使用 Keil 新建工程模板

1. 内容

本文档描述使用 Keil C51（版本 9.52）新建工程模板的方法，并针对 Keil C51 软件的常见配置信息进行描述。

■ 注意事项：

- 在操作本文档进行的新建工程前，用户电脑需已完成 Keil C51（版本 4.95）软件的安装。
- 本文档给出的新建工程的方法是艾克姆科技推荐用户使用的一种新建工程的方法，当然新建工程的方法不是唯一的。

2. 新建工程模板

2.1. 规划工程文件存放目录

建立工程之前，我们需要先考虑一下工程文件的组织，也就是工程的存放目录。清晰的工程目录既方便我们管理工程中的各个文件，也方便日后的维护和移植。我们可以根据自己的习惯和喜好来建立自己的工程存放目录，但是也不要太随意，文件目录应该一目了然，目录中各个文件夹的名字要能准确地指示里面的内容。

下面是我们建立工程时使用的工程目录，供大家参考。

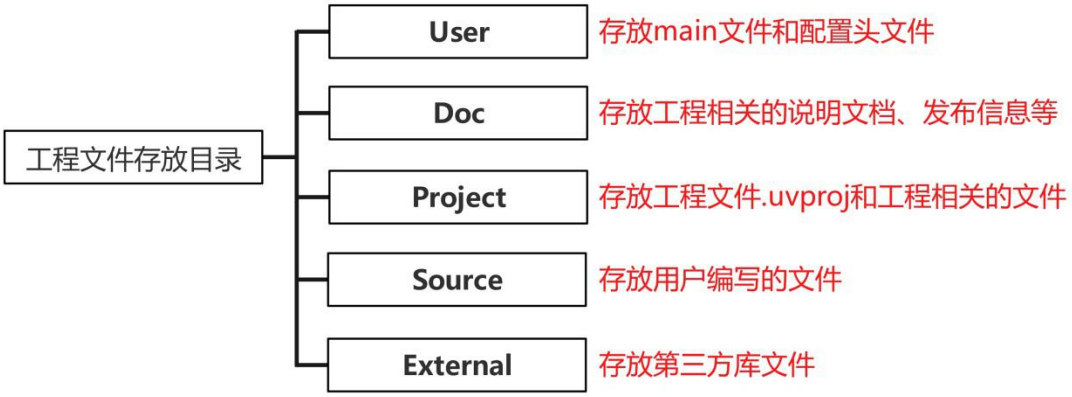


图 1：工程文件存放目录

2.2. 新建工程

因为一般的项目都会用到 LED 指示灯，所以下面我们通过新建一个流水点灯的工程来作为我们的工程模板，以此来说明工程建立和配置的步骤。工程名取为：led_blinky，工程存放到桌面。

1. 按照上文中描述的工程文件存放目录新建用于存放工程各个部件的文件夹。

1) 在桌面新建一个名字为 led_blinky 的文件夹。

图 2：新建工程文件夹

2) 在 led_blinky 文件夹下面新建如下图所示的 5 个文件夹。

图 3：建好的工程文件存放目录

3) 将 15W4KxxS4.h 头文件拷贝到 User 文件夹中。

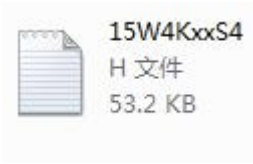


图 4：拷贝 15W4KxxS4.h 头文件

✧ 注：15W4KxxS4.h 头文件是艾克姆科技设计、适用于 STC15W4K32S4 系列单片机的头文件，用户直接复制调用即可。

4) 在 Project 文件夹中再新建 Output 文件夹和 List 文件夹。

图 5：新建 Output 文件夹和 List 文件夹

✧ 注：Output 文件夹用于存放工程编译生成的 HEX 文件等，List 文件夹用于存放工程 lst 文件。

2. 双击打开 Keil uVision4 软件。

图 6：打开 Keil C51 软件

3. 点击【Project】，在弹出的下拉菜单中选择【New uVision Project】。

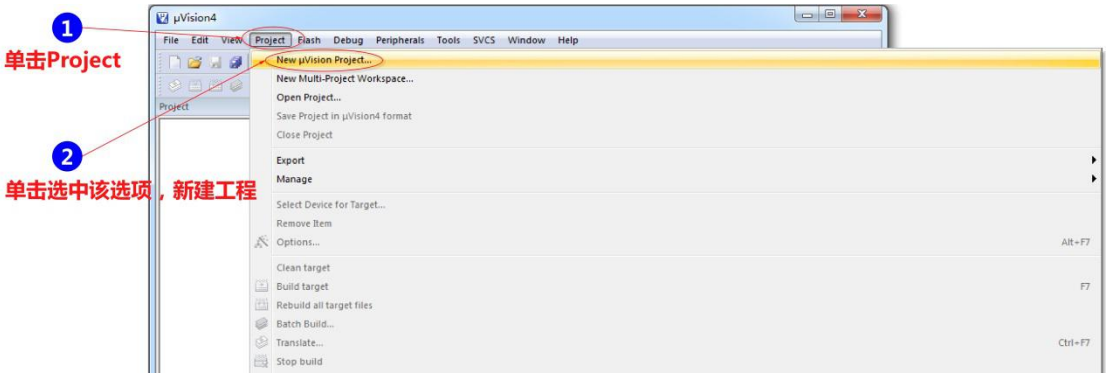


图 7：Keil C51 新建工程

4. 选择新建工程的保存路径，并给新工程命名。

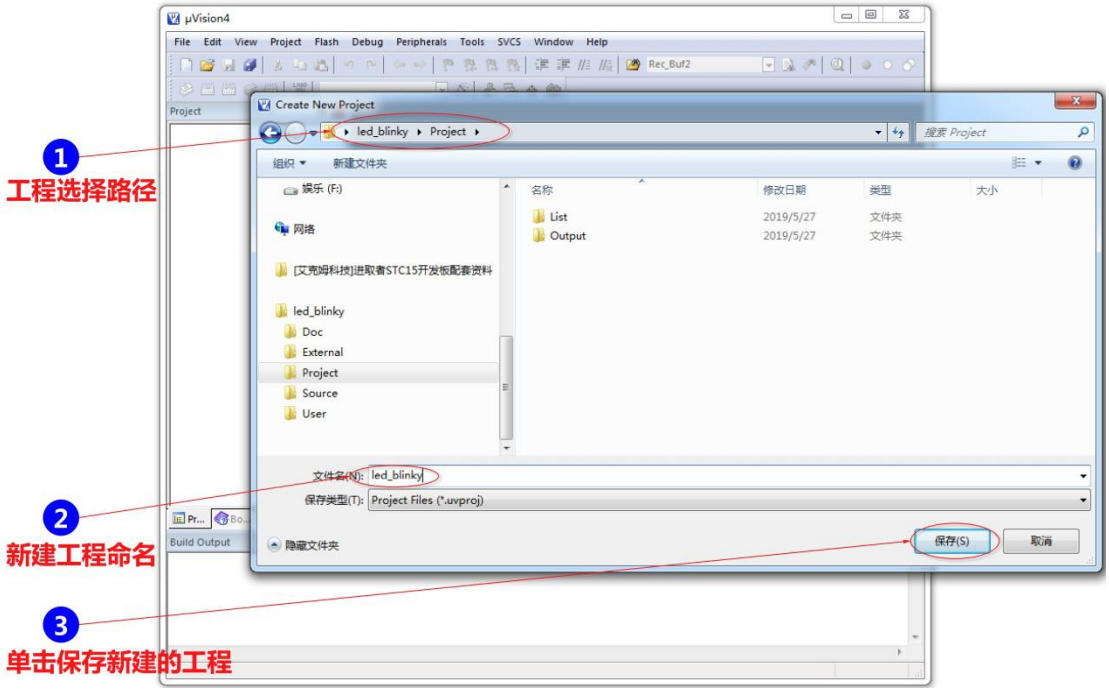


图 8： Keil C51 中新建工程命名

✧ 注：新建工程的保存路径设置为：“...\ led_blinky\Project”。

5. 添加单片机型号，需选择【STC MCU Database】。

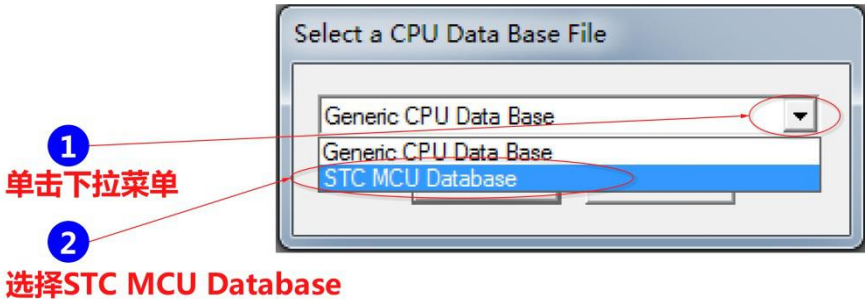


图 9：选择单片机型号

✧ 注：只有在 Keil 仿真设置界面中正确操作“添加型号和头文件到 Keil 中、添加 STC 仿真器驱动到 Keil 中”才会有【STC MCU Database】选项。

6. 选择【STC MCU Database】后，点击【OK】。

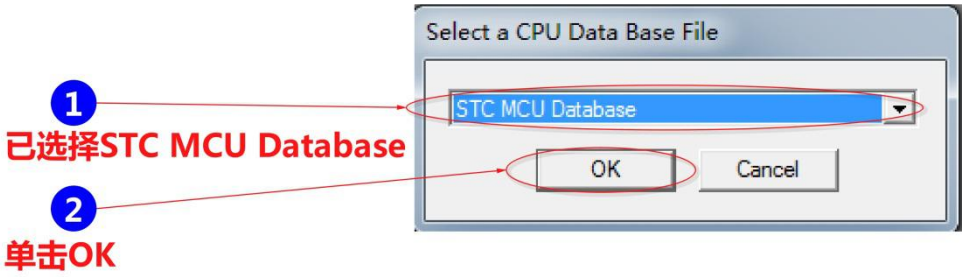


图 10：选中【STC MCU Database】

7. 完成上述操作之后，STC 的单片机型号就可以浏览了。

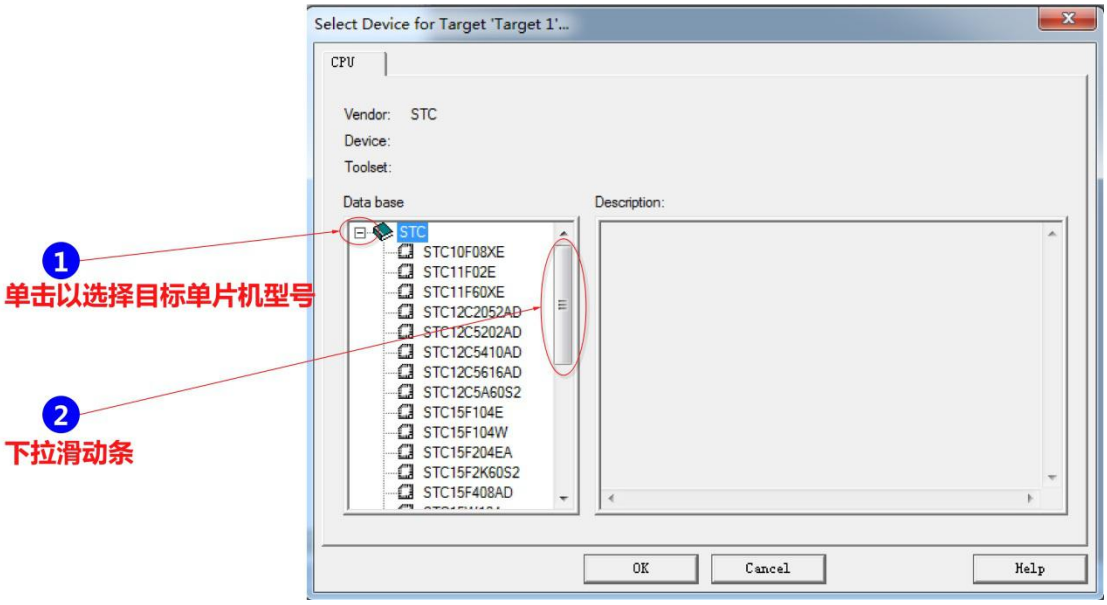


图 11：查找合适的 MCU 型号

8. 选择 MCU 型号为【STC15W4K32S4】，如下图。

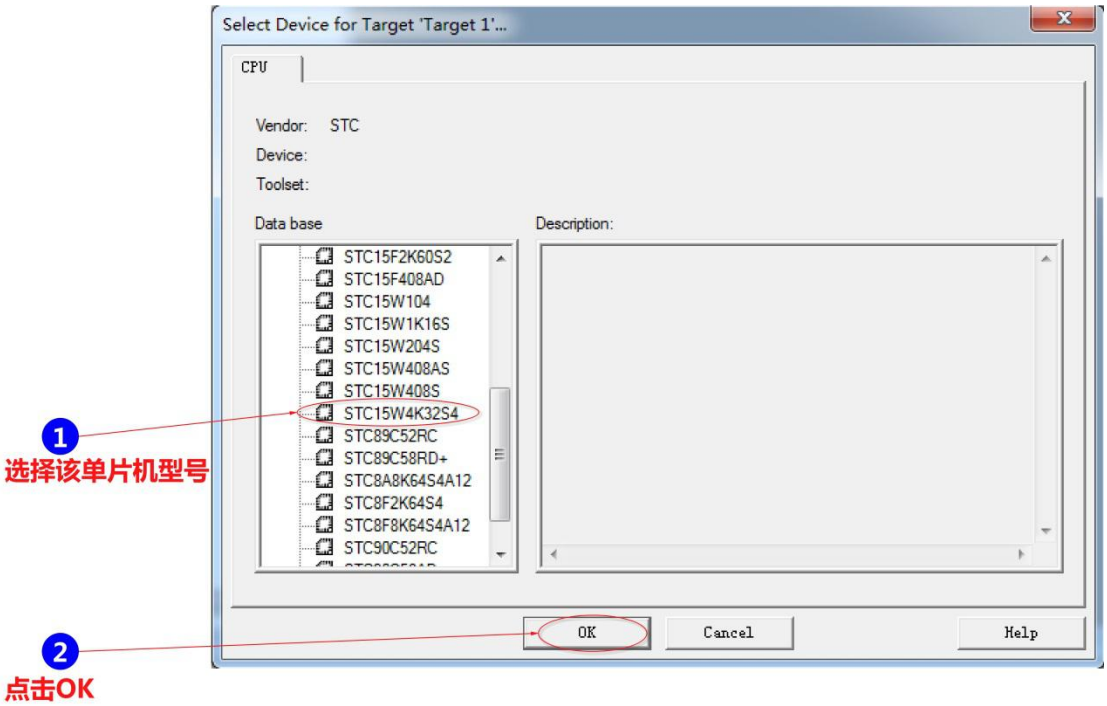


图 12：选择单片机型号

✧ 注：这里的选项 STC15W4K32S4 代表的意思是 STC15W4K32S4 系列，STC15W4K56S4 和 IAP15W4K58S4 均属于 STC15W4K32S4 系列。

9. 当出现的下面提示框时，请选择【否】。

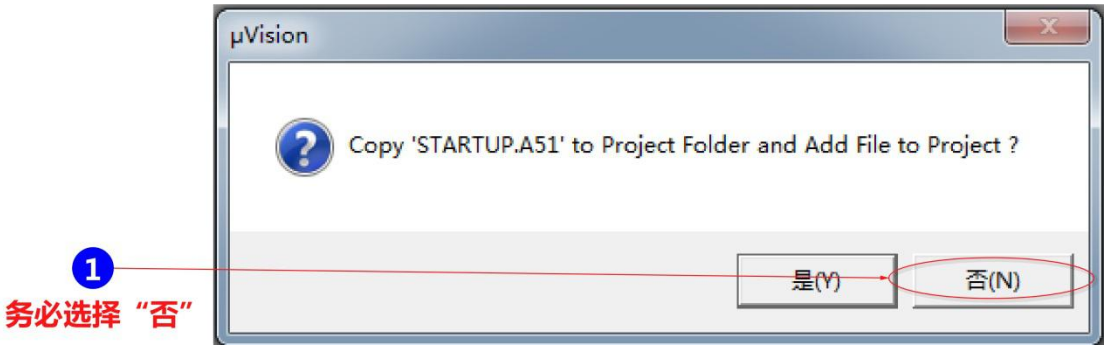


图 13： Keil C51

✧ 注：startup.a51 是 keil C51 的启动代码，这里不需要添加。

10. 目标设备名称命名，并对目标设备下包含组分类并命名。

工程的目标设备是 IK-STC15 开发板，所以示例可以命名目标设备为 IK-STC15。工程目标设备下可包含 3 个组，用来归类加入到工程的文件，如果是 main 文件，可以加入到“USER”组，如果是用户自己编写的文件，可以加入到“SOURCE”组，如果是工程相关的说明文档、发布信息的文档，可以加入到“DOC”组。

✧ **注意事项 1：**如果使用了第三方的库，可以再新建“EXTERNAL”组，并将文件加入到该组。一般用不到，所以示例没有添加“EXTERNAL”组。

✧ **注意事项 2：** MDK 中的工程目录不需要完全和工程文件存放目录一样。

1) 目标设备名称默认是 Target1，可根据需要进行修改。示例修改命名为 IK-STC15。

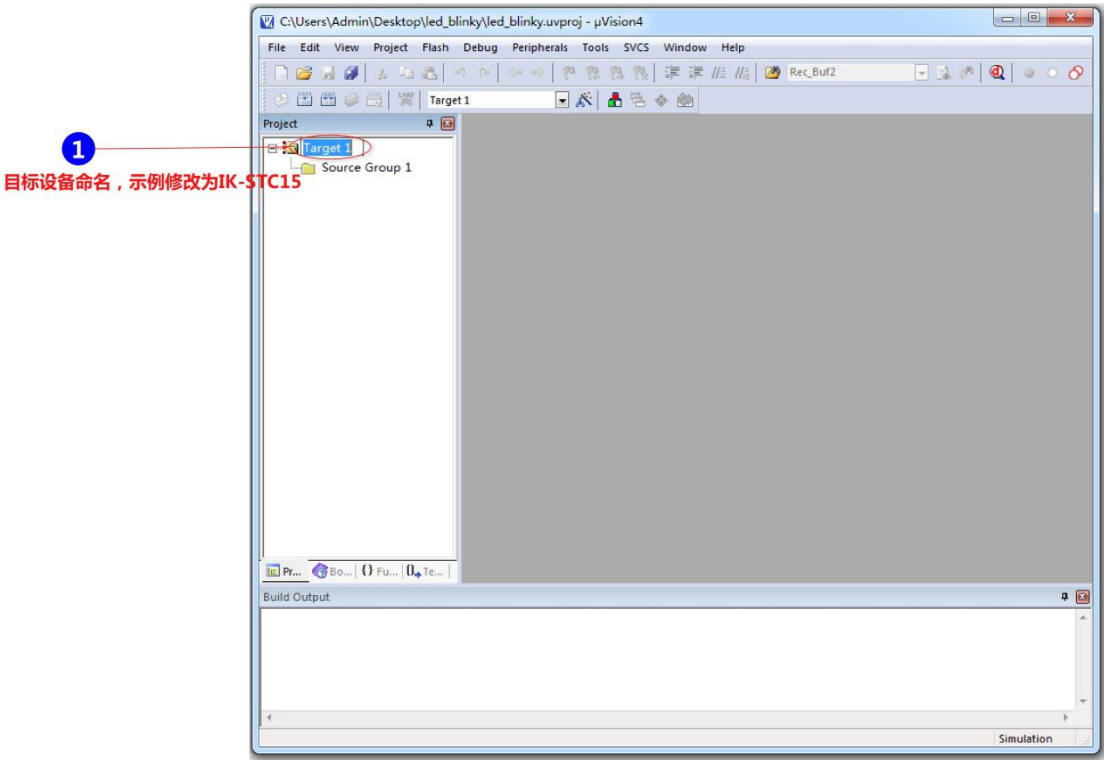


图 14： Keil C51 中命名目标设备名称

2) 修改目标设备下第 1 个组命名为 USER。

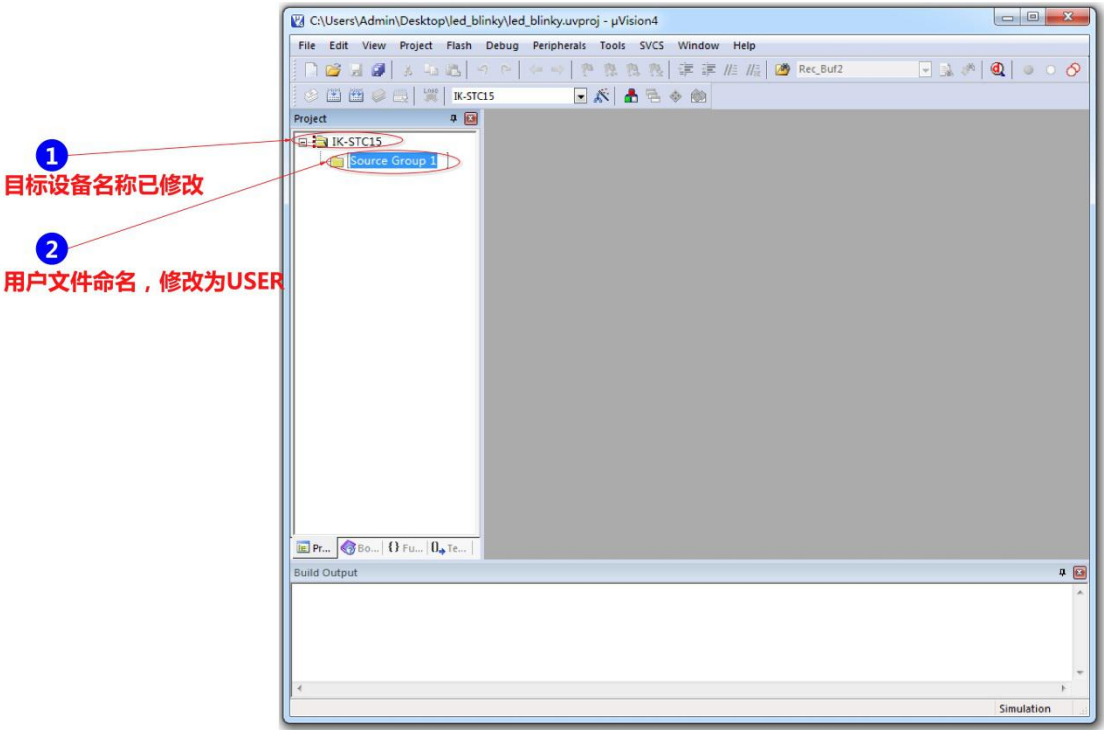


图 15：目标设备新建组重命名

3) 在目标设备下新添加其他组，可重新命名这些组。

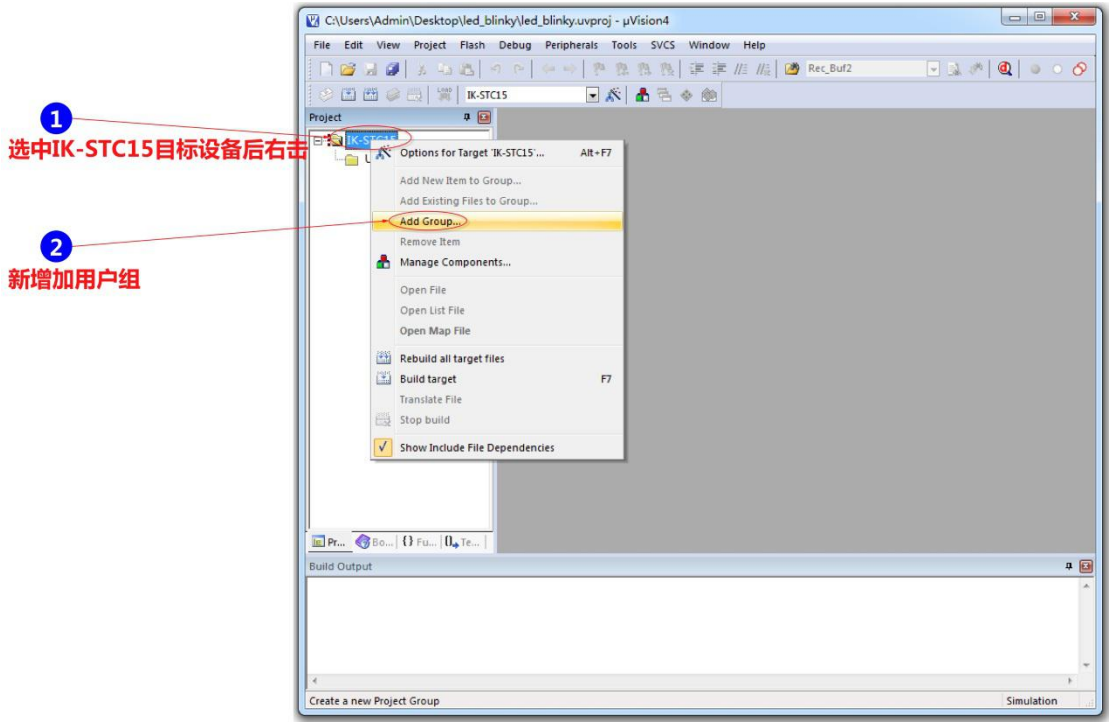


图 16：目标设备添加新建组

4) 完成目标设备新建 3 个组，并分别命名为 USER、SOURCE 和 DOC。

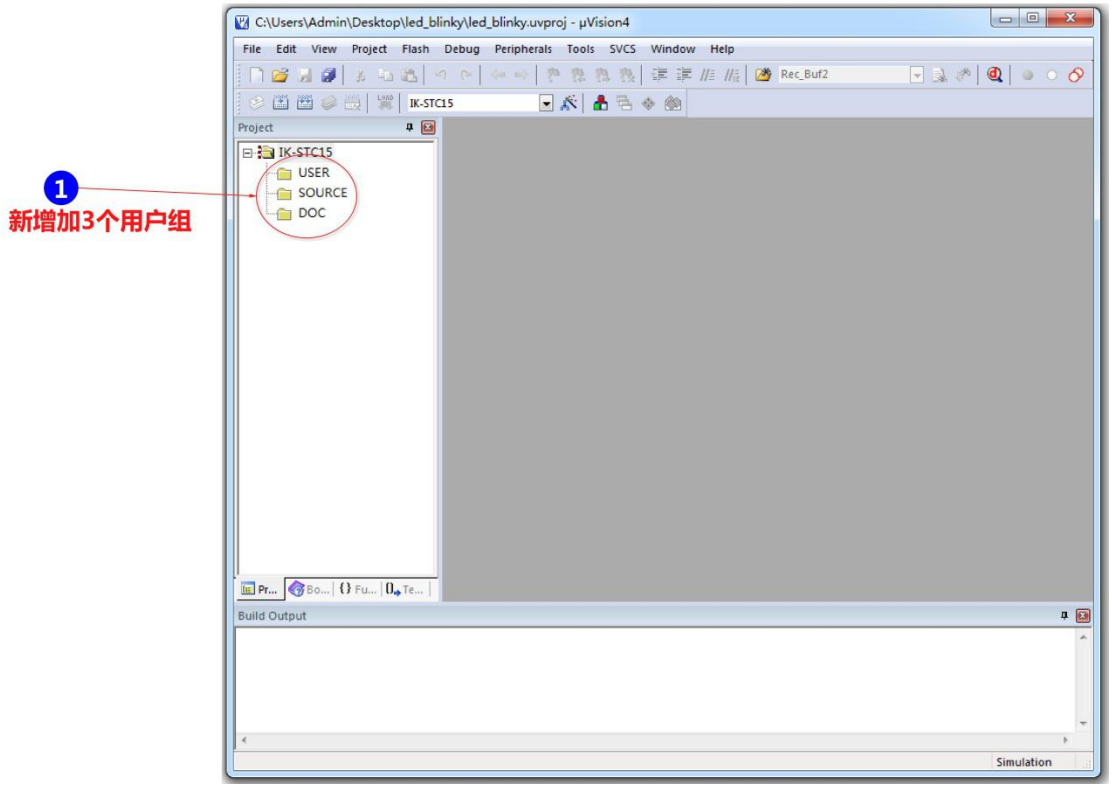


图 17：目标设备新建 3 个组

11. 新建 main.c 文件并添加到工程。

通常，工程都会包含一个“main.c”文件，用于存放 C 程序的入口函数（main()函数），所以，我们需要先新建一个“main.c”文件并添加到工程。

1) 执行“File→New”新建文件，如下图：

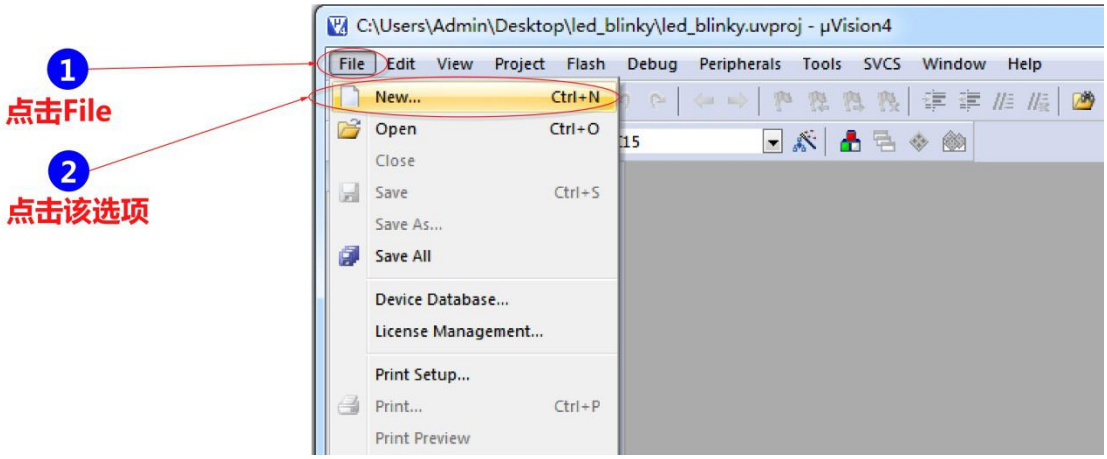


图 18：Keil C51 中新建文件

2) 新建初始文件是默认以 Text1 命名的，如下图：

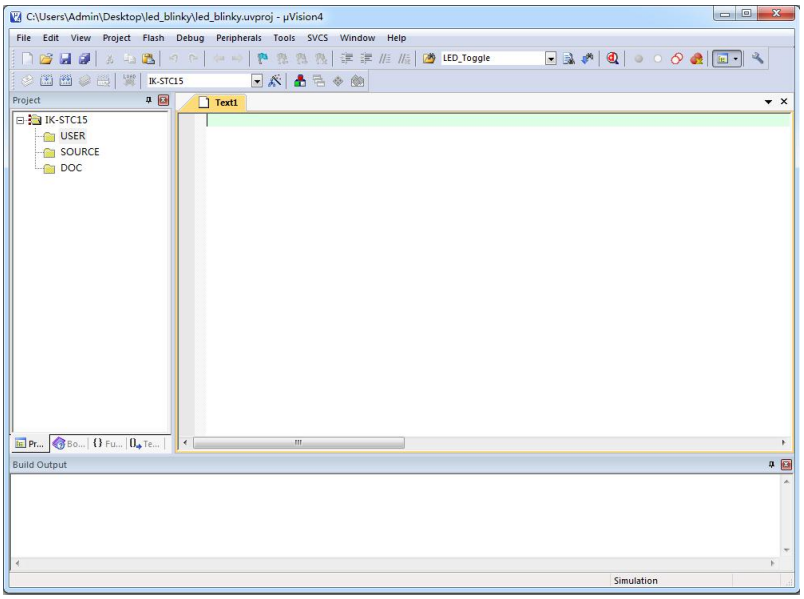


图 19：Keil C51 中新建文件

3) 在 Text1 文件上输入需要写的代码，如下图：

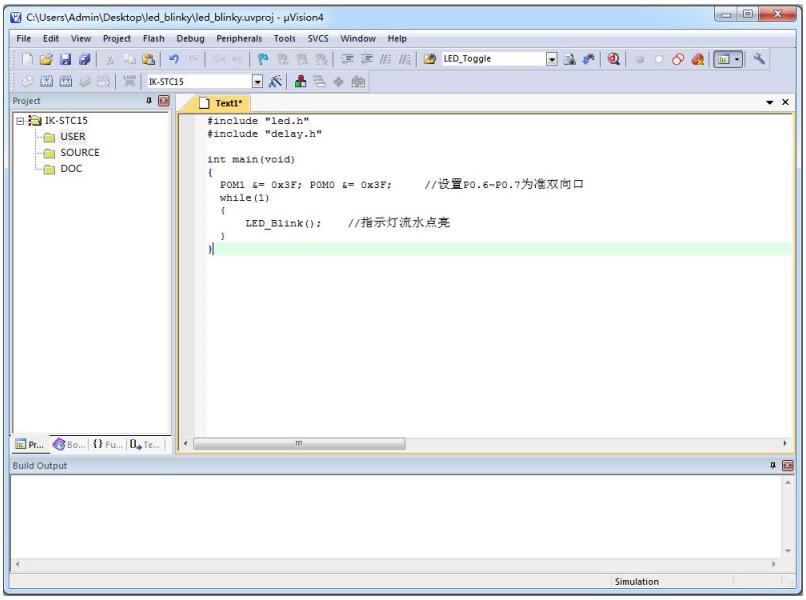


图 20：Keil C51 中新建文件写入代码

4) 点击保存按钮“Save”将 Text1 文件按路径保存到 User 文件夹。

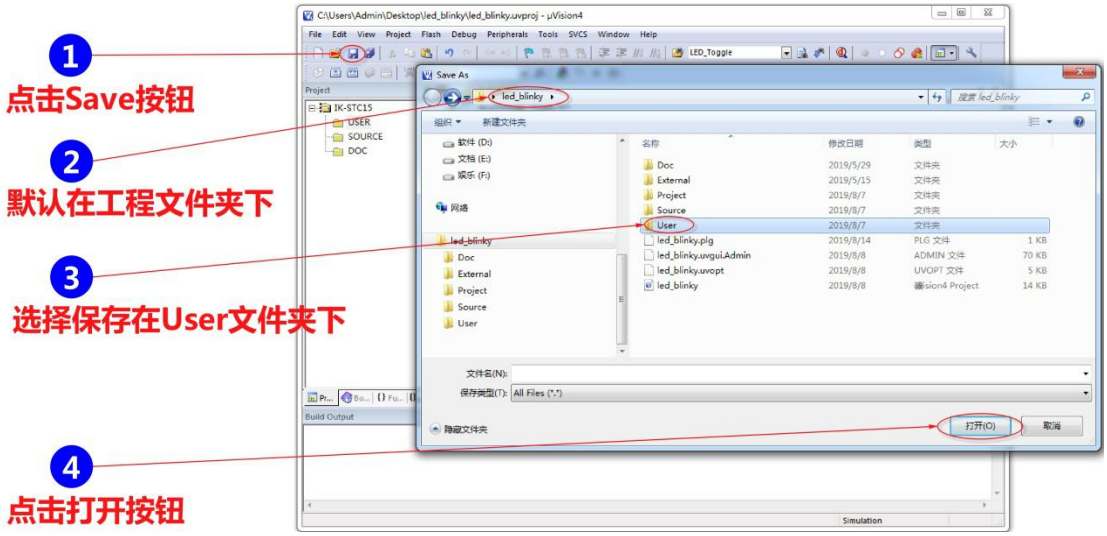


图 21：Keil C51 中保存新建文件

5) 将 Text1 文件命名为“main.c”。

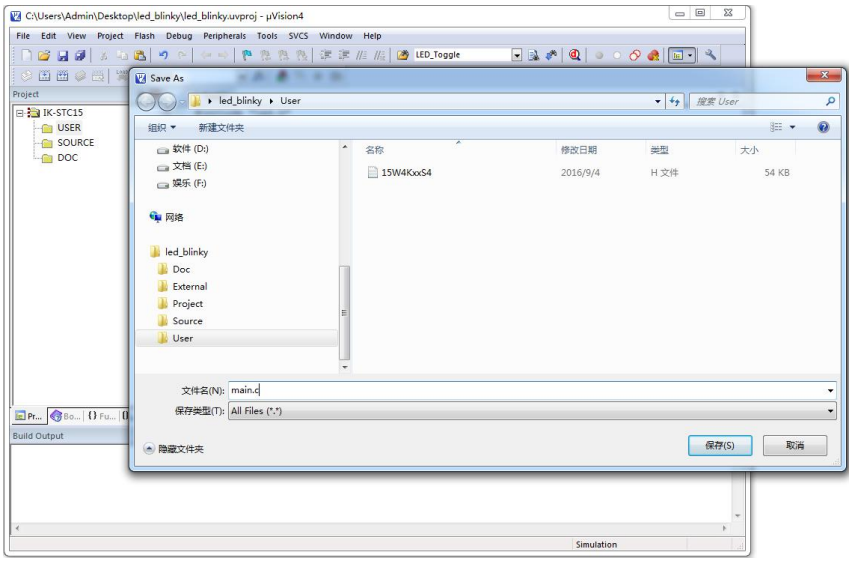


图 22：Keil C51 中重命名新建文件

6) 保存后的“main.c”文件在 Keil 中被打开如下。

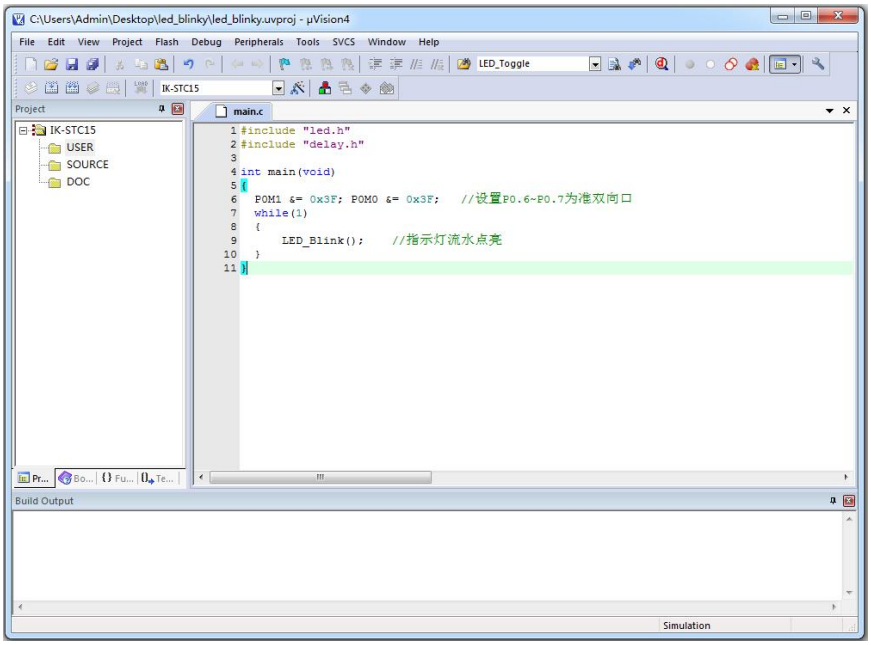


图 23：Keil C51 中新建文件完成

✧ 注：此时在工程文件存放目录 User 中可以看到刚保存的 main.c 文件。

7) 在工程 USER 组中添加已存在文件。

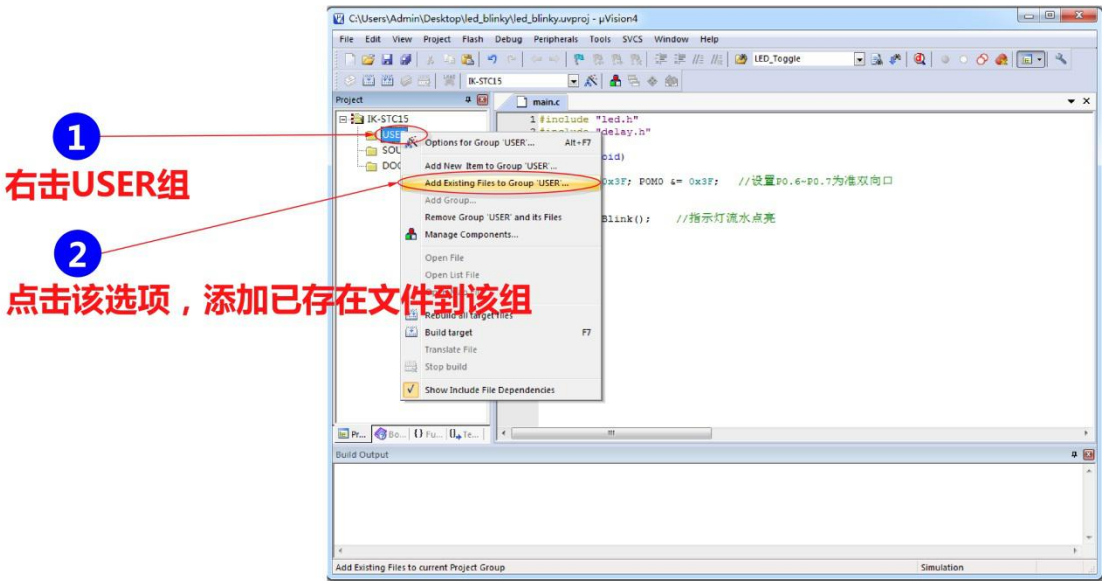


图 24：Keil C51 中添加新建文件到对应组

8) 按路径找到工程文件存放目录 User 文件夹。

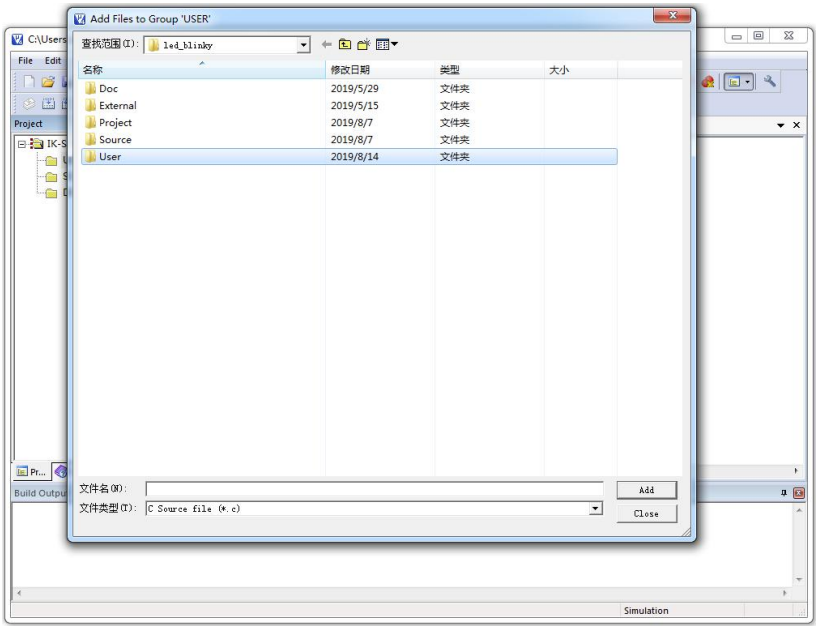


图 25：Keil C51 中添加新建文件到对应组

9) 将 User 文件夹中的“main.c”文件选中添加到工程。

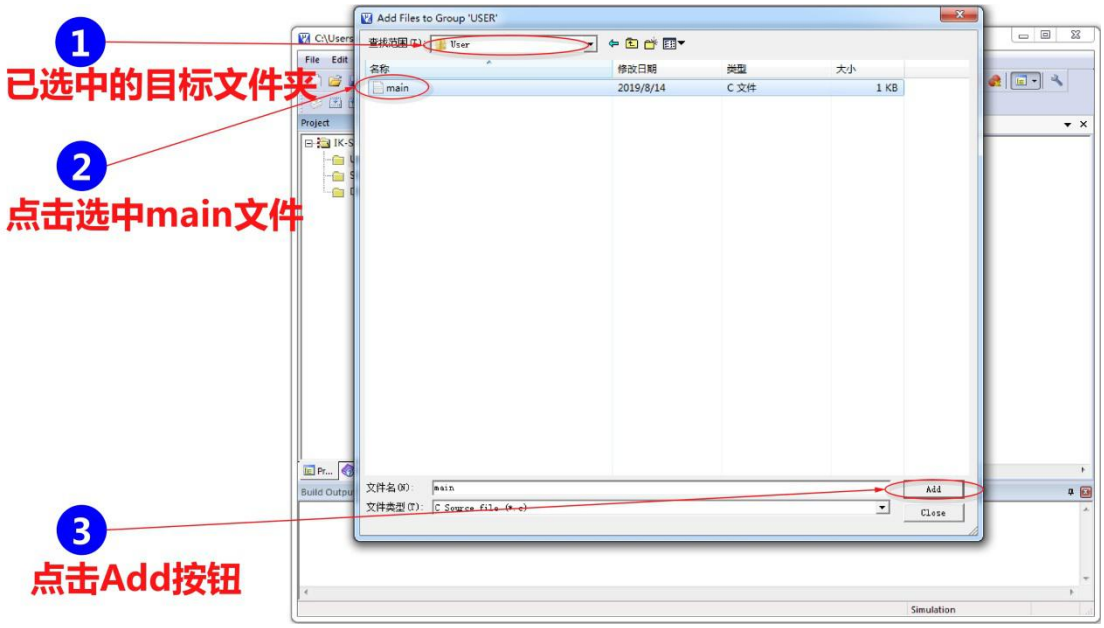


图 26：Keil C51 中添加新建文件到对应组

10) “main.c”文件已被成功添加到工程 USER 组。

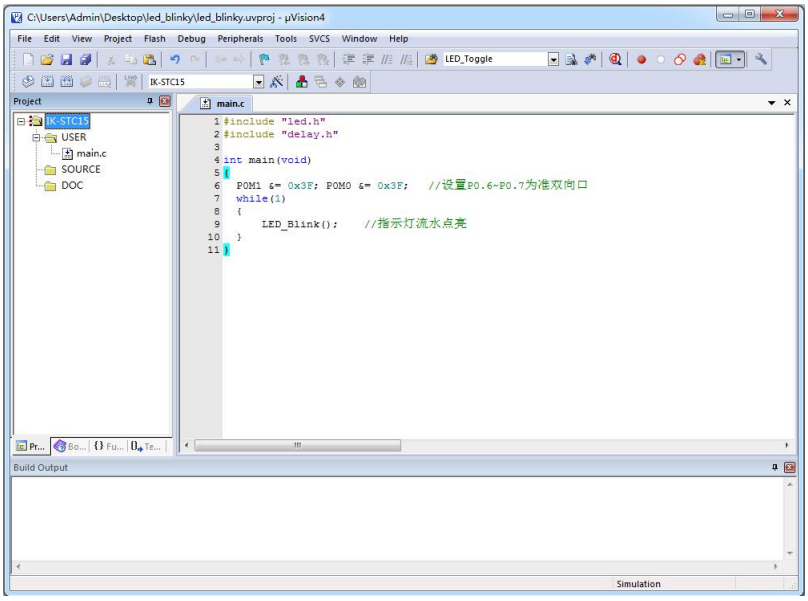


图 27：Keil C51 中成功添加新建文件到对应组

✧ 注：USER 组添加进文件后，其前面便有了“+”的图标，点击“+”展开可以看到 USER 组下的成员。

12. 新建 delay.h、delay.c、led.h、led.c 文件，并将 delay.c 和 led.c 文件添加到工程。

新建 delay.h、delay.c、led.h、led.c 文件的方法和新建 main.c 文件基本一致，都是执行“File →New”新建文件，输入代码，保存文件到工程文件存放目录 Source 中，最后再添加到工程 SOURCE 组中。

1) 保存 delay.h 文件时注意此时的文件类型，如下图：

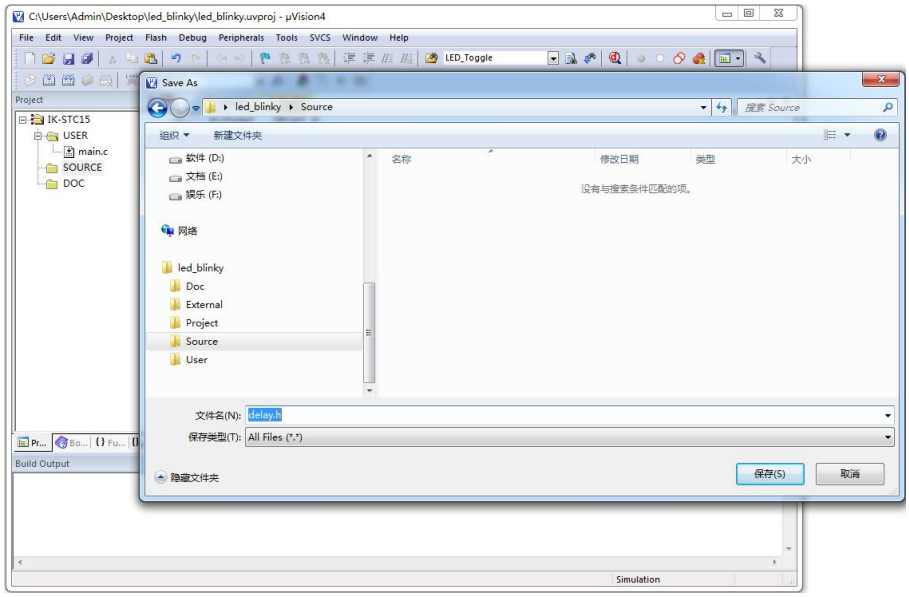


图 28：Keil C51 中新建 delay.h 文件

2) 保存后的“delay.h”文件在 Keil 中被打开如下。

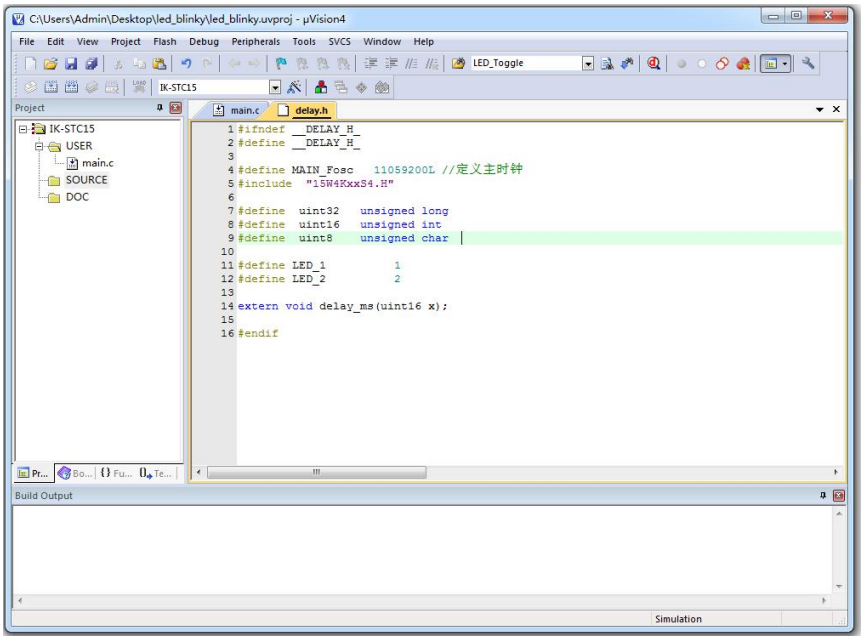


图 29：Keil C51 中完成新建 delay.h 文件

✧ 注：此时在工程文件存放目录 Source 中可以看到刚保存的 delay.h 文件。

3) 保存后的“delay.c”文件在 Keil 中被打开如下。

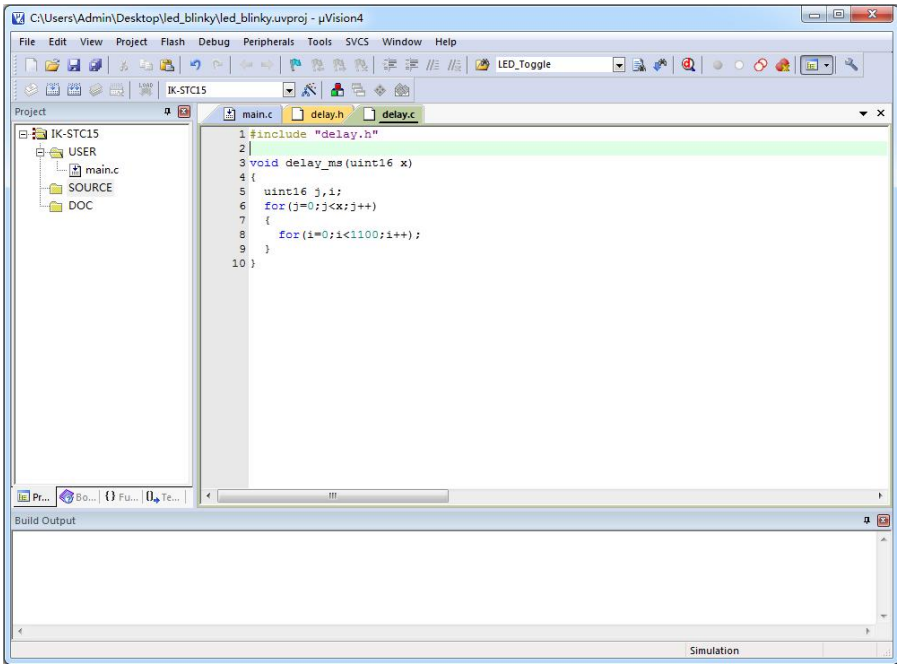


图 30：Keil C51 中完成新建 delay.c 文件

✧ 注：此时在工程文件存放目录 Source 中可以看到刚保存的 delay.c 文件。

4) 保存后的“led.c”文件在 Keil 中被打开如下。

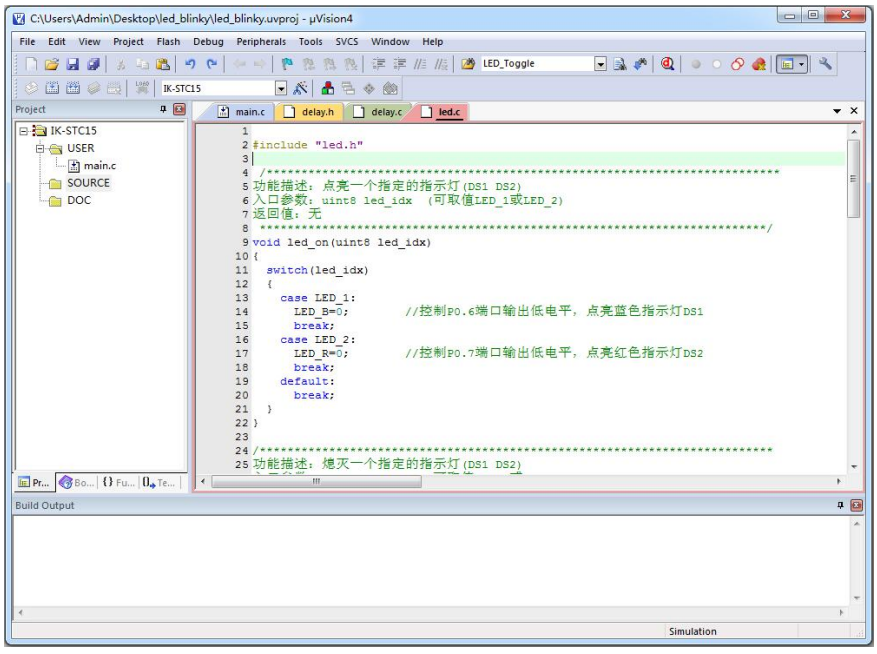


图 31：Keil C51 中完成新建 led.c 文件

✧ 注：此时在工程文件存放目录 Source 中可以看到刚保存的 led.c 文件。

5) 保存后的“led.h”文件在 Keil 中被打开如下。

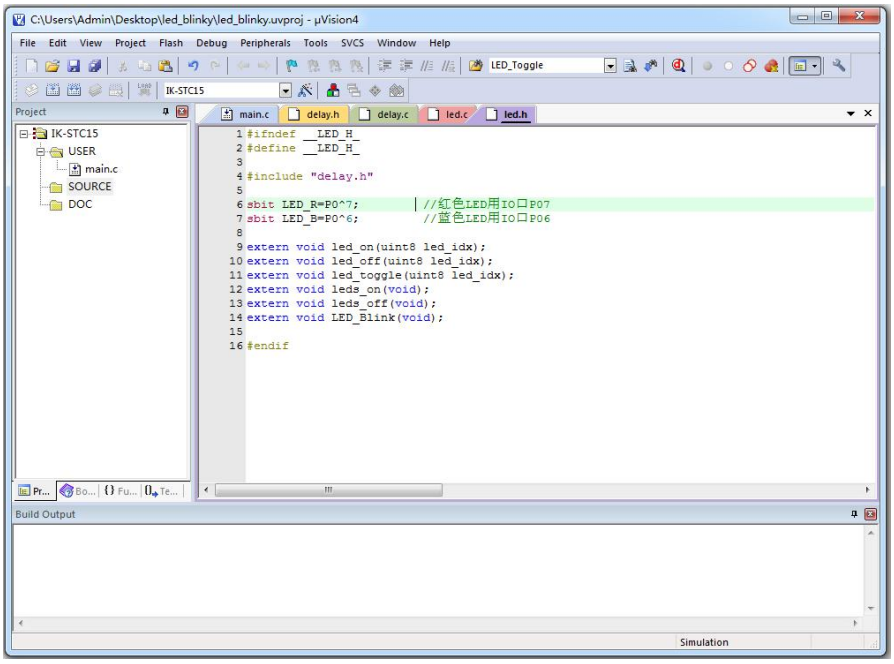
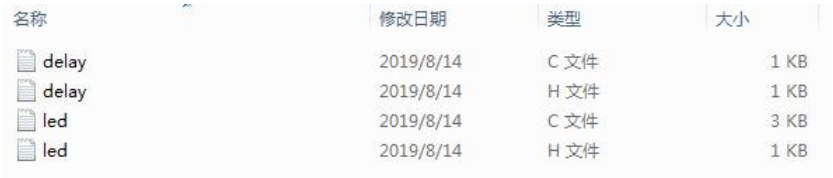


图 32：Keil C51 中完成新建 led.h 文件

✧ 注：此时在工程文件存放目录 Source 中可以看到刚保存的 led.h 文件。

6) 打开工程文件存放目录 Source 文件夹，可以看到 delay.h、delay.c、led.h、led.c 文件。



名称	修改日期	类型	大小
delay	2019/8/14	C 文件	1 KB
delay	2019/8/14	H 文件	1 KB
led	2019/8/14	C 文件	3 KB
led	2019/8/14	H 文件	1 KB

图 33：打开工程文件存放目录 Source 文件夹

7) 在工程 SOURCE 组中添加已存在文件。

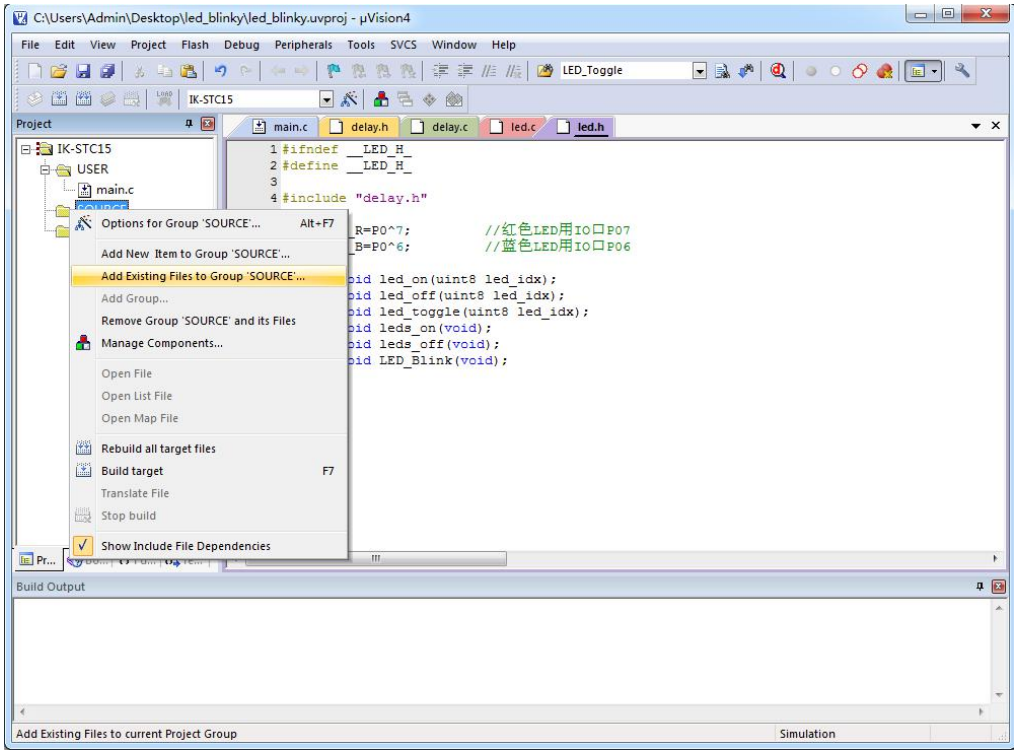


图 34：Keil C51 中添加已存在文件

8) 按路径找到工程文件存放目录 Source 文件夹，并将该文件夹中的 delay.c 和 led.c 文件选中添加到工程。

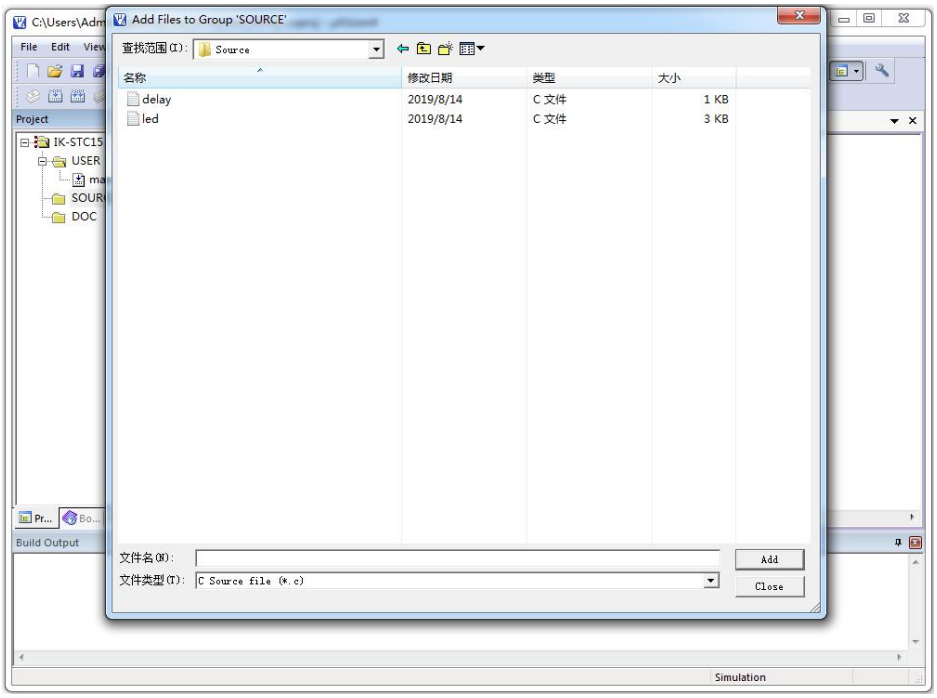


图 35：Keil C51 中添加已存在文件

9) delay.c 和 led.c 文件已被成功添加到工程 SOURCE 组。

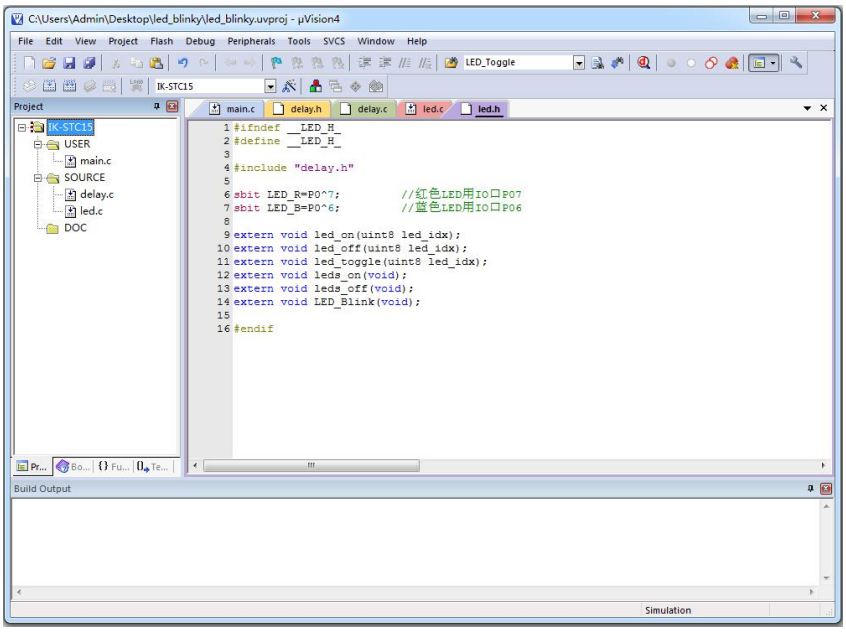


图 36：Keil C51 中添加已存在文件

✧ 注：SOURCE 组添加进文件后，其前面便有了“+”的图标，点击“+”展开可以看到 SOURCE 组下的成员。

13. 新建 ReadMe.txt 文件，并将 ReadMe.txt 文件添加到工程。

新建 ReadMe.txt 文件的方法和新建 main.c 文件基本一致，都是执行“File→New”新建文件，输入内容，保存文件到工程文件存放目录 Doc 中，最后再添加到工程 DOC 组中。

✧ **注意事项：**ReadMe.txt 文件只是用于工程相关的说明或发布信息等，没有实际执行的代码，所以一个工程中是可以没有该文件的，不会影响 HEX 代码的生成与运行。

1) 保存 ReadMe.txt 文件时注意此时的文件类型，如下图：

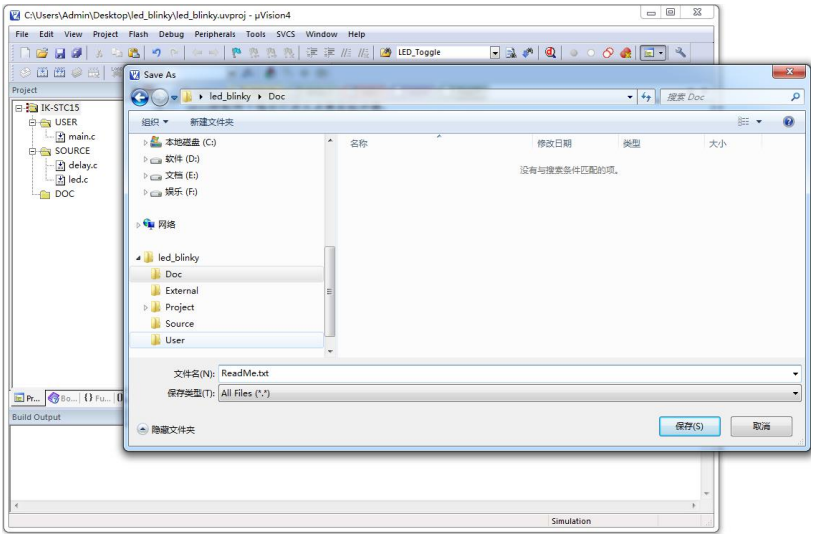


图 37：Keil C51 中新建文件

✧ 注：此时在工程文件存放目录 Doc 中可以看到刚保存的 ReadMe.txt 文件。

2) 保存后的“ReadMe.txt”文件在 Keil 中被打开如下。

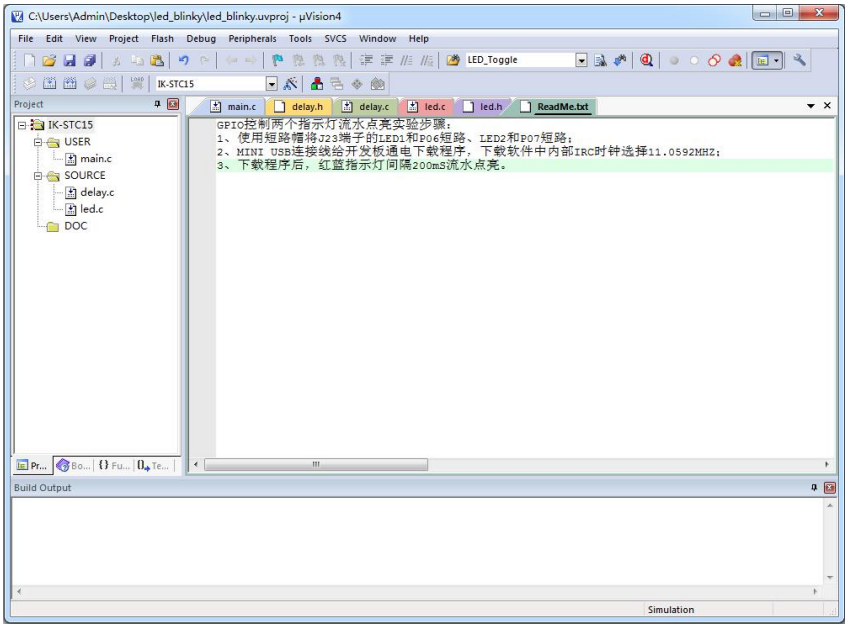


图 38：Keil C51 中新建文件

3) 在工程 DOC 组中添加已存在文件。

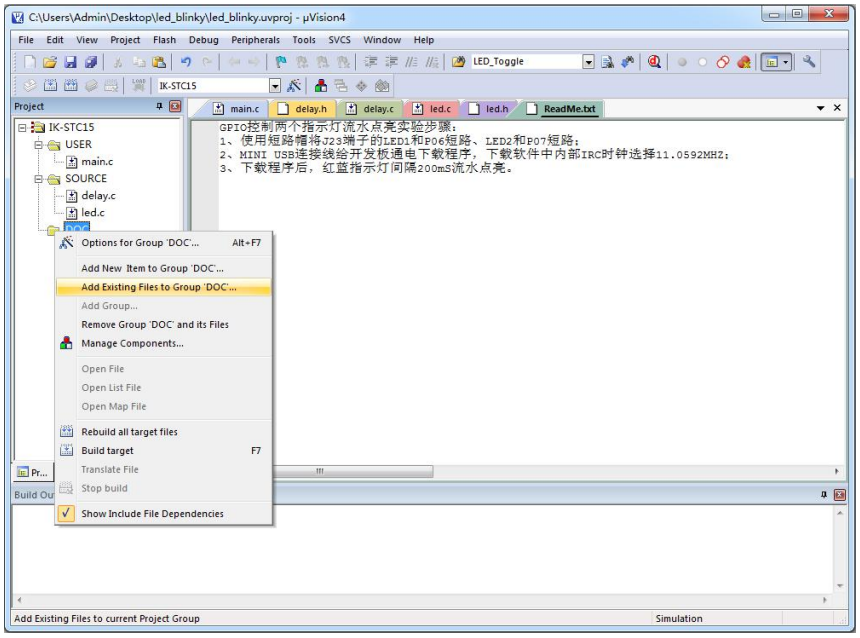


图 39：Keil C51 中添加已存在文件

4) 按路径找到工程文件存放目录 Doc 文件夹，并将该文件夹中的 ReadMe.txt 文件选中添加到工程。

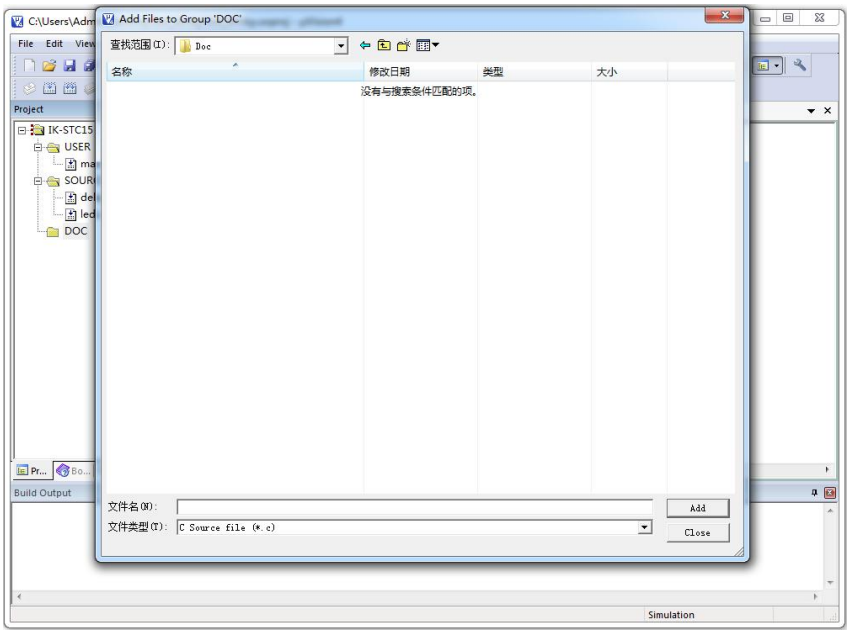


图 40：Keil C51 中添加已存在文件

✧ 注：此时在工程文件存放目录 Doc 中并没有看到 ReadMe.txt 文件。

5) 点击文件类型下拉菜单，可以看到很多种文件类型。

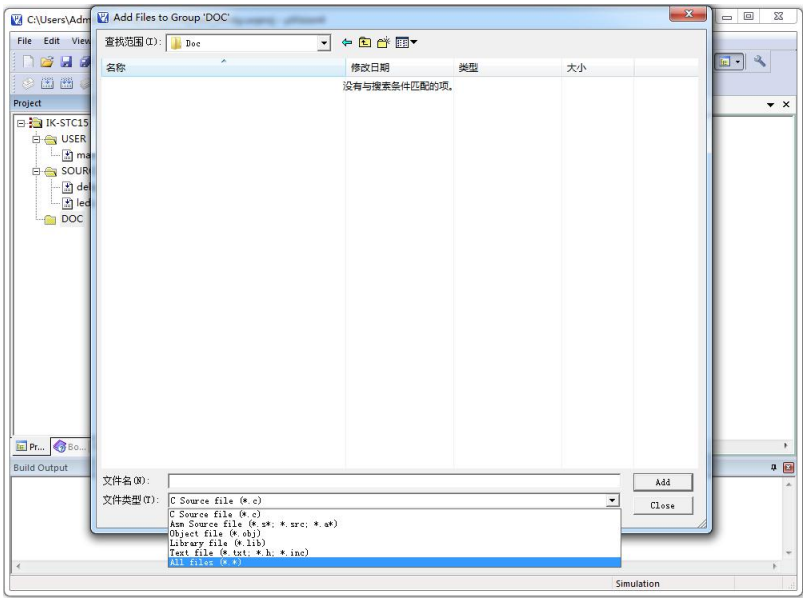


图 41：Keil C51 中添加已存在文件

6) 选择【All files】文件类型，这样可以看到“ReadMe.txt”文件了。

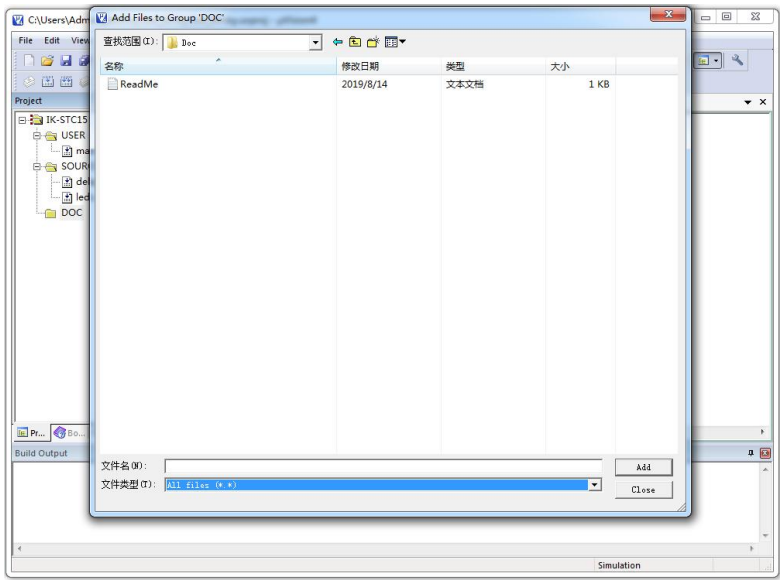


图 42：Keil C51 中添加已存在文件

✧ 注：此时可以在工程文件存放目录 Doc 中看到 ReadMe.txt 文件。

7) ReadMe.txt 文件已被成功添加到工程 DOC 组。

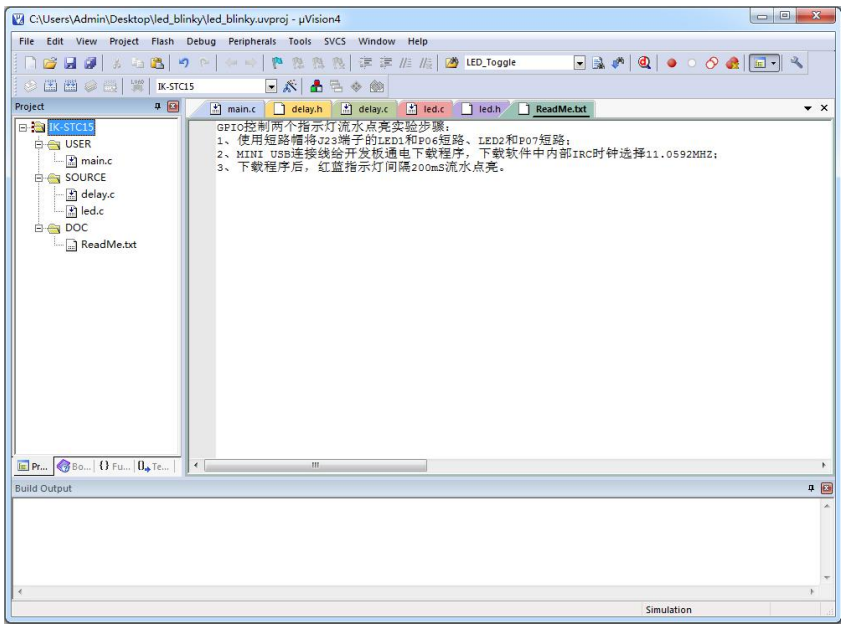


图 43：Keil C51 中添加已存在文件

✧ 注：DOC 组添加进文件后，其前面便有了“+”的图标，点击“+”展开可以看到 DOC 组下的成员。

14. 选择工程配置窗口的【Target】界面进行配置。

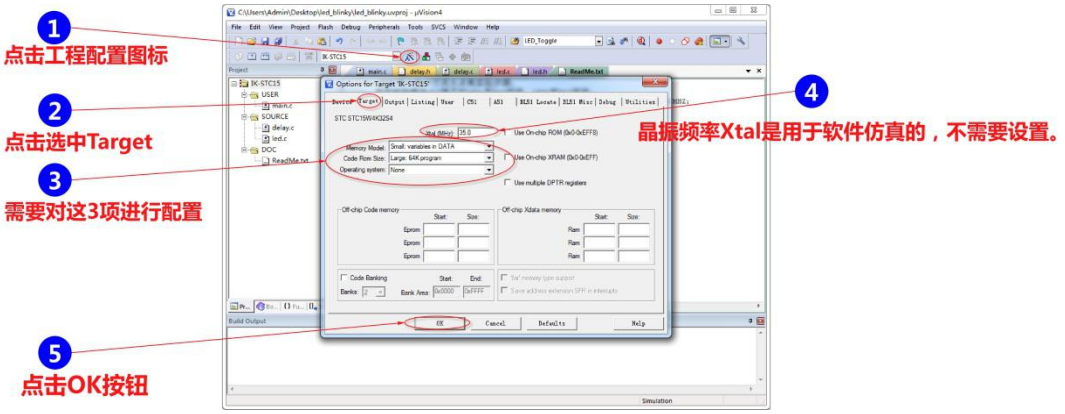


图 44：Keil C51 工程配置 Target 界面

✧ **注意事项 1：**关于 Memory Model 选项。

Small:variables in DATA：所有变量都在的内部 RAM。

Compact:variables in PDATA：使用部分扩展 RAM。

Larget:variables in XDATA：使用全部扩展 RAM。

✧ **注意事项 2：**关于 Code Rom Size 选项。

Small:program 2k or less：只用低于 2K 的程序空间。

Compact:2k functions,64k program：单个函数的代码量不能超过 2K，整个程序可以使用 64K 程序空间。

Larget:64k program：可用全部 64K 空间。

✧ **注意事项 3：**关于 Operating system 选项。

None：不使用实时操作系统。

RTX-51 Tiny：使用 RTX-51 实时操作系统 TINY 版本。

RTX-51 Full：使用 RTX-51 实时操作系统 FULL 版本。

15. 完成工程配置窗口的【Target】配置。

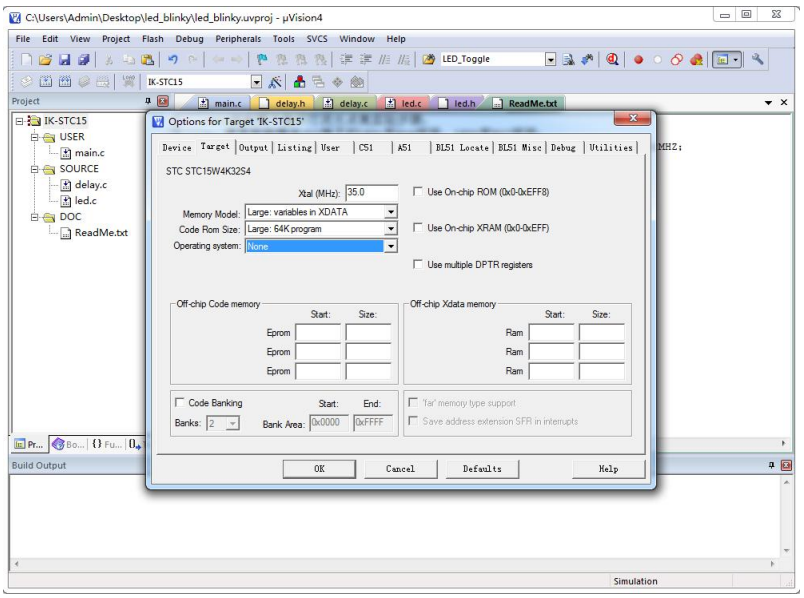


图 45：Keil C51 工程配置 Target 界面完成

16. 选择工程配置窗口的【Output】界面进行配置。

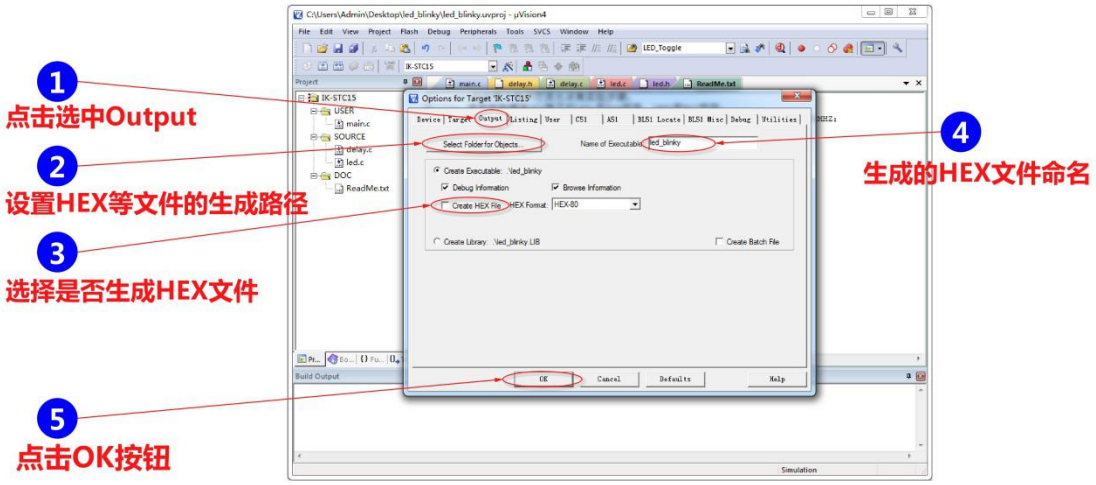


图 46：Keil C51 工程配置 Output 界面

✧ **注意事项 1：**设置 HEX 文件的生成路径为：“...\ led_blinky\Project\Output”。

✧ **注意事项 2：**务必勾选 Create HEX File 前面的选项，即选择生成 HEX 文件。

17. 完成工程配置窗口的【Output】配置。

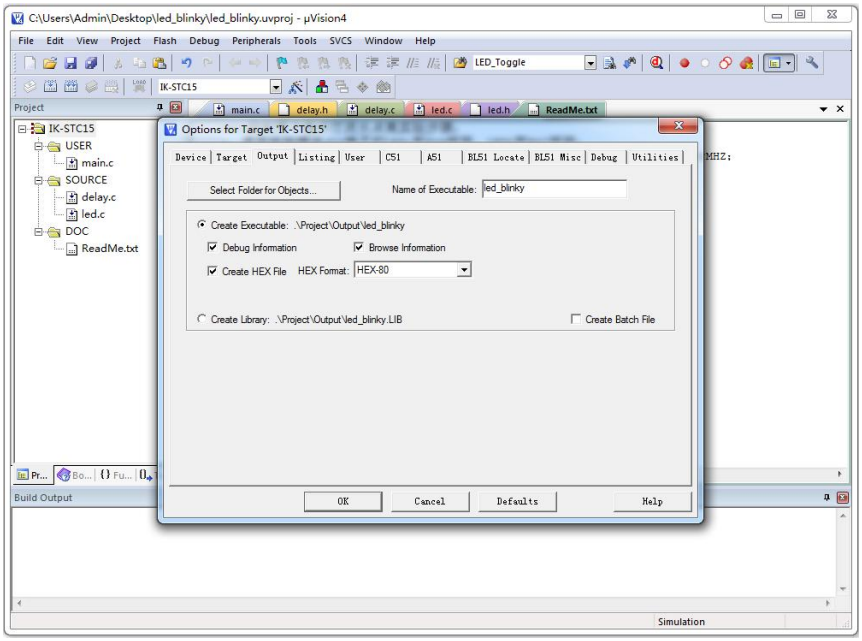


图 47：Keil C51 工程配置 Output 界面完成

18. 选择工程配置窗口的【Listing】界面进行配置。

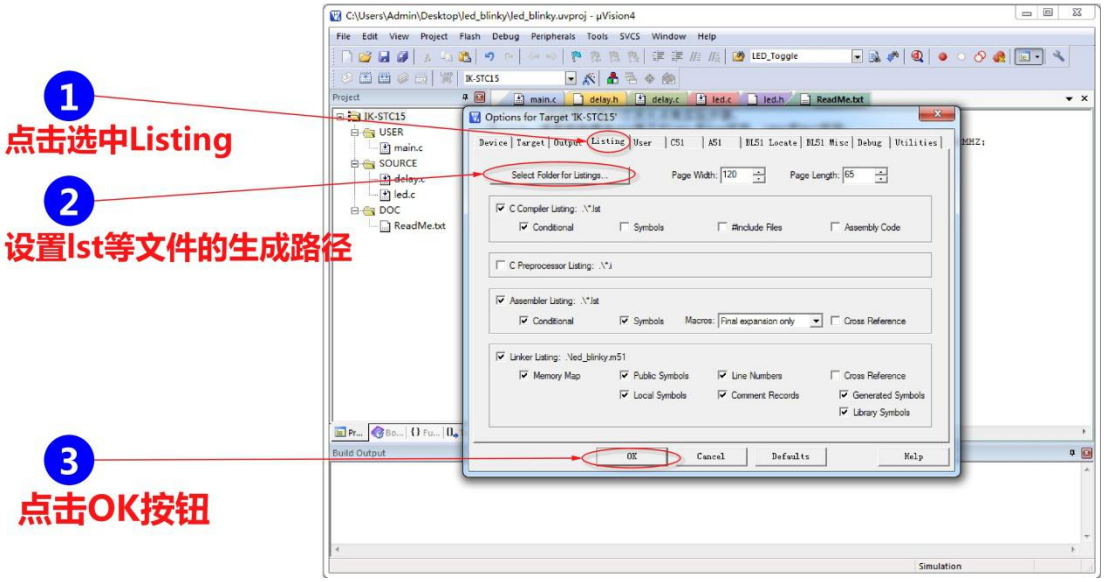


图 48：Keil C51 工程配置 Listing 界面

✧ **注意事项 1：**设置 lst 文件的生成路径为：“...\ led_blinky\Project\List”。

✧ **注意事项 2：**Assamble Code 选项的功能是生成 C 语言源程序所对应的汇编代码。

19. 完成工程配置窗口的【Listing】配置。

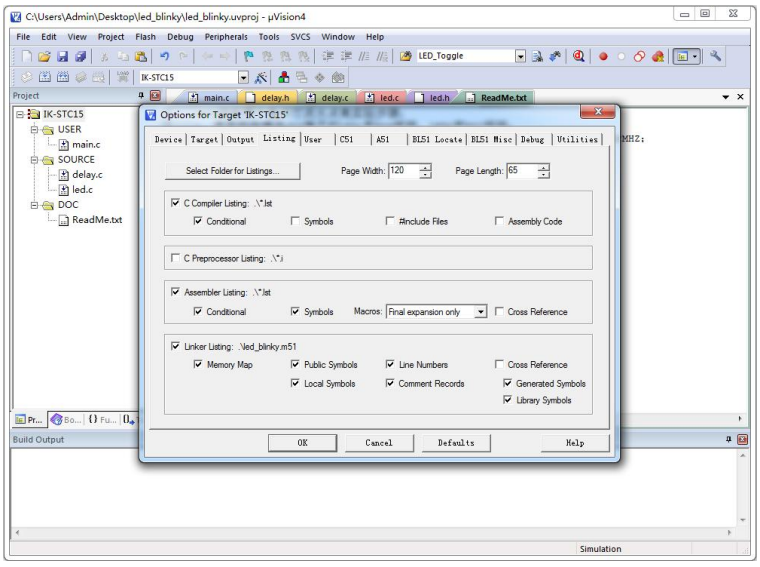


图 49：Keil C51 工程配置 Listing 界面完成

20. 选择工程配置窗口的【C51】界面进行配置。

工程配置窗口的【C51】界面，可供配置项比较多，包括全局宏定义、优化等级、编译优先级和头文件包含路径选择等。优化等级和编译优先级可使用默认即可，全局宏定义需要时可添加，比较关键的是头文件包含路径选择。

✧ **注意事项 1：**关于 Emphasis 选项。

Favor speed：生成的代码速度最快。

Favor size：生成的代码量最少。

<default>：默认，速度优先。

✧ **注意事项 2：**所有工程中用到的头文件必须都将其工程目录路径添加进来。

1) 工程配置窗口的【C51】界面配置项介绍。

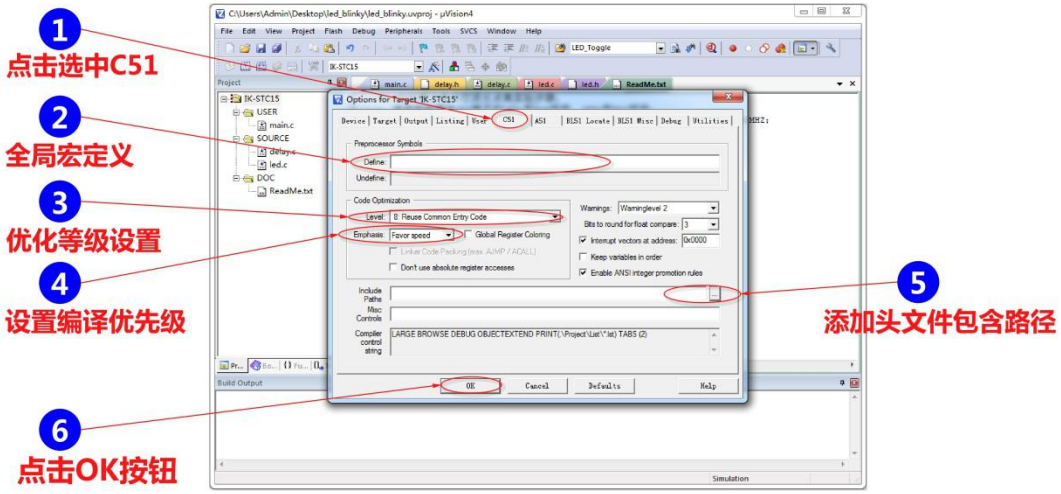


图 50：Keil C51 工程配置 C51 界面

2) 该工程头文件包含路径列表如下。

表 2：头文件包含路径

序号	路径	描述
1	...\ led_blinky\Source	led.h 和 delay.h 头文件在该路径，所以要包含。
2	...\ led_blinky\User	15W4KxxS4.h 头文件在该路径，所以要包含。

3) 添加 led.h 和 delay.h 头文件路径。

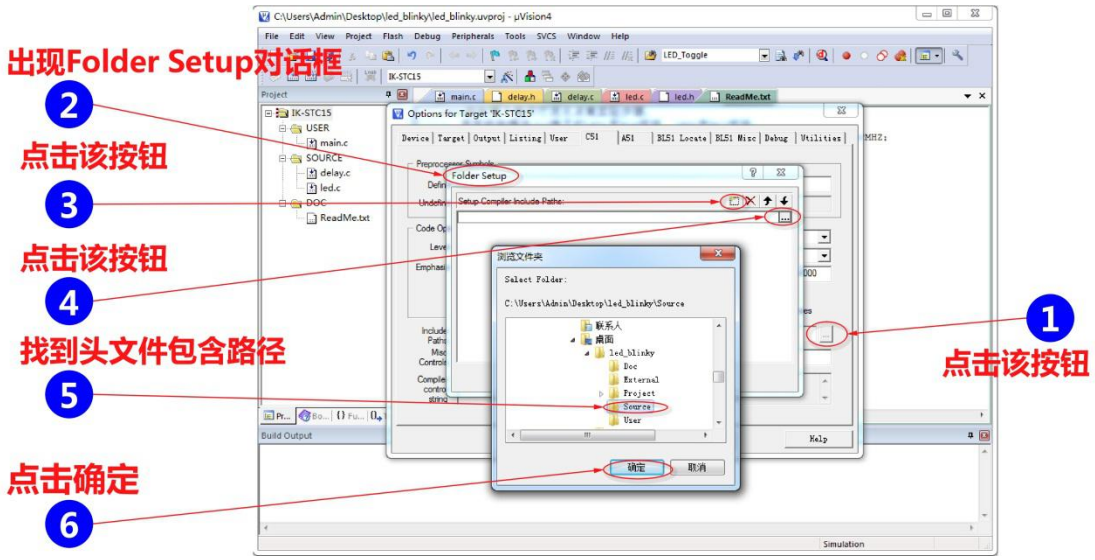


图 51：Keil C51 中添加 led.h 和 delay.h 头文件路径

✧ 注：led.h 和 delay.h 头文件所在路径为：“...\ led_blinky\ Source”。

4) 添加 15W4KxxS4.h 头文件路径。

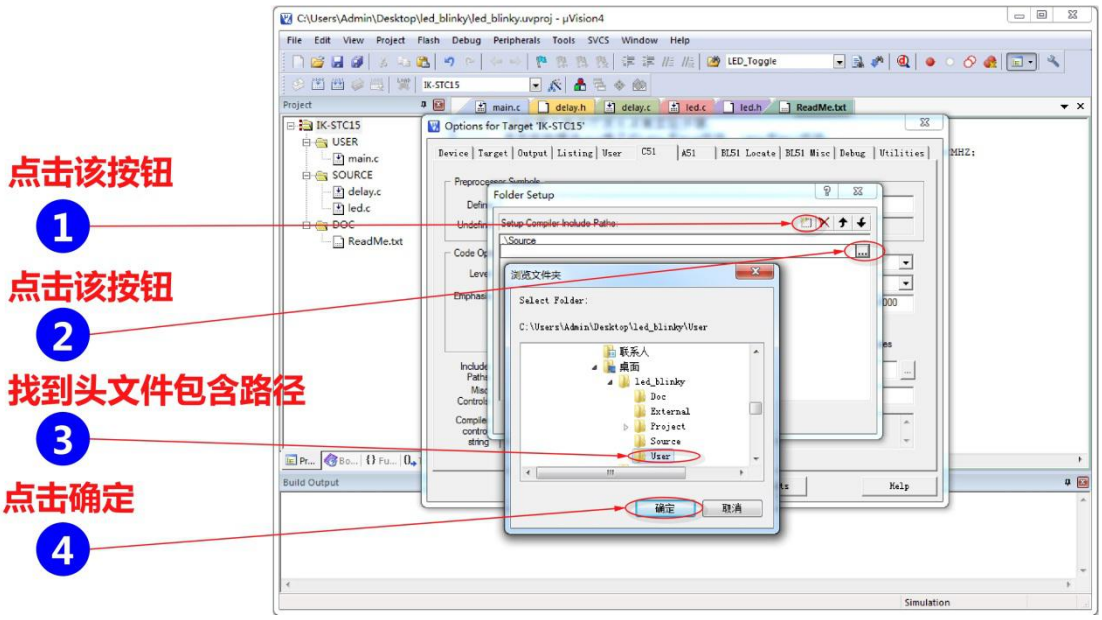


图 52：Keil C51 中添加 15W4KxxS4.h 头文件路径

✧ 注：15W4KxxS4.h 头文件所在路径为：“...\ led_blinky\ User”。

5) 点击下图【OK】按钮，完成工程配置窗口的【C51】配置。

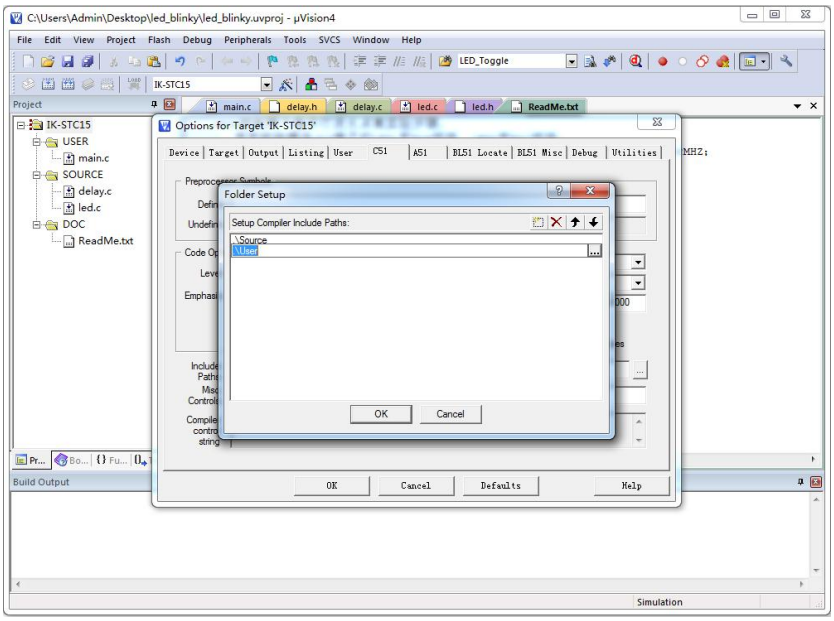


图 53：Keil C51 中完成工程配置窗口的【C51】配置

21. 选择工程配置窗口的【BL51 Misc】界面进行配置。

Keil C51 打开程序源文件编译时，往往会有一些函数在本项目应用工程中没有被调用，而出现报警。但其实这些功能函数非常有用，可能在扩展的应用中就会用到。所以，若简单的删除或屏蔽掉这些功能函数，是不明智的，那怎么办呢？可按照下图对【BL51 Misc】界面进行配置。

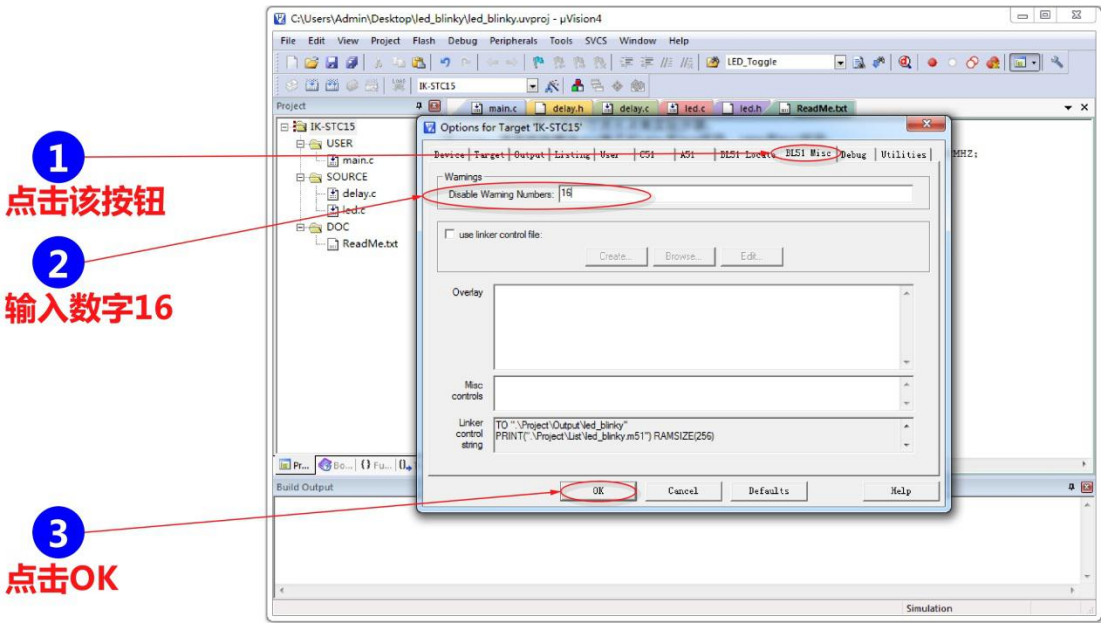


图 54：Keil C51 中工程配置窗口的【BL51 Misc】配置

✧ 注：Disable Warning Numbers 输入数字 16，代表使用 Keil 的消除有未调用的函数而出现的警告的功能，输入其他数字及代表功能不做介绍。

2.3. 编译新建的工程

新建一个流水点灯的工程 led_blinky 已经完成，下面我们可通过编译工程，来验证下新建的工程是不是没有问题。

下面介绍下 Keil 软件的 Build 和 Rebuild 按钮实现编译的区别。

✧ **Build：** 设置是只编译工程中上次修改的文件及其他依赖于这些修改过的文件的模块，同时重新链接生成可执行文件。如果工程之前没编译链接过，他会直接调用 Rebuild All。另外在技术文档中，Build 实际上是指 increase build，即增量编译。

✧ **Rebuild：** 不管工程的文件有没有编译过，都会对工程中所有文件重新进行编译生成可执行文件，因此时间较长。

1. 任选下图界面的编译按钮对工程进行编译。

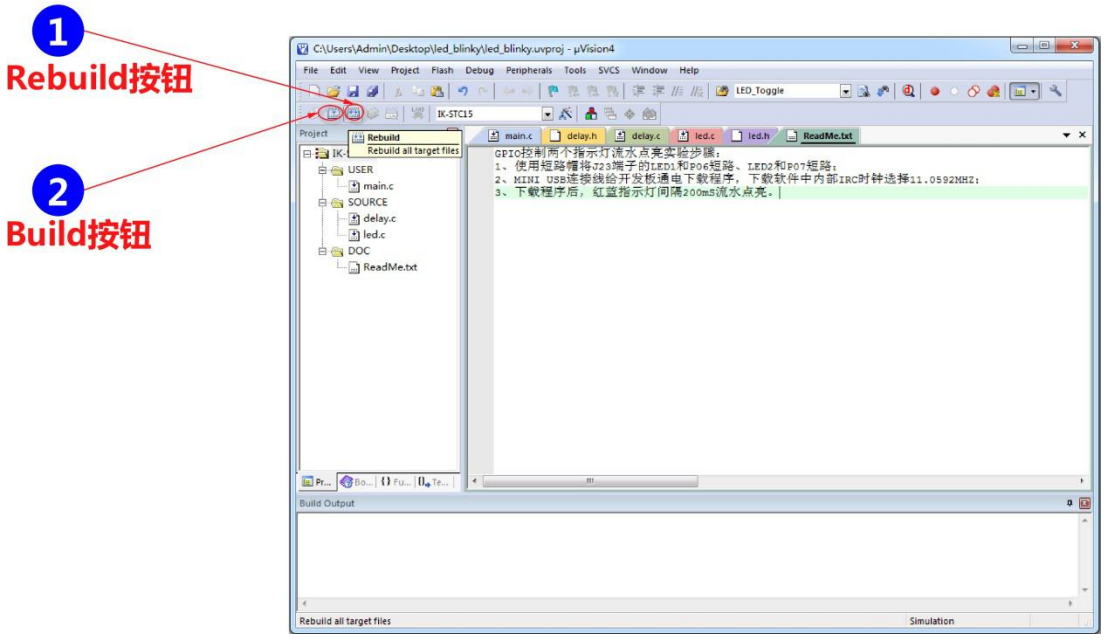


图 55：Keil C51 编译工程选项

✧ 注：首次编译工程，建议操作 Rebuild 按钮。

2. 完成对工程的编译。

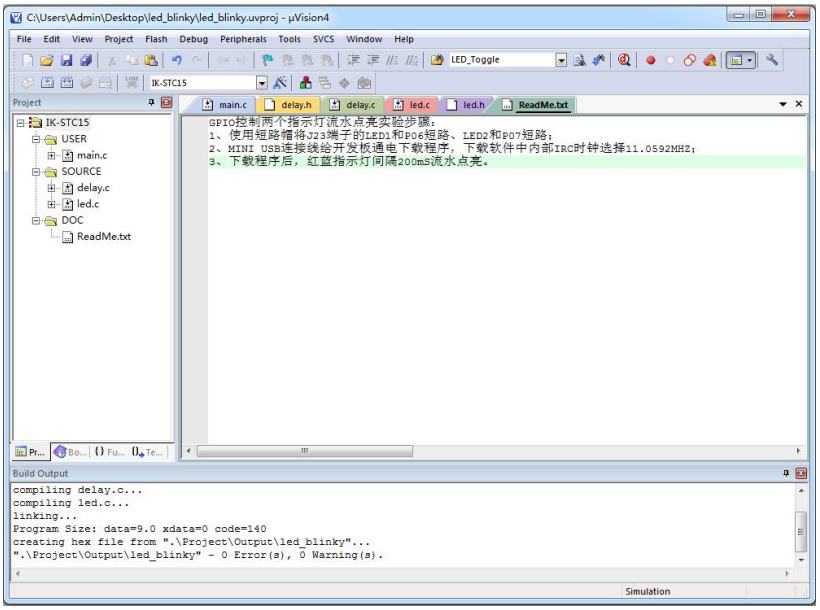


图 56：Keil C51 编译工程完成

✧ 注：可以看到编译结果是 0 错误，0 报警。