

STM32 浅谈之 IAP

我一点都不懂 IAP，看了数据手册，那都是讲如何操作芯片 FLASH 的，根本没有任何帮助！

手头有 IAP 例程，但是要搞懂需要先知道 IAP 原理。如果原理不懂，即使你是大神也束手无策。刚开始查看原理，都说的很模糊。偶尔碰到了 STM32 学校的视频教程，下载看了一下，对 IAP 原理了解。视频教程在优酷里面就可以找到，百度视频一搜就很多。其实 IAP 根本没什么，很简单。我使用的是 STM32F107VC，256KB flash，64kb RAM

IAP 简单介绍

IAP 需要两个程序，一个程序用于引导。查看是否更新程序，是否需要下载程序，是否正常运行。判断的这个关键字，存在 STM32 内部 FLASH 的一个地方，只需要读写即可。处理完成后需要跳转至 APP 程序。第二个程序，即 APP 程序，我们正常需要运行的程序。

IAP 中 APP 程序配置以及修改

找之前能正常运行的程序。

最好找个点亮 LED 的，这样看着直观

对之前工程进行更改

首先需要设置一个程序偏移，因为两个程序，第一个必须是 BOOT 程序，第二个程序为 APP 程序。他们都存放在 STM32 的内部 FLASH 中，所以这两个程序是不能重叠的。给 BOOT 留一些空间，首先要保证你的 APP 程序在 flash 中可以存储。根据实际情况来设定你程序的偏移。我的程序编译之后为 150KB 多。BOOT 程序才 7KB 多。我给 BOOT 程序 64K 的存储空间。这样我 APP 存储空间还有 190kb 左右，我的程序才 150kb，足可以放下。

在 main 程序添加中断向量的偏移。

```
#ifdef IAP
    // 应用 IAP 时需调整中断向量表
    NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x10000);
#endif
```

```

85 *
86 * Argument(s) :
87 *
88 * Return(s) : none.
89 *
90 * Caller(s) :
91 *
92 * Note(s) : none.
93 *
94 *-----
95 //
96 int main(void)
97 {
98     struct tm cur_time; // 定义当前时间结构体变量
99     uint8_t csw = 0;
100     struct tm Signin_time;
101     struct LED;
102     // 应用IAP时需清楚中断向量表
103     NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x10000);
104     #endif
105
106     System_Init();
107     // 显示运行正常
108     Display_Run(p_DeviceID, 0); // 系统主循环
109     while(1)
110     {
111         // 重新加载独立看门狗定时器
112         IWDG_ReloadCounter(); // 喂狗 超时时间设置为200
113         Function_Key(); // 查看是否有按键产生
114         Check_LED_Light(); // 查看是否有LED背光
115         // 判断是否需要更新ID
116         if (Write_dongbiao_ID == 1)
117         {
118             m2SpexX_erase_page(0x10000);
119             debug_info(DEBUGOUT, DEBUG1, "dongbiao ID Erase OK. \r\n");
120             m2SpexX_write_page(0x10000, (uint8_t *)dongbiao_id_temp); // 将ID写入flash 0x10000
121             Write_dongbiao_ID = 0;
122         }
123     }
124 }

```

这个偏移量不是随便来的，这个偏移跟你的程序存储偏移一致。由于程序在上电后先跳至复位中断中，进行初始化。

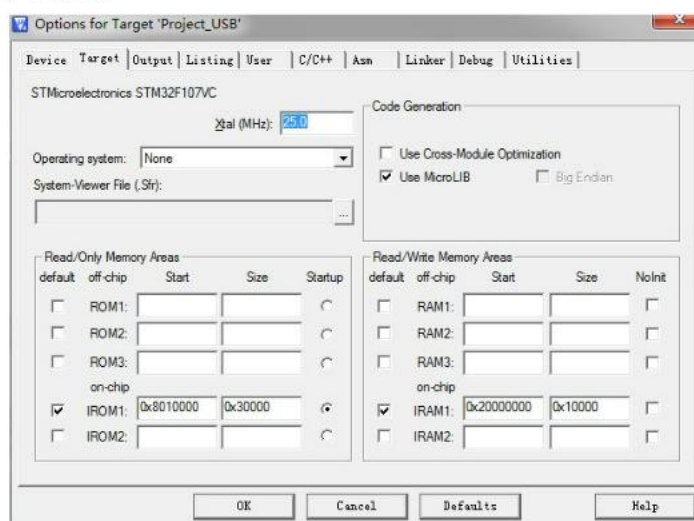
很多时候因为在别的地方有同样偏移设置，导致偏移未成功，程序总是跳转到硬件错误中断中！一般在中断初始化中!!!

```

98 *
99 * Argument(s) : none.
100 *
101 * Return(s) : none.
102 *
103 * Caller(s) : Application.
104 *
105 *-----
106 //
107 void NVIC_Configuration(void)
108 {
109     NVIC_InitTypeDef NVIC_InitStructure;
110     // NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x0);
111
112     // 定时器2中断配置
113     NVIC_InitStructure.NVIC_IRQChannel = TIM2_IRQn;
114     NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
115     NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
116     NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
117     NVIC_Init(&NVIC_InitStructure);
118
119     // 看门狗中断配置
120     NVIC_InitStructure.NVIC_IRQChannel = WWDG_IRQn;
121     NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 3;
122     NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
123     NVIC_Init(&NVIC_InitStructure);
124 }
125
126 //
127 //-----
128 *
129 * GPIO_Config()
130 * Description : GPIO configuration (LED) .
131 *
132 * Argument(s) : none.
133 *

```

在 KEIL 中配置程序偏移。



在 keil 中需要配置 IROM1 中的值，默认为 Start:0x0800 0000 Size:0x40000 将程序偏移 0x10000，所以 Start:0x0801 0000 Size:0x30000

内存是不需要进行配置的！ 如果将内存配置偏移后，有可能导致你的程序堆栈溢出，程序跳入硬件错误！

```
void HardFault_Handler(void)
{
    /* Go to infinite loop when Hard Fault exception occurs */
    while (1)
    {
    }
}
```

这样之后可以使用 H-JTAG 或者 J-LINK 下载程序！你复位后程序是不运行的！为了查看我们程序中断偏移量或者我们的程序下载进去是否能运行。我们添加一个设置：

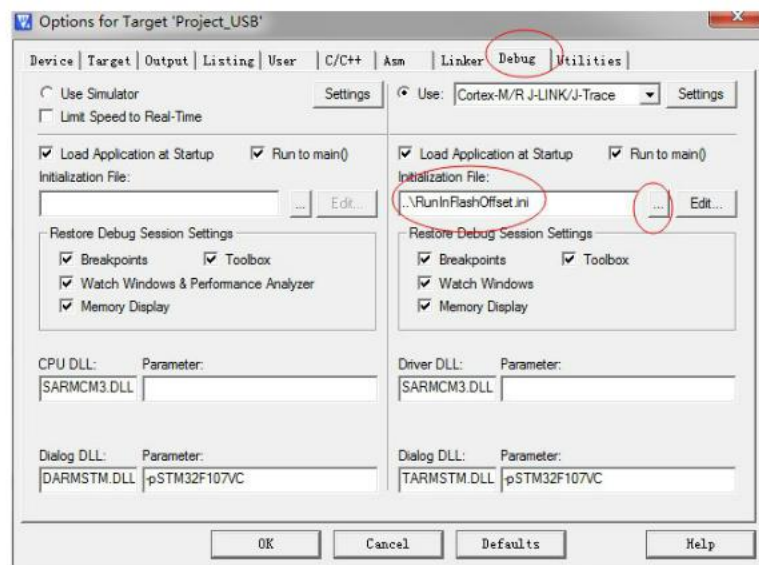
在工程文件中新建一个文本，重命名为：RunInFlashOffset.ini

里面只有两个配置：

```
SP = _RDWORD(0x08010000);          // Setup Stack Pointer
PC = _RDWORD(0x08010004);          // Setup Program Counter
```

其中 SP 为你程序偏移后的地址！PC 为要执行的地址，是程序起始地址的下一条语句！也是存放复位中断函数的地方。

然后在 keil 中进行添加！



在 DEBUG 中添加刚才新建的文件，这样在线调试就 OK 了!!! 只能在线调试，复位后程序任然不允许的！

程序跳转

在 BOOT 中将程序写入 FLASH 后，需要进行跳转！

程序：

```
/* Jump to user application */
JumpAddress = *(__IO uint32_t*) (ApplicationAddress + 4);
Jump_To_Application = (pFunction) JumpAddress;
/* Initialize user application's Stack Pointer */
__set_MSP(*(__IO uint32_t*) ApplicationAddress);
```

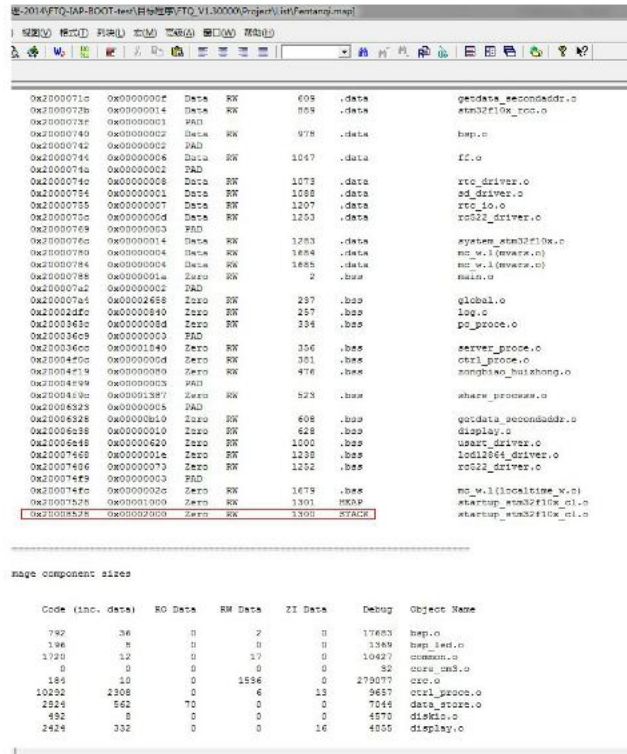

Jump_To_Application();

程序就这样，不要怀疑，他可以跳转！

我很不理解为什么要设置堆栈指针，为什么程序的起始地址为堆栈指针呢？

我们在 APP 工程中查看（APP 更改过偏移），有一个文件：Fentanqi.map（即*.map）

在 LIST 文件夹下或者别的地方。打开这个文件之后你会对你自己的程序有一个实在的了解！它不需要解释，你一眼就可以看到你的全局变量是如何存储，堆在哪里，栈在哪里。程序存储哪里，程序多大！



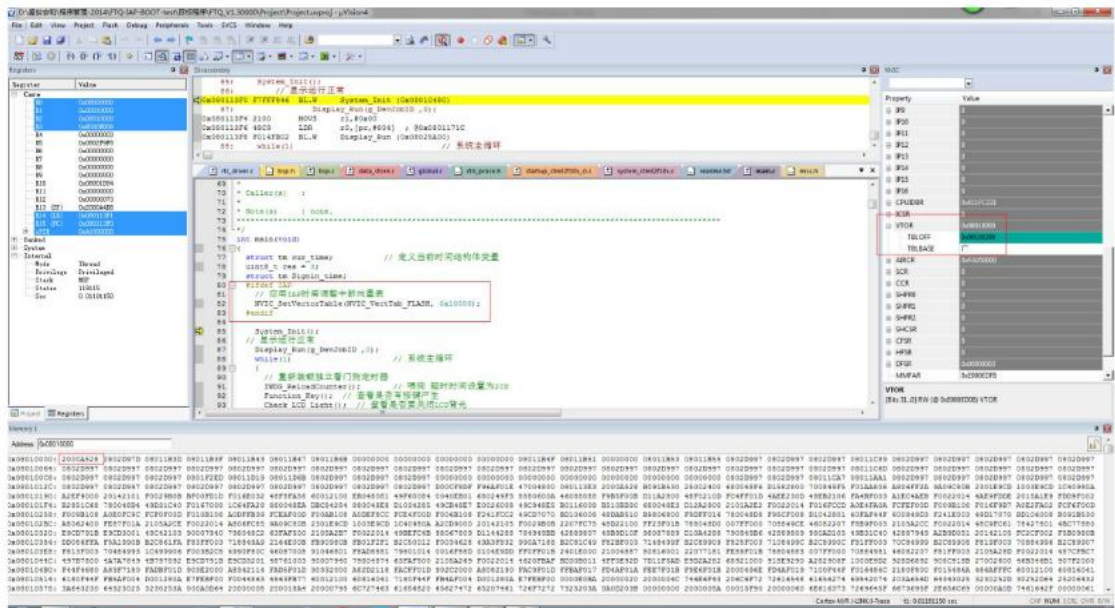
Address	Size	Type	Name
0x20000710	0x00000002	Data RW	608 .data gendata_secondaddr.o
0x20000720	0x000000014	Data RW	858 .data stm32f10x_rcc.o
0x20000730	0x000000003	DATA	
0x20000740	0x000000002	Data RW	978 .data bsp.o
0x20000742	0x000000002	DATA	
0x20000744	0x000000006	Data RW	1047 .data ff.o
0x20000746	0x000000002	DATA	
0x2000074c	0x000000008	Data RW	1073 .data rtc_driver.o
0x20000754	0x000000001	Data RW	1088 .data sd_driver.o
0x20000755	0x000000007	Data RW	1207 .data rtc_io.o
0x2000075c	0x000000004	Data RW	1253 .data rtc2_driver.o
0x20000769	0x000000003	DATA	
0x2000076c	0x000000014	Data RW	1283 .data system_stm32f10x.o
0x20000780	0x000000004	Data RW	1884 .data mc_w1[hware.o]
0x20000784	0x000000004	Data RW	1885 .data mc_w1[hware.o]
0x20000788	0x000000001a	Zero RW	2 .bss main.o
0x200007a2	0x000000002	DATA	
0x200007a4	0x000000008	Zero RW	287 .bss global.o
0x200007d0	0x000000009	Zero RW	287 .bss log.o
0x200007d3	0x000000008	Zero RW	334 .bss pg_proc.o
0x200007d8	0x000000003	DATA	
0x200007dc	0x000000140	Zero RW	356 .bss server_proc.o
0x200007e0	0x00000000d	Zero RW	361 .bss ctrl_proc.o
0x200007e4	0x00000000d	Zero RW	476 .bss songtian_huizhong.o
0x200007f0	0x000000003	DATA	
0x200007f4	0x000000187	Zero RW	523 .bss share_process.o
0x200007f8	0x000000005	DATA	
0x20000828	0x000000010	Zero RW	608 .bss gendata_secondaddr.o
0x20000838	0x000000010	Zero RW	628 .bss display.o
0x20000848	0x000000020	Zero RW	1200 .bss USART_driver.o
0x20000850	0x00000001e	Zero RW	1238 .bss led12064_driver.o
0x20000854	0x000000017	Zero RW	1252 .bss rtc2_driver.o
0x20000858	0x000000003	DATA	
0x2000085c	0x00000000d	Zero RW	1679 .bss mc_w1[localtime_v.o]
0x20000860	0x000000100	Zero RW	1301 .bss startup_stm32f10x.o
0x20000864	0x000000000	Zero RW	1301 .bss startup_stm32f10x.o

Code (inc. data)	RO Data	RW Data	ZI Data	Debug	Object Name
792	36	0	2	0	17653 bsp.o
196	8	0	0	0	1389 bsp_led.o
1720	12	0	17	0	10427 common.o
0	0	0	0	0	52 ctrl.o
184	10	0	1536	0	279077 ctrl.o
10252	2308	0	6	13	9657 ctrl_proc.o
234	562	70	0	0	7445 data_store.o
452	8	0	0	0	4570 display.o
2424	332	0	0	16	4555 display.o

上图中你可以看到栈在内存的最后面
它的起始地址为:0x2000 8528 截止地址为: 0x2001 0000
程序在 flash 中的存储，第一个为栈顶地址！



Address	Value
0x20008528	0x00000000
0x20008529	0x00000000
0x2000852a	0x00000000
0x2000852b	0x00000000
0x2000852c	0x00000000
0x2000852d	0x00000000
0x2000852e	0x00000000
0x2000852f	0x00000000
0x20008530	0x00000000
0x20008531	0x00000000
0x20008532	0x00000000
0x20008533	0x00000000
0x20008534	0x00000000
0x20008535	0x00000000
0x20008536	0x00000000
0x20008537	0x00000000
0x20008538	0x00000000
0x20008539	0x00000000
0x2000853a	0x00000000
0x2000853b	0x00000000
0x2000853c	0x00000000
0x2000853d	0x00000000
0x2000853e	0x00000000
0x2000853f	0x00000000
0x20008540	0x00000000
0x20008541	0x00000000
0x20008542	0x00000000
0x20008543	0x00000000
0x20008544	0x00000000
0x20008545	0x00000000
0x20008546	0x00000000
0x20008547	0x00000000
0x20008548	0x00000000
0x20008549	0x00000000
0x2000854a	0x00000000
0x2000854b	0x00000000
0x2000854c	0x00000000
0x2000854d	0x00000000
0x2000854e	0x00000000
0x2000854f	0x00000000
0x20008550	0x00000000
0x20008551	0x00000000
0x20008552	0x00000000
0x20008553	0x00000000
0x20008554	0x00000000
0x20008555	0x00000000
0x20008556	0x00000000
0x20008557	0x00000000
0x20008558	0x00000000
0x20008559	0x00000000
0x2000855a	0x00000000
0x2000855b	0x00000000
0x2000855c	0x00000000
0x2000855d	0x00000000
0x2000855e	0x00000000
0x2000855f	0x00000000
0x20008560	0x00000000
0x20008561	0x00000000
0x20008562	0x00000000
0x20008563	0x00000000
0x20008564	0x00000000
0x20008565	0x00000000
0x20008566	0x00000000
0x20008567	0x00000000
0x20008568	0x00000000
0x20008569	0x00000000
0x2000856a	0x00000000
0x2000856b	0x00000000
0x2000856c	0x00000000
0x2000856d	0x00000000
0x2000856e	0x00000000
0x2000856f	0x00000000
0x20008570	0x00000000
0x20008571	0x00000000
0x20008572	0x00000000
0x20008573	0x00000000
0x20008574	0x00000000
0x20008575	0x00000000
0x20008576	0x00000000
0x20008577	0x00000000
0x20008578	0x00000000
0x20008579	0x00000000
0x2000857a	0x00000000
0x2000857b	0x00000000
0x2000857c	0x00000000
0x2000857d	0x00000000
0x2000857e	0x00000000
0x2000857f	0x00000000
0x20008580	0x00000000
0x20008581	0x00000000
0x20008582	0x00000000
0x20008583	0x00000000
0x20008584	0x00000000
0x20008585	0x00000000
0x20008586	0x00000000
0x20008587	0x00000000
0x20008588	0x00000000
0x20008589	0x00000000
0x2000858a	0x00000000
0x2000858b	0x00000000
0x2000858c	0x00000000
0x2000858d	0x00000000
0x2000858e	0x00000000
0x2000858f	0x00000000
0x20008590	0x00000000
0x20008591	0x00000000
0x20008592	0x00000000
0x20008593	0x00000000
0x20008594	0x00000000
0x20008595	0x00000000
0x20008596	0x00000000
0x20008597	0x00000000
0x20008598	0x00000000
0x20008599	0x00000000
0x2000859a	0x00000000
0x2000859b	0x00000000
0x2000859c	0x00000000
0x2000859d	0x00000000
0x2000859e	0x00000000
0x2000859f	0x00000000
0x200085a0	0x00000000
0x200085a1	0x00000000
0x200085a2	0x00000000
0x200085a3	0x00000000
0x200085a4	0x00000000
0x200085a5	0x00000000
0x200085a6	0x00000000
0x200085a7	0x00000000
0x200085a8	0x00000000
0x200085a9	0x00000000
0x200085aa	0x00000000
0x200085ab	0x00000000
0x200085ac	0x00000000
0x200085ad	0x00000000
0x200085ae	0x00000000
0x200085af	0x00000000
0x200085b0	0x00000000
0x200085b1	0x00000000
0x200085b2	0x00000000
0x200085b3	0x00000000
0x200085b4	0x00000000
0x200085b5	0x00000000
0x200085b6	0x00000000
0x200085b7	0x00000000
0x200085b8	0x00000000
0x200085b9	0x00000000
0x200085ba	0x00000000
0x200085bb	0x00000000
0x200085bc	0x00000000
0x200085bd	0x00000000
0x200085be	0x00000000
0x200085bf	0x00000000
0x200085c0	0x00000000
0x200085c1	0x00000000
0x200085c2	0x00000000
0x200085c3	0x00000000
0x200085c4	0x00000000
0x200085c5	0x00000000
0x200085c6	0x00000000
0x200085c7	0x00000000
0x200085c8	0x00000000
0x200085c9	0x00000000
0x200085ca	0x00000000
0x200085cb	0x00000000
0x200085cc	0x00000000
0x200085cd	0x00000000
0x200085ce	0x00000000
0x200085cf	0x00000000
0x200085d0	0x00000000
0x200085d1	0x00000000
0x200085d2	0x00000000
0x200085d3	0x00000000
0x200085d4	0x00000000
0x200085d5	0x00000000
0x200085d6	0x00000000
0x200085d7	0x00000000
0x200085d8	0x00000000
0x200085d9	0x00000000
0x200085da	0x00000000
0x200085db	0x00000000
0x200085dc	0x00000000
0x200085dd	0x00000000
0x200085de	0x00000000
0x200085df	0x00000000
0x200085e0	0x00000000
0x200085e1	0x00000000
0x200085e2	0x00000000
0x200085e3	0x00000000
0x200085e4	0x00000000
0x200085e5	0x00000000
0x200085e6	0x00000000
0x200085e7	0x00000000
0x200085e8	0x00000000
0x200085e9	0x00000000
0x200085ea	0x00000000
0x200085eb	0x00000000
0x200085ec	0x00000000
0x200085ed	0x00000000
0x200085ee	0x00000000
0x200085ef	0x00000000
0x200085f0	0x00000000
0x200085f1	0x00000000
0x200085f2	0x00000000
0x200085f3	0x00000000
0x200085f4	0x00000000
0x200085f5	0x00000000
0x200085f6	0x00000000
0x200085f7	0x00000000
0x200085f8	0x00000000
0x200085f9	0x00000000
0x200085fa	0x00000000
0x200085fb	0x00000000
0x200085fc	0x00000000
0x200085fd	0x00000000
0x200085fe	0x00000000
0x200085ff	0x00000000
0x20008600	0x00000000
0x20008601	0x00000000
0x20008602	0x00000000
0x20008603	0x00000000
0x20008604	0x00000000
0x20008605	0x00000000
0x20008606	0x00000000
0x20008607	0x00000000
0x20008608	0x00000000
0x20008609	0x00000000
0x2000860a	0x00000000
0x2000860b	0x00000000
0x2000860c	0x00000000
0x2000860d	0x00000000
0x2000860e	0x00000000
0x2000860f	0x00000000
0x20008610	0x00000000
0x20008611	0x00000000
0x20008612	0x00000000
0x20008613	0x00000000
0x20008614	0x00000000
0x20008615	0x00000000
0x20008616	0x00000000
0x20008617	0x00000000
0x20008618	0x00000000
0x20008619	0x00000000
0x2000861a	0x00000000
0x2000861b	0x00000000
0x2000861c	0x00000000
0x2000861d	0x00000000
0x2000861e	0x00000000
0x2000861f	0x00000000
0x20008620	0x00000000
0x20008621	0x00000000
0x20008622	0x00000000
0x20008623	0x00000000
0x20008624	0x00000000
0x20008625	0x00000000
0x20008626	0x00000000
0x20008627	0x00000000
0x20008628	0x00000000
0x20008629	0x00000000
0x2000862a	0x00000000
0x2000862b	0x00000000
0x2000862c	0x00000000
0x2000862d	0x00000000
0x2000862e	0x00000000
0x2000862f	0x00000000
0x20008630	0x00000000
0x20008631	0x00000000
0x20008632	0x00000000
0x20008633	0x00000000
0x20008634	0x00000000
0x20008635	0x00000000
0x20008636	0x00000000
0x20008637	0x00000000
0x20008638	0x00000000
0x20008639	0x00000000



我要写的也只有这些了，希望对你有帮助，不懂了再找找别的资料！