FRAM 与 MSP-EXP430FR5739 实验板初体验

本文由 Ken Lee 独立整理完成，首发于 QQ 空间，后期会在 EDN 网站 newofcortexm3 博客与其他网站、论坛更新，转载请注明作者和出处。作者水平有限，难免存在错误，不足之处请指正。另：由于个人原因，原署名斌斌-龙臻和 Mr_D_prince 不再使用，由 Ken Lee 和 newofcortexm3 替代。

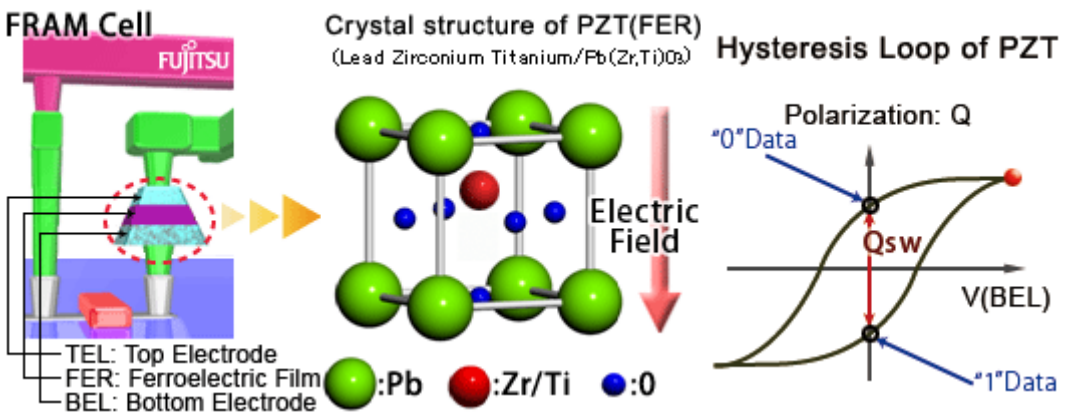
这是久违了的技术文章。

一、 FRAM

什么是 FRAM？FRAM 是指铁电随机存取存储器 （Ferroelectric Random AccessMemory）。这是最新的非易失性存储器技术，它将 SRAM 的速度、写入寿命及低功耗与闪存存储能力相结合，将传统器件上的两个存储器替换为一个统一的代码和数据存储空间。FRAM 的功耗极低，在 1.5V 下即可进行编程，而不是像闪存那样需要 10-14V。因此器件上无电荷泵。这也意味着在 FRAM 上编程无需擦除周期，具备比闪存速度更快、功耗更低的存储优势。

铁电存储技术最早是在 1921 年提出的，但直到 1993 年，美国 Ramtron 国际公司才成功开发出第一个 4K 位的铁电存储器 FRAM 产品，目前所有的 FRAM 产品均由 Ramtron 公司制造或授权。最近，FRAM 又有新的发展，采用了 0.35 m 工艺，推出了 3V 产品，开发出“单管单容”存储单元的 FRAM，最大密度可达 256K 位。

通常用作典型的铁电质材料是 PZT (Lead zirconate titanate-锆钛酸铅)。下面的图表（1）解释了 PZT 晶体结构，在点阵中具有锆和钛，作为两个稳定点。它们可以根据外部电场在两个点之间移动。一旦位置设定，即使再出现电场，它也将不会再有任何移动。顶部和底部的电极安排了一个电容器。那么，电容器划分了底部电极电压和极化，超越了磁滞回线。数据以“1”或“0”的形式存储。



图（1）：PZT 晶体结构和 FRAM 工作原理

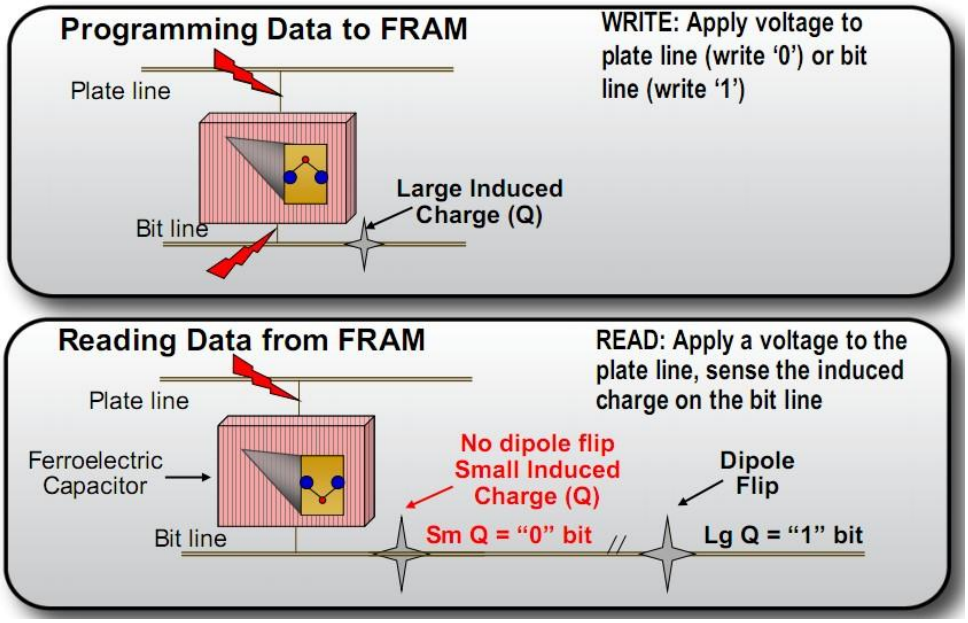
1. 当加置磁场时就会产生极化。（锆/钛离子在晶体中向上或向下移动）

2. 即使在不加置磁场的情况下，也能保持电极。

3. 两个稳定的状态以“0”或“1”的形式存储。

一般说来，这种铁电材料的数据存储性能取决于 PZT 中心原子的极化。由于在晶体中浮动的中心原子实际上能够在 1 毫微秒内达到稳定状态，因此这种极化活动能够以极低的功耗非常快速地出现。其性质非常像 DRAM，其中各个单元均采用单晶体管和单电容器。这意味着 FRAM 内存单元所需面积减少很多，且可采用处理器架构进行类似处理。但不同的是，其如 DRAM 中的电荷一样，不可能泄漏分子极化，在没有电场时状态保持不变。这使得 FRAM 为非易失性，即断电时，FRAM 上的数据不会丢失。

单个 FRAM 单元可被视为一个偶极电容器(dipolecapacitor)，它由两个电极板和之间的铁电材料薄膜构成。存储“1”或“0”时（写入 FRAM 时），只需中心原子按电场方向移动极化。这使得 FRAM 速度非常快，易于写入，能够充分满足写入寿命的要求。类似于 FRAM 写入，从 FRAM 读取也需要对铁电材料电容器施加一个电场。根据中心原子所处状态，重新极化，从而引起出一个大的电荷击穿。这个电荷随即与已知参数进行对比，从而判断中心原子的状态。存储的数据位“1”或“0”从感应电荷推断得出。在读取数据的过程中，按电场方向移动的中心原子会失去当前的状态。因此每次读取后，都需要进行写回操作，恢复存储单元的状态。读写过程如下图（2）所示。



图（2）：FRAM 读/写过程

常见的存储器主要有 EEPROM，SRAM 等，其分类如图（3）所示，与传统存储器相比（对比见图（4），图（5）），FRAM 具有下列优势：

非易失性

- 即使没有上电，也可以保存所存储的信息。
- 与 SRAM 相比，无需电池（环保产品）

更高速度写入

- 像 SRAM 一样，可改写

不要求改写命令

- 对于擦/写操作，无等待时间

写入循环时间 =读取循环时间

写入时间： EEPROM 的 1/30,000

具有更高的耐受力

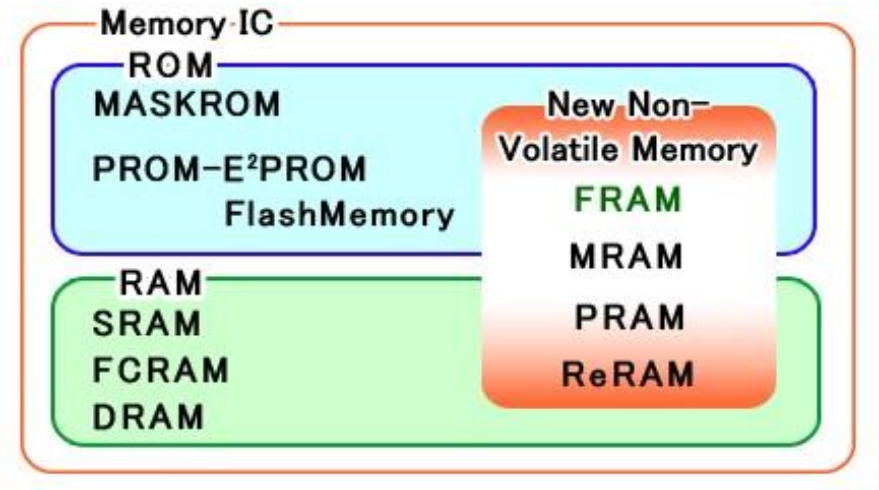
- 确保最大 10^{12} 循环（100 万兆循环）/位的耐久力

耐久力：超过 100 万次的 EEPROM

具有更低的功耗

- 不要求采用充电泵电路

功耗：低于 1/400 的 E^2PROM



图（3）：存储器分类

	FRAM	SRAM	EEPROM	Flash
Non-volatile Retains data without power	Yes	No	Yes	Yes
Write speeds	10ms	<10ms	2secs	1 sec
Average active Power [μA/MHz]	110	<60	50mA+	230
Write endurance	100 Trillion+	Unlimited	100,000	10,000
Dynamic Bit-wise programmable	Yes	Yes	No	No
Unified memory Flexible code and data partitioning	Yes	No	No	No

图（4）：对比 1

	FRAM	E^2PROM	Flash	SRAM
存储器类别	非易失性	非易失性	非易失性	易失性
晶胞结构*1	1T1C/2T2C	2T	1T	6T
数据改写方法	改写	擦除+写入	扇面擦除+写入	改写
写入循环时间	150ns*2	5ms	10μs	55ns
耐久力	最大 10^{12}（1万兆次循环*3）*2	10^6（100万次循环）	10^5（10万次循环）	无限制
写入操作电流	5mA（典型值）*2 15mA（最大值）*2	5mA（最大值）	20mA（最大值）	8mA（典型值） -
待机电流	5μA（典型值）*2 50μA（最大值）*2	2μA（最大值）	100μA（最大值）	0.7μA（典型值） 3μA（最大值）

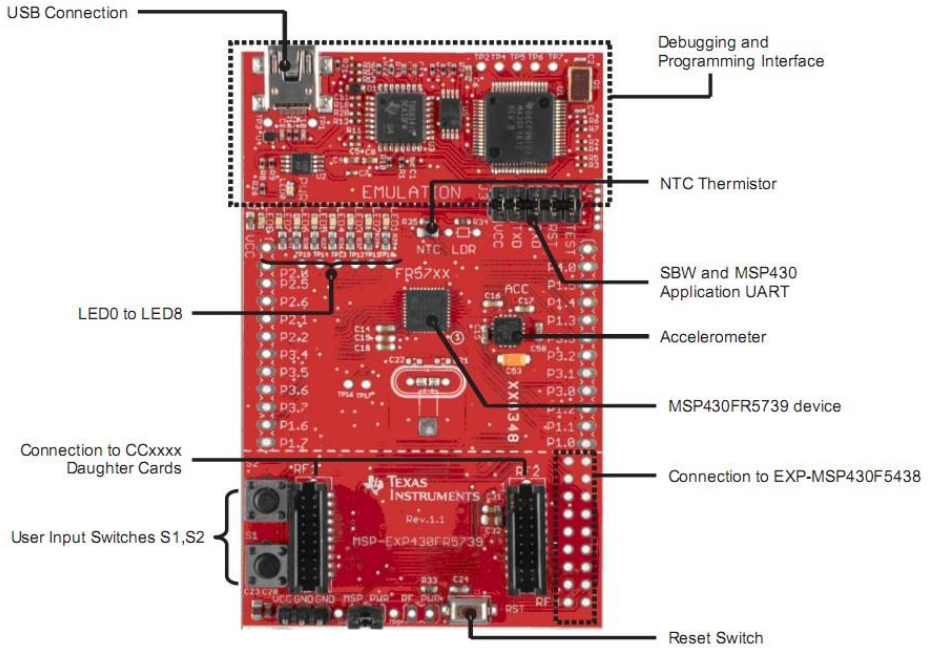
*1) T=晶体管. C=电容器
*2) 256Kb独立的FRAM存储器的技术规格
*3) 读写操作的总循环

图（5）：对比 2

注：图（1），（3），（5）引自富士通半导体，图（2），（4）来自德州仪器的资料。

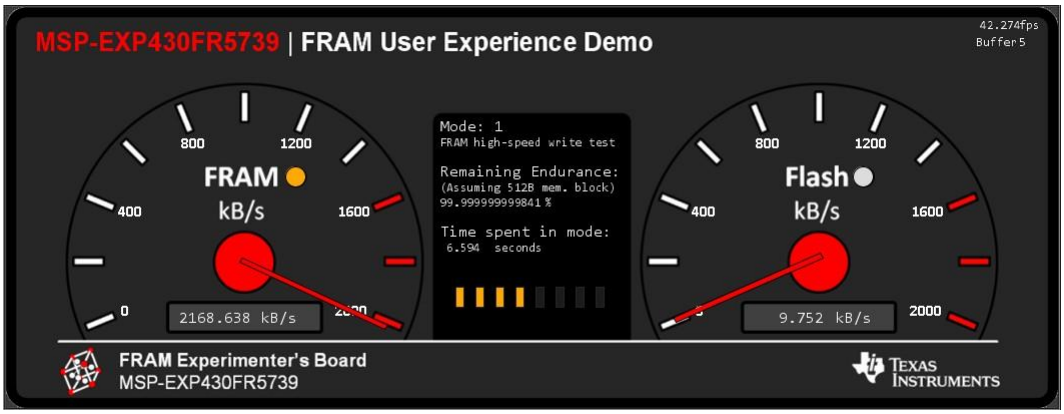
二、 MSP-EXP430FR5739 实验板

去年就购买了一块铁电开发板，因为时间关系一直没细细把玩。 MSP-EXP430FR5739 实验板支持新一代采用集成铁电随机存取存储器 (FRAM) 的 MSP430 微处理器设备，板上资源如图（6）所示：



图（6）：MSP-EXP430FR5739 实验板

出厂附带的 demo 包含了测试 FRAM 特性的 4 种模式：模式 1 - 写入速度最大化，模式 2 - 闪存写入速度仿真，模式 3 - 通过写入进行快速采样（使用加速计），模式 4 - 通过写入进行快速采样（使用热敏电阻）。其操作方式很简单，上电后按一次 S1，LED0 亮，再按 S2，进入 FRAM 高速写入速度测试模式。上电后按两次 S1， LED0~1 亮，再按 S2，进入模拟 FLASH 写入速度测试模式。上电后按三次 S1，LED0~2 亮，再按 S2，进入三轴加速度计测试模式。上电后按四次 S1，LED0~3 亮，再按 S2，进入温度写入测试模式。用户示例程序中附带了用 Java 开发的上位机程序，在四个模式中均可看到往 FRAM 中写数据的速度。如图（7）所示，左侧仪表盘表示了当前模式 1 下的，FRAM 的写数据速度，右侧仪表盘显示是模式 2 下的模拟 Flash 的写入速度。



图（7）：示例中的上位机

浏览了下位机的代码，发现其逻辑还是很简单的。其实现过程如下：

1. 处理器和外设初始化，此时可观察到板上的 LED 灯闪烁几个周期。LED 的电路设计见手册（slau343）17 页，或者见硬件设计压缩包（slac502），LED 的控制见函数 StartUpSequence，这个功能的实现与 Lab1 的实现基本一致。

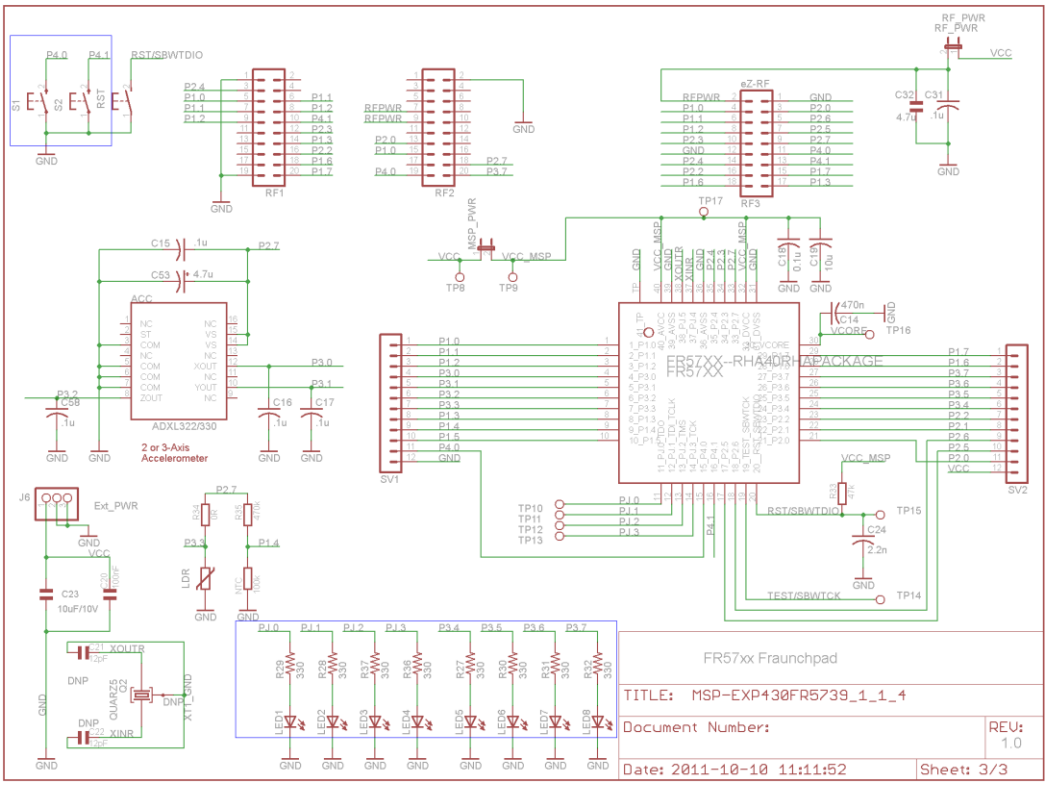
2. 系统进入死循环，等待按键 S1，S2 中断。S1，S2 分别与 P4.0，P4.1 口相连。其中断服务程序是__interrupt void Port_4(void)函数。使能与禁止中断函数分别为 EnableSwitches、DisableSwitches。在进行调用之前必须对其初始化，这个过程在上一步中完成。中断配置如下：

```
P4OUT |= BIT0 +BIT1;            // Configure pullup resistor
P4DIR &= ~(BIT0 + BIT1);        // Direction = input
P4REN |= BIT0 + BIT1;           // Enable pullup resistor
P4IES &= ~(BIT0+BIT1);          // P4.0 Lo/Hi edge interrupt
P4IE = BIT0+BIT1;               // P4.0 interrupt enabled
P4IFG = 0;                      // P4 IFG cleared
```

3. 程序根据按键 S1 的次数进入不同的模式。每个模式执行的逻辑基本类似，都要向 FRAM 写数据，控制 LED 输出，向上位机发送数据。不过模式 2，模式 3 和模式 4 在写数据之前都得完成一件任务。模式 2 得先切换时钟，从而模拟出向 Flash 写数据的时钟频率。模式 3 和模式 4 则先配置加速度传感器和温度传感器的端口，然后进行 AD 转换。

这个过程中，向 FRAM 写数据由函数 FRAM_Write 完成，如果起始地址在可利用的范围之内，该函数向存储区写入 0x12345678，共 512 个字节。LEDSequenceWrite 函数控制 LED，分别对 P3、PJ 口输出高/低电平。向上位机发送数据的任务则交给了函数 TXData。该函数发送的数据包以 0xFA 开始，以 0xFE 结束。第二位数据和第三位数据分别告诉上位机当前处于什么模式，有多少个 LED 灯点亮。利用串口调试工具监视串口，模式 1 下，下位机发送的数据首个数据为 0xFA 0x01 0x00 0x00 0x00 0x00 0xFE，第三位数据位发生变化，从 0x00 递升至 0x07，然后再次从 0x00 递升，表示了 8 个 LED 的周期性循环。模式 2 接收到的数据除了第二位是 0x02 外，变化基本一致，不过变化的速率明显下降了许多。以此对比突出 FRAM 的写入速度。模式 3 和模式 4 的第二位数据分别是 0x03，0x04，第三位数据相同，是 0x34。当程序从当前模式推出时，上位机收到的数据为 0xFA 0x55 0x55 0x55 0x55 0x04 0xFE。

其他程序的细节就不再累述了。上文中提到的 LED 和按键 S1、S2 的电路部分请见图（8），或者参考德州仪器提供的原理图。



图（8）：实验板核心部分原理图

对于 Lab2，原先的工程在主程序部分是需要我们自己添加的。当然 TI FAE 也提供了解决方法，如下文的代码中加粗斜体所示，详请可参考 Lab2_solution.c。

MPUCTL0 = ***MPUPW***; // Write PWD to access MPU registers

MPUSEG = ***0x0804***; // B1 = 0xC800; B2 = 0xD000

// Borders are assigned to segments

MPUSAM &= ~***MPUSEG2WE***; // Segment 2 is protected from write

MPUSAM |= ***MPUSEG2VS***; // Violation select on write access

MPUCTL0 = ***MPUPW+MPUENA+MPUSEGIE+MPULOCK***; // Enable NMI & MPU protection

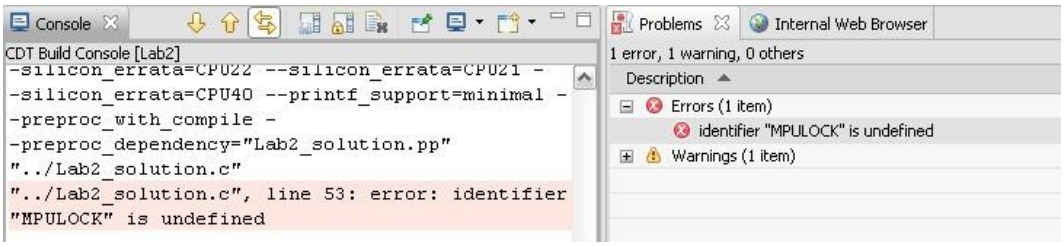
.

.

.

ptr = (unsigned int *)***0xC802***;

但是将 Lab2.c 中需补充的部分按如上所述修改，或者将 Lab2.c Exclude from build，将 Lab2_solution.c Include Build 之后依然存在编译没法通过的问题，如图（9）所示。



图（9）：Lab2 编译问题

通过查手册 slau272（MSP430FR57xx Family Users Guide），了解到 MPULOCK 在寄存器 Memory Protection Unit Control 0 (MPUCTL0)中，具体位置见图（10）。因此只要在头文件 msp430fr5739.h 中添加 #define MPULOCK (0x0002) 即可解决编译问题。当然为什么 TI FAE 没添加这句话，是否有其他的考虑，我还在思索中。

Memory Protection Unit Control 0 (MPUCTL0)

15	14	13	12	11	10	9	8
MPUPW, Read as 096h Must be written as 0A5h							
7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MPULOCK	MPUENA
r-0	r-0	r-0	rw-[0]	r-0	r-0	rw[0]	rw-[0]

MPUPW	Bits 15–8	MPU password. Always read as 096h. Must be written with 0A5h or a PUC is generated on word write. After a correct password is written, all MPU registers are accessible. An incorrect password written in byte mode disables MPU register access and no PUC is generated.
Reserved	Bit 7-5	Reserved. Always read 0.
Reserved	BIt4	Reserved for future usage. Must always be written with 0.
Reserved	Bit 3-2	Reserved. Always read 0.
MPULOCK	Bit 1	MPU lock. If this bit is set, access to all MPU registers except MPUCTL1 are locked and are read only until a BOR occurs. BOR automatically sets this bit to 0. 0 Open 1 Locked
MPUENA	Bit 0	MPU enable. This bit enables the MPU operation. This bit can be set any time with word write and the correct password, if MPULOCK is not already set 0 Disabled 1 Enabled

图（10）：MPUCTL0 寄存器描述

三、 总结

那么 FRAM 写入速度究竟有多快？这取决于应用程序。了解 FRAM 访问速度的关键是了解系统级限制。 通常，起限制作用的不是 FRAM 访问速度本身，而是通过有线或无线外围通讯设施写入 FRAM 时的数据处理开销或协议开销。变换影响访问速度的三种常见因素： DMA 的使用， 系统速度，数据块大小。但是对比于传统的存储器，其写入的快速性已经有了质的飞跃。另外，即使在 85 摄氏度下未启用状态下，FRAM 的写入寿命至少 10 年。

MSP-EXP430FR5739 实验板内部集成了 FRAM 存储器，附带的 demo 显著的论证了 FRAM 的快速性。其板上资源的丰富和低功耗的特性为不少应用提供了解决方案。也是一款优秀的，足以让用户了解 FRAM 特性的入门级实验板。