

嵌入式作業系統 LAB 4

系所：通訊四

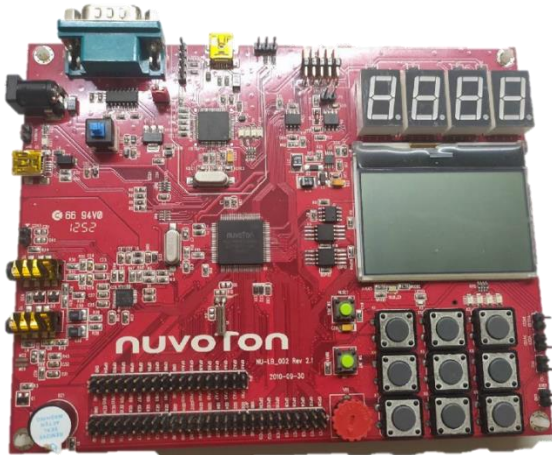
學號：409430030

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>

Basic:

首先下圖 1，26~27 行宣告了 `vTaskPrintParkingSpaces` 和 `CountParkingSpaces`，前者負責列印停車位資訊，後者用於計算停車位的數量。

29 行，宣告了一個名為 `parkingSemaphore` 的信號量變數，初始值為 `NULL`。

30 行，宣告了一個整數變數 `parking_spaces`，初始值為 2，代表停車位的數量。

```
14 #include <stdio.h>
15 #include "NUC100Series.h"
16
17 #include "FreeRTOS.h"
18 #include "task.h"
19 #include "queue.h"
20 #include "croutine.h"
21 #include "timers.h"
22 #include "semphr.h"
23
24 #define PLL_CLOCK      50000000
25
26 static void vTaskPrintParkingSpaces(void* pvParameters);
27 static void CountParkingSpaces(void* pvParameters);
28
29 SemaphoreHandle_t parkingSemaphore=NULL;
30 int parking_spaces=2;
31
32 void vStartThreadTasks( void );
33
34 /* Function prototype declaration */
35 void SYS_Init(void);
36 void UART0_Init(void);
37
```

▲圖 1

下圖 2 的部分為處理 Debounce 的部分，在 LAB3 中有詳細提到，這裡不多加贅述。

```
135 void Debounce(){
136     /* Configure PE.5 as Quasi-bidirection mode and enable interrupt by falling edge trigger */
137     GPIO_SetMode(PA, BIT0, GPIO_PMD_QUASI);
138     GPIO_SetMode(PA, BIT1, GPIO_PMD_QUASI);
139     GPIO_SetMode(PA, BIT2, GPIO_PMD_QUASI);
140     GPIO_SetMode(PA, BIT3, GPIO_PMD_QUASI);
141     GPIO_SetMode(PA, BIT4, GPIO_PMD_QUASI);
142     GPIO_SetMode(PA, BIT5, GPIO_PMD_QUASI);
143
144     GPIO_EnableInt(PA, 0, GPIO_INT_FALLING);
145     GPIO_EnableInt(PA, 1, GPIO_INT_FALLING);
146     GPIO_EnableInt(PA, 2, GPIO_INT_FALLING);
147     //NVIC_EnableIRQ(GPAB_IRQn);
148
149     PA3=0; PA4=1; PA5=1;
150     /* Enable interrupt de-bounce function and select de-bounce sampling cycle time is 128 clocks of LIRC clock */
151     GPIO_SET_DEBOUNCE_TIME(GPIO_DBCLKSRC_LIRC, GPIO_DBCLKSEL_128);
152     GPIO_ENABLE_DEBOUNCE(PA, BIT0 | BIT1 | BIT2);
153 }
```

▲圖 2

接著看到下圖 3 的 main 函數部分，看到第 171 行，

parkingSemaphore = xSemaphoreCreateCounting(5, 2)，其創建了一個計數型信號量 parkingSemaphore，初始值為 5，剩下 2 個空位。

vStartThreadTasks()，啟動了一些任務或執行緒。根據之前的程式碼，可能包括處理停車位的狀態、打印停車位資訊等。

vTaskStartScheduler()，啟動了 FreeRTOS 的任務排程器。這是一個多任務處理器，用於管理並行執行的任務。

```
155 int main(void)
156 {
157     /* Unlock protected registers */
158     SYS_UnlockReg();
159     /* Init system, IP clock and multi-function I/O. */
160     SYS_Init();
161     /* Lock protected registers */
162     SYS_LockReg();
163     GPIO_Init(); // LED initial
164     UART0_Init();
165
166     Debounce();
167
168     printf("-----LAB4-Basic-----\r\n");
169
170     // Create a semaphore for parking spaces with an initial count of 5
171     parkingSemaphore = xSemaphoreCreateCounting(5, 2); // 5 spaces, 2 spaces leave
172
173     vStartThreadTasks();
174     vTaskStartScheduler();
175
176     while(1);
177
178 }
```

▲圖 3

下圖 4 為創建的兩個 TASK，一個負責計數目前空位剩多少，一個負責處理進出的車輛。

```
181 void vStartThreadTasks( void )
182 {
183     xTaskCreate(vTaskPrintParkingSpaces, "vTaskMsgProl",128,NULL,1,( xTaskHandle * ) NULL );
184     xTaskCreate(CountParkingSpaces, "CountParkingSpaces",128,NULL,1,( xTaskHandle * ) NULL );
185 }
186
```

▲圖 4

下圖 5 有兩個函式 getInButtonPressed() 和 getOutButtonPressed()，模擬了兩種按鈕的行為，分別是 "get in" 按鈕和 "get out" 。

首先介紹 getInButtonPressed()，

printf("get in\n")，在按下 "get in" 按鈕時，輸出 "get in"。

if(xSemaphoreTake(parkingSemaphore, 0) == pdTRUE)，這段程式碼嘗試取得停車位的信號量。

xSemaphoreTake() 函式試圖取得 parkingSemaphore 信號量，第二個參數是超時時間，在這裡設置為 0 代表不等待。

如果成功取得信號量 pdTRUE，代表有空的停車位，printf("Car in!\n")，輸出 "Car in!"，表示車子進入停車位。parking_spaces--;; 停車位數量減 1。

如果無法取得信號量 pdFALSE，代表停車位已滿，輸出 "No Parking Space!"，表示沒有空的停車位。

PC12=0，用來控制 LED 燈，這裡是設置 PC12 為 0 表示停車場已滿。

再來看 getOutButtonPressed() 的部分，

printf("get out\n");: 在按下 "get out" 按鈕時，輸出 "get out"。

if(parking_spaces==5)，檢查停車場是否已經滿車（停車位數量是否等於 5），

如果已經沒有任何車子在裡面，printf("No Car inside!\n")，表示沒有車輛在停車場內。

如果還有空位，就使用 xSemaphoreGive(parkingSemaphore)，釋放停車位的信號量。parking_spaces++，停車位數量加一，表示有一個車輛離開了停車場。

```

188 // Function to simulate "get in" button press
189 void getInButtonPressed() {
190     printf("get in\n");
191     if(xSemaphoreTake(parkingSemaphore, 0) == pdTRUE) {
192         printf("Car in!\n");
193         parking_spaces--;
194     } else {
195         printf("No Parking Space!\n");
196         PC12=0;
197     }
198 }
199
200 // Function to simulate "get out" button press
201 void getOutButtonPressed() {
202     printf("get out\n");
203     if(parking_spaces==5){
204         printf("No Car inside!\n");
205     }else{
206         xSemaphoreGive(parkingSemaphore);
207         parking_spaces++;
208     }
209 }
210

```

▲圖 5

最後看到下圖 6，

CountParkingSpaces(void* pvParameters)這個任務的功能是定期顯示停車位的數量。

vTaskPrintParkingSpaces(void* pvParameters) 任務負責處理按鈕的按下事件以及控制 LED 的狀態。

```

211 static void CountParkingSpaces(void* pvParameters)
212 {
213     while(1)
214     {
215         //Task1 count per second (use vTaskDelay)
216         printf("%d parking spaces~\n",parking_spaces);
217         vTaskDelay(5000);
218     }
219 }
220
221 static void vTaskPrintParkingSpaces(void* pvParameters)
222 {
223     while(1)
224     {
225         if(GPIO_GET_INT_FLAG(PA, BIT2)) //key1
226         {
227             //printf("i am key1~~~\n");
228             GPIO_CLR_INT_FLAG(PA, BIT2);
229             getInButtonPressed();
230         }
231         if(GPIO_GET_INT_FLAG(PA, BIT1)) //key2
232         {
233             GPIO_CLR_INT_FLAG(PA, BIT1);
234             getOutButtonPressed();
235         }
236         if(parking_spaces==0){
237             PC12=0;
238         }else{
239             PC12=1;
240         }
241     }
242 }
243

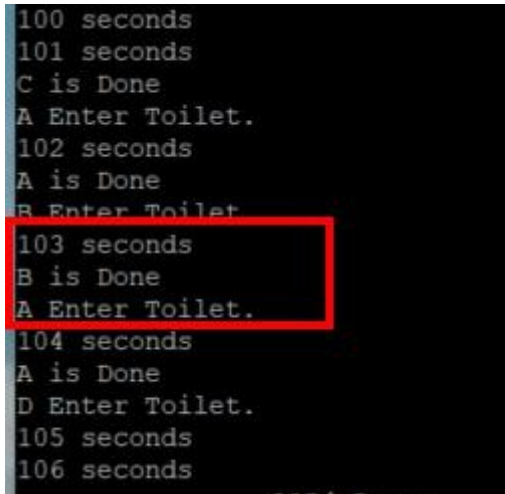
```

▲圖 6

Bonus:

In the 103rd second, who is in the toilet?

從下圖 7 紅框處可知，第 103s 的時候為 A 在廁所中。



```
100 seconds
101 seconds
C is Done
A Enter Toilet.
102 seconds
A is Done
B Enter Toilet.
103 seconds
B is Done
A Enter Toilet.
104 seconds
A is Done
D Enter Toilet.
105 seconds
106 seconds
```

▲圖 7

<心得與收穫>

這次 LAB 展示了嵌入式系統中使用 FreeRTOS 多任務處理，以及利用 Semaphore、Mutex 等觀念模擬了一個停車場管理系統，在 OS 的課程中有學過相關概念，因此在這次實作的過程也比較順利，這樣的經驗對於設計具有複雜操作、多功能性以及資源管理等嵌入式應用是非常有用的。