** **

**嵌入式原理及应用Ⅱ：嵌入式系统**

**实验报告一**

|  |  |  |
| --- | --- | --- |
| **指导老师** | ： |  |
| **姓名** | ： |  |
| **班级** | ： |  |
| **学号** | ： |  |

二〇二一年十一月

**目录**

[实验一 ADS集成开发环境的熟悉 1](#_Toc87108256)

[实验二 ARM 汇编程序的设计 5](#_Toc87108257)

[程序源码： 5](#_Toc87108258)

[AXD调试截图： 6](#_Toc87108259)

[实验三 ARM 汇编与 C 混合编程 8](#_Toc87108260)

[程序流程图： 8](#_Toc87108261)

[实验四 ARM 中断实验 9](#_Toc87108262)

[程序流程图： 9](#_Toc87108263)

[思考题 10](#_Toc87108264)

[体会和建议 12](#_Toc87108265)

**第一篇 windows环境下裸机开发实验**

# 实验一 ADS集成开发环境的熟悉

实验目标：ADS使用入门，熟悉开发板，能自主连接硬件、烧写程序。

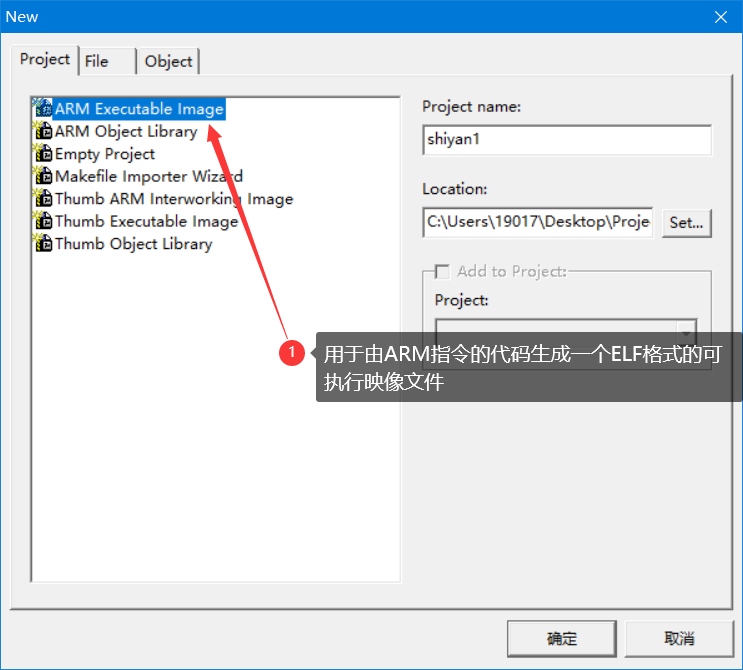
硬件操作：

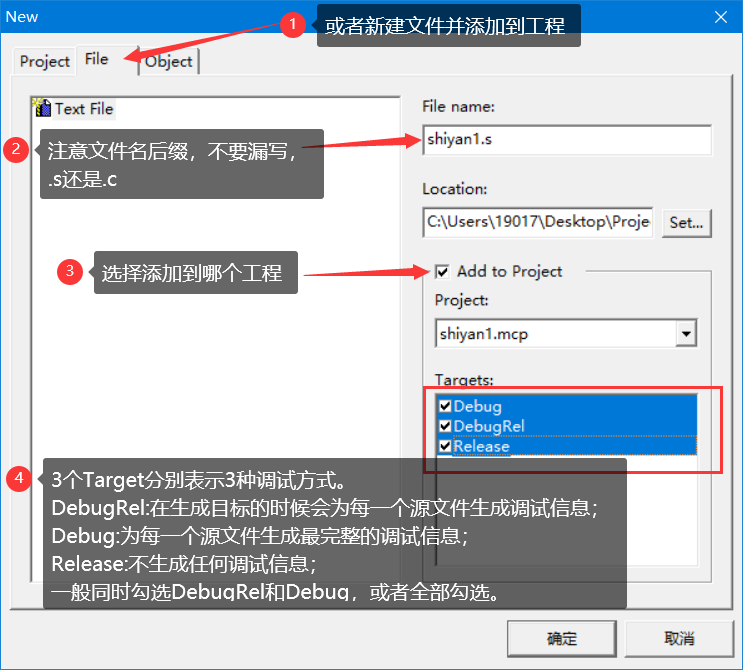
①开发板—(串口线、USB线)—电脑，开发板—(电源线)—插座。

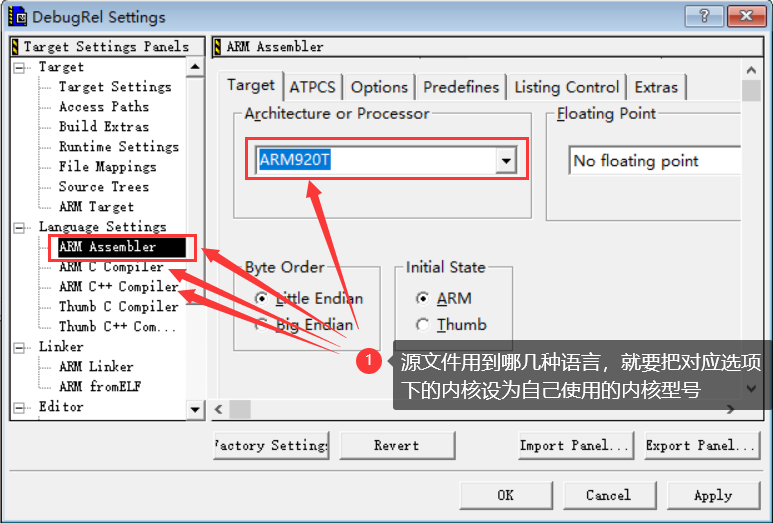
②开发板打到Nor Flash启动档。

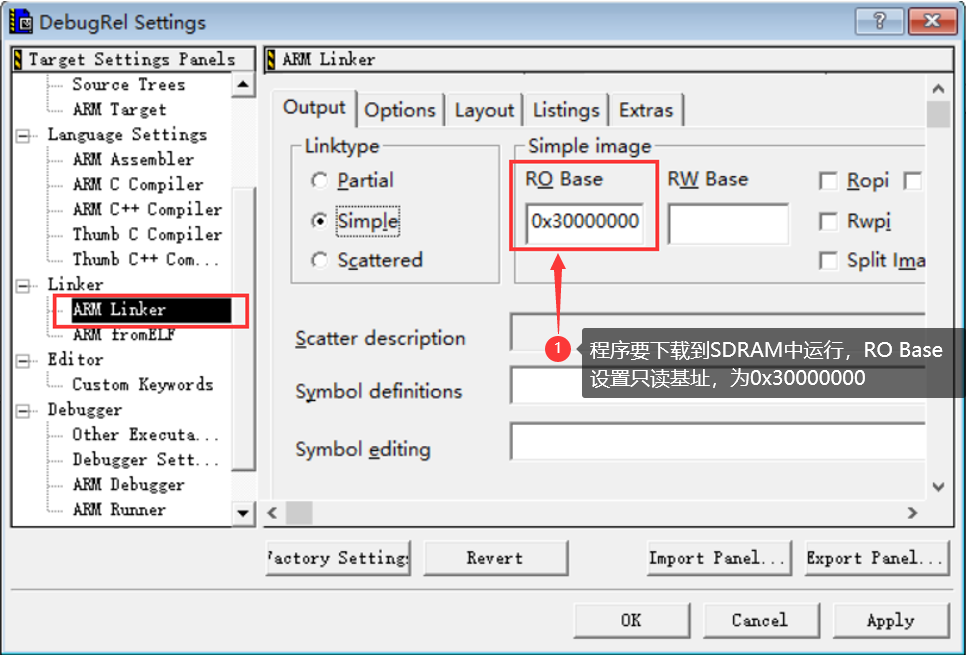
操作内容：

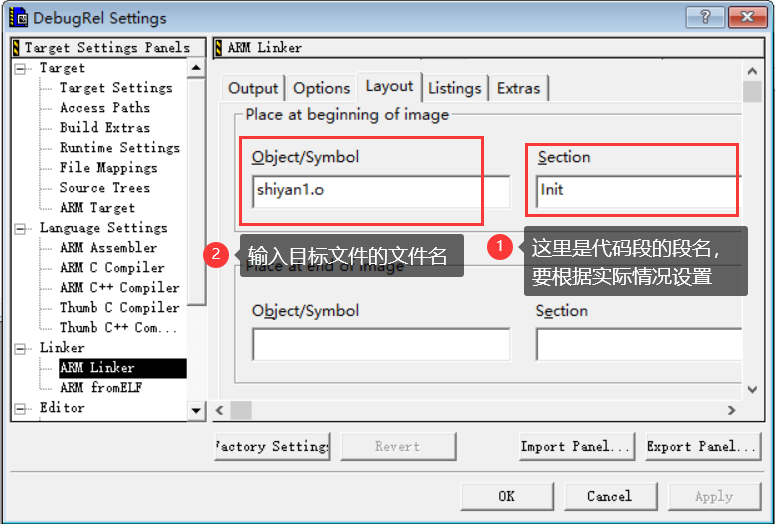
1. 在ADS中创建工程、为工程添加源文件、设置编译环境“DebugRel Settings”、编译。

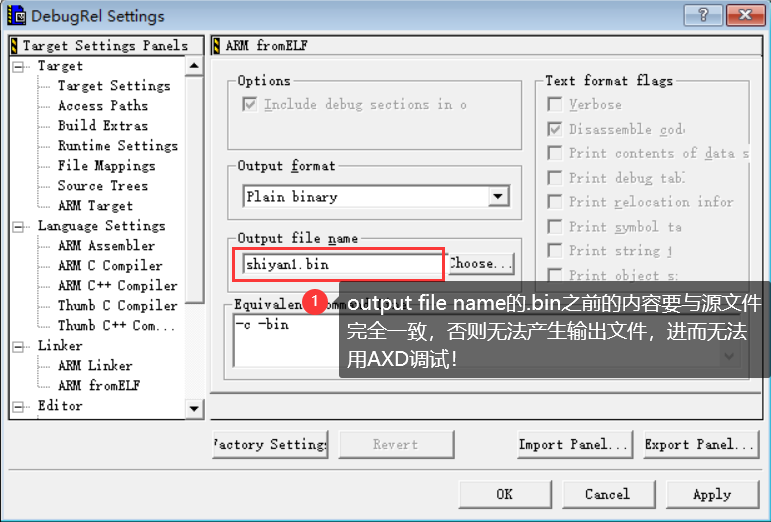












1. 编辑源文件，然后点击ADS主界面窗口的“Make”按钮编译源文件，当“Errors&Warnings”窗口的最后一行输出“Translation to Plain binary format successful.”这句话就表示成功输出.bin文件了。
2. 点击工程管理窗口的“Debug”按钮，打开AXD，配置环境选择内核“ARM920T”、“Load Iamge”中加载调试映像文件.axf
3. 下载可执行程序.bin到开发板：在完成连线、选择Nor Flash档之后，先使用串口超级终端“SecureCRT5.1”,选择第[7]个选项：“download program (uCOS-II or TQ2440\_Test)to SDRAM and run it”, 开发板厂家写好并固化在Nor Flash中的启动程序会把我们烧录的程序从Nor Flash拷贝到SDRAM（起始地址0x30000000）中运行。

最后使用USB下载软件“DNW”把.bin文件下载到开发板，程序即可运行。

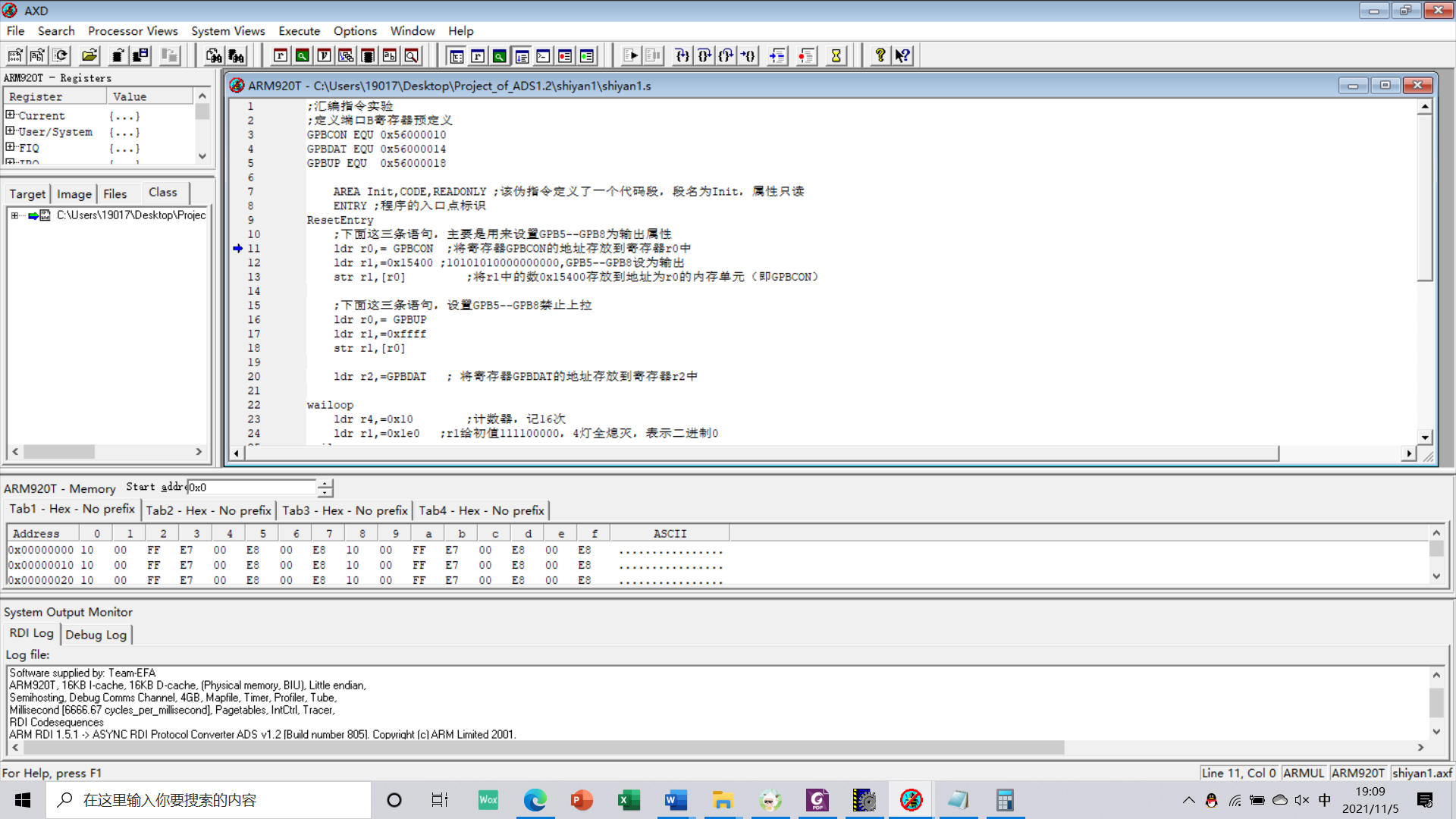
# 实验二 ARM 汇编程序的设计

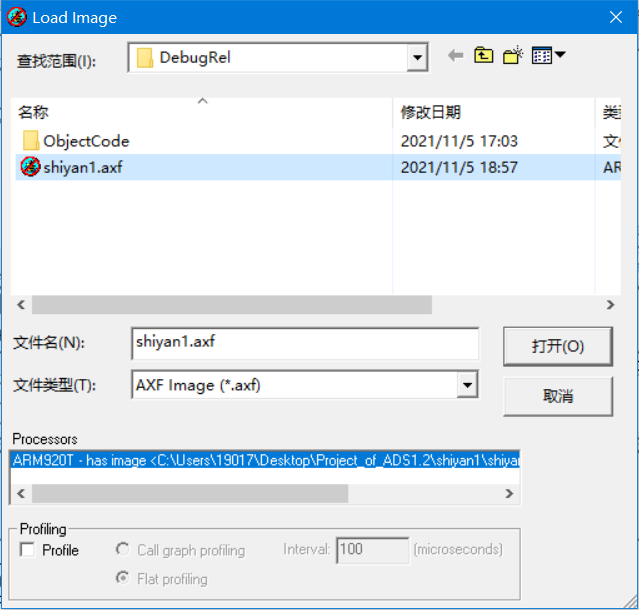
实验目标：用ADS编写汇编程序，实现TQ2440开发板上四个LED灯的二进制流水灯点亮。

## 程序源码：

1. ;汇编指令实验
2. ;定义端口B寄存器预定义
3. GPBCON EQU 0x56000010
4. GPBDAT EQU 0x56000014
5. GPBUP EQU  0x56000018
7. AREA Init,CODE,READONLY ;该伪指令定义了一个代码段，段名为Init，属性只读
8. ENTRY ;程序的入口点标识
9. ResetEntry
10. ;下面这三条语句，主要是用来设置GPB5--GPB8为输出属性
11. ldr r0,= GPBCON  ;将寄存器GPBCON的地址存放到寄存器r0中
12. ldr r1,=0x15400 ;10101010000000000,GPB5--GPB8设为输出
13. str r1,[r0];将0x15400存放到地址为r0的内存单元（即GPBCON）
14. ;下面这三条语句，设置GPB5--GPB8禁止上拉
15. ldr r0,= GPBUP
16. ldr r1,=0xffff
17. str r1,[r0]
18. ldr r2,=GPBDAT;将寄存器GPBDAT的地址存放到寄存器r2中
19. wailoop
20. ldr r4,=0x10  ;计数器，记16次
21. ldr r1,=0x1e0 ;r1给初值111100000，4灯全熄灭，表示二进制0
22. neiloop
23. str r1,[r2];使GPB5输出低电平，GPB6--GPB8输出高电平，LED1亮
24. bl delay
25. sub r1,r1,#0x20 ;r1每次减100000，16次后减为0
26. sub r4,r4,#1 ;计数器每次减1
27. cmp r4,#0
28. bne neiloop ;计数器不到0就继续循环
30. b wailoop;r4=0,r1=0,4灯全亮,表示二进制15,进入下一循环
32. ;下面是延迟子程序
33. delay
34. ldr r3,=0xbffff ;设置延迟的时间
35. delay1
36. sub r3,r3,#1 ;r3=r3-1
37. cmp r3,#0x0 ;将 r3 的值与 0 相比较
38. bne delay1 ;r3 不为 0就继续调用 delay1,否则执行下一条语句
39. mov pc,lr ;lr里装有调用子程序时PC的备份，现在赋给pc，返回调用处
40. END  ;表示汇编结束

## AXD调试截图：

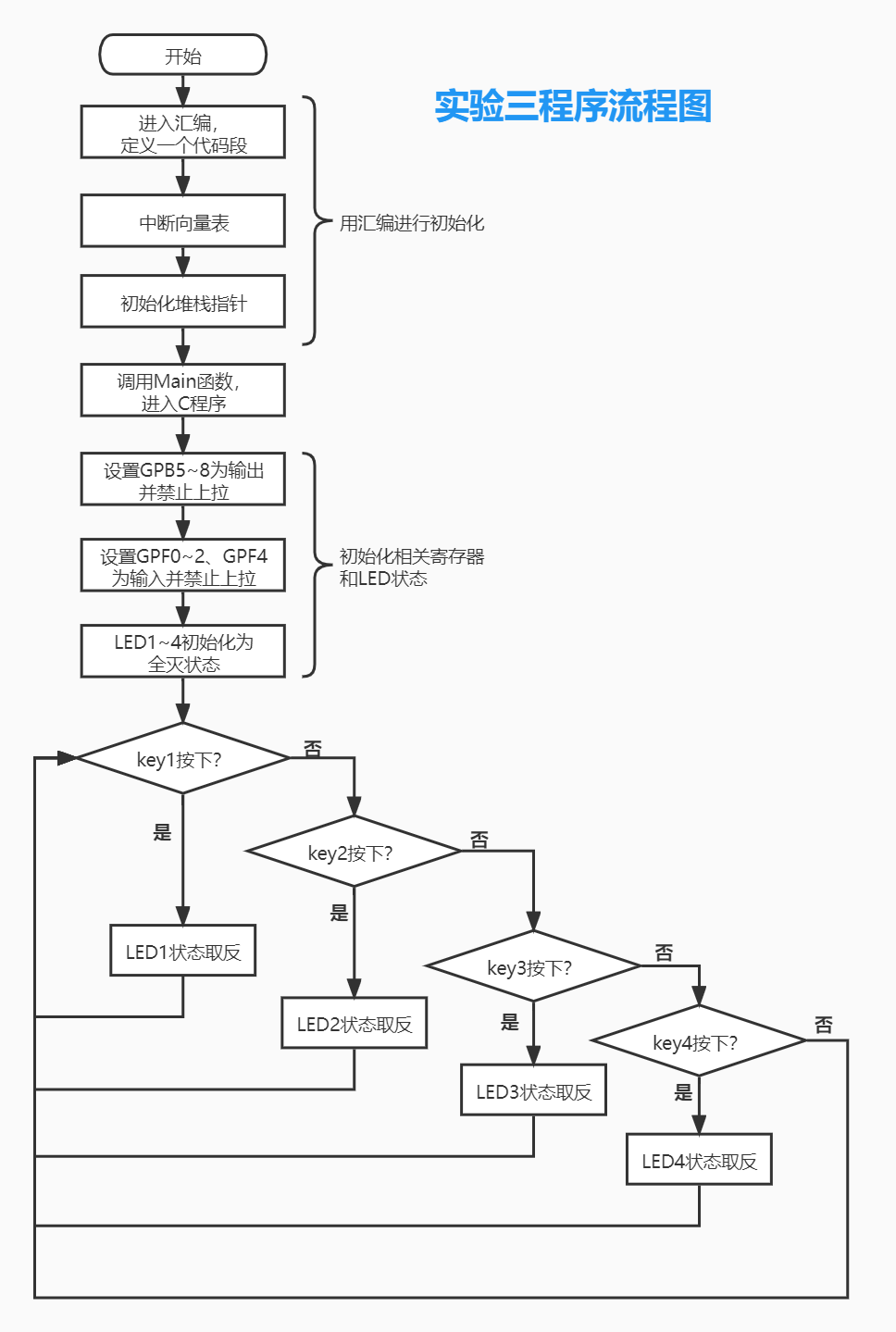




# 实验三 ARM 汇编与 C 混合编程

实验目标：C和汇编混合编程，用查询的方式实现4个按键控制4个LED灯。

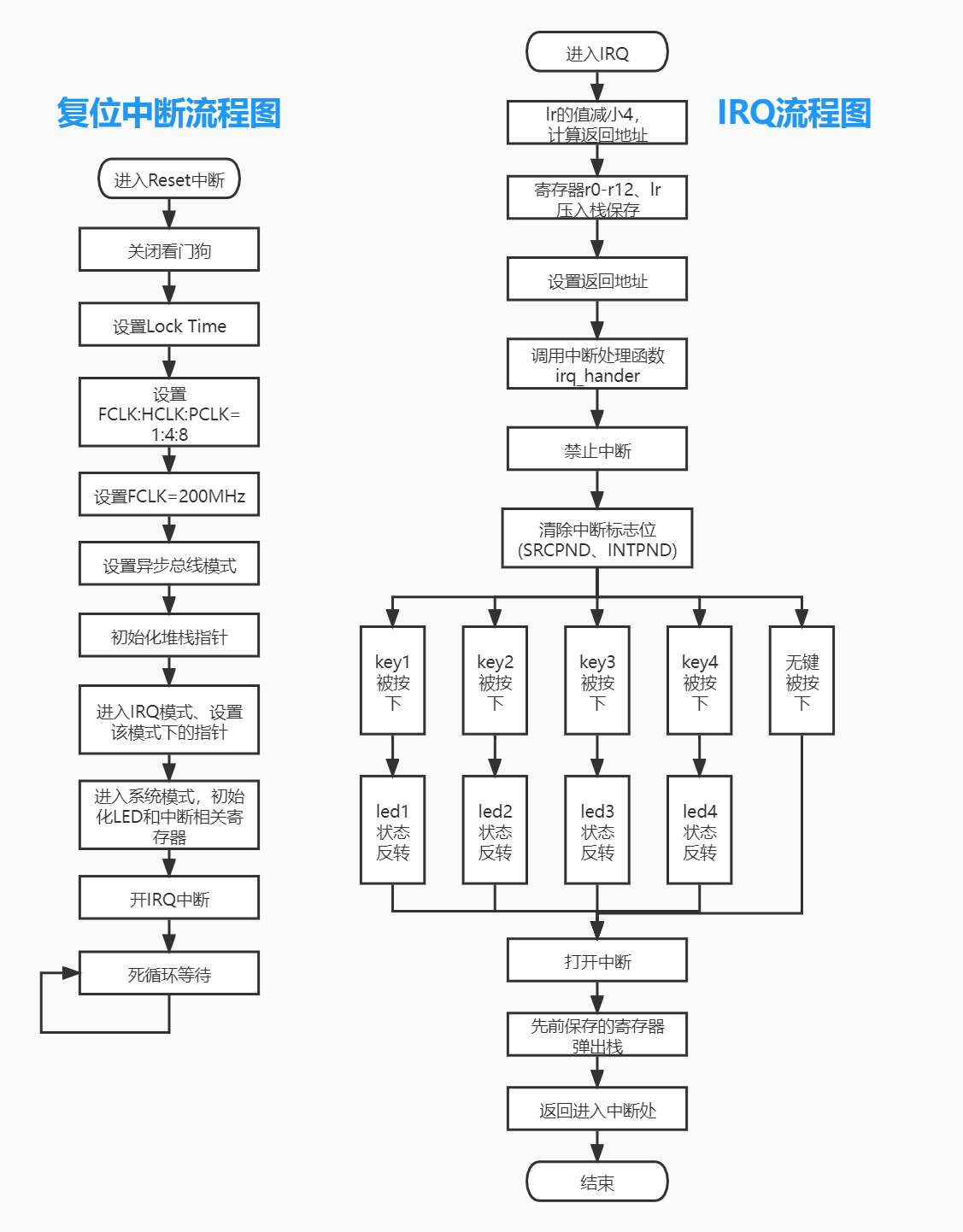
## 程序流程图：



# 实验四 ARM 中断实验

实验目标：C和汇编混合编程，用中断的方式实现4个按键控制4个LED灯的亮灭

## 程序流程图：



# 思考题

1. 在嵌入式系统编程当中，汇编语言和C语言分别有什么优势？是否可以完

全摈弃其中一种语言？为什么？

答：C语言的优点是：结构较好，便于理解，程序可读性强易维护，不完全依赖于硬件环境，可移植性强，且有大量的支持库。汇编的优点是：执行效率高，更符合实时性的要求，可以进行一些“精细处理”。

不可以完全摒弃他们中的任何一个。因为汇编能完成一些C语言做不到、不易做的事，例如开机时硬件系统的初始化，包括CPU状态的设定、中断的使能、主频的设定以及RAM的控制参数及初始化，一些中断处理方面也可能涉及汇编。另外一个使用汇编的地方就是一些对性能非常敏感的代码块，这不能依靠C编译器生成代码，而要手工编写汇编才能达到优化的目的。而程序的主体部分一般还是用更容易的C语言编写。

1. ARM汇编调用C语言以及C语言调用ARM汇编时，如何传递参数？实验二, 三程序中参数是如何传递的？

答：当参数不超过4个时，使用寄存器R0~R3来传递参数；当参数超过4个时，还可以使用数据栈来传递。在参数传递时，将所有参数看作是存放在连续的内存字单元中的宇数据。然后，依次将各字数据传送到寄存器R0、Rl、R2、R3中，如果参数多于4个，将剩余的字数据传送到数据栈中，入栈的顺序与参数顺序相反，即最后一个字数据先入栈。

实验三中只有C程序调用汇编的delay函数涉及参数传递，C程序中的“delay(0x1000)”的形参0x1000存入寄存器R0，从而传递给汇编里的delay函数。

1. c语言中和汇编语言中是如何操作寄存器的？

答：在C语言中，若要使用寄存器A（地址0x48000000），则使用一个句子：“#define A (\*(volatile unsigned long \*) 0x48000000”，把寄存器地址强制转换成volatile unsigned long类型的指针（volatile 限定编译器不对这个指针指向的存储单元进行优化,每次都直接到存储器里取值），然后再用一个“\*”解引用，这样变量“A”就表示寄存器地址里的值了，于是，就能通过给A赋值操作寄存器了。

在汇编语言中，使用“两ldr一str”的三句话组合来操作寄存器。第一句“LDR Rm,=0x**xxxxxxxx**”把目标寄存器地址存到寄存器Rm，第二句“LDR Rn,=0x**xxxxxxxx**”把一个数存到寄存器Rn，第三句“STR Rn,[Rm]”寄存器间接寻址把Rn里的数存到目标寄存器里。

1. 比较实验二、三和四中ADS下的工程设置的有何异同，并分析其理由。

答：实验二仅使用了汇编语言，所以在“DebugRel Settings-Language Settings”目录下只需要将“ARM Assembler”的“Target”栏下芯片内核选为ARM920T,而实验三同时用了汇编和C语言，所以“ARM Assembler”和“ARM C Compiler”下都要设置为ARM920T。

实验二、三都是下载程序到SDRAM并运行，所以RO Base和USB下载程序DNW中的“Download Address”都要设置为SDRAM的起始地址0x30000000。而实验四因为是Nor Flash下载程序，NAND Flash启动运行程序，程序的前4k会被拷贝至Steppingstone(本质是SRAM),而Steppingstone映射的地址空间是(nGCS0)0x00000000,所以RO Base要设置为0x00000000；而在SecureCRT5.1中选项[a]是把程序下载到Nand Flash里（起始地址0x00000000）,所以USB下载程序DNW中的“Download Address”也要设置为0x00000000。

1. 在中断实验中为什么要把可执行程序下载到NANDFlash中运行，而不是直

接下载到SDRAM中运行？如果直接下载到SDRAM中运行会发生什么情况？

答：中断向量表位于地址0x30000000以下（比如IRQ中断向量地址为0x00000018、FIQ中断向量地址为0x0000001c），而SDRAM是映射到地址0x30000000之后。此程序中使用了中断， 若程序下载到SDRAM中，则会因中断向量地址不对（实际值=原本理论值+0x30000000）而无法通过中断向量表跳转到相应的中断处理程序，则程序无法正常执行。故要将程序下载到NAND Flash 中运行。

1. 结合实验，叙述NAND Flash启动的流程。

答：上电后，NAND Flash的前4k被拷贝到Steppingstone（本质是速度快的SRAM，映射到地址空间nGCS0 0x00000000,大小4k），然后CPU开始执行Steppingstone中的引导程序。在引导程序中，将程序的剩余部分拷贝到SDRAM中，然后，跳到SDRAM中运行。

# 体会和建议

①要重点理解Nor Flash启动和Nand Flash启动这两种过程，发生了什么、涉及哪些存储单元、相关地址是多少，还有在SecureCRT5.1中的选项[7]和[a]分别执行了什么操作，发生了什么（例如[7]下载程序到SDRAM，程序是不经过Nor Flash的），这样才能理解ADS里RO Base和DNW里Download Address为什么这么设置的原因。

②遇到问题自己看书、百度、查阅资料仍不能解决的就要多和同学交流、多问老师，只有不断解决遇到的问题、不懂的地方，才会对原理有更全面深刻的理解、才能进步，若是遇到问题一味回避，不久就会积重难返，无力回天，十分感谢老师对我大大小小问题的悉心解答，以及同学们的热心帮助，这些都给了我莫大的鼓励和信心，使我完成了本次实验。