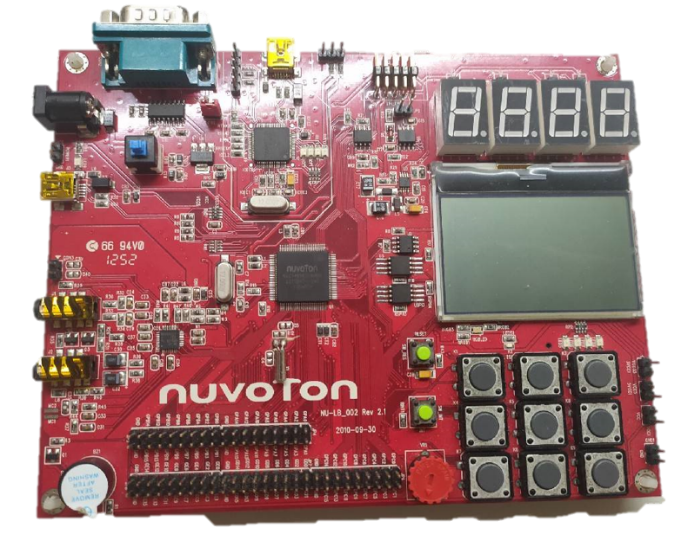
**嵌入式作業系統 LAB 5**

系所：通訊四 學號 :409430030 姓名:翁佳煌

**<實驗器材及環境>**

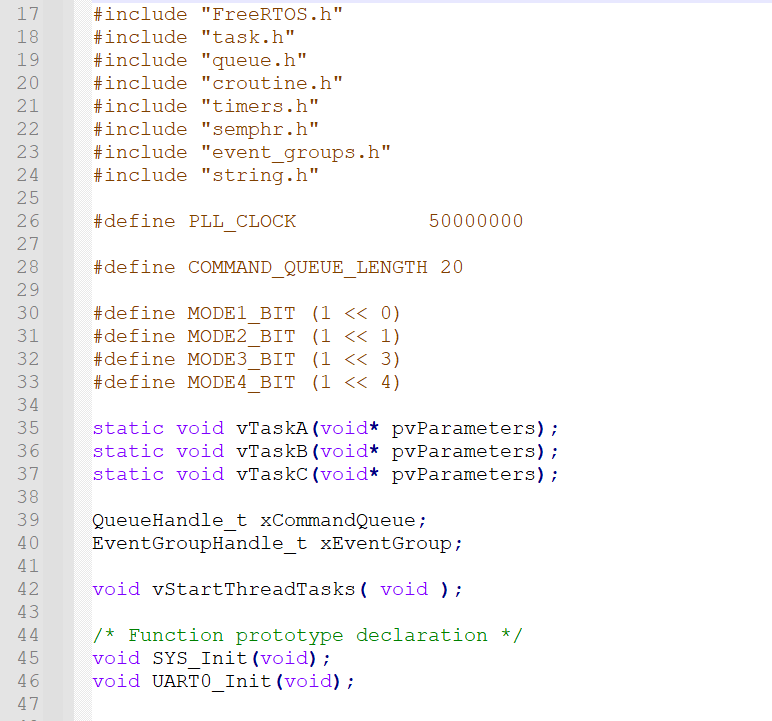
NUC 140 開發板 FreeRTOSv10.4.1



**<實驗過程與方法>**

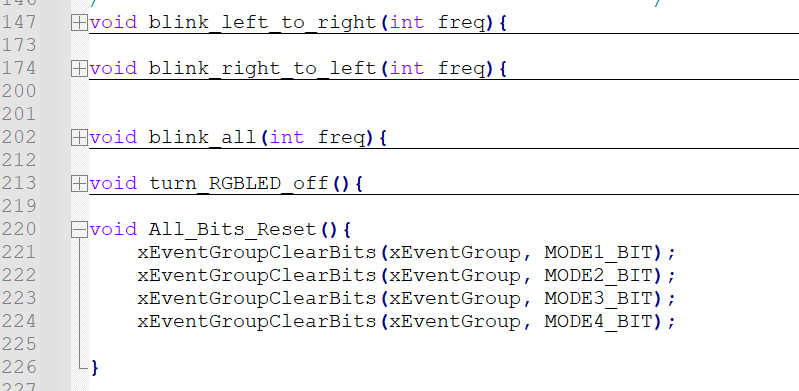
下圖1首先要記得include第23行的event\_group.h，這次LAB將會使用到這個函式庫，而其餘則和之前的LAB實驗相同。

第30~33行，MODE1\_BIT、MODE2\_BIT、MODE3\_BIT、MODE4\_BIT: 定義了事件群組（xEventGroup）中不同模式或標誌的位元表示。例如，MODE1\_BIT代表位元0，MODE2\_BIT代表位元1，MODE3\_BIT代表位元3，MODE4\_BIT代表位元4。



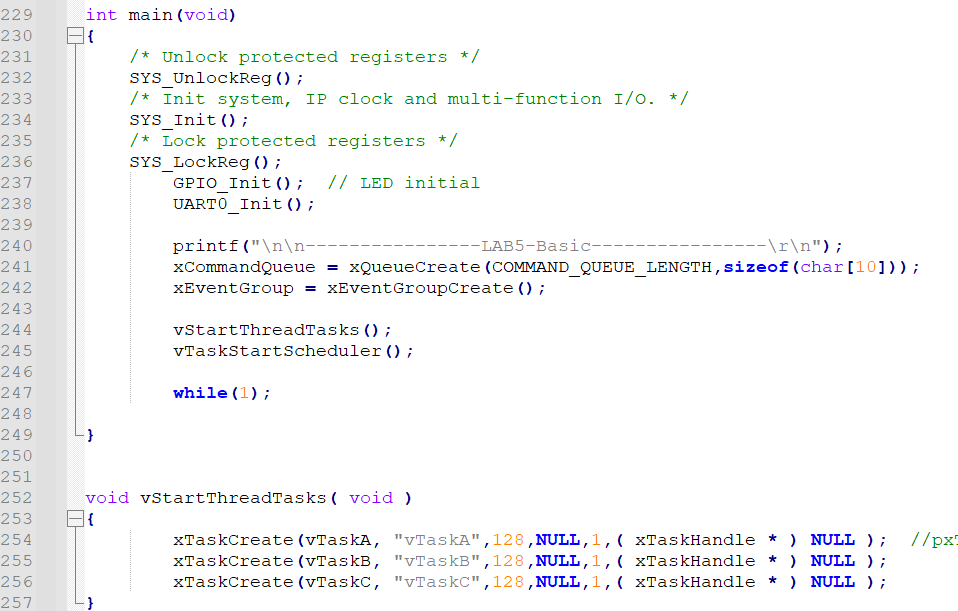
▲圖1

下圖2，147~213行的函式在前幾此LAB都有提到，因此這裡不多加贅述，值得注意的是220行的函式，當呼叫 All\_Bits\_Reset() 函數時，它會使用xEventGroupClearBits() 函數來清除事件群組中相應位元的值，將其設置為零。這可以在需要重置或清除所有模式或標誌的位元時使用，以確保它們處於初始狀態，不再處於任何已設置的狀態。



▲圖2

下圖3為main函數，xEventGroup = xEventGroupCreate(); 創建一個事件群組 xEventGroup。vStartThreadTasks(); 呼叫函數 vStartThreadTasks()，用於創建並啟動FreeRTOS任務。vTaskStartScheduler(); 開始 FreeRTOS 任務調度器。



▲圖3

下圖4這個 vTaskA 是一個 FreeRTOS 任務，負責處理命令的輸入、解析和執行相應的操作。

char command[10];：建立一個長度為 10 的字元陣列，用來存放用戶輸入的命令。

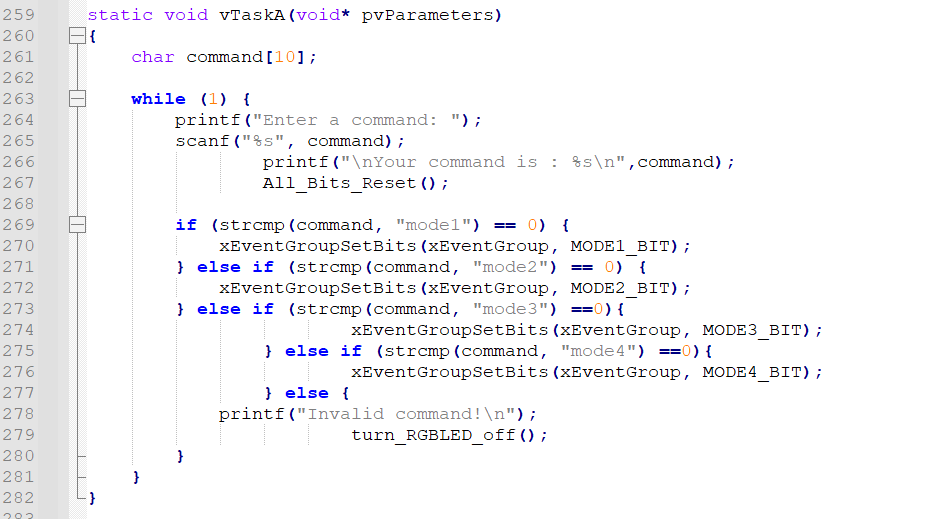
主循環：

如果用戶輸入的命令是 "mode1"，則呼叫 xEventGroupSetBits(xEventGroup, MODE1\_BIT); 函數，設置事件群組 xEventGroup 中的 MODE1\_BIT 位元。

如果命令是 "mode2"、"mode3" 或 "mode4"，分別設置對應的事件群組位元。

如果命令不符合任何已知命令，則顯示 "Invalid command!" 訊息，並呼叫 turn\_RGBLED\_off(); 函數。

此函數將持續等待用戶輸入命令，解析並根據命令執行相應的操作。如果命令是已知的 "mode1"、"mode2"、"mode3" 或 "mode4"，則會設置相應的事件群組位元，否則會印出無效命令並執行特定操作。



▲圖4

下圖5這個函數 vTaskB 是另一個 FreeRTOS 任務，負責根據事件群組中的位元狀態執行相應的操作。

EventBits\_t bits = xEventGroupWaitBits(xEventGroup, MODE1\_BIT | MODE2\_BIT | MODE3\_BIT | MODE4\_BIT, pdFALSE, pdFALSE, portMAX\_DELAY);：它會等待事件群組 xEventGroup 中的指定位元，其中 portMAX\_DELAY 表示等待時間無限期，直到有指定的位元被設置。

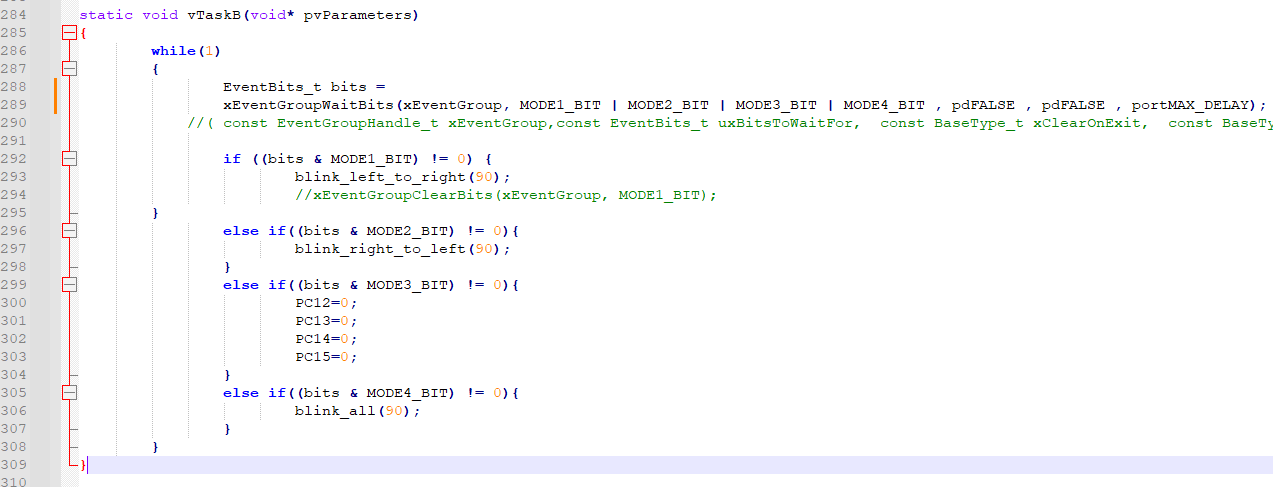
如果 MODE1\_BIT 位元被設置，則調用 blink\_left\_to\_right(90); 函數來執行LED向右閃爍的動作。

如果 MODE2\_BIT 位元被設置，則調用 blink\_right\_to\_left(90); 函數來執行LED向左閃爍的動作。

如果 MODE3\_BIT 位元被設置，則將所有LED持續亮起。

如果 MODE4\_BIT 位元被設置，則調用 blink\_all(90); 函數來執行所有LED一起閃爍的動作。

此任務持續監視事件群組中的位元狀態，並根據不同的狀態執行相應的操作。



▲圖5

下圖6這個函數 vTaskC 是另一個 FreeRTOS 任務，與前面的TaskB相似，根據事件群組中的位元狀態執行相應的操作。

EventBits\_t bits = xEventGroupWaitBits(xEventGroup, MODE1\_BIT | MODE2\_BIT | MODE3\_BIT | MODE4\_BIT, pdFALSE, pdFALSE, portMAX\_DELAY);：等待事件群組 xEventGroup 中的指定位元，portMAX\_DELAY 表示等待時間無限期，直到有指定的位元被設置。

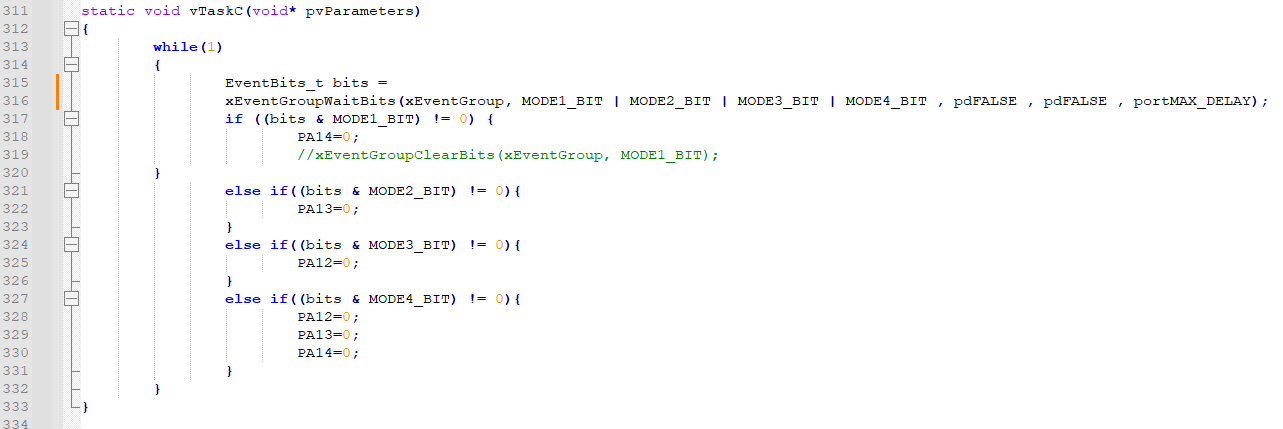
如果 MODE1\_BIT 位元被設置，則將 PA14 設置為0。

如果 MODE2\_BIT 位元被設置，則將 PA13 設置為0

如果 MODE3\_BIT 位元被設置，則將 PA12 設置為0。

如果 MODE4\_BIT 位元被設置，則同時將 PA12、PA13 和 PA14 設置為0。

此任務持續監視事件群組中的位元狀態，並根據這些位元的狀態執行不同的操作。這些操作是控制特定的GPIO腳位，從而影響硬體的行為或狀態。



▲圖6

**<心得與收穫>**

這次LAB5是最後一次嵌入式系統這堂課的實作作業，我從中學習到Event Group的用法以及他的好處，通過 Event Group，你能夠將不同的事件綁定在一起，以便任務在某些特定事件發生時進行同步操作。這種機制為嵌入式系統提供了彈性和準確性，同時也提高了系統的效率和可靠性。

最後謝謝助教以及老師這學期的認真教學和付出，讓我收穫良多。