

微算機系統實習

個人報告

Final Project
期末專案

組別： 13

班級、姓名與學號：

醫工三 B812110004 葉芸茜

日期：2024.06.17

1. 方案構思

最近天氣變得好炎熱，尤其是每天早晨 6 點多的時候我家的窗戶都會直接照進刺眼的陽光，想繼續睡都會被閃死，但我又很不想下床去拉窗簾，於是我希望設計一個能直接用網頁控制窗簾敞開度的系統(因為我都把手機放在我床旁邊哈哈)，這樣我就能不下床就能關上窗簾了！(如果對於我為什麼不直接一開始就一直把窗簾拉上，那是因為我每次都拉上但隔天家人又會把它拉開)

另外一點則是天氣好熱好想開冷氣，但每一個家電都分開操控實在很麻煩耶，假如能把它跟窗簾整合控制就好。另外還能加上利用光敏電阻和溫溼度感測器來感應亮度和濕度，整合這些資訊，能夠更好地打造適宜的環境，也能節電節省資源耗用。因此，"EaseControl 自在控"系統出現了！我們設計了以下幾點功能：

(1)項目一：設計 GUI 介面顯示於畫面

我們希望能構建一個可以調控冷氣溫度及窗簾敞開度的系統，並能達到以下幾點優點

a. 遠程控制：例如在回家途中可以提前開啟冷氣，使室內溫度在到家時已經舒適。或是假如已經出門在外，忘記關閉家電，可以透過這個系統遠程關閉，避免電力耗費。

b. 整合控制：將多個家居電器合併操作，提高生活便利性。

首先我們希望能設計一個 GUI 介面，介面會顯示當前冷氣溫度及窗簾敞開度，下方可以選擇要調控冷氣還是窗簾，選擇後會跳出原本隱藏的調控欄位。

a. 選擇 "air conditioner" 後可選擇要調節的溫度(範圍 5~40 度 C)。

b. 選擇 "curtain" 後可選擇要敞開的程度(範圍 0~100，0 為全關，100 為全開)。

調整完畢提交後，會更新選擇的項目的狀態於網頁上。另外，若選擇 "air conditioner" 後 LED1 亮起，選擇 "curtain" 後 LED2 亮起。

(2)項目二：使用到光敏電阻、LED、MCP3008

我們希望能構建一個可以實時偵測窗外亮度的系統，這樣帶來的好處有

a. 根據窗外亮度調節室內照明。

b. 根據窗外亮度調節窗簾的開合，保持室內適當的光線。

c. 分析外部亮度和天氣變化之間的關係。

設計的功能包括，在 detect you! 欄位選擇 "light" 後可偵測窗外亮度，並顯示於網頁上。根據實時亮度可控制 LED3、LED4 的狀態，光敏電阻閾值範圍如下

a. adc_value: > 450 -> LED3、LED4 閃爍

b. adc_value: 151~450 -> LED3 閃爍

c. adc_value: <=150 -> 無變化

(3)項目三：驅動程式(driver)控制嵌入式設備

在項目一時當我們選擇功能提交後，會開啟對應的 LED，在此處我們使用 driver 去控制 LED 的狀態。通過掛載驅動程式來實現，控制 LED 時會顯示系統訊息在終端上(包含掛載、開檔、讀檔、寫檔、卸載等...)。預計達到以下功能

- (1) 調控 air-conditioner 溫度時，LED1 亮
- (2) 調控 curtain 敞開度時，LED2 亮
- (3) 點選 reset，關閉所有 LED

(4)項目四：使用溫溼度感測器 DHT11、MCP3008

我們希望能構建一個可以實時偵測當前溫度濕度的系統，其優點如下

- a. 根據當前溫度濕度調整家電，保持室內適宜，並避免耗電
- b. 分析溫溼度和天氣變化之間的關係

設計的功能包括，在 detect you! 欄位選擇" temperature" 後可偵測目前室內溫度，並顯示於網頁上。

(5)項目五：寄送 Email

希望可以有一個功能能夠寄送郵件來通知目前的冷氣狀態及窗簾敞開度，除了可以用來記錄自己調整的狀態，也可以寄送給他人並留下附註的話語。我們提供兩個輸入框供使用者輸入

- a. 填寫郵件信箱
- b. 可附註在信中的話

按下提交鍵，則系統會寄送目前的冷氣溫度及窗簾狀態，以及使用者自填的附註話語到填寫的郵件信箱。

2. 實驗步驟

(1)項目一：設計 GUI 介面顯示於畫面

- a. 原本網頁預設狀態
 - 當前冷氣溫度(預設為 25 度)
 - 當前窗簾敞開度(預設為 50 度)

Your data NOW!

your air-conditioner's temp

25

degree

your curtain open percentage

50

%

```
1 <div class="input">
2     <div class="dataShow">
3         <h2>Your data NOW!</h2>
4         your air-conditioner's temp<div id="tempShow" method="post">
5             000 </div> degree
6         <hr align=center width=30% >
7         your curtain open percentage<div id="openShow" method="post">
8             000
9         </div> %
10    </div>
11 </div>
```

b. 點選功能會顯示下方可調欄位

點選對應 radio 會顯示相應欄位，另一個功能欄位維持隱藏

此程式碼可以選擇兩種功能： air-conditioner、curtain

```
1 <form class="input">
2     <div class="functionSelect">
3         <h2>Select what you want to control!</h2>
4         <input type="radio" id="air-con" name="function" value="air" onclick="showOpenClose(); hideSliding();">
5         <label for="air-con">air conditioner</label>
6
7         <input type="radio" id="curtain" name="function" value="cur" onclick="showSliding(); hideOpenClose();">
8         <label for="curtain">curtain</label>
9         <!--<input class="submit" type="submit" name="functionButton" value="submit"-->
10        <hr>
11    </div>
12 </form>
```

```
1 function showSliding() {
2     document.getElementById('slideControl').style.display = 'block';
3     //document.getElementById('slideControl').style.transition = 'all 1s ease-in 0.1s' ;
4 }
5
6 function hideSliding() {
7     document.getElementById('slideControl').style.display = 'none';
8 }
9
10 function showOpenClose() {
11     document.getElementById('openControl').style.display = 'block';
12 }
13
14 function hideOpenClose() {
15     document.getElementById('openControl').style.display = 'none';
16 }
```

● 冷氣調整欄位(5~40，預設 25)

Select what you want to control!

☒ air conditioner
 ☐ curtain

Setting temperature

後端接收冷氣調整的數值，並利用 ajax 更新在當前狀態

```

1 // 窗簾敞開程度
2 app.get("/index1", (req, res) => {
3   console.log(`Received slide value: ${slideValue}`);
4   res.send(`${slideValue}`);
5 })
6
7 app.get("/slideControl", (req, res) => {
8   slideValue = req.query.slideValue;
9   console.log(`Received temperature value: ${slideValue}`);
10  //res.send( req.query.slideValue)
11
12  ledshine("LED1");
13  res.redirect(url.format({
14    pathname: "/",
15  }));
16 })
  
```

前端調整冷氣溫度

因為直接 send response 會回傳預設值 25，所以寫了個 hidden 來抓取實時更新的數值

```

1 <form class="input" action="/openControl" method="get" onsubmit="updateHiddenTemperatureValue();">
2   <div class="open" id="openControl">
3     <h2>Setting temperature</h2>
4     <input type="number" id="temperature" class="kkk" min="5" max="40" oninput="updateTemperatureValue(this.value)">
5     <input type="hidden" id="hiddenTemperatureValue" name="temperature" value="25">
6     <input class="submit" type="submit" name="openButton" value="submit">
7   </div>
8 </form>
9
1  function updateTemperatureValue(value) {
2    document.getElementById('hiddenTemperatureValue').value = value;
3  }
4
5  function updateHiddenTemperatureValue() {
6    const temperature = document.getElementById('temperature');
7    const hiddenTemperatureValue = document.getElementById('hiddenTemperatureValue');
8    hiddenTemperatureValue.value = temperature.value;
9  }
  
```

- 窗簾調整欄位(0~100，預設 50)

Select what you want to control!

☐ air conditioner
 ☒ curtain

Sliding your Curtain

後端接收窗簾調整的數值，並利用 ajax 更新在當前狀態

```

1 // 冷氣溫度
2 app.get('/index2', (req, res) => {
3   console.log(`Received temperature value: ${temperature}`);
4   res.send(`${temperature}`);
5 });
6
7 app.get("/openControl", (req, res) => {
8   temperature = req.query.temperature;
9   console.log(`Received temperature value: ${temperature}`);
10  ledshine("LED2");
11  //res.send( req.query.temperature)
12  res.redirect(url.format({
13    pathname: "/",
14  }));
15 })

```

前端調整窗簾敞開度，大致上與冷氣的作法一樣

```

1 <form class="input" action="/slideControl" method="get" onsubmit="updateHiddenSliderValue();">
2   <div class="slide" id="slideControl">
3     <h2>Sliding your Curtain</h2>
4     <input type="range" id="slider" min="0" max="100" step="1" value="50" oninput="updateSliderValue(this.value)">
5     <span id="sliderValue">50</span>
6     <input type="hidden" id="hiddenSliderValue" name="slideValue" value="50">
7     <input class="submit" type="submit" name="slideButton" value="submit">
8     <hr>
9   </div>
10 </form>

```

c. 提交後可更新當前狀態

ajax 在提交後變更當前狀態

```

1 $('#openShow').css("fontSize", 30);
2 GetResultOpen();
3 function GetResultOpen(){
4   $.ajax({
5     type:"get",
6     url:"/index1",
7     success:function (result){
8       console.log("Item", result)
9       $('#openShow').text(result);
10    },
11    error:function(){
12      alert("error");
13    }
14  })
15 }
16
17 $('#tempShow').css("fontSize", 30);
18 GetResultTemp();
19 function GetResultTemp(){
20   $.ajax({
21     type:"get",
22     url:"/index2",
23     success:function (result){
24       console.log("Item", result)
25       $('#tempShow').text(result);
26    },
27    error:function(){
28      alert("error");
29    }
30  })
31 }

```

- 將當前冷氣從 25 -> 16 度

Your data NOW!

your air-conditioner's temp
25
degree

your curtain open percentage
50
%

Select what you want to control!

☒ air conditioner
☐ curtain

Setting temperature

your air-conditioner's temp
16
degree

your curtain open percentage
50
%

Select what you want to control!

☐ air conditioner
☐ curtain

- 將當前窗簾從 50 -> 74 %

Your data NOW!

your air-conditioner's temp
16
degree

your curtain open percentage
50
%

Select what you want to control!

☐ air conditioner
☒ curtain

Sliding your Curtain

your air-conditioner's temp
16
degree

your curtain open percentage
74
%

Select what you want to control!

☐ air conditioner
☐ curtain

(2)項目二：使用到光敏電阻、LED、MCP3008

- 在 detect you! 欄位選擇 light，可開啟光敏電阻的實時偵測

detect you!

☐ light
☐ temperature

```

1 <form class="input" action="/index4" method="get" onsubmit="return false;">
2   <div class="Detecting">
3     <h2>detect you!</h2>
4     <input type="radio" id="light" name="detecting" value="light">
5     <label for="light">light</label>
6     <input type="radio" id="temp" name="detecting" value="temperature">
7     <label for="temperature">temperature</label>
8     <input class="submit" type="submit" id="detectButton" name="detectButton" value="Submit">
9   </div>
10 </form>
11

```


- b. 選擇 light 後，跳出下方欄位並顯示光敏數值

<div>detect you!</div> <div><input checked="" type="radio"/> light <input type="radio"/> temperature</div> <div>Submit</div>	<div>your light</div> <div>255</div> <div>Finish</div>
---	---

控制數值展示頁面(展示)

```
1 document.getElementById('detectButton').onclick = function () { showDetect() };
2
3 function showDetect() {
4     if (document.getElementById('light').checked) {
5         //alert("light")
6         document.getElementById('DetectShow').style.display = 'block';
7         //alert("light")
8     }
9     else if (document.getElementById('temp').checked) {
10        document.getElementById('DetectShow2').style.display = 'block';
11        //alert("temp")
12    }
13    else {
14        alert("bad");
15    }
16 }
```

控制數值展示頁面(隱藏)

```
1 document.getElementById('detectlightButton').onclick = function () { hideDetect() };
2
3 function hideDetect() {
4     document.getElementById('DetectShow').style.display = 'none';
5 }
```

ajax 持續呼叫/detect1，依據回傳值修改標籤內容

```
1 $("#lightres").css("fontSize", 30);
2     setInterval(GetResult1, 500);
3     function GetResult1(){
4         $.ajax({
5             type:"get",
6             url:"/detect1",
7             success:function (result){
8                 console.log("Item", result)
9                 $('#lightres').text(result);
10            },
11            error:function(){
12                alert("error");
13            }
14        })
15    }
```

/detect1 路由呼叫 detect1()並回傳結果


```

1  app.get("/detect1", (req, res) => {
2    //console.log("dsfsdf")
3    detect1()
4      .then(function (data) {
5        //console.log("send1")
6        res.send(data)
7      })
8      .catch(error => {
9        console.log(err)
10     })
11  })

```

detect2()函數執行 gpio_test.py 可以得到回傳的數值

```

1  function detect1() {
2    light = "light";
3    var p = new Promise(function (resolve, reject) {
4
5      let child_process = require("child_process");
6
7      let process = child_process.spawn('python', [
8        "./gpio_test.py", light
9      ]);
10
11     process.stdout.on('data', (data) => {
12       console.log(`stdout: ${data}`);
13       resolve(data)
14       //return data;
15     });
16
17     process.stderr.on('data', (data) => {
18       console.error(`stderr: ${data}`);
19       reject(data)
20     });
21   })
22   return p
23 }

```

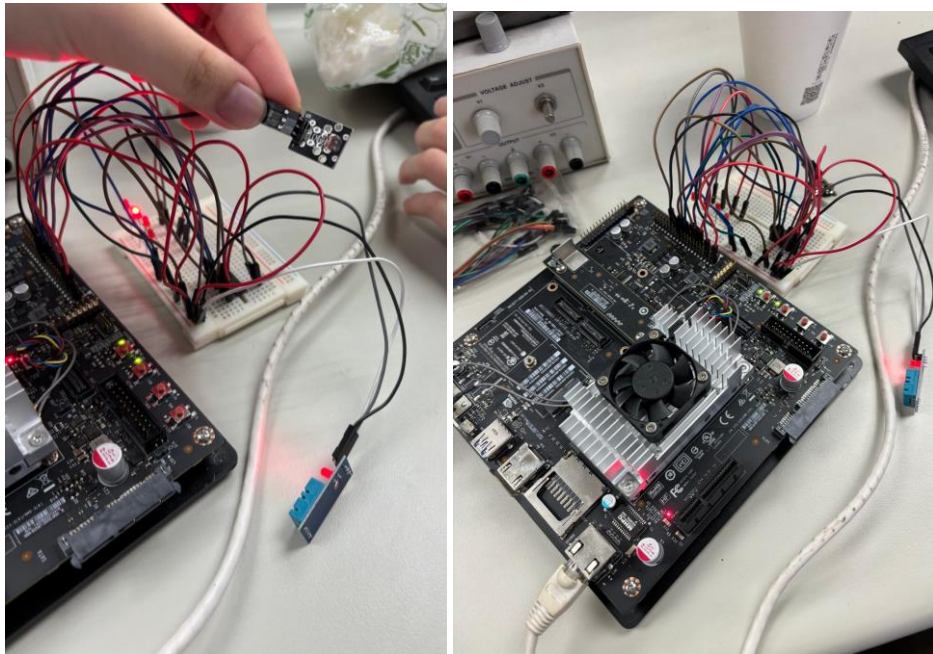
根據光敏電阻控制兩顆 LED 狀態的閾值

```

1  if(adcout > 450):
2      #led2 bright
3      GPIO.output(output2, GPIO.HIGH)
4      #led1 bright
5      GPIO.output(output1, GPIO.HIGH)
6  elif(adcout > 150):
7      #led1 bright
8      GPIO.output(output1, GPIO.HIGH)
9      #led2 dark
10     GPIO.output(output2, GPIO.LOW)
11 else:
12     #led1 dark
13     GPIO.output(output1, GPIO.LOW)
14     #led2 dark
15     GPIO.output(output2, GPIO.LOW)

```

- c. 按下 finish 可結束偵測，隱藏該欄位
- d. 光敏電阻與 MCP3008 的接腳如下



(3) 項目三：驅動程式(driver)控制嵌入式設備

- a. 繼項目一，點選 air-conditioner 並提交按鍵後，呼叫 ledshine 並傳要控制的 LED 參數，Ledshine 會呼叫 controLED，掛載 demo.ko，並呼叫 test.o 來開啟 LED1
冷氣會傳 LED1 給 ledshine

```

1  app.get("/slideControl", (req, res) => {
2      slideValue = req.query.slideValue;
3      console.log(`Received temperature value: ${slideValue}`);
4      //res.send( req.query.slideValue)
5
6      ledshine("LED1");
7      res.redirect(url.format({
8          pathname: "/",
9      }));
10 })

```

冷氣會傳 LED2 給 ledshine

```

1  app.get("/openControl", (req, res) => {
2      temperature = req.query.temperature;
3      console.log(`Received temperature value: ${temperature}`);
4      ledshine("LED2");
5      //res.send( req.query.temperature)
6      res.redirect(url.format({
7          pathname: "/",
8      }));
9  })

```

開啟傳入的 LED 燈

```

1  async function ledshine(LED) {
2      try {
3          await insmod();
4          await controLED(LED, "on");
5          console.log(`on`);
6
7          //setTimeout(() => {
8              //    controLED(LED, "off");
9              //    console.log(`off`);
10             //}, 3000);
11         await rmmmod();
12     } catch (error) {
13         console.error(`Error: ${error.message}`);
14     }
15 }

```

掛載 demo.ko

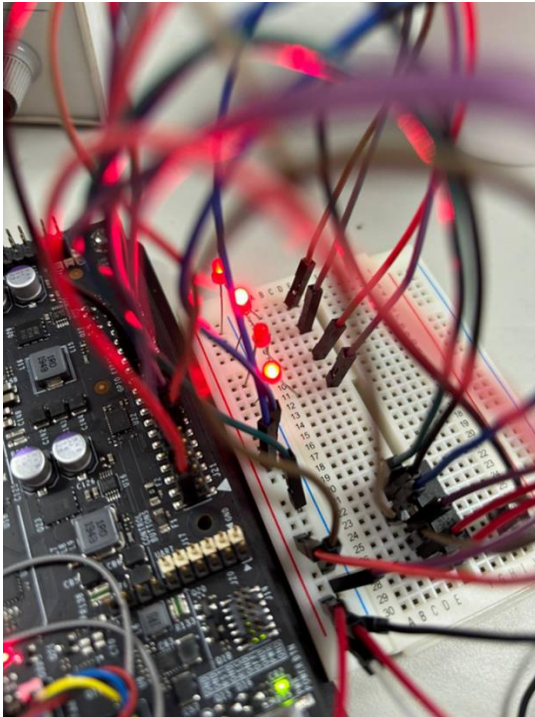
```

1  async function insmod() {
2      return new Promise((resolve, reject) => {
3          const { exec } = require('child_process');
4
5          const command = 'sudo insmod driver/demo.ko';
6
7          exec(command, (error, stdout, stderr) => {
8              if (error) {
9                  console.error(`error: ${error.message}`);
10                 return;
11             }
12             console.log("insmod")
13             resolve();
14         });
15     });
16 }

1  async function controLED(LED, POWER) {
2      console.log(LED)
3      console.log(POWER)
4      return new Promise((resolve, reject) => {
5          let child_process = require("child_process");
6
7          let process = child_process.execFile('sudo', [
8              "./driver/test.o", LED, POWER,
9          ]);
10
11          process.stdout.on('data', (data) => {
12              console.log(`stdout: ${data}`);
13          })
14
15          process.stdout.on('data', (data) => {
16              console.log(`stderr: ${data}`);
17          })
18
19          process.on('exit', (code) => {
20              if (code === 0) {
21                  resolve();
22              } else {
23                  reject(new Error(`Process exited with code ${code}`));
24              }
25          });
26     });
27 }

```

圖為 LED1 開啟，另外點選 curtain 則是開啟 LED2



開啟完對應 LED 後會卸載 demo.ko

```
1  async function rmmod() {
2      return new Promise((resolve, reject) => {
3          const { exec } = require('child_process');
4          const command = 'sudo rmmod driver/demo.ko';
5
6          exec(command, (error, stdout, stderr) => {
7              if (error) {
8                  reject(error);
9                  return;
10             }
11             console.log("rmmod")
12             resolve();
13         });
14     });
15 }
```

dmesg 查看系統訊息


```

7365.137053] <1>demo removed
7365.272853] <1>demo: started
7365.296700] Open: Enter Open function
7365.296737] I/O Control: Enter I/O Control function
7365.296747] Write: Enter Write function
7365.296825] Release: Enter Release function
7365.319789] Open: Enter Open function
7365.319822] I/O Control: Enter I/O Control function
7365.319832] Write: Enter Write function
7365.319909] Release: Enter Release function
7365.347555] Open: Enter Open function
7365.347583] I/O Control: Enter I/O Control function
7365.347594] Write: Enter Write function
7365.348043] Release: Enter Release function
7365.374801] Open: Enter Open function
7365.374856] I/O Control: Enter I/O Control function
7365.374872] Write: Enter Write function
7365.375006] Release: Enter Release function
7365.404038] <1>demo removed
7455.605012] <1>demo: started
7455.631060] Open: Enter Open function
7455.631091] I/O Control: Enter I/O Control function
7455.631102] Write: Enter Write function
7455.631184] Release: Enter Release function
7455.652742] <1>demo removed
7461.159778] <1>demo: started
7461.183832] Open: Enter Open function
7461.183871] I/O Control: Enter I/O Control function
7461.183882] Write: Enter Write function
7461.183961] Release: Enter Release function
7461.211843] <1>demo removed
7512.988532] <1>demo: started
7513.015986] Open: Enter Open function
7513.016013] I/O Control: Enter I/O Control function
7513.016025] Write: Enter Write function
7513.016102] Release: Enter Release function
7513.046663] Open: Enter Open function
7513.046728] I/O Control: Enter I/O Control function
7513.046749] Write: Enter Write function
7513.046906] Release: Enter Release function
7513.073417] Open: Enter Open function
7513.073445] I/O Control: Enter I/O Control function
7513.073456] Write: Enter Write function
7513.073997] Release: Enter Release function
7513.098958] Open: Enter Open function
7513.099007] I/O Control: Enter I/O Control function
7513.099022] Write: Enter Write function
7513.099130] Release: Enter Release function
7513.124512] <1>demo removed

```

b. Reset 來關閉所有 LED

```

1  async function resetLED() {
2      console.log("ddd")
3      try {
4          await insmod();
5          await controLED("LED1", "off");
6          await controLED("LED2", "off");
7          await controLED("LED3", "off");
8          await controLED("LED4", "off");
9          await rmmod();
10     } catch (error) {
11         console.error(`Error: ${error.message}`);
12     }
13 }

```

```

1 <form class="input" action="/ddd" method="get">
2   <div class="reset">
3     <input class="submit" type="submit" id="reset" name="reset" value="Reset">
4   </div>
5 </form>

```

```

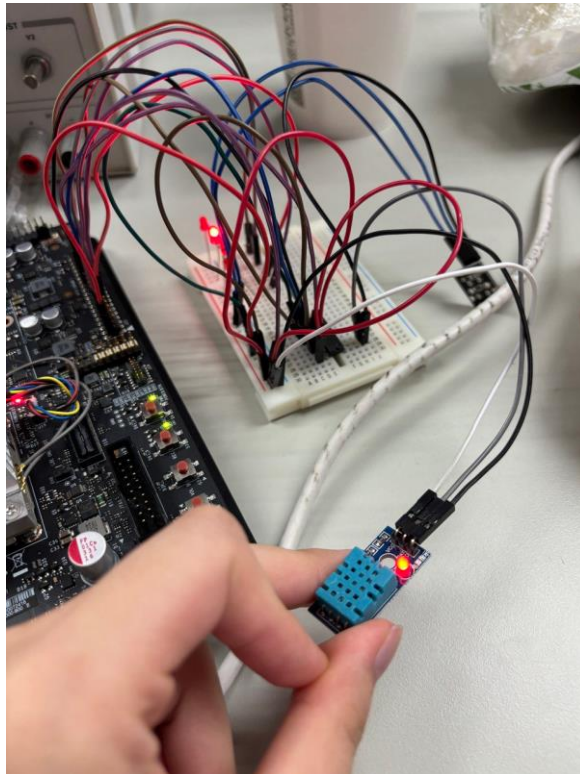
1 app.get("/ddd", (req, res) => {
2   console.log("reset")
3   resetLED();
4   res.redirect(url.format({
5     pathname: "/",
6   }));
7 })

```

(4)項目四：使用溫溼度感測器 DHT11

- a. 與光敏電阻相同，同樣使用 MCP3008 來實現，因為它是八類比數位轉換器，我們使用另一個腳來接 DHT11。

接腳如下

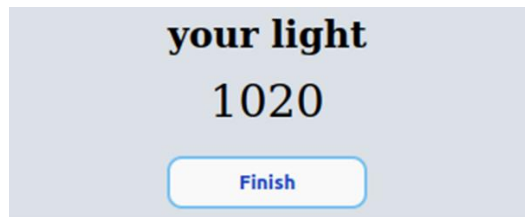


- b. 在 detect you! 欄位選擇 temperature，可開啟溫溼度感測器的實時偵測

detect you!

☐ light
☒ temperature

- c. 選擇 temperature 後，跳出下方欄位並顯示溫溼度



控制數值展示頁面(隱藏)

```
1 document.getElementById('detecttempButton').onclick = function () { hideDetect2() };
2
3 function hideDetect2() {
4     document.getElementById('DetectShow2').style.display = 'none';
5 }
```

ajax 持續呼叫/detect2，依據回傳值修改標籤內容

```
$("#tempres").css("fontSize", 30);

setInterval(GetResult2, 1000);
function GetResult2(){
    $.ajax({
        type:"get",
        url:"/detect2",
        success:function (result){
            console.log("Item", result)
            $('#tempres').text(result);
        },
        error:function(){
            alert("error");
        }
    })
}
```

/detect2 路由呼叫 detect2()並回傳結果

```
1 app.get("/detect2", (req, res) => {
2     //console.log("sfsfs")
3     detect2()
4     .then(function (data) {
5         //console.log("send2")
6         res.send(data)
7     })
8     .catch(error => {
9         console.log(err)
10    })
11 })
```

detect2()函數執行 gpio_test.py 可以得到回傳的數值

```

1  function detect2() {
2      temp = "temp";
3      var p = new Promise(function (resolve, reject) {
4
5          let child_process = require("child_process");
6
7          let process = child_process.spawn('python', [
8              "./gpio_test.py", temp
9          ]);
10
11         process.stdout.on('data', (data) => {
12             console.log(`stdout: ${data}`);
13             resolve(data)
14             //return data;
15         });
16
17         process.stderr.on('data', (data) => {
18             console.error(`stderr: ${data}`);
19             reject(data)
20         });
21     })
22     return p
23 }

```

(5)項目五：寄送 Email

- a. 有兩個輸入框供使用者輸入，包含想寄送的郵件地址，以及想另外附註的話

The form has a light blue background. At the top, it says "email you!" in bold black text. Below that, there are two input fields. The first is labeled "Your email:" and the second is labeled "Your text:". Both fields have a blue border and a dashed blue outline. Below the fields is a horizontal line, and then a rounded rectangular button with the text "submit" in blue.

```

1  <form class="input" action="/sending" method="get">
2      <div class="Email">
3          <h2>email you!</h2>
4          Your email: <input type="text" id="userMail" name="mail" class="message" value="">
5          <br>
6          Your text: <input type="text" id="userText" name="message" class="message" value="">
7          <hr align=center width=50%>
8          <input class="submit" type="submit" name="emailButton" value="submit" onclick="clear();">
9      </div>
10 </form>

```

- b. 傳送的郵件格式如下
 郵件標題會包含當前寄送的時間
 內容包含冷氣及窗簾目前狀態
 下方會顯示附註話語



get 到 /sending 後，呼叫 sendEmail 寄送郵件

```
1 app.get("/sending", (req, res) => {
2   console.log(req.query.mail);
3   console.log(req.query.message);
4   sendEmail(req.query.mail, req.query.message, temperature, slideValue);
5   res.redirect(url.format({
6     pathname: "/",
7   }));
8 })
```

```

1  function sendEmail(mail, userText, mailTemp, mailOpen) {
2      var today = new Date();
3
4      let inputSubject = `[來自自在控小雷擊落客-${today}]`;
5
6      let inputText = `您好
7
8  這是我們的測試郵件，造成您的不便敬請見諒。
9  我是您的自在控小雷擊落客。
10  您的冷氣現在為${mailTemp}度
11  床簾開啟狀態為${mailOpen}%
12  過去的您給自己的提醒是：
13  `;
14
15      let sign = `
16  以上
17  --
18  智慧是人工的 但我對您的愛不是
19
20  Sincerely,
21  Email You
22  -----
23  0952-520-520
24  emailJUSTFORYOUUUUU@gmail.com`;
25
26      //宣告發信物件
27      const transporter = nodemailer.createTransport({
28          service: 'Gmail',
29          auth: {
30              user: 'emailjustforyouuuuu@gmail.com',
31              pass: 'pazbtldskeucvvef' // 這是真的
32          }
33      });
34
35      var options = {
36          //寄件者
37          from: 'Email You <emailjustforyouuuuu@gmail.com>',
38          //收件者
39          to: mail,
40          //副本
41          cc: 'b812110011@tmu.edu.tw',
42          // //密件副本
43          //bcc: 'account4@gmail.com',
44          //主旨
45          subject: inputSubject, // Subject line
46          //純文字
47          text: inputText + userText + sign,
48      };
49
50      //console.log(text);
51
52      //發送信件方法
53      transporter.sendMail(options)
54          .then(info => {
55              //console.log({ info });
56          }).catch(console.error);
57
58      console.log("send finish");
59  }

```

3. 專案心得

透過這次的專案將過去的實驗及課程融合實作出一個智能家居系統，包括使用 driver，使用 ajax 來顯示數值等等。當然這次的實驗中也有遇到一些問

題，例如我們最熟悉的接線問題，在 lab6 中我們也接錯了無數次，這次也發生了跟 lab6 一樣的悲劇，所以一開始的光敏電阻一直顯示 0。另外就是學習到了一些前後端的撰寫，一開始完全不知道可以隱藏欄位，原來可以通過 onclick 來呼叫欄位，且 slider 的數值一開始也沒辦法抓到，每一次回傳都會回傳預設值，後來通過另外建立一個 type hidden 並做實時更新數值來回傳最後提交的數值，最後能順利保存這個變數做進一步使用。另外還有像是寄送 email，它的格式或是我們希望它可以新增的一些資訊，也通過慢慢測試來實現。

這次的專案實作過程中，我們也進一步加深了對於硬體和軟體整合的理解。例如在項目三中，透過撰寫驅動程式來控制嵌入式設備，實際感受到驅動程式的作用及其對硬體的影響。另外，在項目二與項目四中，利用光敏電阻和溫溼度感測器，通過 MCP3008 來進行數據讀取，這過程讓我們更加熟悉了類比數位轉換器的使用及其應用場景。

這次的專案還讓我們有機會學習到更多的網頁設計技巧，特別是如何透過 AJAX 來進行資料的即時更新和顯示，像是之前我們是使用跳轉頁面來顯示，這次則是嘗試在同一個頁面進行更新顯示。

此次設計的智能家居系統，讓我們了解到如何整合多種感測器和設備來實現對家居環境的智能化控制，也對整學期的實驗有了進一步的理解。

4. 小組分工

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