**Cortex-M3 内核HardFault错误调试定位方法**

**1、首先更改startup.s的启动文件，把里面的HardFault\_Handler代码段换成下面的代码：**

HardFault\_Handler\

PROC

IMPORT hard\_fault\_handler\_c

TST LR, #4

ITE EQ

MRSEQ R0, MSP

MRSNE R0, PSP

B hard\_fault\_handler\_c

ENDP

2、然后把hard\_fault\_handler\_c函数放在c文件的代码中。代码如下：

void hard\_fault\_handler\_c(unsigned int \* hardfault\_args)

{

static unsigned int stacked\_r0;

static unsigned int stacked\_r1;

static unsigned int stacked\_r2;

static unsigned int stacked\_r3;

static unsigned int stacked\_r12;

static unsigned int stacked\_lr;

static unsigned int stacked\_pc;

static unsigned int stacked\_psr;

static unsigned int SHCSR;

static unsigned char MFSR;

static unsigned char BFSR;

static unsigned short int UFSR;

static unsigned int HFSR;

static unsigned int DFSR;

static unsigned int MMAR;

static unsigned int BFAR;

stacked\_r0 = ((unsigned long) hardfault\_args[0]);

stacked\_r1 = ((unsigned long) hardfault\_args[1]);

stacked\_r2 = ((unsigned long) hardfault\_args[2]);

stacked\_r3 = ((unsigned long) hardfault\_args[3]);

stacked\_r12 = ((unsigned long) hardfault\_args[4]);

/\*异常中断发生时，这个异常模式特定的物理R14,即lr被设置成该异常模式将要返回的地址\*/

stacked\_lr = ((unsigned long) hardfault\_args[5]);

stacked\_pc = ((unsigned long) hardfault\_args[6]);

stacked\_psr = ((unsigned long) hardfault\_args[7]);

SHCSR = (\*((volatile unsigned long \*)(0xE000ED24))); //系统Handler控制及状态寄存器

MFSR = (\*((volatile unsigned char \*)(0xE000ED28))); //存储器管理fault状态寄存器

BFSR = (\*((volatile unsigned char \*)(0xE000ED29))); //总线fault状态寄存器

UFSR = (\*((volatile unsigned short int \*)(0xE000ED2A)));//用法fault状态寄存器

HFSR = (\*((volatile unsigned long \*)(0xE000ED2C))); //硬fault状态寄存器

DFSR = (\*((volatile unsigned long \*)(0xE000ED30))); //调试fault状态寄存器

MMAR = (\*((volatile unsigned long \*)(0xE000ED34))); //存储管理地址寄存器

BFAR = (\*((volatile unsigned long \*)(0xE000ED38))); //总线fault地址寄存器

while (1);

}

3、执行程序后，若发生内核错误，则程序会运行到最后的while(1)处。此时观察相应的堆栈和故障寄存器值， stacked\_lr即为故障发生时进入故障中断前pc的值，在MDK软件调试状态下，假如stacked\_lr的值为0x1A002D08，在左下方的命令窗口输入“pc = 0x1A002D08”，回车，即可定位发生错误的代码位置。

4、根据内核错误状态寄存器的值，对应下面的说明，也可以看出是发生了何种内核错误。

附录：Cortex-M3内核错误寄存器说明















