

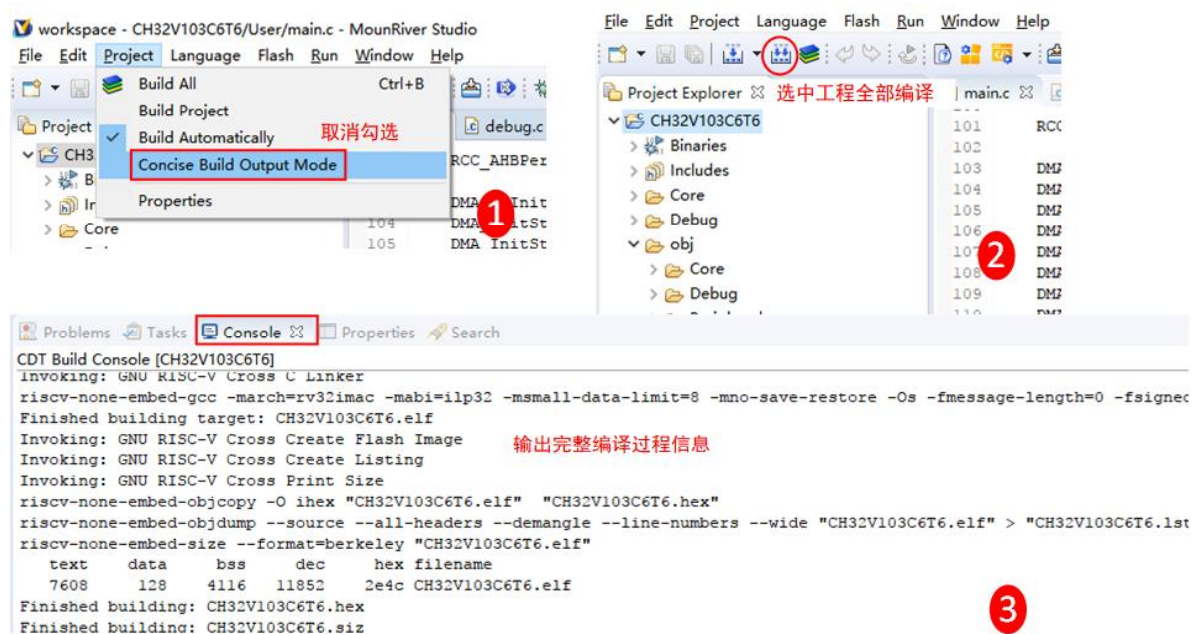
工程配置选项基础

编译过程


使用 MRS 软件时，从代码编写到最后输出目标文件过程，与其它软件编译原理基本相同。简单的讲，包括编译、链接、转换。

- 编译：MRS 软件使用的是 gcc 编译器，将每个 c/c++ 和汇编源文件编译成对应名称的对象文件 “.o” 结尾，其主要内容是从源文件编译得到的机器码，包含了代码、数据、调试信息等。
- 链接：使用 gcc 链接器，把上述生成的各项.o 文件及库文件（如果有）链接成一个映像文件 “.elf” 文件以及存储映像 “.map” 文件。
- 转换：通过 objdump、objcopy 等工具，由 “.elf” 文件转换导出所需的 “.lst” 文件，目标文件 “.hex” 或 “.bin”，其中目标文件可以直接下载到芯片上运行。

MRS 的默认配置中，为了简化内容，将整个编译转换过程信息屏蔽了。如果有人感兴趣，可以通过下面方式调出。



MRS 软件的工程模板中默认的配置选项

选中工程，点击工具栏上的 “” 图标，可以调出工程配置选项窗口。

1. 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。

Architecture 是架构选择，默认使用 RV32I 表示 RISC-V 32 位整数指令架构；其他选项还包括：乘除法扩展指令（RVM），原子指令扩展（RVA），浮点指令扩展（None/RVF/RVFD/RVFDQ），压缩指令扩展（RVC）。

Integer ABI 为 RISC-V 应用程序整数二进制接口。

Floating point ABI 为 RISC-V 应用程序浮点数二进制接口，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。

Tuning 由微架构优化给定处理器的输出，默认是 **rocket**。

Code model，采用的代码模型-**mcmmodel=medlow**，表示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。

Small data limit 设置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** 取决于目标处理器是否支持内存的非对齐访问。

2. Optimization

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，下面说几个常用的。

-O0：无优化(默认)

-O 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。

3. Warnings

这里可以设置 **gcc** 的警告参数选项，以下列出部分：

-fsyntax-only：检查代码中的语法错误，但除此之外不要做任何事情。

-fmax-errors= n：将错误消息的最大数量限制为 n，此时 **gcc** 会缓存，而不是尝试继续处理源代码。如果 n 为 0（默认值），则生成的错误消息数量不受限制。如果还指定了 **-Wfatal-errors**，则重大错误优先于此选项。

-w：禁止所有警告消息。

-Werror：将警告变为错误提示。

-pedantic：允许发出 ANSI/ISO C 标准所列出的所有警告。

-pedantic-errors：允许发出 ANSI/ISO C 标准所列出的警告当做错误。

4. Debugging

默认设置 GDB 调试信息等级为 **-g**。

注意生成静态库时将红框中调为 **None**，否则生成的库会包含调试信息，导致库文件过大。

5. 汇编编译选项

Preprocessor：预处理器选项，定义或取消“宏”，预处理。

Includes：添加参与工程编译的汇编文件路径或文件。

注意：为了方便工程在不同电脑或路径下使用，建议使用相对路径（相对当前工程位置）。

Warnings：设置编译警告条件。

Miscellaneous：其他选项，可扩展其他的编译选项。

6. C 编译选项

Preprocessor：预处理器选项，定义或取消“宏”，预处理。

Includes：添加参与工程编译的汇编文件路径或文件。

注意：为了方便工程在不同电脑或路径下使用，建议使用相对路径（相对当前工程位置）。

Optimization：扩展优化编译选项，需要自己添加

Warnings：设置编译警告条件。

MRS 对 RSIC-V 内核芯片编译支持所采用的工具在安装目录下的“toolchain”文件夹内，里面还有更多的其他工具。可以通过命令行查看每个工具及参数使用方法。

名称

- ☐ riscv-none-embed-as.exe
- ☐ riscv-none-embed-c++.exe
- ☐ riscv-none-embed-c++filt.exe
- ☐ riscv-none-embed-cpp.exe
- ☐ riscv-none-embed-elfedit.exe
- ☐ riscv-none-embed-g++.exe
- ☐ riscv-none-embed-gcc.exe
- ☐ riscv-none-embed-gcc-8.2.0.exe
- ☐ riscv-none-embed-gcc-ar.exe
- ☐ riscv-none-embed-gcc-nm.exe
- ☐ riscv-none-embed-gcc-ranlib.exe
- ☐ riscv-none-embed-gcov.exe
- ☐ riscv-none-embed-gcov-dump.exe
- ☐ riscv-none-embed-gcov-tool.exe
- ☐ riscv-none-embed-gdb.exe
- ☐ riscv-none-embed-gdb-add-index-p
- ☐ riscv-none-embed-gdb-add-index-py
- ☐ riscv-none-embed-gdb-py.exe
- ☐ riscv-none-embed-gprof.exe
- ☐ riscv-none-embed-ld.bfd.exe

```
D:\MRS IDE\MounRiver_Studio\toolchain\RISCV-V Embedded GCC\bin>riscv-none-embed-gcc.exe --help
Usage: riscv-none-embed-gcc.exe [options] file...
Options:
  -pass-exit-codes      Exit with highest error code from a phase.
  --help               Display this information.
  --target-help         Display target specific command line options.
  --help={common|optimizers|params|target|warnings|[' ']} (joined|separate|undocumented)) [...].
                       Display specific types of command line options.
  (Use '-v --help' to display command line options of sub-processes).
  --version            Display compiler version information.
  -dumpspecs            Display all of the built in spec strings.
  -dumpversion          Display the version of the compiler.
  -dumpmachine          Display the compiler's target processor.
  -print-search-dirs    Display the directories in the compiler's search path.
  -print-libgcc-file-name Display the name of the compiler's companion library.
  -print-file-name=<lib> Display the full path to library <lib>.
  -print-prog-name=<prog> Display the full path to compiler component <prog>.
  -print-multiarch      Display the target's normalized GNU triplet, used as
                       a component in the library path.
  -print-multi-directory Display the root directory for versions of libgcc.
  -print-multi-lib       Display the mapping between command line options and
                       multiple library search directories.
  -print-multi-os-directory Display the relative path to OS libraries.
  -print-sysroot         Display the target libraries directory.
  -print-sysroot-headers-suffix Display the sysroot suffix used to find headers.
  -Va,<options>          Pass comma-separated <options> on to the assembler.
  -Vp,<options>          Pass comma-separated <options> on to the preprocessor.
```

查看gcc工具的帮助说明