Configuration

RISC-V 内核芯片烧录配置

 Download for ARM

ARM 内核芯片代码烧录

6. Tool:

 Calculator

计算器

 Device Management

设备管理器

 ISP Tool

ISP 升级工具

7. Windows:

 Show View >

显示视图

 Reset View to Defaults

恢复默认透视图排版

 Preferences

首选项

 Theme

界面主题

8. Help:

 Welcome

欢迎页

 Show Active Keybindings... Ctrl+Shift+L

显示快捷键映射表

 Help Manual

打开帮助文档

 Open Workbench Log

查看 IDE 运行日志

 Language

切换 IDE 界面语言

 Offline Upgrade

IDE 离线升级

 Check Updates

检查在线升级

 About MounRiver Studio

关于

 Visit Website

访问 MounRiver Studio 官网

4.2 快捷工具栏

快捷工具栏如图4.2:



1.  新建各种类型文件
2.  保存当前文件
3.  全部保存
4.  导入待跨核转换的 keil 工程
5.  打开.1d 文件配置界面
6.  弹出快捷工具栏帮助文档
7.  打开 IDE 全局设置
8.  打开工程属性设置

- 9.  打开命令行工具
- 10.  打开终端工具集
- 11.  增量编译当前工程
- 12.  重新编译
- 13.  编译全部工程
- 14.  RISC-V 内核芯片代码烧录
- 15.  ARM 内核芯片代码烧录
- 16.  debug 及其配置页
- 17.  查找及其设置
- 18.  左移选中的代码
- 19.  右移选中的代码
- 20.  添加/取消行注释
- 21.  跳转至下一个断点, 书签等, 点击下拉按钮可配置
- 22.  跳转至上一个断点, 书签等, 点击下拉按钮可配置
- 23.  光标跳转至上次编辑的位置
- 24.  跳转至上一次打开的文件页面
- 25.  返回至跳转之前的文件页面
- 26.  撤销上一步操作
- 27.  取消撤销

4.3 工程目录窗口

包含各个工程的目录结构, 如图4.3

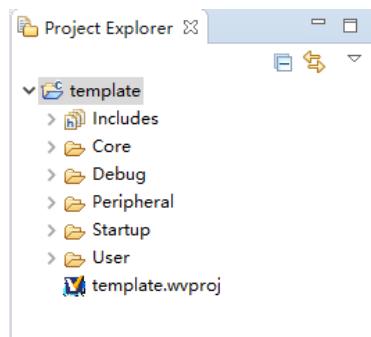


图 4.3

4.4 其他显示窗口

在IDE界面右上角可选择显示模式，点击 可选择显示模式，进入调试模式后会自动切换为调试模式。各种模式显示的窗口不同，都可以在各自模式点击菜单栏中的 Window，如图4.4进行配置

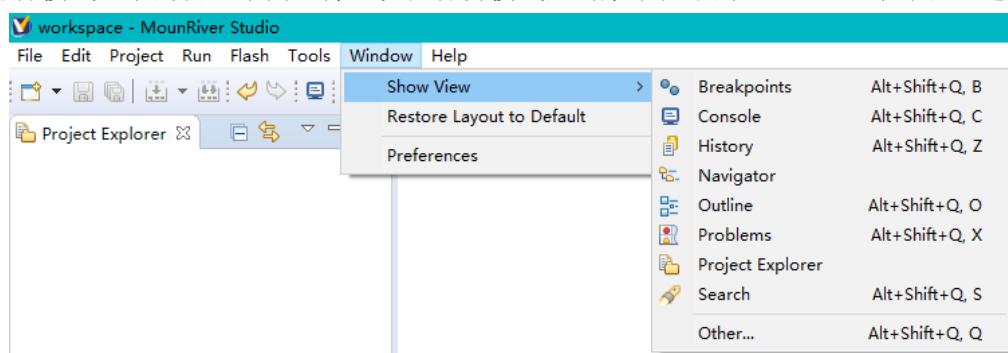


图 4.4

五、MounRiver Studio 工程

5.1 新建工程

点击工具栏 File，按图5.1.1显示点击 MounRiver Project:

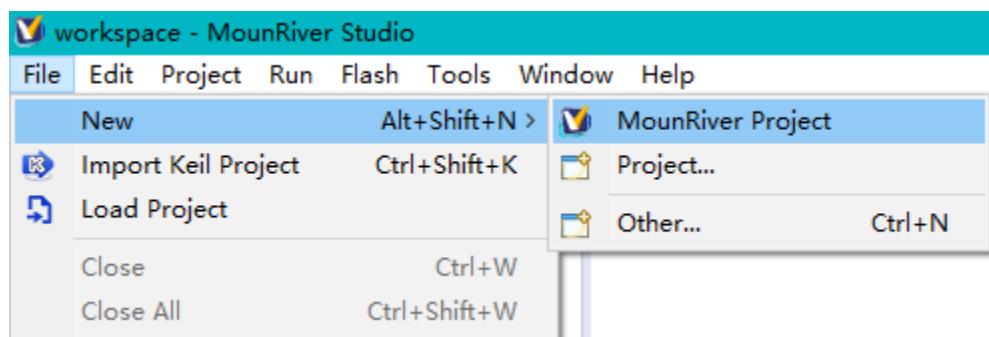


图 5.1.1

出现如下图5.1.2的界面：

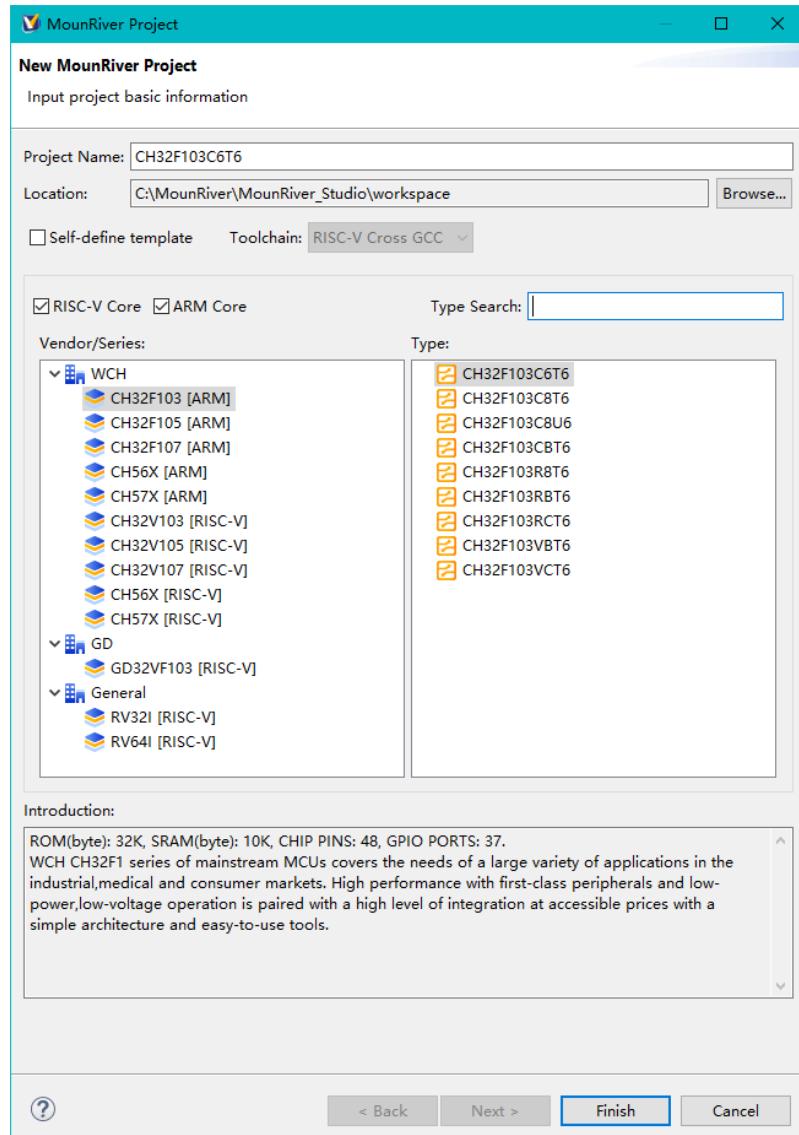


图 5.1.2

Project Name 为工程名称，选择工程默认存放路径，左侧框选择芯片厂商和系列，或者选择通用类型，右侧选择具体的芯片型号，可通过 RISC-V Core 与 ARM Core 勾选项以及 Type Search 快速过滤内置芯片模板；也可勾选 Self-define template，创建自定义工程。点击 **Finish** 完成创建工程。

5.2 打开工程

在建好的工程源码目录中双击名字为 工程名.wvproj 的文件可直接进入 MounRiver Studio。

5.3 导入 KEIL 工程

导入已有的keil工程文件，点击快捷工具栏 ，或者如图5.3.1所示点击 **Import Keil Project**

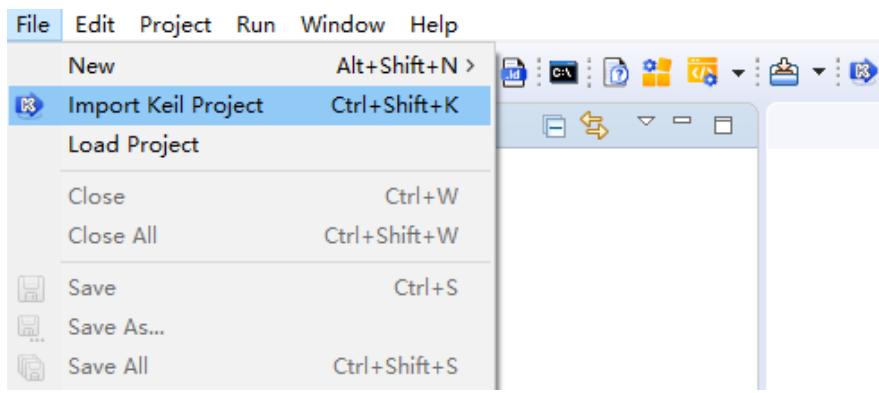


图 5.3.1

显示出如下界面：

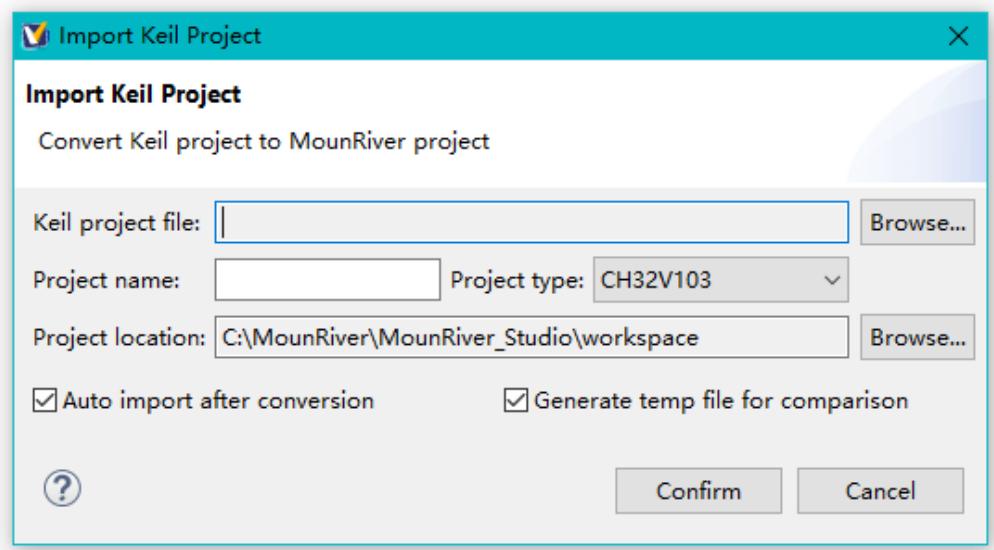


图 5.3.2

1. Keil project file	选择要转换的keil工程
2. Use default workspace location	使用默认保存位置
3. Target project name	转换后的工程名
4. Target project location	转换后的工程存放路径
5. Auto import after conversion	转换后是否直接导入到当前工程目录窗口
6. Generate temp file for comparison	是否生成转换时的临时文件，临时文件只转换keil工程的目录结构，可用于与转换后的工程文件对比 展示转换中的修改之处，建议勾选并做对比

点击 **confirm**，若keil工程中的标准库文件被修改过，会弹出对话框：

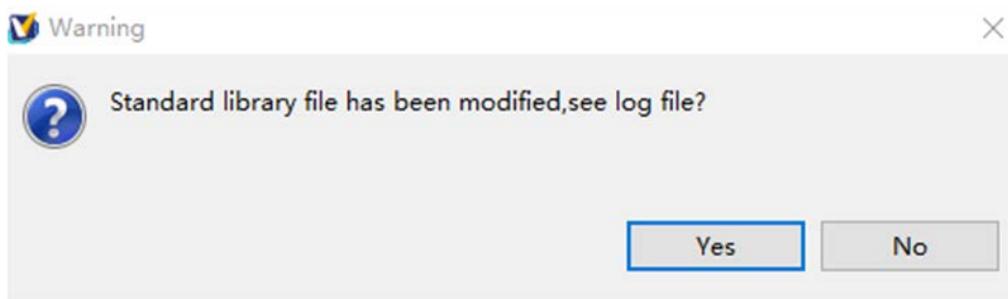


图 5.3.3

点击 **Yes** 可打开转换日志，根据日志查找需要手动修改的文件。

5.4 编译

选中工程目录窗口中的工程，鼠标右键单击，然后点击**build project**，或者点击快捷工具栏中的进行编译，console窗口会显示build过程中产生的信息：



图 5.4.1

若编译成功，则编译过程中产生的文件存放在源码目录下的obj文件中。

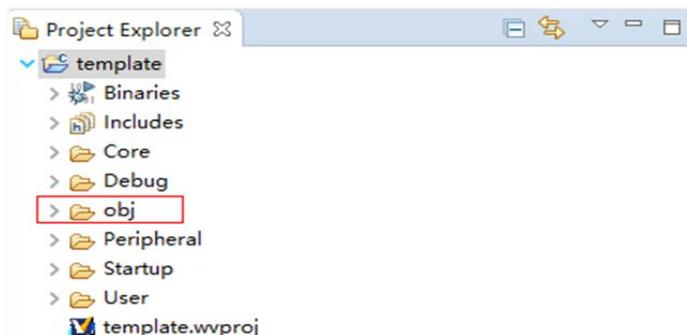


图 5.4.2

如果需要对编译过程做进一步的配置，选中工程目录窗口中的工程，鼠标右键单击，然后点击 properties，以下为常用配置：

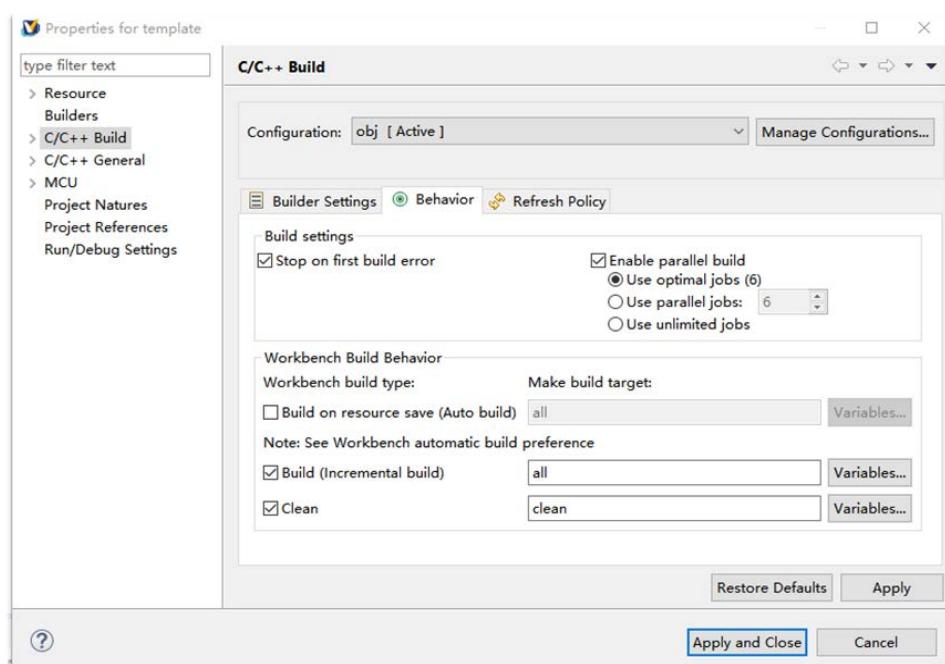


图 5.4.3

选中左侧选项卡 C/C++ Build , 选中右侧选项卡 Behavior:

1. Stop on first build error	编译遇到第一个错误就停止编译
2. Enable parallel build	可选择的编译线程个数
3. Build on resource save (Auto build)	保存文件自动build
4. Build (incremental build)	增量编译
5. Clean	清除build产生的文件

点击左侧选项卡 C/C++ Build 的下拉选项, 选择 Settings, 在右侧弹窗中选择 Tool settings 下的 Target processor。

RISC-V内核芯片属性配置显示如图5.4.4:

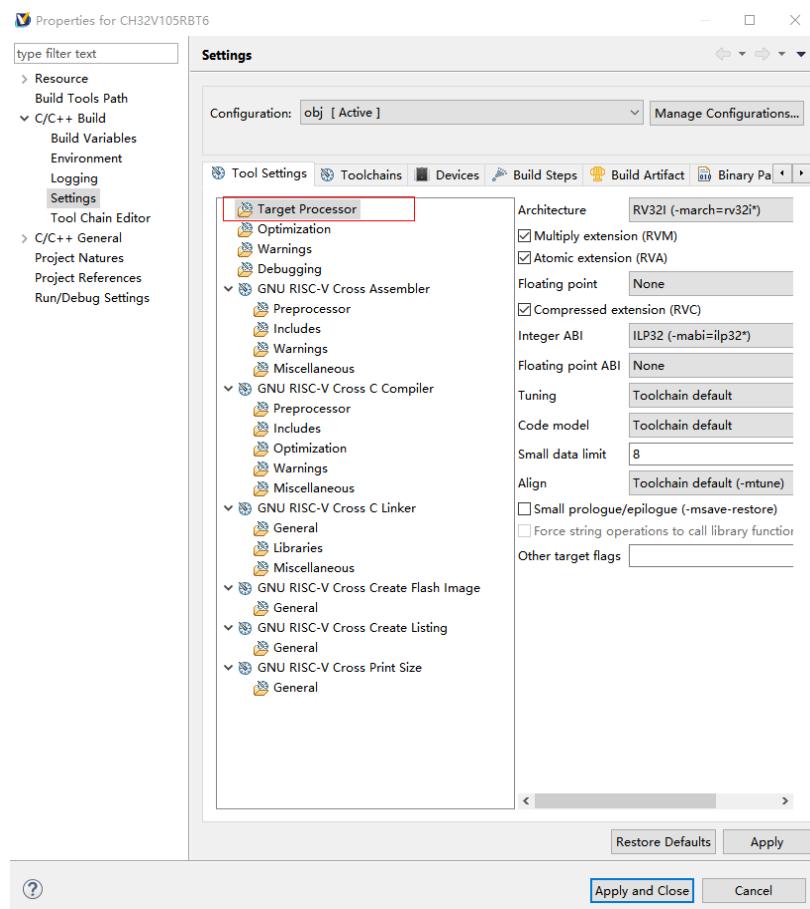


图 5.4.4

Target processor 主要设置目标处理器属性, 详细信息可以在 GCC 文档 3.18.38 RISC-V Options 查看。

1. Architecture 是指令集架构, rv32i 是 RISC-V 基础整数指令集
2. RVM 表示支持乘除法扩展
3. RVA 表示支持原子扩展
4. RVF 表示单精度浮点数扩展
5. RVD 为双精度浮点数扩展
6. RVC 为压缩指令扩展
7. Integer ABI 为 RISC-V 应用程序整数二进制接口

8. Floating point ABI 为 RISC-V 应用程序浮点数二进制接口
9. Tuning 由微架构优化给定处理器的输出, 默认是 rocket
10. Code model
 - mcmode=medlow: 程序及其静态定义的符号必须位于单个 2 GiB 地址范围内, 并且必须位于绝对地址-2 GiB 和+2 GiB 之间。程序可以静态或动态链接。这是默认的代码模型
 - mcmode=medany: 程序及其静态定义的符号可以任何单个 2 GiB 地址范围内。程序可以静态或动态地连接。
11. Small data limit: 在某些目标上将小于 n 字节的全局和静态变量放进一个特殊的段。 Align 中-mstrict-align -mno-strict-align 取决于处理器是否支持内存的非对齐访问

RISC-V 编译器支持多个 ABI, 具体取决于 F 和 D 扩展是否存在。RV32 的 ABI 分别名为 ilp32, ilp32f 和 ilp32d。ilp32 表示 C 语言的整型 (int), 长整型 (long) 和指针 (pointer) 都是 32 位, 可选后缀表示如何传递浮点参数。在 ilp32 中, 浮点参数在整数寄存器中传递; 在 ilp32f 中, 单精度浮点参数在浮点寄存器中传递; 在 ilp32d 中, 双精度浮点参数也在浮点寄存器中传递。

自然, 如果想在浮点寄存中传递浮点参数, 需要相应的浮点 ISA 添加 F 或 D 扩展。因此要编译 RV32I 的代码 (GCC 选项-march=rv32i), 必须使用 ilp32 ABI (GCC 选项-mabi=ilp32)。反过来, 调用约定并不要求浮点指令一定要使用浮点寄存器, 因此 RV32IFD 与 ilp32, ilp32f 和 ilp32d 都兼容。

ARM 内核芯片属性配置显示如图 5.4.5:

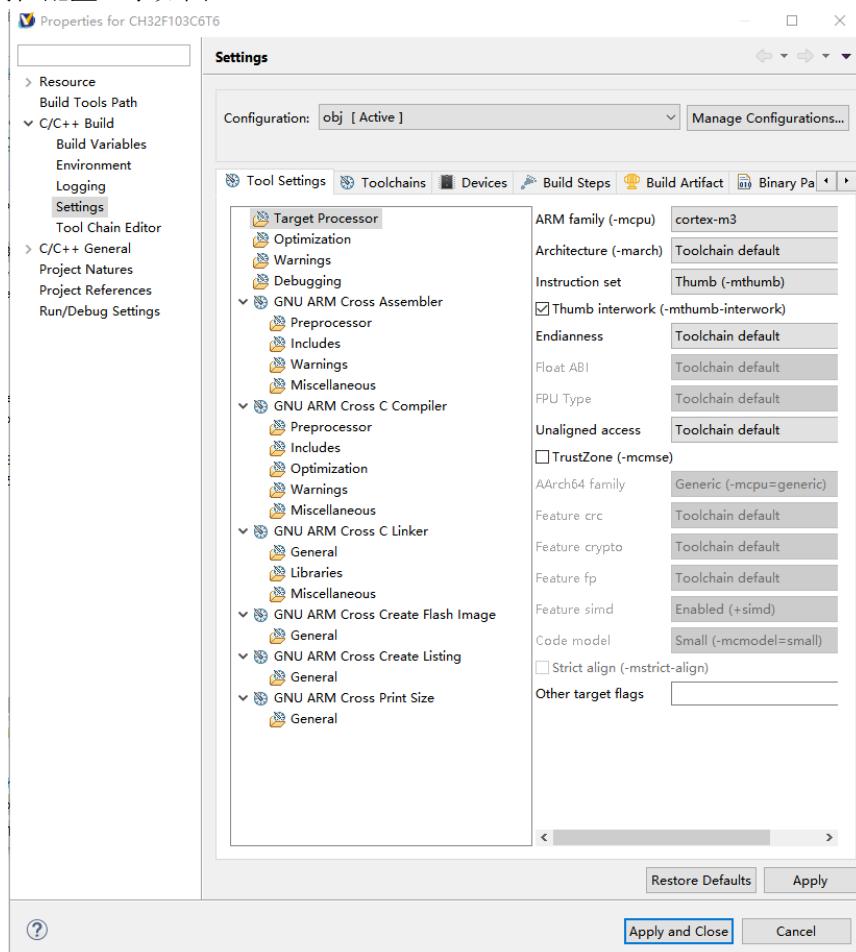


图 5.4.5

1. ARM family 设置芯片内核类型, 如上图 CH32F103 芯片内核为 cortex-m3
2. Architecture 设置指令集架构, 上述芯片为 ARMV7-M, 一般设置完芯片内核, 此选项可选为默认。

3. Instruction set 设置芯片工作模式, thumb 或者 arm 模式。
4. Thumb interwork 生成支持 ARM 和 Thumb 之间调用的代码指令集。如果没有此选项, 则在 v5 之前的架构上, 这两个指令集不能在一个程序中可靠地使用。
5. Endianness 设置大小端, 默认为小端。
6. unaligned access 设置是否允许非对齐访问, 启用 (或禁用) 来自未对齐 16 位或 32 位的地址的 16 位和 32 位值的读取和写入。默认情况下, 未对齐访问为对于所有 ARMv6 之前的版本, 所有 ARMv6-M 和 ARMv8-M Baseline 体系结构, 均禁用; 对于所有其他体系结构, 则启用。设置完芯片内核类型, 其他选型如不清楚可选择默认属性。

点击左侧选项卡 C/C++ Build 的下拉选项, 选择 Settings, 在右侧弹窗中选择 Tool settings 下的 Optimization, 属性页如下图 5.4.6 所示:

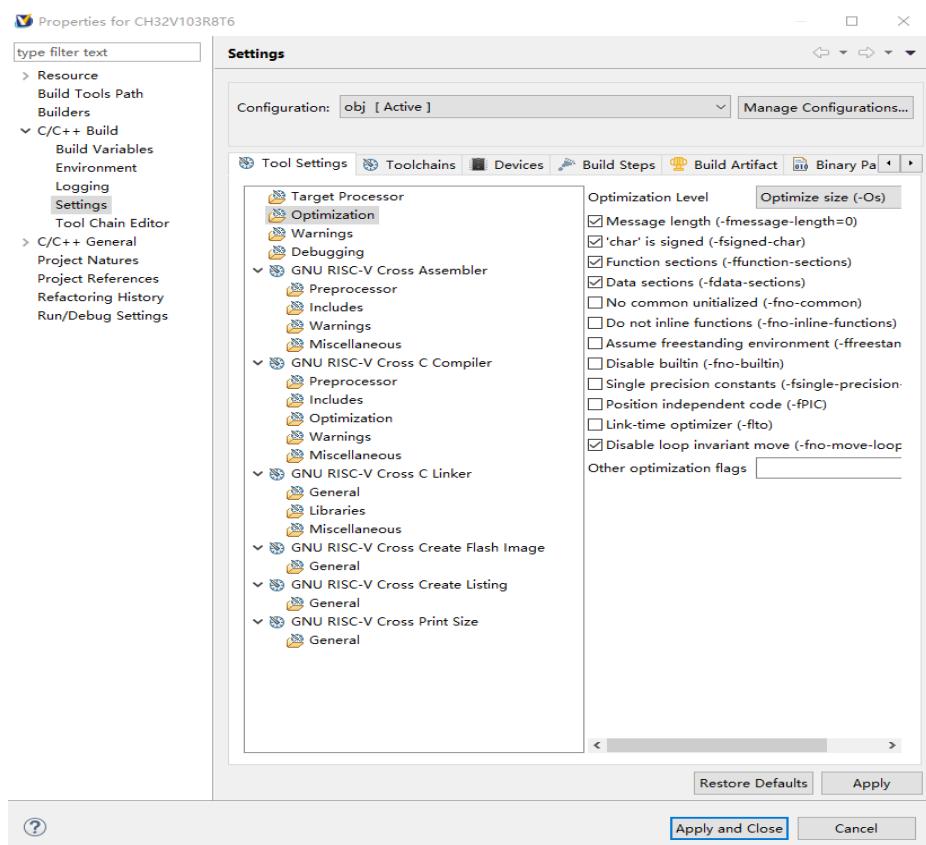


图 5.4.6

该属性页主要是配置 GCC 的优化选项, 想要添加其他优化选项可以写在下方 other optimization flags 中, 常用优化选项含义为:

-00: 无优化(默认)

-0 和 -O1: 使用能减少目标文件大小以及执行时间并且不会使编译时间明显增加的优化。在编译大型程序的时候会显著增加编译时内存的使用。

-O2: 包含 -O1 的优化并增加了不需要在目标文件大小和执行速度上进行折衷的优化。编译器不执行循环展开以及函数内联。此选项将增加编译时间和目标文件的执行性能。

-Os: 专门优化目标文件大小, 执行所有的不增加目标文件大小的 -O2 优化选项。并且执行专门减小目标文件大小的优化选项。

-O3: 打开所有 -O2 的优化选项并且增加部分参数。

点击左侧选项卡 C/C++ Build 的下拉选项，选择 Settings，在右侧弹窗中选择 Tool settings 下的 GNU RISC-V Cross C Linker->Miscellaneous，属性页如下图5.4.7所示：

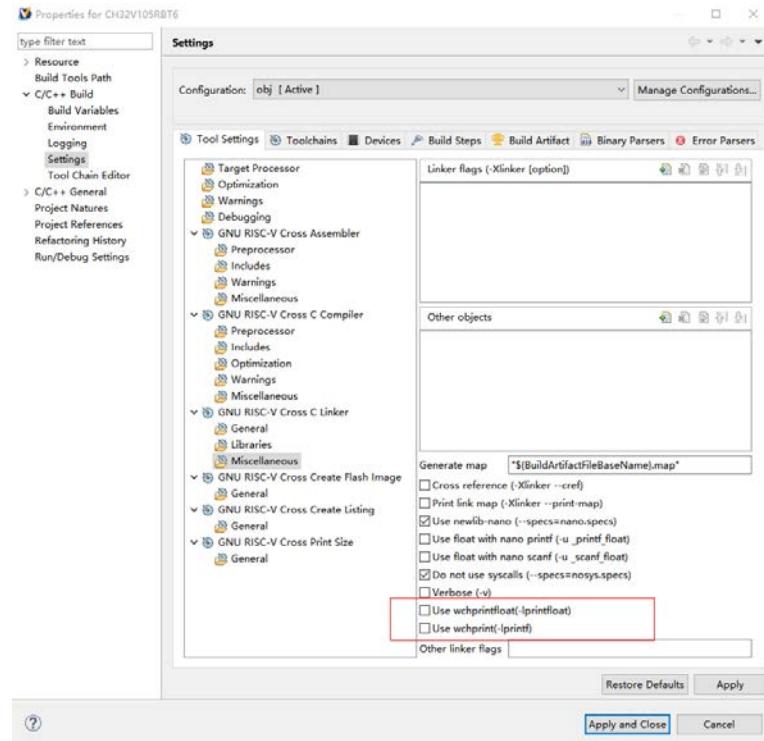


图 5.4.7

此处选项勾选 Use wchprintf，会使用简化版的 printf 函数，减少代码大小，支持常见的 %s, %c, %d, %f, %x, %o 类型，支持 %m.n d, %m.nf, %0m d, %-m d 等常见格式

此处选项勾选 Use wchprintffloat，会增加打印浮点数功能，相比标准库提供的 Use float with nano printf 可以显著减少代码大小。

点击左侧选项卡 C/C++ Build 的下拉选项，选择 Settings，在右侧弹窗中选择 Build Steps，属性页如下图5.4.8所示：

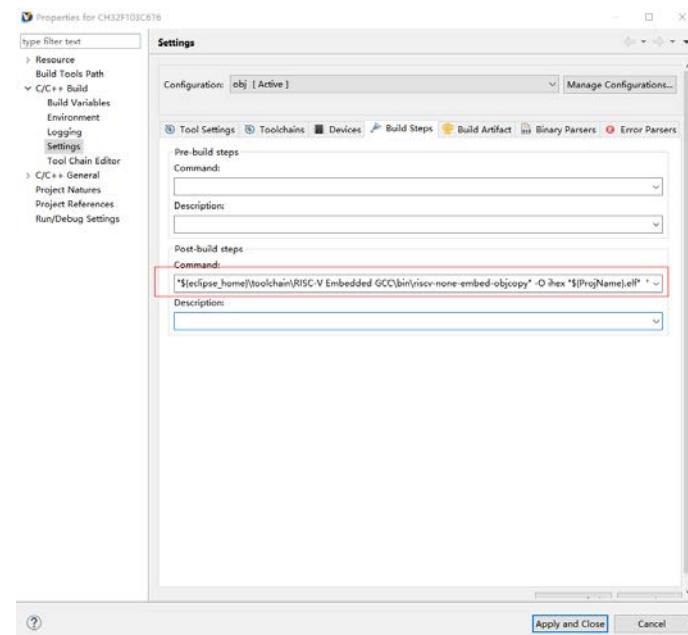


图 5.4.8

如图红圈位置可以增加编译后执行的指令

eg:

```
"${eclipse_home}\toolchain\RISC-V Embedded GCC\bin\riscv-none-embed-objcopy"
-o ihex "${ProjName}.elf" "文件存放路径\\${ProjName}.hex"
```

可用于设置生成 hex 文件位置

```
"${eclipse_home}\toolchain\RISC-V Embedded GCC\bin\riscv-none-embed-objcopy"
-o binary "${ProjName}.elf" "文件存放路径\\${ProjName}.bin"
```

可用于设置生成 bin 文件位置

点击左侧选项卡 C/C++ Build 的下拉选项，选择 Settings，在右侧弹窗中选择 Tool settings 下的 Warnings，显示如图 5.4.9：

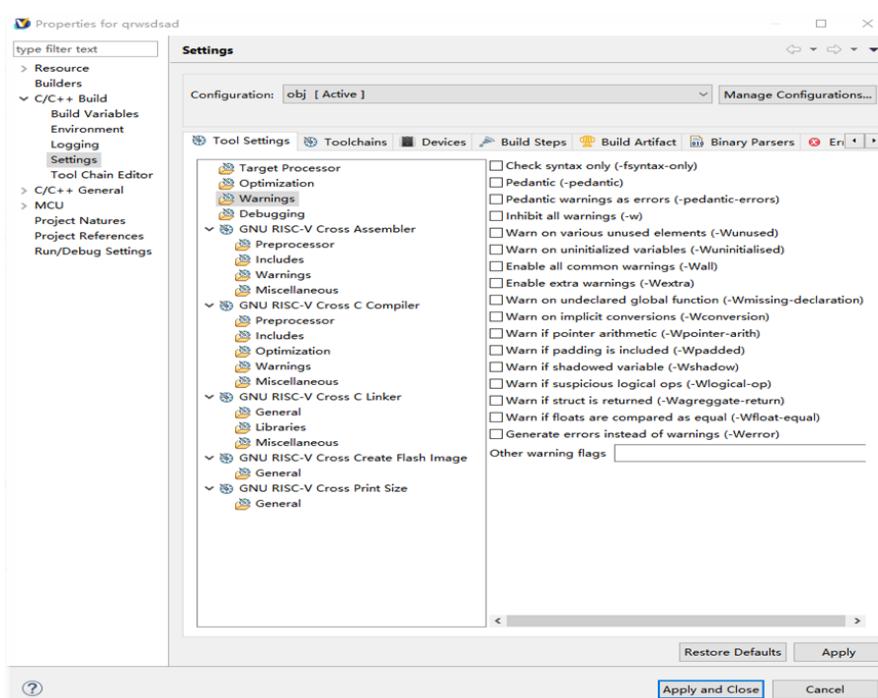


图 5.4.9

1. Check syntax only	只检查语法错误
2. Pedantic	严格执行 ISO C 和 ISO C++ 要求的所有警告
3. Pedantic warnings as errors	ISO C 和 ISO C++ 要求的所有警告显示为错误
4. Inhibit all warnings	禁止全部警告
5. Warn on various unused elements	所有 -Wunused 参数的集合
6. Warn on uninitialized variables	未初始化局部变量的警告
7. Enable all common warning	显示所有警告
8. Enable extra warnings	显示 -Wall 之外的警告
9. Warn on undeclared global function	全局函数在头文件中没有声明的警告
10. Warn on implicit conversion	隐式转换可能改变值的警告
11. Warn if pointer arithmetic	对函数指针或者 void* 类型的指针进行算术操作
12. Warn if padding is included	结构体填充警告
13. Warn if shadow variable	变量或类型声明遮盖影响了另一个变量

14. Warn if suspicious logic ops	可疑的逻辑操作符警告
15. Warn if struct is returned	返回结构、联合或数组时给出警告
16. Warn if floats are compared as equal	浮点值比较相关的警告
17. Generate errors instead of warnings	生成错误 代替警告

仅供参考, 详见GCC警告参数

点击左侧选项卡 C/C++ Build的下拉选项, 选择Settings, 在右侧弹窗中选择Tool settings下的 GNU RISC-V Cross C Linker下的Miscellaneous。显示出如图5.4.10:

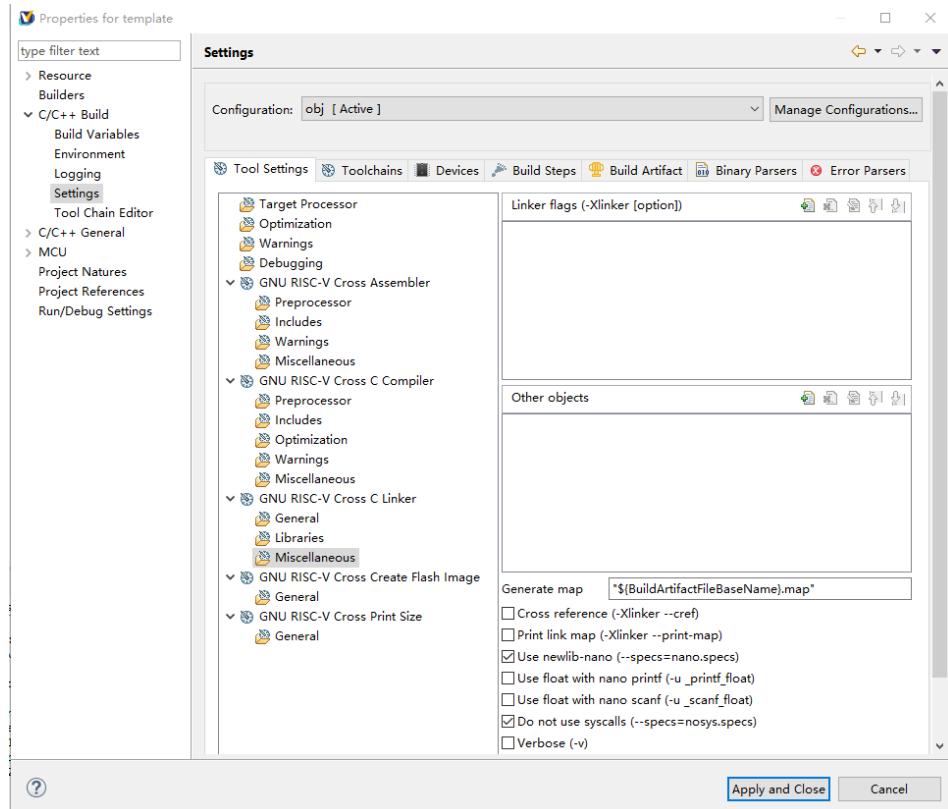


图 5.4.10

在右侧Other objects中, 可另外添加工程中要参与链接的文件, 注意参与链接的文件所在的路径中不要包含括号或者特殊字符, 点击绿色的加号, 如图5.4.11:



图 5.4.11

显示出如图5.4.12:

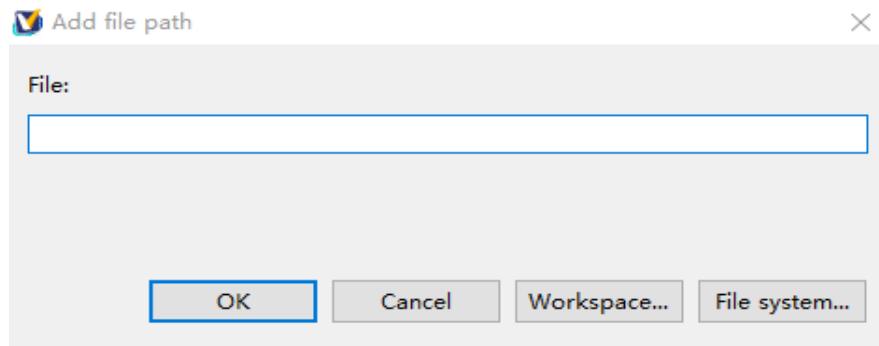


图 5.4.12

点击 **File system...**或**Workspace...**，选中要添加的库文件，点击OK，添加成功则库文件路径会显示在下方窗口。



图 5.4.13

点击左侧选项卡 **C/C++ Build** 的下拉选项，选择 **Settings**，在右侧弹窗中选择 **Tool settings** 下的 **GNU RISC-V Cross C Create FlashImage** 下的 **General**。显示出如图 5.4.14：

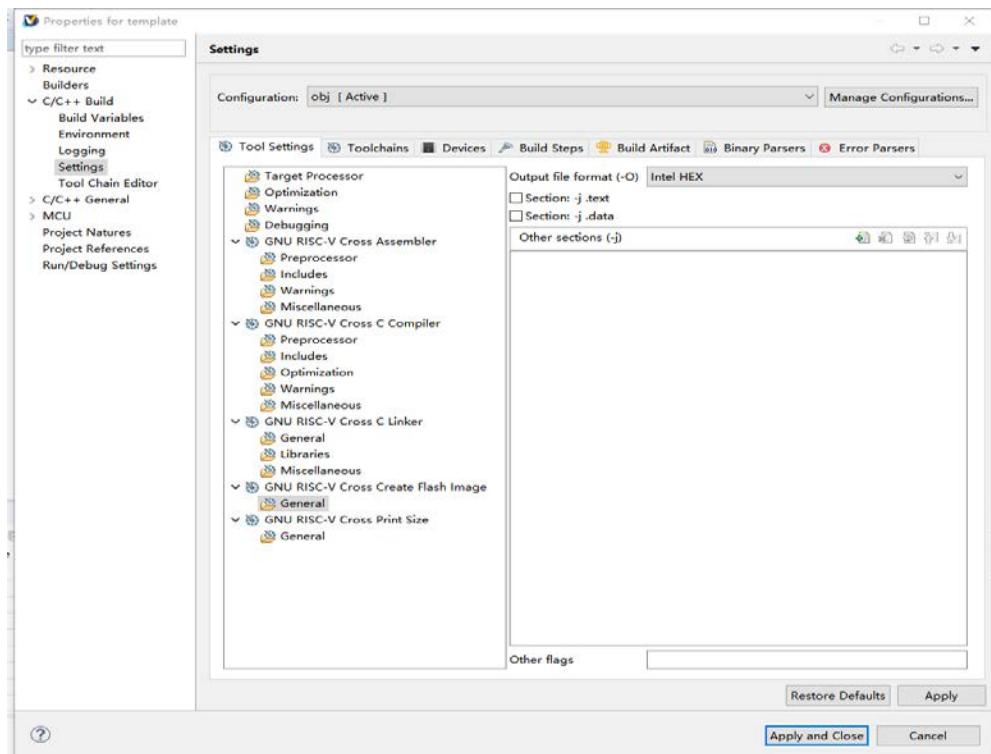


图 5.4.14

在右侧窗口 **output file format(-O)** 中，点击下拉工具可选择生成 Intel HEX 或者生成 Raw binary 文件。

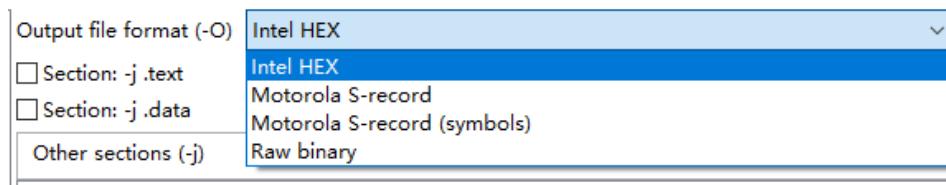


图 5.4.15

点击左侧选项卡 C/C++ Build 的下拉选项，选择 Build Artifact，显示出如图5.4.16：

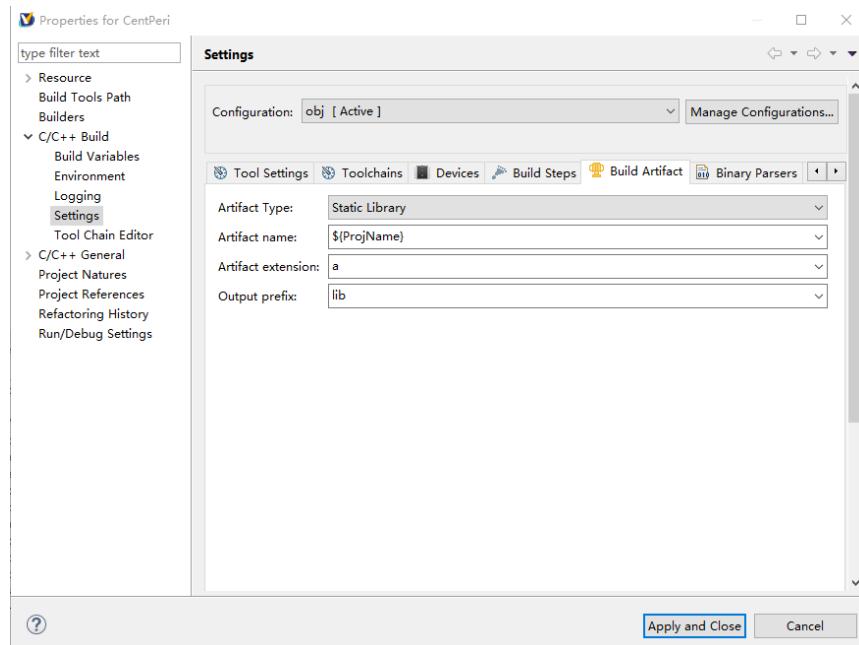


图 5.4.16

Artifact type	可下拉选择生成
executable	可执行文件
Static library	静态库 如果要缩减静态库大小，可选择 Tool Setting 下的 Debugging，设置 Debug level 为NONE
Artifact name	可更改默认生成的可执行文件等文件的名字，删除原来默认名称，填写自定义的名字。
Output prefix	库文件名字前缀

点击左侧选项卡 C/C++ Build 的下拉选项，选择 Settings，在右侧弹窗中选择 Tool settings 下的 GNU RISC-V Cross C Linker 下的 Libraries。显示出如图5.4.17：

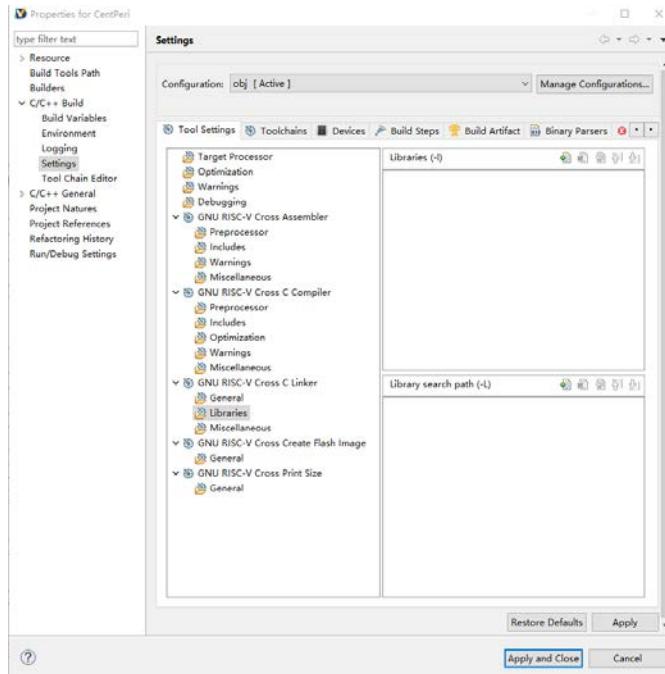


图 5.4.17

在下方方框中点击绿色加号添加库文件路径（路径中不要包含“\$”等特殊字符），上方方框中填入库名，库名不要加 Lib 前缀和. a 后缀，以此完成库文件的添加。

以上所有配置如有更改，必须点击相应配置页面的 **Apply** 或者 **Apply and Close** 选项，否则更改的配置不生效。

5.5 调试

选中工程目录窗口中的工程，如果未编译，则先编译工程，再点击快捷工具栏中 ，进入调试模式。

注:WCH-LINK使用方法详见 [WCH-Link使用说明-V1.1.pdf](#)

5.5.1 快捷工具栏



- 跳过所有断点
- 继续
- 暂挂
- 终止
- 单步跳入
- 单步跳过
- 指令集单步模式
- 单步返回

5.5.2 断点

双击代码行左侧，设置断点，再次双击取消断点。



```

111 * Input      : None
112 * Return     : None
113 ****
114 void ADC1_2_IRQHandler()
115 {
116     if(ADC_GetITStatus( ADC1, ADC_IT_AWD)){
117         printf( "Enter AnalogWatchdog Interrupt\r\n" );
118     }
119     ADC_ClearITPendingBit( ADC1, ADC_IT_AWD );
120 }
121 ****
122 * Function Name : main
123 * Description   : Main program.
124 * Input         : None
125 * Return        : None
126 ****
127 int main(void)
128 {
129     u16 ADC_val;
130
131     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_4);
132
133     Delay_Init();
134     USART_Printf_Init(115200);
135     printf("SystemClock=%d\r\n",SystemCoreClock);
136
137     ADC_Function_Init();
138
139     while(1)
140     {
141         ADC_val = Get_ADC_Val( ADC_Channel_2 );
142         Delay_Ms(500);
143         printf( "%d\r\n", ADC_val );
144         Delay_Ms(2);
145     }
146 }
147
148
149
150
151
152
153
154

```

图5.5.2.1

5.5.3 变量

鼠标悬停在源码中变量之上会显示详细信息，或者选中变量，然后右键单击 `add watch expression`，弹出如图5.5.3.1:

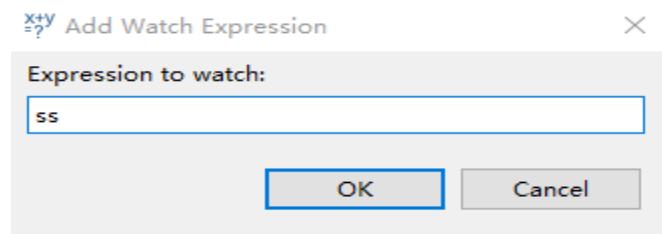


图 5.5.3.1

填写变量名，或者直接点击OK，将刚才选中的变量加入到弹出的:

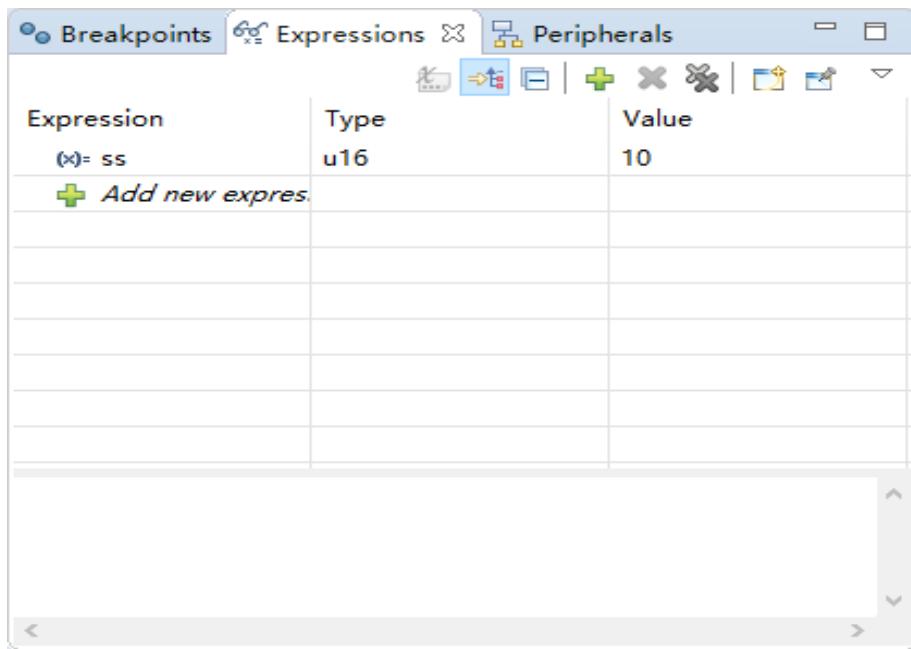
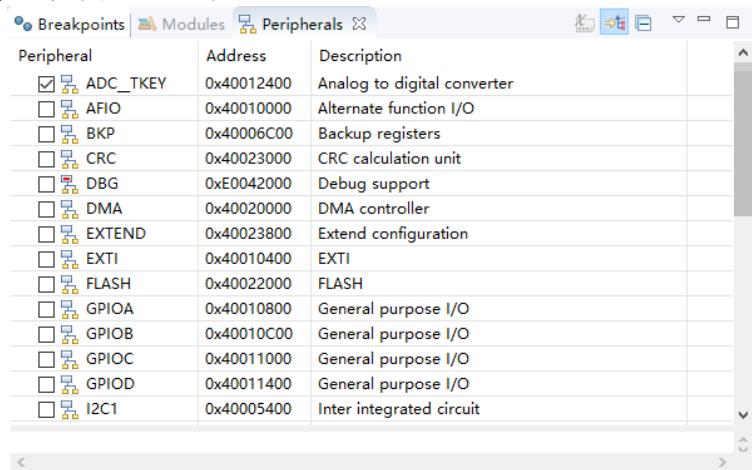


图 5.5.3.2

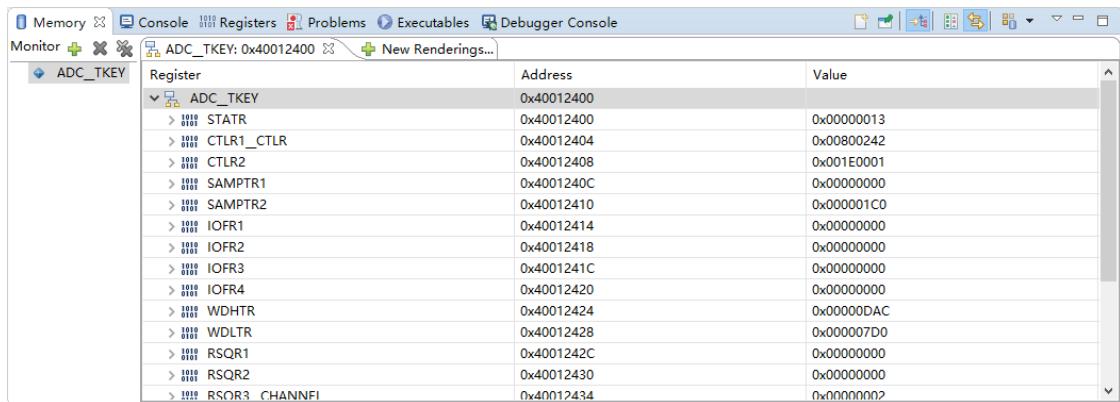
5.5.4 外设寄存器

在IDE界面左下角Peripherals 界面显示有外设列表，勾选外设则在Memory窗口显示其具体的寄存器名称、地址、数值。如图5.5.4.1和5.5.4.2。



Peripheral	Address	Description
ADC_TKEY	0x40012400	Analog to digital converter
AFIO	0x40010000	Alternate function I/O
BKP	0x40006C00	Backup registers
CRC	0x40023000	CRC calculation unit
DBG	0xE0042000	Debug support
DMA	0x40020000	DMA controller
EXTEND	0x40023800	Extend configuration
EXTI	0x40010400	EXTI
FLASH	0x40022000	FLASH
GPIOA	0x40010800	General purpose I/O
GPIOB	0x40010C00	General purpose I/O
GPIOC	0x40011000	General purpose I/O
GPIOD	0x40011400	General purpose I/O
I2C1	0x40005400	Inter integrated circuit

图 5.5.4.1



Register	Address	Value
ADC_TKEY	0x40012400	0x00000013
STATR	0x40012400	0x00000242
CTRL1_CTRL	0x40012404	0x001E0001
CTRL2	0x40012408	0x00000000
SAMPTR1	0x4001240C	0x00000000
SAMPTR2	0x40012410	0x000001C0
IOFR1	0x40012414	0x00000000
IOFR2	0x40012418	0x00000000
IOFR3	0x4001241C	0x00000000
IOFR4	0x40012420	0x00000000
WDHTR	0x40012424	0x00000DAC
WDLTR	0x40012428	0x000007D0
RSQR1	0x4001242C	0x00000000
RSQR2	0x40012430	0x00000000
RSQR3_CHANNFI	0x40012434	0x00000002

图 5.5.4.2

5.6 下载

1. RISC-V 内核芯片代码下载:

点击快捷工具栏中的  箭头，弹出工程烧录配置窗口，如图5.6.1：

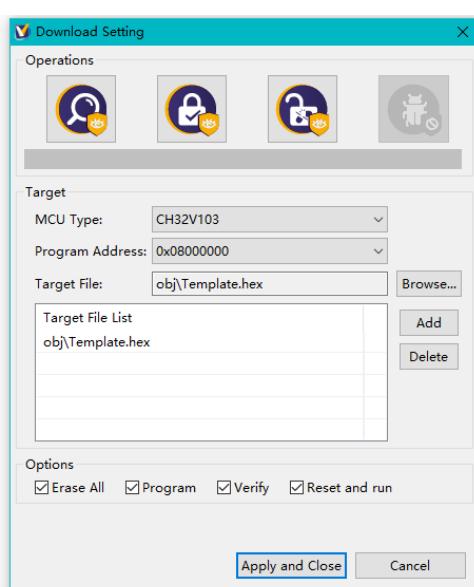


图 5.6.1

1. MCU Type	芯片型号
2. Program address	编程地址
3. EraserAll	全擦
4. Program	编程
5. Verify	校验
6. Reset and run	复位后运行



: 针对 CH32V103 型号, 查询设备读保护状态。



: 针对 CH32V103 型号, 使能设备读保护状态。



: 针对 CH32V103 型号, 解除设备读保护状态。

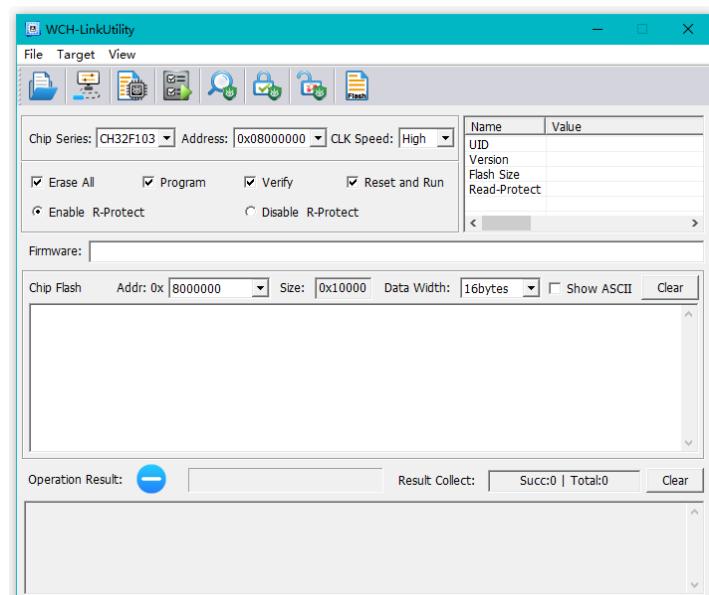


: 针对 CH56x/CH57x 型号, 禁止两线调试接口。

点击“Apply and Close”, 保存烧录配置。设置完毕后当需要进行烧录时, 直接点击工具栏 图标或在资源管理器菜单点击右键“Download”选项, 即可进行代码烧录, 结果显示在 Console 中。

2. ARM 内核芯片代码下载:

点击快捷工具栏或资源区右键菜单中的 按钮, MRS 会检查当前 WCH-Link 是否连接, 且如果其不是最新的固件版本, 会自动进行固件升级操作, 反之则会直接弹出 ARM 内核工程烧录工具窗口:



注：ARM 烧录工具使用详见本手册结尾附录一。

5.7 工程导出为模板

选中一个工程，点击右键菜单“Save As Project Template”，如图 5.7.1 所示：

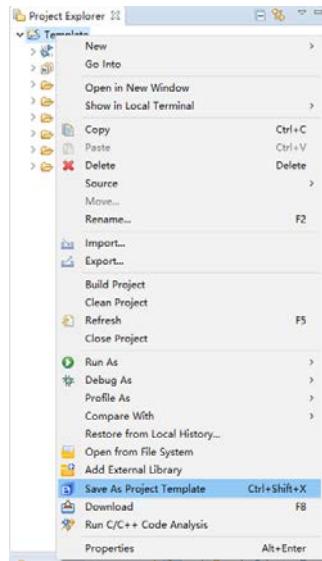


图 5.7.1

弹出导出模板的配置界面，如图 5.7.2 所示。

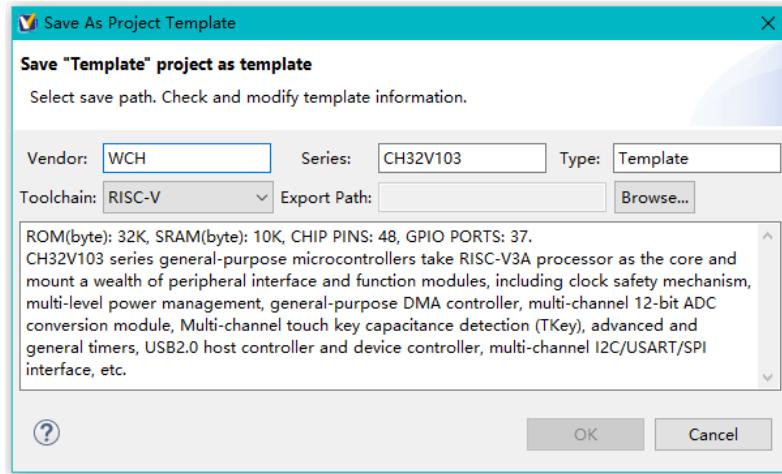


图 5.7.2

1. Vendor 厂商

2. Series 芯片系列

3. Type 芯片型号

4. Toolchain 工具链

5. Export 模板包导出的位置

窗口下部位置可填入芯片描述。点击OK完成工程模板导出。

5.8 工程模板管理

点击主菜单“Project->Template Management”选项，弹出模板管理窗口，如图 5.8.1 所示：

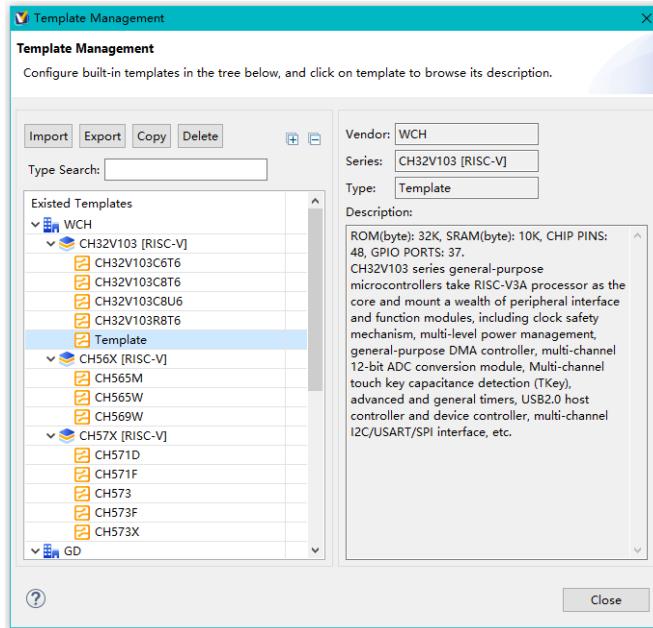


图 5.8.1

1. Import	导入工程模板
2. Export	导出工程模板
3. Copy	复制工程模板
4. Delete	删除工程模板

可以在Type Search内填入字段快速过滤芯片型号。修改后的模板库在新建MounRiver工程时自动刷新，如图5.8.2所示。

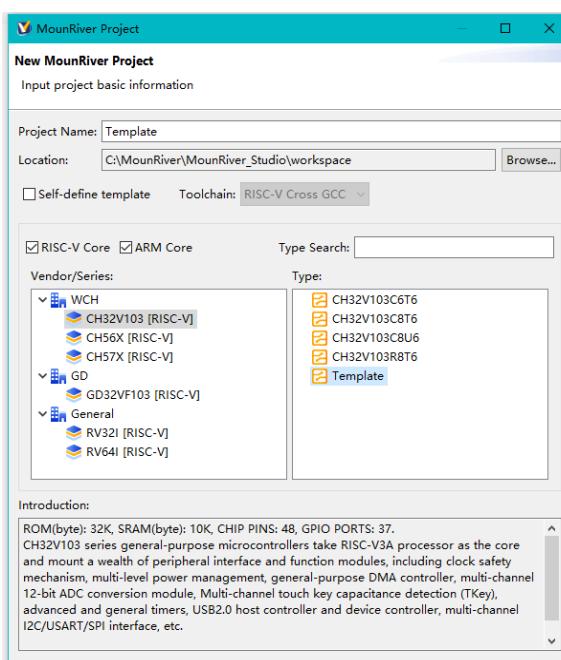


图5.8.2

六、快速问答

Q1：窗口显示框如何恢复默认状态？

A1：点击主菜单Window->Reset View to Defaults，在弹出的对话框中点击OK。

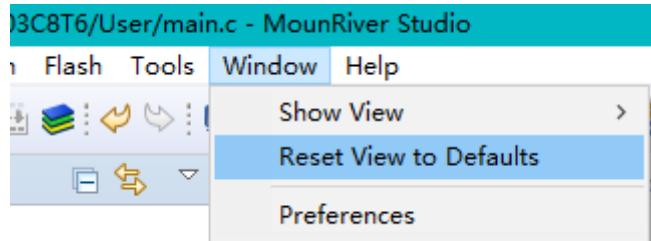


图 6.1

Q2：编译工程，提示“riscv-none-embed-gcc: not found”？

A2：点击工具栏 按钮，在弹出的对话框中点击左侧 Global RISC-V Toolchains Paths 选项，再点击右侧界面 Toolchain folder 项旁边 Browse 按钮，选择 MRS 安装目录\toolchain\RISC-V Embedded GCC\bin，点击“Apply and Close”即可正确编译。

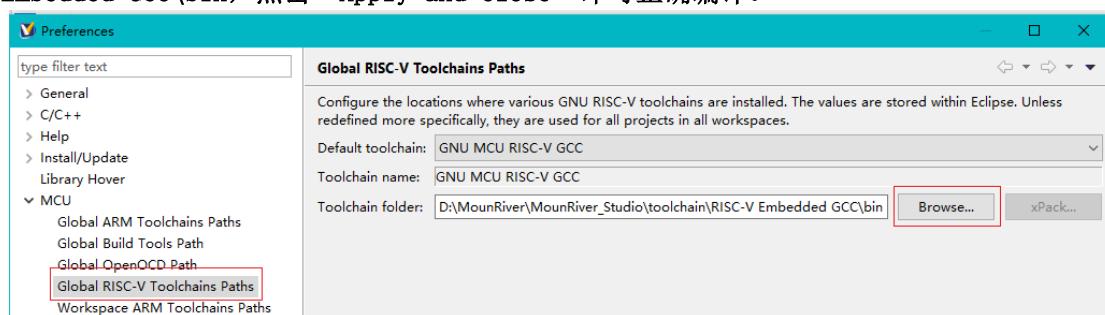


图 6.2

Q3：调试工程时，查看不到详细的寄存器参数监控信息

A3：点击工具栏调试按钮旁边三角形箭头 ，在下拉菜单中选择 Debug Configurations，弹出调试配置窗口。选中左侧 XXX (当前工程名) Built-in Launch 选项，点击右侧 SVD Path 标签页，在 File path 编辑框中填入当前工程对应的.svd文件路径。

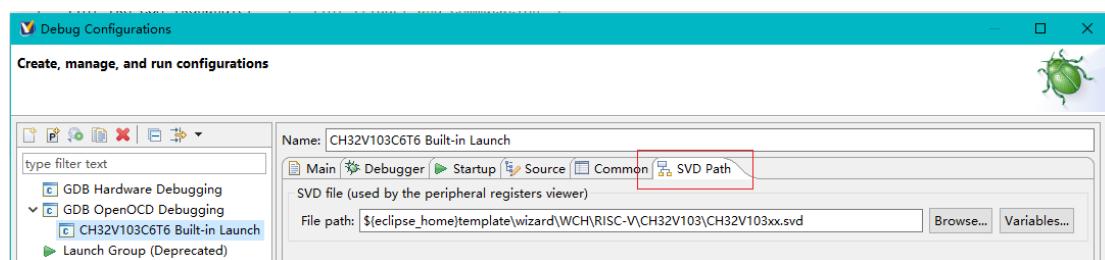


图 6.3

Q4、如何与历史版本文件对比?

A4、选中想要对比的文件，鼠标右键单击，选择Compare With —>Local History，如图6.4.1，在弹出的 History窗口中选择想要对比的历史版本鼠标左键双击，显示如图6.4.2

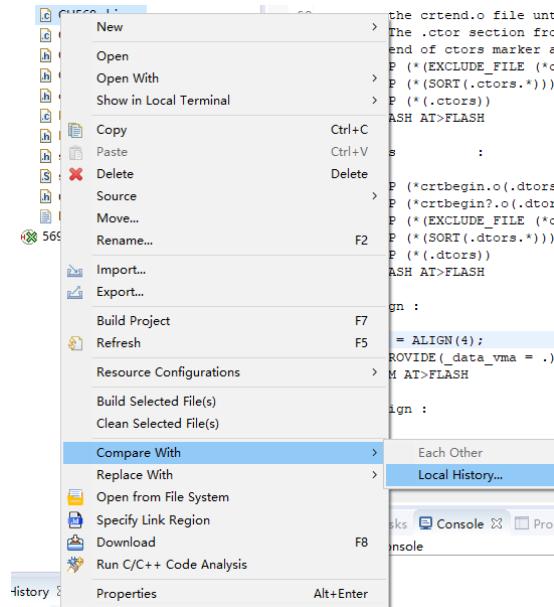


图 6.4.1

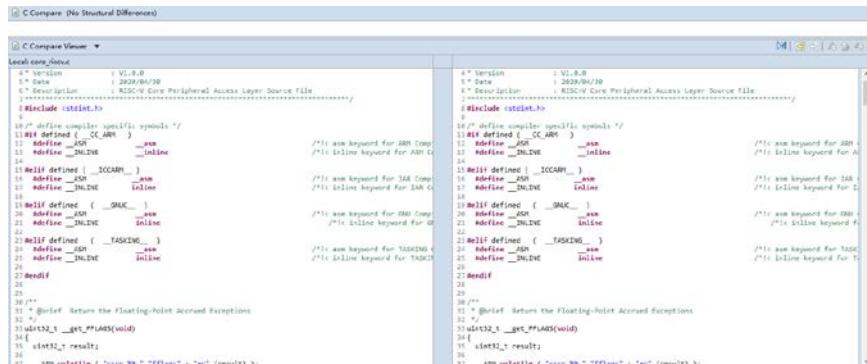


图 6.4.2

Q5、如何添加工程目录以外的文件?

A5、在工程目录空间选中工程，鼠标右键单击，显示如图 6.5.1，选择 Add->Linked Folder，显示如图 6.5.2，选择 Browse...，去添加外部文件，点击 confirm 确认。

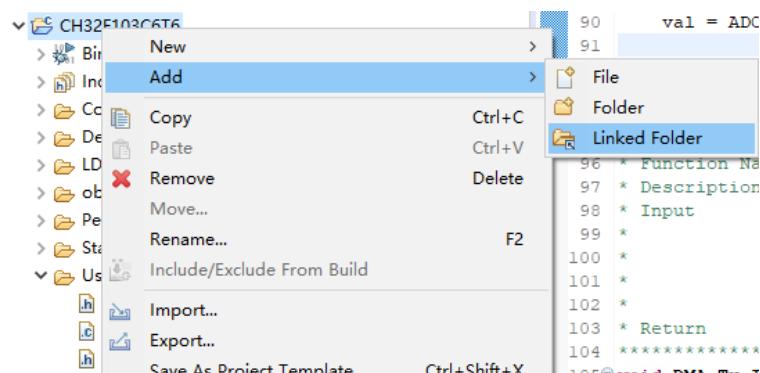


图 6.5.1

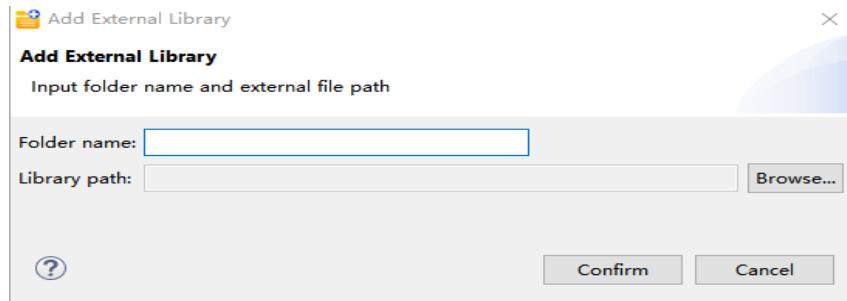


图 6.5.2

Q6、编译选中文件的功能入口从资源区右键菜单消失怎么办？

A6、点击主菜单 Project, 去掉下拉菜单中 Build Automatically 菜单项的勾选, 再次选中资源区文件点击右键, 菜单中 Build Selected File(s) 选项重新出现。

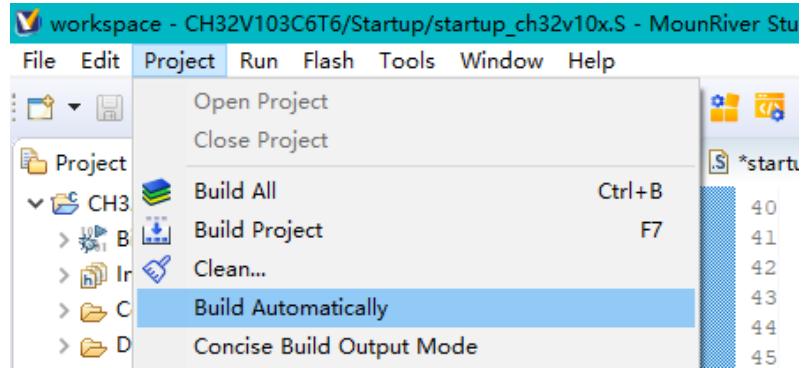


图 6.6

Q7、如何查看 MRS 运行日志？

A7、点击主菜单 “Help->Open Workbench Running Log”

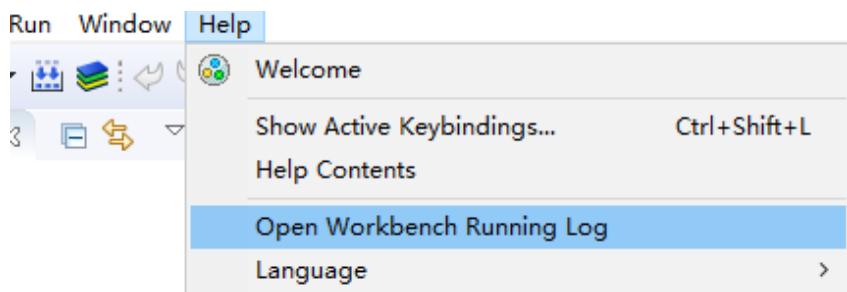


图 6.7

七、更新

7.1 在线更新

选中菜单栏 Help->Check Updates, 如图 7.1, 自动进行下载、更新, 更新完毕, 弹窗提示成功

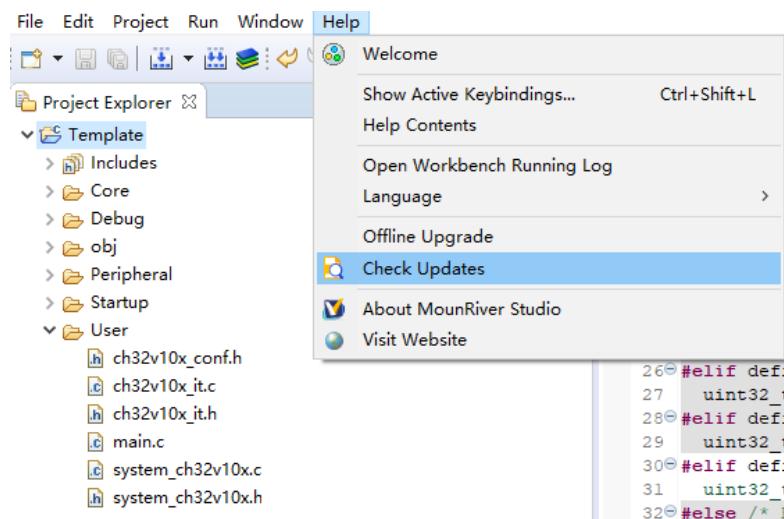


图 7.1

7.2 离线更新

选中菜单栏 Help->Offline Upgrade, 如图 7.2.1, 在图 7.2.2 中选择离线安装包。点击 Update, 自动升级。

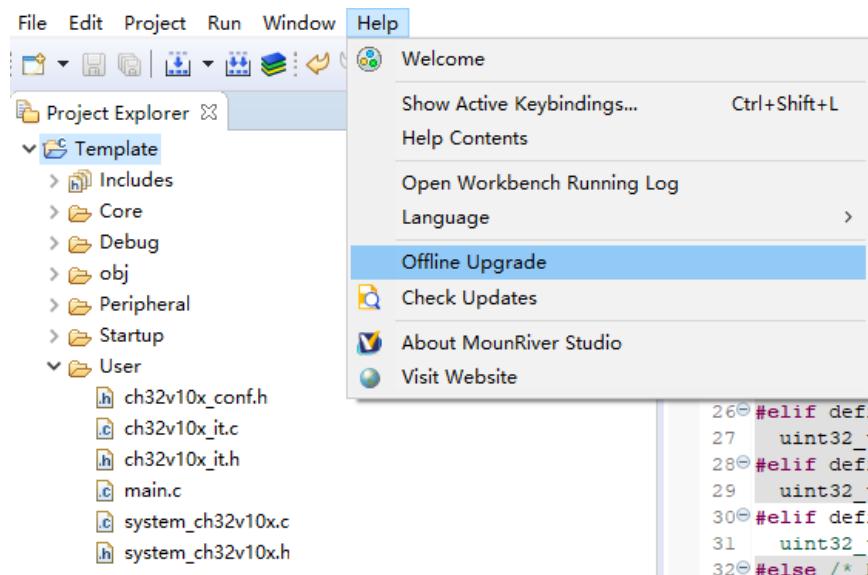


图 7.2.1

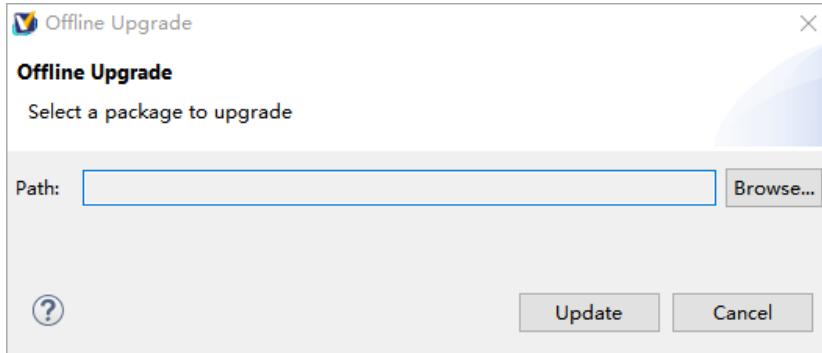


图 7.2.2

八、联系我们

如果您在使用MounRiver Studio过程中有任何反馈和建议，请通过下方邮箱或者网站联系我们

Email: support@mounriver.com

Website: <http://mounriver.com>

附录一：WCH-Link Utility.exe使用说明