第 1-3-1 讲：新建工程模板–新建和配置工程

1. 规划工程存放目录

建立工程之前，我们需要先考虑一下工程文件的组织，也就是工程中的各个文件在电脑中存放的目录。清晰的工程目录不但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，尤其是在我们开发复杂程序的时候，作用更为明显。

为了在学习开发之初就能养成良好的习惯，本书中所有的例子都会规划工程存放目录，而不是把工程中用到的文件都放到一个文件夹里面。

规划工程存放目录的时候，我们可以根据自己的习惯和喜好来建立自己的工程目录，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存放目录，供大家参考。

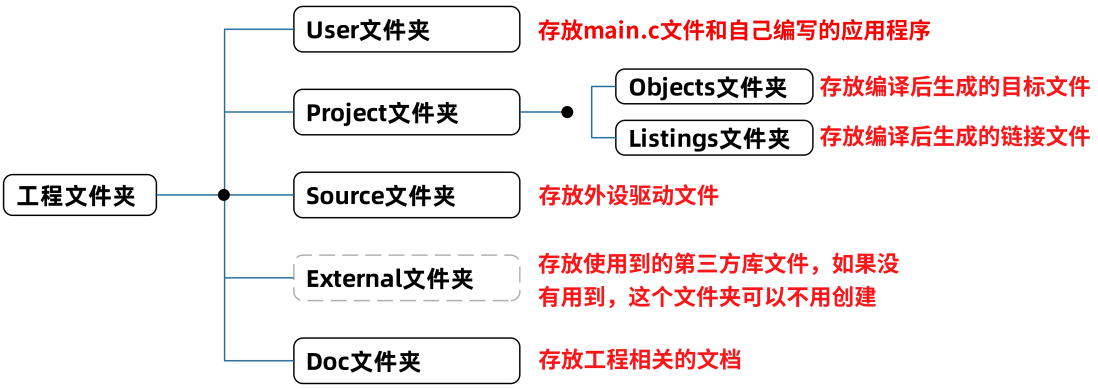

图 1：工程存放目录

其中：

- User 文件夹：存放 main.c 文件和自己编写的应用程序。
- Project 文件夹：存放创建工程生成的相关文件。
  - Objects 文件夹：用于存放工程编译后生成的目标文件。
  - Listings 文件夹：用于存放工程编译后生成的链接文件。
  - ✧ **说明**：如果读者使用的 Keil C51 是 9.60a 版本的，则无需手动创建“Objects”和“Listings”文件夹，新建工程后 Keil 会自动创建这 2 个文件夹。
- Source 文件夹：存放外设驱动文件
- External：存放使用到的第三方库文件，如果没有用到，这个文件夹可以不用创建。
- Doc 文件夹：用于存放说明之类的文档。

2. 建立目录、拷贝头文件

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点灯的工程来作为

我们的工程模板，并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 D 盘。

1. 创建工程存放目录

按照前文所述，首先，在 D 盘新建一个名称为 led_blinky 的文件夹，然后在该文件夹里面新建如下图所示的文件夹，并在 Project 文件夹里面新建 Objects 和 Listings 文件夹。因为本例中没有用到第三方库文件，因此，没有创建 External 文件夹。

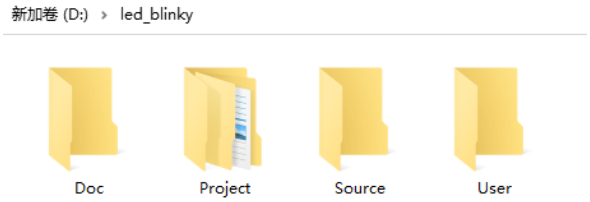

图 2：工程存放目录

2. 拷贝头文件

将 STC8.h 头文件拷贝到 User 文件夹中，可以从教程配套的例子里面拷贝。

✧ 说明：STC8.h 头文件是 STC（宏晶科技）提供的适用于 STC8A8K64D4 系列单片机的头文件，用户直接复制调用即可。

建立工程存放文件夹和拷贝库文件之后，我们就可以开始建立工程了。

3. 新建工程

1. 启动 Keil C51，点击[Project]，在弹出的下拉菜单中选择[New uVision Project]新建工程。

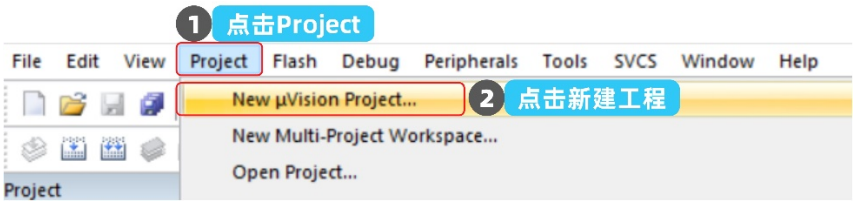

图 3：打开新建工程窗口

2. 设置工程名和工程保存路径（保存到我们之前创建的 led_blinky 文件夹里面的 Project 文件夹），设置完成后点击[保存]。

✧ 注意事项：工程路径和工程名不要包含汉字字符(虽然有些计算机使用汉字字符没有问题，但是还是建议不要使用汉字字符，因为 Keil 对汉字字符的支持比较差)，同时路径不要过深，否则打开工程或仿真时可能会出现问题。

图 4：设置工程保存路径和工程名称

3. 选择器件库：在弹出的窗口中选择[STC MCU Database]后，点击[确定]。

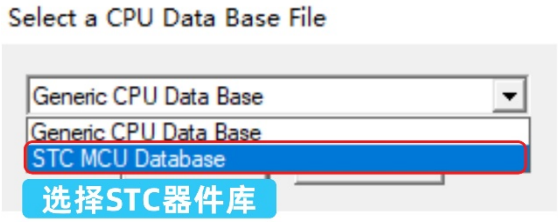

图 5：选择器件库

4. 选择器件：在弹出的窗口中选择单片机的型号，如下图所示。

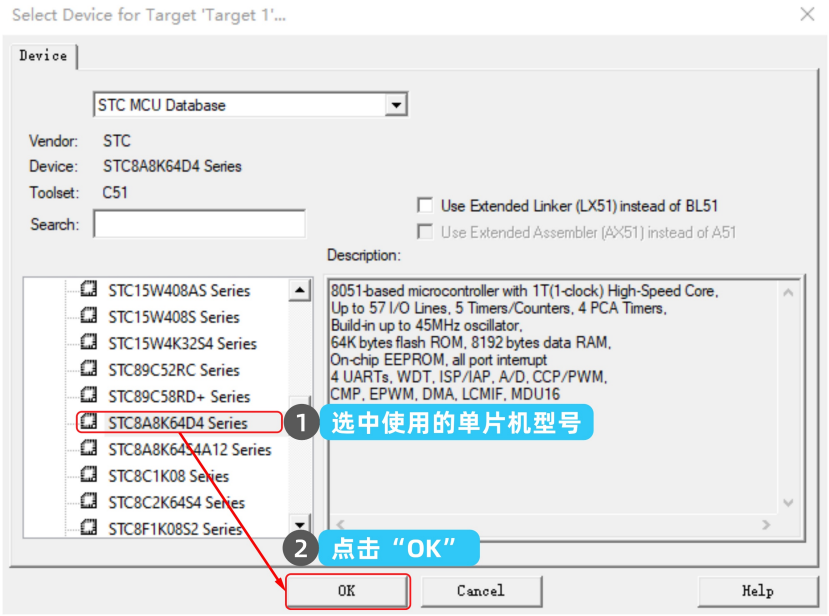

图 6：选择器件

✧ 说明：这里的选项 STC8A8K64D4 Series 表示的意思是 STC8A8K64D4 这一系列的单片机，STC8A8K48D4 也属于 STC8A8K64D4 系列。

5. 当出现的下面提示框时，选择[否]。

startup.a51 是 Keil C51 的启动代码，工程中会默认使用 Keil C51 安装目录中的启动文件，因此，不需要将其添加到工程。如果读者需要修改启动文件，这一步就需要选择[是]，将启动文件添加到工程中修改。

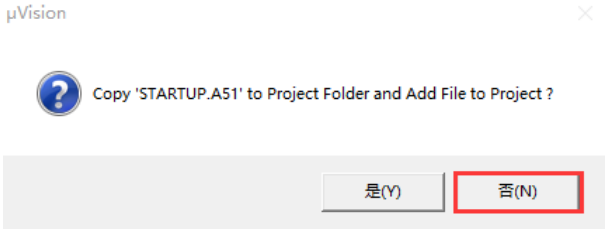

图 7：不添加“STARTUP.A51”

6. 新建的工程界面如下，可以看到，工程是空的，什么都没有，因此接下来我们还要整理 Keil 里面的工程目录、添加文件和编写代码。

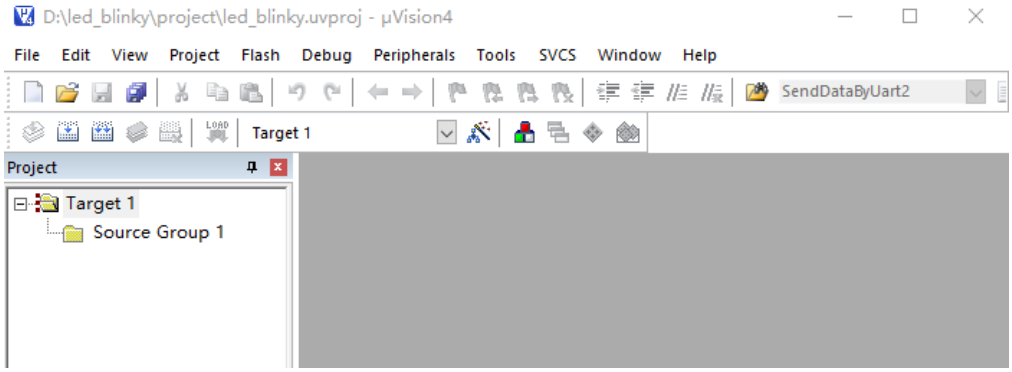

图 3-6：新建的工程

7. 管理 Keil C51 工程目录。

Keil C51 工程目录指的是 Keil C51 中 Project 栏的目录（如下图所示），并不是前文中的工程在电脑中的存放目录，这一点要注意区分。

下图是整理前和整理后的 Keil C51 工程目录，主要操作是修改 Target 或组名称、添加组以及调整组的顺序。这么做的目的是为了目录清晰，方便添加文件和管理文件。

✧ 说明：Keil C51 工程窗口中的工程目录不需要完全和工程文件存放目录一样，这里的目录用于分类显示添加到工程的文件，方便用户管理工程中的文件。

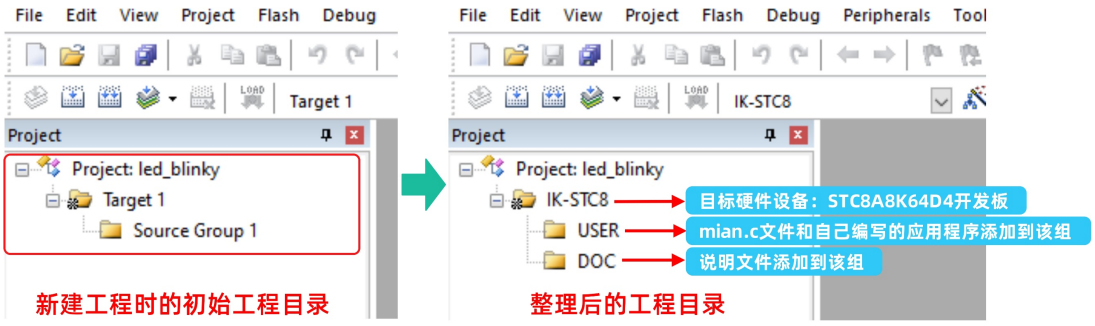

图 8：整理工程目录

整理目录时会用到下面几个操作：

1) 修改 Target 名称（修改组名称的方法一样）：先选中需要修改名称的组，然后单击即可修改名称。注意，不是双击。
2) 添加组：选中 Target(即整理后的目录中的 IK-STC8)，右键选中“Add Group”，即可添加一个组。
3) 调整组/文件的顺序。

如下图，点击工程管理按钮，打开工程管理窗口。

图 9：点击工程管理按钮

按照下图所示根据自己的习惯调整组/文件的顺序，调整时，先选中需要调整的组/文件，然后点击向上/向下的箭头即可将组/文件的位置上移/下移。

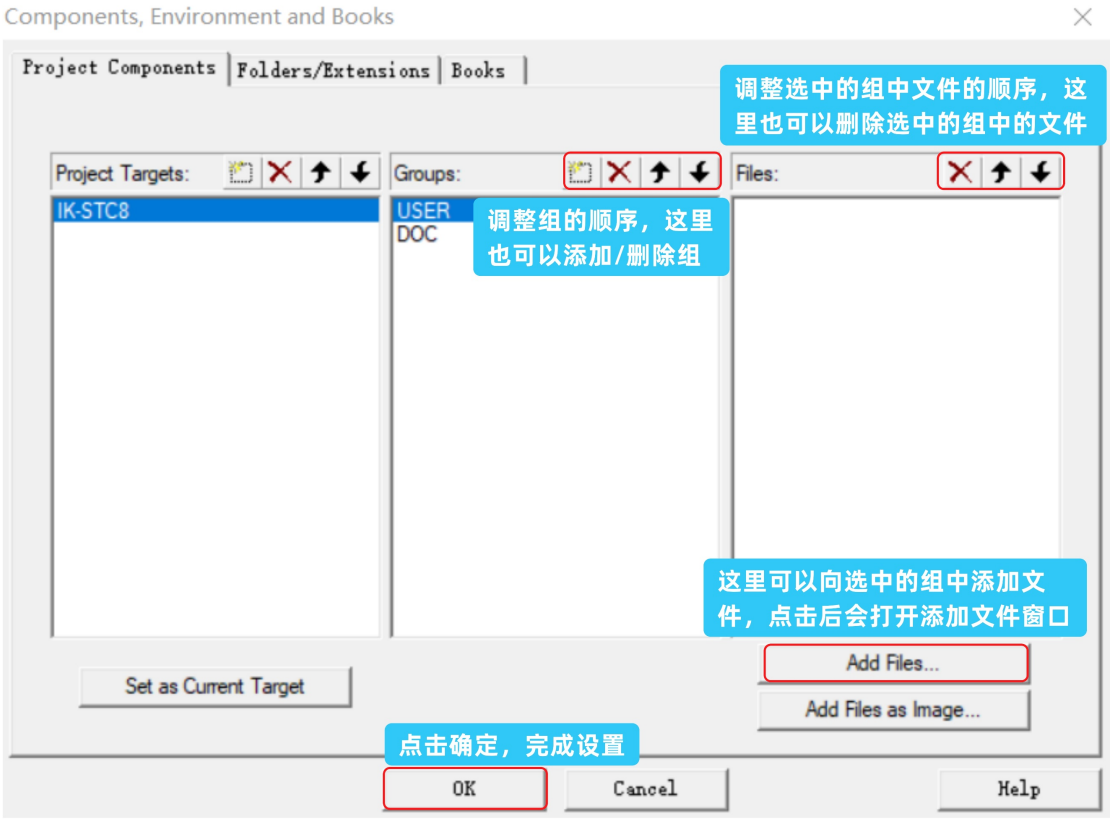

图 10：调整组/文件的顺序

4. 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名称为“main.c”的文件并添加到工程。

如下图，点击新建文件的快捷按键新建文件，也可以执行“File→New”新建文件。

图 11：新建文件

新建的文件如下图所示，新建的文件名称默认是 Text1。

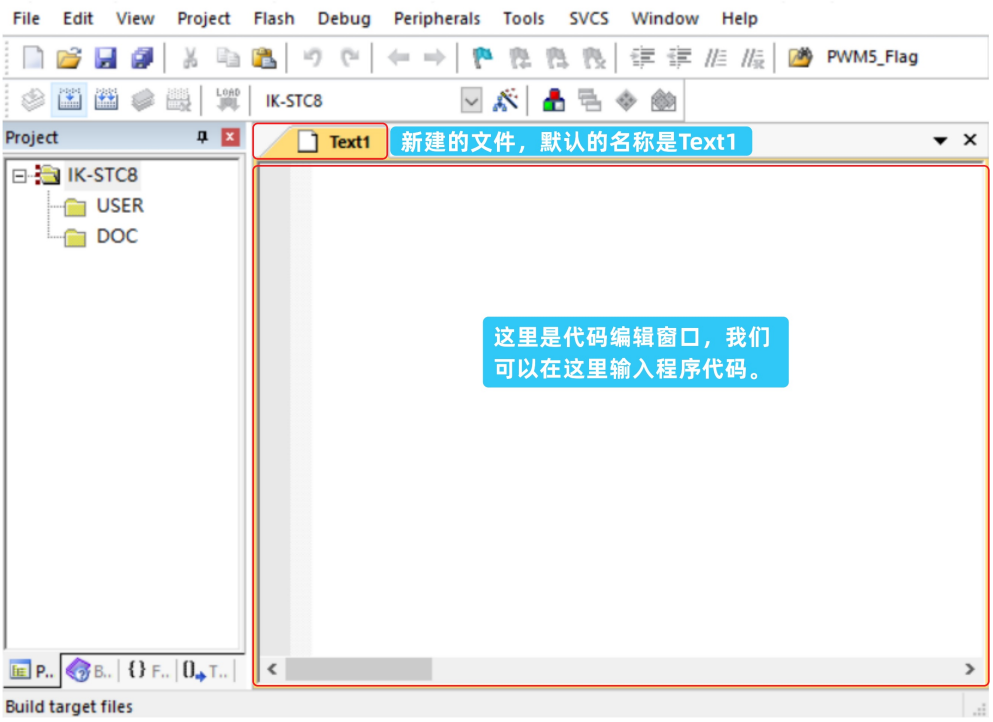

图 12：新建文件的界面

这时，虽然新建了文件，但是他还没有和工程关联起来，接下里，我们还需要保存文件并把该文件添加到工程，从而和工程联系起来。

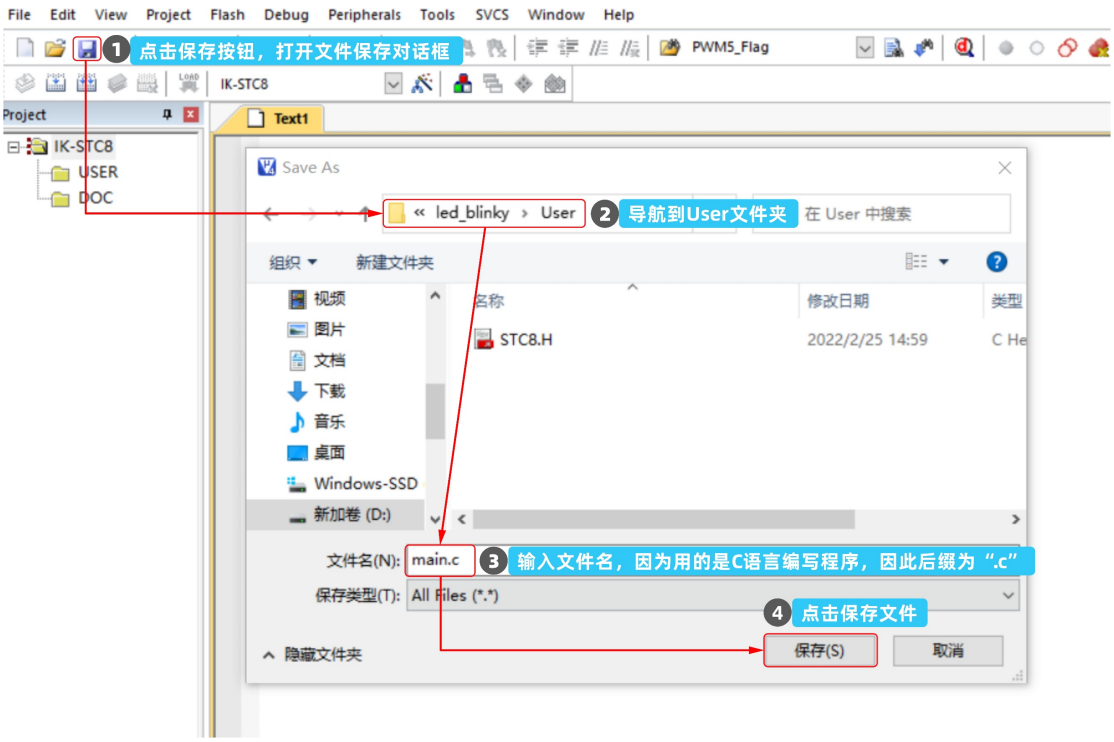

图 13：保存文件

保存 main.c 文件后，按照下图所示，将文件添加到工程，需要添加到哪个组，就双击组名打开添加窗口（也可以选中组名后右键，然后点击“Add Existing Files to Group ‘USER’”）。这里，我们将 main.c 添加到 User 组。

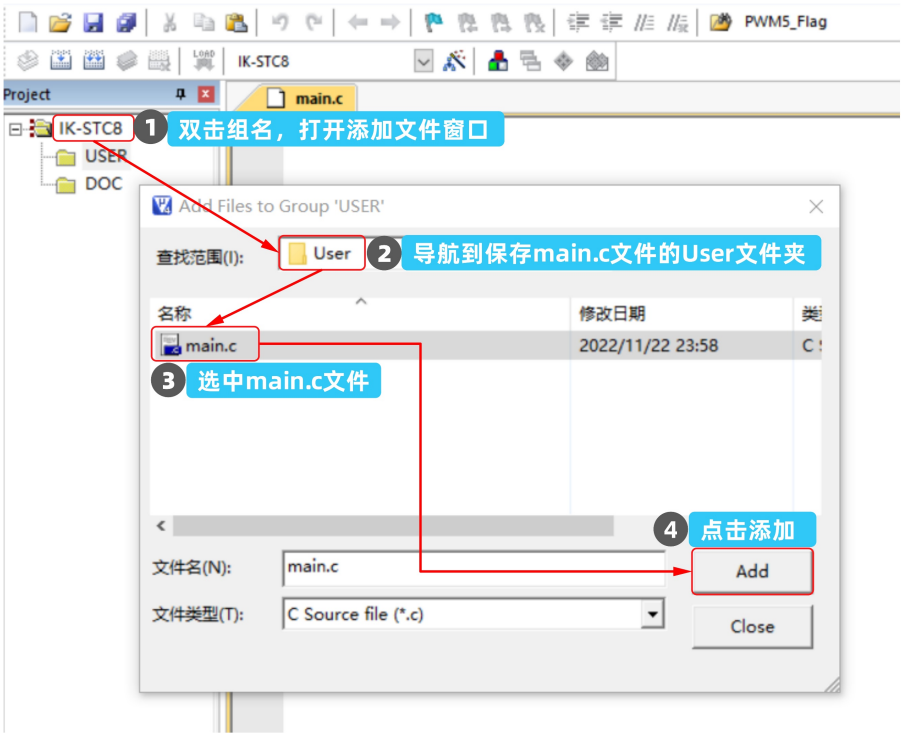

图 14：添加文件到工程

到这里，我们学习了如何新建工程、文件以及如何将文件添加到工程，接下来，还需

要对工程进行配置。

5. 配置工程

点击魔术棒，打开工程配置窗口，即可对工程进行配置。

图 15：点击魔术棒

5.1. 配置“Target”

“Target”配置页面如下图所示。

✧ 说明：Device 选项卡是选择单片机型号，这在新建工程的时候已经设置过了，这里不再赘述。

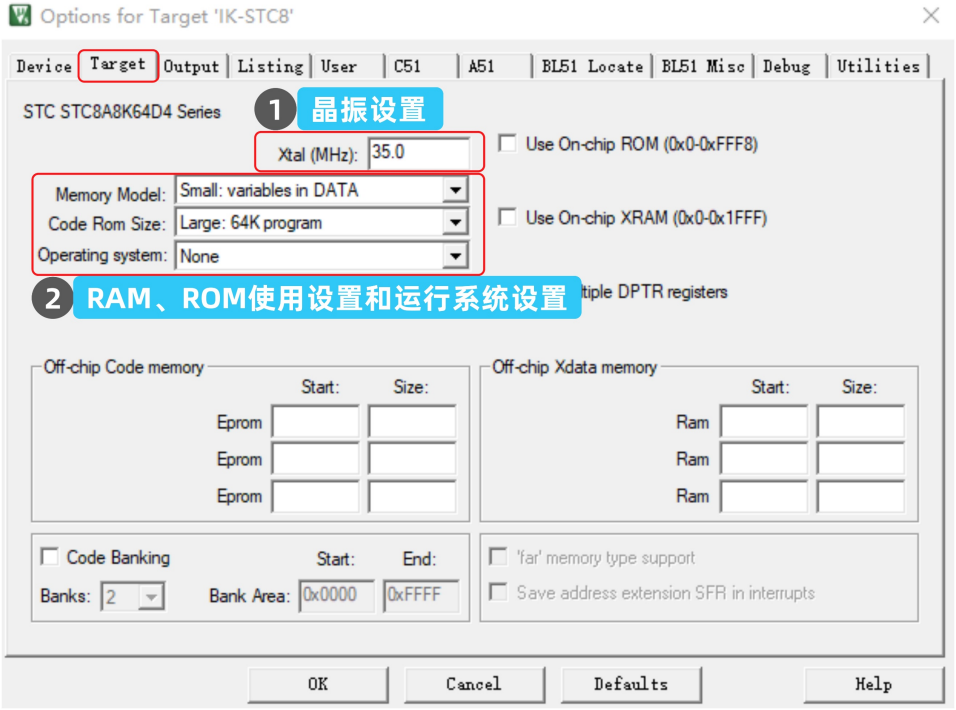

图 16：“Target”配置

“Target”配置页面使用默认配置即可，不过里面的一些设置我们需要了解一下。

1) 晶振频率：晶振频率 Xtal 是用于软件仿真的，设置或不设置对硬件烧写程序和硬件仿真都没有影响。

2) Memory Model 选项，用于设置程序对 RAM 的使用，有下面 3 个可选项。

- Small:variables in DATA：程序中使用的变量默认存储在内部 RAM（idata 类型）。
- Compact:variables in PDATA：程序中使用的变量默认存储在外部 RAM（pdata 类型，8 位寻址）。
- Larget:variables in XDATA：使用全部扩展 RAM（xdata 类型，16 位寻址）。

✧ 说明：通常我们建议使用 Small 模式，他的好处是我们在程序中定义的变量默认都会

存储到内部 RAM，访问速度快。如果内部 RAM 不够用或者我们希望某个变量存储到外部 RAM 时，使用关键字 xdata 修饰该变量即可。由此可见，使用 small 模式不会浪费访问速度快的内部 RAM，也可以通过关键字修饰变量使用外部 RAM，因此，建议使用 small 模式。

3) Code Rom Size 选项，用于设置程序对 ROM 的使用，有下面 3 个可选项。

- Small:program 2k or less：只用低于 2K 的程序空间，适用于片内 ROM 不超过 2K 单片机。
- 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。
- Large:64k program：单个函数或整个工程都可用全部 64K 空间。

我们使用的是 STC8A8K64D4，片内 Flash 为 64KB，因此选择 Large 模式。

4) Operating system 选项，用于设置操作系统，有下面 3 个可选项。

- None：不使用操作系统。
- 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。

通常，我们不会使用 TINY 或 FULL 操作系统，因此，选择 None。

5.2. 配置“Output”

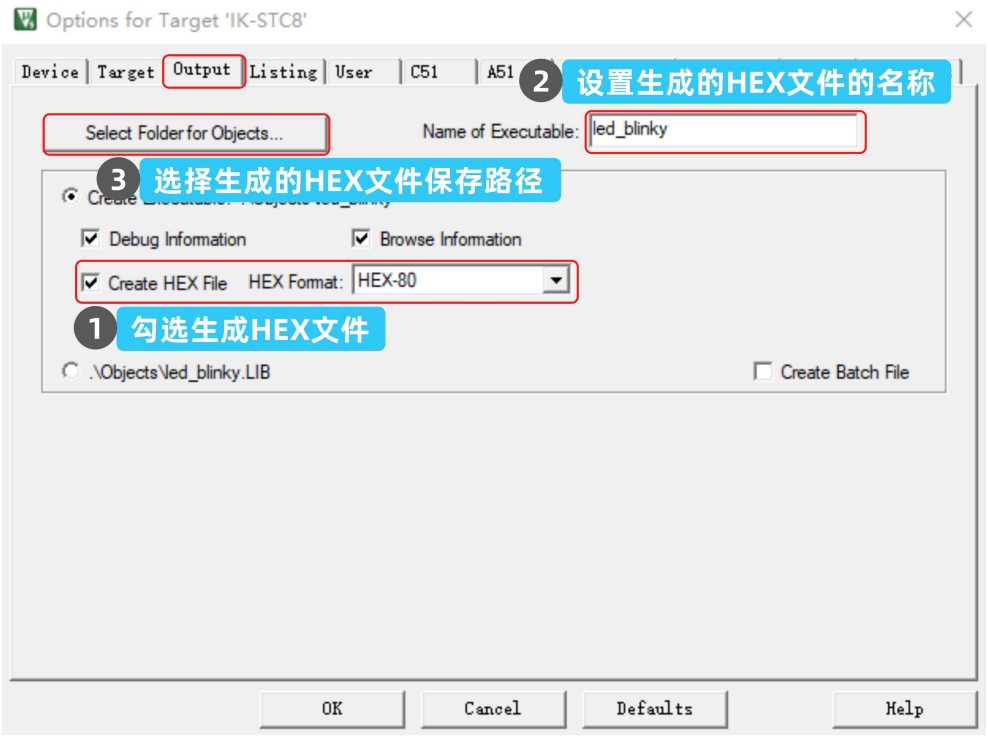

图 17：配置“Output”

主要配置下面几个项目：

1) 勾选“Create HEX File”，程序编译后生成 HEX 文件，我们烧写程序的时候需要使用程序生产的 HEX 文件，因此，这里需要勾选。

2) 设置生成的 HEX 文件名称。

3) 选择生成的目标文件保存路径，点击打开路径设置窗口，导航到工程的“…\Project\Objects”文件夹，然后点击“OK”按钮确定即可。

5.3. 配置 Listing

点击“Select Folder for Listings”按钮，打开路径设置窗口，导航到工程的“…\Project\Listings”文件夹，然后点击“OK”按钮确定即可。

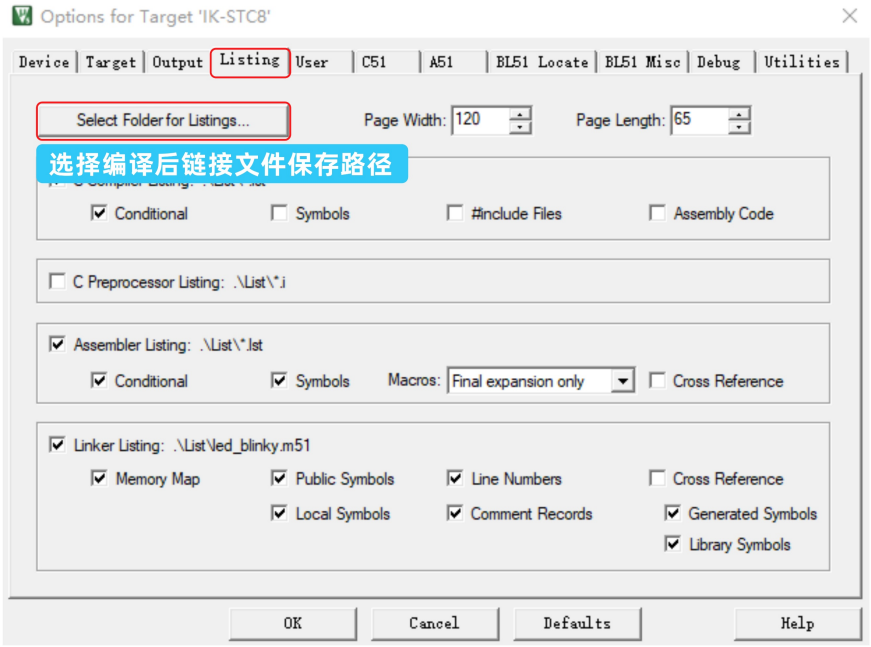

图 18：配置“Listing”

5.4. 配置 C51

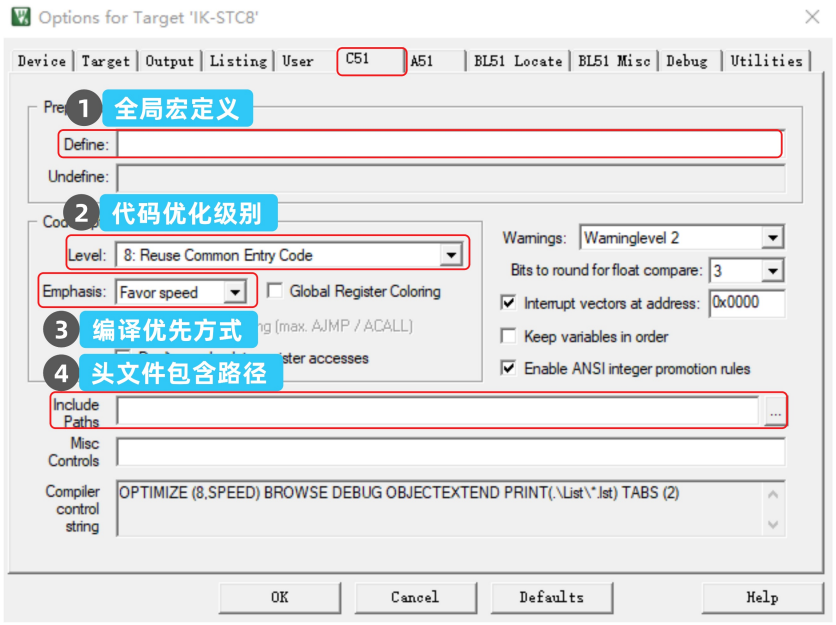

图 19：配置“C51”

“C51”配置页面最常用的是全局宏定义、代码优化级别和头文件包含路径配置，另外，编译优先方式我们也需要了解一下。

1. 全局宏定义

定义被整个工程使用的宏定义，如果有多个宏定义，用空格隔开。注意，这里定义的宏定义，对于整个工程有效。

2. 代码优化级别“Code Optimization”

0 表示不优化，设置越大，优化级别越高。一般仿真调试的时候，优化级别设置为 Level 0（最低），调试完成后，设置为 Level 3（最高），以减小编译后的代码大小。

3. 编译优先方式“Emphasis”，他用于设置编译器优先方式，有一下 3 个可选择项目.

- Favor speed：速度优先，即编译时以代码运行速度快为优先项。
- Favor size：编译后生成的目标文件大小优先，即编译时以目标文件大小为优先项。
- <default>：默认，速度优先。

通常，我们使用的是：速度优先。

4. 头文件包含路径

当我们引用了一个头文件时，就需要告诉编译器这个头文件的路径，否则，编译器可没这么聪明，可以自己在电脑中找到头文件（除非把所有的头文件都放到工程文件“.uvproj”目录，因为 Keil C51 是知道工程文件的路径的，不过，这样做是不推荐的），当然，编译的时候就会因为找不到头文件产生错误。

头文件包含路径有几种设置方法，最常用的方式就是在 Keil C51 开发环境中设置(其他几种方式不方便，也不灵活，极少被使用，故在此不提)，即在工程配置的“C51”页面里面设置，添加头文件包含路径的方法如下。

点击上图中头文件设置右边的添加头文件路径的按钮，打开添加头文件路径的窗口，之后按照下图所示方法添加头文件，如果需要添加多个头文件，重复添加步骤即可。

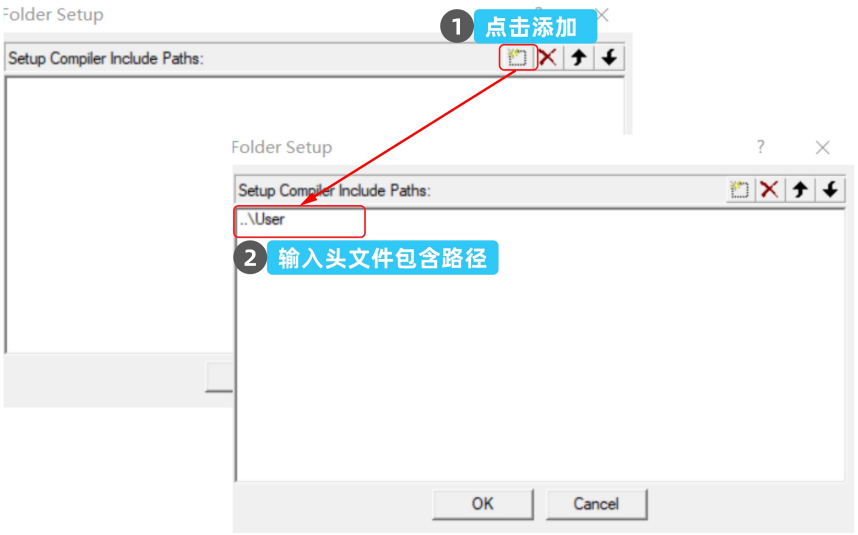

图 20：添加头文件

上图中，我们设置了“STC8.H”这个头文件的路径“..\User”，要明白为什么这样设置，就需要了解头文件包含路径的表示方法。

通常，头文件路径使用相对路径表示，他是相对于工程文件“led_blinky.uvproj”的路径。使用相对路径的优点是：拷贝整个工程到其他盘，不会影响到工程，因为路径是相对的。相对路径中必须要掌握的几种路径表示方法：

- "..\"：表示工程文件所在目录向上一级的目录，"..\"可以连用，“..\..\” 表示工程文件所在目录向上两级的目录，以此类推。
- ".\"：表示的是当前路径。

知道了路径的表示方法，我们再看本例中“STC8.H”头文件的路径，“STC8.H”头文件位于工程的 User 文件夹，如下图所示。

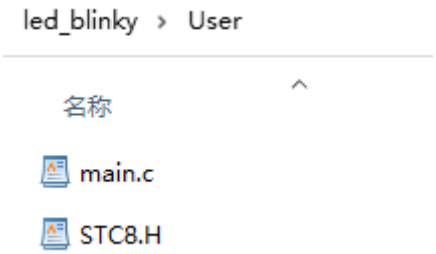

图 21：“STC8.H”头文件的路径

而 User 文件夹在工程中的路径是工程文件所在目录向上一级的目录，因此，用相对路径表示“STC8.H”头文件的路径为：“..\User”。

5.5. 配置“BL51 Misc”

“BL51 Misc”配置页面常用的配置项是设置警告忽略，读者可以根据需要决定是否设置。这里我们设置了忽略 16 号警告，以避免程序中存在未调用的函数而在编译的时候出现警告。

✧ 说明：当我们定义了一个函数，但程序里面没有使用该函数时，即会产生 16 号警告，这个警告对程序的功能是没有影响的。尤其在使用库函数的时候，程序一般不可能调用库里面所有的函数，此时，可以配置忽略次警告。

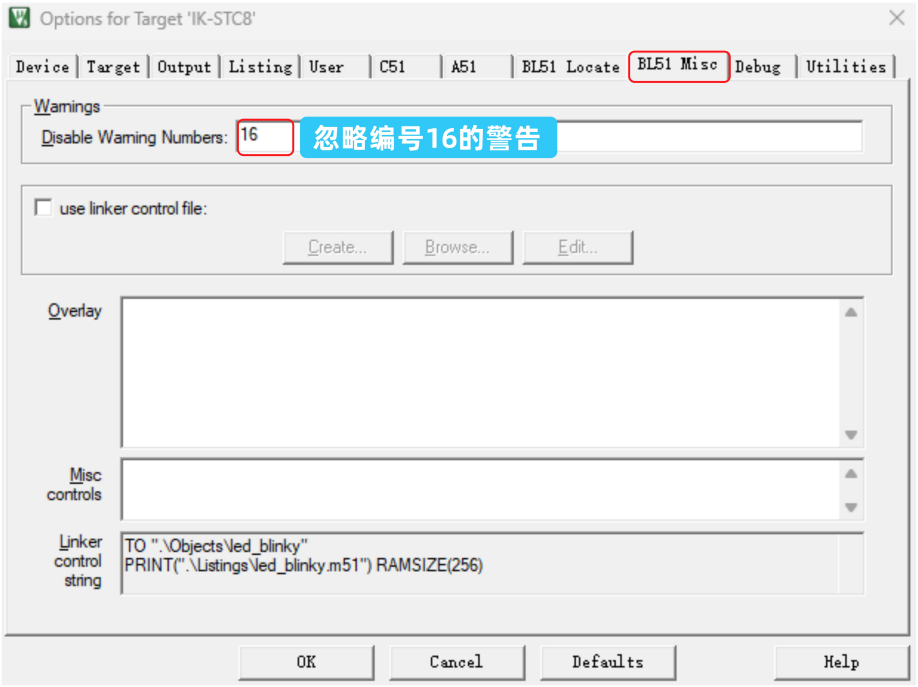

图 22：忽略 16 号警告

到这里，工程配置就完成了，下一章中我们再讲解如何编译工程和下载运行。