

微算機系統實習

MICROPROCESSOR SYSTEMS LAB.

SPRING, 2023

Instructor : Yen-Lin Chen(陳彥霖), Ph.D.

Professor

Dept. Computer Science and Information Engineering
National Taipei University of Technology



作業評分方式

- 實驗部分佔該次實驗的總比例70%
報告總分佔該次實驗的總比例30%
* 當次實驗會因為難易度不同佔學期總成績的實驗分數比例也不同
- 實驗報告上傳格式，公告於i學園
- 以小組為單位繳交報告

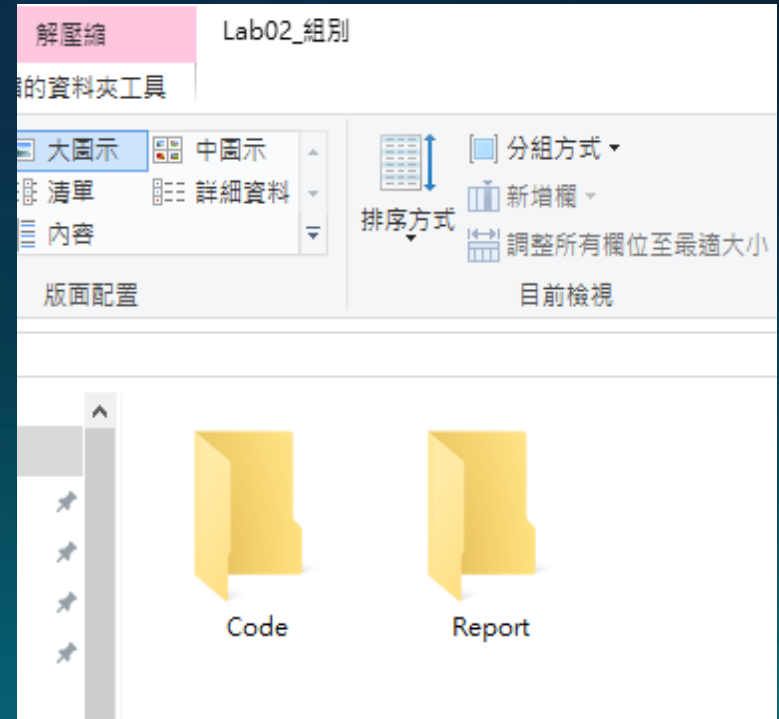
請同學上傳報告時依照上面的格式上傳，

- 一、組別與組員名單
- 二、實驗步驟截圖與說明
- 三、組員貢獻比例 (組員%數加總必須等於100%)
- 四、小組報告 & 個人心得
 - 實驗報告單獨分數為100分，以上第一、三、四點。
 - 每缺少一項報告分數扣20分。第二點缺少報告扣40分。另外第二、四點不完整會依照狀況扣分，最高扣到該項目的上限分數。
 - 貢獻比例分配方式為: 報告總分*人數*組員貢獻比例=組員報告得分

作業繳交格式

- 檔名: 不限，可辨認即可
- 例：Lab05_組別.zip
- 其zip裡要包含如下資料夾
 1. -Code //存放專案程式碼
 2. -Report //存放報告
 3. -影片請嵌入至小組報告內

嚴禁抄襲！！若有n組抄襲則抄襲
組別分數皆為 $1/n$





作業繳交

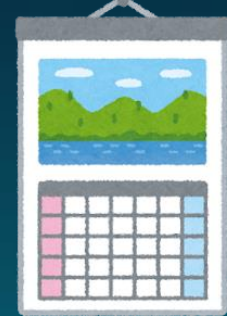
- 以小組為單位繳交報告：
- 實驗：公布實驗後一週 **5/5 (16:00)** 上課結束前驗收
- 報告：公布實驗後隔週的下個星期一 **5/8 (23:59)** 以前上傳
-報告內容：實作過程及心得（**每人150字以上**）

* 若有因為特殊原因繳交時間有變動助教會另外公布
超過時間遲交每隔一週（含一週內）分數打8折，

採累計連乘方式，

實驗與報告打折是分開算的，

最多遲交兩週， <遲交的項目單獨分數> * 0.64



- 舉例：

遲交三天－以遲交一週計算 $\text{<遲交的項目單獨分數>} * 0.8 = \text{該項目得到的分數}$

遲交九天－以遲交兩週計算 $\text{<遲交的項目單獨分數>} * 0.8 * 0.8 = \text{該項目得到的分數}$

- 以上配分與注意事項有問題請聯絡助教



小組報告格式規定

- 實驗內容
 - 此次實驗需要做什麼？
- 實驗流程及結果
 - 預期實驗的結果
 - 實際上的結果
 - 遇到的問題
- 程式碼
- 本次實驗流程細項說明與解決方法
 - 實驗過程
 - 解決方法
- 分工
 - 說明小組成員分工內容與比例。(務必加入)



個人報告內容

- 實驗心得：(150字)
 - 此次實驗的目的？
 - 在實驗上遇到的問題？怎麼解決？
 - 實驗結果跟你預期的內容是否一致？
- 組員貢獻度、工作內容以及你個人覺得合不合理？
 - 陳大明：33%，資料查詢、文書處理、實驗實作。
 - 陳忠義：33%，實驗設計、程式規劃、測試與除錯。
 - 陳小美：33%，小組報告、買飲料、買午餐。

實驗五



遠端週邊控制實習

實驗說明



本次實驗目標

- 學習透過C++和Python控制LED燈和光敏電阻
- 學習如何使用回呼函式（callback function）回傳給前端
光敏電阻的數值
- 學習如何使用jQuery 的\$.ajax() 來定期抓取後端數值
- 學習如何使用html和javascript語法
- 結合感測器與LED進行實驗



LAB 5-2 項目要求

透過網頁控制GPIO上的光敏電阻和及4顆LED燈開關狀態

- **項目一：網頁顯示光敏電阻的值 (45%)**
 - 在麵包板上安裝類比數位訊號轉換器及光敏電阻
 - 撰寫html及後端程式控制光敏電阻
 - 勾選DETECT，點擊Submit切換到偵測畫面，網頁會顯示光敏電阻的值
 - 依照設定光敏電阻的門檻值控制TX2上2顆LED燈開關狀態
- **項目二：切換模式來控制指定LED開關事件 (15%)**
 - 勾選NO_DETECT，點擊Submit會停止顯示光敏電阻的值，並切換畫面
 - **2顆LED燈狀態選項** (可選用Check Box)
 - 燈亮、熄滅選項及Submit按鈕
 - 勾選指定LED燈號，點擊Submit即可控制TX2上的LED燈
 - 點擊Submit後，需在原畫面顯示送出的狀態(在同個畫面)



LAB 5-2 項目要求

- 項目三：網頁控制多顆LED同時閃爍(10%)
 - 須有input欄位可以輸入指定閃爍次數
 - 預設按鈕名稱為Mode_Shine
 - 點擊後，依據指定閃爍次數，以間隔閃爍2組LED
- * 以上皆必須能在瀏覽器上操作

Async vs Sync

- 同步指的是同一時間照順序執行一行指令
- 非同步指同時有不同指令在執行，且執行完馬上換下一個指令，例如：Click, AJAX, SetInterval等
- 解決方法即透過callback, promise

```
function print_word(word){  
  setTimeout(function(){  
    console.log(word)  
  }, Math.floor(Math.random()*100))  
}  
  
function print_all_word(){  
  print_word("a")  
  print_word("b")  
  print_word("c")  
}  
  
print_all_word()
```

Async方法

c
b
a

執行相同程式碼卻得到不同結果

b
a
c

Callback function

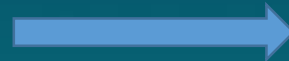
- 利用增加callback function確保每次執行順序唯一
- 第一個function執行完成才繼續呼叫下一個function

```
function print_word(word, callback){  
    setTimeout(function(){  
        console.log(word)  
        callback()  
    }, Math.floor(Math.random()*100)+1)  
}
```

加入Callback

```
function print_all_word(){  
    print_word("a", function(){  
        print_word("b", function(){  
            print_word("c", function(){})  
        })  
    })  
}  
print_all_word()
```

1
2
3



得到相同結果

```
⌵ a  
⌵ b  
⌵ c
```

```
⌵ a  
⌵ b  
⌵ c
```



Callback hell

- callback function可以確保每次執行順序正確
- 但當需要執行的function數量龐大時，會造成程式碼難以維護

```
function print_all_word(){
    print_word("a",function(){
        print_word("b",function(){
            print_word("c",function(){
                print_word("d",function(){
                    print_word("e",function(){
                        print_word("f",function(){
                            print_word("g",function(){
                                print_word("h",function(){
                                    // ...
                                })
                            })
                        })
                    })
                })
            })
        })
    })
}
```



Promise

- Promise也可以確保每次執行順序正確
- 寫法簡潔所以可讀性高，維護起來相對容易

```
function print_word(word){  
    return new Promise(function(resolve,reject){  
        setTimeout(function(){  
            console.log(word)  
            resolve()  
        },Math.floor(Math.random()*100))  
    })  
}
```

加入Promise

```
function print_all_word1(){  
    print_word("a")  
    .then(function(){  
        return print_word("b")  
    })  
    .then(function(){  
        return print_word("c")  
    })  
}
```

透過.then繼續執行下一道指令



Promise

- 新版Promise可以使用更簡潔的寫法，透過=>取代function和return使版面更乾淨

```
function print_word(word){  
    return new Promise(function(resolve,reject){  
        setTimeout(function(){  
            console.log(word)  
            resolve()  
        },Math.floor(Math.random()*100))  
    })  
}
```

```
function print_all_word2(){  
    print_word("a")  
    .then(()=> print_word("b"))  
    .then(()=> print_word("c"))  
}
```

移除function和return



AJAX使用範例

```
<script>
    $("#response").css("fontSize", 40); //加上#表示選擇response這個特定id
    setInterval(GetResult, 1000); //每一秒鐘呼叫一次
    function GetResult() {
        $.ajax({
            type: "get", //透過get的方式接收資料
            url: "/index",
            success: function (result) { //接收server回傳的資料
                $('#response').text(result); //變更id是response的標籤裡的文字
            },
            error: function() { //失敗給出提示
                alert("error");
            }
        })
    }
</script>
```

Index.html



後端傳值範例

```
function test(){  
  return new Promise(function(resolve, reject) {  
    //promise需自帶resolve和reject兩個函數  
  
    let child_process = require("child_process");  
  
    let process = child_process.spawn('python', [  
      | './python/test.py"  
    ]);  
    //成功時從stdout輸出,data抓取spawn執行後得到的結果  
    process.stdout.on('data', (data) => {  
      | console.log(`stdout: ${data}`);  
      | resolve(data)//透過resolve回傳執行結果  
    });  
    //失敗時從stderr輸出  
    process.stderr.on('data', (data) => {  
      | console.error(`stderr: ${data}`);  
      | reject(data)//透過reject回傳錯誤結果  
    });  
  })  
}
```

Index.js

實際執行範例

← → ↻ ⚠ 不安全 | 192.168.137.182:8080

Nodejs

請選擇模式

☐ DETECT ☐ NO_DETECT



Nodejs

感測器偵測中

adc_value= 139

☒ DETECT ☐ NO_DETECT

勾選DETECT，點擊Submit後會切換到偵測畫面，並顯示光敏電阻的值



實際執行範例

Nodejs

感測器偵測中

adc value= 138

☐ DETECT ☒ NO_DETECT



Nodejs

手動控制面板

LED Control Panel

☐ LED1 ☐ LED2

☐ ON ☐ OFF

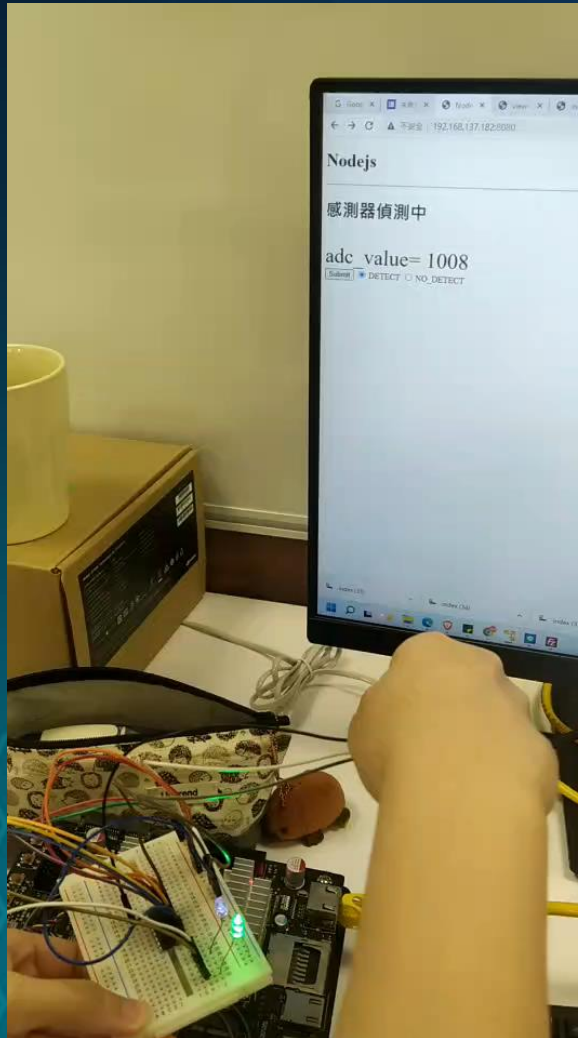
LED Switch Frequency

☐ DETECT ☒ NO_DETECT

勾選NO_DETECT，點擊Submit會停止顯示光敏電阻的值，並切換下個畫面



實際執行範例



加分題

定域監控



加分題：定域監控

故事：

這是一個小偷只在亮度50以上才會出現並且行竊的世界，不要問為什麼，問就是小偷怕黑，並且小偷也不在亮度500以上的時候行竊，因為小偷有吸血鬼的體質，太亮灰飛煙滅，所以我們只在50~500以上監控。

情境：

在正常光源下觀測影像，因為太暗一團黑沒有監控的必要，太亮傷害感光元件。

要求：

- 光源數值50~500才開啟CSI相機或USB相機。
- 可手動開關畫面，並且可以拍照、錄影。

你能夠協助這個世界的警備人員嗎？

