

嵌入式系統
期末專題
Medicine Control System

資科三甲 110816003 林佩瑩

資科三甲 110816032 謝好婕

目錄

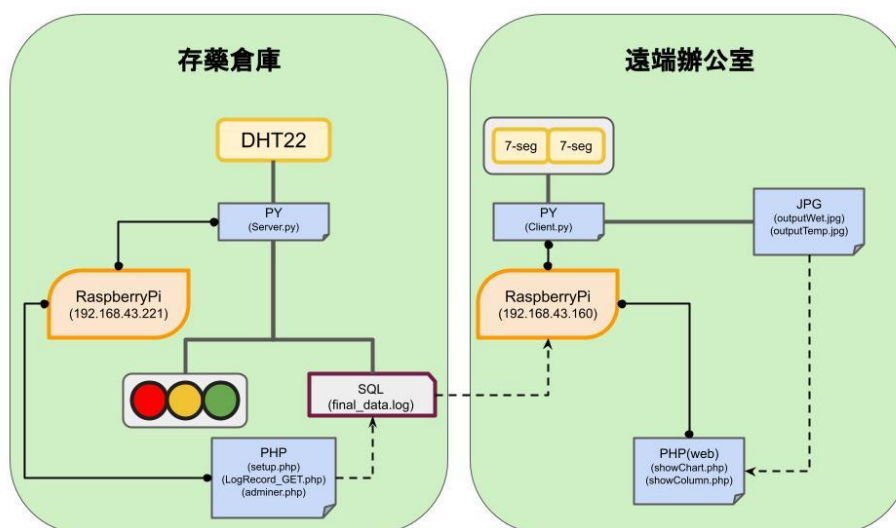
一、小組分工與討論	2
二、Medicine Control System專題目的	2
三、硬體線路圖	3
四、程式說明	4
Server部分	4
setup(.php)	4
LogRecord_GET(.php)	5
adminer(.php)	5
Server(.py)	6
Client部分	7
showChart(.php)	7
showColumn(.php)	7
Client(.py)	8
五、專題DEMO	10
六、心得	11
七、參考資料	11

一、小組分工與討論

直接約在學校圖書館一樓電腦教室討論 OuO!

佩瑩負責 Server 端的程式撰寫、資料庫建立及用戶創建、手繪硬體線路圖；好婕負責 Client 端的程式撰寫、用 fritzing 畫接線圖、用 Google 簡報畫系統方塊圖、書面報告整合。全程都在圖書館一起完成，有問題也可以馬上發現討論解決方法。

二、Medicine Control System 專題目的



因應近年來的肺炎風波，醫藥問題也隨之受到重視，Medicine Control System 提供一個可以遠距監控存藥倉庫的方法，讓員工可以利用 IOT 概念實現人與藥物零接觸的工作環境。

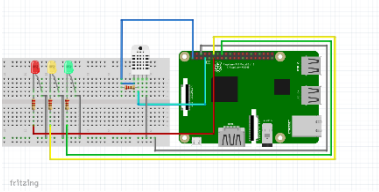
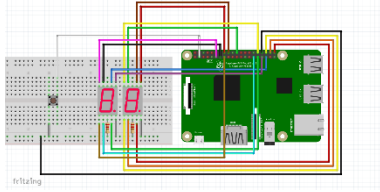
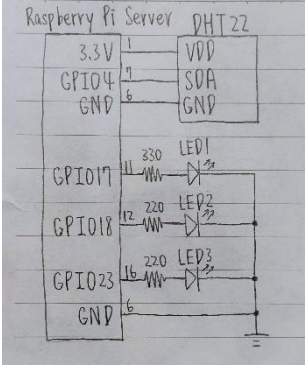
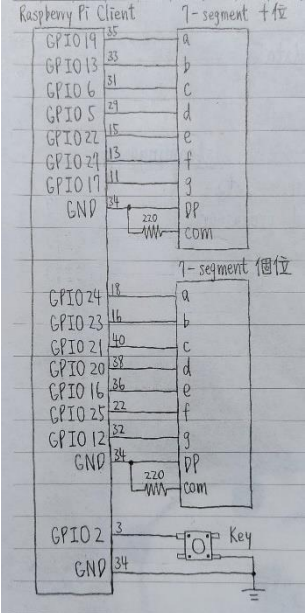
如圖所示，Medicine Control System 主要分為兩個部分，分別是 Server 端的【存藥倉庫】，和 Client 端的【遠端辦公室】。

Server 負責監測存藥環境的溫度與溼度，並判斷當下環境是否符合藥物存放的標準，如果溫度與溼度兩項皆符合標準則亮起綠色 LED，若其中一項未符合標準則亮起黃色 LED，而兩項皆未符合標準則亮起紅色 LED。

用戶需先使用 setup.php 建立資料庫後，再執行 Server.py 程式檔，即可透過 LogRecord_GET.php 將監測的時間、溫度、溼度及燈號資料傳送至資料庫，以提供用戶進行後續的資料應用。

Client 負責擷取遠端資料庫的資料並將其轉為 DataFrame 結構，再以 Matplotlib 套件將資料視覺化，最後輸出為折線圖。Client.py 提供用戶可在執行 python 檔後，按下 0 結束程式；按下 1，透過網頁瀏覽的方式得知倉庫的環境狀態變化(showChart.php、showColumn.php)；按下 2，執行上述功能，並在有七段顯示器及按鈕時，直接讓七段顯示器顯示最新的溫度，而用戶可透過按下按鈕切換為顯示溼度。

三、硬體線路圖

Server 接線圖	Client 接線圖
 (link)	 (link)
Server 硬體線路圖(電路圖)	Client 硬體線路圖(電路圖)
 (link)	 (link)

四、程式說明

以下僅就各檔案內重要內容說明，詳情請見附檔或[專案完整程式碼 \(GitHub\)](#)，若使用網頁瀏覽器可點擊[程式說明\(HackMD 網頁版\)](#)以獲得較佳的瀏覽體驗。

● Server 部分

■ setup(.php)

1. 設定伺服器名稱、使用者帳號及使用者密碼的參數。

```
$host = "localhost";  
$dbuser = "root";  
$dbpasswd = "123456";
```

2. 設定資料庫名稱為 final_data，資料表名稱為 log，以及資料表內的各項欄位名稱，依序為索引值(index_id)、時間(time)、溫度(temp)、溼度(wet)與燈號(light)。

```
$create_db_sql = "CREATE DATABASE IF NOT EXISTS `final_data`";  
$create_table_sql = "CREATE TABLE IF NOT EXISTS `final_data`.`log` (  
    `index_id` INT NOT NULL AUTO_INCREMENT ,  
    `time` TIMESTAMP NOT NULL ,  
    `temp` VARCHAR(20) NOT NULL ,  
    `wet` VARCHAR(20) NOT NULL ,  
    `light` VARCHAR(20) NOT NULL ,  
    PRIMARY KEY (`index_id`)  
)  
ENGINE = InnoDB";
```

3. 建立連線，並建立資料庫及資料表，若皆成功則顯示 **All OK**。

```
$conn = mysqli_connect($host, $dbuser, $dbpasswd);  
  
if(!$conn){  
    echo "Error: Unable to connect to MySQL." . PHP_EOL;  
    echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;  
    echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;  
    die("無法連線至資料庫");  
}  
$ret = mysqli_query($conn, $create_db_sql);  
if(!$ret){  
    echo "Error: Unable to connect to MySQL." . PHP_EOL;  
    echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;  
    echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;  
    die("建立 db 失敗");  
}  
// 設定連線編碼  
mysqli_query($conn, "SET NAMES 'utf8'");  
  
// 建立 table  
$ret = mysqli_query($conn, $create_table_sql);  
  
if(!$ret){  
    echo "Error: Unable to connect to MySQL." . PHP_EOL;  
    echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;  
    echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;  
    die("建立 table 失敗");  
}  
echo "All OK";  
mysqli_close($conn);
```

■ LogRecord_GET(.php)

1. 設定伺服器名稱、使用者帳號、使用者密碼及資料庫名稱的參數。

```
$host = "localhost";  
$dbuser = "root";  
$dbpasswd = "123456";  
$DBNAME = "final_data";
```

2. 如果欄位 temp 有讀取到內容，建立連線，並在資料表內新增一筆資料，其中包含 time, temp, wet, light 四項欄位的內容，若皆成功則顯示 **Records created successfully**。

```
if(isset($_GET['temp'])){  
    $conn = mysqli_connect($host, $dbuser, $dbpasswd, $DBNAME);  
    if (!$conn) {  
        echo "Error: Unable to connect to MySQL." . PHP_EOL;  
        echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;  
        echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;  
        die ("無法連接資料庫");  
    }  
    // 設定連線編碼  
    mysqli_query($conn, "SET NAMES 'utf8'");  
  
    $sql = sprintf("INSERT INTO log (`time`,`temp`,`wet`,`light`)  
        VALUES (NOW(),'%s','%s','%s');",  
        $_GET['temp'],$_GET['wet'],$_GET['light']  
    );  
    $ret = mysqli_query($conn, $sql);  
  
    if (!$ret){  
        echo "Error: Unable to connect to MySQL." . PHP_EOL;  
        echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;  
        echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;  
        die("insert error");  
    }else{  
        echo "Records created successfully\n";  
    }  
    mysqli_close($conn);  
}else{  
    die("Error");  
}
```

■ adminer(.php)

1. 輸入使用者帳號及密碼登入後，即可查看資料庫及資料表的內容，也可以刪除資料庫。由於此程式檔內容與老師上課範例相同，故不作詳細的程式說明。
2. 由管理員帳號登入後編輯 wpuser(Client 端帳號)權限，使其可以閱覽資料表。

權限

建立使用者

帳號	伺服器	
mariadb.sys	localhost	編輯
mysql	localhost	編輯
phpmyadmin	localhost	編輯
pi	localhost	編輯
root	localhost	編輯
username	localhost	編輯
wpuser	localhost	編輯

■ Server(.py)

1. ledOff()

讓三個 LED 都不亮，維持 1 秒。

```
GPIO.output(led1, False)
GPIO.output(led2, False)
GPIO.output(led3, False)
sleep(1)
```

2. led1_On()

只有 led1(紅色 LED)亮起，維持 3 秒。

```
GPIO.output(led1, True)
GPIO.output(led2, False)
GPIO.output(led3, False)
sleep(3)
```

3. led2_On()

只有 led2(黃色 LED)亮起，維持 3 秒。

```
GPIO.output(led1, False)
GPIO.output(led2, True)
GPIO.output(led3, False)
sleep(3)
```

4. led3_On()

只有 led3(綠色 LED)亮起，維持 3 秒。

```
GPIO.output(led1, False)
GPIO.output(led2, False)
GPIO.output(led3, True)
sleep(3)
```

5. sent()

透過 requests.get 函式及 LogRecord_GET.php 傳送溫度、溼度及燈號資料至資料庫，並在螢幕上列印是否傳送成功。

```
r = requests.get(
    'http://{0}/LogRecord_GET.php?temp={1}&wet={2}&light={3}'
    .format(TARGET_URL, temp, humid, light))
print("Server Return Code:", r.status_code)
print(r.text)
```

6. 主函式(僅說明 try 部分的程式碼)

讀取當下的溫度與溼度資料後，讓三個 LED 都不亮，並在螢幕上列印溫度及溼度。

```
temp = dht.temperature
humid = dht.humidity
light = 0
ledOff()
print("Temp: {:.2f}*C Humidity: {:.2f}% ".format(temp, humid))
```

如果溫度介於攝氏 19 至 28 度之間且溼度小於 80%，則在螢幕上列印燈號(3 代表綠色 LED)，並傳送資料給資料庫後，讓綠色 LED 發亮。

如果溫度小於攝氏 19 度或大於攝氏 28 度且溼度小於 80%，或溫度介於攝氏 19 至 28 度之間且溼度大於等於 80%，則在螢幕上列印燈號(2 代表黃色 LED)，並傳送資料給資料庫後，讓黃色 LED 發亮。

如果溫度小於攝氏 19 度或大於攝氏 28 度且溼度大於等於 80%，則在螢幕上列印燈號(1 代表紅色 LED)，並傳送資料給資料庫後，讓紅色 LED 發亮。

```
if 19 <= temp <= 28 and humid < 80:
    light = 3
    print("Light:", light)
    sent()
    led3_On()
```

```

        elif ((temp < 19 or temp > 28) and humid < 80) or (19 <= temp <= 28 and humid >= 80):
            light = 2
            print("Light:", light)
            sent()
            led2_On()
        elif (temp < 19 or temp > 28) and humid >= 80:
            light = 1
            print("Light:", light)
            sent()
            led1_On()

```

● Client 部分

■ showChart(.php)

1. 設定網頁每 5 秒自動刷新。

```
<?php header('refresh: 5;url="http://192.168.43.160/showChart.php"') ?>
```

2. 展示資料視覺化結果，可在導航欄切換至一般表格。

```




```

■ showColumn(.php)

1. 依照 Server 端註冊的帳號密碼設定參數並建立連線。

```

<?php
    $host = "192.168.43.221";
    $dbuser = "wpuser";
    $dbpasswd = "123456";
    $DBNAME = "final_data";
    $conn = mysqli_connect($host, $dbuser, $dbpasswd, $DBNAME);
    if (!$conn) {
        echo "Error: Unable to connect to MySQL." . PHP_EOL;
        echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;
        echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;
        die ("無法連線至資料庫");
    }
    // 設定連線編碼
    mysqli_query($conn, "SET NAMES 'utf8'");
    $sql = "SELECT * FROM final_data.log ORDER BY \"time\" DESC;";
    $ret = mysqli_query($conn, $sql);
    if (!$ret){
        echo "Error: Unable to connect to MySQL." . PHP_EOL;
        echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;
        echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;
        die("error");
    }
    mysqli_close($conn);
?>

```

2. 設定網頁每 5 秒自動刷新

```
<?php header('refresh: 5;url="http://192.168.43.160/showColumn.php"') ?>
```

3. 連線遠端資料庫並展示一般表格，可在導航欄切換至資料視覺化結果。

資料庫欄位[4]為安全代號 light，將 light 代號轉換回 LED 顏色。

```

<?php

while($row = mysqli_fetch_array($ret, MYSQLI_NUM)){
    echo "<tr>";
    echo "<td>".$row[1] . "</td>\n";
    echo "<td>".$row[2] . "</td>\n";
    echo "<td>".$row[3] . "</td>\n";
}

```



```

        if ($row[4]=="1"){
            echo "<td>";
            echo "<font color=\"red\">";
            echo "●";
            echo "</font>";
            echo "</td>\n";
        }
        if ($row[4]=="2"){
            echo "<td>";
            echo "<font color=\"yellow\">";
            echo "●";
            echo "</font>";
            echo "</td>\n";
        }
        if ($row[4]=="3"){
            echo "<td>";
            echo "<font color=\"green\">";
            echo "●";
            echo "</font>";
            echo "</td>\n";
        }

        echo "</tr>";
    }
?>

```

■ Client(.py)

1. getSqlData()

建立連線並擷取遠端資料庫內容，回傳 DataFrame(僅說明 try 內的程式碼)。

變數 loadInitData 表示是否已執行第一次擷取資料，避免七段顯示器執行報錯。

```

print("----- Connect to MariaDB Platform -----")
conn = mariadb.connect(
    user=myUser,
    password=myPassword,
    host=myHost, port=myPort,
    database=myDatabase)
cur = conn.cursor()
sql = "Show tables;"
cur.execute(sql)
allTable = cur.fetchall()
sql = "use {};" .format(myDatabase)
cur.execute(sql)
sql = "SELECT * FROM {};" .format(myTable)
cur.execute(sql)
allColumns = cur.fetchall()
df = pd.DataFrame(allColumns)
conn.close()
print("----- Connect Done -----")
global loadInitData
loadInitData = True
updateNumber(df)
return df

```

2. drawChart(dataframe)

將 dataframe 內容轉為數值(欄位[1]是 time 故以 continue 跳過)。

```

for i in range(5):
    if i==1:
        continue
    dataframe[i] = dataframe[i].astype('float64')

```

依據參數 dataframe 傳入資料繪製折線圖並輸出圖檔(此僅展示其中一張的程式碼)。

```

print("----- Draw Start -----")
print("Now drawing.....")
print(" x = time, y = wet")

```

```
plt.figure(figsize=(PltWidth, PltHeight), dpi=PltDpi)
fig, ax = plt.subplots(1)
fig.autofmt_xdate()
plt.plot(timeList, wetList, '-.')
xfmt = mdates.DateFormatter('%m-%d %H:%M')
ax.xaxis.set_major_formatter(xfmt)
plt.title('humidity', fontsize=FontSize)
plt.xlabel('time', fontsize=FontSize)
plt.ylabel('%', fontsize=FontSize)
plt.savefig("/home/pi/web-jesse/images/outputWet.jpg")
```

繪製結束後將圖檔所在資料夾內容覆寫至 html 資料夾。

```
os.system("sudo cp /home/pi/web-jesse/images/* /var/www/html")
print("----- Update Done -----")
```

3. **updateNumber(dataframe)** (key&7-seg)

擷取資料表最後(最新)一筆資料。

```
global tempNumber
tempNumber = round(float(dataframe[2].tail(1)))
global wetNumber
wetNumber = round(float(dataframe[3].tail(1)))
```

4. **segNumberChange(pin_key)** (key&7-seg)

用於切換七段顯示器數值(溫度溼度)。

```
global nowTemp
nowTemp = not nowTemp
while(not loadInitData):
    print("not loadInitData ... ")
    sleep(1)
sevenSegmentControl(nowTemp)
```

5. **sevenSegmentControl(showTemp)** (key&7-seg)

讓七段顯示器顯示數值。

sevenSegmentStringDict 是紀錄了 0-9 阿拉伯數字所對應 a-g 的字典。

```
if(showTemp):
    getNumber = tempNumber
else:
    getNumber = wetNumber
number10 = floor(getNumber/10)
number = getNumber%10
for pinIndex in range(7):
    value10 = int(sevenSegmentStringDict[number10][pinIndex])
    GPIO.output(sevenSegmentPin10[pinIndex], value10)
    value = int(sevenSegmentStringDict[number][pinIndex])
    GPIO.output(sevenSegmentPin[pinIndex], value)
```

6. **main()** 主函式

詢問使用者是否要使用七段顯示器。

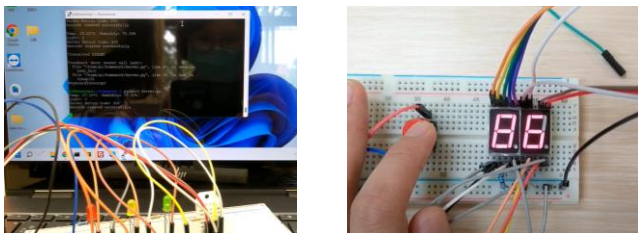
```
:0 結束程式
:1 僅擷取資料並更新網頁圖片
:2 擷取資料並更新網頁圖片、可監聽按鈕切換七段顯示器顯示溫溼度
modeCtrl()
```

根據使用者輸入執行對應函式，每 60 秒刷新一次結果(僅說明 try 內的程式碼)。

```
if(user=='2'):
    segInit()
    GPIO.add_event_detect(pin_key, GPIO.FALLING, callback=segNumberChange, bouncetime=200)
while(True):
    data = getSqlData()
    if(user=='2'):
        sevenSegmentControl(nowTemp)
    drawChart(data)
    sleep(60)
```

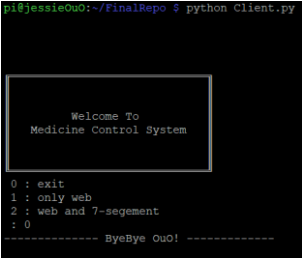
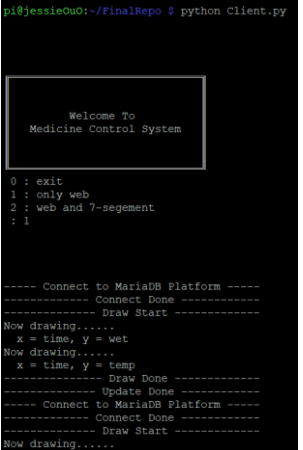

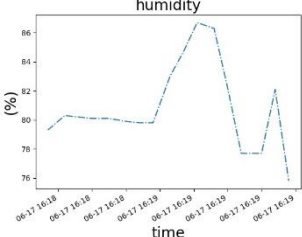
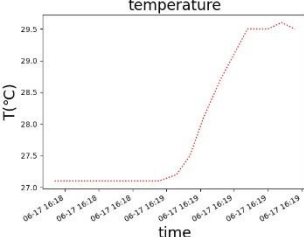
五、專題 DEMO

- DEMO 影片



Server 影片連結：<https://youtu.be/kleWXTKR9NI>
Client 影片連結：<https://youtu.be/CdhzDdG1TBA>

- Client Shell

<p>: 0</p> <p>(結束程式)</p>	<p>: 1</p> <p>(僅擷取資料並更新網頁圖片)</p>	<p>: 2</p> <p>(擷取資料並更新網頁圖片、 監聽按鈕切換七段顯示器顯示溫 溼度)</p>
<pre>pi@jessieOu0:~/FinalRepo \$ python Client.py</pre> 	<pre>pi@jessieOu0:~/FinalRepo \$ python Client.py</pre> 	<pre>pi@jessieOu0:~/FinalRepo \$ python Client.py</pre> 
<p>: 1 或 : 2 的圖片輸出</p>		
		

六、心得

- 佩瑩

經過兩次上傳資料庫的作業練習，在建立資料庫與上傳資料時沒有出現太大的問題，皆能夠順利完成，而在 Client 端要連結 Server 端時才發現沒有想像中的容易。一開始以為只需設定 Client 端就好，在和好婕一起上網搜尋資料後，得知 Server 端與 Client 端都需要進行安裝套件與多個設定步驟(防火牆)，還需要登入資料庫中使用 SQL 語法進行用戶創建的設定。使用這些不熟悉的指令時感覺有些複雜，也經歷了幾次連結失敗後再上網尋找解決方法的狀況，不過最後還是成功地讓 Client 端連上資料庫了。

完成了期末專題後，除了將上課時學到的素材加以運用之外，亦學到了許多課堂上沒有提到的內容，感覺十分充實，也很開心修了這門課，能夠學到有關樹莓派與各種電子材料的使用方法及應用。

- 好婕

這次卡關最久的就是資料庫連線問題，php 只要知道帳號密碼就可以了，但 python 如果要連線就必須先由管理員註冊帳號給予權限(說是保護機制，不能直接用 root 登入)；再來就是網頁不能顯示圖片，因為多年前也有不少人在討論 nginx 伺服器網頁的標籤錯誤，我們起初也認為應該是一樣的原因，改設定檔、重新執行、改變路徑都嘗試了一番卻都沒有成效，最後才發現是 html 的相對圖片路徑與 web-jesse 的相對圖片路徑不同，總之兩個問題最後都一起解決了，可喜可賀可口可樂 `u`)

七、參考資料

- [MariaDB/MySQL 常用指令操作與語法範例](#) / 魏子靖(2017)
- [How to enable Remote access to ……](#) / Webdock(2022)
- [How to connect Python programs to MariaDB](#) / MariaDB(2020)