利用STM32F103ZE微控制器I2C总线DMA读EEPROM一点经验

Stm32f103的i2c真是难学，连续学习了两个星期，终于有了一点进展。现将心得总结一下，供大家交流，不当的地方欢迎指正。

一、硬件平台：野火霸道开发板+野火仿真器

二、软件平台：MDK V5.14和野火多功能调试助手

三、利用资源：I2C和DMA，还有配合的USART

四、预期功能：利用stm32f103的硬件I2C1，向板载的AT24C02 EEPROM写入8个字节数据，然后通过I2C1外设的DMA功能将写入的数据读出，比较写入和读出的数据是否一致。

五、实现方法：具体的方法就不细说了，主要是参考野火的USART和I2C例程，实现I2C 轮询方式数据读写（I2C例程有时会出现超时，主要在busy位操作时，没有在意），然后参照STM32F10x固件库的I2C读写EEPROM例程，希望实现预期功能。

六、问题现象：1.仅I2C，不用dma时，不是每次都能够读写成功，进入debug模式，设置断点运行，停在I2C产生开始信号之前，busy=1；2.开启dma和中断时，不能运行，总是总线忙，busy=1，偶尔能够运行但是读出的数据仅有7位正确，8个字节的WD[0]始终无法正确读出。

七、解决过程：一句话，上网，上论坛，通过连续数天的查找，得出一个结论，大家普遍认为stm32f103的硬件I2C存在BUG，大家也进行了分析，推荐应用软件模拟I2C。由于个人水平有限，理解了个大概，但是无从下手。这个过程中，再次阅读了英文参考手册，感觉这里面的信息量太大了，一时半时没法完全吸收。还有就是，英文的表达方式和汉语的表达方式有差别，很多翻译的资料与英文原版的意思还是有一定差别的。于是，想到查找英文论坛资料，看看擅长电子设计的外国人有没有遇到这类问题，毕竟stm32是外国人设计的嘛。打开bing国际版，输入stm32 i2c busy等关键词，果然外国人也遇到了类似问题，见：<https://electronics.stackexchange.com/questions/267972/i2c-busy-flag-strange-behaviour/305954>。这里面提到了一个关键词，errata sheet，这篇sheet涉及的微控制器为STM32F100X系列，有可能不适用stm32f103。到st官网一查，stm32f10x果然也有相应的errata sheet（en.CD00197763.pdf），而且发布日期为2015年底，两篇文档中对于遇到的问题都有描述，2.14.7中有问题分析及workaround（理解为权宜之计吧），一共有15步。

八、解决例程：参照errata sheet中方法，编写相应解决busy=1的程序

void I2C\_ClearBusyFlagErratum(void)

{

       GPIO\_InitTypeDefGPIO\_InitStructure;

       I2C\_InitTypeDefI2C\_InitStructure;

       //step 1 Clear PE bit

       I2C\_Cmd(I2C1,DISABLE);

       //step 2 configure thescl and sda i/os as gpio output open drain.high level

       GPIO\_InitStructure.GPIO\_Mode=GPIO\_Mode\_Out\_OD;

       GPIO\_InitStructure.GPIO\_Pin=GPIO\_Pin\_6|GPIO\_Pin\_7;

       GPIO\_InitStructure.GPIO\_Speed=GPIO\_Speed\_50MHz;

       GPIO\_Init(GPIOB,&GPIO\_InitStructure);

       GPIO\_WriteBit(GPIOB,GPIO\_Pin\_6,Bit\_SET);

       GPIO\_WriteBit(GPIOB,GPIO\_Pin\_7,Bit\_SET);

       //step 3 check scl andsda high level in GPIO\_IDR Register

       while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_6)!=Bit\_SET);

       while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_7)!=Bit\_SET);

       //step 4 configure thesda io as gpio output open drain low level

       GPIO\_WriteBit(GPIOB,GPIO\_Pin\_6,Bit\_RESET);

       //step 5 check sda lowlevel

       while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_6)!=Bit\_RESET);

       //step 6 configure thescl io as gpio output open drain low level

       GPIO\_WriteBit(GPIOB,GPIO\_Pin\_7,Bit\_RESET);

       //step 7 check scl lowlevel

       while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_7)!=Bit\_RESET);

       //step 8 configure thescl io as gpio output open drain high  level

       GPIO\_WriteBit(GPIOB,GPIO\_Pin\_7,Bit\_SET);

       //step 9 check scl hihtlevel

       while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_7)!=Bit\_SET);

       //step10 configure thesda io as gpio output open drain high level

       GPIO\_WriteBit(GPIOB,GPIO\_Pin\_6,Bit\_SET);

       //step11 check sda highlevel

       while(GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_6)!=Bit\_SET);

       //step12 configure the scland sda ios as alternate function open drain

       GPIO\_InitStructure.GPIO\_Pin   = GPIO\_Pin\_6|GPIO\_Pin\_7;

       GPIO\_InitStructure.GPIO\_Speed= GPIO\_Speed\_50MHz;

       GPIO\_InitStructure.GPIO\_Mode  = GPIO\_Mode\_AF\_OD;

       GPIO\_Init(GPIOB,&GPIO\_InitStructure);

       //step13 set SWRST bitin CR1 register

       I2C\_SoftwareResetCmd(I2C1,ENABLE);

       //step14 clear SWRST bitin CR1 register

       I2C\_SoftwareResetCmd(I2C1,DISABLE);

       //step15 enable i2c bysetting PE

       I2C\_InitStructure.I2C\_Ack=I2C\_Ack\_Enable;

       I2C\_InitStructure.I2C\_AcknowledgedAddress=I2C\_AcknowledgedAddress\_7bit;

       I2C\_InitStructure.I2C\_ClockSpeed=200000;

       I2C\_InitStructure.I2C\_DutyCycle=I2C\_DutyCycle\_2;

       I2C\_InitStructure.I2C\_Mode=I2C\_Mode\_I2C;

       I2C\_InitStructure.I2C\_OwnAddress1=0x5F;//i2c×&Ocirc;&Eacute;í&micro;&Oslash;&Ouml;·&pound;&raquo;

    I2C\_Init(I2C1,&I2C\_InitStructure);

       I2C\_Cmd(I2C1,ENABLE);

       I2C\_DMALastTransferCmd(I2C1,ENABLE);

}

完整例程附后。

关于8个字节数据之后发送7个问题，有待进一步学习，但是通过多次实验，应该是传输速度匹配的问题，在读**eeprom**之后加一定延迟，居然解决这个问题。

以上为个人的一点总结和心得，不对的地方大家及时纠正。