<http://blog.chinaunix.net/uid-27680183-id-3985913.html>

昨天调试我们新的rtu的时候用外部spi flash的时候出现的一些小问题，因为赶工赶的急，所以想到用f4内置1m的flash作为数据存储器来使用。  
首先在网上搜集了一些资料，也就有一篇资料比较好，我这里吧我看的连接的地址挂上，谢谢这位同学的无私，通过他的代码和stm32的数据  
手册我，吧函数做一下几个函数，希望对各位同学有帮助，<http://www.cnblogs.com/zyqgold/archive/2013/11/09/3416108.html>

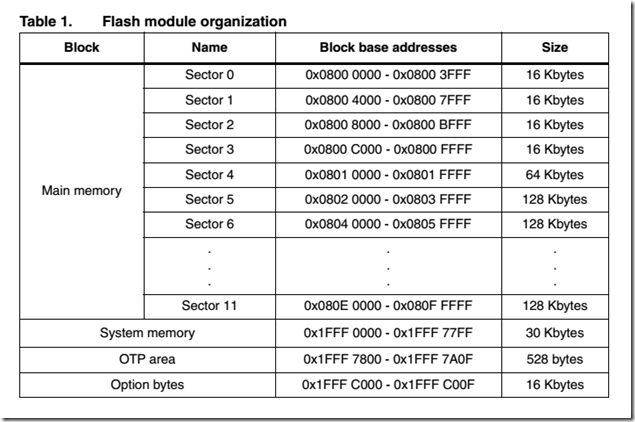
STM32F4Discovery开发帮使用的STM32F407VGT6芯片，内部FLASH有1M之多。平时写的代码，烧写完之后还有大量的剩余。有效利用这剩余的FLASH能存储不少数据。因此研究了一下STM32F4读写内部FLASH的一些操作。

【STM32F4 内部Flash的一些信息】

STM32F407VG的内部FLASH的地址是：0x08000000，大小是0x00100000。

写FLASH的时候，如果发现写入地址的FLASH没有被擦出，数据将不会写入。FLASH的擦除操作，只能按Sector进行。不能单独擦除一个地址上的数据。因此在写数据之前需要将地址所在Sector的所有数据擦除。

在STM32F4的编程手册上可找到FLASH的Sector划分，我们现在只操作Main memory：

[](http://images.cnitblog.com/blog/79656/201311/09212839-babf306dab1b46fdb89aa23d75ddab86.png)

参考Demo中的例子，将FLASH的页的其实地址（基地址）可定义如下：  
  
  
这里的flash的在编程钱如果这一个地址已经写过了请先擦除这个地址，因为现在stm32f4的的flash分为11个部分，每个擦除是按照每个部分一起擦除，因为这你的对每个快的做一个缓冲区才行。  
我的代码如下：  
void get\_from\_flash(void)  
{  
delay\_ms(2);  
OwnFlashReady(0x08008000,getflashdata\_buf,100);  
// SPI\_FLASH\_BufferRead(getflashdata\_buf, 0x000000, 100); //??????  
// SPI\_FLASH\_ChipErase();//????flash????????  
memmove(&RTU\_Basedata\_Only,&getflashdata\_buf[0],18);  
memmove(&BaseData01,&getflashdata\_buf[18],21);  
memmove(&gx\_options01[0],&getflashdata\_buf[39],60);   
}  
  
  
void OwnFlashSave(uint32\_t save\_addr,uint8\_t \*p,uint16\_t number)  
{  
uint32\_t StartSector,EndSector,i, save\_addr\_temp;  
StartSector = GetSector(save\_addr);  
EndSector = GetSector(save\_addr+number);  
save\_addr\_temp = save\_addr;  
FLASH\_Unlock();   
FLASH\_ClearFlag(FLASH\_FLAG\_EOP | FLASH\_FLAG\_OPERR | FLASH\_FLAG\_WRPERR |   
FLASH\_FLAG\_PGAERR | FLASH\_FLAG\_PGPERR|FLASH\_FLAG\_PGSERR);  
  
  
//????flash  
for (i = StartSector; i<=EndSector; i += 8)   
{  
while (FLASH\_EraseSector(i, VoltageRange\_3) != FLASH\_COMPLETE)  
{  
}  
} //????????????×÷?????????????????????í?ó??  
while (save\_addr < number+save\_addr\_temp)  
{  
if (FLASH\_ProgramByte(save\_addr, \*p++) == FLASH\_COMPLETE)   
{  
save\_addr = save\_addr + 1;  
}  
else  
{   
while (1)  
{  
}  
}  
}  
FLASH\_Lock();  
}  
  
  
// void OwnFlashErasure()  
  
  
void OwnFlashReady(uint32\_t read\_addr,uint8\_t \* read\_buf,uint16\_t read\_number)  
{  
uint32\_t Address\_ready;  
Address\_ready = read\_addr;  
while (Address\_ready < read\_addr+read\_number)  
{  
\*read\_buf++ = (\*(\_\_IO uint8\_t\*)Address\_ready);   
Address\_ready = Address\_ready + 1;  
}   
}  
  
  
uint32\_t GetSector(uint32\_t Address)  
{  
uint32\_t sector = 0;  
  
if((Address < ADDR\_FLASH\_SECTOR\_1) && (Address >= ADDR\_FLASH\_SECTOR\_0))  
{  
sector = FLASH\_Sector\_0;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_2) && (Address >= ADDR\_FLASH\_SECTOR\_1))  
{  
sector = FLASH\_Sector\_1;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_3) && (Address >= ADDR\_FLASH\_SECTOR\_2))  
{  
sector = FLASH\_Sector\_2;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_4) && (Address >= ADDR\_FLASH\_SECTOR\_3))  
{  
sector = FLASH\_Sector\_3;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_5) && (Address >= ADDR\_FLASH\_SECTOR\_4))  
{  
sector = FLASH\_Sector\_4;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_6) && (Address >= ADDR\_FLASH\_SECTOR\_5))  
{  
sector = FLASH\_Sector\_5;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_7) && (Address >= ADDR\_FLASH\_SECTOR\_6))  
{  
sector = FLASH\_Sector\_6;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_8) && (Address >= ADDR\_FLASH\_SECTOR\_7))  
{  
sector = FLASH\_Sector\_7;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_9) && (Address >= ADDR\_FLASH\_SECTOR\_8))  
{  
sector = FLASH\_Sector\_8;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_10) && (Address >= ADDR\_FLASH\_SECTOR\_9))  
{  
sector = FLASH\_Sector\_9;   
}  
else if((Address < ADDR\_FLASH\_SECTOR\_11) && (Address >= ADDR\_FLASH\_SECTOR\_10))  
{  
sector = FLASH\_Sector\_10;   
}  
else  
{  
sector = FLASH\_Sector\_11;   
}  
  
  
return sector;  
}

代码不是很复杂，但是如果没有弄懂的话还是要费些时间的，希望各位同学多多指教