微算機系統實習

Final Project

個人報告

組別：19

109590014 沈煒翔

109590015 楊挺煜

109590023 廖堃霖

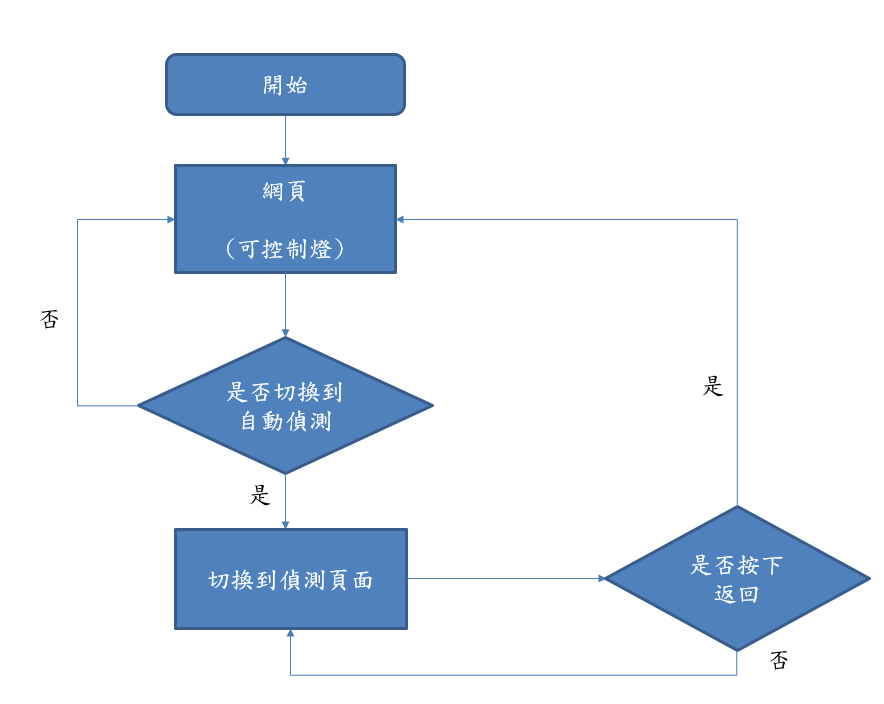
日期：111/06/24

**1. 使用情景**

(1) 遠端控制電燈

(2) 自動感光進行電燈控制

(3) 遠端警示燈(可控制閃爍次數、頻率)

**2. 專案流程圖**  


**3. 專案構思**

(1) LED燈控制：透過字元驅動控制LED燈。

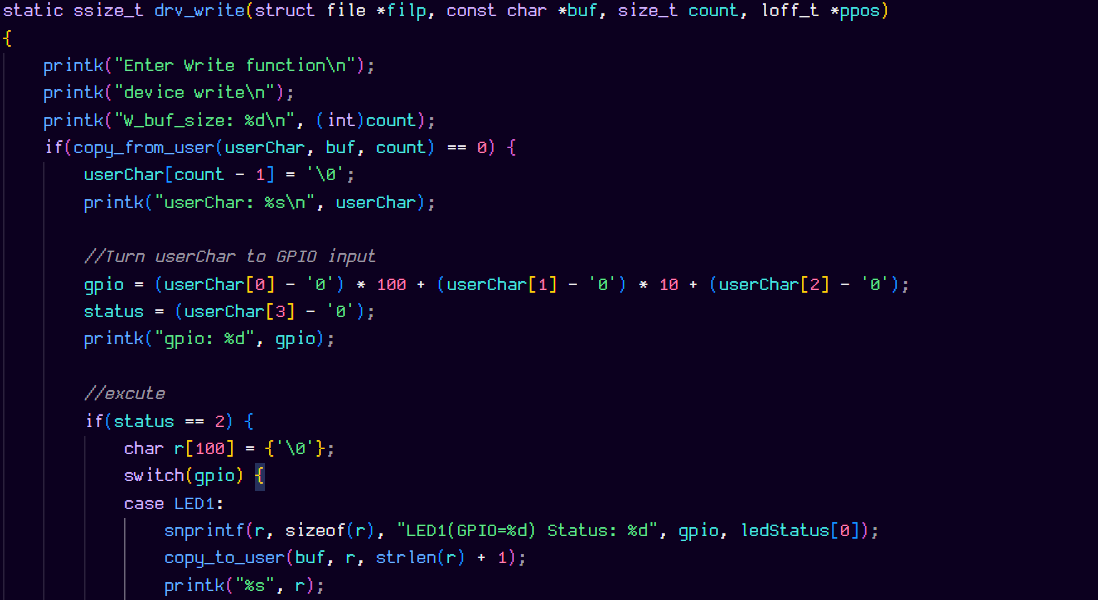
(2) 頻率閃爍LED燈：透過Mutex與Semaphore控制每次的LED開關。

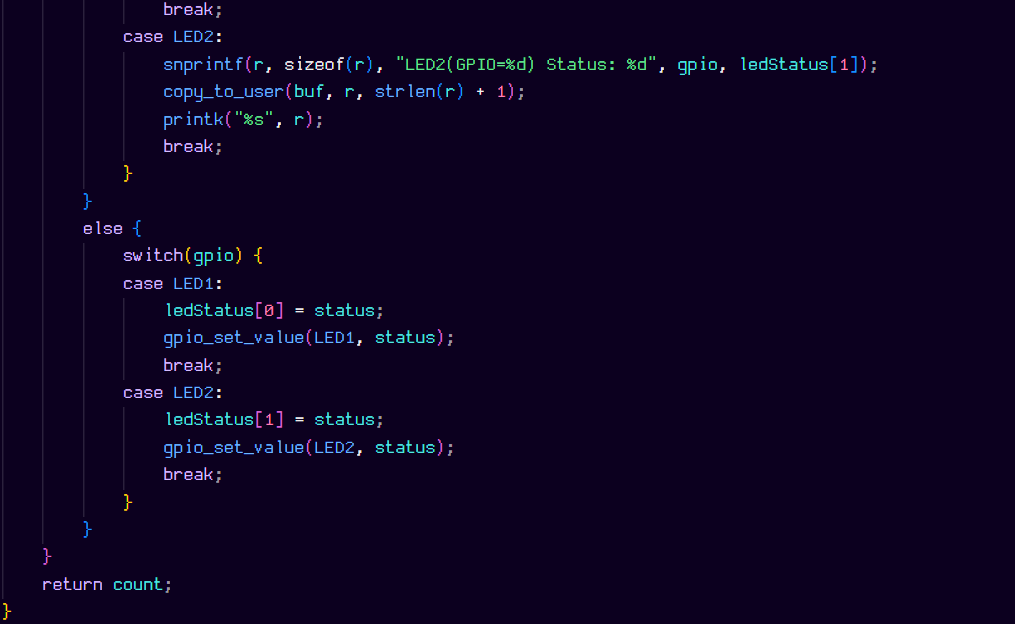
(3) 光敏電阻值偵測：使用Python的GPIO Module讀取光敏電阻的類比訊號。

(4) 網頁：使用NodeJS與NPM架設。

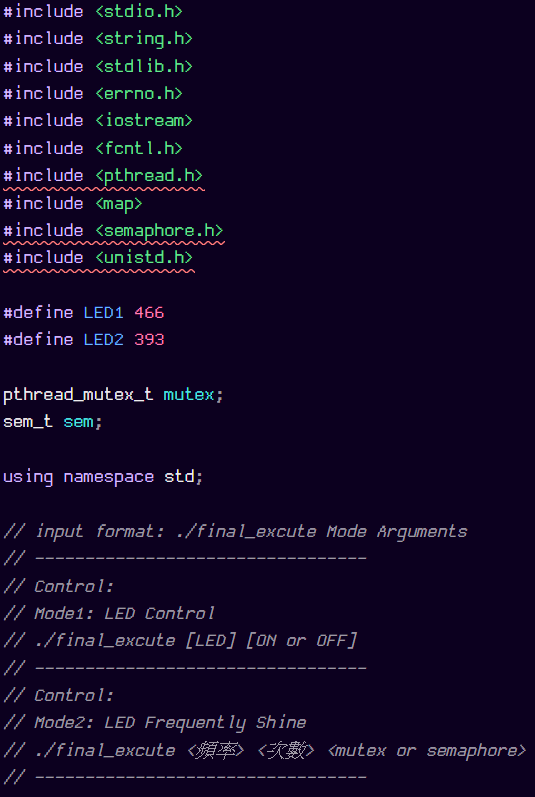
(5) 寄送Email：使用NPM的NodeMailer模組與同學的伺服器發送。

**4. 實驗步驟**

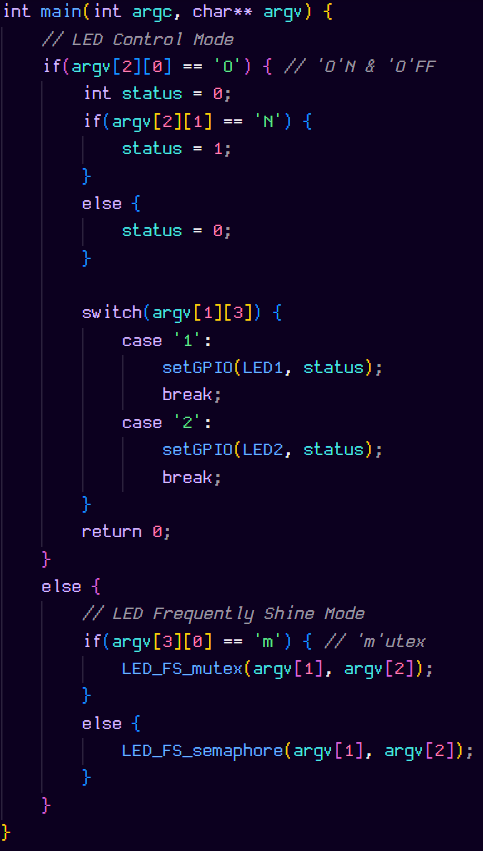


****

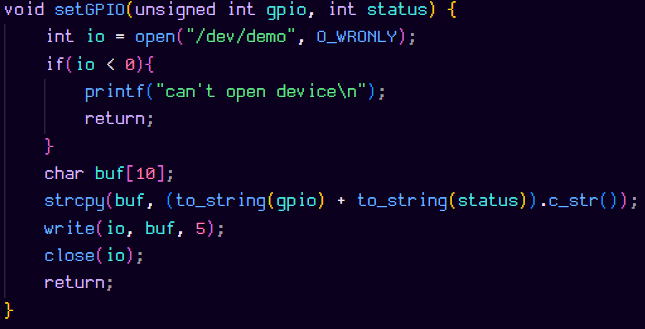
1. 字元驅動write function，基本上與Lab6相同。



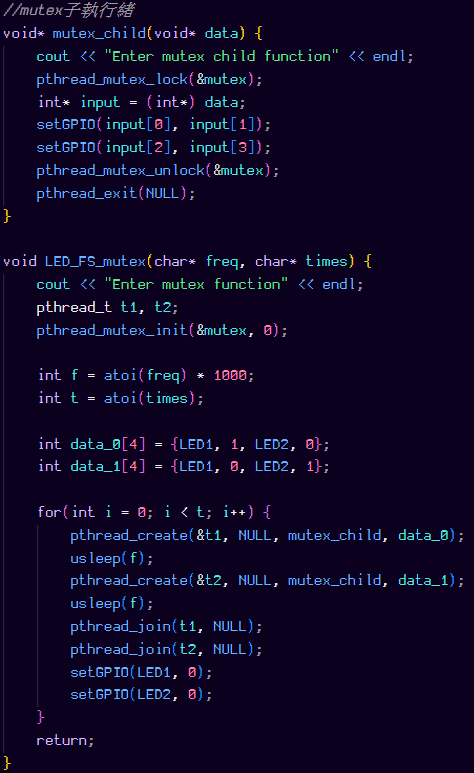
1. 執行檔cpp：Include header、Defines、全域變數、輸入格式註解。



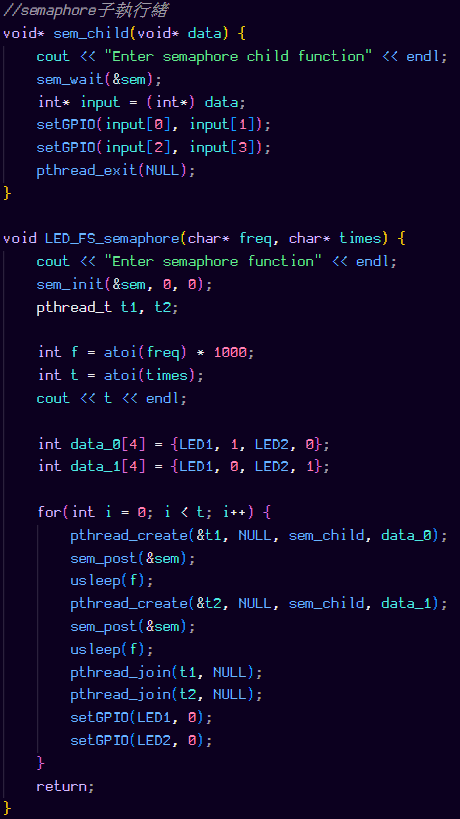
1. 執行檔cpp：main function



1. 執行檔cpp：控制GPIO的function



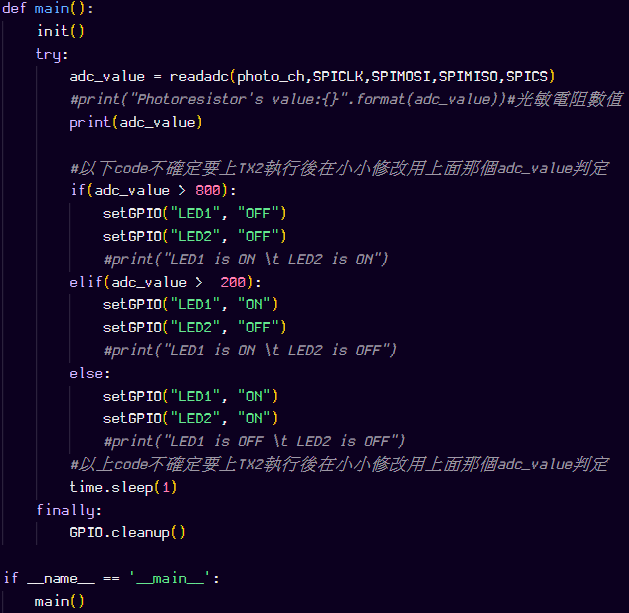
1. Mutex模式的function



1. Semaphore模式的function



1. final.py 設定LED function



1. final.py 透過adc\_value控制LED(其餘部份與Lab5相同)



1. control.js 切換模式寄送Email程式(其餘部份與Lab4大致相同)

**5. 專案心得**

這次的專案我們大概是照著老師給的基礎指示去完成，因為有其他的專案還沒有做完，所以時間很趕。不過這次的專案還是有遇到一些小問題。

1. 閃爍的頻率：以往要讓程式睡眠一段時間是用Sleep函式可是這個函式輸入integer單位為秒，也就是說最少只能停一秒。最後找到一個usleep()，他可以在Arm上的Linux執行而且他的單位是微秒。可以很精細的控制頻率，
2. 光敏控制LED：最一開始我是打算直接使用Lab5的程式，但是總是在Initial GPIO腳位的時候報出了Permission denied的錯誤。試了很多次最後發現應該是驅動已經有先Export取得權限了，接著就想說透過Python去修改”/dev/demo”的內容，可是不管怎麼試寫入”4661”(466為gpio，1為輸出高電位) userChar永遠只讀的到”466”而已，我找不出問題，不過我在想應該跟Python的字串處理方式有關。最後使用subprocess.check\_call()去執行cpp編譯出來的執行檔。

我負責的部份比較主要的問題是這兩個，感謝另外兩位組員分擔了不少工作。

**6. 小組分工**

109590014 沈煒翔 33% 網頁前後端，伺服器，製作報告，DeBug，上機測試

109590015 楊挺煜 33% 執行檔程式製作，驅動製作，製作報告，上機測試

109590023 廖堃霖 33% DeBug，上機測試，程式製作，製作報告