Lab 6 An external switch generates a GPIO interrupt

Open project "EdgeInterrupt_4C123" from D:\Keil\EE319KwareSpring2016

We will begin with a simple example that counts the number of rising edges on Port F bit 4 . The initialization requires many steps.
(a) The clock for the port must be enabled.
(b) The global variables should be initialized.
(c) The appropriate pins must be enabled as inputs.
(d) We must specify whether to trigger on the rise, the fall, or both edges. In this case we will trigger on the rise of PF4. 清理触发器标志，避免开机干扰
(e) It is good design to clear the trigger flag during initialization so that the first interrupt occurs due to the first rising edge after the initialization has been run. We do not wish to count a rising edge that might have occurred during the power up phase of the system.
(f) We arm the edge-trigger by setting the corresponding bits in the **IM** register.
(g) We establish the priority of Port F by setting bits 23 – 21 in the **NVIC_PRI7_R** register as listed in Table 1. We activate Port F interrupts in the NVIC by setting bit 30 in the **NVIC_EN0_R** register, Table 2. There is no need to unlock PF4.

Table 1. The LM3S/TM4C NVIC registers. Each register is 32 bits wide. Bits not shown are zero.

Address	31 – 29	23 – 21	15 – 13	7 – 5	Name
0xE000E400	GPIO Port D	GPIO Port C	GPIO Port B	GPIO Port A	NVIC_PRI0_R
0xE000E404	SSI0, Rx Tx	UART1, Rx Tx	UART0, Rx Tx	GPIO Port E	NVIC_PRI1_R
0xE000E408	PWM Gen 1	PWM Gen 0	PWM Fault	I2C0	NVIC_PRI2_R
0xE000E40C	ADC Seq 1	ADC Seq 0	Quad Encoder	PWM Gen 2	NVIC_PRI3_R
0xE000E410	Timer 0A	Watchdog	ADC Seq 3	ADC Seq 2	NVIC_PRI4_R
0xE000E414	Timer 2A	Timer 1B	Timer 1A	Timer 0B	NVIC_PRI5_R
0xE000E418	Comp 2	Comp 1	Comp 0	Timer 2B	NVIC_PRI6_R
0xE000E41C	GPIO Port G	GPIO Port F	Flash Control	System Control	NVIC_PRI7_R
0xE000E420	Timer 3A	SSI1, Rx Tx	UART2, Rx Tx	GPIO Port H	NVIC_PRI8_R
0xE000E424	CAN0	Quad Encoder 1	I2C1	Timer 3B	NVIC_PRI9_R
0xE000E428	Hibernate	Ethernet	CAN2	CAN1	NVIC_PRI10_R
0xE000E42C	uDMA Error	uDMA Soft Tfr	PWM Gen 3	USB0	NVIC_PRI11_R
0xE000ED20	SysTick	PendSV	--	Debug	NVIC_SYS_PRI3_R

Table 2. Some of the TM4C NVIC interrupt enable registers. There are five such registers defining 139 interrupt enable bits.

Address	31	30	29-7	6	5	4	3	2	1	0	Name
0xE000E100	G	F	…	UART1	UART0	E	D	C	B	A	NVIC_EN0_R
0xE000E104			…						UART2	H	NVIC_EN1_R

Using edge triggering to synchronize software to hardware centers around the operation of the trigger flags, **RIS**. A busy-wait interface will read the appropriate **RIS** bit over and over, until it is set. When the **RIS** bit is set, the software will clear the **RIS** bit (by writing a one to the corresponding **IC** bit) and perform the desired function. With interrupt synchronization, the initialization phase will arm the trigger flag by setting the corresponding **IM** bit. In this way, the active edge of the pin will set the **RIS** and request an interrupt. The interrupt will suspend the main program and run a special interrupt service routine (ISR). This ISR will clear the **RIS** bit and perform the desired function. At the end of the ISR it will return, causing the main program to resume. In particular, five conditions must be simultaneously true for an edge-triggered interrupt to be requested:

Table 3. Edge-triggered modes.

DIR	AFSEL	PMC	IS	IBE	IEV	IME	Port mode
0	0	0000	0	0	0	0	Input, falling edge trigger, busy wait
0	0	0000	0	0	1	0	Input, rising edge trigger, busy wait
0	0	0000	0	1	-	0	Input, both edges trigger, busy wait
0	0	0000	0	0	0	1	Input, falling edge trigger, interrupt
0	0	0000	0	0	1	1	Input, rising edge trigger, interrupt
0	0	0000	0	1	-	1	Input, both edges trigger, interrupt

- The trigger flag bit is set (RIS)
- The arm bit is set (IME)
- The level of the edge-triggered interrupt must be less than BASEPRI
- The edge-triggered interrupt must be enabled in the NVIC_EN0_R
- The I bit, bit 0 of the special register PRIMASK, is 0

This initialization is shown to enable interrupts in step (i). However, in most systems we would not enable interrupts in the device initialization. Rather, it is good design to initialize all devices in the system, then enable interrupts. All ISRs must acknowledge the interrupt by clearing the trigger flag that requested the interrupt. For edge-triggered PF4, the trigger flag is bit 4 of the **GPIO_PORTF_RIS_R** register. This flag can be cleared by writing a 0x10 to **GPIO_PORTF_ICR_R**.

要求：

把触发引脚从 PF4 改为 PF0，重新验证实验结果，分析实验现象。