Interrupts Activity

Lecture 19

Josh Brake

Harvey Mudd College

Learning Objectives

By the end of this lecture you will be able to:

- Configure interrupts with a variety of peripherals
  - Toggle LED with button press
  - Toggle LED based on timer

Button Interrupt

Button Interrupt

- What GPIO pin do you want to use?
- Steps to configure:
  - EXTI mux in SYSCFG peripheral
  - Interrupt generation settings in EXTI peripheral
  - Globally enable interrupts
  - Set Interrupt Mask Register
  - Select rising/falling edge trigger
  - Turn on the interrupt in the `NVIC_ISER` (NB: The bits in the NVIC registers correspond to the interrupt position in the vector table).

12.3 Interrupt and exception vectors

The grey rows in *Table 46* describe the vectors without specific position.

Refer to device datasheet for availability of each peripheral.

Table 46. STM32L41xxx/42xxx/43xxx/44xxx/45xxx/46xxx vector table

Position	Priority	Type of priority	Acronym	Description	Address
-	-	-	-	Reserved	0x0000 0000
-	-3	fixed	Reset	Reset	0x0000 0004
-	-2	fixed	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000 0008
-	-1	fixed	HardFault	All classes of fault	0x0000 000C
-	0	settable	MemManage	Memory management	0x0000 0010
-	1	settable	BusFault	Pre-fetch fault, memory access fault	0x0000 0014
-	2	settable	UsageFault	Undefined instruction or illegal state	0x0000 0018
-	-	-	-	Reserved	0x0000 001C - 0x0000 0028
-	3	settable	SVCall	System service call via SWI instruction	0x0000 002C
-	4	settable	Debug	Monitor	0x0000 0030
-	-	-	-	Reserved	0x0000 0034
-	5	settable	PendSV	Pendable request for system service	0x0000 0038
-	6	settable	SysTick	System tick timer	0x0000 003C
0	7	settable	WWDG	Window Watchdog interrupt	0x0000 0040
1	8	settable	PVD_PVM	PVD/PVM1/PVM2(1)/PVM3/PVM4 through EXTI lines 16/35/36/37/38 interrupts	0x0000 0044
2	9	settable	RTC_TAMP_STAMP /CSS_LSE	RTC Tamper or TimeStamp /CSS on LSE through EXTI line 19 interrupts	0x0000 0048
3	10	settable	RTC_WKUP	RTC Wakeup timer through EXTI line 20 interrupt	0x0000 004C
4	11	settable	FLASH	Flash global interrupt	0x0000 0050
5	12	settable	RCC	RCC global interrupt	0x0000 005C
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 005C
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000 005C
8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000 0060
9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000 0064
10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000 0068

RM0394 p. 321

Use the reference manual and datasheet for the MCU to answer the following questions.

What bit (register name, address, and bit index) needs to be enabled to turn on the clock domain for accessing the system configuration registers (SYSCFGEN)?

`SYSCFGEN` is bit 0 in `RCC->APB2ENR` memory address `0x60` from base address of RCC which is `0x4002 1000`.

What register within the SYSCFG register needs to be configured to enable interrupts from your desired GPIO pin?

`SYSCFG_EXTICR<X>`. Need to find the correct bits within the register to configure the mux to pass through the desired GPIO pin.

Interrupts Worksheet Cont.

What is the assembly instruction used to globally enable interrupts? Can this instruction be replaced by a memory-mapped load/store operation?

`__enable_irq()` is defined as `__ASM volatile ("cpsie i" : : : "memory")`. These instructions are special instructions in the ISA and cannot be replaced by memory-mapped read/writes.

Interrupts Worksheet Cont.

What registers in EXTI need to be configured to trigger interrupts on the falling edge?

The interrupt mask register (IMR) and trigger selection registers (`EXTI_RTSR` and `EXTI_FTSR`).

What is the base address for the EXTI registers?

`0x4001 0400`

Interrupts Worksheet Cont.

What register needs to be configured in the Nested Vector Interrupt Controller to enable the interrupt?

Interrupt Set-Enable Register (`ISER1`) at address `0xE000E104`.

Interrupts Worksheet Cont.

What is the name of the interrupt handler for IRQHandler? It is defined as `<x>_IRQHandler` where is the name of the acronym in the vector table. Hint: look in the Vector table.

`<EXTI_Name>_IRQHandler`

Interrupts Worksheet Cont.

Inside the interrupt handler, what registers do we need to check to see if there is an interrupt pending? How do we reset it to clear the pending status?

We need to check the pending register `EXTI_PR`. To reset, we write the bit to 1 (a little strange, but true).

Timer Interrupt

Timer Interrupt Configuration

Configure `TIM6` to generate interrupts. Configure the interrupt handler to toggle an LED.