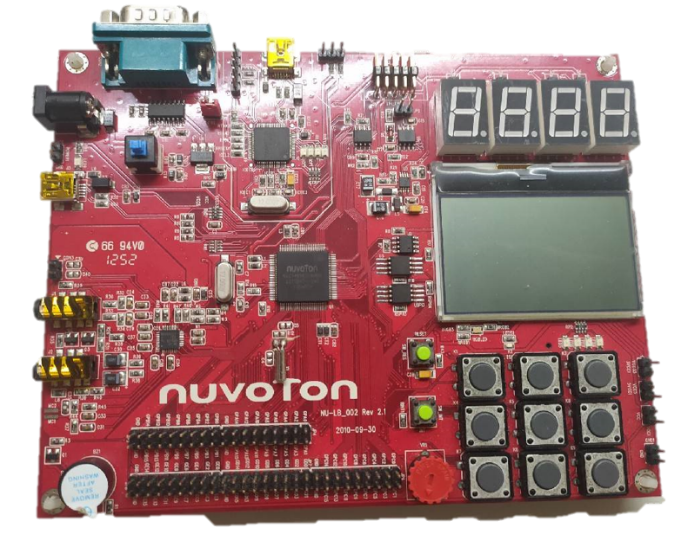
**嵌入式作業系統 LAB 1**

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

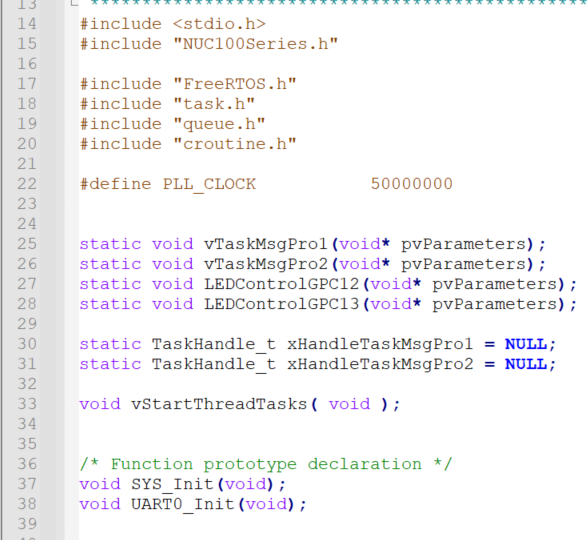


**<實驗過程與方法>**

下**圖1**的14~23行，程式開始時引入了必要的標頭檔。這些標頭檔通常包括系統標頭檔、FreeRTOS 相關標頭檔和自定義標頭檔，以及PLL 運行時的時脈頻率，這將用於系統時鐘設定。

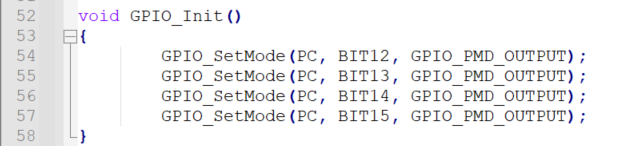
再來第25~28行，這一部分宣告了多個靜態函數，這些函數將作為 FreeRTOS 任務運行。

SYS\_Init以及UART0\_Init都是做簡單的初始化動作，這在微處理的課程中多次提到，這邊就不再贅述。



▲圖1

下**圖2**的部分是因為Bonus指定要控制LED閃爍，因此這個函數是用來初始化嵌入式系統中的 GPIO（通用輸出/輸入）引腳，以確保這些引腳的正確操作模式。在這個函數中，使用了 GPIO\_SetMode 函數，來設置多個引腳（PC12、PC13、PC14 和 PC15）的操作模式為 GPIO\_PMD\_OUTPUT，這意味著它們將被配置為數字輸出引腳，以便後續的 LED 控制。

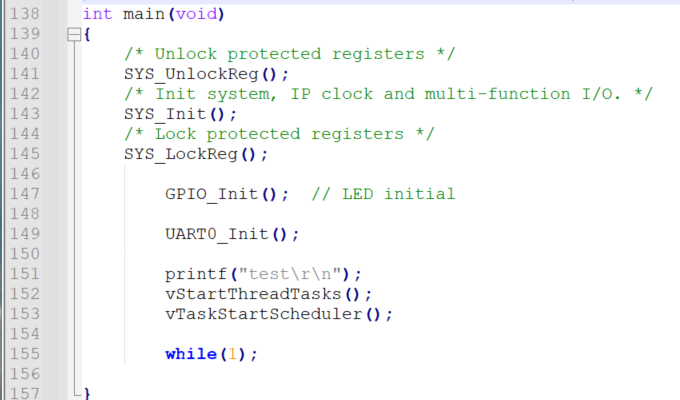


▲圖2

接下來看到下**圖3**的138~157行的main主程式，**SYS\_UnlockReg()**是用來解除保護註冊的鎖定，以便後續的設定和初始化操作能夠存取需要的系統註冊。在某些嵌入式系統中，某些系統註冊可能會被鎖定，以防止不必要的更改。**SYS\_Init()**用於初始化嵌入式系統的主要設定，包括系統時脈設定、外部硬體設定等。這個函數的目的是設定系統的運行環境，以確保系統能夠正確運行。**SYS\_LockReg()**這一行程式碼是用來重新鎖定保護註冊，以確保系統在進行運行時不會被不必要地修改。這通常在系統初始化後調用，以提高系統的安全性。**GPIO\_Init()**在上**圖2**中有提到，用於初始化 GPIO（通用輸出/輸入）引腳的設定。

再來看到第152行的**vStartThreadTasks()，**用於初始化和創建多個執行緒（task）或任務。這些執行緒將在作業系統中運行，執行不同的任務。這些執行緒是使用 FreeRTOS 函數庫創建的。

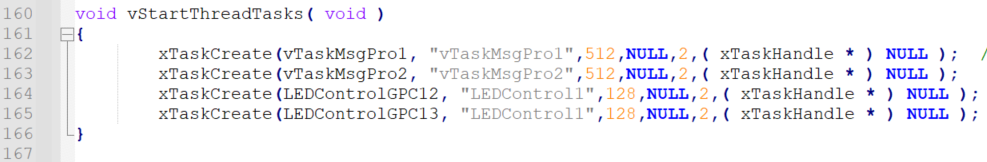
第153行的**vTaskStartScheduler()，**這是 FreeRTOS 函數庫的一個函數，用於啟動作業系統的調度器。一旦調度器啟動，將開始執行各個執行緒，並按照其優先級執行。

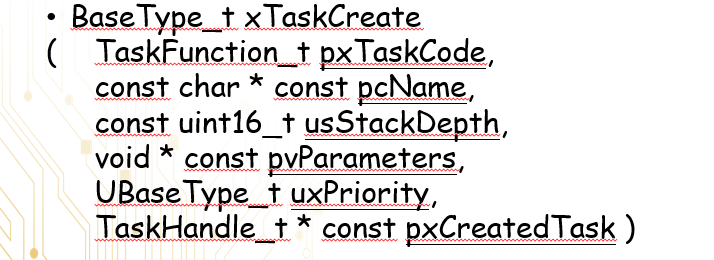


▲圖3

下**圖4**的160~166行，是使用 FreeRTOS 函數庫創建和管理多個執行緒（task）或任務，以實現並行執行不同的任務。這裡創建了四個不同的執行緒，分別為 vTaskMsgPro1、vTaskMsgPro2、LEDControlGPC12 和 LEDControlGPC13。這四個執行緒將在相同的優先級下運行。

xTaskCreate()會創建一個新的執行緒，並指定其執行的函數（例如 vTaskMsgPro1）、名稱、stack深度、參數、Priority以及一個指向執行緒的指標。





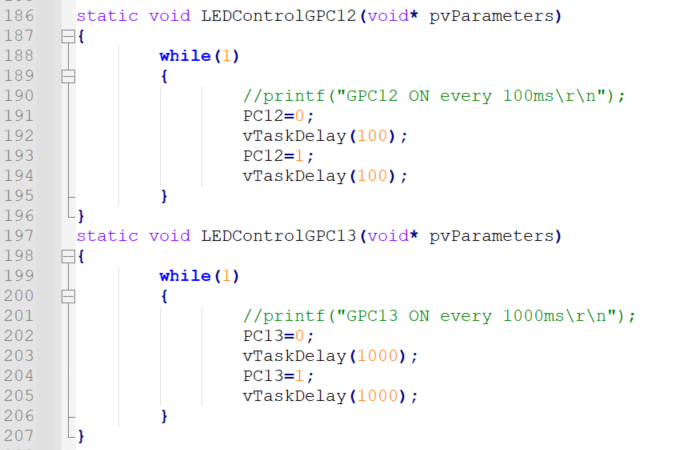
▲圖4

下**圖5**的168~184行是實作LAB1中Basic的重要部分，vTaskMsgPro1和vTaskMsgPro2是執行緒函數，不斷地執行一個無窮迴圈，分別每次都會輸出 "Task1 -> 500ms"以及"Task2 -> 1000ms"，然後使用 vTaskDelay 函數使其暫停 500ms和1000ms。



▲圖5

下**圖6**的186~207行，是Bonus控制LED的主要部分，LEDControlGPC12和LEDControlGPC13執行緒分別控制 PC12以及PC13 引腳，分別為每隔100ms和1000ms切換引腳狀態，即 ON 和 OFF，達到LED閃爍的效果。



▲圖6

**<心得與收穫>**

由於之前修過微處理與介面系統設計和作業系統相關課程，這次的 LAB1 在Code方面並未遇到太大的困難。主要問題是一開始建環境時遇到一些困難，感謝助教願意撥出時間指導我，解決了我在環境配置方面的問題，這使我能夠順利完成本次 LAB。希望在未來的學習和實驗中繼續積累更多的知識和經驗，並不斷提升自己的技能。