

Embedded Studio 8.10d RISC-V 应用开发示例

SEGGER Embedded Studio 8.10d 同时支持 ARM 和 RISC-V 处理器，并集成了市场主流处理器相关库文件。以下内容通过LED显示例程，介绍利用 Embedded Studio 8.10d 开发RISC-V处理器 GD32VF103应用程序的过程。

1. 建立环境

软件开发和调试环境包括SEGGER Embedded Studio 8.10d, 仿真器 JLINK EDU mini (或GD-Link) 和评估板GD32VF13V-EVAL。

1.1. 软件环境

SEGGER Embedded Studio (以下简称为Embedded Studio) 版本已经更新到V10.8d, 支持Windows, MacOS和Linux三种操作。

下载地址: <https://www.segger.com/downloads/embedded-studio/>。下载页面如图 1.1所示。

选择与开发软件平台所使用的操作系统一致的版本, 点击右侧download, 下载软件。

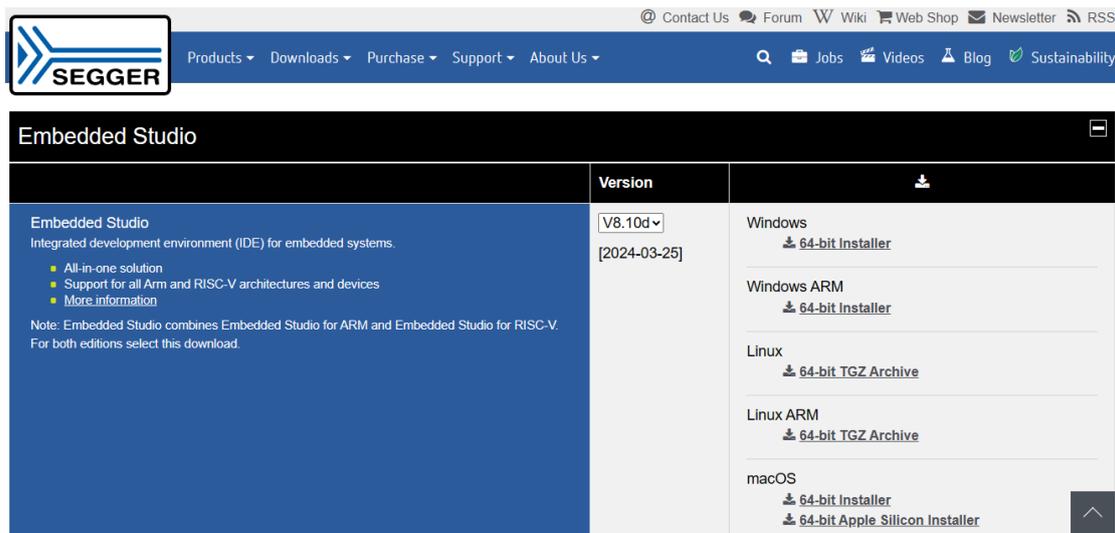


图1.1 SEGGER Embedded Studio 下载页面

下载完成后, 直接运行所下载的安装程序, 进行安装。安装过程比较简单, 在出现的所有对话框中, 直接点击next即可完成。

1.2. 硬件环境

如图 1.2 所示, GD32VF103V-EVAL 评估版采用 GD32VF103BT6 作为主控制器。评估板从 mini USB 接口或电源适配器接口输入 5V 电源。评估板提供了 JTAG 和 GD-Link 两种调试接口, 以及 LCD、LED、按键和串口等输入/输出设备。

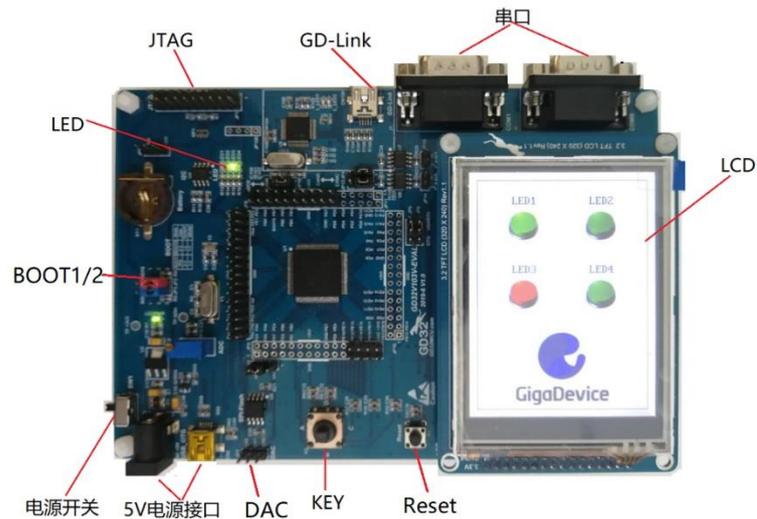


图 1.2 GD32VF103V-EVAL 评估版

JLINK EDU mini 是 SEGGER 公司为学校教育开发的一款小巧仿真器。如果图 1.3 所示，图的右侧是 microUSB 接口，与运行 Embedded Studio 的 PC 相连。图的左侧是 10 pin JTAG 接口，使用转接电缆连接到评估板的 20 pin JTAG 端。



图 1.3 JLINK EDU mini

Embedded Studio 自带 JLINK 驱动。将仿真器的 JTAG 端连接到评估板，MicroUSB 端口连接到 PC 的 USB 接口。连接评估板电源并打开电源开关，Embedded Studio 就能够将程序下载到评估板，在评估板上运行和调试程序。

接下来，通过打印“Hello RISC-V!”的简单程序，说明 Embedded Studio RISC-V 应用程序开发过程。过程可分为创建工程，设置工程属性，添加文件和编辑源程序，生成可执行程序，以及调试等步骤。

2. 创建工程

启动 Embedded Studio，建立一个新的项目 gpio_led。首先选择处理器和工程类型，如图 2.1 所示，

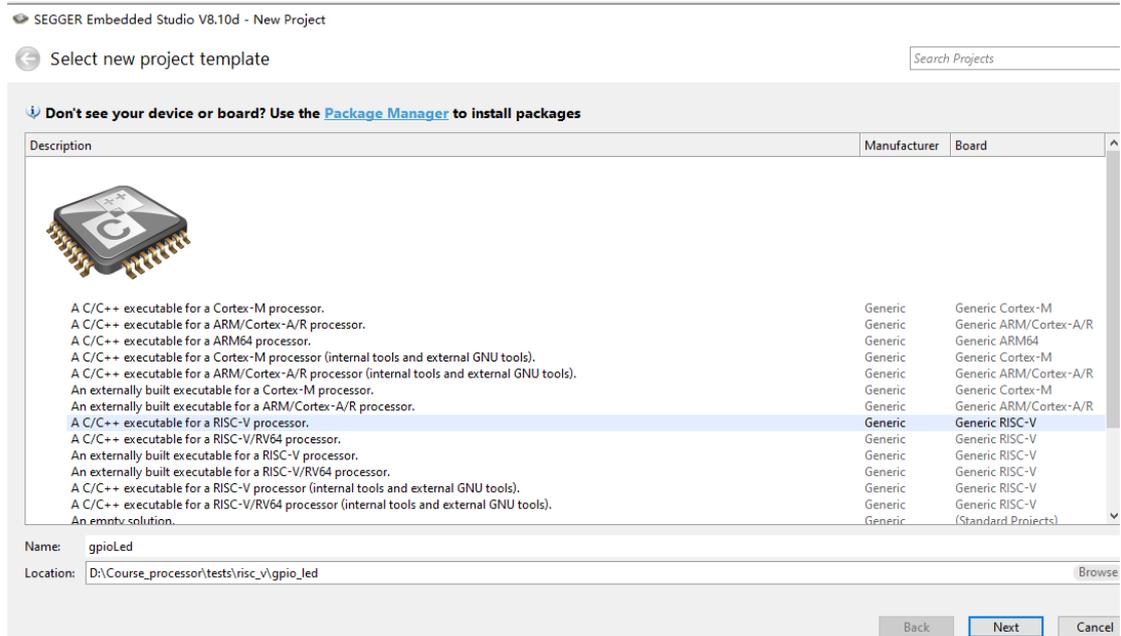


图2.1 处理器和工程类型选项

在图2.1的对话框中，选择“A C/C++ executable for RISC-V processor”，然后点击“next”，进入选择处理器型号对话框，如图2.2所示：

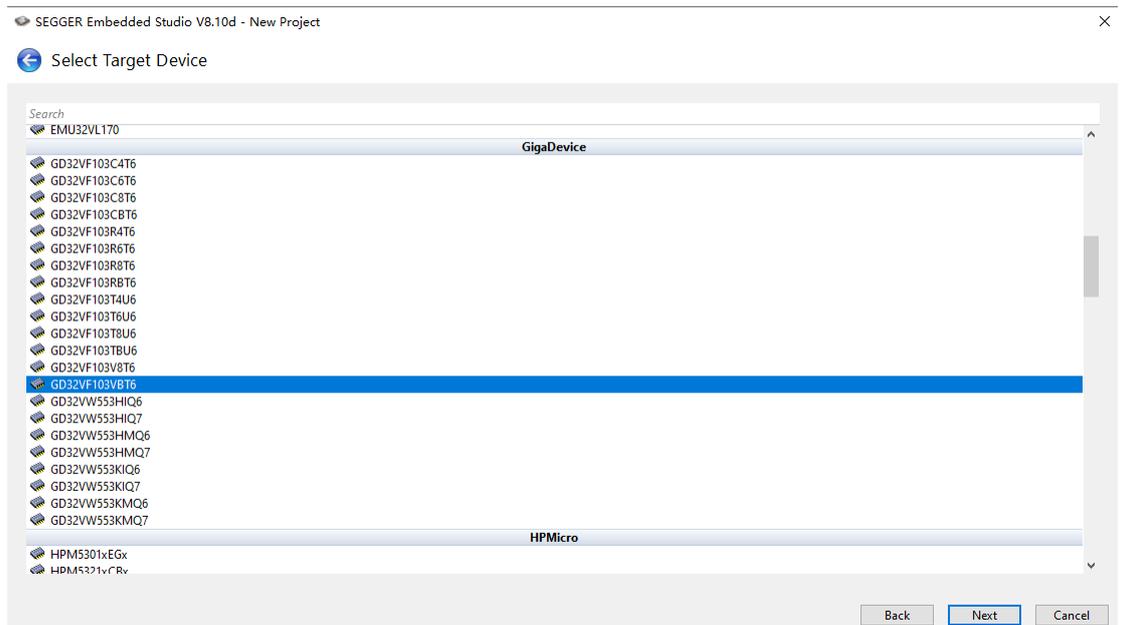


图2.2 处理器型号选项

在对话框中寻找厂商GigaDevice，选择处理器“gd32VF103VBT6”，然后点击“next”，进入工具对话框，如图2.3所示：

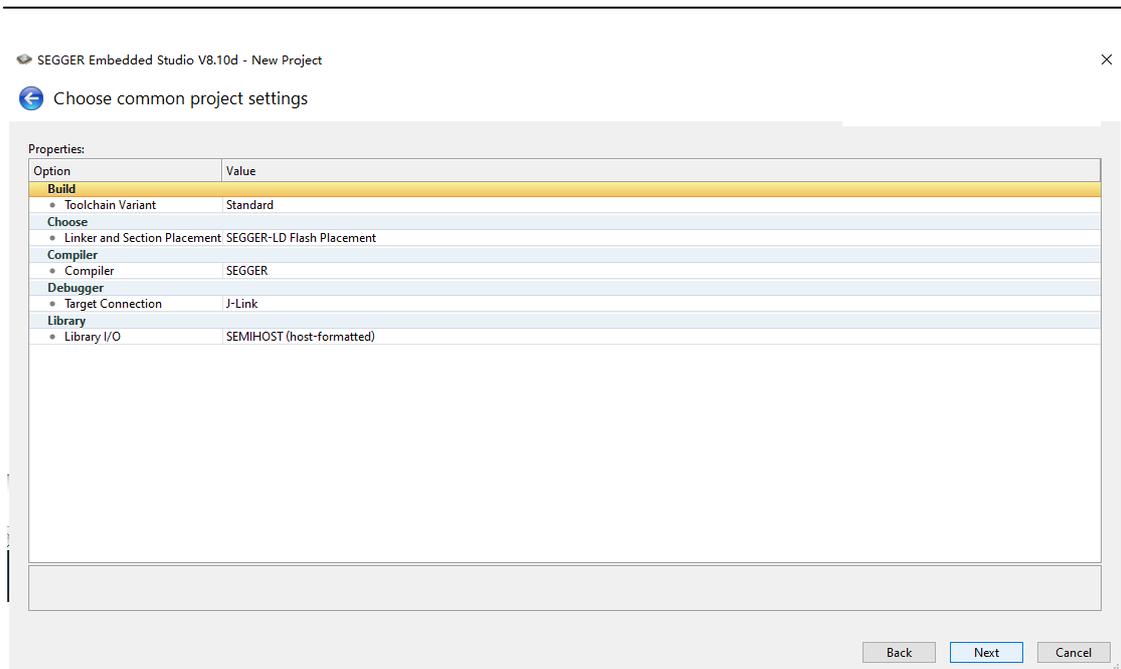


图2.3 工具选项

点击“next”，进入图2.4 对话框。

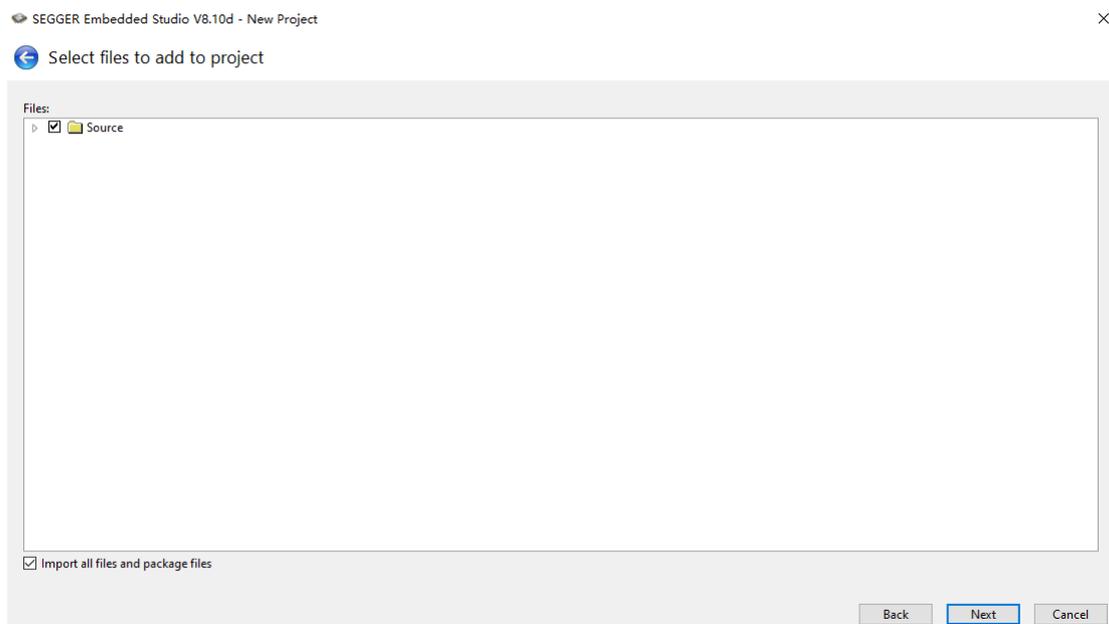


图2.4 生成对话框

点击“next” 调试和发布设置选择对话框，如图2.5:

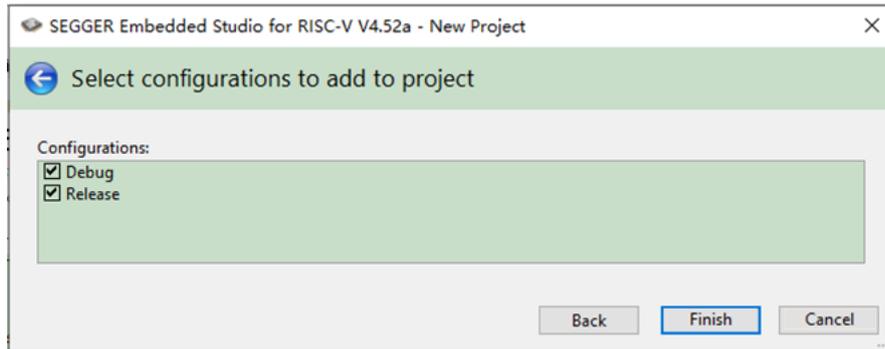


图 2.5 调试和发布设置选择对话框

点击”next“，生成 project 框架。如图 2.6

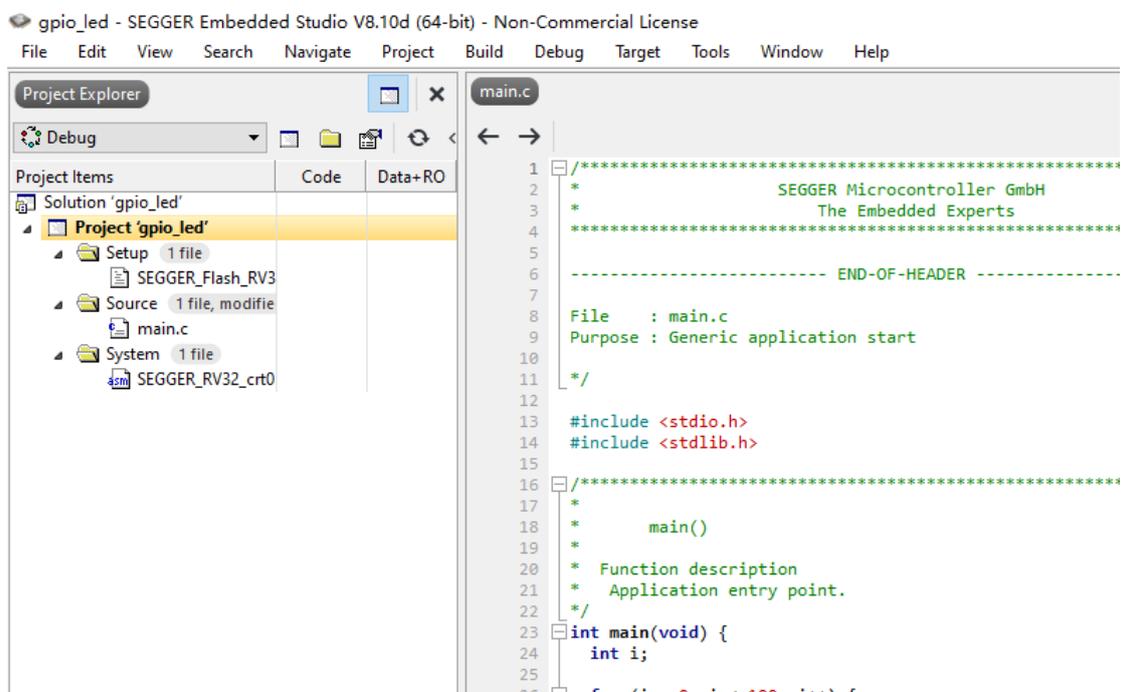


图2.6 初始工程

3. 工程设置

工程创建后，需要在 Embedded Studio 中设置程序生成工具和调试工具，工程中文件路径，目标系统存储资源，程序优化等选项。

在 Embedded Studio 中，如果解决方案中含有多个工程，可以设置方案中所有工程的选项，也可以独立设置所选工程的选项。

图 3.1 中，在工程导航栏选中“project ‘gpio_led’”，则设置当前解决方案中所有工程的选择。在菜单中选中 Project->Options，则进入选择设置（Option）对话框。

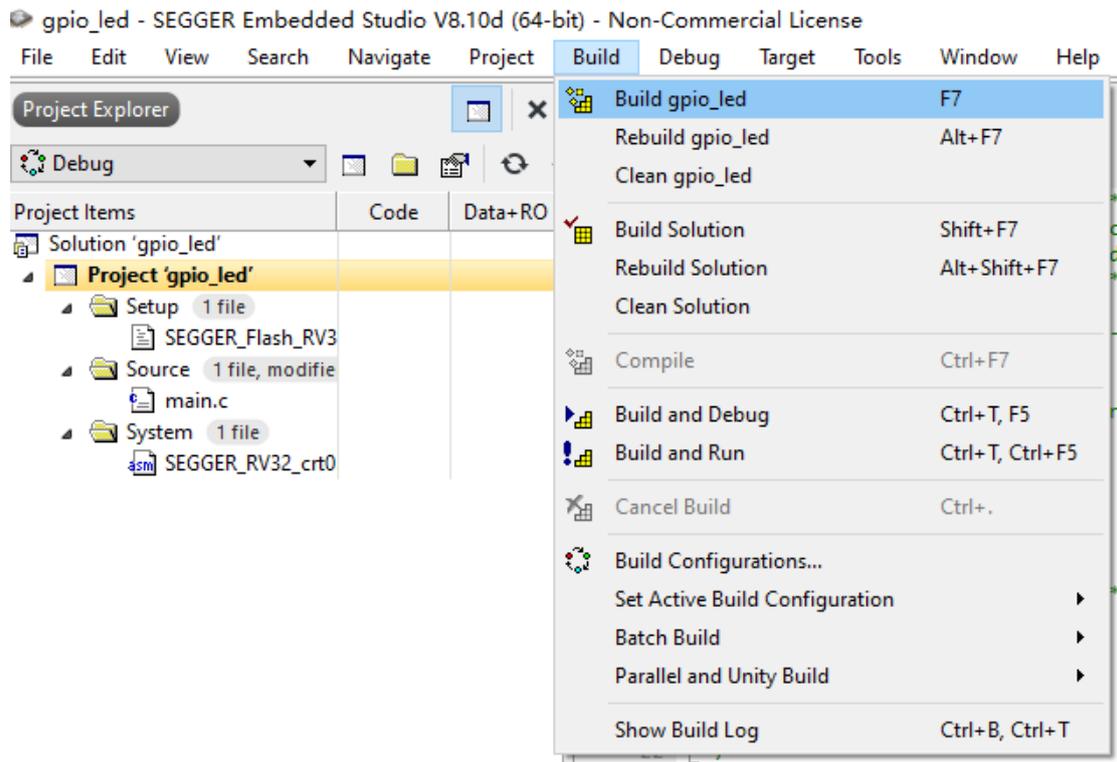


图3.1 选项设置菜单

如果在图3.1工程导航栏中选中“Project 'gpio_led' ”，则进入选项设置对话框后，只设置所选工程的选项。

3.1. 程序生成选项

图 3.2 是选择设置对话框，其中“1”是“debug”或“Release”选项，选择目标程序是调试或发行版本。两种版本工程的属性选项相同，但选项的值可以不同。

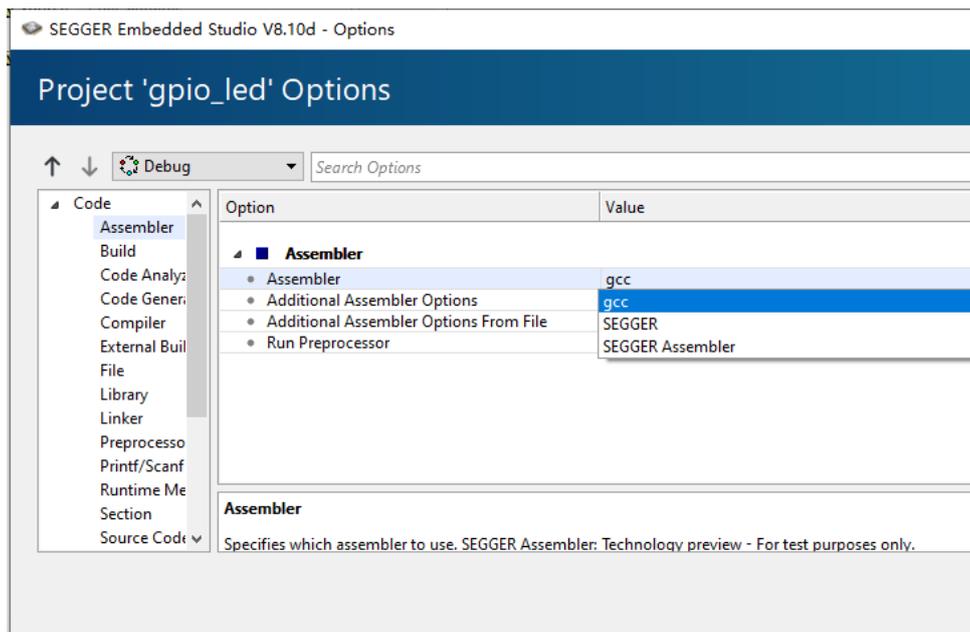


图 3.2 选择设置对话框

选项包括两个部分，“2”是代码（Code），“4”是调试（Debug）选项。代码（code）部分设置程序运行环境属性和程序生成工具，调试（Debug）部分选择调试工具，并设置属性。

如图 3.2 所示，“3”是汇编器（Assemble）选项，选择第 1 项，采用 gcc 汇编器。

代码生成（Code Generation）有多个条目，图 3.3 中列出了三个非常重要的条目。6 个程序优化（Optimization Level）选项，对应于 C 编译器不同的优化等级，缺省值为 none。5 个应用程序接口（RISC-V ABI）选项，选择不同数据类型和参数传递规则，当前工程中选 ilp32。10 个处理器指令架构（RISC-V ISA）选项，选择不同指令模块组合，当前工程选 rv32ima。

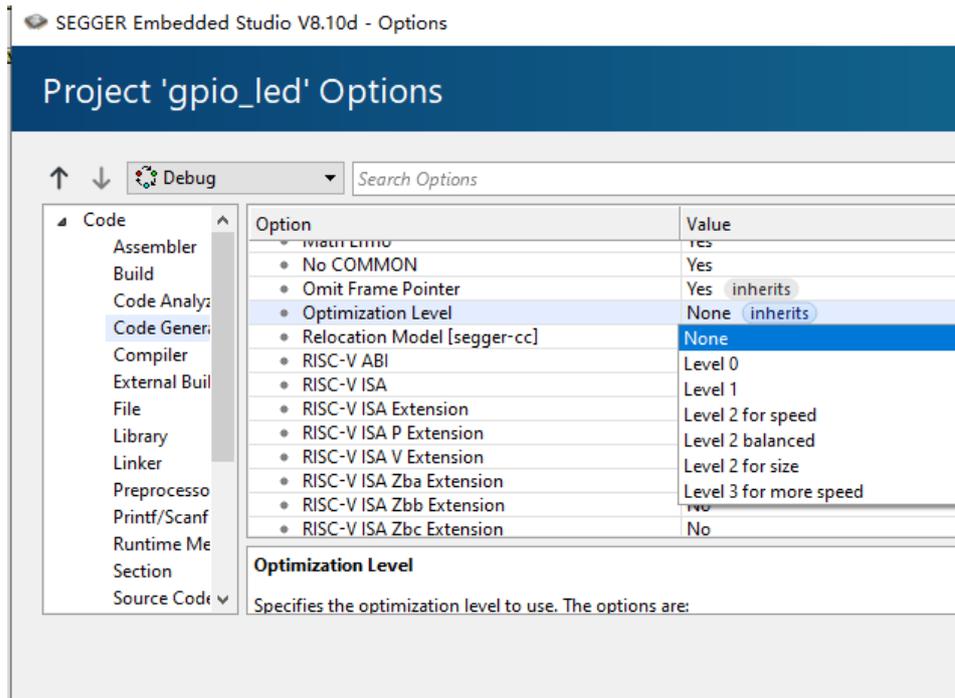


图 3.3 Code Generation 选项

在图 3.4 中，有三个编译器（Compiler）选项，选择 gcc 编译器。

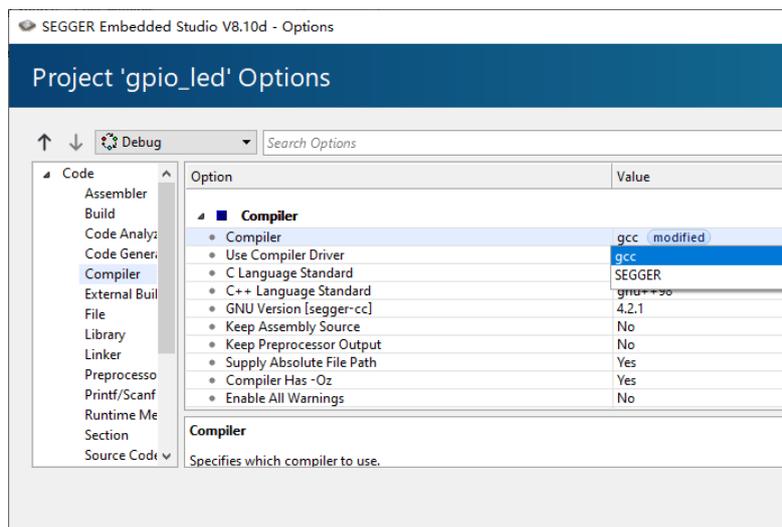


图 3.4 编译器选项

图 3.5 是链接器 (Linker) 的设置内容, 可以添加用户定义的链接脚本文件 (Use Manual Linker Script), 段放置文件 (Section Placement File) 等。链接器必须指定程序入口 (Entry Point)。

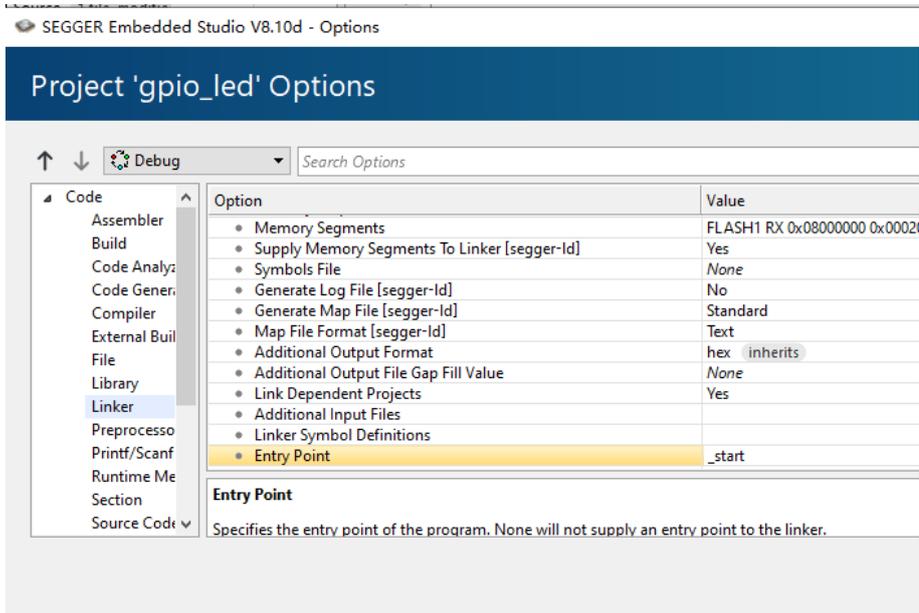


图 3.5 链接器选项

在图 3.6 中, 设置程序运行时变量和动态存储空间, 栈 (stack) 和堆 (heap) 的大小。

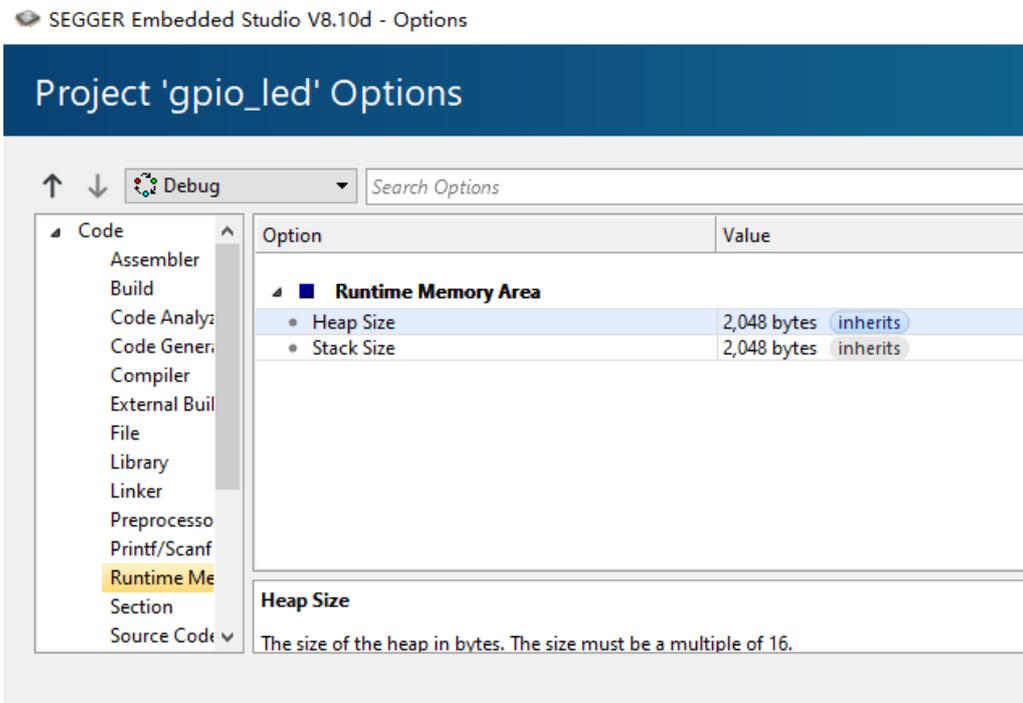


图 3.6 运行时内存分配

3.2. 调试设置

如图 3.7 所示, 调试设置中最重要的选择是连接对象。如果没有外接开发板, 则选择

Simulator，用模拟器调试程序。如果需要连接开发板，在目标板上运行和调试程序，则选择 J-Link 或 GDB Server。

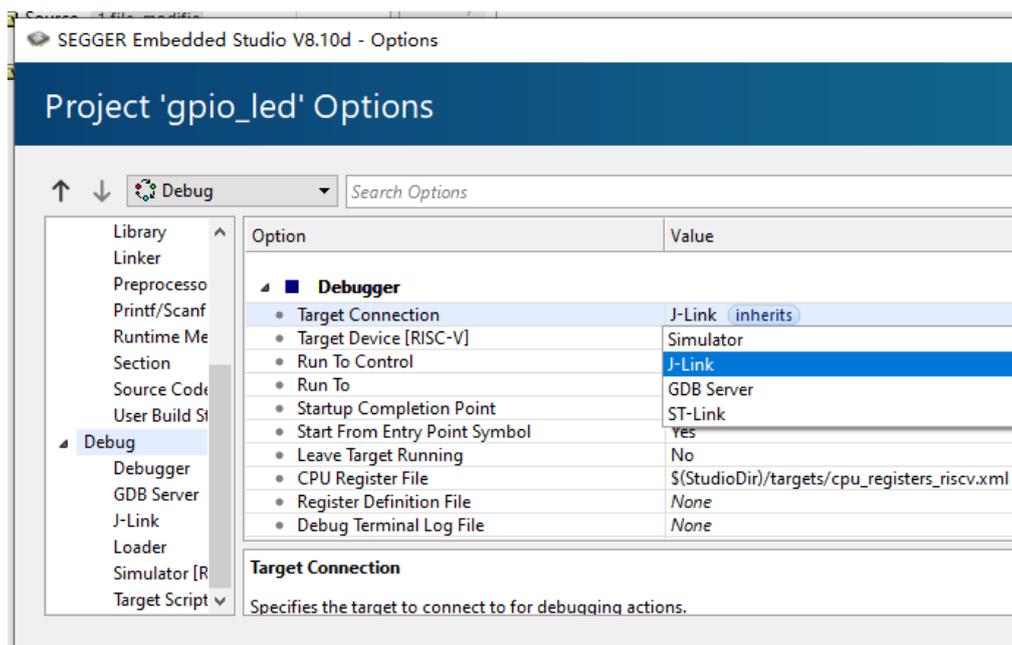


图 3.7 调试器设置

完成代码选择和调试器设置后，编写源程序，生成可执行程序，并进行调试。

4. 添加工程文件

在 Embedded Studio 创建的示例工程“gpio_led”中添加 gd32vf103lib 目录，将 gd32vf103 和 gd32vf103v-EVAL 函数库源文件添加到该目录下，如图 4.1 所示：

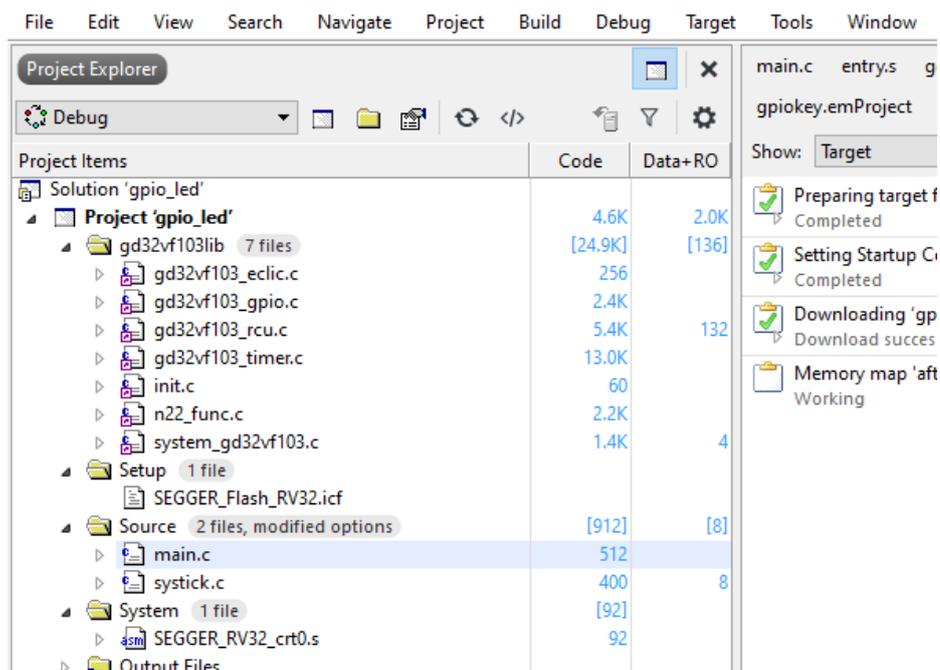


图 4.1 添加库函数源文件

将函数库的“include”路径添加到，预编译选项中。

打开“option -> code -> preprocessor -> user include directories”，如图 4.2 所示。

SEgger Embedded Studio V8.10d - Options

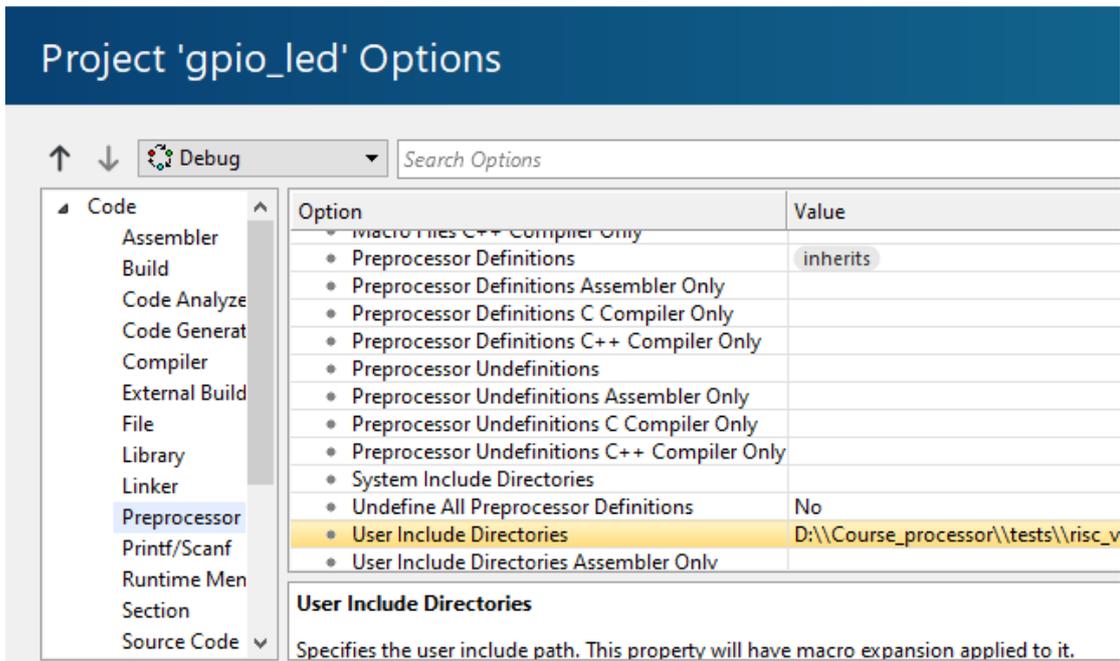


图 4.2 添加用户 include 目录

5. Build & Debug

编辑 main.c 和相关程序，点击“build”，选择“build XXX”，生成可执行目标文件。如图 5.1。

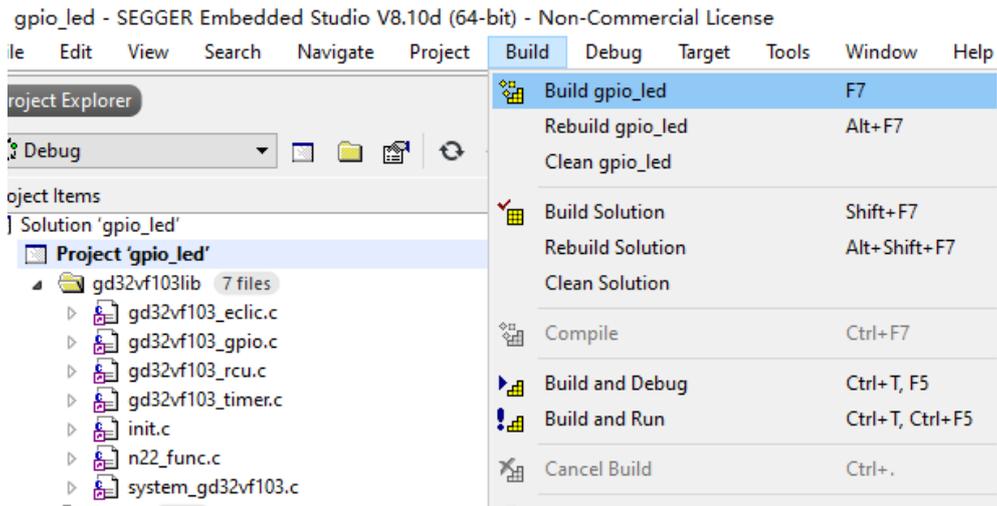


图 5.1 build 目标文件

Build 成功后，调试目标程序。首先链接调试器，如图 5.2 所示。

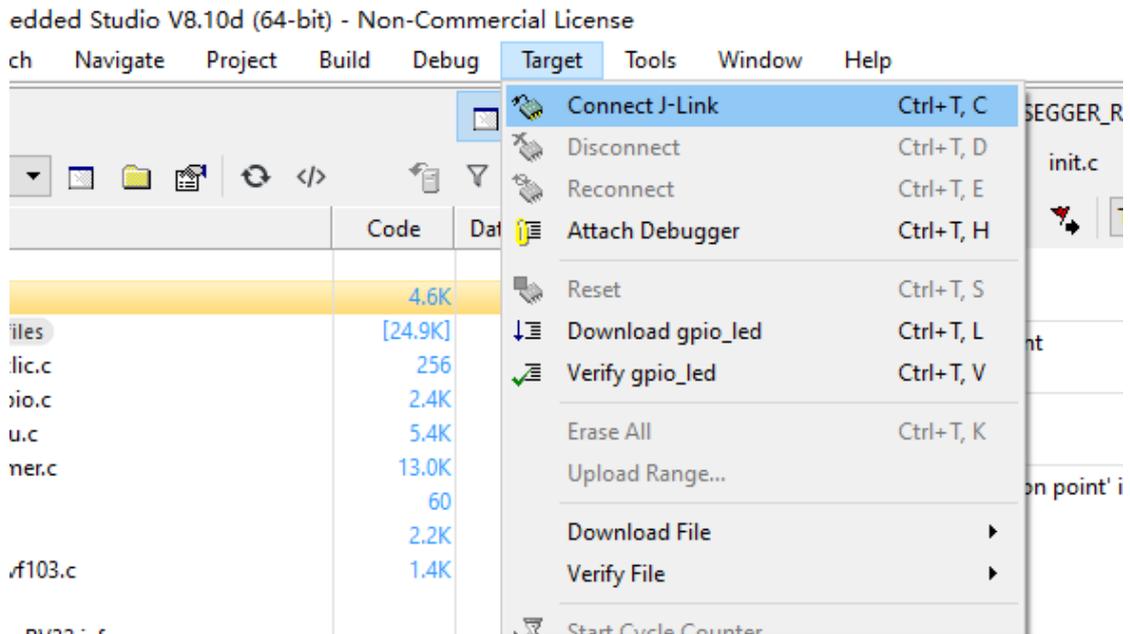


图 5.2 连接调试器

连接成功后，点击“Debug -> go”，下载程序并调试。如图 5.3 所示：

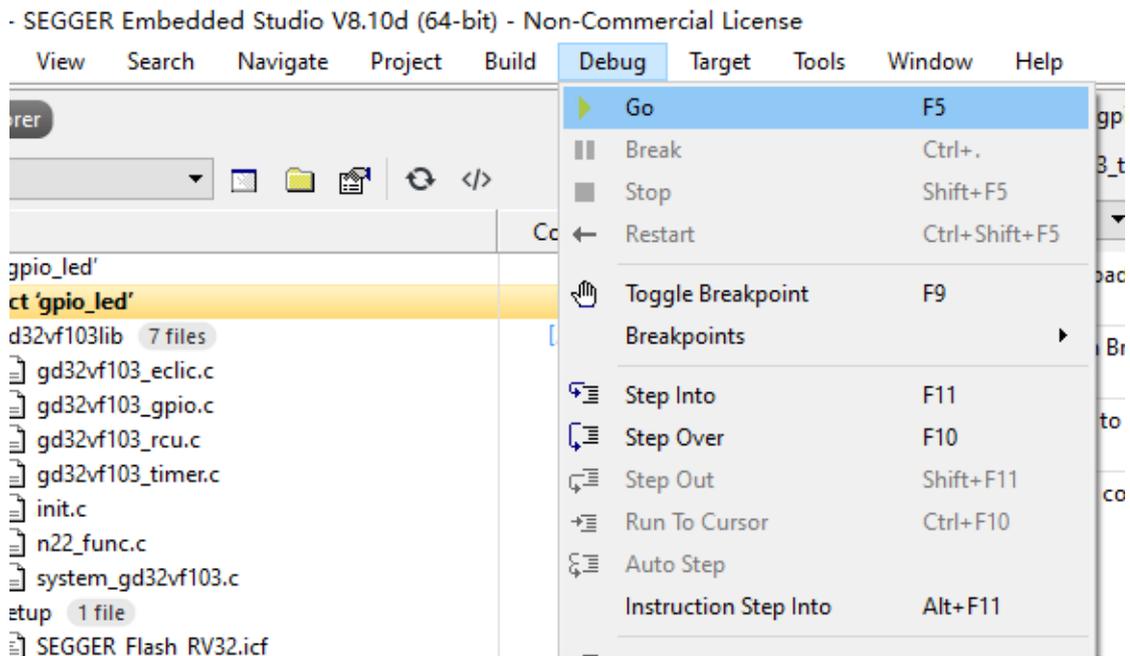


图 5.3 调试程序

至此，完成了整个工程创建、设置、编程和调试过程。