

2020 Spring 微處理機 LAB 6 Interrupt

Due : 2020/05/20 早上 8:00

PART 1. (10%) 實作題

Lab 6.1.1 Timer interrupt – Change LED sparkling rate:

請完成實驗 錄影紀錄實驗結果並附上程式碼(main.c 及 include 之.h, .c 檔案)

以範例程式 Lab 6.1 為基礎，按按鍵切換閃爍頻率，由 0.5s -> 0.1s -> 0.3s，不可更改 clk rate。

```
// Check if need to change state
if(pos_cnt>debounce_threshold){
    if(last_button_state==0){
        // Pressed button - Pos edge
        SysTick->CTRL ^= (1 << SysTick_CTRL_ENABLE_Pos);
        x=1+x;
        //pressed 1 time, change freq to 0.1s bright led
        if(x==1){
            SystemClock_Config_Interrupt(10, 1000000);
            //x=1+x;
        }
        //pressed 2nd times, change freq to 0.3s bright led
        else if(x==2){
            SystemClock_Config_Interrupt(10, 3000000);
            //x=1+x;
        }
        //pressed third times, turn back to the initial freq and reset to 0.5s bright led
        else {
            x=3-x;
            SystemClock_Config_Interrupt(10, 5000000);
        }
    }
}
```

PART 2. (40%) 實作題

Lab6.2: Keypad external interrupt

請完成實驗 錄影及截圖紀錄實驗結果並附上程式碼(main.c 及 include 之.h, .c 檔案)

- 將 Lab6 中所實作的鍵盤掃描程式改成利用 SysTick 與外部中斷 EXTI 完成 (取代掃描迴圈)。
 - 將 Column output 掃描由 SysTick interrupt handler 完成，中斷時間間隔 0.1s，當 SysTick 中斷發生時更改 scan 的 column。
 - 在 SysTick interrupt handler 中設定並啟動 keypad row 的 4 個 input 腳的 EXTI 外部中斷。並使用負邊緣觸發(Negative trigger)。
 - 當 EXIT 中斷發生時讀取 4 個 input 的值，並根據目前 column 掃描狀態判斷是哪個鍵按下。
- 在主程式中依使用者所按下的按鍵值利用 lab6 的 display()顯示至 7 段顯示器上。

設定 interrupt:

```
void EXTI_Setup() {
    //Enable SYSCFG CLK
    RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;
    //Select output bits
    SYSCFG->EXTICR[0] &= ~SYSCFG_EXTICR1_EXTI3_Msk;
    SYSCFG->EXTICR[0] |= (1 << SYSCFG_EXTICR1_EXTI3_Pos);
    SYSCFG->EXTICR[1] &= ~SYSCFG_EXTICR2_EXTI4_Msk;
    SYSCFG->EXTICR[1] |= (1 << SYSCFG_EXTICR2_EXTI4_Pos);
    SYSCFG->EXTICR[1] &= ~SYSCFG_EXTICR2_EXTI5_Msk;
    SYSCFG->EXTICR[1] |= (1 << SYSCFG_EXTICR2_EXTI5_Pos);
    SYSCFG->EXTICR[1] &= ~SYSCFG_EXTICR2_EXTI6_Msk;
    SYSCFG->EXTICR[1] |= (1 << SYSCFG_EXTICR2_EXTI6_Pos);

    //Enable interrupt
    EXTI->IMR1 |= EXTI_IMR1_IM3;
    EXTI->IMR1 |= EXTI_IMR1_IM4;
    EXTI->IMR1 |= EXTI_IMR1_IM5;
    EXTI->IMR1 |= EXTI_IMR1_IM6;

    //Enable Falling Edge
    EXTI->FTSR1 |= EXTI_FTSR1_FT3;
    EXTI->FTSR1 |= EXTI_FTSR1_FT4;
    EXTI->FTSR1 |= EXTI_FTSR1_FT5;
    EXTI->FTSR1 |= EXTI_FTSR1_FT6;

    //Enable NVIC**
    NVIC_EnableIRQ(EXTI3_IRQn);
    NVIC_EnableIRQ(EXTI4_IRQn);
}
```

PART 3. (50%) 實作題

Lab6.3: 製作簡單鬧鐘

請完成實驗 錄影及截圖紀錄實驗結果並附上程式碼(main.c 及 include 之.h, .c 檔案)

利用 SysTick timer、User button 和喇叭設計一個簡單的鬧鐘，利用 keypad 輸入計時鬧鐘倒數時間並即時顯示至 7-Seg LED，數字代表設定幾秒(ex: 2 為 2 秒)，輸入為 0 時則沒反應，繼續等待下次輸入，按下 User button 則代表時間輸入完畢

- 啟動一秒觸發一次 interrupt 的 SysTick timer 開始倒數，利用 7-seg LED 顯示目前倒數的時間秒數，當時間到後，喇叭便會響起(在 SysTick interrupt handler 中利用 while loop 讓喇叭持續發出聲音，頻率自訂)
- 直到使用者按下 User button 後才會停止發出聲音並回到等待使用者輸入狀態，注意 SysTick 開始計時到使用者關閉喇叭的期間，keypad 不會有任何作用。(程式會由 user button 觸發一個 nested interrupt)

Note:

- 1.注意 SysTick timer 中斷和 User button 外部中斷的 Priority 關係。
2. SysTick clock source 設定為 10MHz
- 3.請練習如何設定 interrupt 的 priority

本作業參考自: DCP1155 Microprocessor System Lab 2016

曹孝櫟教授 國立交通大學 資訊工程學系 Lab7

時間到了就會響：

```
1      num_dot/=10;
2      }
3
4      for(int i = 0 ; i< num_d ;i++){
5          if(i==0)
6              send_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin, 3,0x80 + num%10 );
7          else
8              send_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin, i+3, num%10);
9              num/=10;
10         }
11         //
12         if(now_time==TIME_SEC){
13             timer_enable(TIM2);
14             input = 1;
15             timer_init(TIM2, 200000/784-1, 19); //4M/20/(200000/f)=f
16             break;
17         }
18     }
19 }
20 ~
```

