

# **FT61F02X**

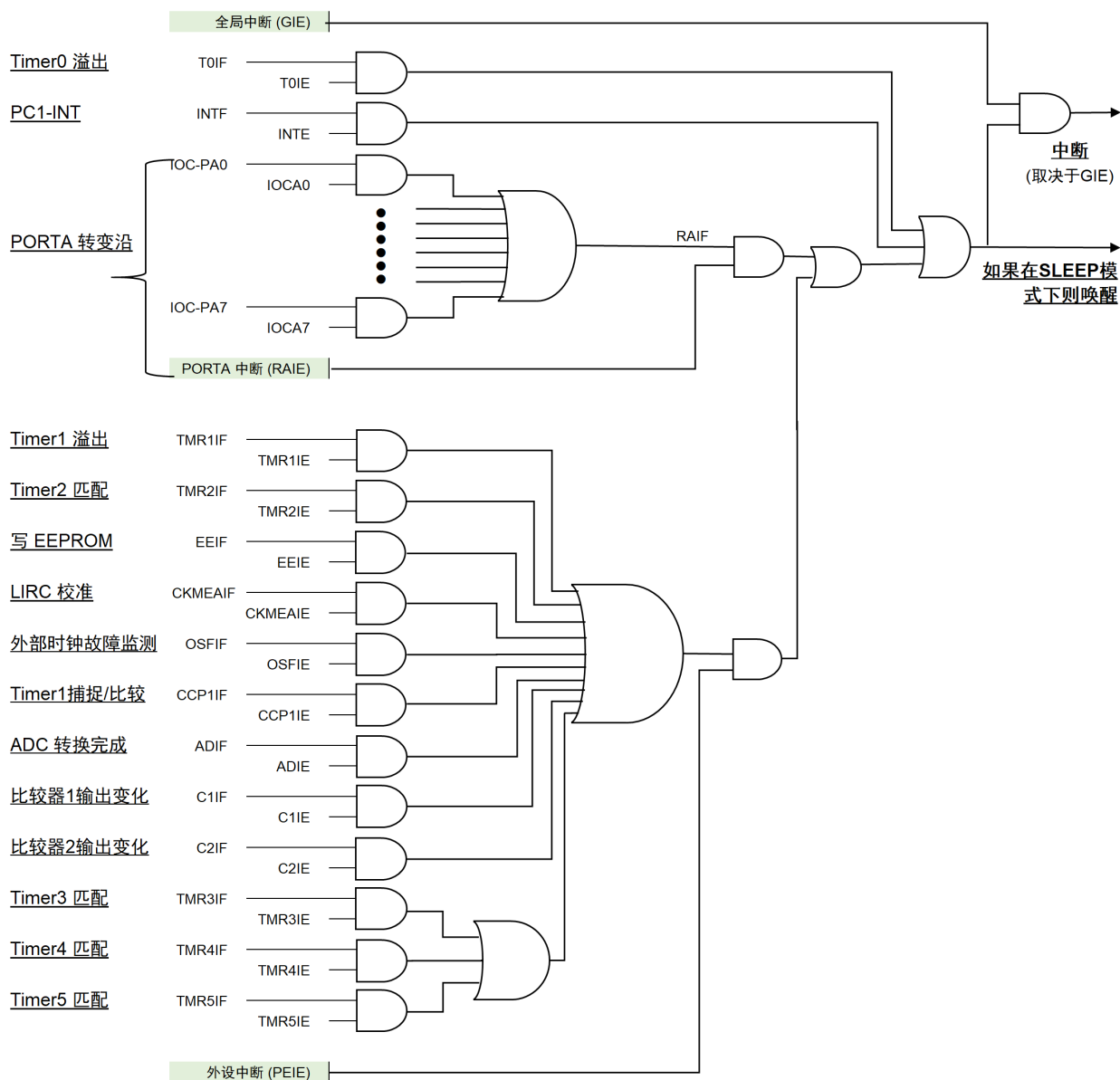
## **PA\_INT Application note**

## 目录

1. 中断(INTERRUPTS) .....	3
1.1. 中断相关寄存器汇总 .....	5
1.2. PC1-INT 和 PORTA 端口变化中断 .....	7
2. 应用范例 .....	8
联系信息 .....	12

## FT61F02x PA\_INT 应用

### 1. 中断(INTERRUPTS)



注：由于系统时钟在休眠期间暂停，只有不依赖于系统时钟的外设会将器件从休眠中唤醒

图 1-1 中断结构框图

CPU 支持 15 个中断源，分为 2 组：

#### 1) 非外设中断 (Timer0 和 I/O)

- Timer0 溢出
- PC1-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 溢出
- Timer2 与 PR2 匹配
- DATA EEPROM 写完成
- LIRC 和 HIRC 交叉校准完成
- 故障保护时钟监控器
- Timer1 捕捉/比较
- ADC 转换完成
- 比较器 1 / 比较器 2 输出变化
- Timer3 / Timer4 / Timer5 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, TMRxIE(x=1,2,3,4,5), EEIE, CKMEAIE, CxIE(x=1,2), OSFIE, ADIE, CCP1IE。
- 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。
- Timer0 中断无法从睡眠状态下唤醒 CPU。

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(RETI)”指令退出 ISR。在 RETI 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 RETI 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	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000
PIE2	0x8D	-	-	-	-	-	-	ADIE	CCP1IE	---- --00
PIR2	0x0D	-	-	-	-	-	-	ADIF	CCP1IF	---- --00
PWM3CR0	0x10F	P3INTS	P3PER[2:0]			P3CKSRC[2:0]			P3BZR	0000 0000
PWM3CR1	0x110	P3EN	P3POL	TMR3PS[2:0]			TMR3ON	TMR3IE	TMR3IF	0000 0000
PWM4CR0	0x115	P4INTS	P4PER[2:0]			P4CKSRC[2:0]			P4BZR	0000 0000
PWM4CR1	0x116	P4EN	P4POL	TMR4PS[2:0]			TMR4ON	TMR4IE	TMR4IF	0000 0000
PWM5CR0	0x11B	P5INTS	P5PER[2:0]			P5CKSRC[2:0]			P5BZR	0000 0000
PWM5CR1	0x11C	P5EN	P5POL	TMR5PS[2:0]			TMR5ON	TMR5IE	TMR5IF	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	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	PC1-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	PC1-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
CKMEAIE	LIRC和HIRC交叉校准完成中断		PIE1[6]		RW-0
C2IE	比较器2中断		PIE1[4]		RW-0
C1IE	比较器1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
TMR1IE	Timer1溢出中断		PIE1[0]		RW-0
ADIE	ADC转换完成中断		PIE2[1]	0x8D	RW-0
CCP1IE	CCP1捕捉 / 匹配中断		PIE2[0]		RW-0

表 1-3 PIEx 寄存器

名称	状态	寄存器	地址	复位值
EEIF	EEPROM 写完成标志位	PIR1[7]	0x0C	RW-0
CKMEAIF	LIRC和HIRC交叉校准完成标志位	PIR1[6]		RW-0
C2IF	比较器2中断标志位	PIR1[4]		RW-0
C1IF	比较器1中断标志位	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
TMR1IF	Timer1溢出标志位	PIR1[0]		RW-0
ADIF	ADC 转换完成标志位	PIR2[1]	0x0D	RW-0
CCP1IF	CCP1发生捕捉 / 匹配标志位	PIR2[0]		RW-0

表 1-4 PIRx 寄存器

名称	状态	寄存器	地址	复位值
P3INTS	Timer3中断选择位	PWM3CR0[7]	0x10F	RW-0
P4INTS	Timer4中断选择位	PWM4CR0[7]	0x115	RW-0
P5INTS	Timer5中断选择位	PWM5CR0[7]	0x11B	RW-0
TMR3IE	Timer3中断使能位	PWM3CR1[1]	0x110	RW-0
TMR4IE	Timer4中断使能位	PWM4CR1[1]	0x116	RW-0
TMR5IE	Timer5中断使能位	PWM5CR1[1]	0x11C	RW-0
TMR3IF	Timer3中断标志位	PWM3CR1[0]	0x110	RW-0
TMR4IF	Timer4中断标志位	PWM4CR1[0]	0x116	RW-0
TMR5IF	Timer5中断标志位	PWM5CR1[0]	0x11C	RW-0

表 1-5 Timer3/4/5 中断寄存器

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>PC1 中断沿</u> 1 = <u>上升沿</u> 0 = 下降沿	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 = 关闭上拉/下拉	TRISA[7:0]	0x85	RW-11111111
IOCA	<u>PORTA 端口变化中断</u> 1 = 使能      0 = <u>关闭</u>	IOCA[7:0]	0x96	RW-00000000

表 1-6 OPTION, TRISA 和 IOCA 寄存器

## 1.2. PC1-INT 和 PORTA 端口变化中断

名称	PC1-INT	PORTA 端口变化中断
通道数	仅PC1	PA0 – PA7 (多达8个通道)
I/O 设置	TRISC[1] = 1; ANSEL[4] = 0; CMCON0[2:0] = 000	TRISA[x] = 1; ANSEL[x] = 0; CMCON0[2:0] = 000
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 1-7 PC1-INT 和 PORTA 端口变化中断之间的区别

PC1-INT 和 PORTA 端口变化中断均为外部 I/O 中断。如果设置正确，PC1-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。

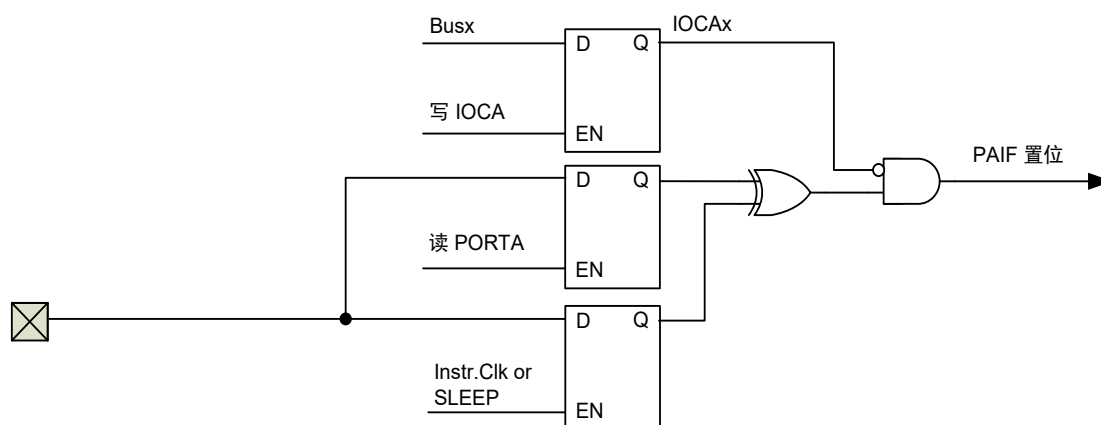


图 1-2 PORTA 转变沿中断

## 2. 应用范例

```
//*****
/* 文件名: TEST_61F02x_RA_INT.c
* 功能:    FT61F02x-PA 口电平中断功能演示
* IC:      FT61F023 SOP16
* 晶振:    16M/2T
* 说明:    程序中 DemoPortOut(PA3)输出 100 帧 50Hz 的占空比为 50%的方波后,
*          MCU 进入睡眠,等待中断的发生; 当每次 PA2 电平变化中断触发后,
*          重复以上流程;
*
*          FT61F023  SOP16
*          -----
*  VDD-----|1(VDD)  (VSS)16|-----GND
*  NC-----|2(PA7)   (PA0)15|-----NC
*  NC-----|3(PA6)   (PA1)14|-----NC
*  NC-----|4(PA5)   (PA2)13|-----RAINT<--电平变化中断 IN
*  NC-----|5(PC3)   (PA3)12|-----DemoPortOut
*  NC-----|6(PC2)   (PC0)11|-----NC
*  NC-----|7(PA4)   (PC1)10|-----NC
*  NC-----|8(PC5)   (PC4)09|-----NC
*          -----
*/
//*****
#include "SYSCFG.h"
//***** 宏定义 *****
#define unchar      unsigned char
#define DemoPortOut PA3

unchar FCount;
unchar ReadAPin;
/*-----*/
* 函数名: interrupt ISR
* 功能:    中断处理函数
* 输入:    无
* 输出:    无
*-----*/
void interrupt ISR(void)
{
    //PA 电平变化中断
    if(PAIE && PAIF)
    {
        ReadAPin = PORTA;    //读取 PORTA 数据清 PAIF 标志
        PAIF = 0;           //清 PAIF 标志位
        PAIE = 0;           //暂先禁止 PA2 中断
    }
}
```

```

        IOCA2 = 0;                //禁止 PA2 电平变化中断
    }
}
/*-----
* 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/2=8MHz,0.125μs
    INTCON = 0;                 //暂禁止所有中断
    PORTA = 0B00000000;
    TRISA = 0B00000100;        //PA 输入输出 0-输出 1-输入
                                //PA3->输出 PA2->输入

    PORTC = 0B00000000;
    TRISC = 0B00000000;        //PC 输入输出 0-输出 1-输入
    WPUA = 0B00000000;        //PA 端口上拉控制 1-开上拉 0-关上拉
    WPUC = 0B00000000;        //PC 端口上拉控制 1-开上拉 0-关上拉

    OPTION = 0B00001000;        //Bit3=1,WDT MODE,PS=000=WDT RATE 1:1
    MSCKCON = 0B00000000;
    //Bit6->0,禁止 PA4, PC5 稳压输出
    //Bit5->0,TIMER2 时钟为 Fosc
    //Bit4->0,禁止 LVR
    CMCON0 = 0B00000111;        //关闭比较器, CxIN 为数字 IO 口
}
/*-----
* 函数名: DelayUs
* 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 μs
* 输出:   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 快 1%  
 \* 输入: Time 延时时间长度 延时时长 Time ms  
 \* 输出: 无

-----\*/

void DelayMs(unsigned char Time)

```
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197);
        }
    }
}
```

/\*-----

\* 函数名: PA2\_Level\_Change\_INITIAL  
 \* 功能: PA 端口(PA2)电平变化中断初始化  
 \* 输入: 无  
 \* 输出: 无

-----\*/

void PA2\_Level\_Change\_INITIAL(void)

```
{
    ANSEL = 0B00000000;    //关闭中断引脚模拟输入功能
    TRISA2 = 1;            //设置 A2 上拉
    ReadAPin = PORTA;      //清 PA 电平变化中断
    PAIF = 0;              //清 PA INT 中断标志位
    IOCA2 = 1;             //使能 PA2 电平变化中断
    PAIE = 1;              //使能 PA INT 中断
}
```

/\*-----

\* 函数名: main  
 \* 功能: 主函数  
 \* 输入: 无  
 \* 输出: 无

-----\*/

void 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);           //10ms
    DemoPortOut = 0;
    DelayMs(10);
}
PA2_Level_Change_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.