



Lab.8 STM32 Interrupt and Exception

0516076 施威綸

1. Lab objectives 實驗目的

- Understand how to set STM32 SysTick timer.
- Understand the setting and principle of NVIC and External

2. Steps 實驗步驟

2.1. SysTick timer interrupt setting

Set the clock source as HSI's SysTick timer, and use the SysTick timer interruption to darken the LED(one time for every three seconds).

- (a) Set up system clock configurations and configure systick
- (b) Set timer for 0.5 sec
- (c) Switch the LED on PA5

```
int main() {
    //NVIC_EnableIRQ(15);
    SystemClock_Config();
    SysTick_Config(749999); // sysTick reload value 250,000 * 3sec.
    GPIO_init();
    while(1) LED_switch();
}

void SystemClock_Config() {
    RCC->CR |= RCC_CR_HSION; // turn on HSI
    while(RCC->CR & RCC_CR_HSIRDY); // wait until HSI ready
    SET_REG(RCC->CFGR, RCC_CFGR_SW, RCC_CFGR_SW_HSI); // configure HSI as SYSCLK
    SET_REG(RCC->CFGR, RCC_CFGR_SWS, RCC_CFGR_SWS_HSI);
    SET_REG(RCC->CFGR, RCC_CFGR_HPRE, RCC_CFGR_HPRE_DIV64); // SYSCLK divided by 64
}
```

2.2. Multiple External Interrupt setting

Set the interruption as input source for input pin of keypad. LED will keep lighting when button is not pressed, otherwise program will be triggered into the handler. LED will flash the time you enter from the keypad in the handler(light and dark each 0.5 second), after that LED will back to the initial state.

- (a) Set up external interrupt handler for EXTI 10~15(GPIOx 10~15)
- (b) Set EXTI and NVIC configuration
- (c) Set interrupt priority between systick and EXTI
- (d) Scan the keypad input.



(e) Switch LED

```

void SysTick_init() {
    SysTick_Config(4000); // 1ms
    NVIC_SetPriority (SysTick_IRQn, 0); // set highest SysTick priority
}

void NVIC_config() {
    // IRQn = 40
    NVIC->IP[40] |= (uint8_t)0xF0u; // lowest interrupt priority
    NVIC->ISER[1] |= (uint32_t)0x00000100u;
    //NVIC->ISPR[1] |= (uint32_t)0x00000100u;
}

// interrupt handler implementation
void EXTI15_10_IRQHandler() {
    //EXTI->PR1 &= ~(EXTI_PR1_PIF10 & EXTI_PR1_PIF15 & EXTI_PR1_PIF14 & EXTI_PR1_PIF11);
    int n = keypad_scan();
    for(int i=0; i<n; i++) {
        GPIOA->BRR |= GPIO_BRR_BR5;
        timer = 1;
        while(timer != 0);
        GPIOA->BSRR |= GPIO_BSRR_BS5;
        timer = 1;
        while(timer != 0);
    }
    GPIOB->BSRR |= (GPIO_BSRR_BS10 | GPIO_BSRR_BS15 | GPIO_BSRR_BS14 | GPIO_BSRR_BS11);
}

void SysTick_Handler() {
    timer = (timer+1)%200;
}

void System_config() {
    RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;

    SYSCFG->EXTICR[2] |= SYSCFG_EXTICR3_EXTI10_PB;
    SYSCFG->EXTICR[3] |= SYSCFG_EXTICR4_EXTI15_PB;
    SYSCFG->EXTICR[3] |= SYSCFG_EXTICR4_EXTI14_PB;
    SYSCFG->EXTICR[2] |= SYSCFG_EXTICR3_EXTI11_PB;
}

void EXTI_config() {
    EXTI->IMR1 |= (EXTI_IMR1_IM10 | EXTI_IMR1_IM15 | EXTI_IMR1_IM14 | EXTI_IMR1_IM11);
    EXTI->RTSR1 |= (EXTI_RTSR1_RT10 | EXTI_RTSR1_RT15 | EXTI_RTSR1_RT14 | EXTI_RTSR1_RT11);
    //EXTI->PR1 |= (EXTI_PR1_PIF10 | EXTI_PR1_PIF15 | EXTI_PR1_PIF14 | EXTI_PR1_PIF11);
}

```

2.3. Alarm

Use SysTick timer, user button and buzzer to design a simple alarm clock. Take the keypad as input of the time for alarming, every single number is represented to the second(ex. 2 for 2 second). When input is zero, it won't have any response until next input come in. After enter your input, Systick timer will start counting and buzzer will alarm(using delay to make buzzer alarm, and you can set your own frequency) when the counting finish. After that, when user press user button, buzzer will stop alarming and wait for the next input. Note that keypad cannot work between start counting and user stop the buzzer.

- (a) Same as lab8.2.
- (b) Set buzzer.



3. Results and analysis 實驗結果與分析

3.1. SysTick timer interrupt setting

SysTick is a built-in timer with interrupt, if no special use but only counting and timing, SysTick is an easy way to do it.

3.2. Multiple External Interrupt setting

The systick configure function inside the API has done something not expected(set up the lowest priority). This took me so much time debugging. I have to look in detail every time using a new function.

3.3. Single and multi buttons

Nothing special.

4. Conclusions and ideas 心得討論與應用聯想

The buzzer & PWM in lab8-3 really drive me crazy, and took me a lot of time. This is actually from lab 7, maybe I should learn from my teammate. Unfortunately no one knows how it works and I had to read all the documents and references from the beginning.