



## 實驗七 STM32 Clock and Timer

0516039 羅文笙

### 1. Lab objectives 實驗目的

瞭解 STM32 的各種 clock source 使用與修改

瞭解STM32的timer使用原理

瞭解STM32的PWM使用原理與應用

### 2. Steps 實驗步驟

#### (1) Modify system initial clock

利用先前lab所實作的assembly 版本GPIO\_init與delay\_1s初始化GPIO與delay。修改SYSCLK的clock source以及相關的prescaler使得CPU frequency(HCLK)為1MHz。當使用者按下user button便依以下順序改變CPU system clock(HCLK), 1MHz -> 6MHz -> 10MHz -> 16MHz -> 40MHz -> 1MHz ->...

#### (2) 計時器

完成main.c中的Timer\_init()與Timer\_start(),並使用STM32 timer實做一個計時器會從0上數(Upcounting) TIME\_SEC秒的時間。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。使用polling的方式取得 timer CNT register值並換算成時間顯示到7-SEG LED上。≤  
TIME\_SEC ≤ 10000.00 (超過範圍請直接顯示 0.00)

#### (3) Music keypad

請利用timer產生並輸出Duty cycle為50%的PWM訊號,並以Lab6中的keypad為鍵盤,當使用者在按下不同keypad按鍵時產生特定頻率的PWM方波給蜂鳴器,沒按鍵或按到沒功能的鍵時請不要發出聲音。本次實驗會需要設定



GPIOx\_AFRH、GPIOx\_AFRL、TIMx\_CCER、  
TIMx\_CCMR1、TIMx\_CCR1...等registers。

(3)-1 Music 音色實驗 在上一實驗中的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。

### 3. Results and analysis 實驗結果與分析

#### (1) Modify system initial clock

一開始先將 SYSCLK設成HSI16，再用system的prescaler將16MHZ除以16，將系統的CLK設定成1MHZ，之後再利用PLL1,PLL2,PLL3來設定頻率，公式為:PLL1/PLL2/PLL3

#### (2)計時器

將timer 設定好upcounting, psc 設成40000(-1)，arr設成100(-1)，如此一來就可以得到一個100HZ的波可以使timer每0.01秒做一次count的+1，來讓小數點後的數字直接顯示在7段顯示器上，當數到arr的上限的時候，TIMx->SR的最低位元會顯示為1，當程式如果讀到這邊為1時，就將秒數+1，並將該位數顯示到7段顯示器上，另外，需要注意的是，要顯示小數點，所以在第三位（也就是秒數的最低位）的時候將要傳送給max7219的值和0b10000000做bitand，來將D7給打開，如此一來就可以顯示小數點了。

#### (3)Music keypad

各自設定CCMR,CCER,CCR1，先將CCMR設成PWN Mode 1,然後用CCER enable CCR1並且設成Output mode,將 CCR1設成50,然後根據所對應使用的timer設定AFR register,根據每一個按鍵要輸出的頻率,計算按到按鍵的時候PSC的變化,PSC的計算方式為  $(ARR * PSC / 40000000) = (1 / \text{頻率})$ ,當按下按鍵後就



打開timer，讓蜂鳴器發出對應的頻率，當沒下按鍵時就關掉timer，這樣蜂鳴器就不會繼續叫了。

### (3)-1 Music 音色實驗

將按鍵設定在‘\*’和‘#’上，\*字號用來將聲音給調大，而#字號則用來將聲音給調小，調大的意思就是將CCR1+5而調小就是將CCR1-5。

## 4. Conclusions and ideas 心得討論與應用聯想

這次要設定的東西很多，不只是要將程式寫出來，還要事先設定好硬體設備，我覺得這次的作業份量有點大，如果可以給兩個禮拜的時間話，我覺得會比較好，這次的作業內容大概就是我們這組最後的期末作業的概念，所以也幸好有這次的lab作業，不然期末作業一開始也真的是不知從何下手。