

嵌入式系統實驗 TESSEL 2

第十組 許鎔巘 林彥廷

說明

我們這次練習的內容主要涵蓋了 Tessel 2 input/output 方向的操作，因為沒有 demo 展示，主要會以電腦截圖穿插在內文中加以解釋。

實驗工具

Tessel 2 x1

Logitech cam x1

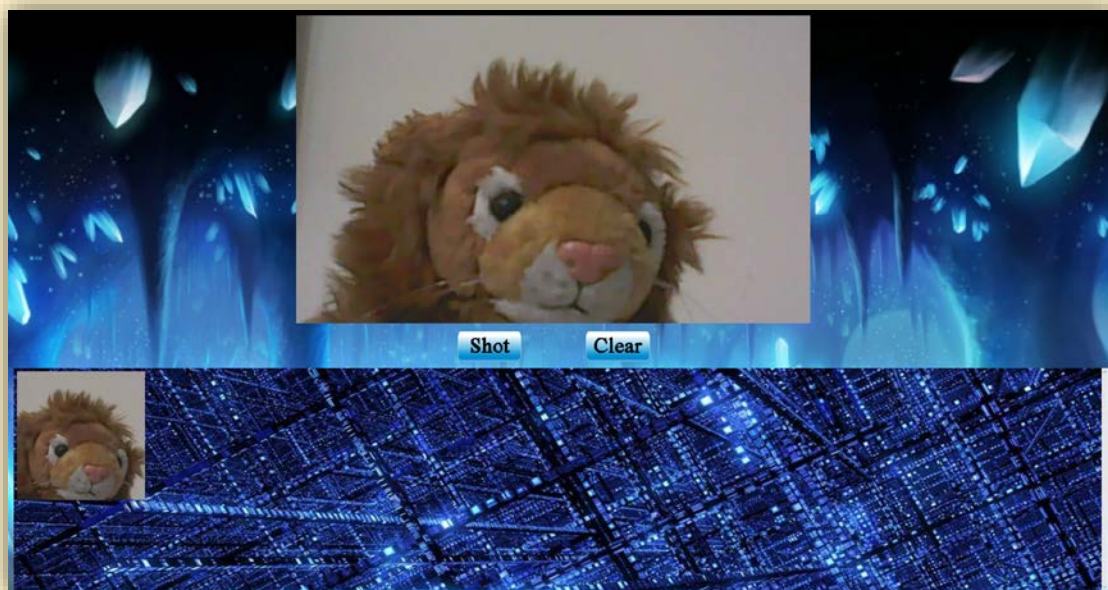
input 方向練習

我們將 cam 連接在 Tessel 2 上，然後將 Tessel 2 連上一般的 wifi，當連線完成後，Tessel 2 會自己開設一個區域無線網路，並會把存放在 Tessel 2 裡的內容架設在它開的 port 上，藉由在 port 上寫一個網頁，將攝影捕捉到的畫面放置在這個網頁上，實踐 physical→Tessel 2→web 這個方向的練習

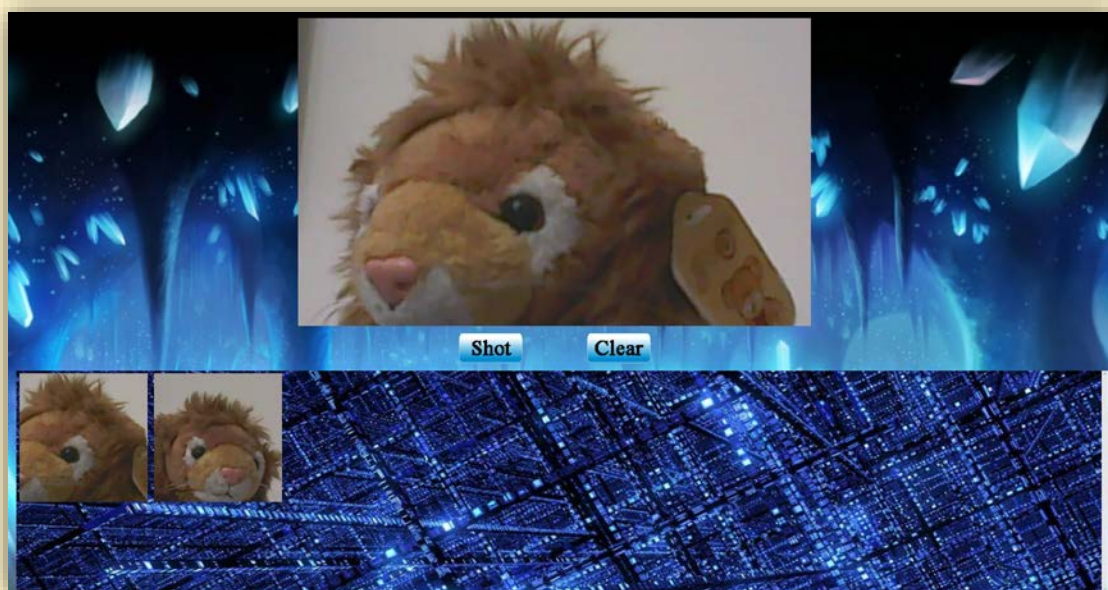


output 方向練習

當頁面順利出現後，我們在裡面寫了一個 shot 與 clear 的按鈕，按下 shot 可以拍下瞬間的照片，按下 clear 可以清除 local memory 網路暫存的圖片，我們把拍下來圖片放在攝影畫面的下方。



▼ 當拍攝的照片超過一張時，最即時拍攝的照片會安排在左下角第一張。



▼ 當我們點擊下方的圖片時，欲觀賞的圖片會放大。



點選照片可以下載圖片，並以 jpeg 的格式儲存，另外，按下紅色叉叉會跳回上一頁面，按下藍色垃圾桶則是將此照片刪除。

input 架構 physical→Tessel 2→web

```
12 http.listen(port, function(){
13   tessell.led[2].on();
14   setInterval(function() { tessell.led[2].toggle(); }, 500);
15   console.log("Server running at http://192.168.1.101:" + port);
16 });
17
18 app.get("/", function(req, res){
19   res.sendFile(__dirname + "/index.html");
20 });
```

- ▲ 12 行至 20 行是用來架設網站，其中 13、14 行裡的 led[2]是希望在開機完後可以讓這顆 led 燈持續閃爍，表示已經完成連線，並可以透過開啟 Tessel 2 開設的無線網路來連接我們指定的 port 當網頁開起順利後，cam 會將捕捉的畫面持續投射在 <http://192.168.1.101:8000> 中，運作的原理是 cam 透過 Tessel 2 回傳一個 camera.url 給 web，讓 web 接到後可以顯示畫面。

output 架構 Tessel 2→physical

```
43     var shotBtn      = document.createElement("img");
44     shotBtn.src       = "/image/btn_shot/png";
45     shotBtn.style.marginRight = "50px";
46     shotBtn.onclick   = function() { socket.emit("shot"); };
47     var clearBtn     = document.createElement("img");
48     clearBtn.src      = "/image/btn_clear/png";
49     clearBtn.style.marginLeft = "50px";
```

```
115 socket.on("shot", function(photoData) {
116     (function() {
117         var photoImg = new Image();
118         photoImg.src = "data:image/jpeg;base64," + photoData;
119
120         var canvas    = document.createElement("canvas");
121         canvas.id      = ("000" + imageID++).slice(-4);
122         canvas.style.margin = "5px";
123         var ctx        = canvas.getContext("2d");
124         ctx.canvas.width  = displaySize;
125         ctx.canvas.height = displaySize;
126         ctx.drawImage(photoImg, (displaySize-cameraWidth*displaySize/cameraHeight)*0.5, 0,
127             cameraWidth*displaySize/cameraHeight, displaySize);
```

- ▲ 當我們按下 shot 按鈕時，網頁的 socket 會回傳指令給 Tessel 2，並開啟一個空白的 canvas 帆布，將 shot 擷取的瞬間畫面存放在 canvas 上。

```
129     canvas.onclick = function() {
130         var largeCanvas = document.createElement("canvas");
131         var ctx          = largeCanvas.getContext("2d");
132         ctx.canvas.width = cameraWidth;
133         ctx.canvas.height = cameraHeight;
134         ctx.drawImage(photoImg, 0, 0);
135         photoLargeDiv.appendChild(largeCanvas);
136         backDiv.style.display = "block";
137         photoLargeDiv.style.display = "block";
138         closeBtn.style.display = "block";
139         deleteBtn.style.display = "block";
```

- ▲ 當我們點擊存放的圖片時，網頁會執行下一行 var largeCanvas 產生一個放大的圖片顯示，此時背景的畫面顯示亮度會暗掉，讓使用者可以完整地觀看照片而不受攝影畫面的影響。

```

141     largeCanvas.onclick = function() {
142         var downloadLink = document.createElement("a");
143         downloadLink.href = largeCanvas.toDataURL("image/jpeg");
144         downloadLink.download = "IMG" + canvas.id + ".jpg";
145         document.body.appendChild(downloadLink);
146         downloadLink.click();
147     }

```

- ▲ 當我們點擊放大的圖片時，網頁會自動下載圖片存成 jpeg 檔，手機板與電腦版都可以支持此動作，電腦版需下載最新版的瀏覽器才可以完成下載的動作。

```

149     backDiv.onclick = closeBtn.onclick = function() {
150         photoLargeDiv.style.display = "none";
151         backDiv.style.display = "none";
152         closeBtn.style.display = "none";
153         deleteBtn.style.display = "none";
154         photoLargeDiv.removeChild(photoLargeDiv.firstChild);
155     }

```

- ▲ 當我們點擊大圖片的叉叉時，原本 block 的動作全部改回執行 none，模擬讓使用者介面回復到上一個攝影時介面的動作。

```

157     deleteBtn.onclick = function () {
158         photoDisplayDiv.removeChild(canvas);
159         backDiv.onclick();
160     }
161 }

```

- ▲ 按下垃圾桶則是將此照片從 local memory 裡刪除。

```

50     clearBtn.onclick = function() {
51         while(photoDisplayDiv.firstChild)
52             photoDisplayDiv.removeChild(photoDisplayDiv.firstChild);
53         imageID = 1;
54     };

```

- ▲ 當我們按下 clear 按鈕時，photoDisplayDiv 會刪除所有存放在 local memory 的圖片，可以一次性刪除。