

FT61F13X

INT Application note

目录

1. 中断(INTERRUPTS)	3
1.1. 中断相关寄存器汇总.....	4
1.2. PA4-INT 和 PORTA 端口变化中断	6
2. 应用范例.....	7
联系信息	11

FT61F13x INT 应用

1. 中断(INTERRUPTS)

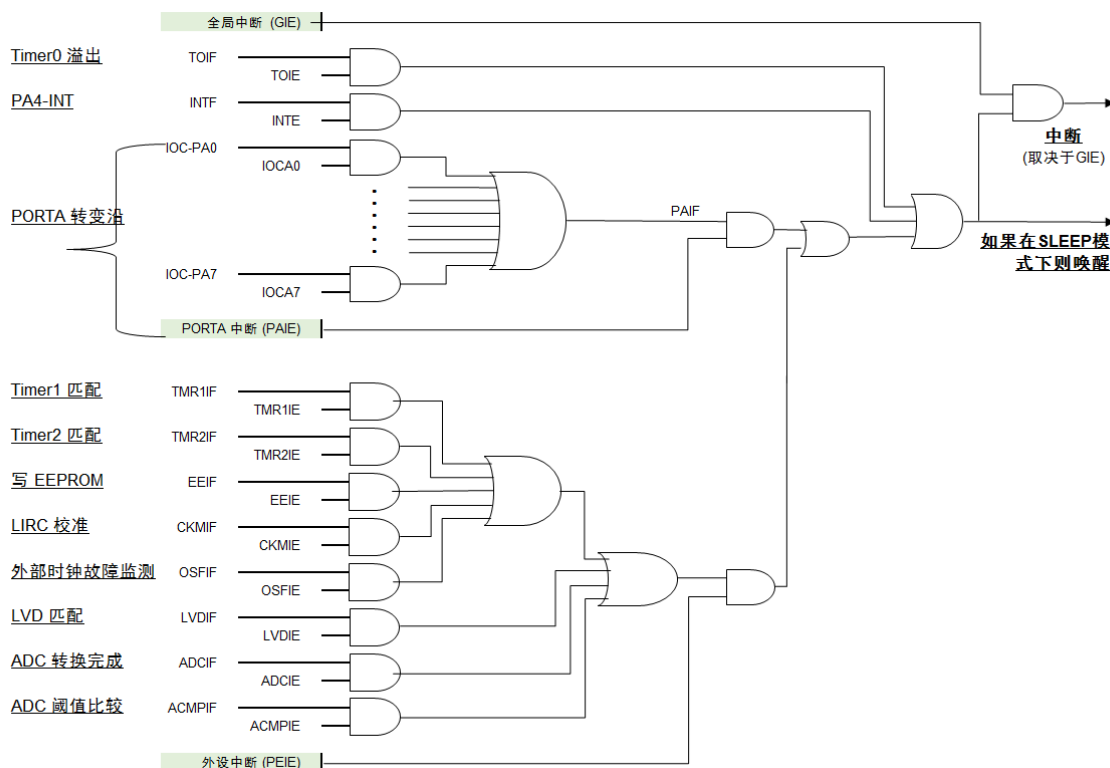


图 1-1 中断结构框图

CPU 支持 11 个中断源，分为 2 组：

1) 非外设中断 (Timer0 和 I/O)

- Timer0 溢出
- PA4-INT (自动上升沿或下降沿中断)
- PORTA 端口变化中断 (软件控制)

2) 外设中断

- Timer1 与 PR1 匹配
- Timer2 后分频器溢出
- DATA EEPROM 写完成
- LIRC 和 HIRC 交叉校准完成
- 故障保护时钟监控器
- LVD 条件匹配
- ADC 转换完成
- ADC 阈值比较匹配

与其他 Timers 不同，WDT 溢出不会产生中断。除外部 I/O 中断外，其他中断请参阅相应章节。

产生中断时，PC 跳转并执行“中断服务程序(ISR)”。中断的关闭/使能有多层控制：

- 每个中断源均有其独立的中断使能位：T0IE, INTE, IOCAx, TMR1IE, TMR2IE, EEIE, CKMIE, LVDIE, ADCIE, ACMPIE。
- 8 个 PAX 中断输入共用一个端口中断使能位：PAIE (PORTA Interrupt Enable)。
- 外设中断有一个总中断使能位：PEIE (PEripheral Interrupt Enable)。
- 如果关闭以上所有控制位，将不会执行从睡眠中唤醒。
- 所有中断都由全局中断使能位控制：GIE (Global Interrupt Enable)。与其他使能位不同，当关闭全局中断使能位时，依然允许从睡眠中唤醒。
- 关闭中断使能位并不影响中断标志位的置位。

中断处理时序如下：

- 自动设置“GIE = 0”，从而关闭中断。
- 返回地址被压入堆栈，程序指针 PC 加载 0x0004 地址。
- 发生中断后的 1 – 2 个指令周期，跳转到“中断服务程序(ISR)”开始处理中断。
- 执行“从中断返回(RETl)”指令退出 ISR。在 RETl 之前必须清除所有中断标志位。
- 当 ISR 完成时，PC 返回到中断前的地址，如果在 SLEEP 模式下，则返回到 SLEEP 指令后紧跟的地址。
- 在执行 RETl 时自动设置“GIE = 1”，从而使能中断。

注： 中断过程中，只有返回 PC 地址被自动保存在堆栈上。如果用户需要保存其他重要的寄存器值(如 W, STATUS 寄存器等)，必须通过指令将这些值正确地写入临时寄存器中，建议使用 GPR 的最后 16 个 bytes 作为临时寄存器，因为所有 bank 共用这 16 个 bytes，而不需要切换 bank 以节省代码。

1.1. 中断相关寄存器汇总

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值 (RW)
INTCON	0x0B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
PIR1	0x0C	EEIF	CKMIF	LVDIF	ACMPIF	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
TRISA	0x85	PORTA 方向控制								1111 1111
IOCA	0x96	PORTA 端口变化中断设置								0000 0000

表 1-1 中断相关寄存器地址和默认值

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, 各中断独立使能位适用) 0 = <u>全局关闭</u> (唤醒不受影响)	INTCON[7]	0x0B 0x8B 0x10B 0x18B	RW-0
PEIE	外设总中断	1 = 使能 (各中断独立使能位适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]		RW-0
T0IE	Timer0 溢出中断	1 = 使能 0 = <u>关闭</u> (无唤醒)	INTCON[5]		RW-0
INTE	PA4-INT 外部中断		INTCON[4]		RW-0
PAIE	PORTA 端口变化中断		INTCON[3]		RW-0
T0IF	Timer0 溢出中断标志位	1 = Yes (锁存) 0 = <u>No</u>	INTCON[2]		RW-0
INTF	PA4-INT 外部中断标志位		INTCON[1]		RW-0
PAIF	PORTA 端口变化中断标志位		INTCON[0]		RW-0

表 1-2 INTCON 寄存器

名称	状态		寄存器	地址	复位值
EEIE	EE 写完成中断	1 = 使能 0 = <u>关闭</u> (无唤醒)	PIE1[7]	0x8C	RW-0
CKMIE	LIRC和HIRC交叉校准完成中断		PIE1[6]		RW-0
LVDIE	LVD 中断		PIE1[5]		RW-0
ACMPIE	ADC 阈值比较匹配中断		PIE1[4]		RW-0
TMR1IE	Timer1与PR1匹配中断		PIE1[3]		RW-0
OSFIE	外部振荡器故障中断		PIE1[2]		RW-0
TMR2IE	Timer2与PR2匹配中断		PIE1[1]		RW-0
ADCIE	ADC 转换完成中断		PIE1[0]		RW-0

表 1-3 PIE1 寄存器

名称	状态		寄存器	地址	复位值
EEIF	EEPROM 写完成标志位	1 = Yes (锁存) 0 = <u>No</u>	PIR1[7]	0x0C	RW-0
CKMIF	LIRC和HIRC交叉校准完成标志位		PIR1[6]		RW-0
LVDIF	LVD 中断标志位		PIR1[5]		RW-0
ACMPFIF	ADC 阈值比较匹配标志位		PIR1[4]		RW-0
TMR1IF	Timer1与PR1匹配标志位		PIR1[3]		RW-0
OSFIF	外部振荡器故障标志位		PIR1[2]		RW-0
TMR2IF	Timer2与PR2匹配标志位		PIR1[1]		RW-0
ADCIF	ADC 转换完成标志位		PIR1[0]		RW-0

表 1-4 PIR1 寄存器

名称	状态	寄存器	地址	复位值
/PAPU	<u>PORTA 上拉</u> 1 = <u>全局关闭</u> 0 = 由 WPUA 控制	OPTION[7]	0x81	RW-1
INTEDG	<u>PA4 中断沿</u> 1 = <u>上升沿</u> 0 = <u>下降沿</u>	OPTION[6]		RW-1
TRISA	<u>PORTA I/O 数字输出 (方向控制)</u> 1 = <u>输入 (关闭数字输出)</u> 0 = <u>关闭上拉/下拉</u>	TRISA[7:0]	0x85	RW-11111111
IOCA	<u>PORTA 端口变化中断</u> 1 = <u>使能</u> 0 = <u>关闭</u>	IOCA[7:0]	0x96	RW-00000000

表 1-5 OPTION, TRISA 和 IOCA 寄存器

1.2. PA4-INT 和 PORTA 端口变化中断

名称	PA4-INT	PORTA 端口变化中断
通道数	仅PA4	PA0 – PA7 (多达8个通道)
I/O 设置	TRISA[4] = 1; ANSEL0[4] = 0	TRISA[x] = 1; ANSEL0[x] = 0
其他设置	INTEDG, INTE, GIE, INTF	IOCA, PAIE, GIE, PAIF
触发	上升沿或下降沿 (二选一)	0 → 1 或 1 → 0
需要软件监控?	No	需要

表 1-6 PA4-INT 和 PORTA 端口变化中断之间的区别

PA4-INT 和 PORTA 端口变化中断均为外部 I/O 中断，而两者对 PA4 均适用。如果设置正确，PA4-INT 将在后台运行而无需监督。PORTA 端口变化中断则需要持续的软件监控。对于 PORTA 端口变化中断：

1. 将输入寄存器值锁存到端口变化中断锁存器中(通过读 PORTA)。
2. 当输入电平变化时，输入寄存器值与锁存器值之间的差异将置位 PAIF。
3. 输入寄存器的锁存过程(即读 PORTA 的过程)将更新用于比较的参考电平，如果在 PAIF 置位后立即读 PORTA 即可清除端口变化中断的触发条件。当端口不匹配事件不再存在时，PAIF 可通过指令清除。

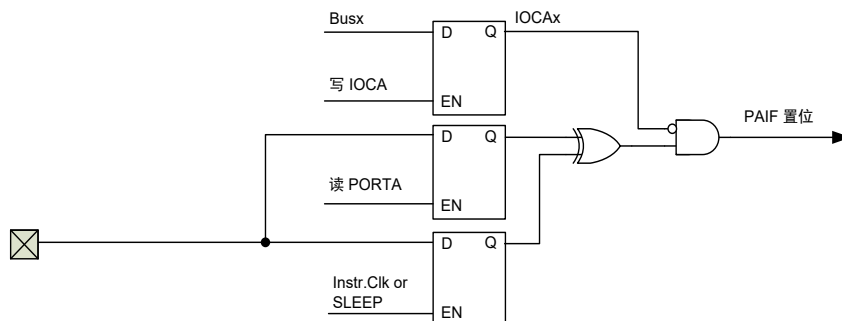


图 1-2 PORTA 转变沿中断

2. 应用范例

```
// Project:   test_61F13x_INT.c
// Device:   FT61F13X
// Memory:   Flash 3Kx14b, EEPROM 12x8b, SRAM 256x8b
// Company:
// Version:
// Date:
/* 文件名: test_61F13x_INT.c
* 功能:   FT61F13x-INT 功能演示
* IC:     FT61F135 SOP20
* 晶振:   16MHz/2T
* 说明:   程序中 DemoPortOut(PA2)输出 100 帧 50Hz 的占空比为 50%的方波后,
*         MCU 进入睡眠, 等待外部中断(PA4)的发生;
*         当外部中断触发后, 重复以上流程;
*
*         FT61F135  SOP20
*         -----
* VDD-----|1(VDD)  (VSS)20|-----VSS
* NC-----|2(PC1)  (PA0)19|-----NC
* NC-----|3(PC0)  (PA1)18|-----NC
* NC-----|4(PB7)  (PA2)17|---DemoPortOut
* NC-----|5(PB6)  (PA3)16|-----NC
* NC-----|6(PB5)  (PA4)15|-----INT
* NC-----|7(PB4)  (PA5)14|-----NC
* NC-----|8(PB3)  (PA6)13|-----NC
* NC-----|9(PB2)  (PA7)12|-----NC
* NC-----|10(PB1) (PB0)11|-----NC
*
*         -----
*/
//=====
#include "SYSCFG.h";
//=====
//Variable definition
//=====
#define  uchar      unsigned char

#define  DemoPortOut  PA2

uchar  ReadAPin;
uchar  FCount=0;
```

```
/*-----
 * 函数名: DelayUs
 * 功能:   短延时函数
 * 输入:   Time 延时时间长度 延时时长 Time  $\mu$ s
 * 返回:   无
 *-----*/
void DelayUs(unsigned char Time)
{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}

/*-----
 * 函数名: DelayMs
 * 功能:   短延时函数--16M-2T--大概快 1%左右.
 * 输入:   Time 延时时间长度 延时时长 Time ms
 * 返回:   无
 *-----*/
void DelayMs(unsigned char Time)
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197);    //快 1%
        }
    }
}

/*-----
 * 函数名: POWER_INITIAL
 * 功能:   上电系统初始化
 * 输入:   无
 * 输出:   无
 *-----*/
void POWER_INITIAL (void)
{
    OSCCON = 0B01110001;    //IRCF=111=16MHz/2T=8MHz, 0.125 $\mu$ s
    INTCON = 0;             //暂禁止所有中断
    OPTION = 0B00001000;    //Bit3=1 WDT, Bit[2:0]=000= WDT RATE 1:1

    PORTA  = 0B00000000;
```



```

    TRISA  = 0B00010000;    //PA 输入输出 0-输出 1-输入
                             //PA2-OUT PA4-IN

    PORTC  = 0B00000000;
    TRISC  = 0B00000000;    //PC 输入输出 0-输出 1-输入

    WPUA   = 0B00010000;    //PA 端口上拉控制 1-开上拉 0-关上拉
                             //开 PA4 上拉
    WPUC   = 0B00000000;    //PC 端口上拉控制 1-开上拉 0-关上拉

    ANSEL0 = 0B00000000;
}
/*-----
 * 函数名: interrupt
 * 功能:   中断程序
 * 输入:   无
 * 输出:   无
-----*/
void interrupt ISR(void)
{
    if(INTF==1)
    {
        ReadAPin = PORTA;
        INTF = 0;
        INTE = 0;
    }
    DelayMs(10);           //延时 10ms
}
/*-----
 * 函数名: INT_INITIAL
 * 功能:   中断初始化函数
 * 输入:   无
 * 输出:   无
-----*/
void INT_INITIAL(void)
{
    TRISA4 =1;             //设置 PA4 输出

    ReadAPin = PORTA;      //读清除错误匹配

    INTEDG =1;             //设置 PA4 INT 为上升沿触发
    INTF =0;               //清 PA4 INT 中断标志位
    INTE =1;              //使能 PA4 INT 中断
}
/*-----

```

* 函数名: main
* 功能: 主函数
* 输入: 无
* 输出: 无

-----*/

```
main()
{
    POWER_INITIAL();           //系统初始化
    while(1)
    {
        for(FCount=0;FCount<100;FCount++)
        {
            //输出 100 次波形
            DemoPortOut = 1;
            DelayMs(10);       //延时 10ms
            DemoPortOut = 0;
            DelayMs(10);
        }
        INT_INITIAL();         //初始化外部中断
        GIE = 1;               //开总中断
        SLEEP();               //睡眠
    }
}
```

联系信息**Fremont Micro Devices Corporation**

#5-8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

#16, 16/F, Block B, Veristrong Industrial Centre,
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.