

微處理機 Lab8 Report

一、 實驗名稱：STM32 Interrupt and Exception

二、 實驗目的：

- 瞭解 STM32 SysTick timer 設定。
- 瞭解 STM32 NVIC 和 External interrupt 設定和原理。

三、 實驗步驟：

1. SysTick timer interrupt & DS18B20 & One wire protocol。
2. Keypad external interrupt。

四、 實驗結果與分析：

1. SysTick timer interrupt & DS18B20 & One wire protocol：

第一題算是兩題中較難的一題，我們覺得它難的地方就在於 one-wire protocol 的實作以及數位溫度計的操作，實作的原理其實不會太難，講義上都有講解，唯一讓我們卡住的點在於在 reset 的最後，除了要等 60ns 去收 input 值外，還要另外等大概 420ns，讓整個 reset 的過程完整，一開始忘了做這部份，導致初始化沒有成功，困擾我們很久。而另外一個問題就是 delay 的問題，這邊要 delay 的時間都是微秒級，況且我們是使用預設的 system clock rate(4Mhz)，只要 4 個 cycle 就 1 微秒了，加上我們一開始是用呼叫函數的方式在操作 GPIO，導致常常指令做完，那個資料維持的時間或規定的時間長度也跑掉了，為了解決這個問題，我們只好放棄使用函數的方式，改為類似組語那樣的寫法直接去存取 GPIO 相關的暫存器，讓必要的存取不要浪費太多的 clock cycle 導致無法正確讀取到資料。

因為溫度變化不明顯的關係，我們也採取輸出小數點的方式，不過這邊有個問題卡住我們就是當我們要做浮點數運算時，不知為何的都會被 trap 到無窮迴圈內，最後才發現，要把 FPU unit 開啟，不然做浮點數運算一定會 trap 到無窮迴圈內，因為 CPU 認不得那些指令。

2. Keypad external interrupt :

這題算是兩題中比較簡單的一題，這題的做法是一開始先將 SysTick 給開起來，並設定好它所需的 clock source 以及 reloader 的值。這邊一開始遇到的困難的是找不到 SysTick 相關的 register，最後找了資料才發現 SysTick 以及 NVIC 都是處理器內建的 register，難怪在 STM32 的 reference 找不到。接著就要設定 EXTI 模組，這邊我們一共使用四個 EXTI 模組(EXTI12, EXTI13, EXTI14, EXTI15)，要設定 EXTI 模組須先開啟 SYSCFG 的功能(經由 APB2ENR 來開啟)，接著選各個 EXTI 模組中斷的 source port，接著設定 EXTI 的 IMR register，讓我們想開啟中斷的 EXTI 模組不會被 mask 掉，再來設定什麼時候會 trigger 此中斷，我們這邊設定的是上緣觸發，當電壓被拉上來時會觸發中斷發生。最後是設定 NVIC 以及把中斷 routine 填在 startup_stm32.s 中，設定 NVIC 讓我們設定的中斷可以被 CPU 看到以及將 EXTI 中斷 routine 填在正確的位置，這邊我們也卡了很久，最後發現是沒有將 routine 填在正確位置，填在正確位置及設好優先級後就可以執行了。

在執行的時候我們發現了另外一個問題，就是當我們按某個按鍵後，常常顯示出來的值都不是正確數值，而是跟它同一個 row 的其他按鍵數值，一直不懂該怎麼解決這個問題以及問題發生的原因，我們的改善方法是把 Display 這個函數放在 EXTI_handler 裡面，只有當中斷發生時才執行，這樣顯示錯誤的機率就會降低不少，

不過可能還需要仔細了解這個問題的成因才會有更好的解決辦法。

五、心得討論與應用聯想：

這次的作業雖然時間給比較多，但因為最近作業太多，所以我們仍然也只有幾天能夠做，先審題後，發現第二題比較簡單，所以先做第二題，只要開好 Interrupt 表後，大致上即可使用，但我們卻發現數字顯示出來是同一排的其他數字，因此判斷是又被 systick 占掉，所以要讓它不能用，所以便把寫值這件事寫在 Interrupt 內；另一題因為溫度計的協議較複雜，且 GPIO 的函式好像也不能直接使用，所以實際上也 DEBUG 了許久，好不容易讀到值了，卻也發現值不太對，最後再修正許多後，終於完成。