

实验三:外部中断/定时器实验

1. 实验目的

- 掌握外部中断引脚功能设置及外部中断工作模式设置。
- 学习中断服务函数的编写。
- 掌握定时器外设的操作原理和编程。

2. 实验设备

- 硬件：PC 机一台
Mini2440 ARM 实验板一套
J-link 仿真器一套
- 软件：WindowsXP 系统，Keil uVision 4.0 集成开发环境

3. 实验内容

(1) 将 GPIO0 设置为低电平触发外部中断；然后等待中断事件。中断服务程序利用计数器 (R2) 计算中断发生的次数。使用 Keil uVision 的调试功能单步、全速运行程序，设置断点，打开寄存器窗口监视寄存器，观察计数器的变化。

(2) 使用 GPIO 读取 Mini2440 实验板上的按键状态，观察按键输入的抖动现象。

4. 实验预习要求

- (1) 学习 ARM 中断工作原理和编程方法；
- (2) 查阅 S3C2440 芯片手册，了解中断处理系统的结构和原理。

5. 实验步骤

(一) 外部中断实验：

(1) 开发板按键 1 连接到 EINT8/GPIO0，见图 3-1，本实验中该引脚被配置成外部中断输入功能。

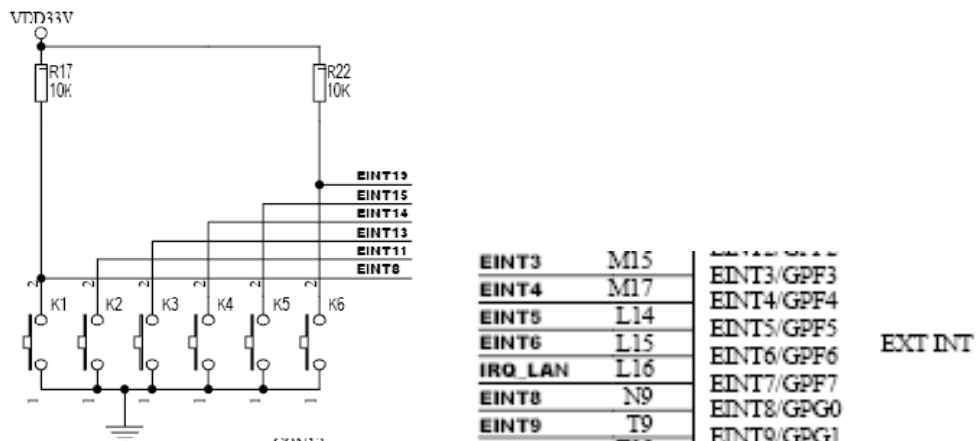


图 3-1 按键输入

(2) 启动 Keil uVision, 新建一个工程 ex03-1。不需要系统提供的 Startup 文件。建立汇编源文件 ex03-1.s, 编写实验程序, 然后添加到工程中。设置工程选项, 存储器映射。设置工程调试选项。建立仿真初始化文件 RAM.ini。

具体步骤参考实验二。

(3) S3C2440 中断控制器原理见图 3-2。

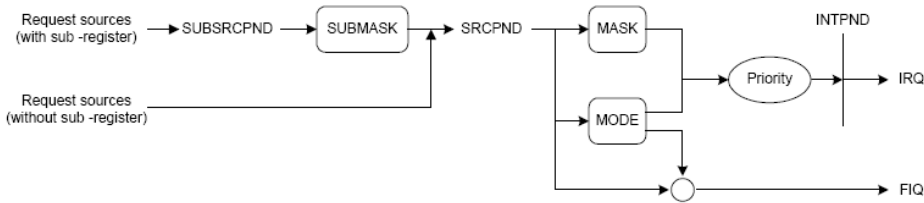


图 3-2 中断控制器原理

- (4) 实验程序分为三个部分：
- a) 设置中断向量表；
 - b) 初始化 ARM 处理器、中断控制器、I/O 输入引脚的设置，允许中断发生和处理，然后主程序进入空循环，等待中断事件；
 - c) 准备中断处理程序，对中断事件进行相应的处理；
- (5) 编译链接工程。连接实验板电源、J-link 仿真器，进行仿真调试。全速执行程序，按下按键 1，然后暂停程序执行，观察计数值的变化。

参考：GPIOG 控制寄存器

Register	Address	R/W	Description	Reset Value
GPGCON	0x56000060	R/W	Configures the pins of port G	0x0
GPG4	[9:0]		00 = Input 10 = EINT[12] 01 = Output 11 = LCD_PWRDN	
GPG3	[7:6]		00 = Input 10 = EINT[11] 01 = Output 11 = nSS1	
GPG2	[5:4]		00 = Input 10 = EINT[10] 01 = Output 11 = nSS0	
GPG1	[3:2]		00 = Input 10 = EINT[9] 01 = Output 11 = Reserved	
GPG0	[1:0]		00 = Input 10 = EINT[8] 01 = Output 11 = Reserved	

(一) 定时器实验：

(1) 本实验使用 S3C2440 片内定时器 0，见图 3-3。PCLK 时钟经过预分频作为递减计数器的时钟，当计数器的值减小到 0 时，将触发中断事件。

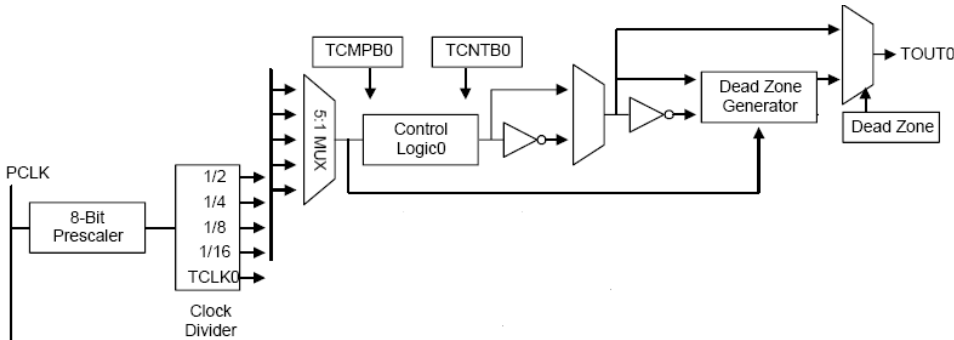


图 3-3 按键输入

(2) 启动 Keil uVision, 新建一个工程 ex03-2。不需要系统提供的 Startup 文件。建立汇编源文件 ex03-1.s, 编写实验程序, 然后添加到工程中。设置工程选项, 存储器映射。设置工程调试选项。建立仿真初始化文件 RAM.ini。

具体步骤参考实验二。

(3) 实验程序分为三个部分:

a) 设置中断向量表;

b) 初始化 ARM 处理器、中断控制器、定时器的设置, 允许中断发生和处理, 然后主程序进入空循环, 等待中断事件;

c) 准备中断处理程序, 对中断事件进行相应的处理;

(5) 编译链接工程。连接实验板电源、J-link 仿真器, 进行仿真调试。全速执行程序, 然后暂停程序执行, 观察计数值的变化。

参考: 定时器控制寄存器

Register	Address	R/W	Description	Reset Value
TCFG0	0x51000000	R/W	Configures the two 8-bit prescalers	0x00000000

TCFG0	Bit	Description	Initial State
Reserved	[31:24]		0x00
Dead zone length	[23:16]	These 8 bits determine the dead zone length. The 1 unit time of the dead zone length is equal to that of timer 0.	0x00
Prescaler 1	[15:8]	These 8 bits determine prescaler value for Timer 2, 3 and 4	0x00
Prescaler 0	[7:0]	These 8 bits determine prescaler value for Timer 0 and 1.	0x00

TIMER CONTROL (TCON) REGISTER

Register	Address	R/W	Description	Reset Value
TCON	0x51000008	R/W	Timer control register	0x00000000

TCON (Continued)

TCON	Bit	Description	Initial state
Reserved	[7:5]	Reserved	
Dead zone enable	[4]	Determine the dead zone operation. 0 = Disable 1 = Enable	0
Timer 0 auto reload on/off	[3]	Determine auto reload on/off for Timer 0. 0 = One-shot 1 = Interval mode(auto reload)	0
Timer 0 output inverter on/off	[2]	Determine the output inverter on/off for Timer 0. 0 = Inverter off 1 = Inverter on for TOUT0	0
Timer 0 manual update (note)	[1]	Determine the manual update for Timer 0. 0 = No operation 1 = Update TCNTB0 & TCMPB0	0
Timer 0 start/stop	[0]	Determine start/stop for Timer 0. 0 = Stop 1 = Start for Timer 0	0

TIMER 0 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB0/TCMPB0)

Register	Address	R/W	Description	Reset Value
TCNTB0	0x5100000C	R/W	Timer 0 count buffer register	0x00000000

TCNTB0	Bit	Description	Initial State
Timer 0 count buffer register	[15:0]	Set count buffer value for Timer 0	0x00000000

6. 实验参考程序

GPIO 输出实验的参考程序见程序清单 3.1。GPIO 输入实验的参考程序见程序清单 2.2。

程序清单 3.1 外部中断实验参考程序

```
NOINT      EQU    0x80                ; 1000 0000

BIT_EINT8_23 EQU    (0x1<<5)
EINT8      EQU    (0x1<<8)

GPGCON     EQU    0x56000060          ;Port G control register
EXTINT1    EQU    0x5600008c          ;External interrupt control register 1

INTMSK     EQU    0x4a000008          ;Interrupt mask control
EINTMASK   EQU    0x560000a4          ;External interrupt mask

SRCPND     EQU    0x4a000000          ;Interrupt request status
INTPND     EQU    0x4a000010          ;Interrupt request status
EINTPEND   EQU    0x560000a8          ;External interrupt pending

        AREA RESET, CODE, READONLY    ;声明代码段 RESET
        ENTRY                          ;表示程序入口
        CODE32                          ;声明 32 位 ARM 指令

        b    Reset

Undef     b    Undef                    ;handler for Undefined mode
SWI       b    SWI                      ;handler for SWI interrupt
Pabort    b    Pabort                  ;handler for PAbort
Dabort    b    Dabort                  ;handler for DAbort
        b    .                          ;reserved
        b    IRQ                       ;handler for IRQ interrupt
FIQ       b    FIQ                     ;handler for FIQ interrupt

Reset

        LDR    R1,=GPGCON               ;set GPIO portG(0) as external interrupt 8
        LDR    R0,=0x2
        STR    R0, [R1]

        LDR    R1,=EXTINT1              ;set external interrupt 8 as 'Falling edge triggered'
        LDR    R0,=0x2                  ;bit[2:0] = 010
        STR    R0, [R1]

        LDR    R1,=INTMSK              ;
```

```

        LDR    R0, [R1]           ;get interrupt mask register
        BIC    R0, #0x20         ;clear EINT8 mask
        STR    R0, [R1]         ;set interrupt mask register

        LDR    R1, =EINTMASK ;
        LDR    R0, [R1]         ;get external interrupt mask register
        BIC    R0, #0x100       ;clear EINT8 mask
        STR    R0, [R1]         ;set external interrupt mask register

        mrs    r0, cpsr         ;get cpsr
        bic    r0, r0, #NOINT   ;clear irq mask
        msr    cpsr_cxsf, r0    ;set cpsr

Loop
        B      Loop

IRQ     LDR    R1, =EINTPEND     ;clear External interrupt pending bit
        LDR    R0, =EINT8
        STR    R0, [R1]

        LDR    R1, =SRCPND      ;clear interrupt source pending bit
        LDR    r0, =BIT_EINT8_23
        STR    R0, [R1]

        LDR    R1, =INTPND      ;clear interrupt pending bit
        STR    R0, [R1]
        LDR    R0, [R1]

        add    r2, r2, #0x1     ;increase counter (R2)
        subs   pc, lr, #0x4     ;return from interrupt: pc = r14 - 4

        END

```

程序清单 3.2 定时器实验参考程序

```

NOINT      EQU    0x80          ; 1000 0000

BIT_TMR_INT EQU    (0x1<<10)
TICKS_RATE EQU    1
PCLK       EQU    12000000
TCNTB0_VAL EQU    ((PCLK/(2*255*TICKS_RATE)) - 1)

TCFG0      EQU    0x51000000    ;Timer configuration register0
TCON       EQU    0x51000008    ;Timer control register
TCNTB0     EQU    0x5100000c    ;Timer count buffer 0

```

INTMSK EQU 0x4a000008 ;Interrupt mask control

SRCPND EQU 0x4a000000 ;Interrupt request status

INTPND EQU 0x4a000010 ;Interrupt request status

AREA RESET, CODE, READONLY

ENTRY

CODE32

b Reset

Undef b Undef ;handler for Undefined mode

SWI b SWI ;handler for SWI interrupt

Pabort b Pabort ;handler for PAbort

Dabort b Dabort ;handler for DAbort

b . ;reserved

b IRQ ;handler for IRQ interrupt

FIQ b FIQ ;handler for FIQ interrupt

Reset

LDR R1,=INTMSK ;

LDR R0,[R1] ;get interrupt mask register

BIC R0,#BIT_TMR_INT ;clear Timer0 int mask

STR R0,[R1] ;set interrupt mask register

mrs r0,cpsr ;get cpsr

bic r0,r0,#NOINT ;clear irq mask

msr cpsr_cxsf,r0 ;set cpsr

LDR R1,=TCFG0 ;set timer0 prescaler

LDR R0,=0xff ;prescaler value: 0xff

STR R0,[R1]

LDR R1,=TCNTB0 ;set timer counter value

LDR R0,=TCNTB0_VAL ;

STR R0,[R1]

LDR R1,=TCON ;

LDR R0,=0x2 ;manual load

STR R0,[R1] ;

LDR R0,=0x9 ;start as autoload mode

STR R0,[R1] ;

Loop

B Loop

IRQ LDR R1,=SRCPND ;clear interrupt source pending bit

 LDR r0,=BIT_TMR_INT

 STR R0, [R1]

 LDR R1,=INTPND ;clear interrupt pending bit

 STR R0, [R1]

 add r2, r2, #0x1 ;increase counter (R2)

 subs pc, lr, #0x4 ;return from interrupt: pc = r14 - 4

END

7. 思 考

- (1) 能否使用多个外部中断，如何区分中断来源？
- (2) 定时器产生中断的最大时间间隔是多大？

8. 选作内容

- (1) 用按键控制指示灯的状态。
- (2) 用定时器控制指示灯的闪烁速度。