



YC31-b

向量表更新说明

Yichip Microelectronics

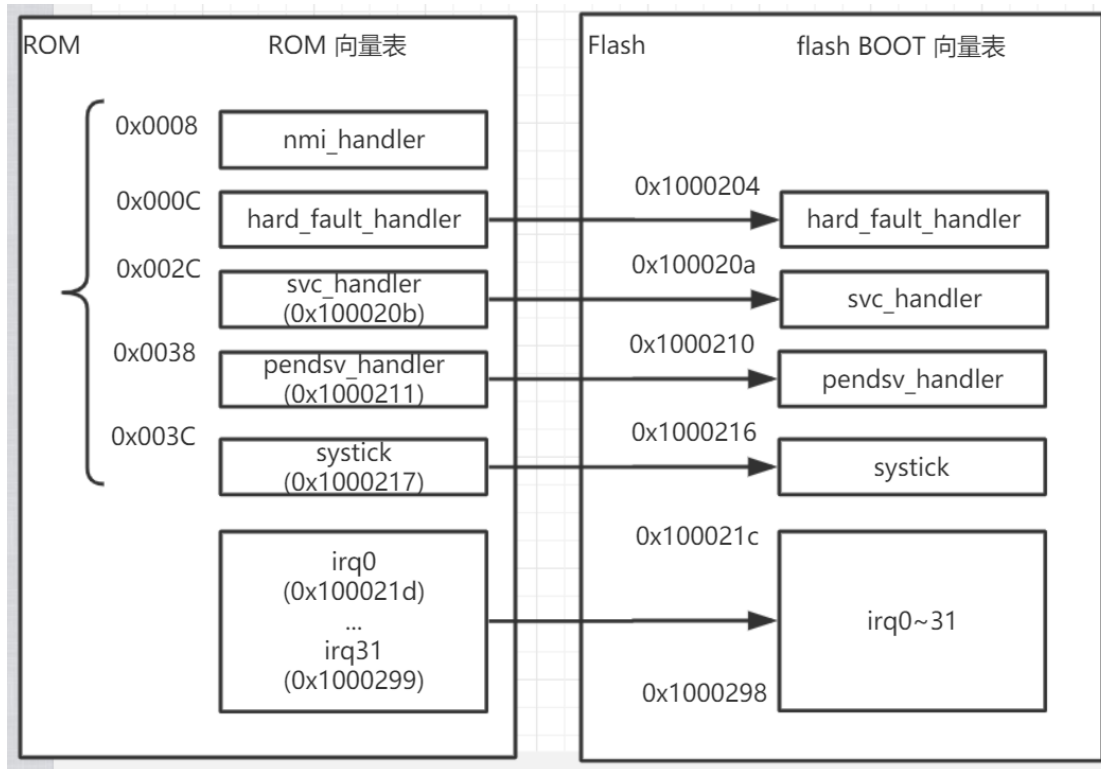
©2014

[illegible]

1.更新原理

由于 CM0 没有 VTOR 寄存器 我们是通过跳转的方式执行的

- 中断服务函数从 ROM 调转到 flash



Rom 中的中断向量表存储的地址是指向 flash 的固定地址的，例如：

向量表 0x3C 的地址存储的是“systick handle” 的服务函数地址 0x1000217

- Flash 中中断服务函数跳转（见 boot_flash_stat.s 的说明）
Flash 中的中断服务函数要按照上图中的地址一致，才能让中断顺利执行。
中断服务函数分成两种：

1. 系统中断服务函数
 - a) Hard_fault
 - b) Svc
 - c) Pendsv
 - d) systick
2. 外部中断服务函数
 - a) Irq0~31

外部中断跳转到 ISR,

```
7 references
irq0      PROC
          EXPORT irq0
          movs r0,#4*0
          b isr
          ENDP
```

系统中断跳转到 ISR1,

```
1 reference
pre_pendsv_handler PROC
          EXPORT pre_pendsv_handler
          movs r0,#6*2
          b isr1
          nop
          ENDP
```

根据对 ENABLEAPP 中的寄存器“0xf8564”(上电默认值为 2)的值判断当前中断服务函数是执行 BOOT 中的还是 APP 中的。具体见下图

系统中断服务函数:

```
4 references
isr1      PROC
          LDR R1,= ENABLEAPP
          LDR R1,[R1]
          LDR R1,[R1]
          CMP R1,#2
          BNE isr1_app
          LDR R1,=hard_fault_handler
          ADDS R0,R0,R1
          BX R0
          ENDP

1 reference
isr1_app  PROC
          LDR R1,= APPSTARTADDR
          LDR R1,[R1]
          ADDS R1,#4
          ADDS R1,R1,R0
          BX R1
          ENDP
```

得到寄存器地址

得到寄存器的值

不相等

start_flash函数占四个字节

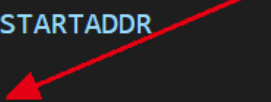
外部中断服务函数:

```
isr      PROC
        LDR R1,= ENABLEAPP
        LDR R1,[R1]
        LDR R1,[R1]
        CMP R1,#2
        BNE isr_app
        ldr r1,isr_table
        ldr r0,[r0, r1]
        blx r0
        ENDP

14 references
isr_ret  PROC
        movs r0,#0
        subs r0,#7
        mov lr,r0
        bx lr
        ENDP

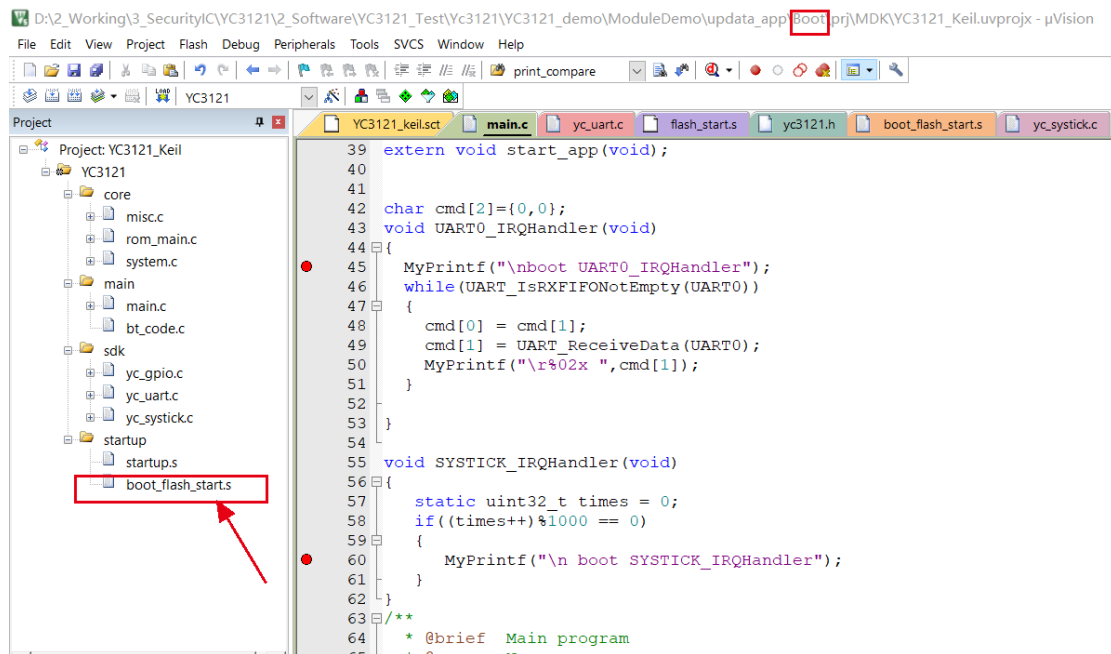
1 reference
isr_app  PROC
        LDR R1,= APPSTARTADDR
        LDR R1,[R1]
        ADDS R1,#156
        BX R1
        ENDP
```

下一级app 函数 "isr" 的偏移地址



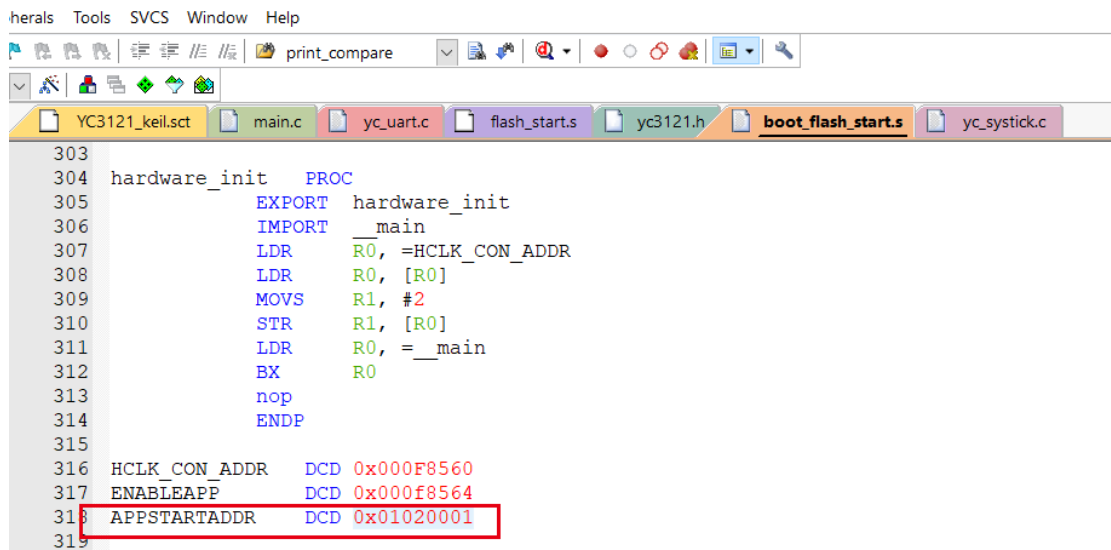
2.更新方法

1.step 1 boot 使用 “boot_flash_start.s”

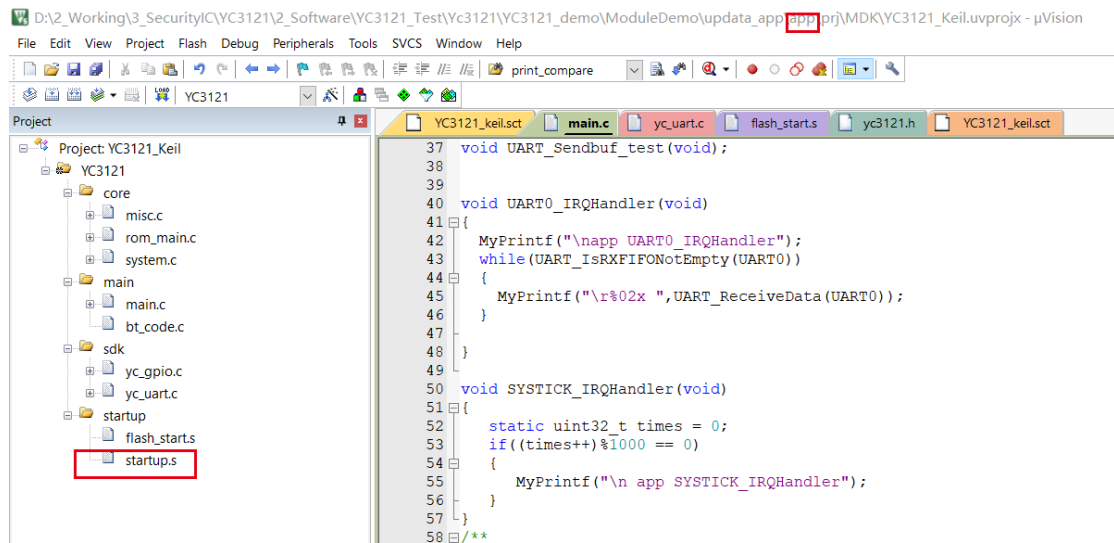


将 APPSTARTADDR 修改 为 APP 的起始地址

software\YC3121_Test\YC3121\YC3121_demo\ModuleDemo\updata_app\Boot\prj\MDK\YC3121_Keil.uvprojx - uVision



2.step2: APP 使用

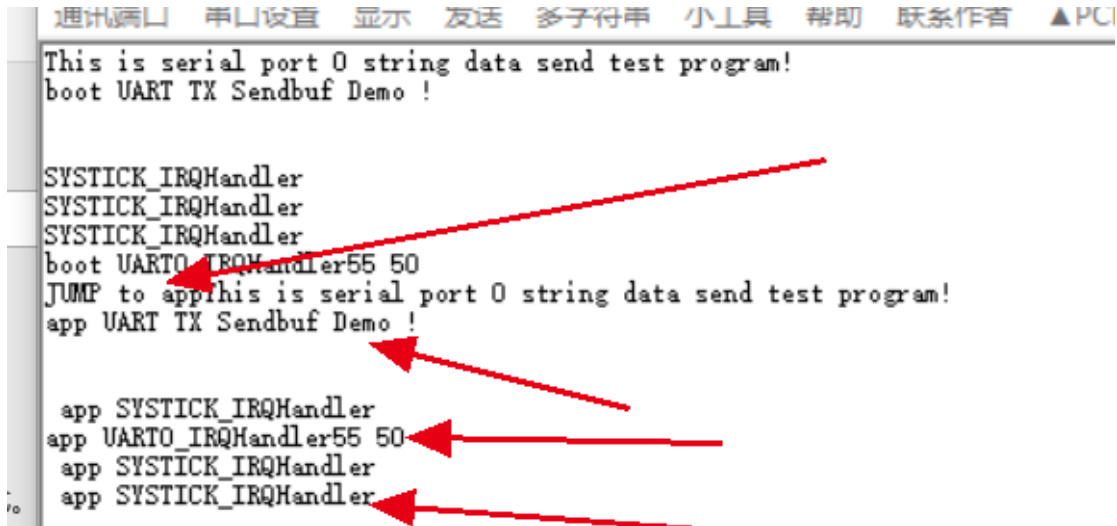


```

1 ;LR_IROM2 0x4000 0x4000{
2 ;   ER_IROM2 0x4000 {
3 ;       rom_main.o (+RO)
4 ;   }
5 ;}
6 LR_IROM3 0x1020000 0x200{
7     ER_IROM3 0x1020000 {
8         flash_start.o (|.flash_start|,+RO)
9     }
10 }
11
12 LR_IROM4 0x1020200 0x1000000{
13     ER_IROM4 0x1020200 {
14         flash_start.o (|.flash_start|,+RO)
15         *(InRoot$$Sections)
16         .ANY (+RO)
17     }
18
19     RW_IRAM2 0x00020000 0x010000 { ; RW data
20     .ANY (+RW +ZI)
21     }
22 }
23 }
24

```

3.如果使用串口一起下载 APP 和 BOOT 双击 genbin.bat 生成 flash.bin（需要安装 perl 脚本环境）



```
通信窗口 串口设置 显示 发送 多子符串 小工具 帮助 联系作者 ▲PC
This is serial port 0 string data send test program!
boot VART TX Sendbuf Demo !

SYSTICK_IRQHandler
SYSTICK_IRQHandler
SYSTICK_IRQHandler
boot VARTO_IRQHandler55 50
JUMP to appthis is serial port 0 string data send test program!
app VART TX Sendbuf Demo !

app SYSTICK_IRQHandler
app VARTO_IRQHandler55 50
app SYSTICK_IRQHandler
app SYSTICK_IRQHandler
```

4. 中