

# 嵌入式實驗二報告

## Input Output 練習

組員：電機三 林尚謙

電機三 楊耀程

指導教授：鄭振牟

繳交日期：2017/05/05

## 主題介紹：Smart furniture(智慧家電)

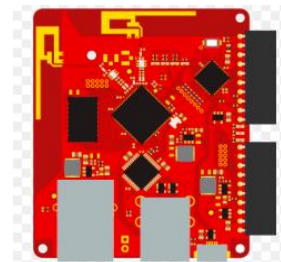
1. 具有接收環境資訊的功能，可讀取環境溫濕度(input)
2. 具有監控的功能，可傳送即時影像(input)
3. 接受到的資訊能在 web 上顯現
4. 在 web 上有 message 的功能，使用者群組可互相聊天
5. 能從 web 上的開關控制實體 led 的開與關(output)

## Smart Furniture 架構概述



Client : family (web)  $\longleftrightarrow$  Server : furniture (Tessel)

整體架構可分為 client 端、server 端，client 端即為瀏覽器上的網頁(可由家庭的使用者來打開)，server 端即為 Tessel 2 所執行的 javascript 程式。Client 會先與 server 建立 socket 連線，在互動發生後(送出訊息、點擊按鈕等)，會由 client 端向 server 發出請求，再由 server 做出對應的回應，而 database 則負責提供 server 存取資料。我們的 client 端與 server 端分別由一個檔案(index.html 與 server.js)構成。



## 操作步驟

1. 確保 server(Tessel) 和 client 連接到的是同一個 wifi 網域
2. 先輸入 t2 wifi 找到他們連接到的 wifi IP (本範例為:192.168.43.105)

```
PS D:\Users\derek\Downloads\lab2-master\lab2-master> t2 wifi
INFO Looking for your Tessel...
INFO Connected to derek.
INFO Connected to "derek"
INFO IP Address: 192.168.43.105
INFO Signal Strength: (9/70)
INFO Bitrate: 72mbps
```

3. 輸入 t2 run server.js 來執行程式
4. 在 client(browser)輸入 wifiIP:3000 本範例為: 192.168.43.105:3000
5. 完成

## 使用到的輔助工具

1. socket io
2. climate-si7020(溫溼度感測器)
3. USB camera
4. Tessel 2 板子上的 led

## 特色與功能說明

### Message 功能

作為智慧家電，同個家庭使用者之間有聊天溝通的功能，能夠讓家電的使用更方便，使用者可以討論家電要不要開，要開幾度。

### 實作架構

延續實驗一所做的聊天室，以 socket io 作為溝通的主要基礎，能夠多人聊天並即時反應(與實驗一報告重複之部分在此省略)。

client	server
<pre>\$('#textbox').submit(function() {   if (\$('#m').val() != '') {     socket.emit('chat message', \$('#m').val());     \$('#m').val(''); // clean input   }   return false; });</pre>	<pre>socket.on('chat message', function(msg) { //receive msg from client   console.log(socket.username + ":" + msg);   var mytime = mygetTime();   io.emit('chat message', {     u_name: socket.username,     u_word: msg,     u_time: mytime   }); //when 'chat message'   records.insert({ u_name: socket.username, u_word: msg, u_time: mytime }); });</pre>

## 實際操作畫面



Input 端練習(physical world→Tessel→web)

一、溫溼度→climate sensor→Tessel→web

作為智慧家電，能夠及時測定周圍環境的溫度濕度，並呈現在網頁上讓使用者一目瞭然，每三十秒更新一次。

## 實作架構

1. 以 socket io 引發事件，每隔三十秒，client 端會發出一個要求 climate 資訊的請求

```
setInterval(function() {  
    climateon();  
}, 30000); //update every 30 sec
```

```
function climateon() {  
    socket.emit("climate on", {});  
};
```

2. server 端(Tessel)收到請求後，就會透過 climate sensor 讀取當下的溫度跟濕度，並回傳給 client

```
socket.on('climate on', function() {  
    myclimate();  
    console.log('climate on');  
});
```

```
function myclimate() {  
    climate.readTemperature('f', function(err, temp) {  
        climate.readHumidity(function(err, humid) {  
            socket.emit('update climate',{  
                Degree: temp.toFixed(1),  
                Humidity: humid.toFixed(1)  
            });  
            console.log('Degrees:', temp.toFixed(4) + 'F', 'Humidity:', humid.toFixed(4) + '%RH');  
        });  
    });  
};
```

3. client 端收到溫度濕度後，在網頁上呈現。

```
socket.on('update climate', function(data) {  
    var temp = document.getElementById('Temperature');  
    var humid = document.getElementById('Humidity');  
    var degree = ((data.Degree-32)*(5/9)).toFixed(1);  
    var rh = data.Humidity;  
    temp.innerHTML = "溫度： " + degree + "度C";  
    humid.innerHTML = "濕度： " + rh + "%RH";  
});
```

## 二、影像→USB camera→Tessel→web

能夠透過 USB 相機，以連續的 frame 轉播家中的情況，並在網頁上呈現，達到監控的效果。可以由網頁上的開關決定網頁上是否要顯示監控到的影像。

## 實作架構

1. server 端不斷發送影像的串流出去

```
app.get('/stream', (request, response) => {  
    response.redirect(camera.url);  
});  
  
app.use(express.static(__dirname));
```

2. 由 client 端 html 檔接收，並顯現在 web 上

```
<div id="camblock">  
      
</div>
```

3. 由 client 端 web 上的 camera on/off 按鈕，選擇是否要讓影像的畫面在 web 上顯現

```
$('#cameraswitch').click(function() {  
    var cs = document.getElementById('cameraswitch');  
    if (cs.checked) {  
        var camblock = document.getElementById("camblock");  
        var msgblock = document.getElementById("message_block");  
        camblock.innerHTML = '';  
        camblock.style.display = 'inline-block';  
        msgblock.style.display = 'inline-block';  
    }  
});
```

## Output 端練習(Tessel→physical world)

### 一、web→Tessel→led (tessel2 的 led[3])

能夠透過 web 上 led on/off 的按鈕，來操控 Tessel 板子上 led 的亮與暗，web 上開關的狀態會隨著 Tessel 本身 led 的亮暗作相對應的更新，不會發生現在燈是亮的，網頁上的開關卻是 off，這樣奇怪的狀況。此 led 可以象徵家電的 on/off，只要在同一個內網上，使用者可藉由手機或電腦的 browser，來操控電器的開關。

## 實作架構

1. 當 client 端開啟網頁，會執行 readLed 的函式，跟 server 請求讀現在 led 值的狀況。

```
function readLed() {  
    socket.emit("read led");  
};
```

2. Server 讀取 led 的值並回傳，client 網頁上的開關也會作出相對應的變化。

(不會發生燈是亮的，網頁上的開關卻是 off 的狀況)

```
socket.on('read led', function() {  
    socket.emit('set led',{  
        isOn: tessel.led[3].isOn  
    });  
});
```

```
socket.on('set led', function(data) {  
    var ledswitch = document.getElementById("ledswitch");  
    console.log(data.isOn);  
    ledswitch.checked = data.isOn;  
});
```

3. 當 client 端的開關被打開或關掉，會通知 server(Tessel)，使 server 的 led 做出相對應的變化(on/off)

```
$('#ledswitch').click(function() {  
    socket.emit("led switch");  
});
```

```
socket.on('led on', function() {  
    tessel.led[3].on();  
    console.log('led 3 on');  
});  
  
socket.on('led off', function() {  
    tessel.led[3].off();  
    console.log('led 3 off');  
});
```