

重庆交通大学信息科学与工程学院

《嵌入式系统基础A》课程

作业报告（第三周）

班 级： 物联网工程2001

姓名-学号： 周帅康-632007060321

实验项目名称： ARM汇编程序入门实践

实验项目性质： 设计性

实验所属课程： 《嵌入式系统基础》

实验室(中心)： 南岸校区语音大楼

指导教师： 姜路

完成时间： 2022 年 10 月 2 日

一、实验内容和任务

一、搭建并配置Keil嵌入式开发环境，完成一个基于STM32汇编程序的编写。 1) 记录build生成的 hex文件各段的大小，了解Hex文件格式及其前8个字节内容含义； 2) 学习在没有硬件条件下进行仿真调试的方法，观察ARM寄存器变化状况。

二、实验要求

1. 分组要求：每个学生独立完成，即1人1组。
2. 程序及报告文档要求：具有较好的可读性，如叙述准确、标注明确、截图清晰等。
3. 项目代码上传github，同时把项目完整打包为zip文件，与实验报告（Markdown源码及PDF文件）、作业博客地址一起提交到学习通。

三. 实验过程介绍（此处可以填博客内容）

<下面是一个参考范例，需要换成你对应的作业博客markdown内容。 生成报告时删除此句>

@[toc]

一个简单stm32程序编写以及调试

一.环境配置

1.MDK的安装

MDK (Microcontroller Development Kit) 是针对ARM处理器，特别是Cortex-M内核处理器的最佳开发工具

① 软件包的下载

首先需要下载安装mdk5软件和stm32包

- ① MDK安装包
- ② stm32相关包
- ③ MDK破解软件

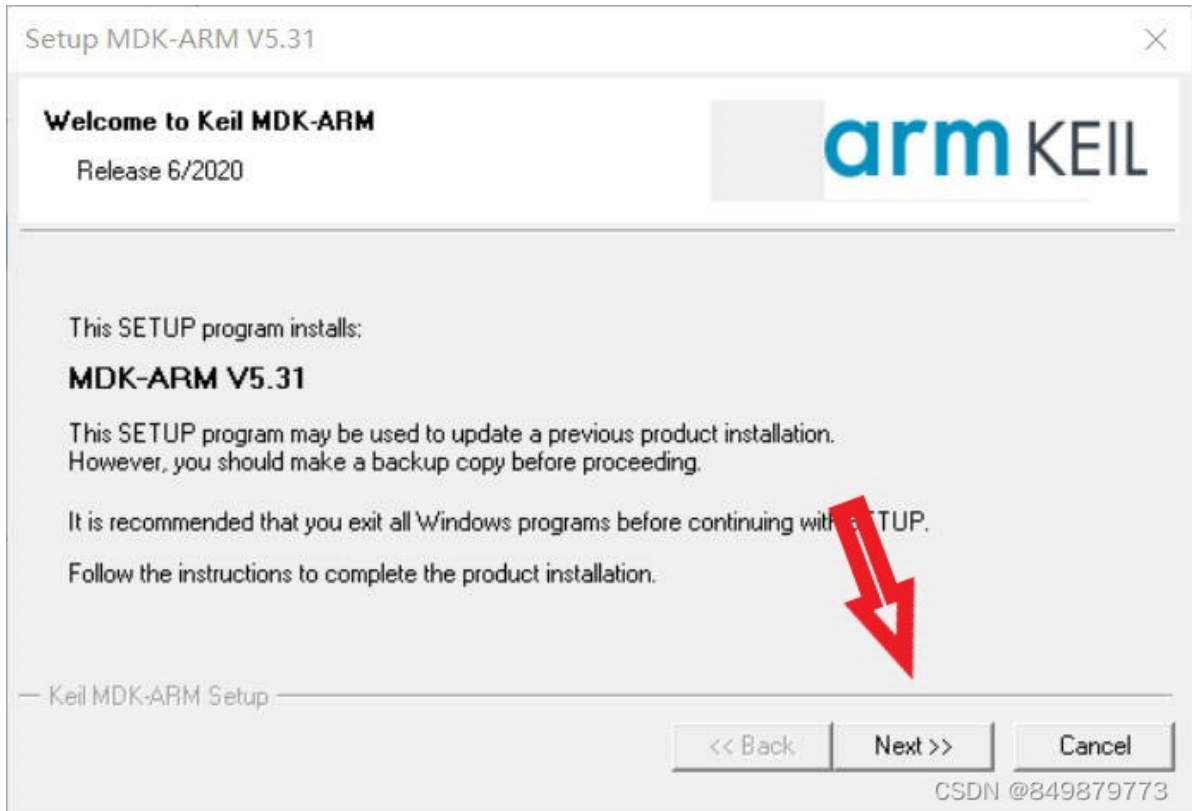
链接: <https://pan.baidu.com/s/123ET4Ch8t6GcpNLMosfHMQ>

提取码: 1234

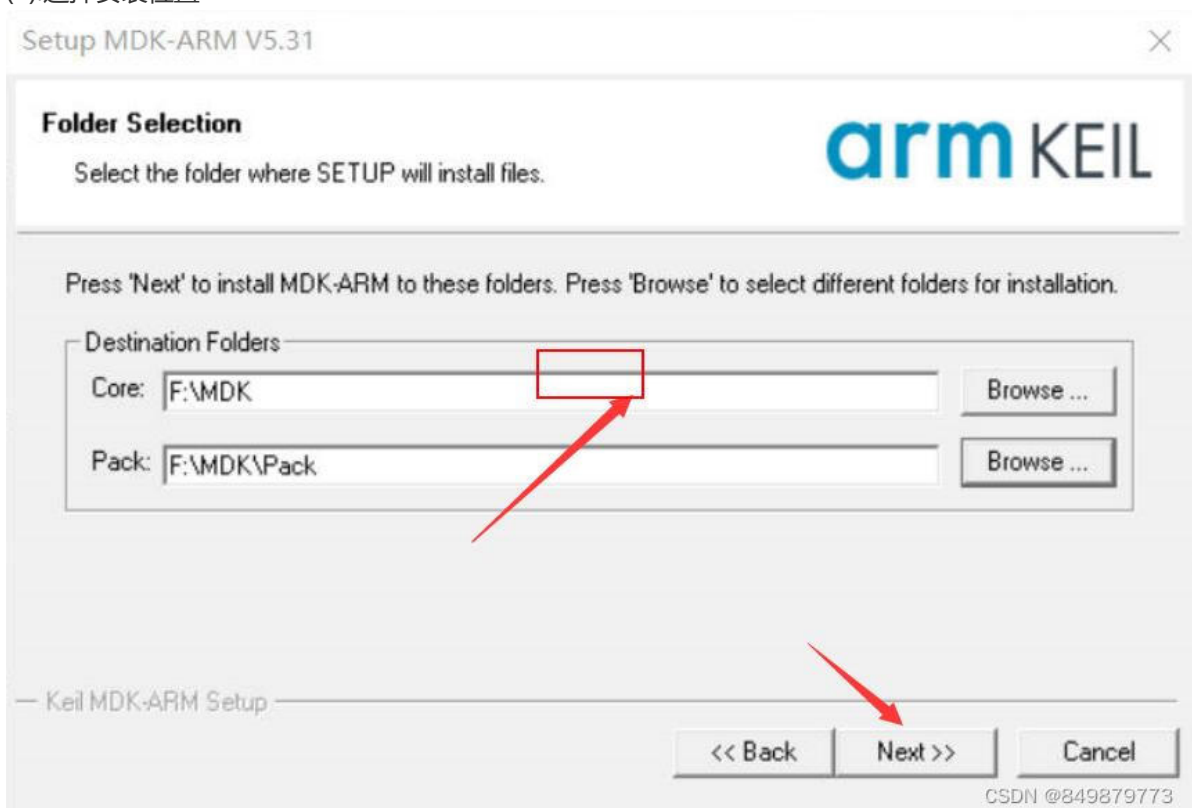
下载压缩包解压后进行后面的步骤

② MDK的安装

(1).打开mdk531.exe文件，点击下一步



(2).选择安装位置



(3).填写注册信息

Setup MDK-ARM V5.31

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:

填好后点击next

<< Back Next >> Cancel

Keil MDK-ARM Setup

CSDN @849879773

Setup MDK-ARM V5.31

Setup Status

MDK-ARM Setup is performing the requested operations.

Install Files ...

Installing cpp_psfnu.b.

■■■■■

<< Back Next >> Cancel

Keil MDK-ARM Setup

CSDN @849879773

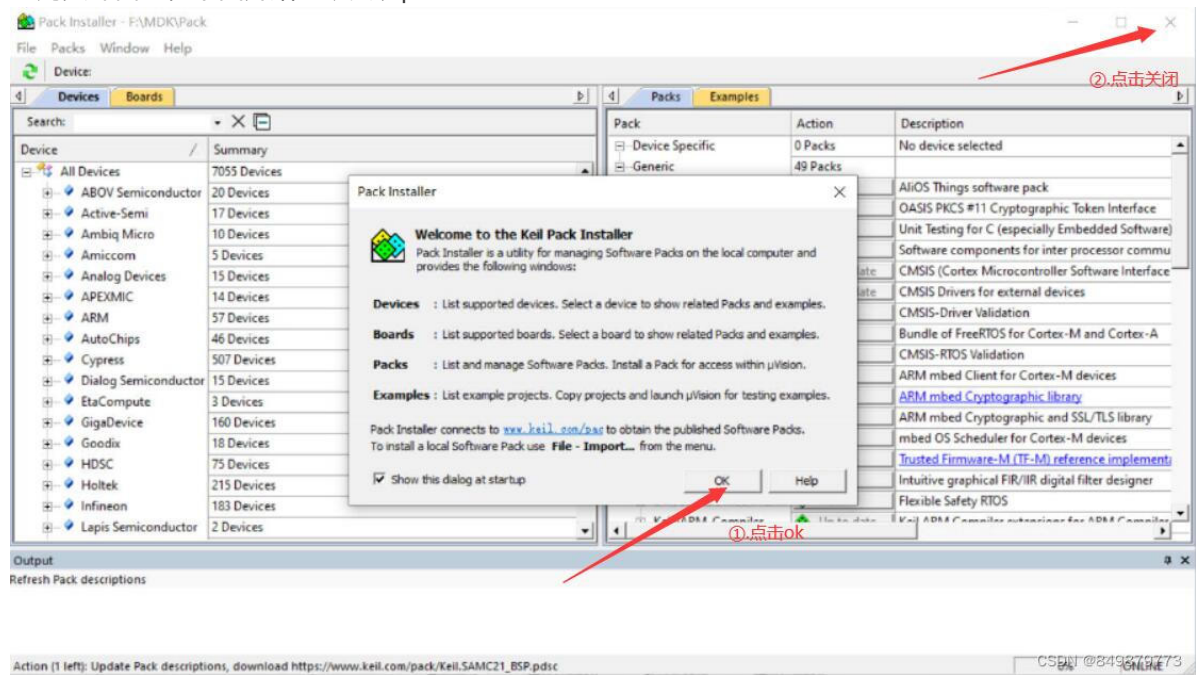
(4).点击安装



(5).完成安装

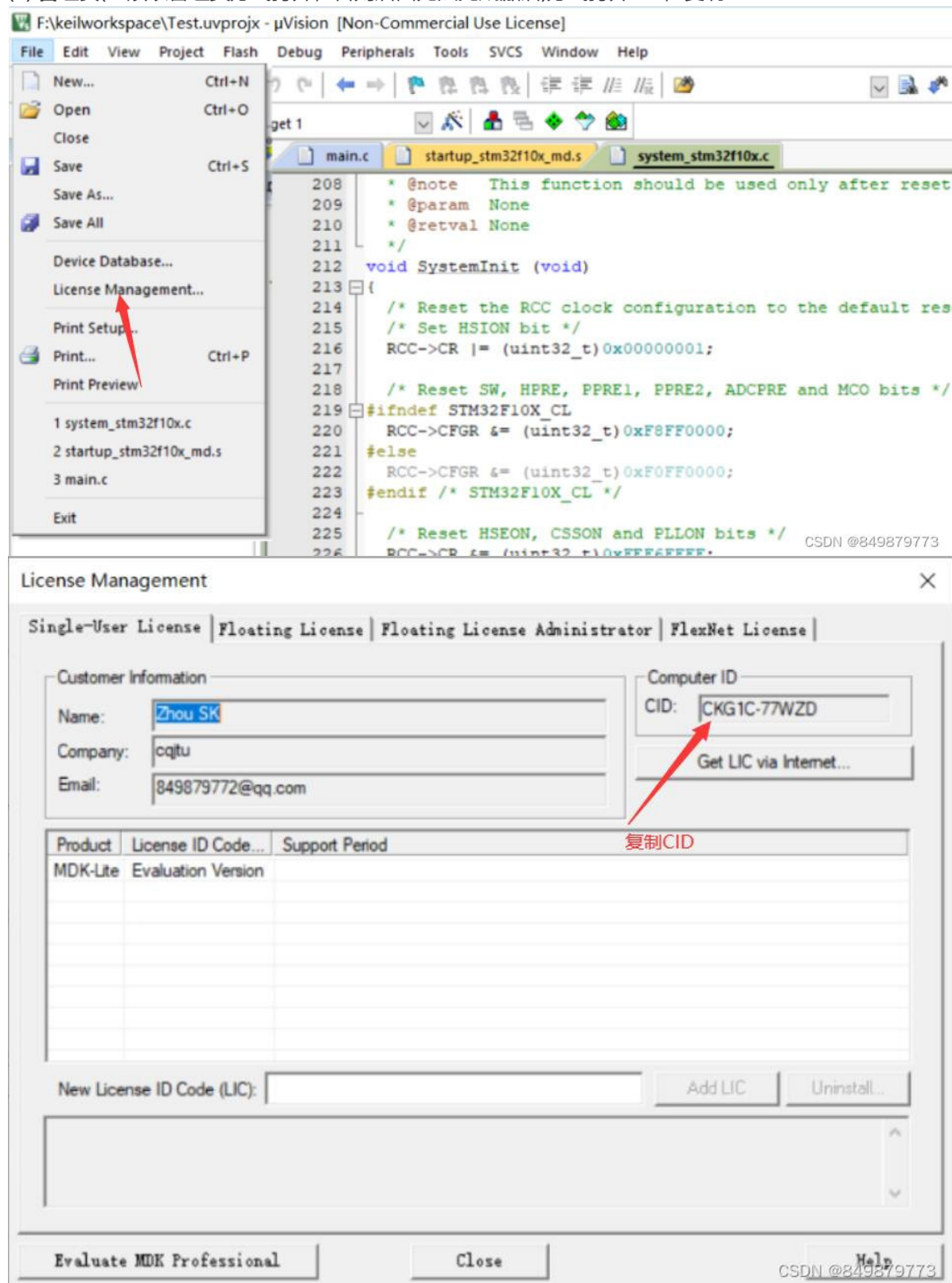


(6).点击OK后，鼠标会变成转圈圈的，因为正在进行在线安装各种pack，但会安装失败，不用着急，右上角关掉窗口，下面开始手动安装pack包。

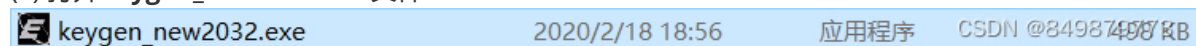


2. MDK的破解

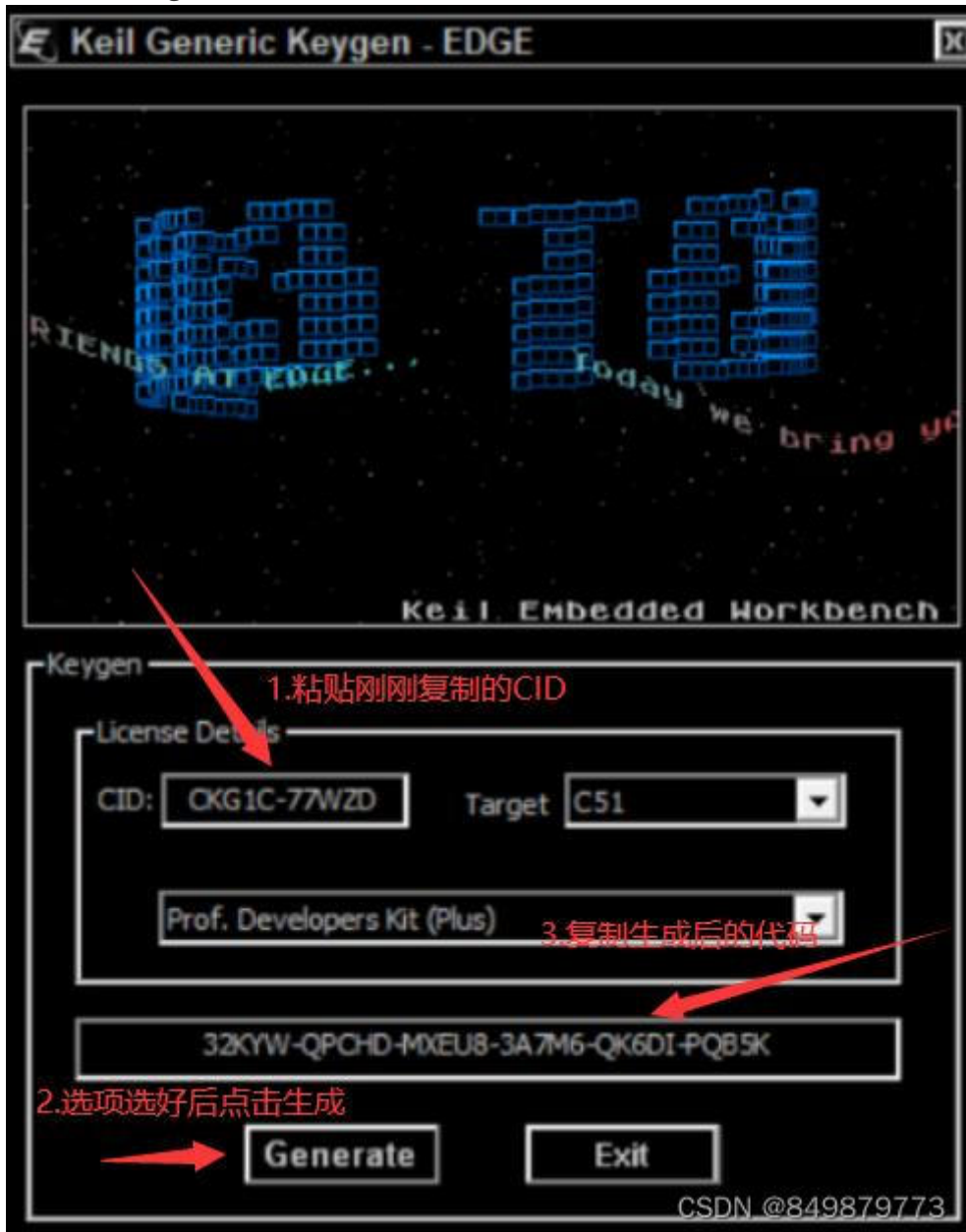
(1).管理员(必须以管理员方式打开, 否则后面无法完成激活)方式打开keil, 复制cid



(2).打开keygen_new2032.exe文件



(3).打开第一步辅助的CID粘贴到对应的位置，
按图上选择target和版本后点击Generate生成激活码



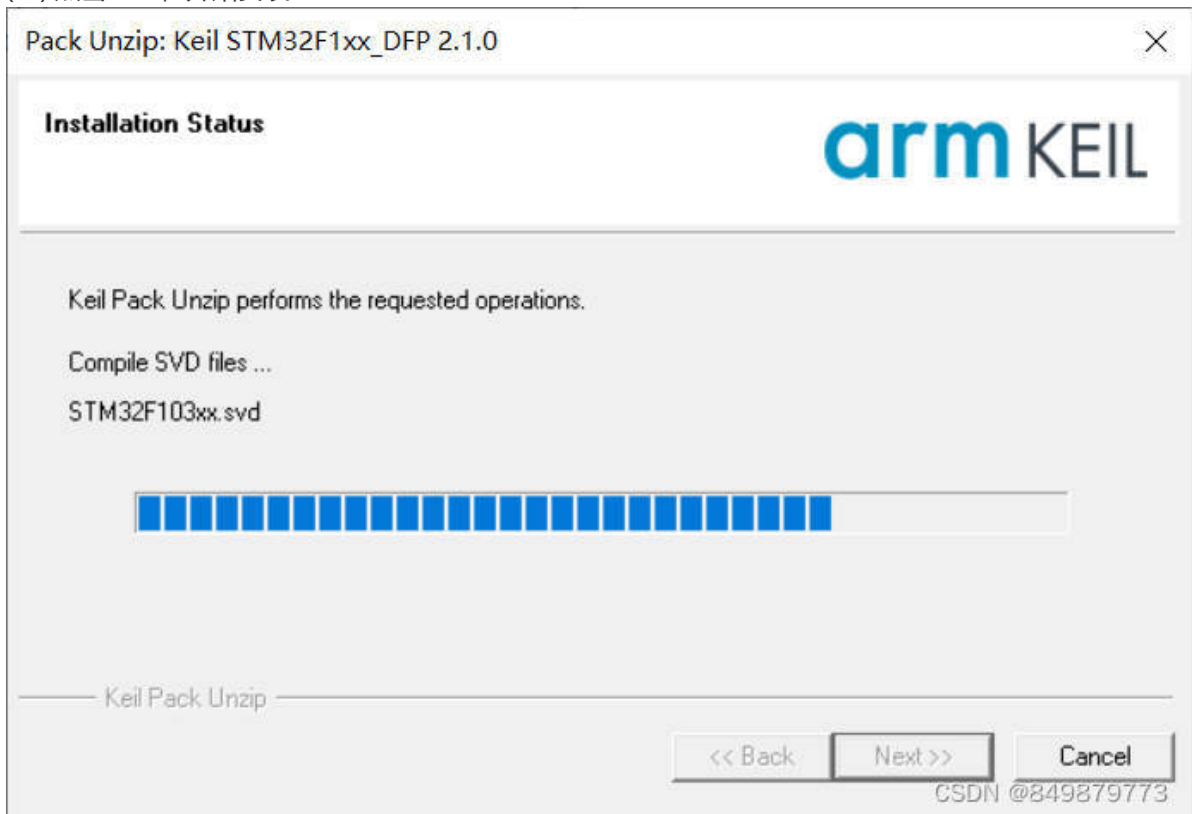
(4).回到keil，把激活码粘贴上去，破解成功

3.stm32 pack的安装

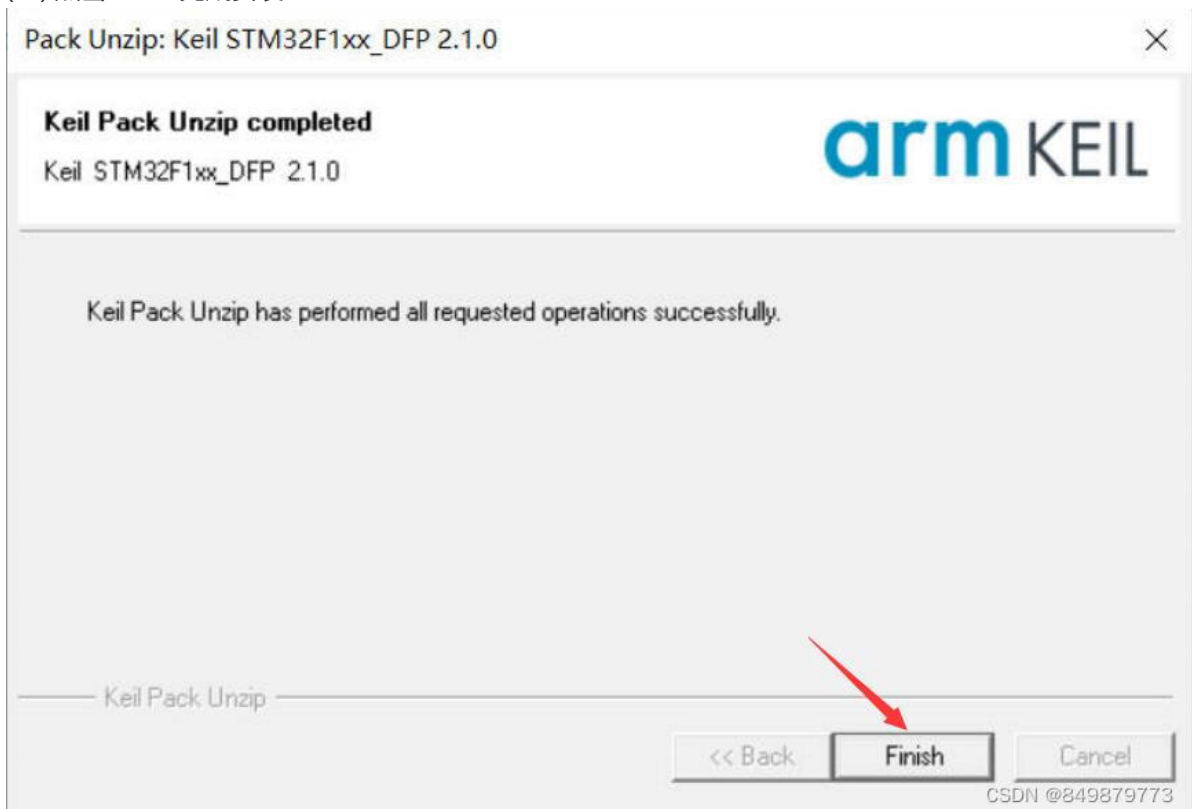
(一).打开Keil.STM32F1xx_DFP.2.1.0.pack文件



(二).点击next, 开始安装

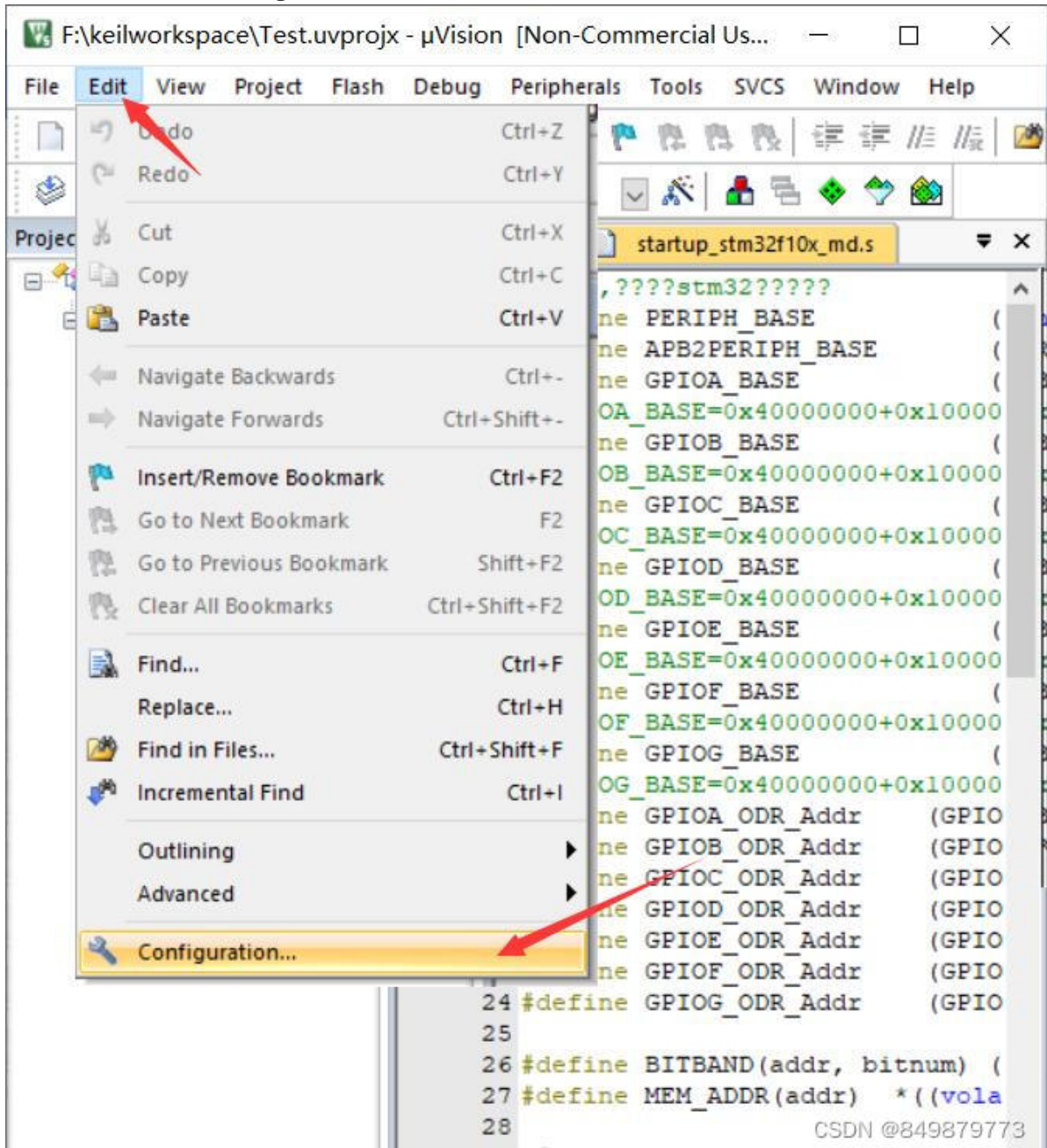


(三).点击finish完成安装

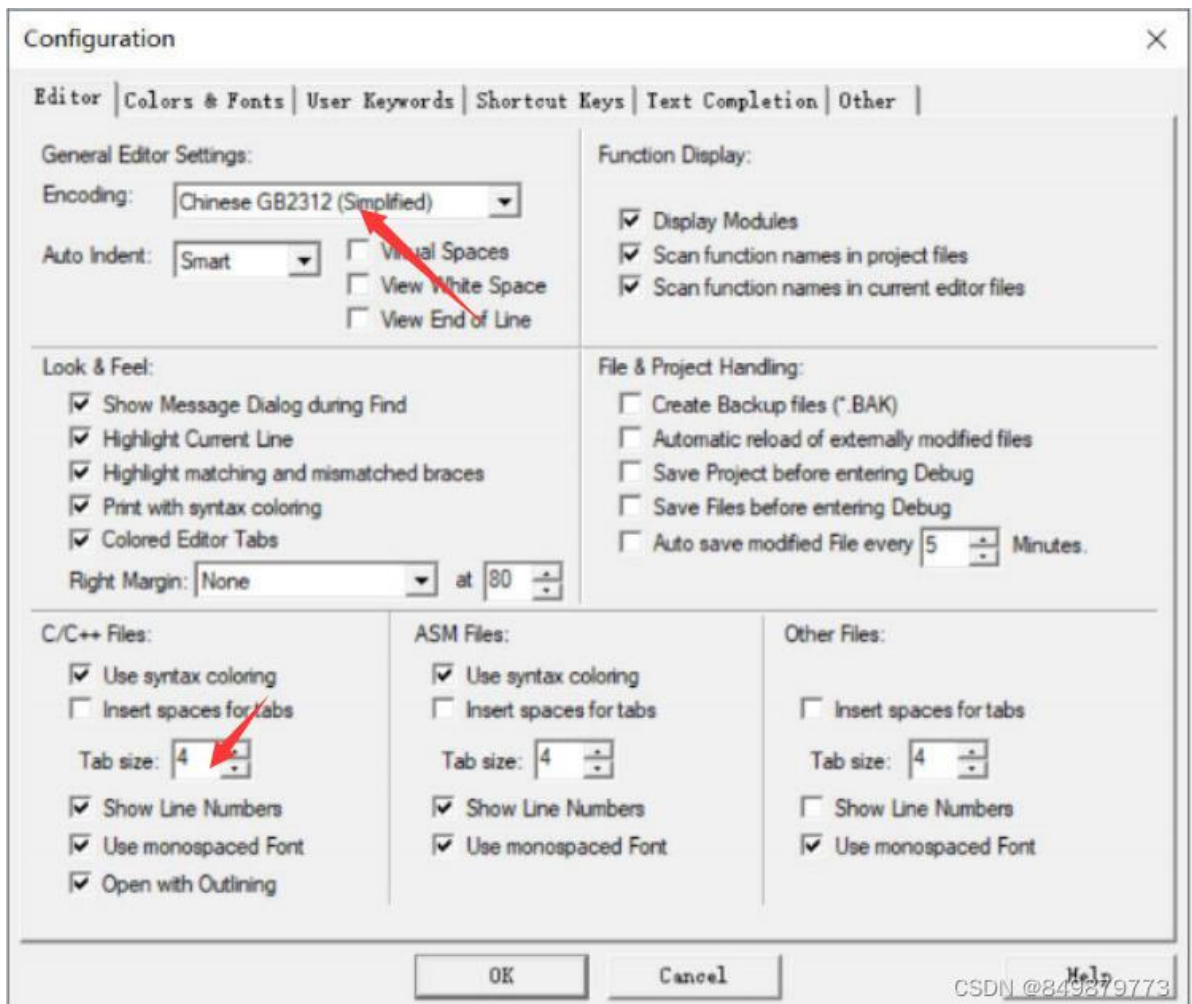


4.keil的简单设置

(1) 首先点击Edit→Configuration进入设置界面。



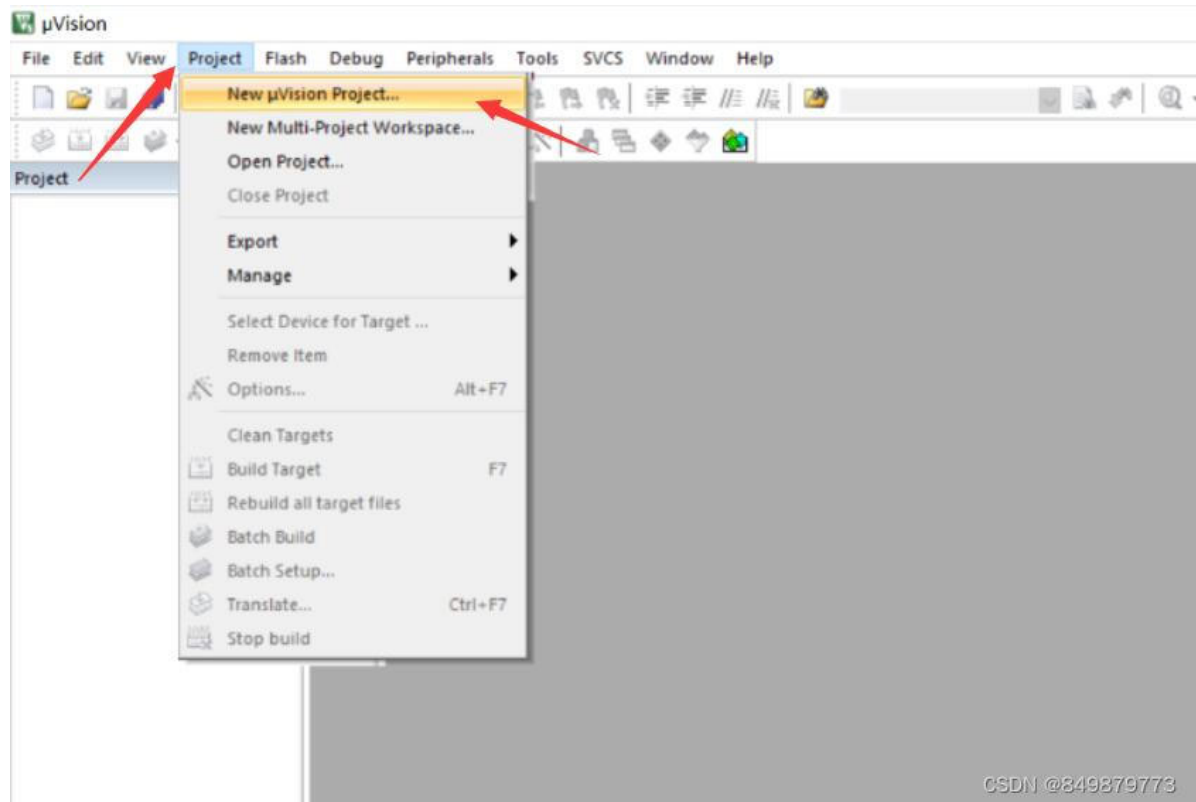
(2) 设置编码形式为ChineseGB2312(Simplified)如果不设置，你从其它地方粘贴过来的代码含有中文的话，就会出现乱码，然后设置Tab size为4。



二.一个stm32简单程序的编译

1.新建一个工程

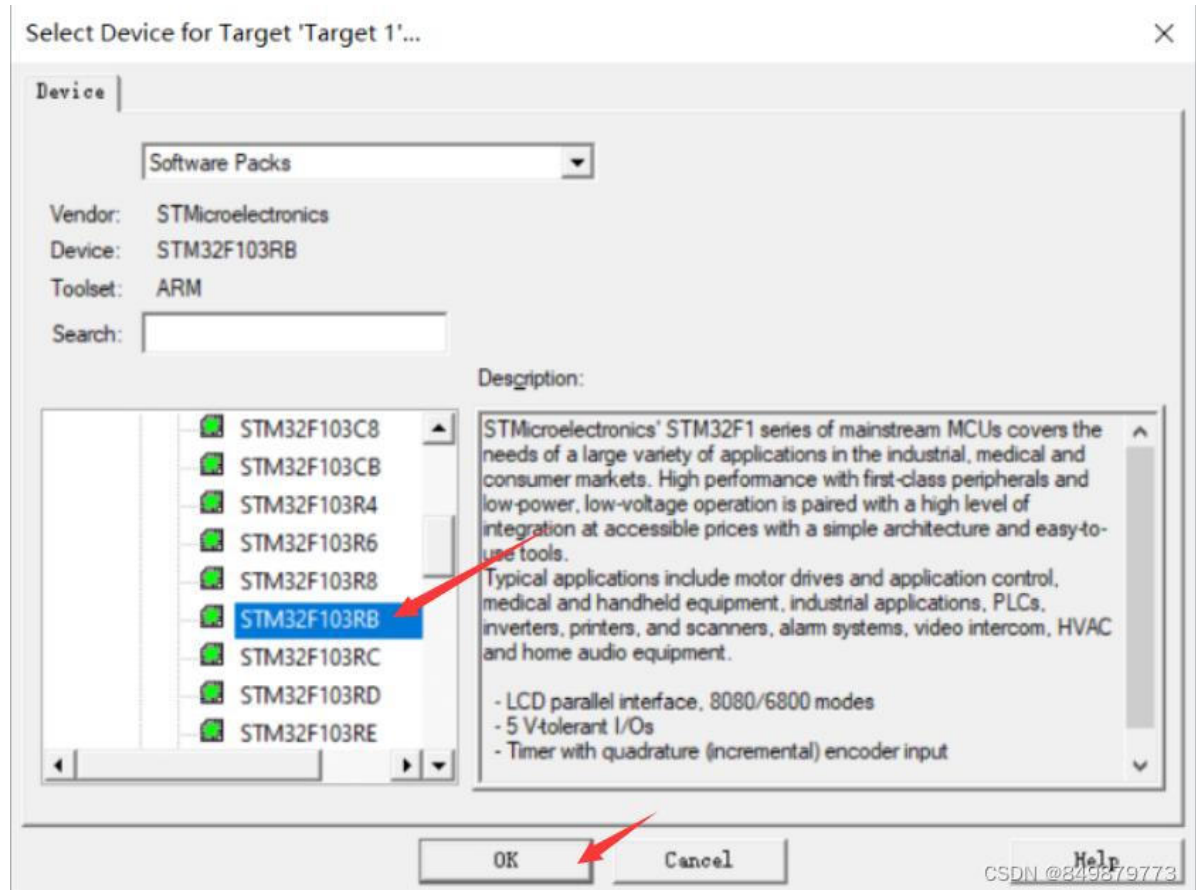
(1)打开keil uVision5,并新建一个工程



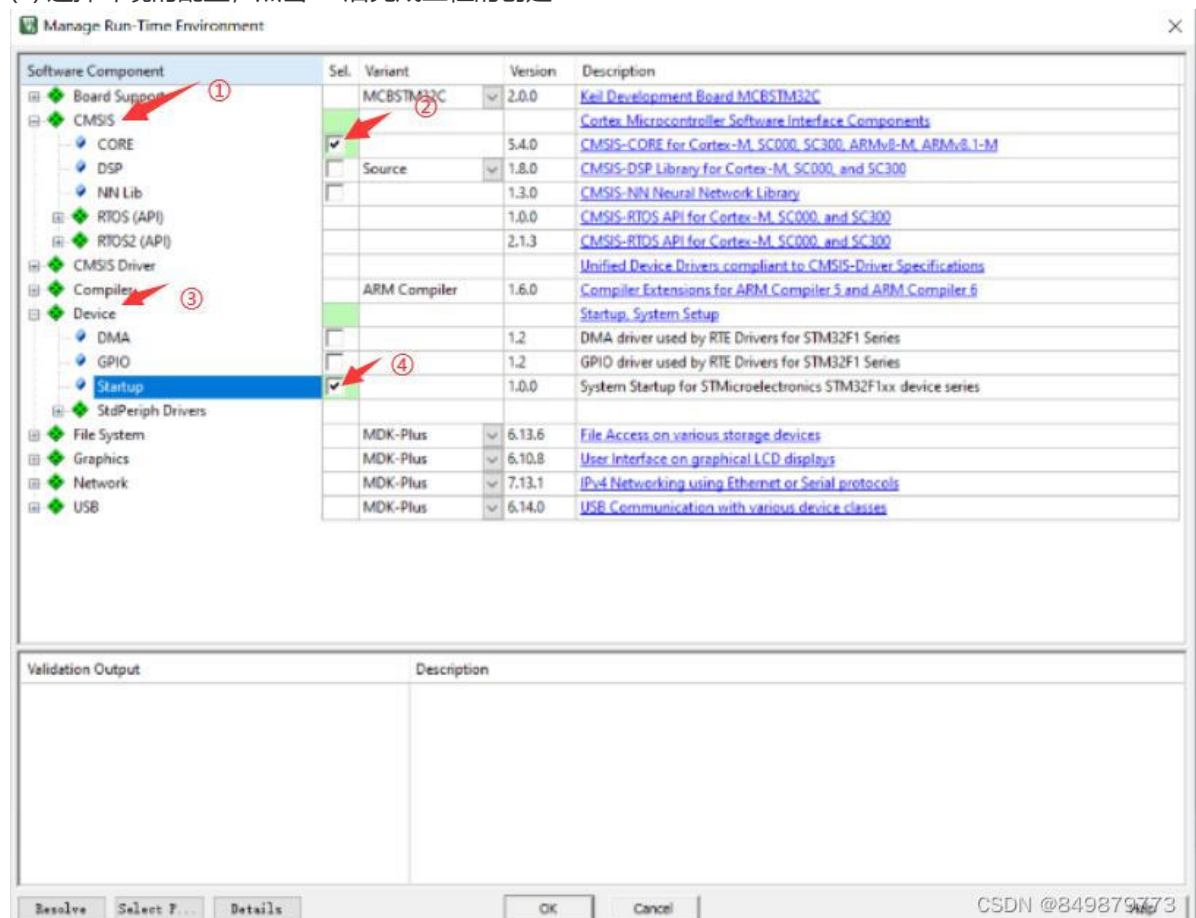
(2).建立一个名为Test的工程



(3).选择STM32F103C8芯片，点击OK

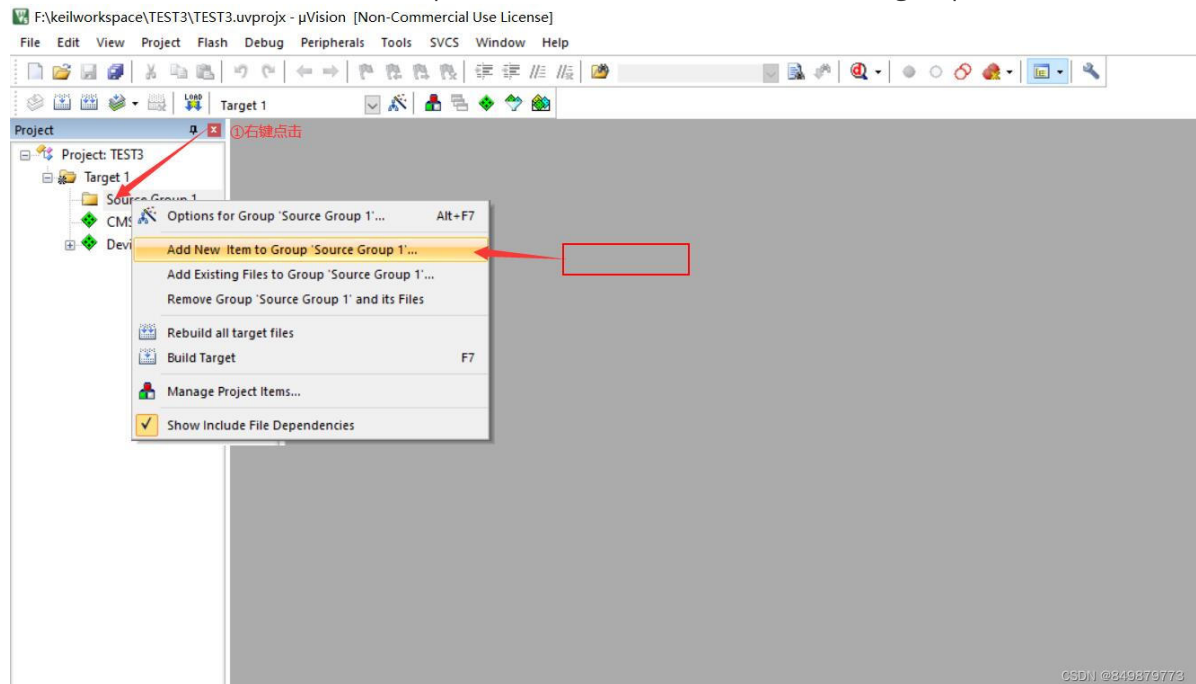


(4).选择环境的配置，点击OK后完成工程的创建

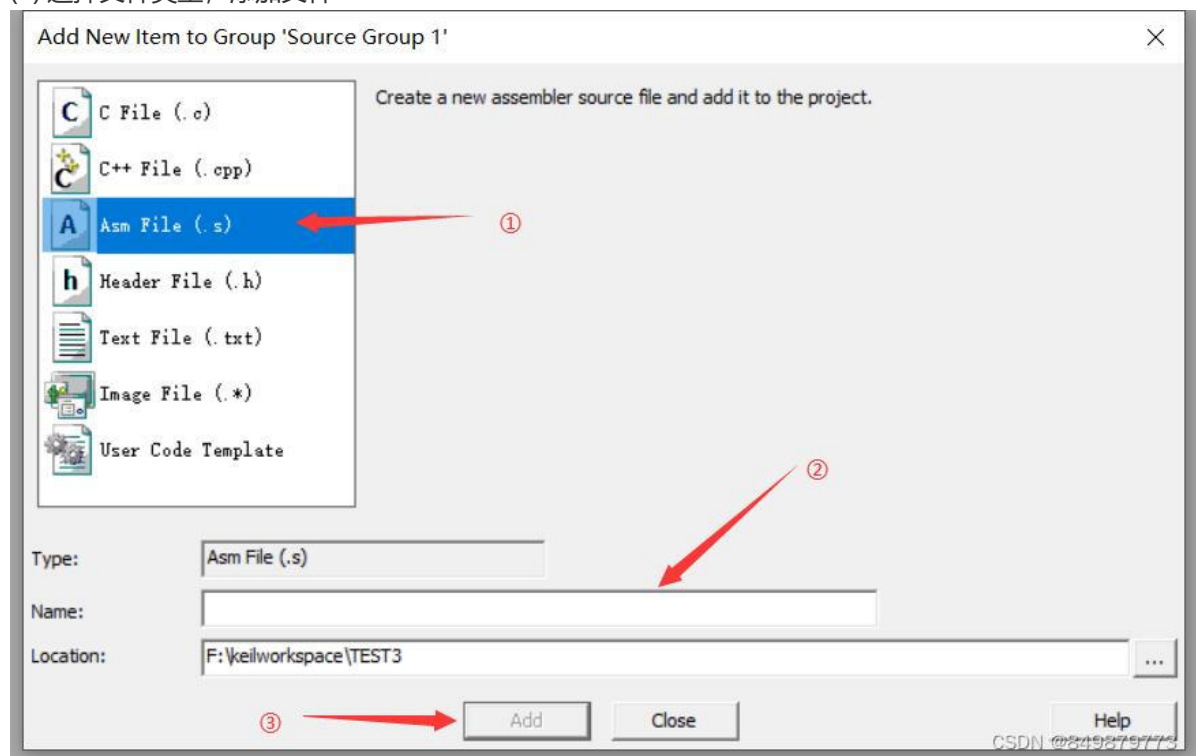


2.新建一个.s文件

(1) 工程创建完毕后，对Source Group文件点击右键再点击ADD new item to group



(2) 选择文件类型，添加文件



(3).键入代码

```
AREA MYDATA, DATA

AREA MYCODE, CODE
ENTRY
EXPORT __main

__main
```



```

MOV R0, #10
MOV R1, #11
MOV R2, #12
MOV R3, #13
;LDR R0, =func01

```

```

BL func01
;LDR R1, =func02
BL func02

```

```

BL func03
LDR LR, =func01
LDR PC, =func03
B .

```

```

func01
MOV R5, #05
BX LR

```

```

func02
MOV R6, #06
BX LR

```

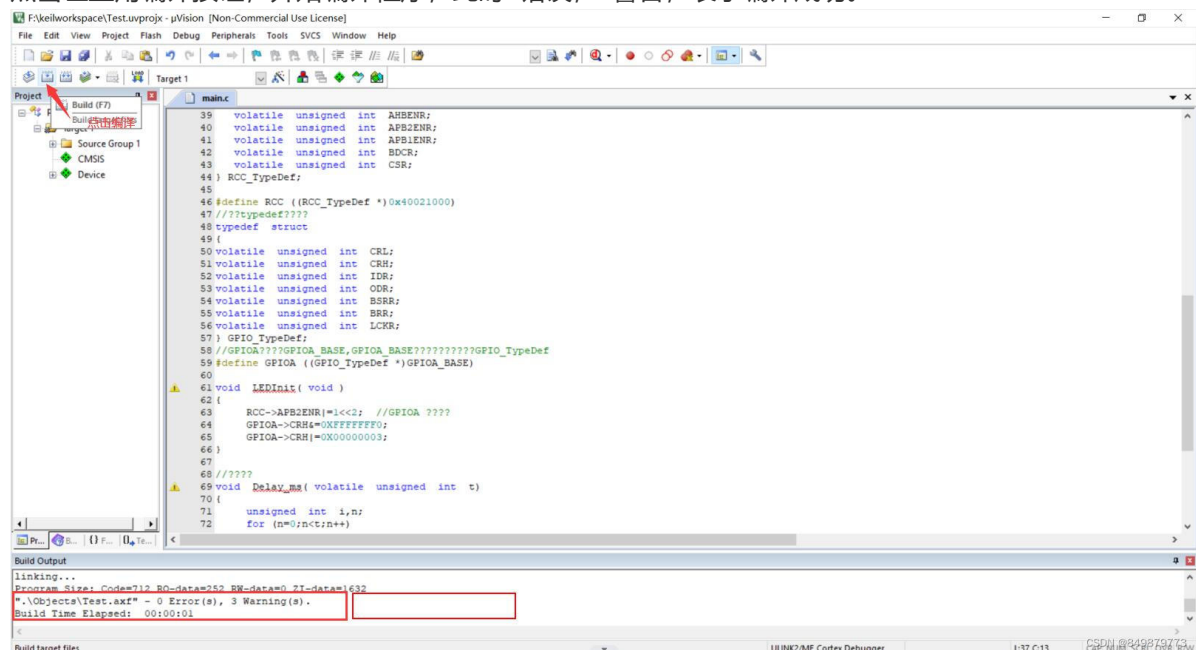
```

func03
MOV R7, #07
MOV R8, #08
BX LR

```

3.编译

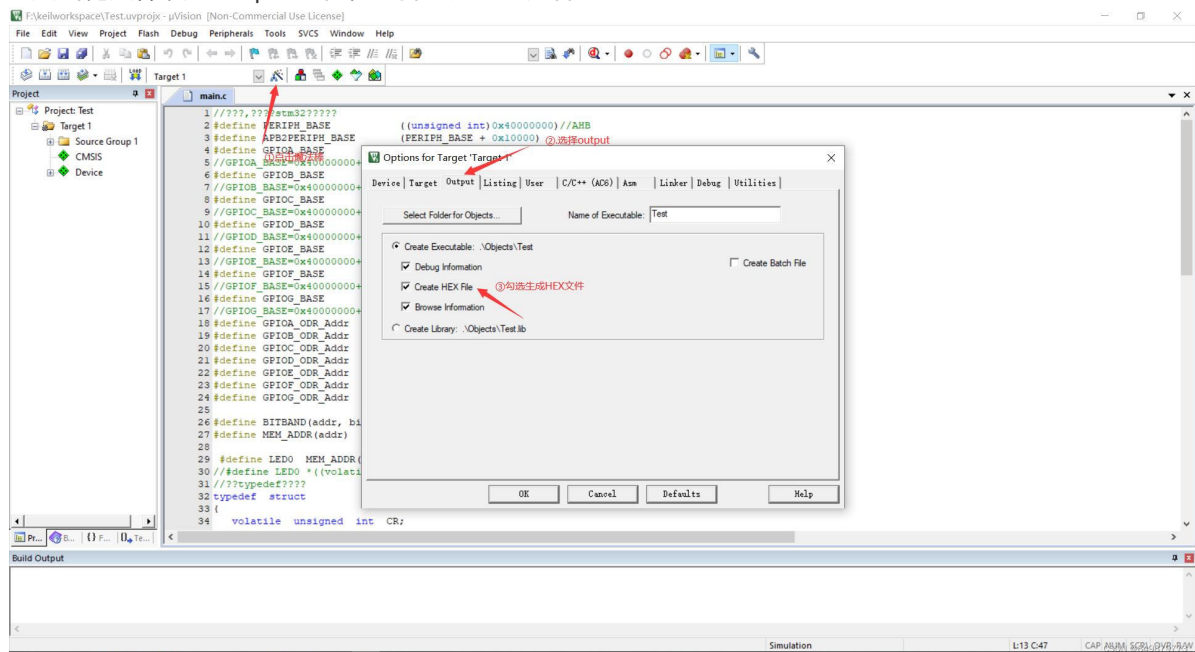
点击左上角编译按钮，开始编译程序，此时0错误，0警告，表示编译成功。



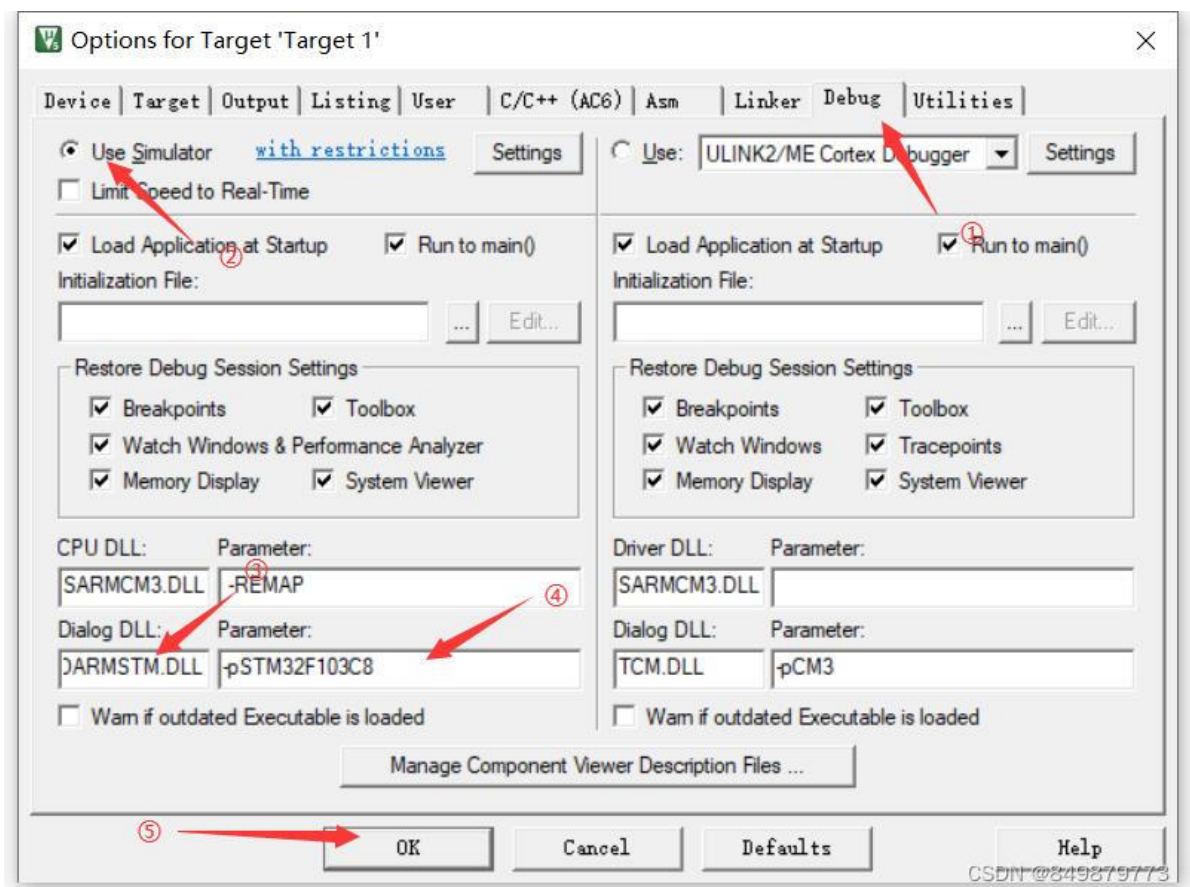
4.stm32程序仿真调试

(一)调试设置

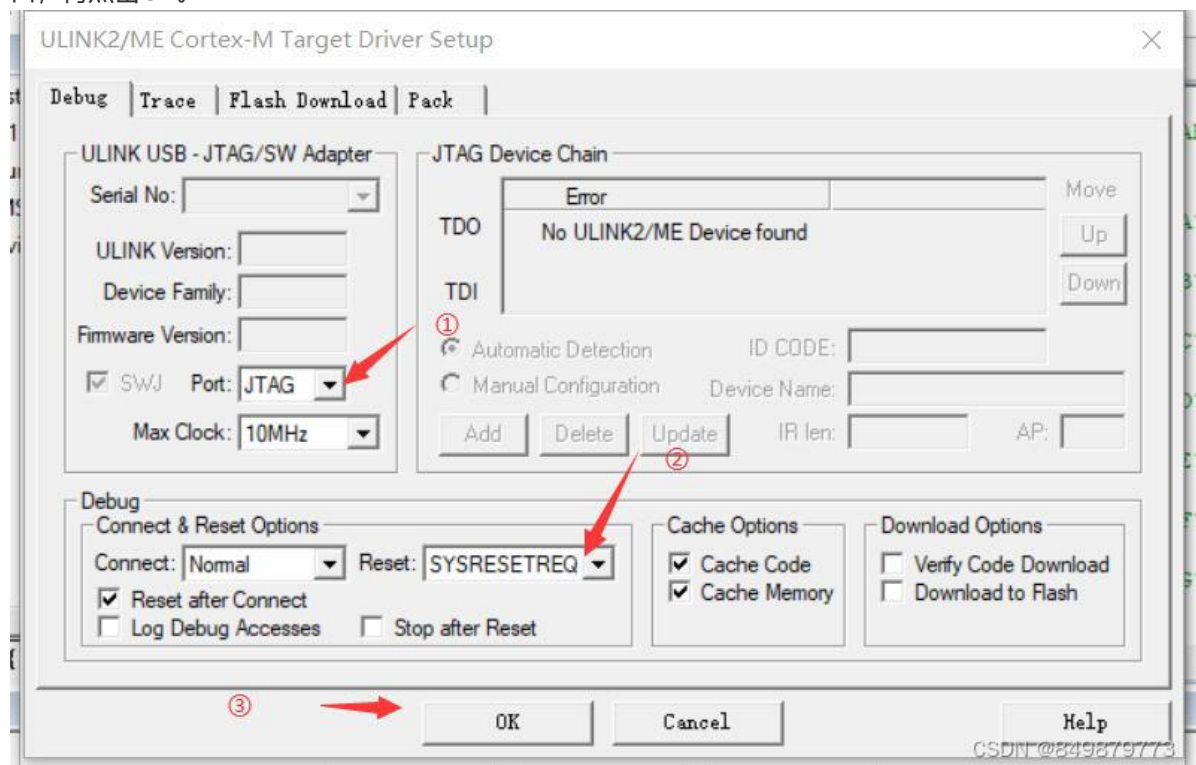
1.点击魔法棒设置output一栏，选择生成HEX文件



2.点击 Debug 界面。选择 Use Simulator，代表选择仿真器进行调试；同时勾选Run to main()防止程序不运行主函数而进入死循环（默认情况是勾选了的）在下方的 Dialog DLL 一栏中覆写 DARMSTM.DLL，在一旁的Parameter中覆写-pSTM32F103C8，用于设置支持STM32F103C8的软硬件仿真。

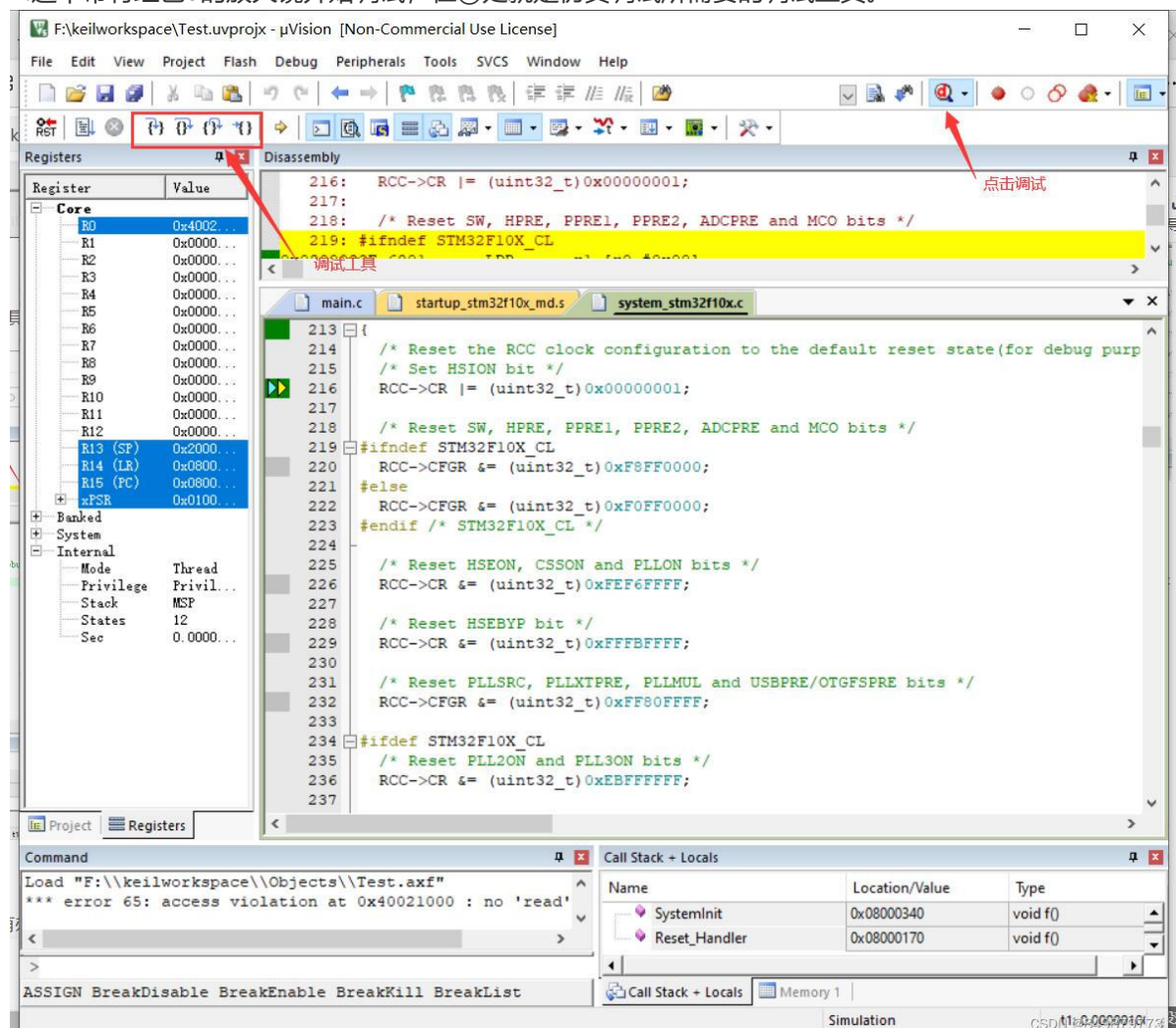


3.确定一下Port是JTAG，Reset可以设置为Autodetect或SYSRESETREQ，然后点击OK返回上一级窗口，再点击OK。

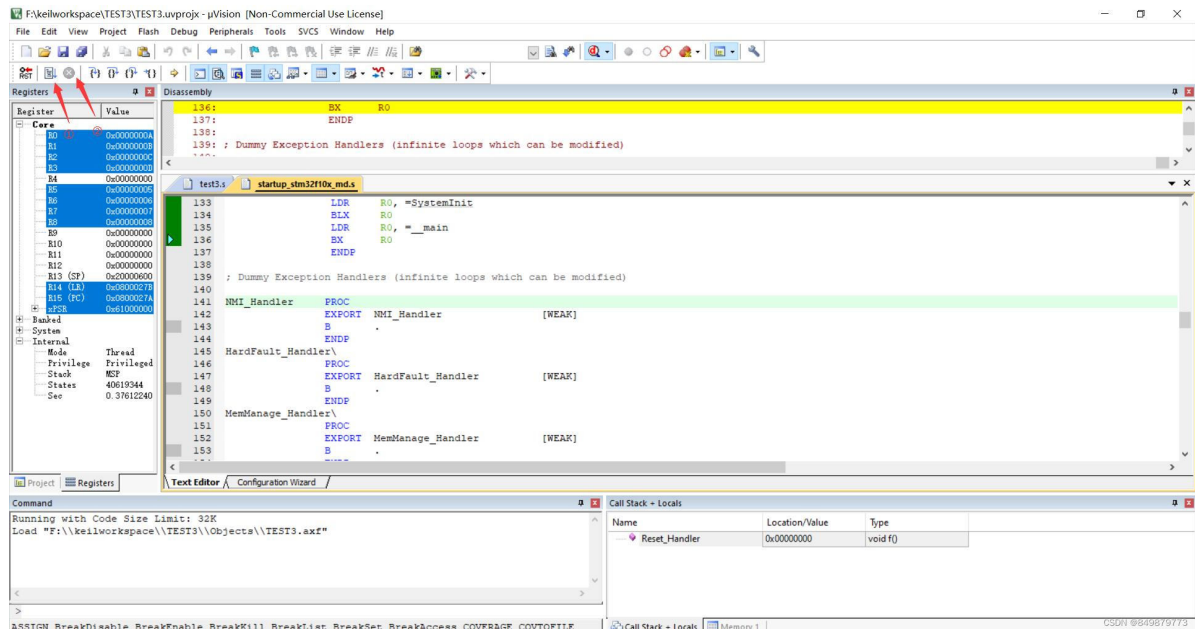


(二)开始调试

1.选中带有红色d的放大镜开始调试，在②处就是仿真调试所需要的调试工具。



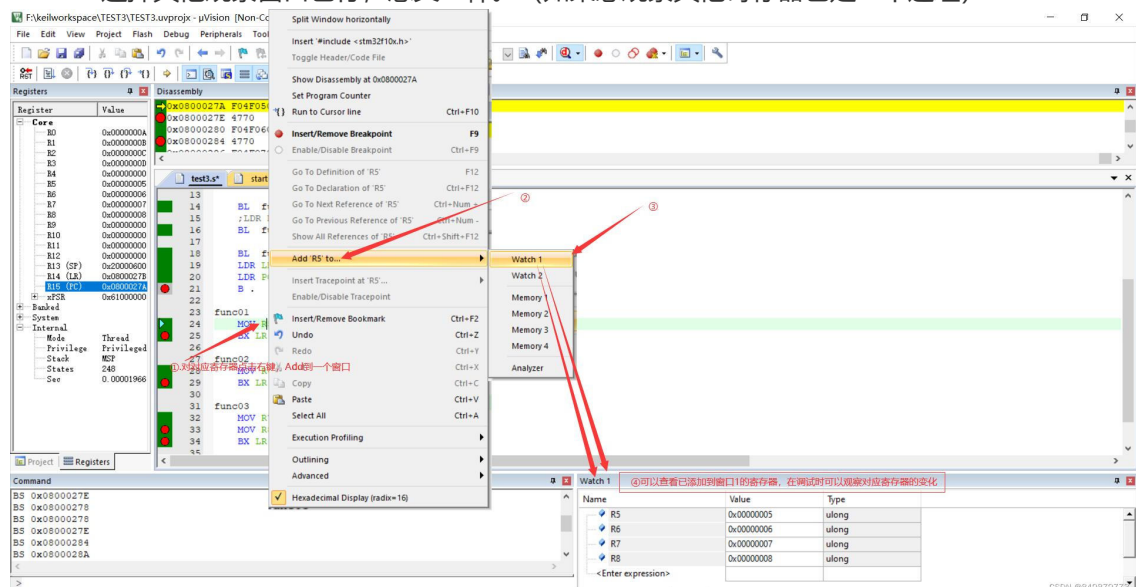
2. 调试运行



运行结果：结果符合预期，寄存器 R5, R6, R7, R8 的值和程序设置一致。

3. 单步调试

- 1. 进入调试状态后，转到TEST.s文件，分别右键R5、R6、R7、R8->Add R to Watch，不选择Watch1选择其他观察窗口也行，意义一样。（如果想观察其他寄存器也是一个道理）



- 2. 在内容变化之前打断点，在右下角窗口观察之前显示的寄存器的变化



Watch 1		
Name	Value	Type
R5	0x00000005	ulong
R6	0x00000006	ulong
R7	0x00000007	ulong
R8	0x00000008	ulong
<Enter expression>		

CSDN @849879773

三.分析Hex文件

hex文件在项目目录中的object里面，以项目名.hex命名

1.文件内容:

```
:020000040800F2
:1000000000060020ED000008F5000008F7000008D9
:10001000F9000008FB000008FD00000800000000D7
:100020000000000000000000000000FF000008C9
:10003000010100080000000003010008050100089C
:100040000701000807010008070100080701000870
:100050000701000807010008070100080701000860
:100060000701000807010008070100080701000850
:100070000701000807010008070100080701000840
:100080000701000807010008070100080701000830
:100090000701000807010008070100080701000820
:1000A0000701000807010008070100080701000810
:1000B0000701000807010008070100080701000800
:1000C00007010008070100080701000807010008F0
:1000D00007010008070100080701000807010008E0
:1000E00007010008070100080701000809488047C8
:1000F00009480047FEE7FEE7FEE7FEE7FEE7FEE70A
:10010000FEE7FEE7FEE7FEE704480549054A064B21
:1001100070470000FD0100085502000800000020A3
:1001200000060020000200200002002070477047F7
:100130007047000080B500F001F880BD82B041F248
:1001400004000021C4F202000191009150F8041C47
:1001500041F4803140F8041C50F8041C01F40031D3
:100160000091019901310191009919B90199B1F5F5
:10017000A06FF1D150F8041C890354BF0021012164
:1001800000910099012936D142F20001C4F2020126
:100190000A6842F010020A600A6822F003020A604C
:1001A0000A6842F002020A600168016001680160A9
:1001B000016841F480610160016821F47C110160F3
:1001C000016841F4E811016050F8041C41F08071AD
:1001D00040F8041C50F8041C8901FBD5016821F08B
:1001E00003010160016841F002010160016801F052
:1001F0000C010829FAD102B07047000080B541F225
:100200000000C4F202000168002241F00101016017
:100210004168CFF6FF021140416001684FF6FF725E
:10022000CFF6F66211400160016821F4802101607F
:10023000416821F4FE0141604FF41F018160FFF726
:1002400079FF4EF60850CEF200004FF000610160D9
:1002500080BD00004FF00A004FF00B014FF00C0280
:100260004FF00D0300F009F800F00AF800F00BF869
:10027000DFF81CE0DFF81CF0FEE74FF005057047E3
:100280004FF0060670474FF007074FF00808704719
:080290007B0200088702000850
```

:04000005080000ED02
:00000001FF

2.文件大小

Program Size: Code=428 RO-data=236 RW-data=0 ZI-data=1536 CSDN @849879773

3.内容分析：扩展线性地址记录:020000040800F2

- 扩展线性地址记录（hex 文件的第一排十六进制）也叫作 32 位地址记录或 HEX386 记录
- 这些记录包含数据地址的高 16 位
扩展线性地址记录总是有两个数据字节，外观如下（这里我通过标记方便对应原始数据）
- 当一个扩展线性地址记录被读取，存储于数据域的扩展线性地址被保存，它被应用于从 Intel HEX 文件读取来的随后的记录
- 线性地址保持有效，直到它被另外一个扩展地址记录所改变
- 通过把记录当中的地址域与被移位的来自扩展线性地址记录的地址数据相加获得数据记录的绝对存储器地址

内容	描述
:020000040800F2	扩展线性地址记录
02	这个记录当中数据字节的数量
0000	地址域，对于扩展线性地址记录,这个域总是 0000
04	记录类型 04 (扩展线性地址记录)
0800	是地址的高 16 位
F2	是这个记录的校验和，计算方法：01h + NOT(02h + 00h + 00h + 04h + 08h + 00h)

4.内容分析：数据分析

Intel HEX文件由任意数量以回车换行符结束的数据记录组成，比如第一行数据记录：

:10000000600600206D0100087501000877010008F6

内容	描述
10	记录当中数据字节的数量
0000	数据将被下载到存储器当中的地址
00	记录类型(数据记录)
600600206D0100087501000877010008	数据
F6	是这个记录的校验和

5.内容分析：文件尾

在文件的最后一排，是一个文件的结束标志（END OF FILE RECORD）：
:00000001FF

内容	描述
00	记录的长度为 0
0000	LOAD OFFSET为0000
01	TYPE = 01
FF	校验和为FF

四.总结

对于完全没接触过stm32实物的时候，模拟仿真调试是一个很好的方法来熟悉或者掌握有关的流程。对于keil的调试来说，只能一步一步的进行调试过程，需要提前在需要的地方打上断点。如果要观察相应寄存器的变化的时候还需要提前调出观察窗口，跟随调试过程一步一步观察不同寄存器的变化过程。整个流程下来有很多细节需要掌握，否则在后面的过程中很有可能会因为这样那样的问题报错。期望在拿到实物后能对keil的理解以及掌握的更好。

五.参考

https://blog.csdn.net/ChenJ_1012/article/details/120520933

https://blog.csdn.net/qq_45659777/article/details/120496577