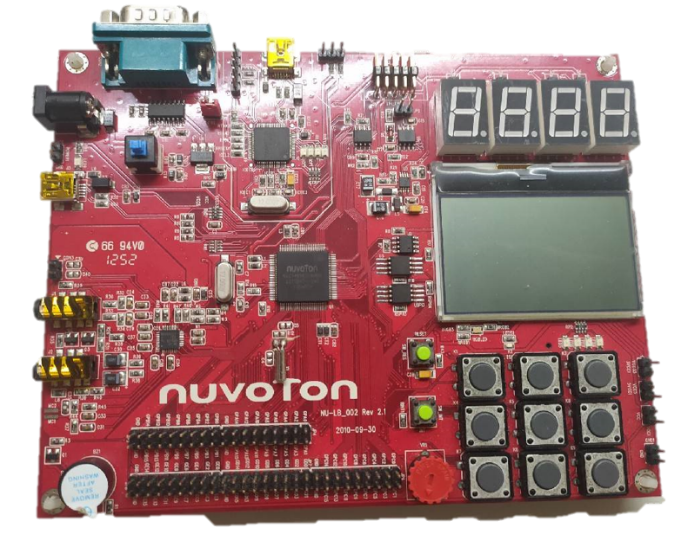
微處理機系統與介面技術 LAB 4

系所：通訊三 學號 :409430030 姓名:翁佳煌

**<實驗器材>**

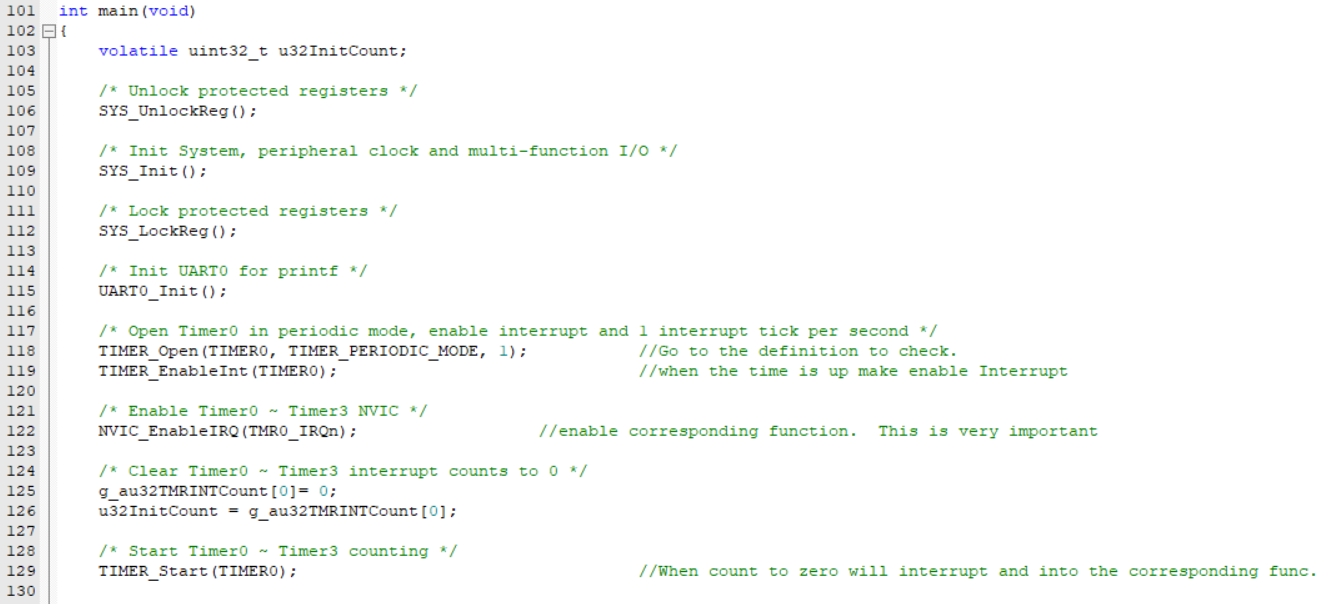
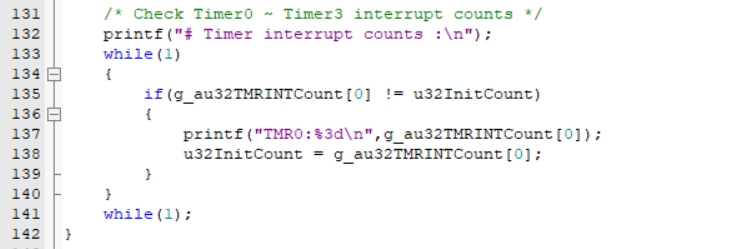
NUC 140 開發板



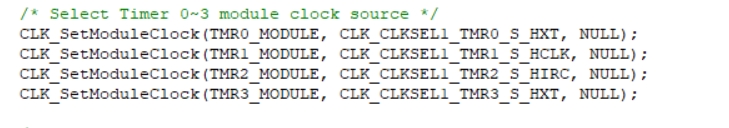
PL2303 USB to UART線

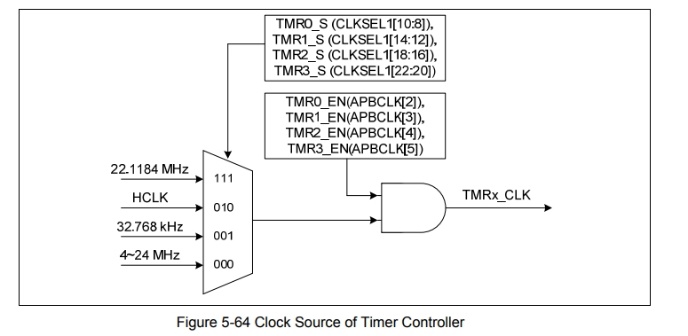
**<實驗過程與方法>**

首先實作Basic的部份，下**圖1.1**中，main中呼叫SYS\_Init()，一開始必然要先使用CLK\_EnableModuleClock(TMR0\_MODULE)(在此以TWM0為例子)，此外，這函式中寫到的CLK\_SetModuleClock(**圖1.2)，**是負責選擇timer的clock source，要注意不能選錯，否則之後的計數會錯誤。回到**圖1.1**中第118行TIMER\_OPEN，裡面三個參數分別放timer(在此實驗為TIMER0)、Mode(在此實驗為連續模式)、Freq(在此實驗為1)，使用Go to definition去查看的話可發現(**圖1.3**)此處為決定如何計數與prescale是如何算出1秒的部分。回到**圖1.1，**第119行則是設置當時間數完時Enable中斷，而第122行則是設置中斷發生時對應的function。第129行TIMER\_Start則是開始計數，當數到0時，觸發中斷。中斷發生後會跑到**圖1.4**第32行的函式**，**第37行為負責清除設置起來的Interrupt\_Flag，以便下次中斷發生時再次使用，第39行則是負責儲存每秒計數的值，而回到**圖1.1**的第133~142行，此部分為負責印出每秒count的數值。

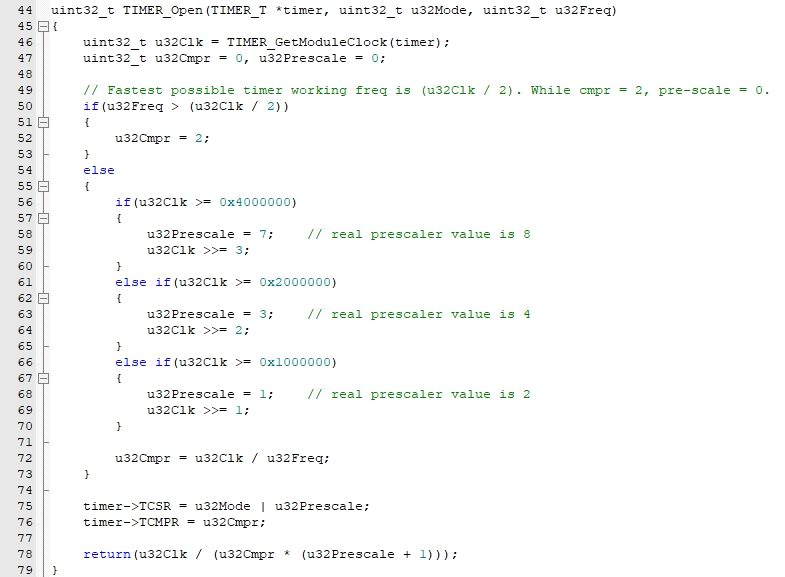
▼圖1.1

▼圖1.2





▼圖1.3

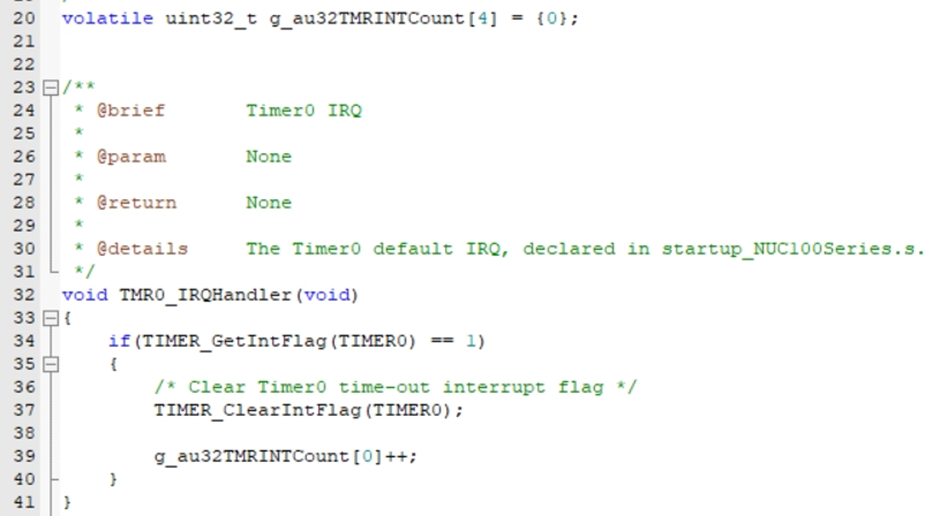
****

TIMER\_ONESHOT\_MODE(One–Shot模式)：計數只會進行一次，數到0後就發送中斷訊號。

TIMER\_PERIODIC\_MODE(Periodic 模式)：計數不斷循環，每數到0就發送中斷訊號。

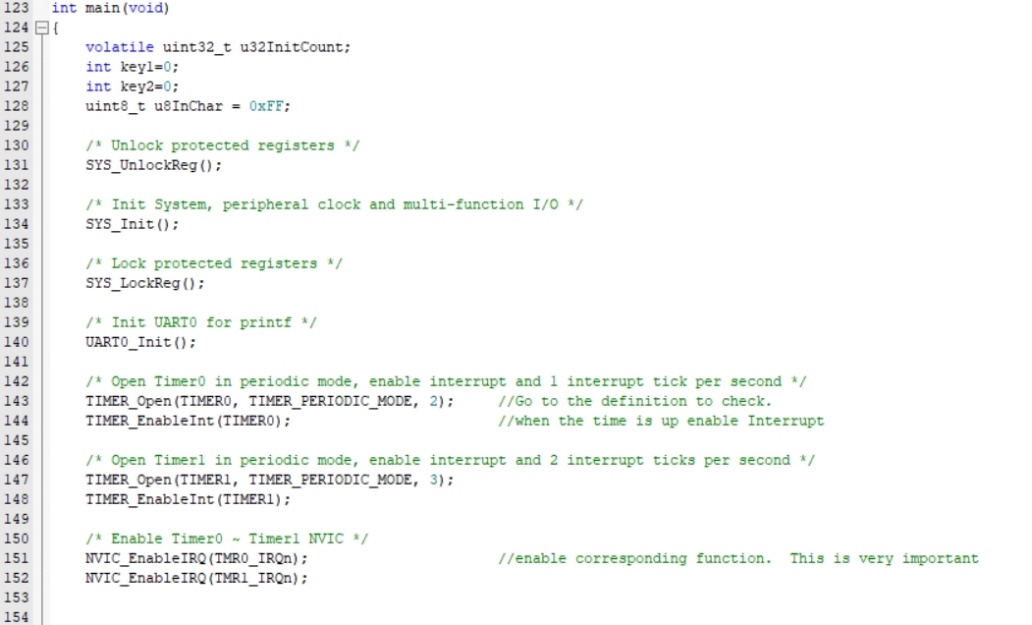
TIMER\_TOGGLE\_MODE (Toggle 模式): 設定CEN的值來控制計數器是否要持續計數，1 就是持續0則反之。在這狀態下每次數到閥值就會發送一個中斷訊號，然後回到初始值再重新數一次。只要interrupt flag被清除，TIMER就會回到初始值重新進行計數。

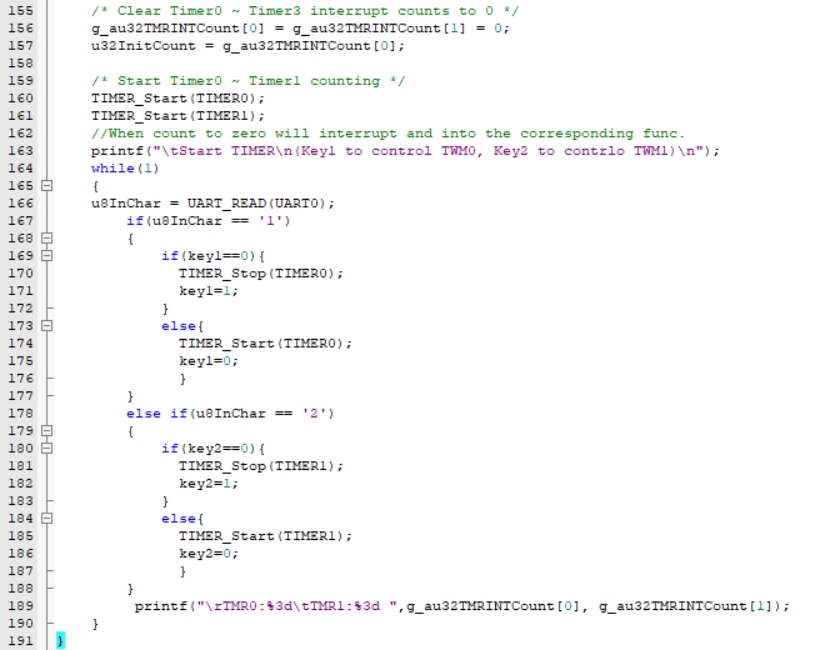
▼圖1.4



再來為Bonus的部分，這裡的概念與Basic的部分幾乎一樣，只是多了一個Timer和控制TIMER的部分，首先如下**圖1.5**第143以及第147行，TIMER\_Open裡面，頻率的部分，前者選擇一秒數2次，後者為一秒數3次，第160~161行開始兩個TIMER的計數，當數到0時發生中斷(這部分與Basic相似)。至於控制TIMER的部分，首先第126~127行我先設置兩個變數key1與key2為0，用來表示尚未按下按鍵，接著再第166行讀入使用鍵盤所輸入的字元，如果輸入為1，且如果key1為0，則TIMER暫停，如果key1為1則繼續開始計數，輸入為2則與上述同理。

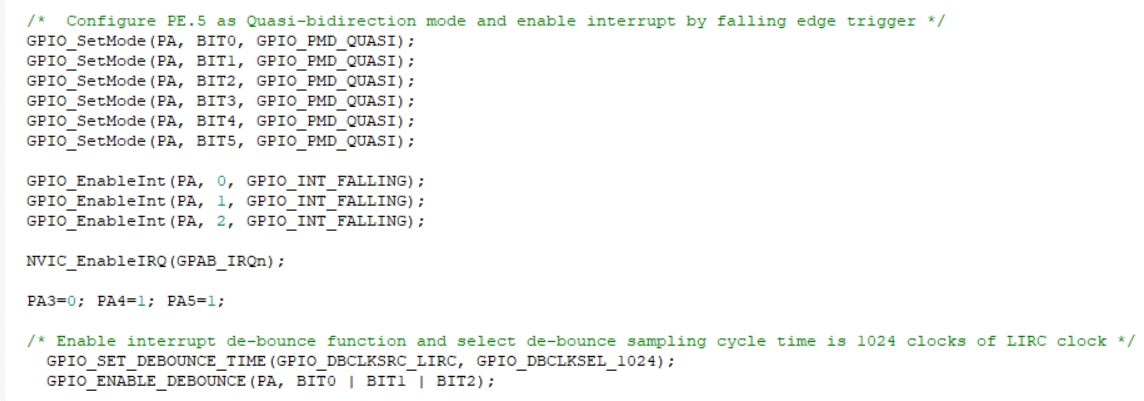
▼圖1.5

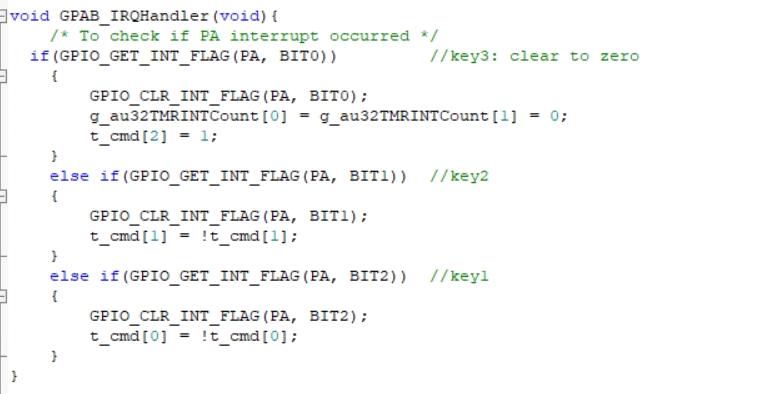
****

****

除了以上控制方法，還可使用九宮格的按鈕，設置GPIO，利用按鈕按下來發生中斷，如下**圖1.6**，首先要先設置GPIO\_SetMode，Quasi-bidirection為雙向模式，GPIO\_EnableInt則是當falling edge trigger發生時enable中斷，換句話說，當按下按鈕，也就是中斷發生時，會進入GPAB\_IRQHandler函式裡頭，這部分為負責控制TIMER的暫停與繼續計數。

▼圖1.6

****

****

**<心得與收穫>**

這次的主題我認為最難理解的部分就是他如何算出1秒，雖然應用很簡單，只需要去手動更改數值就可以，但其背後的運算我看了Datasheet的內容，也還是一知半解的感覺，實在是讓我頭痛。另外，這次的實驗與我們這組要報告的PWM有部分名詞有些重疊，像是當初看到timer、prescaler等等的名詞都不太懂他們到底負責做什麼，但做完這次的LAB也有初步的認識與了解。此外，在Bonus的部分，我誤以為是要用電腦鍵盤上的數字鍵1和2來控制，但後來有重新理解，並運用MCU上九宮格的按鈕來設置兩個GPIO控制TIMER，GPIO的部分原來可以如此深度的應用，我這才發現我對GPIO許多模式不太熟悉，看來必須再回頭看這部分的資料。