



复旦微电子

FM3316/FM3313/FM3312

低功耗系列 MCU

应用笔记

中断使用说明

AN0005

V1.0



本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。

在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。

采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。

未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站(<http://www.fmsh.com/>)。

如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。

商 标

上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。



目录

| | |
|-------------------------------|-----------|
| 1 说明 | 5 |
| 2 原理 | 6 |
| 2.1 处理过程 | 6 |
| 2.2 中断使能控制 | 6 |
| 2.3 全局中断控制 | 6 |
| 2.4 内核中断控制 | 6 |
| 2.5 外设中断控制 | 7 |
| 2.6 中断优先级 | 7 |
| 2.7 中断编程注意 | 10 |
| 2.8 中断编程常见错误 | 10 |
| 3 实现方法 | 12 |
| 3.1 参考例程使用说明 | 12 |
| 4 建议的实现步骤 | 13 |
| 版本信息 | 14 |
| 上海复旦微电子集团股份有限公司销售及服务网点 | 15 |

表格图片目录

| | |
|--------------------------------------|----|
| 图表 1 : 中断使能寄存器 IE | 6 |
| 图表 2 : 附加中断使能寄存器 AIE | 7 |
| 图表 3 : RTC 模块中断寄存器 | 7 |
| 图表 4 : 中断优先级寄存器 IPH0/IPL0 | 8 |
| 图表 5 : SFR IPH0 (S:0B7H) 寄存器 | 8 |
| 图表 6 : SFR IPL0 (S:0B8H) 寄存器 | 8 |
| 图表 7 : SFR AIPH (S:0F7H) 寄存器 | 8 |
| 图表 8 : SFR AIPL (S:0F8H) 寄存器 | 9 |
| 图表 9 : 中断优先级 | 9 |
| | |
| 图 1 : UART 中断 | 12 |
| 图 2 : 内核中断 | 12 |
| 图 3 : 全局中断 | 12 |
| 图 4 : 中断处理函数 | 12 |
| 图 5 : 实现步骤 | 13 |



1 说明

本文档为 FM3316/FM3313/FM3312 系列低功耗 MCU 的应用笔记，用于说明芯片中断的使用说明。FM3316/FM3313/FM3312 系列是复旦微电子公司开发的低功耗 MCU 芯片，请联系复旦微电子公司提供更多相关文档支持设计开发。

2 原理

2.1 处理过程

FM3316 的中断系统类似标准 8xC251，但是延迟周期更短，中断典型处理过程如下：

- 中断请求信号产生。
- 中断信号被 CPU 中断系统采样，并寄存到标志缓冲区。
- 中断系统比较中断标志的优先级，为高优先级的中断置起中断标志。
- 中断标志置起告知控制单元中断当前指令执行流程，控制单元在完成当前指令执行前保存 3bytesPC 和 PSW1 寄存器(S:0D1h)，并将中断向量地址装载到 PC。
- 软件服务程序执行中断任务，以 RETI 指令作为结束，重载 PC 和 PSW1 寄存器值，程序从中断前的位置继续执行。

2.2 中断使能控制

FM3316 的中断使能控制可分为三级：

第一级：全局中断控制；

第二级：内核中断控制；

第三级：外设中断控制。

只有当三级中断全部使能时，中断功能才可正常工作。

2.3 全局中断控制

全局中断控制由 CPU 的 SFR 寄存器 IE 的 bit7 控制。

EA = 0; //关闭全局中断；

EA = 1; //打开全局中断。

当 EA=0，除两个不可屏蔽中断 TRAP 和 NMI 外所有的中断都被屏蔽。当 EA=1 时，除 TRAP 和 NMI 中断外，其它中断的使能由它们各自的使能位决定。

2.4 内核中断控制

内核中断控制由 CPU 的 SFR 寄存器 IE 的 bit0~bit6 和 AIE 的 bit0~bit6 控制。每个 bit 控制一个中断的中断使能，具体见下表。

图表 1：中断使能寄存器 IE

| Bit | 助记符 | 功能描述 |
|-----|-----|-------------------------------------|
| 7 | EA | 全局中断使能,不影响 TRAP 和 NMI 中断（这两个中断不可屏蔽） |
| 6 | EC | PCA 中断使能 |
| 5 | ET2 | 内核定时器 2 中断使能 |

| Bit | 助记符 | 功能描述 |
|-----|-----|---|
| 4 | ES | 保留 |
| 3 | ET1 | 内核定时器 1 中断使能 |
| 2 | EX1 | 外部中断 1 (int1) 使能 (GPIO 引脚中断) |
| 1 | ET0 | 内核定时器 0 中断使能 |
| 0 | EX0 | 外部中断 0 (int0) 使能 (RTC、LTBC、LPTIM、LVD) |

图表 2：附加中断使能寄存器 AIE

| Bit | 助记符 | 功能描述 |
|-----|------|----------------------------|
| 7 | -- | 未实现：读为 0 |
| 6 | AIE6 | 附加中断 6 使能 (DMA) |
| 5 | AIE5 | 附加中断 5 使能 (Flash 控制器) |
| 4 | AIE4 | 附加中断 4 使能 (DISP) |
| 3 | AIE3 | 附加中断 3 使能 (SPI/I2C/7816) |
| 2 | AIE2 | 附加中断 2 使能 (UART0/1/2/3) |
| 1 | AIE1 | 附加中断 1 使能 (ADC) |
| 0 | AIE0 | 附加中断 0 使能 (ET1/2/3/4) |

2.5 外设中断控制

外设的中断控制由各个外设模块内的寄存器实现，以 RTC 模块的秒中断为例，只需将 RTCIE2 的 SEC_IE (bit4) 写 1 即可使能外设中断控制，其他的外设中断控制也可以此类推。

图表 3：RTC 模块中断寄存器

| 名称 | RTCIE2 | | | | | | | |
|-----|---------|-------------|--------|--------|--------|--------|--------|-------------|
| 地址 | 01:00A3 | | | | | | | |
| 位 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 位名 | -- | HOUR_I E | MIN_IE | SEC_IE | 2HZ_IE | 4HZ_IE | 8HZ_IE | 16HZ_ IE |
| 位权限 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W- 0 |

2.6 中断优先级

FM3316 的中断优先级支持四级可配置，每个中断使用两个比特控制优先级，由 C251 内核的 IPH0/IPL0，AIPH/AIPL 四个寄存器控制。以 PCA 中断为例，PCA 中断的优先级由 IPH0 的 IPHC (bit6) 和 IPL0 的 IPLC (bit6)，两个比特控制，如下表所示，其他中断的优先级可以此类推。

图表 4：中断优先级寄存器 IPH0/IPL0

| IPHC | IPLC | 中断优先级 | 中断优先级说明 |
|------|------|-------|---------|
| 0 | 0 | 0 | 最低优先级 |
| 0 | 1 | 1 | 次低优先级 |
| 1 | 0 | 2 | 次高优先级 |
| 1 | 1 | 3 | 最高优先级 |

图表 5：SFR IPH0 (S:0B7H) 寄存器

| Bit | 助记符 | 功能描述 |
|-----|-------|-----------------------|
| 7 | -- | 未实现：读为 0 |
| 6 | IPHC | PCA 中断优先级高位 |
| 5 | IPHT2 | 定时器 2 中断优先级高位 |
| 4 | IPHS | 串口中断优先级高位 |
| 3 | IPHT1 | 定时器 1 中断优先级高位 |
| 2 | IPHX1 | 外部中断 1 (int1) 优先级高位 |
| 1 | IPHT0 | 定时器 0 中断优先级高位 |
| 0 | IPHX0 | 外部中断 0 (int0) 优先级高位 |

图表 6：SFR IPL0 (S:0B8H) 寄存器

| Bit | 助记符 | 功能描述 |
|-----|-------|-----------------------|
| 7 | -- | 未实现：读为 0 |
| 6 | IPLC | PCA 中断优先级低位 |
| 5 | IPLT2 | 定时器 2 中断优先级低位 |
| 4 | IPLS | 串口中断优先级低位 |
| 3 | IPLT1 | 定时器 1 中断优先级低位 |
| 2 | IPLX1 | 外部中断 1 (int1) 优先级低位 |
| 1 | IPLT0 | 定时器 0 中断优先级低位 |
| 0 | IPLX0 | 外部中断 0 (int0) 优先级低位 |

附加中断优先级寄存器 AIPH/AIPL。

图表 7：SFR AIPH (S:0F7H) 寄存器

| Bit | 助记符 | 功能描述 |
|-----|-------|--------------|
| 7 | -- | 未实现：读为 0 |
| 6 | AIPH6 | 附加中断 6 优先级高位 |
| 5 | AIPH5 | 附加中断 5 优先级高位 |
| 4 | AIPH4 | 附加中断 4 优先级高位 |

| Bit | 助记符 | 功能描述 |
|-----|-------|--------------|
| 3 | AIPH3 | 附加中断 3 优先级高位 |
| 2 | AIPH2 | 附加中断 2 优先级高位 |
| 1 | AIPH1 | 附加中断 1 优先级高位 |
| 0 | AIPH0 | 附加中断 0 优先级高位 |

图表 8 : SFR AIPL (S:0F8H) 寄存器

| Bit | 助记符 | 功能描述 |
|-----|-------|--------------|
| 7 | -- | 未实现：读为 0 |
| 6 | AIPL6 | 附加中断 6 优先级低位 |
| 5 | AIPL5 | 附加中断 5 优先级低位 |
| 4 | AIPL4 | 附加中断 4 优先级低位 |
| 3 | AIPL3 | 附加中断 3 优先级低位 |
| 2 | AIPL2 | 附加中断 2 优先级低位 |
| 1 | AIPL1 | 附加中断 1 优先级低位 |
| 0 | AIPL0 | 附加中断 0 优先级低位 |

除 TRAP 和 NMI 中断外，每一个中断源都可编程为高优先级或低优先级，处于同一优先级的两个中断不会相互嵌套，但是一个中断服务程序可以被一个更高优先级的中断所中断。如果有两个或两个以上的中断请求源都被编程为同一优先级，则它们的优先级由下表中的一般优先级来控制，数字越小，一般优先级越高。一般优先级仅适用于两个处于同一优先级的中断同时到达时，中断系统最终仲裁判断的依据，最终仅会有一个中断服务程序被触发（一般优先级高的那个）。

图表 9 : 中断优先级

| 中断源 | 中断描述 | 一般优先级 | 中断入口地址 | 使能位 | 优先级控制位 | 标志位 | 标志位清除 |
|-------|-----------|-------|----------|--------|--------------------------|----------|------------------|
| TRAP | TRAP 指令中断 | 1 | FF:007BH | -- | -- | -- | -- |
| NMI | NMI 中断 | 2 | FF:003BH | -- | -- | -- | -- |
| Int_0 | 外部中断 0 | 3 | FF:0003H | IE.EX0 | IPH0.IPHX0 IPL0.IPLX0 | TCON.IE0 | 边沿触发为硬件清除，否则软件清除 |
| T/C0 | 定时器 0 中断 | 4 | FF:000BH | IE.ET0 | IPH0.IPHT0 IPL0.IPLT0 | TCON.TF0 | 硬件清除 |
| Int_1 | 外部中断 1 | 5 | FF:0013H | IE.EX1 | IPH0.IPHX1 IPL0.IPLX1 | TCON.IE1 | 边沿触发为硬件清除，否则软件清除 |
| T/C1 | 定时器 1 中 | 6 | FF:001BH | IE.ET1 | IPH0.IPHT1 | TCON.TF1 | 硬件清除 |

| 中断源 | 中断描述 | 一般优先级 | 中断入口地址 | 使能位 | 优先级控制位 | 标志位 | 标志位清除 |
|-------------|----------|-------|----------|----------|--------------------------|-------------------------|-------|
| | 断 | | | | IPL0.IPLT1 | | |
| T/C2 | 定时器 2 中断 | 8 | FF:002BH | IE.ET2 | IPH0.IPHT2 IPL0.IPLT2 | T2CON.TF2 T2CON.EXF2 | 软件清除 |
| PCA | PCA 中断 | 9 | FF:0033H | IE.EC | IPH0.IPHC IPL0.IPLC | CCON.CF CCON.CCFx | 软件清除 |
| Intextra_n0 | 附加中断 0 | 10 | FF:0043H | AIE.AIE0 | AIPH.AIPH0 AIPL.AIPL0 | AIF.AIF0 | 软件清除 |
| Intextra_n1 | 附加中断 1 | 11 | FF:004BH | AIE.AIE1 | AIPH.AIPH1 AIPL.AIPL1 | AIF.AIF1 | 软件清除 |
| Intextra_n2 | 附加中断 2 | 12 | FF:0053H | AIE.AIE2 | AIPH.AIPH2 AIPL.AIPL2 | AIF.AIF2 | 软件清除 |
| Intextra_n3 | 附加中断 3 | 13 | FF:005BH | AIE.AIE3 | AIPH.AIPH3 AIPL.AIPL3 | AIF.AIF3 | 软件清除 |
| Intextra_n4 | 附加中断 4 | 14 | FF:0063H | AIE.AIE4 | AIPH.AIPH4 AIPL.AIPL4 | AIF.AIF4 | 软件清除 |
| Intextra_n5 | 附加中断 5 | 15 | FF:006BH | AIE.AIE5 | AIPH.AIPH5 AIPL.AIPL5 | AIF.AIF5 | 软件清除 |
| Intextra_n6 | 附加中断 6 | 16 | FF:0073H | AIE.AIE6 | AIPH.AIPH6 AIPL.AIPL6 | AIF.AIF6 | 软件清除 |

2.7 中断编程注意

FM3316 所使用的 C251 内核的寄存器组共有四组，由特殊功能寄存器 PSW 里的 RS1、RS0 位决定当前使用的寄存器组。在编写中断服务程序时，可以使用 using x 来指定当前中断服务程序运行时使用那组寄存器。

- using 0 时设置 RS1=0，RS0 =0，用第 0 组寄存器（BANK0）
- using 1 时设置 RS1=0，RS0 =1，用第 1 组寄存器（BANK1）
- using 2 时设置 RS1=1，RS0 =0，用第 2 组寄存器（BANK2）
- using 3 时设置 RS1=1，RS0 =1，用第 3 组寄存器（BANK3）

在进入中断前，通过切换寄存器组，可以方便的保护主程序被打断处寄存器组的数据不被中断里的语句破坏，但是不正确的使用寄存器组也会给程序带来隐患。

2.8 中断编程常见错误

- 为中断服务程序指定了第 0 组寄存器

例如：void int0_int(void) interrupt 0 using 0

因为复位后默认选择 BANK0，所以 main()函数使用的就是第 0 组寄存器，如果中断服务程序也使用 BANK0，

中断服务程序会改变寄存器组中的数据。中断返回后，主程序恢复运行时，寄存器 R0-R7 的值已经被中断服务程序改变，导致程序运行异常。

- 为不同优先级的中断服务程序指定了同一组寄存器

例如：

```
void int0_int( void ) interrupt 0 using 1    //低优先级中断
```

```
void int1_int( void ) interrupt 2 using 1    //高优先级中断
```

当高优先级的中断嵌套正在使用 BANK1 运行的低优先级中断时，与前一种错误一样，高优先级中断的服务程序会修改 BANK1 中的数据，低优先级中断恢复运行时数据已经被破坏，导致程序运行异常。对于不同优先级的中断服务程序，需要指定不同的寄存器组。

- 不使用 using x 指定寄存器组

例如：void int0_int(void) interrupt 0

没有用 using 指定寄存器组，编译器会默认分配第 0 组寄存器，进入中断前会额外压栈保护第 0 组寄存器的 R0~R7 中会被中断服务程序破坏的寄存器，中断返回前再恢复，虽然不会导致程序运行异常，但是会降低运行效率。

- 在中断服务程序和主程序中调用同一个函数

Keil 的 C51/C251 编译器编译得到的函数天生就是不可重入的，不能被多个程序同时调用，所以编写中断服务程序时不可调用主程序中的函数，以免数据被破坏，导致程序运行异常。当需要同时调用时，要将这个函数申明为可重入函数。C51 编译器采用了一个扩展关键字 reentrant 作为定义函数时的选项，需要将一个函数定义为可重入函数时，只要在函数后面加上关键字 reentrant 即可。对于简单函数也可直接复制一个实现相同功能的不同名函数分别给主程序和中断服务程序调用。

3 实现方法

3.1 参考例程使用说明

以 uart 中断举例：

- uart 初始化后，打开 uart 外设中断

```
UART_Init(UART2, &init);
UART_TX_Interrupt_Enable(UART2);
UART_RX_Interrupt_Enable(UART2);
UART_TX_Enable(UART2);
UART_RX_Enable(UART2);
```

图 1：UART 中断

- 打开 uart 外设挂靠的内核中断

```
AIF2 = 0;           //中断标志清0
AIE2 = 1;           //打开中断允许
EA = 1;
```

图 2：内核中断

- 打开全局中断

```
AIF2 = 0;           //中断标志清0
AIE2 = 1;           //打开中断允许
EA = 1;
```

图 3：全局中断

- 中断处理函数

```
//UART0/1/2/3
void Eint2_int (void) interrupt 10 using 1
{
    unsigned char Temp_IF, Temp;
    AIF2 = 0;

    Temp_IF = UARTIE&UARTIF; //任何对寄存器的操作都会导致其清零，必须先缓存
    if(Temp_IF)
    {
        //UART0
        if( (Temp_IF&B0000_0010) ) //接收
        {
            Temp = RXREG0; //从UART0数据寄存器中取接收数据
            //用户中断代码
            if(RXSTA0&B0000_1110) //出错
            {
                RXSTA0&=~B0000_1110;
            }
            TXREG0 = Temp;
        }
        if( (Temp_IF&B0000_0001) ) //发送
        {
            //用户中断代码
            UARTIF = B1111_1110;
        }
    }
}
```

图 4：中断处理函数

4 建议的实现步骤

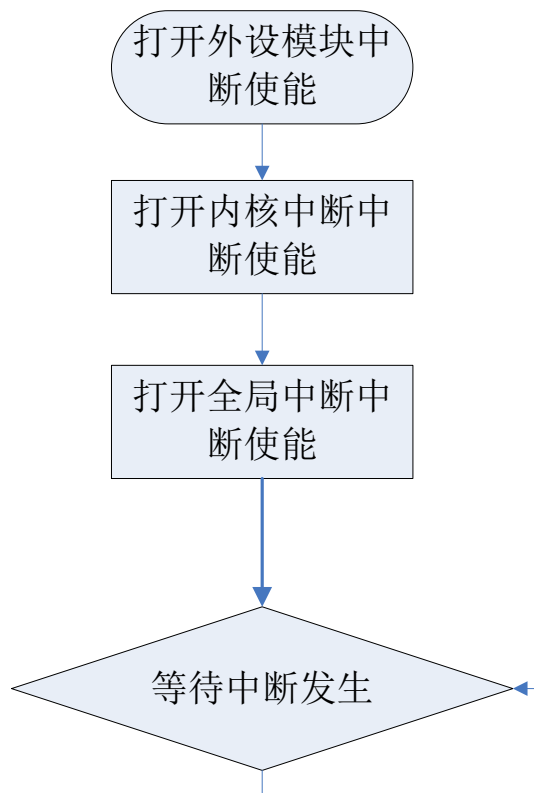


图 5：实现步骤



版本信息

| 版本号 | 发布日期 | 更改说明 |
|-----|---------|------|
| 1.0 | 2018.05 | 首次发布 |



上海复旦微电子集团股份有限公司销售及服务中心

上海复旦微电子集团股份有限公司

地址：上海市国泰路 127 号 4 号楼

邮编：200433

电话：(86-021) 6565 5050

传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司

地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室

电话：(852) 2116 3288 2116 3338

传真：(852) 2116 0882

北京办事处

地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室

邮编：100007

电话：(86-10) 8418 6608

传真：(86-10) 8418 6211

深圳办事处

地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室

邮编：518028

电话：(86-0755) 8335 0911 8335 1011 8335 2011 8335 0611

传真：(86-0755) 8335 9011

台湾办事处

地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室

电话：(886-2) 7721 1889

传真：(886-2) 7722 3888

新加坡办事处

地址：237, Alexandra Road, #07-01, The Alexcier, Singapore 159929

电话：(65) 6472 3688

传真：(65) 6472 3669

北美办事处

地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA

电话：(480) 857-6500 ext 18

公司网址：<http://www.fmsb.com/>