

AI8051U 库函数如何建立一个项目

更新日期：2025 年 1 月 22 日

目录

1、安装 Keil C251（已经安装可以跳过此步）	2
下载必要的文件	2
安装流程	2
2、建立一个新工程	5
新建一个项目文件夹	5
新建工程	5
选择单片机	5
设置 4Byte 字节对齐	6
设置 Xsmall 模式	7
使用 REMOVEUNUSED 功能	7
选择输出 HEX 文件功能	8
3、按需添加工程文件	9
添加文件	9
添加 main.c 文件（包含 main 函数的 C 文件）	9
添加头文件定义（可从例程开始复制）	10
简单修改，编译	11
下载时的注意事项	12
4、反馈渠道	13
官方网站： https://www.stcaimcu.com/	13
库函数相关反馈渠道：	13
https://www.stcaimcu.com/forum.php?mod=viewthread&tid=13629	13

1、安装 Keil C251（已经安装可以跳过此步）

下载必要的文件

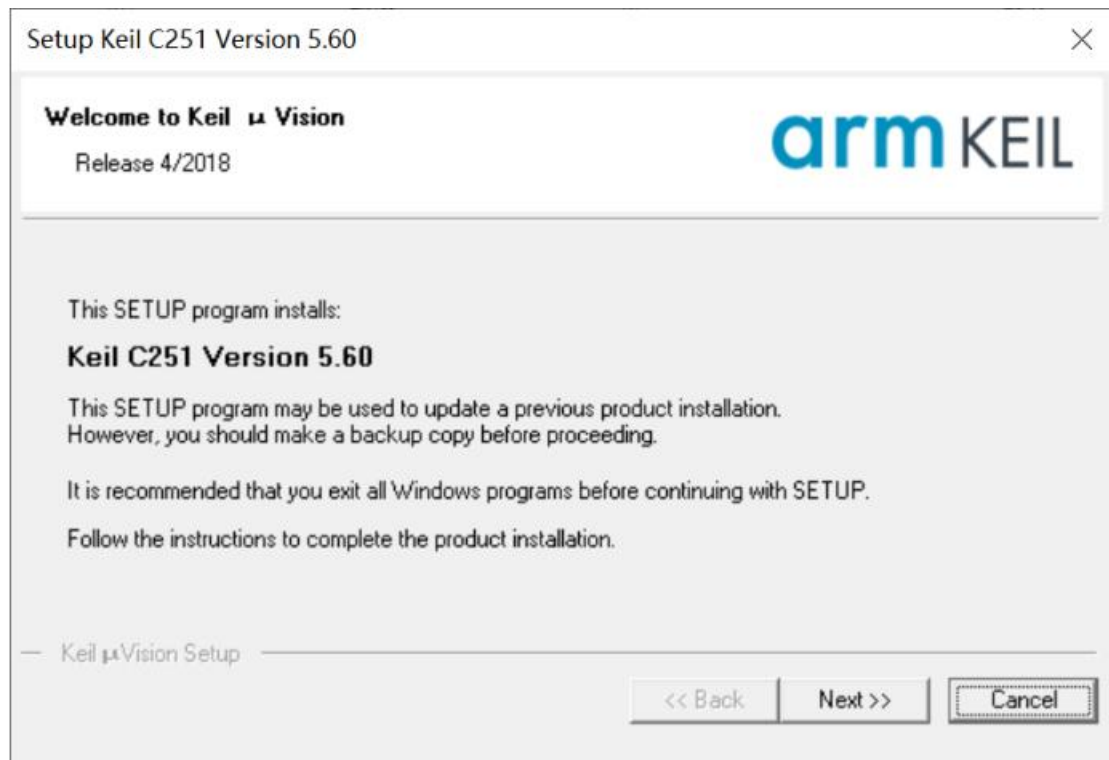
Keil 官网下载链接: [Keil Product Downloads](#)（点击跳转浏览器下载）

AI8051U 库函数: [前进中的 Ai8051U 专属库函数](#), 听取建议, 不断提高 - [软件库函数/原理图库/PCB 库/最小包装 国芯技术交流网站](#) - [AI32 位 8051 交流社区](#)

安装流程

信息随便填写, 点确定后进入下载页面进行下载。

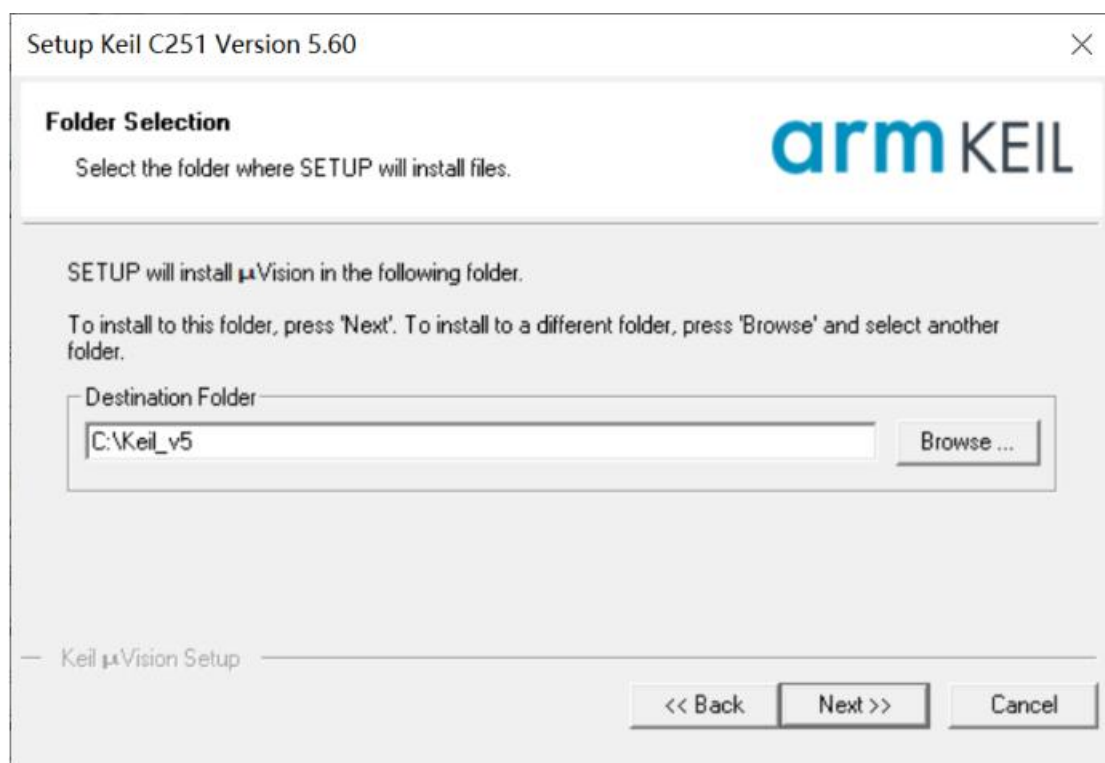
双击下载的安装包开始安装, 点击 “Next” :



勾选 “I agree to all the terms of the preceding License Agreement”, 然后点击 “Next” :



选择安装目录，然后点击“Next”：



填写个人信息，然后点击“Next”：

Setup Keil C251 Version 5.60

Customer Information

Please enter your information.

Please enter your name, the name of the company for whom you work and your E-mail address.

First Name:

Last Name:

Company Name:

E-mail:

— Keil μ Vision Setup —

<< Back Next >> Cancel

安装完成，点击“Finish”结束。

Setup Keil C251 Version 5.60

Keil μ Vision Setup completed

Keil C251 Version 5.60

μ Vision Setup has performed all requested operations successfully.

☒ Show Release Notes.

☒ Add example projects to the recently used project list.

— Keil μ Vision Setup —

<< Back Finish Cancel

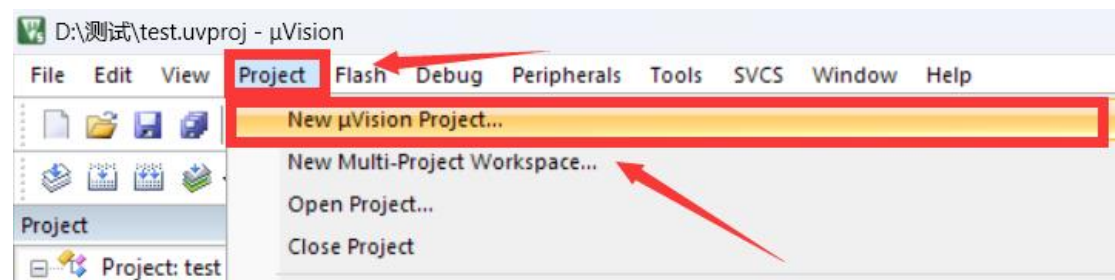
2、建立一个新工程

新建一个项目文件夹

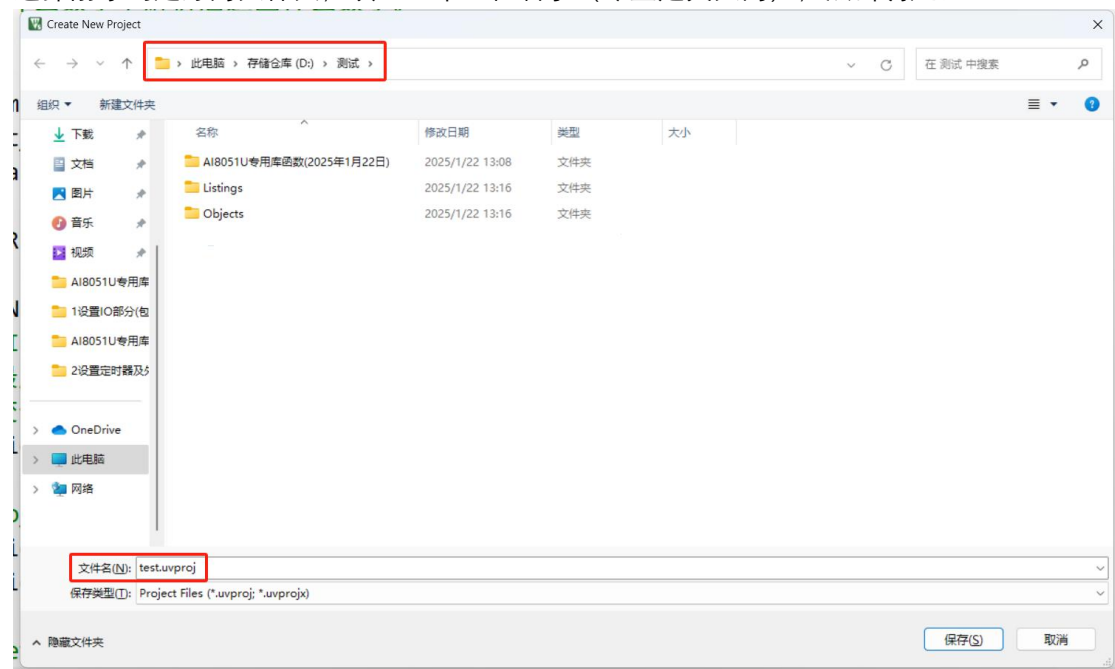
在一个纯英文目录下（例如 D 盘根目录）新建一个文件夹，用于存放项目的文件。不直接在桌面创建是因为路径有可能包含中文。

新建工程

打开 Keil 软件，在上方的'Project'选项中，找到 New uVision Project 选项，进行新建工程。

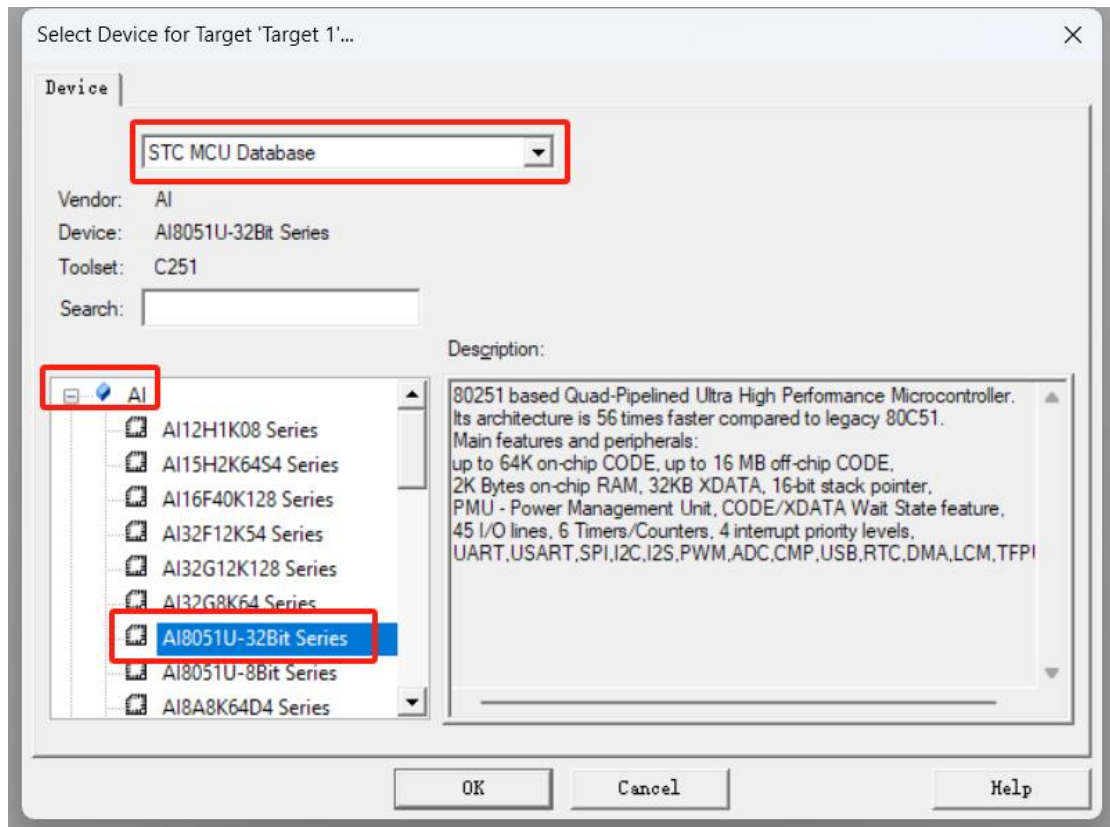


选择刚才创建好的文件夹，填入一个工程名字（尽量是英文的），点击确定。



选择单片机

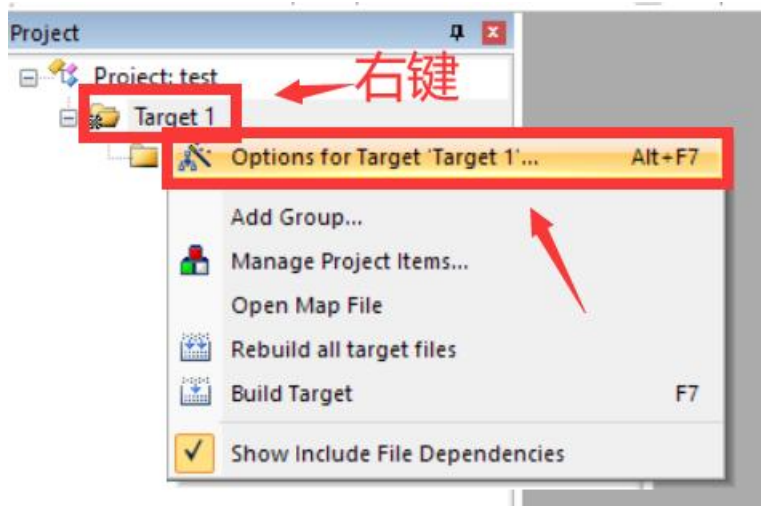
在上方的 Database 选项中选择'STC MCU Database'，然后展开 AI 选项，找到其中的 'AI8051U-32Bit'选项点击，最后点击'OK'以完成单片机选择。

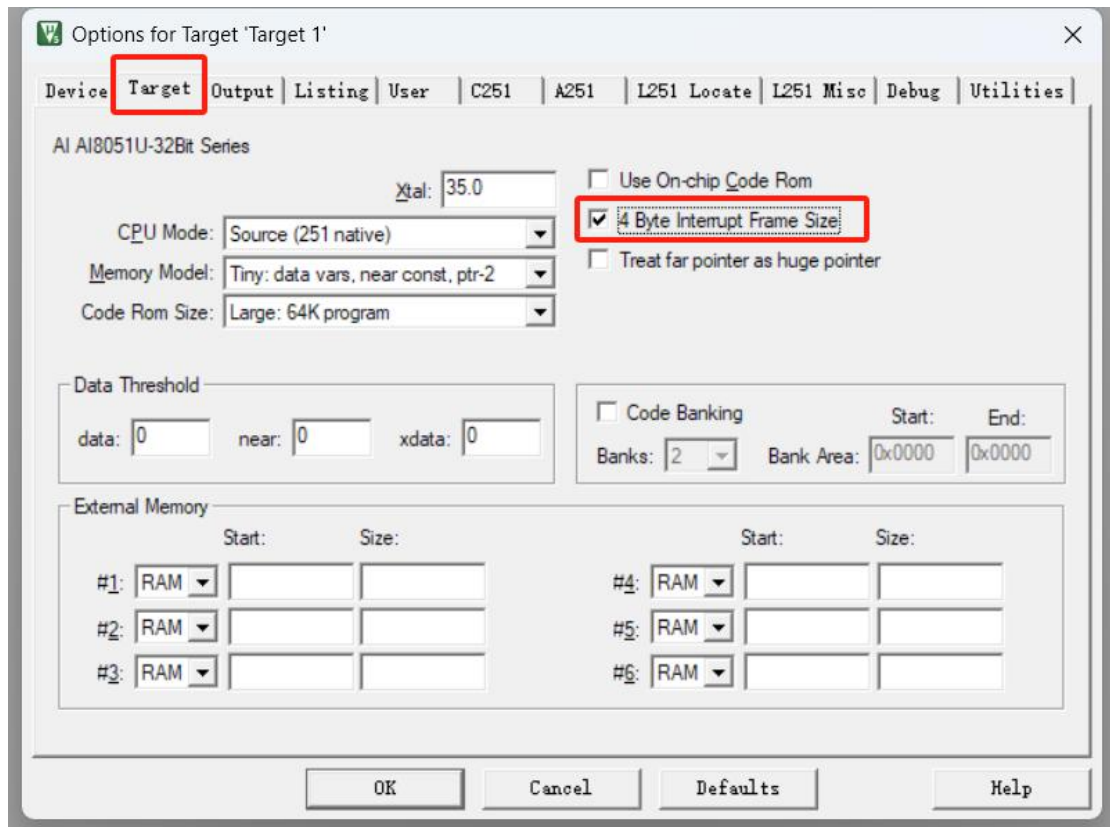


设置 4Byte 字节对齐

选择完单片机后，项目是空白的。此时需要设置一下工程的选项。

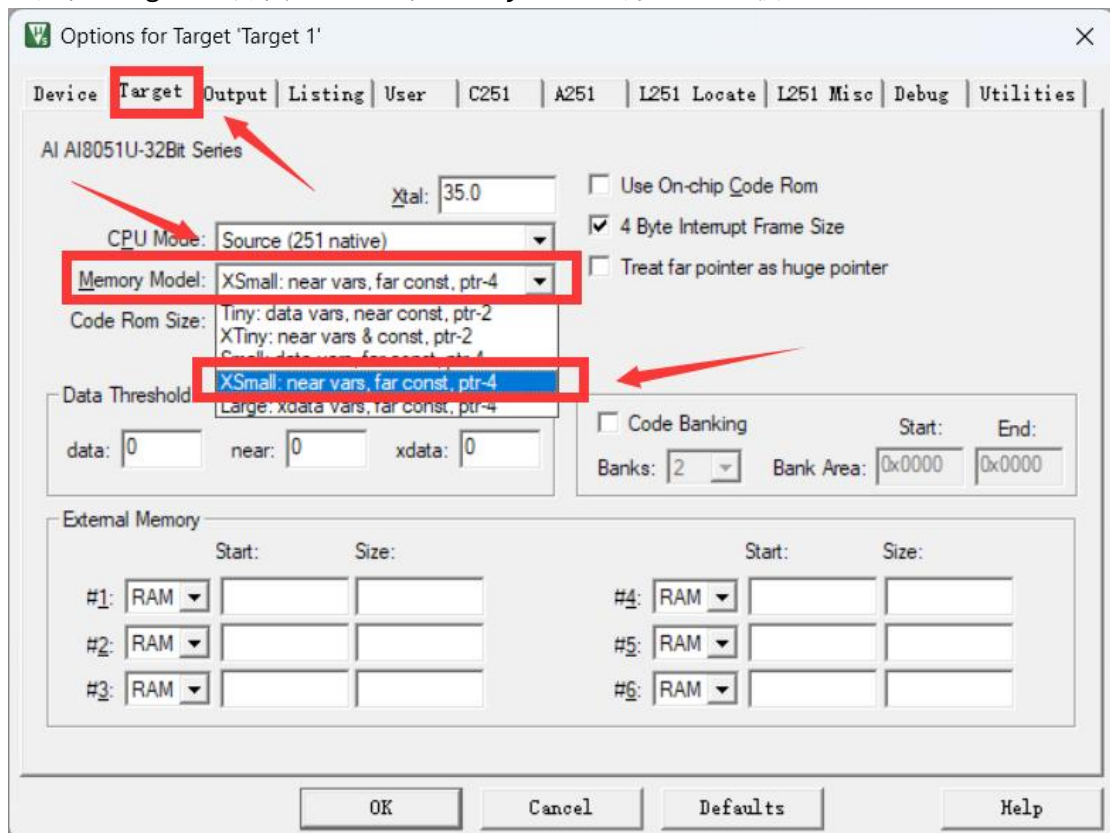
在左侧的 Project 选项卡中，右键 Target 1 文件夹图标，然后选择 'Options for Target 'Target 1'...' 选项，进入后默认就是 Target 选项卡，如图所示点击 '4Byte Interrupt Frame Size' 选项，以规范程序内部调用指针时候的对齐，防止出现奇怪的问题。





设置 Xsmall 模式

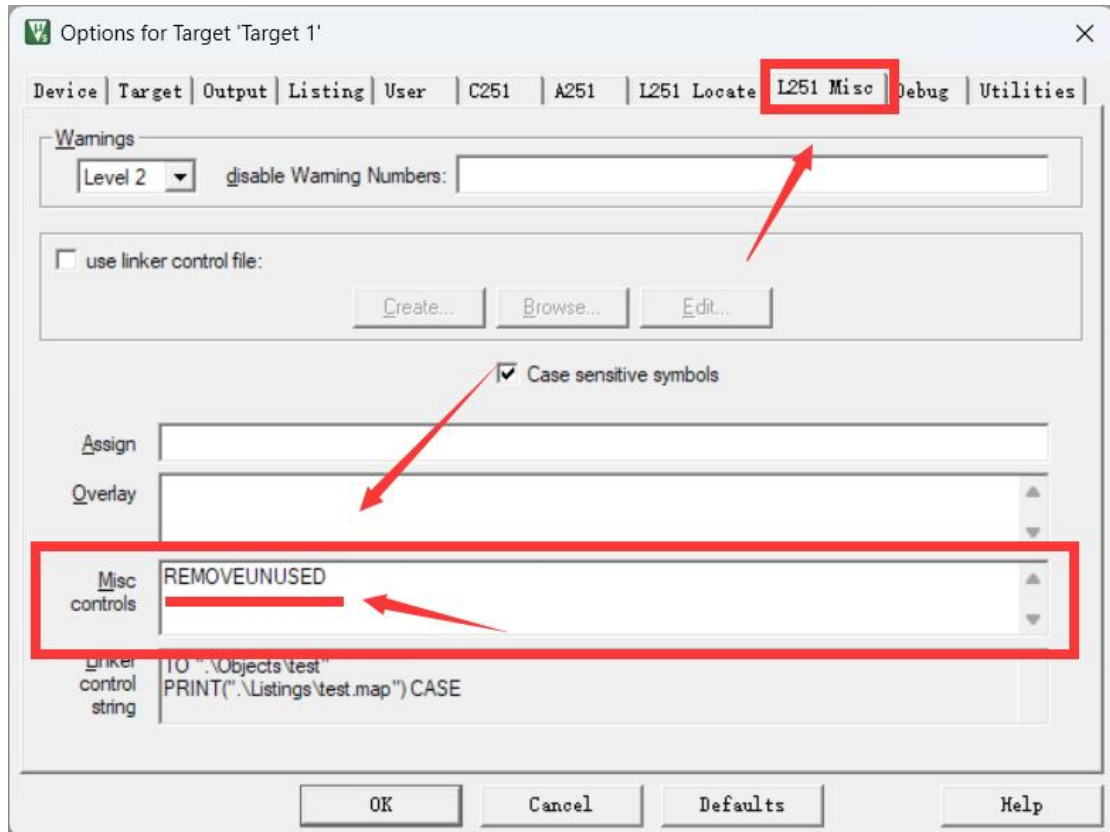
还是在 Target 选项卡，这次选择 Memory Model 为 'XSmall' 模式



使用 REMOVEUNUSED 功能

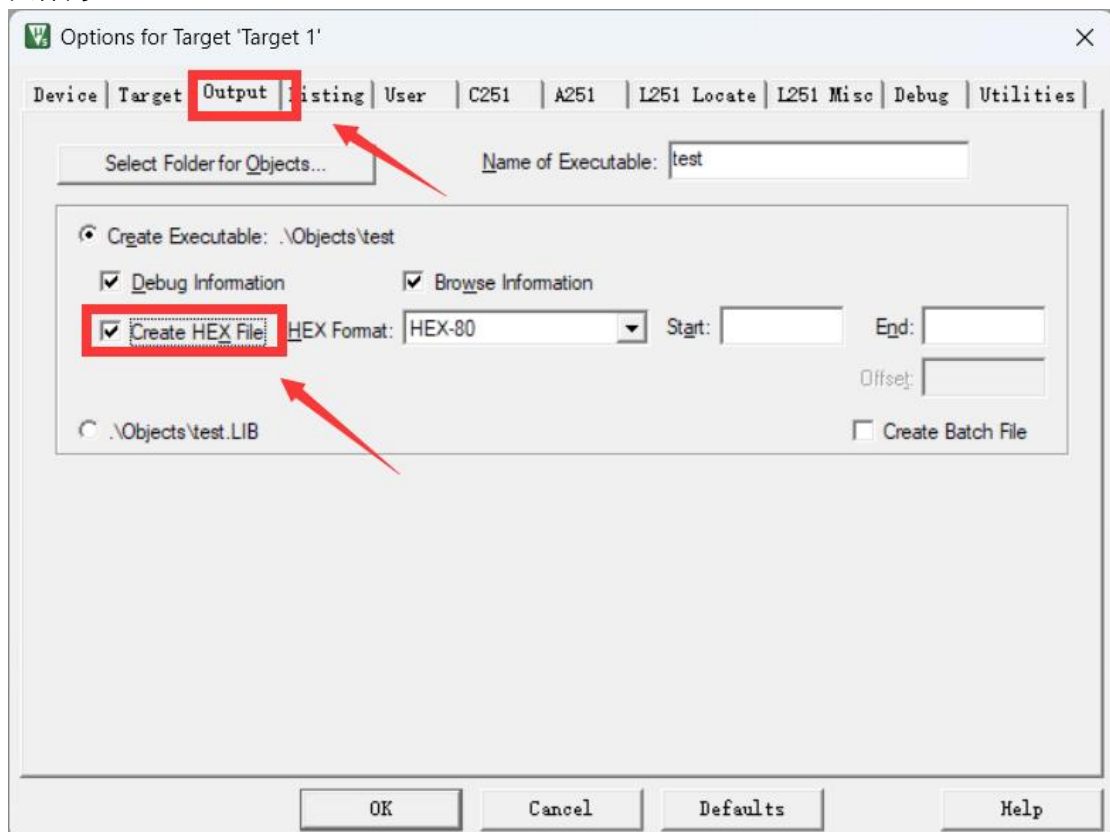
接下来切换到 L251 Misc 选项卡，在 Misc controls 输入框中，输入大写的

'REMOVEUNUSED'，以实现不使用的函数不进行调用。



选择输出 HEX 文件功能

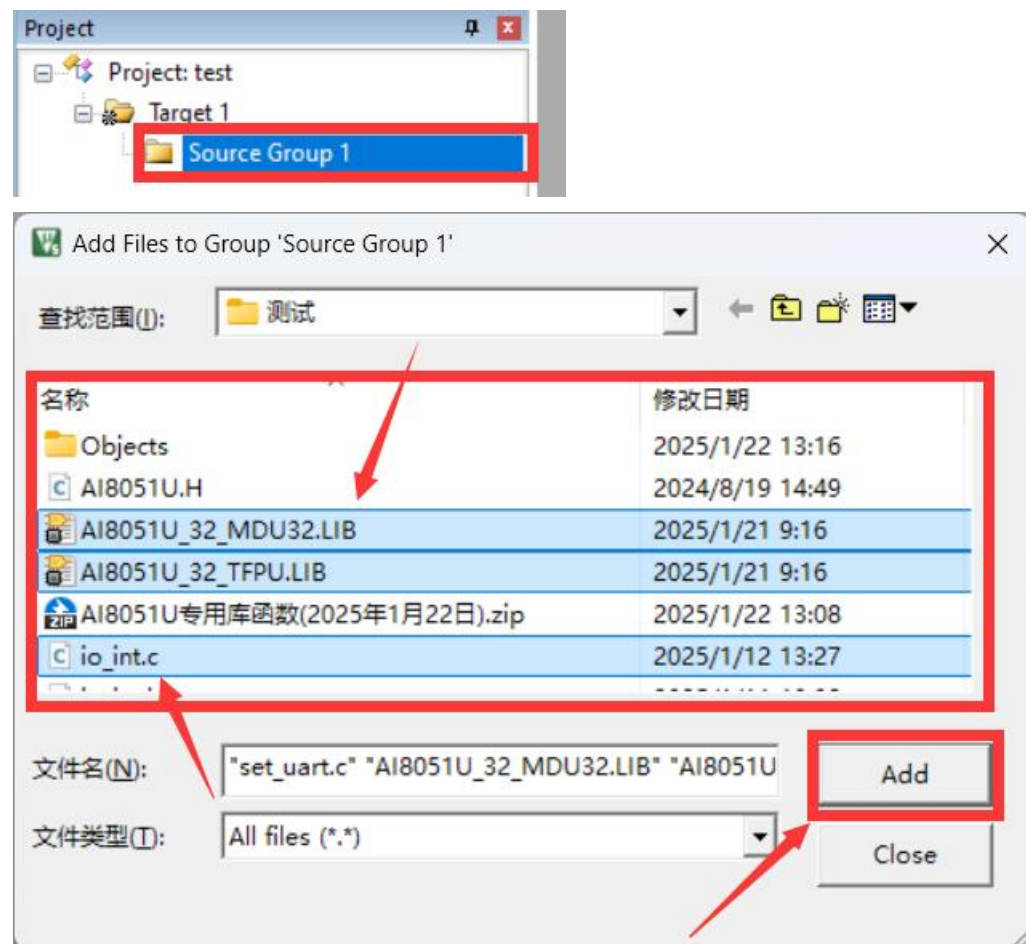
接下来，切换到 Output 选项卡，勾选 'Creat HEX File' 选项，否则编译后是不会创建 HEX 文件的。



3、按需添加工程文件

添加文件

关闭上面的'Options for Target 'Target 1'设置界面，回到主界面，在左侧的 Project 栏中，双击 Source Group 1 文件夹，即可打开 Add Files to Group 'Source Group 1' 界面，在这个界面中选择需要使用的库函数的.c 文件，还有相应的数学库.LIB 文件，点击 Add 按钮即可添加成功。选择多个文件的时候按住键盘的 Ctrl 键，点击需要选择的文件即可（此步需要先复制库函数的.c 和.h 文件到刚才创建的文件夹内，按需赋值）。

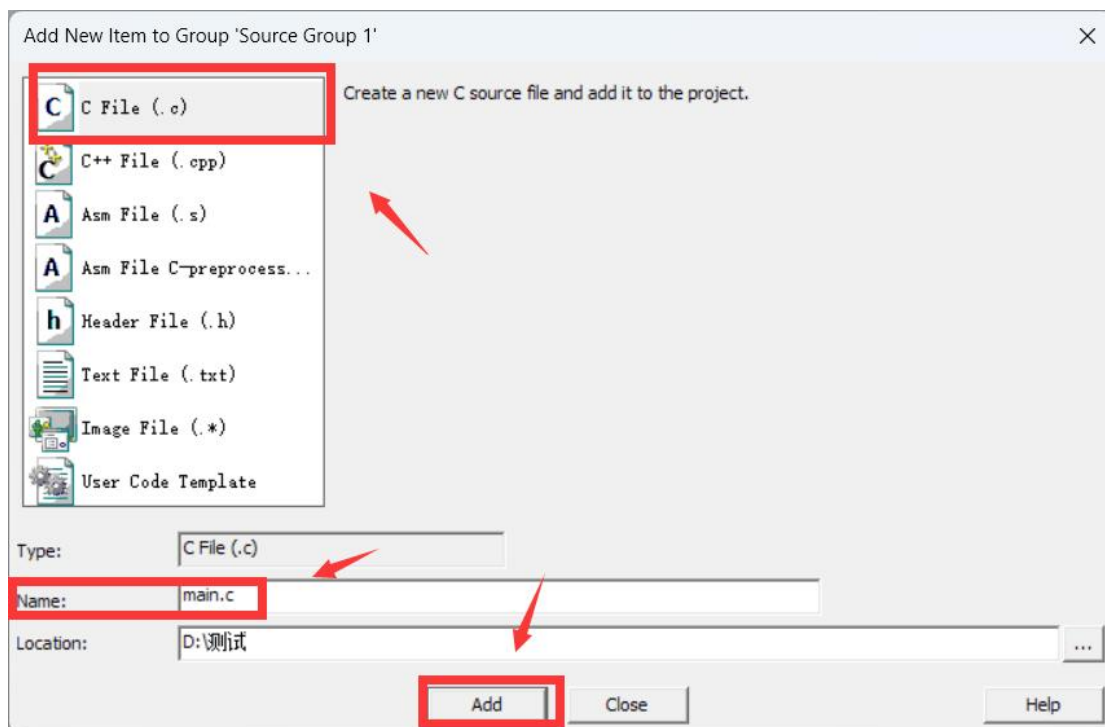
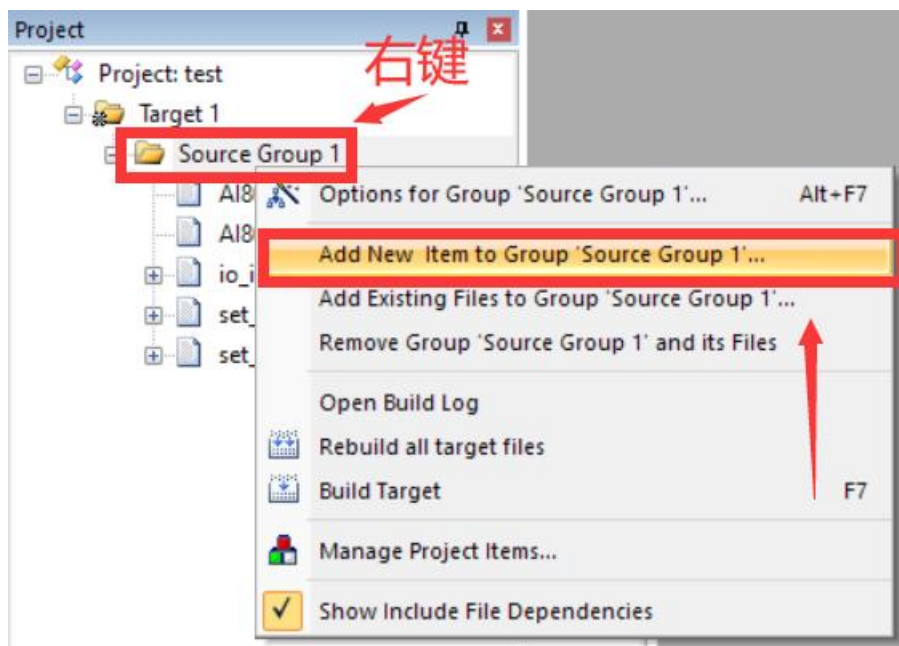


添加 main.c 文件（包含 main 函数的 C 文件）

添加完成函数库文件后，还缺少一个 main 文件来调用这些库函数。

还是在左侧的 Project 栏中，右键 Source Group 1 文件夹，选择 'Add New Item to Group 'Source Group 1'...' 选项。

在弹出来的界面中，选择 C File，并在底下的 Name 栏中填入相应的名字，这里填入'main.c'（可以更改为其他的名字，不是非要 main.c 这个名字的）。



添加头文件定义（可从例程开始复制）

添加好 main.c 后，我们还需要引入对应的头文件定义，这里我们直接复制对应独立例程里面的全部程序过来。

```

1 #include "AI8051U.h"
2 #include "io_int.h" //设置IO独立中断的库函数，依赖于set_io.h
3 #include "set_io.h" //设置IO模式的库函数，可以阅读H文件中的详细说明，或者直接看以下例程的使用
4 #include "set_uart.h" //设置串口的库函数，丰富的使用方法详见H文件中的详细说明
5 #include "stdio.h"
6
7 /*
8 使用说明：
9 按下P32，串口1打印hello world。需要使用USB转串口接在TyepC下载口上，波特率是115200bps
10 使用串口1发送 "cnt:123" (英文符号，引号不包括在内)，应该返回send num:123。
11 即发送什么数字，解析后返回什么数字。
12 */
13
14 char tmp_str[5] = {'a', 'b', 'c', 'd', 'e'};
15 int cnt_dat = 0;
16 void main(void)
17 {
18     EAXFR = 1; // 允许访问扩展寄存器
19     WTST = 0;

```

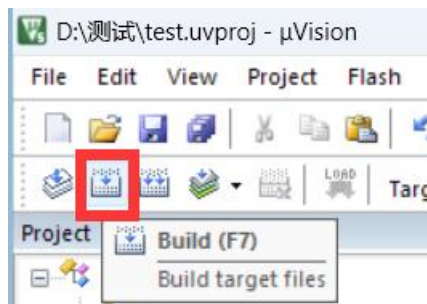
简单修改，编译

我们修改一下例程的程序，原来输出的 Hello World，这里我们改成 Test。

按一下 F7 或者界面左上角的 Build 图标，即可得到编译输出。

看到底部的 Build Output 栏输出 0 Error 0 Warning 代表没有错误

Creating hex file 代表成功创建了 hex 文件。



```

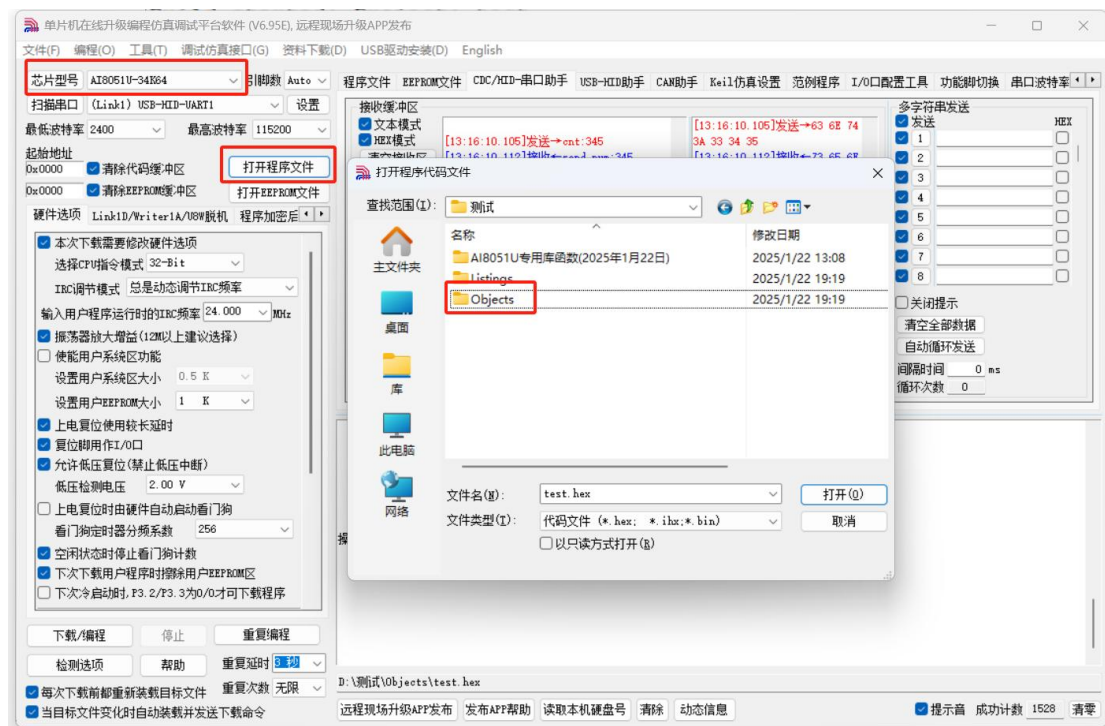
40 if (get_uart_state(Uart1))
41 {
42     // 注意：使用sscanf需要引入stdio.h
43     sscanf(_uart1_rx_buff, "cnt:%d", &cnt_dat); // 缓冲区可以查看set_uart.h中缓冲区的定义
44     // sscanf用法，第一个参数是缓冲区，第二个参数是格式化字符串，第三个参数是变量地址
45     uart_printf(Uart1, "send num:%d\r\n", (int)cnt_dat); // 串口1打印解析到的数据并显示
46 }
47 if (get_ioint_satae(Pin32)) // 按下P32，串口1打印hello world
48 {
49     uart_printf(Uart1, "Test!\r\n");
50     // 普通printf用法，内嵌printf函数，可以通过第一个参数实现打印串口的选择
51     // 本printf自带长度校验和串口忙标志，超过长度会不打印，请到set_uart.h中改变Uart_Tmp_Max
52     // 如果连续调用printf，在第一个printf没有完成发送的情况下，后续的printf会被丢弃
53     // 如果想知道对应的串口发送是否忙，可以使用tx_state[Uart1]这样子来查询（这个是串口1的）
54
55     // uart_printf(Uart1,Hex_Mode,0x0f); //输出0x0f单字节，类似直接给SBUF值
56     // uart_printf(Uart1,Buff_Mode,tmp_str,5); //输出字符串tmp_str，5个字节
57 }
58 }

```

```
Build Output
Build started: Project: test
Build target 'Target 1'
linking...
Program Size: data=81 edata+hdata=406 xdata=486 const=780 code=13273
creating hex file from ".\Objects\test"...
".\Objects\test" - 0 Error(s), 0 Warning(s).
Build time elapsed: 00:00:00
```

下载时的注意事项

下载的时候，需要打开 STC-ISP，选择正确的单片机型号。然后点击‘打开程序文件’，Keil 生成的 HEX 文件默认放在 Objects 文件夹内了（HEX 文件具体生成在哪里，上一步中的 Build Output 输出信息中有给出），双击进入文件夹后，选择相应的 HEX 文件，点击 OK 即可选择。最后点击 STC-ISP 左下角的‘下载/编程’即可下载。



4、反馈渠道

官方网站: <https://www.stcaimcu.com/>

库函数相关反馈渠道:

<https://www.stcaimcu.com/forum.php?mod=viewthread&tid=13629>