微算機系統實習
MICROPROCESSOR SYSTEMS LAB.
SPRING, 2021

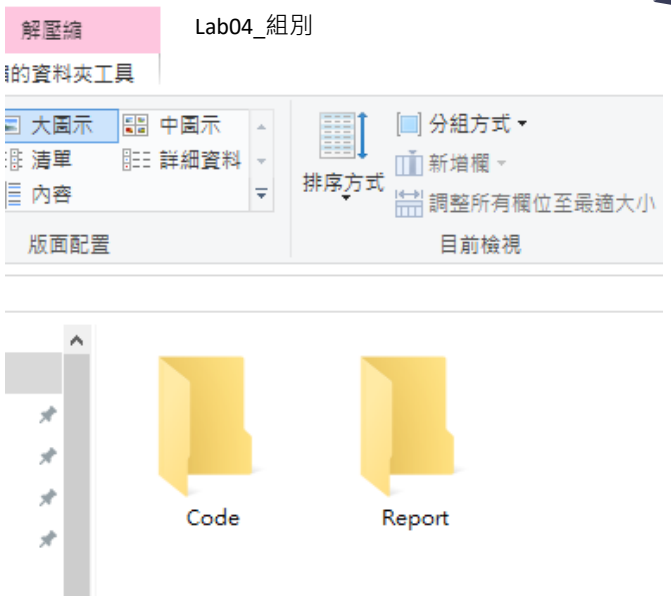
Instructor：Yen-Lin Chen(陳彥霖), Ph.D.
Professor
Dept. Computer Science and Information Engineering
National Taipei University of Technology

實驗四

透過CGI網頁控制GPIO

作業繳交格式

- 檔名: Lab04_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



作業繳交

- 基本繳交時間

  -實驗：4/23 (18:00)上課結束前驗收

  -報告：4/29 (23:59)以前上傳

  * 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

  - 舉例：

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

    遲交九天 - 以遲交兩週計算 <遲交的項目單獨分數>*0.8*0.8 = 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

實驗說明

本次實驗目標

- 利用Nodejs架設web伺服器
- 安裝npm開源套件
- 撰寫html基礎語法
- 了解前後端參數傳遞概念
- 利用child_process執行子程序
- 學習透過CGI網頁，控制GPIO上的LED燈

LAB4 項目要求

透過網頁控制GPIO上的4個LED

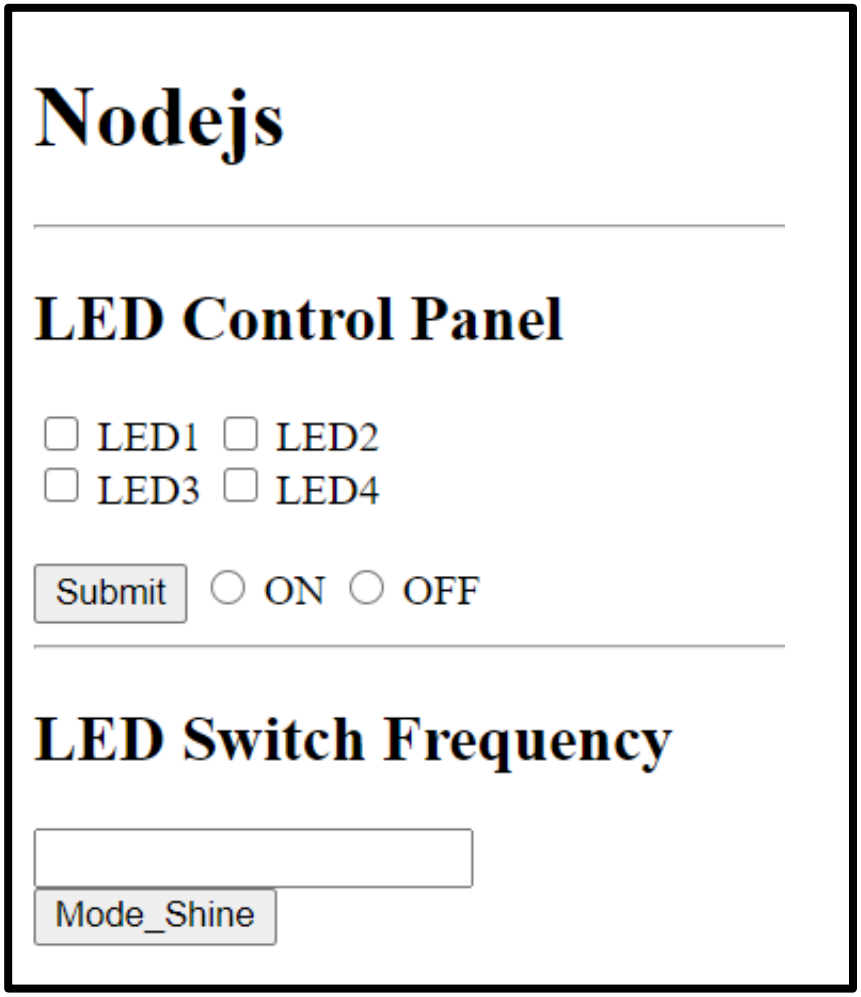
- **項目一：利用Nodejs架設web伺服器(20%)**
  - 撰寫html及後端程式
  - 在TX2上架設web伺服器
  - 能在**Windows**電腦瀏覽器上查看網頁
- **項目二：網頁控制指定LED開關事件(25%)**
  - 4顆LED燈狀態選項 (可選用Check Box)
  - 燈亮、熄滅選項及Submit按鈕
  - 勾選指定LED燈號，點擊Submit即可控制TX2上的LED

LAB4 項目要求

- **項目三：網頁控制多顆LED同時閃爍(25%)**
  - 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windows瀏覽器上操作

實驗執行範例

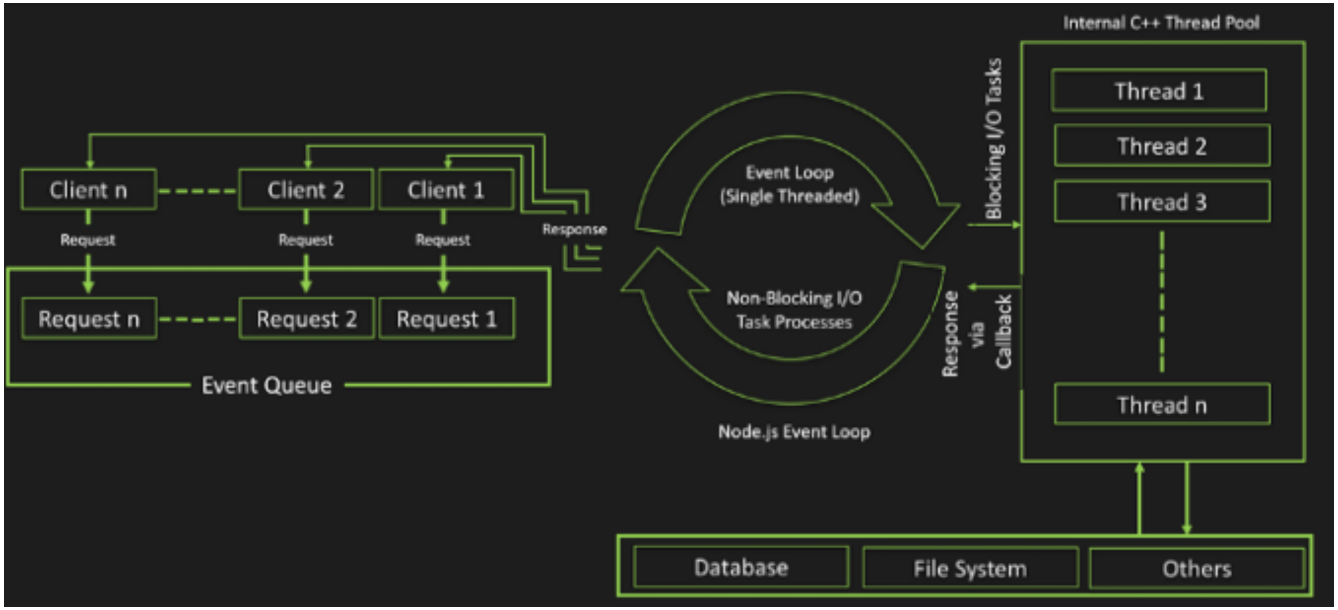


執行範例UI
（僅供參考，同學可自行加入CSS及JavaScript，排版可與此不同）

如何利用Nodejs執行子程序

child_process介紹

- child_process 為Node.js內建的模組，可開啟執行多個子程序，在多個子程序之間可以共享記憶體空間並互相通訊來實現資訊的交換。
- 透過系統命令設定父子程序來優化 CPU 計算的問題、或進行多程序開發等，極大化 Node.js 的開發能力。



child_process API

- 以下為幾個child_process常用的API
  - child_process.spawn ()
  - child_process.spawnSync()
  - child_process.exec()
  - child_process. execFile()

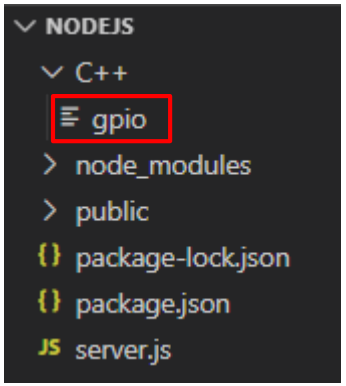
- API說明及操作方法，可參考官方網站
  - https://nodejs.org/api/child_process.html?fbclid=IwAR1UunjxPktROWgpOYDtxd53DQtho0r4ll2eiFEyTErfp5UZIDX9YN6mH2Y#child_process_child_process_exec_command_options_callback

利用child_process執行子程序

server.js

```
1  const express = require("express");
2  const app = express();
3
4  // simple route
5
6  app.use(express.static('./public'));
7
8  app.get("/index", (req, res) => {
9    controlLED(req.query.LED, req.query.POWER)
10   res.send("Successfully Requested")
11 });
12
13 function controlLED(LED, POWER){
14
15   let child_process = require("child_process");
16
17   let process = child_process.execFile('sudo', [
18     "./C++/gpio", LED, POWER
19   ]);
20   process.stdout.on('data', (data) => {
21       console.log(`stdout: ${data}`);
22   });
23   process.stderr.on('data', (data) => {
24       console.error(`stderr: ${data}`);
25   });
26 }
27
28 // set port, listen for requests
29 const PORT = process.env.PORT || 8080;
30 app.listen(PORT, () => {
31   console.log(`Server is running on port ${PORT}.`);
32 });
```

引入模組



專案架構

*若以sudo執行程式，則必須在TX2上輸入權限密碼

利用**execFile()**執行操控LED程式

此例即是執行sudo ./C++/gpio **LED** **POWER**

LED與**POWER**即是客戶端傳進的參數

監聽此程式的輸出內容並印出

監聽此程式的錯誤訊息並印出