

FT60F01X

PA_INT Application note

目录

1. PA_INT 相关寄存器的设置.....	3
2. 应用范例.....	6
联系信息	10

FT60F01x PA_INT 应用

1. PA_INT 相关寄存器的设置

PORTA的每个端口都可以被单独设置成一个中断源(端口状态变化触发中断)。控制IOCAx寄存器里的位就可使能或关断这些端口的中断功能。端口状态变化触发中断的功能在上电复位时无效的。

当端口状态变化触发中断的功能被使能时，当前端口电平值会与上次所读取数据寄存器的旧值作对比。所有错误匹配结果会被或在一起形成中断标志位(INTCON寄存器中的RAIF标志位)。

该中断可以将芯片从睡眠状态中唤醒。用户需要在中断服务程序中执行以下程序来清除该标志位:

A) 对PORTA进行一次读或写动作，这将结束任何错误匹配的状态。

B) 清零RAIF标志位。

错误匹配的条件会一直设置 RAIF 位。对 PORTA 做一次读就可以结束任何错误匹配的状态，使得 RAIF 能被清零。数据寄存器里保持的上一次读的值不会被/MCLR 或低电压复位所影响。只要错误匹配状态存在，RAIF 位就会被置 1。

相关寄存器的各个位定义如下：

1) PORTA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	PA5	PA4	PA3	PA2	PA1	PA0
Reset	-	-	0	0	x	0	0	0

Bit7~Bit6: -

Bit5: PORTA5 数据

Bit4: PORTA4 数据

Bit3: PORTA3 只有输入功能，不存在相应的数据寄存器

Bit2: PORTA2 数据

Bit1: PORTA1 数据

Bit0: PORTA0 数据

2) TRISA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	TRISA[5]	TRISA[4]	-	TRISA[2]	TRISA[1]	TRISA[0]
Reset	-	-	1	1	-	1	1	1

Bit7~ Bit6: N/A, 读 0

Bit5~ Bit4: PORTA<5:4>输入/输出状态控制寄存器

1: 端口为输入状态

0: 端口为输出状态

Bit3: - (注意: 仅为输入口, 读 1)

Bit2~Bit0: PORTA<2:0>输入/输出状态控制寄存器

1: 端口为输入状态

0: 端口为输出状态

3) WPUA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	WPUA5	WPUA4	-	WPUA2	WPUA1	WPUA0
Reset	-	-	1	1	-	1	1	1

Bit7~Bit6、Bit3: N/A, 读 0

Bit5~Bit4、Bit2~Bit0: Port A 弱上拉使能

1: 使能PORTA 端口弱上拉

0: 关闭PORTA 端口弱上拉

4) IOCA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-		IOCA[5:0]					
Reset	-		6'h00					

Bit7~Bit6、Bit3: N/A, -

Bit5~ Bit0: Port A 端口状态触发中断设置

1: 使能端口状态触发中断

0: 关闭端口状态触发中断

5) OPTION 寄存器

Bit	7	6	5	4	3	2	1	0
Name	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
Reset	1	1	1	1	1	1	1	1

Bit7: PORTA 口上拉使能位

1: 上拉功能被禁止

0: 上拉功能使能

Bit6: 触发中断边沿选择位

1: PA2/INT 上升沿触发中断

0: PA2/INT 下降沿触发中断

Bit5: Timer0 时钟选择位

1: PA2/T0CKI管脚输入时钟

0: 内部指令周期Fosc/4

Bit4: Timer0 时钟边沿选择位

1: PA2/T0CKI管脚由高到底变化时计数增加

0: PA2/T0CKI管脚由低到高变化时计数增加

Bit3: 预分频分配位

1: 预分频器分配给WDT

0: 预分频器分配给Timer0

Bit2~Bit0: 预分频大小设置位

Bit2: Bit0	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

6) INTCON 寄存器

Bit	7	6	5	4	3	2	1	0
Name	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF
Reset	0	0	0	0	0	0	0	0

Bit7: 全局中断使能位

1: 使能全局中断

0: 关闭全局中断

Bit6: 外设中断使能位

1: 使能外设中断

0: 关闭外设中断

Bit5: Timer0 定时器溢出中断使能位

1: 使能Timer0中断

0: 关闭Timer0中断

Bit4: 外部中断使能位

1: 使能PA2/INT外部中断

0: 关闭PA2/INT外部中断

Bit3: PORTA端口状态变化中断使能位

1: 使能PORTA端口状态变化中断使能

0: 关闭PORTA端口状态变化中断使能

Bit2: Timer0定时器溢出中断标志位

1: Timer0寄存器溢出（必须软件清零）

0: Timer0寄存器无溢出

Bit1: 外部中断标志位

1: PA2/INT外部中断发生（必须软件清零）

0: PA2/INT外部中断无发生

Bit0: PORTA端口状态改变中断标志位

1: PORTA<5:0>至少有一个端口状态发生了改变（必须软件清零）

0: PORTA<5:0>没有一个端口发生状态改变

2. 应用范例

```
//*****
/* 文件名: TEST_60F01x_PA_INT.c
* 功能:    FT60F01x-PA 口电平中断功能演示
* IC:      FT60F011A SOP8
* 晶振:    16M/4T
* 说明:    程序中 DemoPortOut(PA4)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;
*
*          FT60F011A  SOP8
*          -----
*  VDD-----|1(VDD)  (GND)8|-----GND
*  NC-----|2(PA2)  (PA4)7|----DemoPortOut
*  NC-----|3(PA1)  (PA5)6|-----NC
*  NC-----|4(PA3)  (PA0)5|-----NC
*          -----
*/
//*****
#include "SYSCFG.h"
//*****宏定义*****
#define unchar    unsigned char

#define DemoPortOut PA4

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 PA0 中断
        IOCA2 = 0;          //禁止 PA0 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 = 0B11101111;    //PA 输入输出 0-输出 1-输入
                           //PA4->输出
    WPUA = 0B00000000;    //PA 端口上拉控制 1-开上拉 0-关上拉

    OPTION = 0B00001000;    //Bit3=1 WDT MODE,PS=000=1:1 WDT RATE
    MSCKCON = 0B00000000;
    //Bit4=0,禁止 LVR(60F01x O 版之前)
    //Bit4=0,LVREN 使能时,开启 LVR(60F01x O 版及 O 版之后)
    //Bit4=1,LVREN 使能时,工作时开启 LVR,睡眠时自动关闭 LVR(60F01x O 版及 O 版后)
}

/*-----
* 函数名称: DelayUs
* 功能:     短延时函数 --16M-4T--大概快 1%左右.
* 输入参数: Time 延时时间长度 延时时长 Time μs
* 返回参数: 无
-----*/

void DelayUs(unsigned char Time)
{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}

/*-----
* 函数名称: DelayMs
* 功能:     短延时函数
* 输入参数: Time 延时时间长度 延时时长 Time ms
* 返回参数: 无
-----*/

void DelayMs(unsigned char Time)
{

```

```

    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(98);          //快 1%
        }
    }
}
/*-----
* 函数名称: DelayS
* 功能:      短延时函数
* 输入参数: Time 延时时间长度 延时时长 Time S
* 返回参数: 无
-----*/
void DelayS(unsigned char Time)
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<10;b++)
        {
            DelayMs(100);
        }
    }
}
/*-----
* 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)
{
    TRISA2 = 1;          //设置 PA2 输入
    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.