微算機系統實習

第 13 組專案報告

LAB 04

GPIO 遠端週邊控制實習

組別： 13

班級、姓名與學號：

醫工三 B812110004 葉芸茜

醫工三 B812110011 湯青秀

日期：2024.04.16

一、實驗內容：

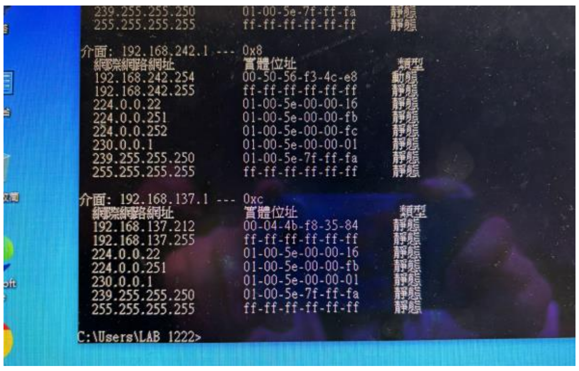
1. 利用 Nodejs 架設 web 伺服器
2. 安裝 npm 開源套件
3. 撰寫 html 基礎語法
4. 了解前後端參數傳遞概念
5. 利用 child_process 執行子程序
6. 學習透過 CGI 網頁，控制 GPIO 上的 LED 燈
   a. 項目一：利用 Nodejs 架設 web 伺服器(20%)
      i. 撰寫 html 及後端程式
      ii. 在 TX2 上架設 web 伺服器
      iii. 能在電腦瀏覽器上查看網頁
   b. 項目二：網頁控制指定 LED 開關事件(25%)
      i. 4 顆 LED 燈狀態選項 (可選用 Check Box)
      ii. 燈亮、熄滅選項及 Submit 按鈕
      iii. 勾選指定 LED 燈號，點擊 Submit 即可控制 TX2 上的 LED
      iv. 點擊 Submit 後，需在原畫面顯示送出的狀態(在同個畫面)
   c. 項目三：網頁控制多顆 LED 同時閃爍(25%)
      i. 須有 input 欄位可以輸入指定閃爍次數
      ii. 預設按鈕名稱為 Mode_Shine
      iii. 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)

二、實驗過程及結果：

1. 實驗過程

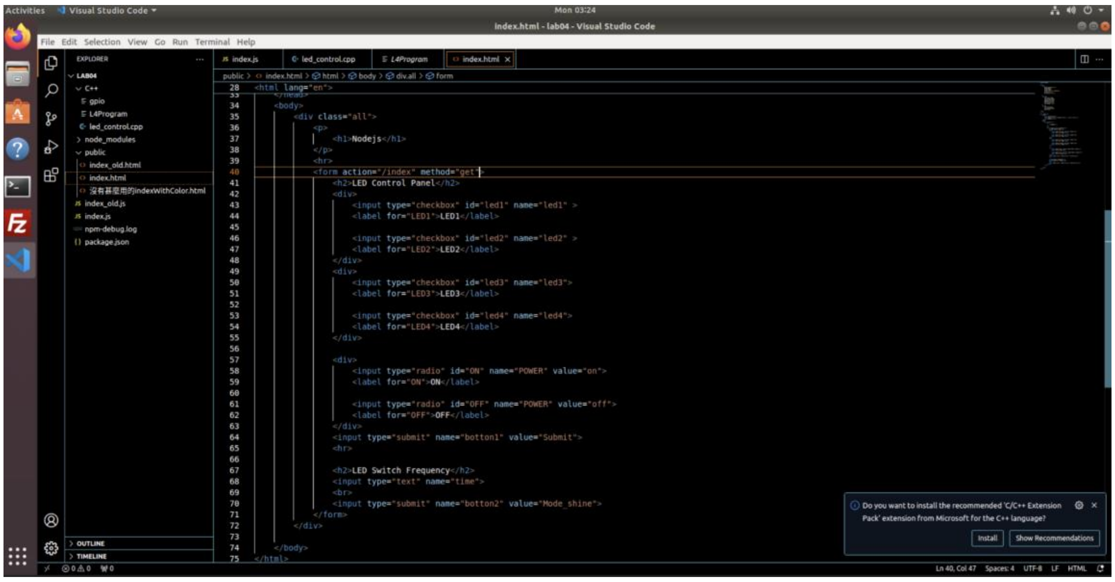
   因為本實驗在 tx2 需共用網路，所以需使用網路線來進行連接，可在 cmd 輸入指令查看對應 tx2 ip，此 ip 之後可在瀏覽器進行輸入來查看網頁



```
  239.255.255.250       01-00-5e-7f-ff-fa     靜態
  255.255.255.255       ff-ff-ff-ff-ff-ff     靜態

介面: 192.168.242.1 --- 0x8
  網際網路網址          實體位址              類型
  192.168.242.254       00-50-56-f3-4c-e8     動態
  192.168.242.255       ff-ff-ff-ff-ff-ff     靜態
  224.0.0.22            01-00-5e-00-00-16     靜態
  224.0.0.251           01-00-5e-00-00-fb     靜態
  224.0.0.252           01-00-5e-00-00-fc     靜態
  230.0.0.1             01-00-5e-00-00-01     靜態
  239.255.255.250       01-00-5e-7f-ff-fa     靜態
  255.255.255.255       ff-ff-ff-ff-ff-ff     靜態

介面: 192.168.137.1 --- 0xc
  網際網路網址          實體位址              類型
  192.168.137.212       00-04-4b-f8-35-84     靜態
  192.168.137.255       ff-ff-ff-ff-ff-ff     靜態
  224.0.0.22            01-00-5e-00-00-16     靜態
  224.0.0.251           01-00-5e-00-00-fb     靜態
  230.0.0.1             01-00-5e-00-00-01     靜態
  239.255.255.250       01-00-5e-7f-ff-fa     靜態
  255.255.255.255       ff-ff-ff-ff-ff-ff     靜態

C:\Users\LAB 1222>
```

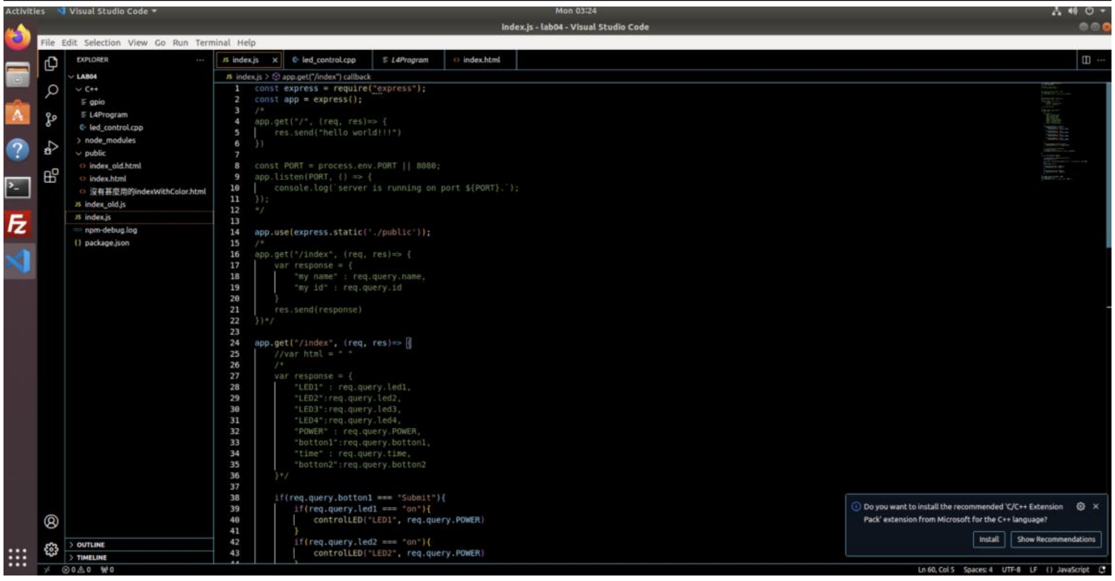
(1)項目一

(a)　撰寫 html 及後端程式


index.html - lab04 - Visual Studio Code

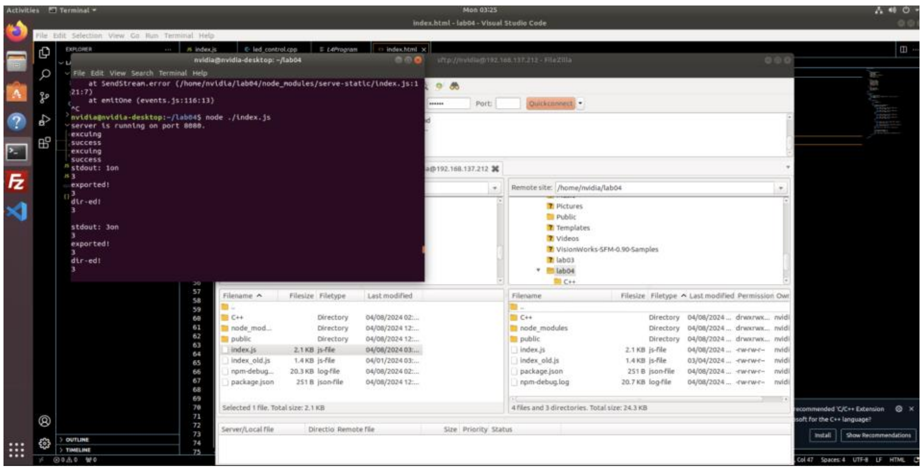
```html
<html lang="en">
    <body>
        <div class="all">
            <p>
                <h1>Nodejs</h1>
            </p>
            <hr>
            <form action="/index" method="get">
                <h2>LED Control Panel</h2>
                <div>
                    <input type="checkbox" id="led1" name="led1" >
                    <label for="LED1">LED1</label>

                    <input type="checkbox" id="led2" name="led2" >
                    <label for="LED2">LED2</label>
                </div>
                <div>
                    <input type="checkbox" id="led3" name="led3">
                    <label for="LED3">LED3</label>

                    <input type="checkbox" id="led4" name="led4">
                    <label for="LED4">LED4</label>
                </div>

                <div>
                    <input type="radio" id="ON" name="POWER" value="on">
                    <label for="ON">ON</label>

                    <input type="radio" id="OFF" name="POWER" value="off">
                    <label for="OFF">OFF</label>
                </div>
                <input type="submit" name="botton1" value="Submit">
                <hr>

                <h2>LED Switch Frequency</h2>
                <input type="text" name="time">
                <br>
                <input type="submit" name="botton2" value="Mode_shine">
            </form>
        </div>

    </body>
</html>
```

Do you want to install the recommended 'C/C++ Extension Pack' extension from Microsoft for the C++ language?


index.js - lab04 - Visual Studio Code

```javascript
const express = require("express");
const app = express();
/*
app.get("/", (req, res)=> {
    res.send("hello world!!!")
})

const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
    console.log(`server is running on port ${PORT}.`);
});
*/

app.use(express.static('./public'));
/*
app.get("/index", (req, res)=> {
    var response = {
        "my name" : req.query.name,
        "my id" : req.query.id
    }
    res.send(response)
})*/

app.get("/index", (req, res)=> {
    //var html = " "
    /*
    var response = {
        "LED1" : req.query.led1,
        "LED2":req.query.led2,
        "LED3":req.query.led3,
        "LED4":req.query.led4,
        "POWER" : req.query.POWER,
        "botton1":req.query.botton1,
        "time" : req.query.time,
        "botton2":req.query.botton2
    }*/

    if(req.query.botton1 === "Submit"){
        if(req.query.led1 === "on"){
            controlLED("LED1", req.query.POWER)
        }
        if(req.query.led2 === "on"){
            controlLED("LED2", req.query.POWER)
```

(b)　在 TX2 上架設 web 伺服器


nvidia@nvidia-desktop: ~/lab04

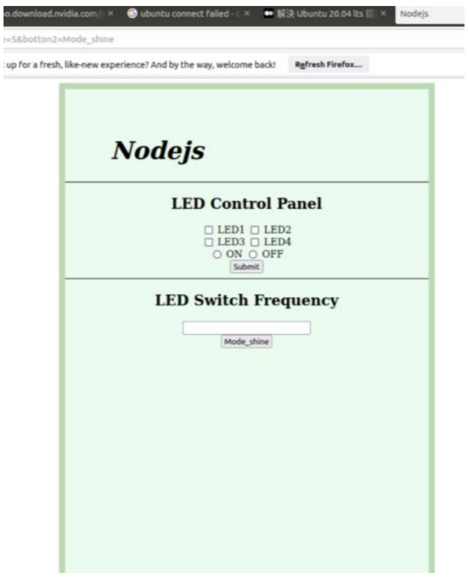
```
    at SendStream.error (/home/nvidia/lab04/node_modules/serve-static/index.js:1
21:7)
    at emitOne (events.js:116:13)
^C
nvidia@nvidia-desktop:~/lab04$ node ./index.js
server is running on port 8080.
excuing
success
excuing
success
stdout: 1on
3
exported!
3
dir-ed!
3

stdout: 3on
3
exported!
3
dir-ed!
3
```

Remote site: /home/nvidia/lab04

Filename	Filesize	Filetype	Last modified	Permission
..				
C++		Directory	04/08/2024 ...	drwxrwx...
node_modules		Directory	04/08/2024 ...	drwxrwx...
public		Directory	04/08/2024 ...	drwxrwx...
index.js	2.1 KB	js-file	04/08/2024 ...	-rw-rw-r--
index_old.js	1.4 KB	js-file	03/04/2024 ...	-rw-rw-r--
package.json	251 B	json-file	04/08/2024 ...	-rw-rw-r--
npm-debug.log	20.7 KB	log-file	04/08/2024 ...	-rw-rw-r--

4 files and 3 directories. Total size: 24.3 KB

（c）　能在電腦瀏覽器上查看網頁



Nodejs

LED Control Panel

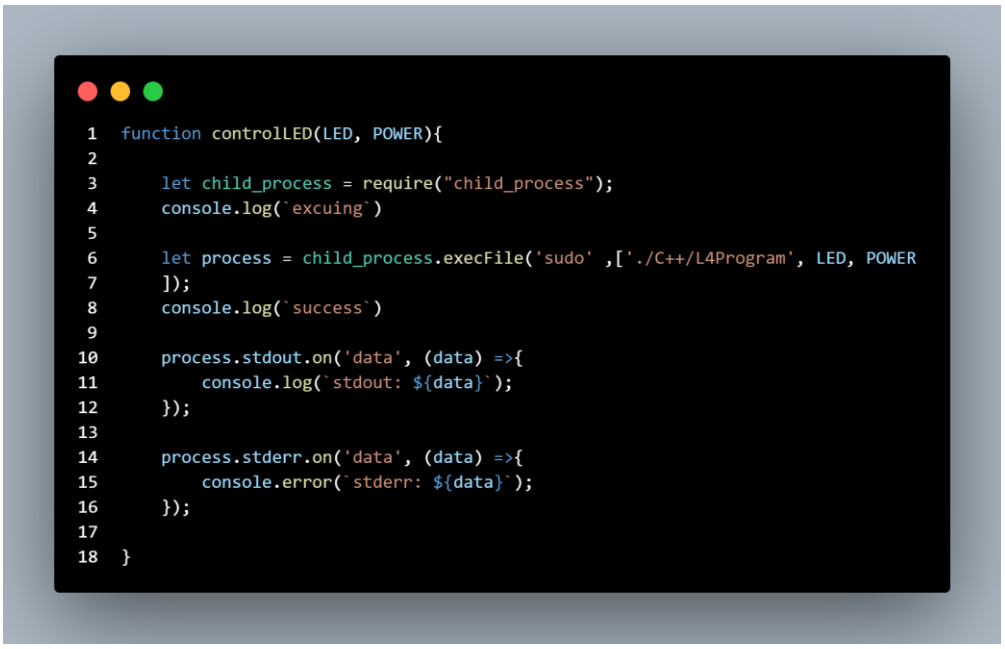
☐ LED1 ☐ LED2
☐ LED3 ☐ LED4
○ ON ○ OFF
Submit

LED Switch Frequency

Mode_shine

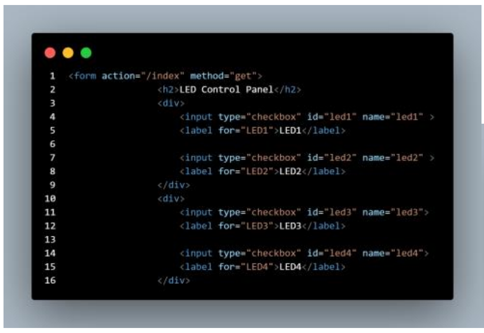
(2)項目二

進行 LED 操作的子程式如下。其中 L4Program 為 led_control.cpp 編譯出來的執行檔。

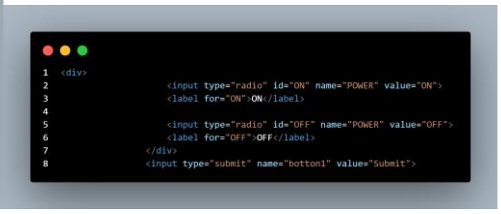


```
function controlLED(LED, POWER){

    let child_process = require("child_process");
    console.log(`excuing`)

    let process = child_process.execFile('sudo' ,['./C++/L4Program', LED, POWER
    ]);
    console.log(`success`)

    process.stdout.on('data', (data) =>{
        console.log(`stdout: ${data}`);
    });

    process.stderr.on('data', (data) =>{
        console.error(`stderr: ${data}`);
    });

}
```

i. html(checkbox & radio & button)

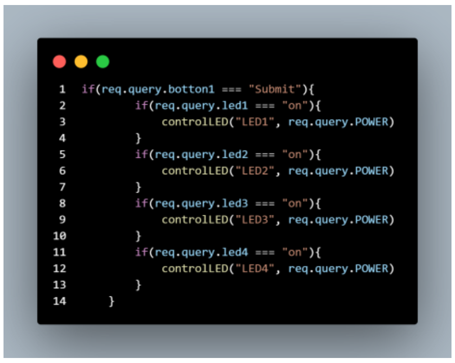


```
<form action="/index" method="get">
                <h2>LED Control Panel</h2>
                <div>
                    <input type="checkbox" id="led1" name="led1" >
                    <label for="LED1">LED1</label>

                    <input type="checkbox" id="led2" name="led2" >
                    <label for="LED2">LED2</label>
                </div>
                <div>
                    <input type="checkbox" id="led3" name="led3">
                    <label for="LED3">LED3</label>

                    <input type="checkbox" id="led4" name="led4">
                    <label for="LED4">LED4</label>
                </div>
```



```
<div>
                    <input type="radio" id="ON" name="POWER" value="ON">
                    <label for="ON">ON</label>

                    <input type="radio" id="OFF" name="POWER" value="OFF">
                    <label for="OFF">OFF</label>
                </div>
                <input type="submit" name="botton1" value="Submit">
```

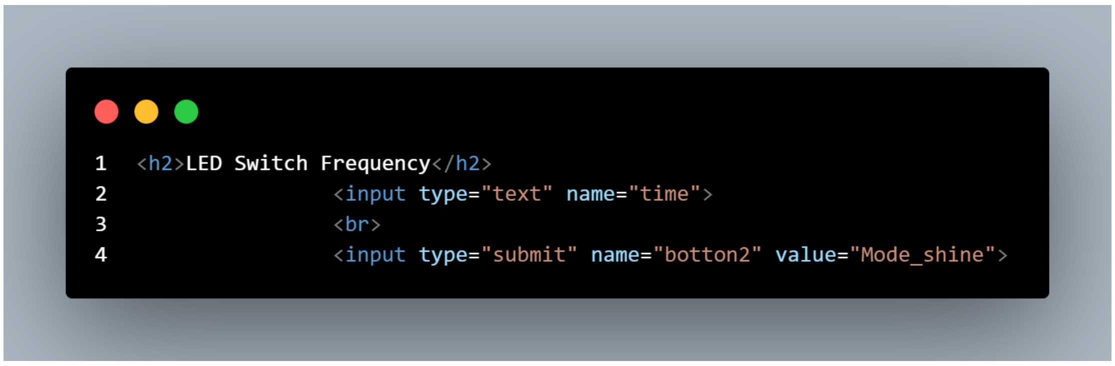
ii. js



```
if(req.query.botton1 === "Submit"){
        if(req.query.led1 === "on"){
            controlLED("LED1", req.query.POWER)
        }
        if(req.query.led2 === "on"){
            controlLED("LED2", req.query.POWER)
        }
        if(req.query.led3 === "on"){
            controlLED("LED3", req.query.POWER)
        }
        if(req.query.led4 === "on"){
            controlLED("LED4", req.query.POWER)
        }
    }
```

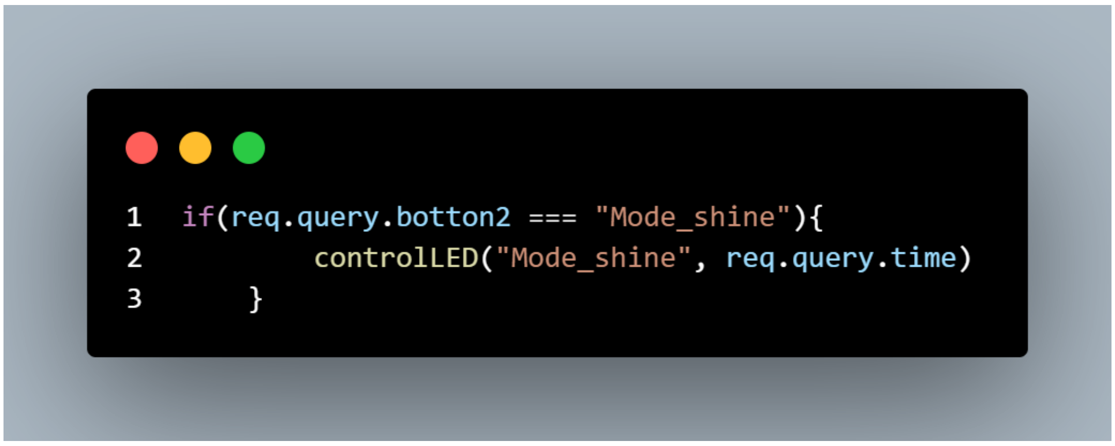
(3)項目三

i. html(text & button)



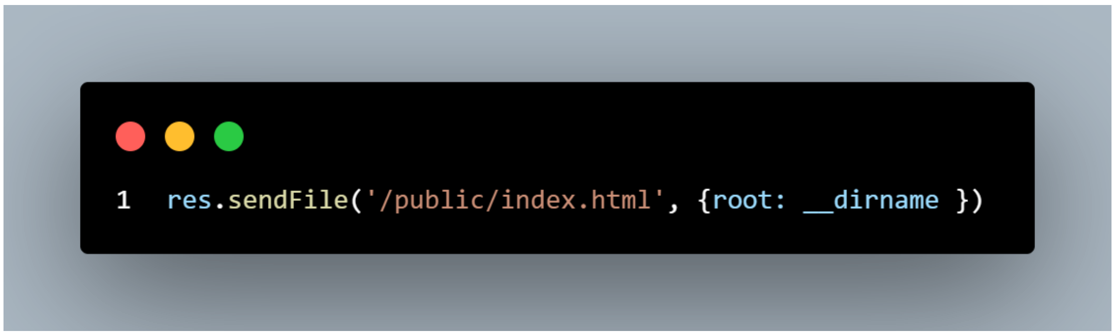
```
<h2>LED Switch Frequency</h2>
                <input type="text" name="time">
                <br>
                <input type="submit" name="botton2" value="Mode_shine">
```

ii. js



```
if(req.query.botton2 === "Mode_shine"){
          controlLED("Mode_shine", req.query.time)
      }
```

※ 最後回傳



```
res.sendFile('/public/index.html', {root: __dirname })
```

2. 預期實驗結果

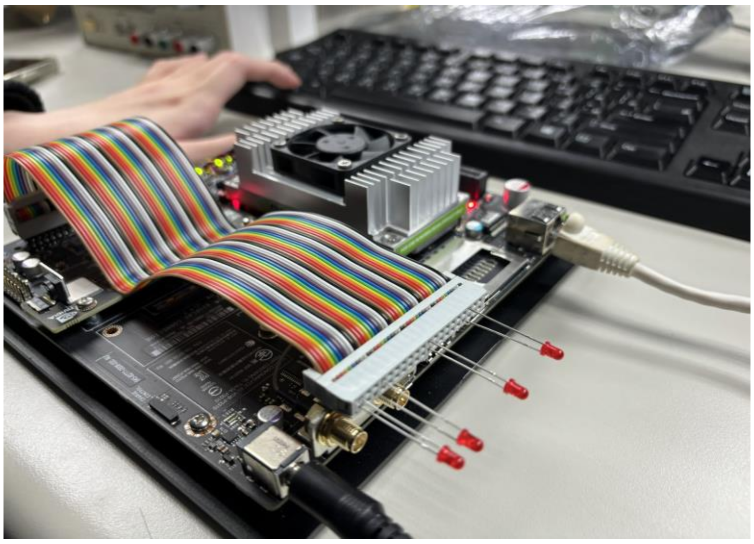
能在 tx2 的電腦瀏覽器上查看網頁，並利用所寫網頁控制指定 led 開關事件(利用 checkbox 選擇目標 led、radio 勾選 ON or OFF，並按鍵提交)，可勾選燈亮或熄滅再按提交。提交後 tx2 上相應的 led 會做出對應動作。

另外，可於 input 欄位輸入指定閃爍次數，並點擊按鍵來控制間隔閃爍 2 組 led 的次數。

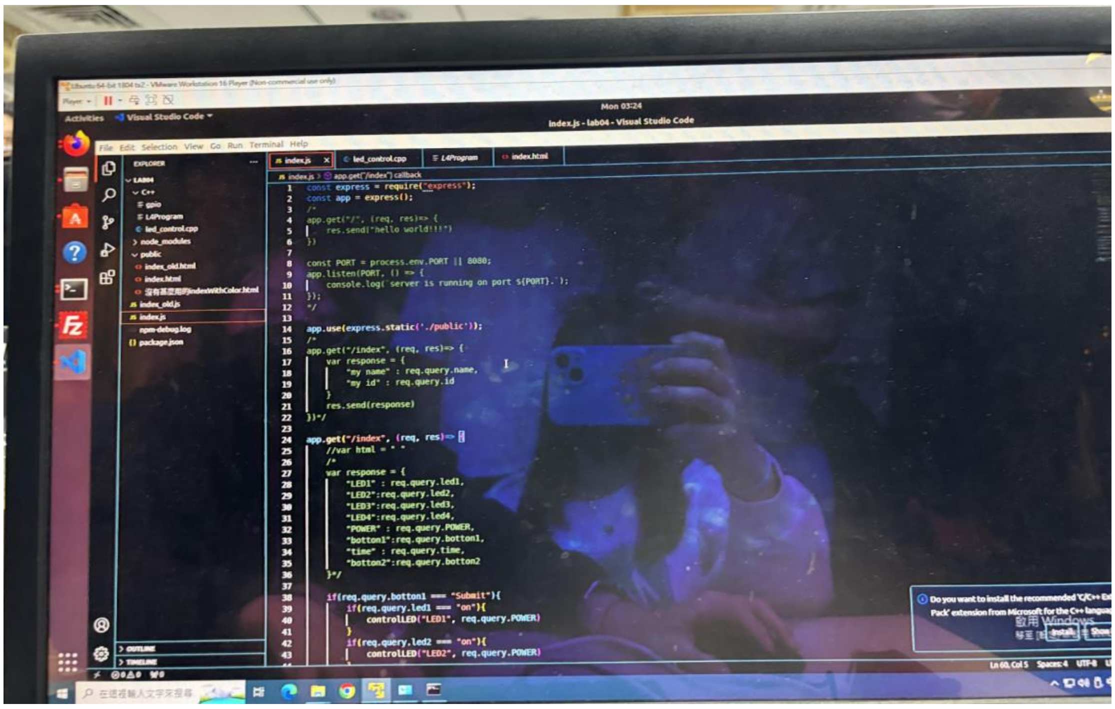
3. 實際上的結果

https://drive.google.com/file/d/1zYEFGyqG9nopRhrkEPGTjE1L5FhhrNxs/view?usp=drive_link（操作影片連結）

(1)led 接到 tx2

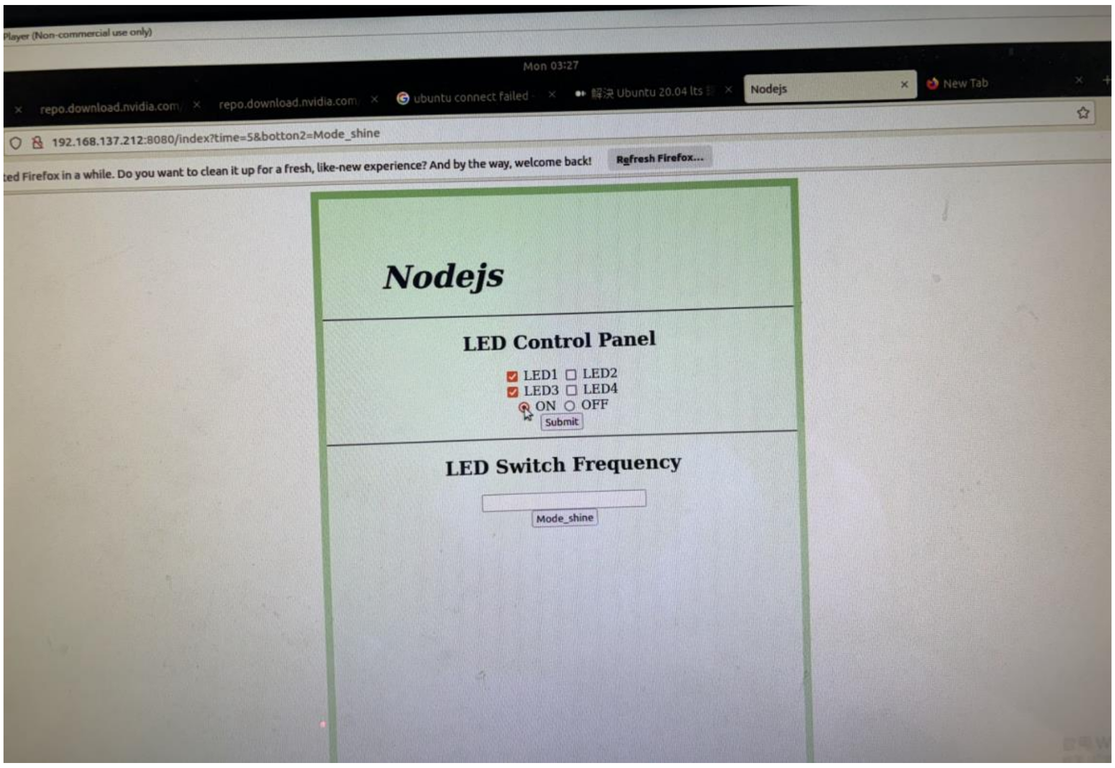


(2)程式碼畫面

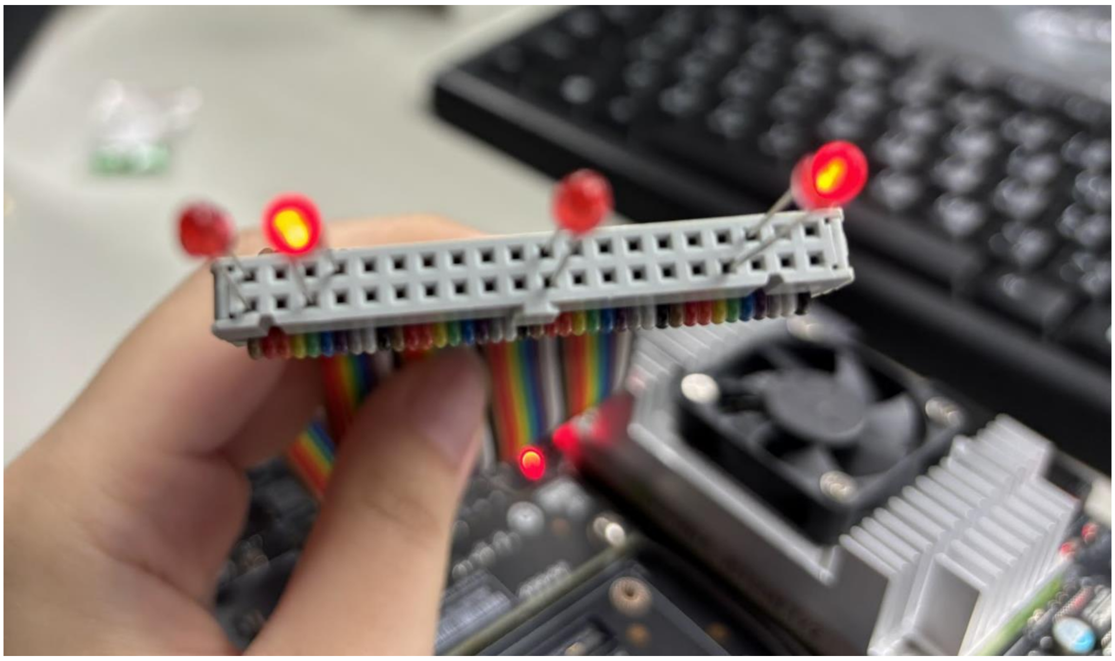


```
const express = require("express");
const app = express();
/*
app.get("/", (req, res)=> {
    res.send("hello world!!!")
})

const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
    console.log(`server is running on port ${PORT}.`);
});
*/

app.use(express.static('./public'));
/*
app.get("/index", (req, res)=> {
    var response = {
        "my name" : req.query.name,
        "my id" : req.query.id
    }
    res.send(response)
})*/

app.get("/index", (req, res)=> {
    //var html = " "
    /*
    var response = {
        "LED1" : req.query.led1,
        "LED2":req.query.led2,
        "LED3":req.query.led3,
        "LED4":req.query.led4,
        "POWER" : req.query.POWER,
        "botton1":req.query.botton1,
        "time" : req.query.time,
        "botton2":req.query.botton2
    }*/

    if(req.query.botton1 === "Submit"){
        if(req.query.led1 === "on"){
            controlLED("LED1", req.query.POWER)
        }
        if(req.query.led2 === "on"){
            controlLED("LED2", req.query.POWER)
```

(3)實際在瀏覽器操作



(4)控制 led 亮燈畫面



4. 遇到的問題&問題怎麼解決

(1) 不會寫前後端，只能自己想辦法上網查

(2) tx2 沒網路，網路共用設置相反導致抓不到 ip，後來跑回去翻講義才設置好

(3) 在最後為了不讓頁面跳開，花了很多時間嘗試，最後還是選擇直接回傳 html 檔案

三、程式碼

index. js

```
const express = require("express");
const app = express();
/*
app.get("/", (req, res)=> {
    res.send("hello world!!!")
})

const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
    console.log(`server is running on port ${PORT}.`);
});
*/

app.use(express.static('./public'));
/*
app.get("/index", (req, res)=> {
    var response = {
        "my name" : req.query.name,
        "my id" : req.query.id
    }
    res.send(response)
})*/

app.get("/index", (req, res)=> {
    //var html = " "
    /*
    var response = {
        "LED1" : req.query.led1,
        "LED2":req.query.led2,
        "LED3":req.query.led3,
        "LED4":req.query.led4,
        "POWER" : req.query.POWER,
        "botton1":req.query.botton1,
        "time" : req.query.time,
        "botton2":req.query.botton2
    }*/

    if(req.query.botton1 === "Submit"){
        if(req.query.led1 === "on"){
            controlLED("LED1", req.query.POWER)
        }
        if(req.query.led2 === "on"){
            controlLED("LED2", req.query.POWER)
        }
        if(req.query.led3 === "on"){
            controlLED("LED3", req.query.POWER)
        }
        if(req.query.led4 === "on"){
            controlLED("LED4", req.query.POWER)
        }
    }

    if(req.query.botton2 === "Mode_shine"){
        controlLED("Mode_shine", req.query.time)
    }

    //res.send("Successfully Requested")
    //res.send('response')
    res.sendFile('/public/index.html', {root: __dirname })

})

function controlLED(LED, POWER){

    let child_process = require("child_process");
    console.log(`excuing`)

    let process =
child_process.execFile('sudo' ,['./C++/L4Program', LED, POWER
    ]);
    console.log(`success`)

    process.stdout.on('data', (data) =>{
        console.log(`stdout: ${data}`);
    });

    process.stderr.on('data', (data) =>{
        console.error(`stderr: ${data}`);
    });

}
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
    console.log(`server is running on port ${PORT}.`);
});

        if(req.query.led2 === "on"){
            controlLED("LED2", req.query.POWER)
        }
        if(req.query.led3 === "on"){
            controlLED("LED3", req.query.POWER)
        }
        if(req.query.led4 === "on"){
            controlLED("LED4", req.query.POWER)
        }
    }

    if(req.query.botton2 === "Mode_shine"){
        controlLED("Mode_shine", req.query.time)
    }

    //res.send("Successfully Requested")
    //res.send('response')
    res.sendFile('/public/index.html', {root: __dirname })

})

function controlLED(LED, POWER){

    let child_process = require("child_process");
    console.log(`excuing`)

    let process =
child_process.execFile('sudo' ,['./C++/L4Program', LED, POWER
    ]);
    console.log(`success`)

    process.stdout.on('data', (data) =>{
        console.log(`stdout: ${data}`);
    });

    process.stderr.on('data', (data) =>{
        console.error(`stderr: ${data}`);
    });

}
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
    console.log(`server is running on port ${PORT}.`);
});
```

index.html

```
<!DOCTYPE html>

<style>
    .all{
        background:#e8fcef;
        width:600px;
        height:800px;
        border:10px solid #b3dbb1;
        margin:0 auto;
        text-align: center;

    }
    .all h1{
        font-size: 40px;
        font-weight:500px;
        padding-left:75px;
        padding-top:50px;
        font-style:italic;
        text-align: left;
    }
    .all p{
        font-size:20px;
        font-weight:200px;
        opacity:0.5;
    }
</style>

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=\, initial-
scale=1.0">
        <title>Nodejs</title>
    </head>
    <body>
        <div class="all">
            <p>
                <h1>Nodejs</h1>
            </p>
            <hr>
            <form action="/index" method="get">
                <h2>LED Control Panel</h2>
                <div>
                    <input type="checkbox" id="led1" name="led1"
>
                    <label for="LED1">LED1</label>

                    <input type="checkbox" id="led2" name="led2"
>
                    <label for="LED2">LED2</label>
                </div>
                <div>
                    <input type="checkbox" id="led3"
name="led3">
                    <label for="LED3">LED3</label>

                    <input type="checkbox" id="led4"
name="led4">
                    <label for="LED4">LED4</label>
                </div>

                <div>
                    <input type="radio" id="ON" name="POWER"
value="on">
                    <label for="ON">ON</label>
                    <input type="radio" id="OFF" name="POWER"
value="off">
                    <label for="OFF">OFF</label>
                </div>
                <input type="submit" name="botton1"
value="Submit">
                <hr>

                <h2>LED Switch Frequency</h2>
                <input type="text" name="time">
                <br>
                <input type="submit" name="botton2"
value="Mode_shine">
            </form>
        </div>

    </body>
</html>
```

led_control.cpp

```
#include <vector>
#include <iostream>
#include <string>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>

using namespace std;

int gpio_export(unsigned int gpio);
int gpio_unexport(unsigned int gpio);
int gpio_set_dir(unsigned int gpio, string dirStatus);
int gpio_set_value(unsigned int gpio, int value);

int main(int argc, char *argv[]){
      cout << argv[1][3] << argv[2] << endl;
      if(argv[1][0] == 'L'){
      if(argv[2][1] == 'n'){
            switch(argv[1][3]){
            case '1':
                  gpio_export(396);
                  gpio_set_dir(396, "out");
                  gpio_set_value(396,1);
                  break;
            case '2':
                  gpio_export(429);
                  gpio_set_dir(429, "out");
                  gpio_set_value(429,1);
                  break;
            case '3':

                  gpio_export(395);
                  gpio_set_dir(395, "out");
                  gpio_set_value(395,1);
                  break;
            case '4':
                  gpio_export(393);
                  gpio_set_dir(393, "out");
                  gpio_set_value(393,1);
                  break;
      }}
      else{
            switch(argv[1][3]){
            case '1':
                  gpio_set_value(396, 0);
                  gpio_unexport(396);
                  break;
            case '2':
                  gpio_set_value(429, 0);
                  gpio_unexport(429);
                  break;
            case '3':
                  gpio_set_value(395, 0);
                  gpio_unexport(395);
                  break;
            case '4':
                  gpio_set_value(393, 0);
                  gpio_unexport(393);
                  break;
}
      }}
      else{
            int len = std::stoi(argv[2]);
      for (int i = 0; i < 2*len; i++){
            cout << "i:" << i << endl;
            if(i % 2 == 0){
                  gpio_export(396);
                  gpio_set_dir(396, "out");
                  gpio_set_value(396,1);
                  gpio_export(429);
                  gpio_set_dir(429, "out");
                  gpio_set_value(429,1);

                  gpio_set_value(395, 0);
                  gpio_unexport(395);
                  gpio_set_value(393, 0);
                  gpio_unexport(393);
                  sleep(1);
            }
            else{
                  gpio_export(395);
                  gpio_set_dir(395, "out");
                  gpio_set_value(395,1);
                  gpio_export(393);
                  gpio_set_dir(393, "out");
                  gpio_set_value(393,1);

                  gpio_set_value(396, 0);
                  gpio_unexport(396);
                  gpio_set_value(429, 0);
                  gpio_unexport(429);
                  sleep(1);
}
}
                  gpio_set_value(395, 0);
                  gpio_unexport(395);
                  gpio_set_value(393, 0);
                  gpio_unexport(393);
                  gpio_set_value(396, 0);
                  gpio_unexport(396);
                  gpio_set_value(429, 0);
                  gpio_unexport(429);
}

      return 0;
}


int gpio_set_value(unsigned int gpio, int value){
      int fd;
      char buf[64];

      snprintf(buf, sizeof(buf), "/sys/class/gpio/gpio%d/value",
gpio);

      fd = open(buf, O_WRONLY);
      cout << fd << endl;
      if(fd < 0){
            perror("gpio/value");
            return fd;
}
      if(value == 0)
            write(fd, "0", 2);
      else
            write(fd, "1", 2);
      close(fd);
      return 0;
}

int gpio_set_dir(unsigned int gpio, string dirStatus){
      int fd;
      char buf[64];

      snprintf(buf, sizeof(buf),
"/sys/class/gpio/gpio%d/direction", gpio);

      fd = open(buf, O_WRONLY);
      cout << fd << endl;
      if(fd < 0){
            perror("gpio/direction");
            return fd;
}
      if(dirStatus == "out")
            write(fd, "out", 4);
      else
            write(fd, "in", 3);

      close(fd);
      cout << "dir-ed!" << endl;
      return 0;
}

int gpio_unexport(unsigned int gpio){
      int fd, len;
      char buf[64];

      fd = open("/sys/class/gpio/unexport", O_WRONLY);
      if(fd <0){
            perror("gpio/export");
            return fd;
      }
      len = snprintf(buf, sizeof(buf), "%d", gpio);
      write(fd, buf, len);
      close(fd);
      return 0;
}

int gpio_export(unsigned int gpio){
      int fd, len;
      char buf[64];


      fd = open("/sys/class/gpio/export", O_WRONLY);
      cout << fd << endl;
      if(fd <0){
            perror("gpio/export");
            return fd;
      }
      len = snprintf(buf, sizeof(buf), "%d", gpio);
      write(fd, buf, len);
      close(fd);
      cout << "exported!" << endl;
      return 0;
}
```

四、本次實驗過程說明與解決方法:

1. 實驗過程

   撰寫前後端程式碼 → 連線 tx2 → 將專案目錄送到 tx2 → 確認 tx2 是否已安裝所需套件 → 在瀏覽器執行 → 測試是否符合目標

2. 解決方法

   遇到最大的問題還是對於網頁設計的不熟悉，最後也只能靠自己四處尋找資料來解決。

五、分工:

學號、組員	貢獻比例	工作內容
B812110004 葉芸茜	50%	文書處理、實驗設計與實作、程式規劃、測試與除錯
B812110011 湯青秀	50%	文書處理、實驗設計與實作、程式規劃、測試與除錯