

# 嵌入式系統實驗 實驗二：Tessel 2

B02901083 施承佑 B02901083 陳冠宇

## 一、 INPUT 方向練習：

### 1. 使用模組：

- (1) Tessel 2 主控板
- (2) accel-mma84 module
- (3) socket.io

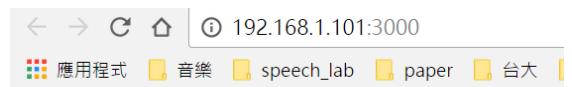
### 2. 功能：

- (1) 將 accelerometer 所觀測到的 X、Y、Z 三個方向的加速度，輸出到 Tessel 2 主控板所架設的網站上。

### 3. 架構：

- (1) 利用 Tessel 2 主控板建立一個 server，當 accelerometer 回傳資料時，利用 socket.io 傳輸到 client 端的網站上。

```
1 var tessel = require('tessel');
2 var accel = require('accel-mma84').use(tessel.port['A']);
3 var app = require('express')();
4 var http = require('http').Server(app);
5 var io = require('socket.io')(http);
6 var port = process.env.PORT || 3000;
7
8 app.get('/', function(req, res){
9   res.sendFile(__dirname + '/index.html');
10 });
11
12 var data = new Object();
13 accel.on('ready', function () {
14   accel.on('data', function (xyz) {
15     data.x = xyz[0].toFixed(2);
16     data.y = xyz[1].toFixed(2);
17     data.z = xyz[2].toFixed(2);
18     io.emit('accel', data);
19   });
20 });
21
22 accel.on('error', function(err){
23   console.log('Error:', err);
24 });
25
26 http.listen(port, function(){
27   console.log('listening on *:' + port);
28 });
29
```



## Accelerometer on Tessel 2

- x-direction: 0.05
- y-direction: 0.05
- z-direction: 1.01



## 二、 OUTPUT 方向練習：

### 1. 使用模組：

- (1) Tessel 2 主控板
- (2) Servo-pca9685 module
- (3) Socket.io

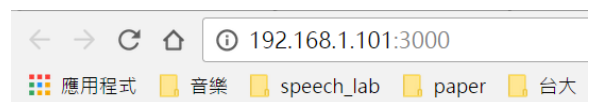
### 2. 功能：

- (1) 在 Tessel 2 主控板架設的網站上，輸入數值來改變 servo 馬達的位置。

### 3. 架構：

- (1) 在 Tessel 2 主控板建立一個 server，利用 socket.io 來與 client 端溝通。當 client 端傳送了一個更改位置的 request 時，調整 servo 馬達到更改的位置上。

```
1 var tessel = require('tessel');
2 var servolib = require('servo-pca9685');
3
4 var servo = servolib.use(tessel.port['A']);
5
6 var servo1 = 1;
7 var position = 0;
8
9 var app = require('express')();
10 var http = require('http').Server(app);
11 var io = require('socket.io')(http);
12 var port = process.env.PORT || 3000;
13
14 app.get('/', function(req, res){
15   res.sendFile(__dirname + '/index.html');
16 });
17
18 http.listen(port, function(){
19   console.log('listening on *:' + port);
20 });
21
22 io.on('connection', function(socket){
23   socket.on('position', function(pos){
24     servo.configure(servo1, 0.05, 0.12, function () {
25       servo.move(servo1, pos);
26     });
27   });
28 });
```



## Servo Controller

