2020 Spring 微處理機 LAB 5 Clock and Timer

Due ： 2020/05/13 早上 8:00

PART 1. (10%) 實作題

Lab 5.1 Modify system initial clock:

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

˙在範例程式 Lab 5.1 中新增 20 M Hz 的 system clock 頻率, 讓 LED 閃爍頻率做出對應改變。

新增 20 M Hz 的 system clock 頻率:

```
void SystemClock_Config(int speed){
    // system clock -> MSI
    RCC->CFGR &= ~RCC_CFGR_SW_Msk;
    RCC->CFGR |= RCC_CFGR_SW_MSI;

    while(!(((RCC->CFGR & RCC_CFGR_SWS_Msk)>> RCC_CFGR_SWS_Pos) == 0));    // Make sure system clock is ready

    RCC->CR &= ~RCC_CR_PLLON;                // Disable PLL
    while((RCC->CR & RCC_CR_PLLRDY) != 0);    // Make sure PLL is ready (unlocked)

    // Set PLL to MSI
    RCC->PLLCFGR &= ~RCC_PLLCFGR_PLLSRC_Msk;
    RCC->PLLCFGR |= RCC_PLLCFGR_PLLSRC_MSI;

    // R / 0~3  2,4,6,8
    // N * 8~86 8~86
    // M / 0~7  1~8
    // initial
    int set_R=0, set_N=0, set_M=0;
    // Change R N M
    if(speed==40){
        set_R = 1;
        set_N = 40;
        set_M = 0;
    }
    else if(speed==20){
        set_R = 0;
        set_N = 10;
        set_M = 0;
    }
```

```
int main(){
    // Cause we want to use floating points we need to init FPU
    FPU_init();

#ifdef lab_modify_system_clock

    if(init_led(LED_gpio, LED_pin) != 0){
        // Fail to init led
        return -1;
    }
    if(init_button(BUTTON_gpio, BUTTON_pin) != 0){
        // Fail to init button
        return -1;
    }

    int speed=0, trans[6]={1, 6, 10, 16, 20, 40}; // Add a new 20M Hz CLCK SPEED
    SystemClock_Config(trans[speed]);
```

PART 2. (40%) 實作題

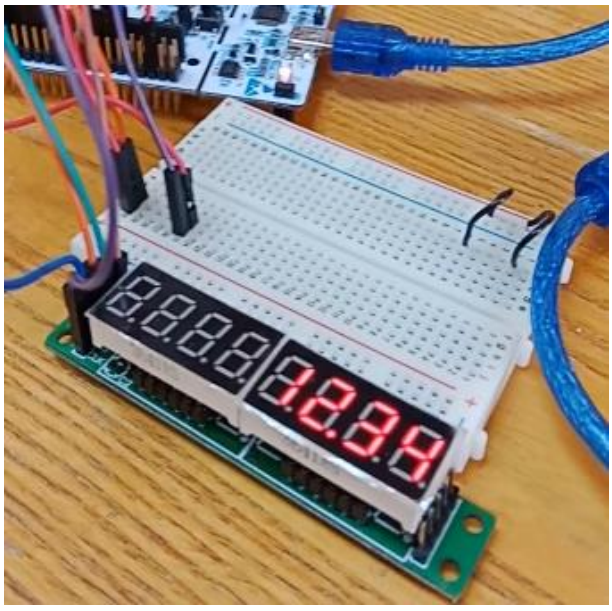
Lab 5.2 Timer

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

- 使用 STM32 timer 實做一個計時器會從 0 上數(Upcounting) TIME_SEC 秒(自訂)的時間。顯示到小數點以下第二位，結束時 7-SEG LED 停留在 TIME_SEC 的數字。(建議使用擁用比較高 counter resolution 的 TIM2~TIM5 timer), 取得 timer CNT register 值並換算成時間顯示到 7-SEG LED 上。
- 0.01 ≤ TIME_SEC ≤ 10000.00 (超過範圍請直接顯示 0.00)
- 舉例: TIME_SEC 為 12.7 時的 demo 影片： https://goo.gl/F9hh35
- Note: 7-SEG LED 驅動請利用之前 Lab 所實作的 GPIO_init()、7-segment_init()與 Display()等等函式呈現(須改成可呈現 2 個小數位)。

```
182 int main(){
183     FPU_init();
184     if(init_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin) != 0)
185     {
186         //Fail to init 7seg
187         return -1;
188     }
189     //Set Decode Mode to Code B decode mode
190     send_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin, SEG_ADDRESS_DECODE_MODE, 0xFF);
191     //Set Scan Limit to all digits
192     send_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin, SEG_ADDRESS_SCAN_LIMIT, 0x07);
193     //Wakeup 7seg
194     send_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin, SEG_ADDRESS_SHUTDOWN, 0x01);
195     //Set brightness
196     send_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin, SEG_ADDRESS_ITENSITY, 0x05);
197     //Clear the digits
198     for(int i=1;i<=8;i++){
199         send_7seg(SEG_gpio, DIN_pin, CS_pin, CLK_pin, i, 15);
200     }
201
202     double TIME_SEC = 12.34;
```

在 Part2 src 裡，我將其他.c 的 function 整合到 main.c 以方便整理。



PART 3. (40%) 實作題

Lab 5.3 電子琴

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

- 製作電子琴，用 Timer 製作出所需要的頻率波形並連到變頻喇叭發出 Do, Re, Mi 等音符，八個 keypad 按鍵 0~7 分別對應投影片中音階頻率表中第三度的 Do ~Si 和高音 Do。
-

注意

1.輸出正反變換一次為一個週期。或者可使用 PWM mode 輸出，參考 Reference Manual 26.3.11 PWM mode。

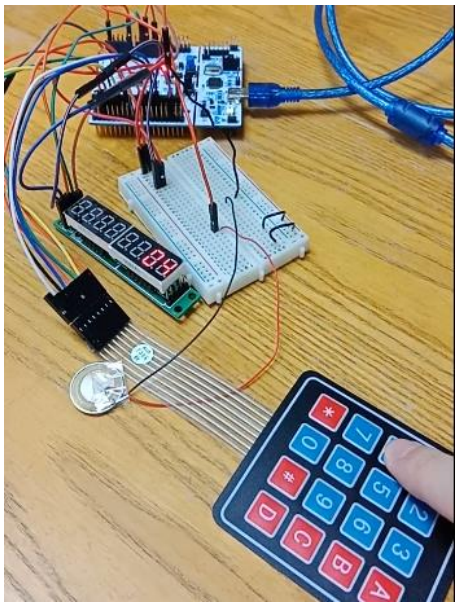
2.喇叭接 Vcc 和輸出接腳 or GND 和輸出接腳 都可。

	X0	X1	X2	X3
Y0	Do	Re	Mi	
Y1	Fa	So	La	
Y2	Si	HDo		
Y3				

Keypad 對應音名

音名	Do	Re	Mi	Fa	So	La	Si	HDo
頻率(Hz)	261.6	293.7	329.6	349.2	392.0	440.0	493.9	523.3

音名頻率對應表



```
int i;
while(1){
    display_number(SEG_gpio,DIN_pin,CS_pin,CLK_pin,0,0);
    int input = 0;
    for(int i=0;i<4;i++)
    {
        for(int j=0;j<4;j++)
        {
            if(check_keypad_input_one(ROW_gpio, COL_gpio, ROW_pin, COL_pin, i, j))
            {
                display_number(SEG_gpio, DIN_pin, CS_pin, CLK_pin,keypad[i][j],2);
                    if(octave == 3){
                        f = three[i][j];
                    }
                    timer_enable(TIM2);
                    input = 1;
                    timer_init(TIM2, 200000/f-1, 19); //4M/20/(200000/f)=f
                }
            }
```

PART 4. (10%) 問答題

1. 說明 Sysclk 和 timer 內 clk 差異 (5%)
   修改 Sysclk 會讓整個微處理機的運行速度有所改變, 而若只修改 timer 的 clk 只是讓 timer 內的計算有所改變，非影響全部的功能運算速度。

2. 說明如何在 ARM 中設定生成 PWM，參考 Reference Manual 26.3.11 PWM。
Pulse Width Modulation mode allows you to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register.
透過 TIMx_ARR register 和 TIMx_CCRx register 可以分別設定頻率和 duty cycle 以達到生成 PWM 的目的。

PART 4. (20%) 加分題

5.4. Music 音色實驗(15%)
請完成實驗 錄影紀錄實驗結果並附上程式碼(main.s 及 include 之 pin.s 檔案)
在前一實驗(Lab 5.3 電子琴)中的 keypad 增加 2 個功能按鈕用以調整 PWM 輸出的 Duty cycle(範圍 10%~90%，每按一次鍵調整 5%)，觀察是否會影響蜂鳴器所發出的聲音大小或音色。
Note: 須注意頻率與 duty cycle 的關係來設定 timer ARR 與 CCR registers。可用 LED 或者錄音測試 duty cycle 是否有改變, 成功應會看到 LED 隨著 duty cycle 不同而有明暗變化。

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