经过小弟的一个周末的努力，终于发现STM8S不能块编程的原因了。首先STM8S的块编程，需要把程序放在RAM中运行。对于像我这样没学过汇编的，也不了解编译器原理的，只知道用C语言写写程序的朋友。有点摸不着头脑。那么如何让程序在RAM中运行呢？

我们看下STM8S的库函数数就知道了。下面是库函数的代码。

IN\_RAM(void FLASH\_EraseBlock(uint16\_t BlockNum, FLASH\_MemType\_TypeDef FLASH\_MemType));

我刚开始就被这个函数定义给骗了。以为加了IN\_RAM( )，就可以在RAM中运行了。我们看下 IN\_RAM() 的定义

#define IN\_RAM(a) a

在默认情况下，它什么都没干。仔细一下，在默认情况下，是不库函数把在RAM中运行程序的功能给屏蔽了。

/\* Uncomment the line below to enable the FLASH functions execution from RAM \*/

#if !defined (RAM\_EXECUTION)

/\*#define RAM\_EXECUTION (1) \*/

#endif /\* RAM\_EXECUTION \*/

取消掉这个注释就可以了。把它变成

/\* Uncomment the line below to enable the FLASH functions execution from RAM \*/

#if !defined (RAM\_EXECUTION)

#define RAM\_EXECUTION (1)

#endif /\* RAM\_EXECUTION \*/

这样IN\_RAM（）的定义就变成了

#define IN\_RAM(a) \_\_ramfunc a

再去调试程序。就好了。

自己写程序其实就是在函数前面加下 \_\_ramfunc

下面是我测试成功的程序

uint8\_t block\_buff[128];

\_\_ramfunc void WriteBlock(void)

{

uint8\_t i;

FLASH\_Unlock(FLASH\_MEMTYPE\_DATA);

FLASH->CR2 |= FLASH\_CR2\_PRG;

FLASH->NCR2 &= (uint8\_t)(~FLASH\_NCR2\_NPRG);

for(i=0; i<128; i++)

{

\*((PointerAttr uint8\_t\*) (MemoryAddressCast)0x4000 + i) = block\_buff[i];

}

FLASH\_Lock(FLASH\_MEMTYPE\_DATA);

}

祝还不会块编程的朋友们好运！！！！！

但是看门狗我还不知道原因。在接上STLINK调试的时候，看门狗会死机，但是在程序自己运行时，它就正常了