微算機系統實習

實驗4

日期:2021/04/23

1. 組員姓名：

第一組

資工二　108590044　何柏憲

資工二　108590452　林峻霆

1. 實驗步驟截圖與說明：

實驗目標:

• 利用nodejs 架設web伺服器

• 安裝npm套件  
• 撰寫html 和 js  
• 了解前後端參數傳遞  
• 利用 child\_process執行子程序

• 透過CGI網頁控制GPIO上的LED燈

實驗過程:

1. 我們先了解前後端的參數如何傳遞，也上網先複習了html 以及js 應該如何撰寫 (在這裡我們因為之謙修過網頁設計所以大概理解如何寫)

2. 我們將書寫好的程式碼在TX2上執行並從window 端連線測試我們的程式碼是否正確。

3. 結合lab2的執行檔案，以及這次的程式內容不斷連線debug

4. 修改子程序gpio2 以達到配合這次實驗目標

5. 測試：勾選指定的發光二極體，在點擊Submit可控制 TX2上的發光二極體

: 輸入閃爍次數，可間隔閃爍2組LED

以下是我們的程式碼：

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name ="viewport" content="width=device-width, initial-scale=1.0">

<title>Nodejs</title>

</head>

<body>

<p>

<h1>Nodejs</h1>

</p>

<hr>

<form action="/index">

<h1>LED Control Panel</h1>

<input type="checkbox" id="LED1" name="LED1" value="true" >

<label for="LED1"> LED1</label><br>

<input type="checkbox" id="LED2" name="LED2" value="true">

<label for="LED2"> LED2</label><br>

<input type="checkbox" id="LED3" name="LED3" value="true">

<label for="LED3"> LED3</label><br>

<input type="checkbox" id="LED4" name="LED4" value="true">

<label for="LED4"> LED4</label><br>

<input type="submit" value="submit">

<input type="radio" id="on" name="Switch" value="on">

<label for="on">On</label>

<input type="radio" id="off" name="Switch" value="off">

<label for="off">Off</label>

</form>

<hr>

<form action="/index1" method="get">

<h1>LED Switch Frequency</h1>

<label><input type="text" name="frequency"></label><br>

<input type="submit" value="Mode\_Shine">

</form>

</body>

</html>

server.js

const express = require("express")

const app = express()

app.use(express.static("../lab04"))

app.get("/index",(req,res)=>{

let q =req.query;

let response = {

"LED1" :[q.LED1,q.LED2,q.LED3,q.LED4],

"on&off" : q.Switch,

}

response.LED1.forEach((e,i) => {

if(e == "true")

controlLED(i+1,q.Switch);

});

res.send(response);

});

app.get("/index1",(req,res)=>{

let q =req.query;

shine(parseInt(q.frequency));

res.status(200).send("good");

});

let controlLED=(LED,Switch)=>{

let child\_process =require("child\_process");

let process = child\_process.execFile('sudo',["../lab02/gpio",LED,Switch]);

console.log('sudo',["../lab02/gpio",LED,Switch]);

process.stdout.on('data',(data)=>{

console.log(`stdout: ${data}`);

});

process.stderr.on('data',(data)=>{

console.error(`stdout: ${data}`);

});

}

let shine=(times)=>{

let child\_process =require("child\_process");

let process = child\_process.execFile('sudo',["../lab02/gpio2",times]);

// exec(times);

//let p = child\_process.execFile(times);

process.stdout.on('data',(data)=>{

console.log(`stdout: ${data}`);

});

process.stderr.on('data',(data)=>{

console.error(`stdout: ${data}`);

});

}

// set port,listen for requests

const PORT = process.env.PORT || 8080;

app.listen(PORT,()=>{

console.log(`Server is running on port ${PORT}.`);

});

因為我們有修改gpio2.cpp

以下是 gpio2.cpp

#include "gpio.h"

void turnOn(unsigned int target){

gpio\_export(target);

gpio\_set\_dir(target,"out");

gpio\_set\_value(target,1);

}

void turnOff(unsigned int target){

gpio\_set\_value(target,0);

gpio\_unexport(target);

}

int main(int arg,char \*argv[]){

int times = atoi(argv[1]);

int arr\_pin[4] = {396,397,255,481};

for(int i = 0 ; i < times\*2; i++){

if( i % 2 == 0)

{

turnOn(arr\_pin[0]);

turnOn(arr\_pin[1]);

turnOff(arr\_pin[2]);

turnOff(arr\_pin[3]);

}else{

turnOff(arr\_pin[0]);

turnOff(arr\_pin[1]);

turnOn(arr\_pin[2]);

turnOn(arr\_pin[3]);

}

sleep(1);

}

return 0;

}

影片連結：<https://youtube.com/shorts/wi7WhkEauEY?feature=share>

1. 組員貢獻比例：

何柏憲：50%

林峻霆：50%

1. 心得：

何柏憲：

　　這次的實驗有用到網頁的架構，還好之前有修過網頁程式設計，讓在寫網頁時沒有遇到太多困難，主要是後端的部分比較不熟悉。Nodejs主要是用JavaScript下去撰寫，平常都沒怎麼使用這個程式語言，導致一開始在理解程式碼上有些困難。不過後來還是跟我組員順利完成了這次的實驗。下禮拜要考試，有點擔心Makefile的部分，Linux我相當的熟悉所以比較不擔心。

林峻霆：

這次實驗延續了之前( lab2 )的實驗內容，以及上學期修網頁的基礎，我覺得非常的有趣，也往我更加了解前後端參數的傳遞，一開始以為以我們的網頁基礎應該可以快速的解決這次實驗，沒想到還是花了不少時間，所以我們還是上我複習了一下一些前後端的知識，最後還發現之前lab2的執行檔案需要更正也讓我們花了不少時間，希望下次我們可以更快的完成實驗，並且幫助他人，讓助教和教授可以早點下班。