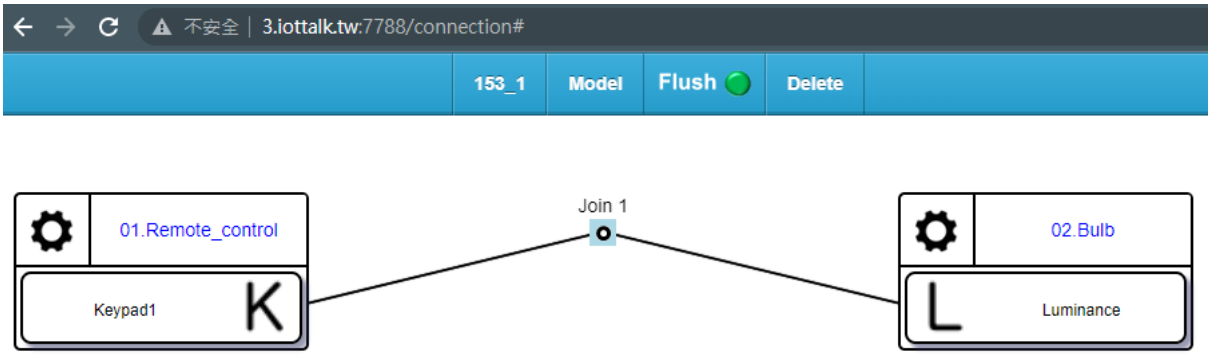
Part1:

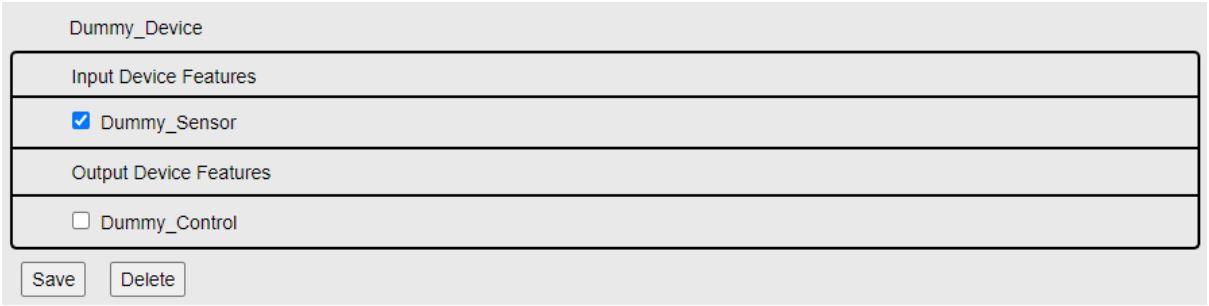
1. 架構圖



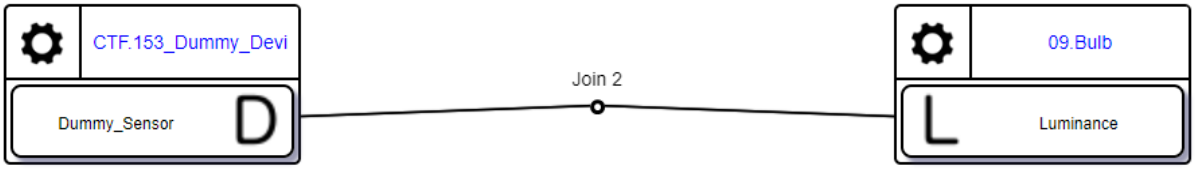
2. 測試 auto scale
    a. 先按 0, 再按 2, 再按 1, 再按 0, 再按 8, 再按 2
    b. 再來依序按 1, 2, 3, 5, 7, 8, 9 之後自己隨便按按測試
3. Demo Video

    https://youtu.be/1z6n2-ttxVo

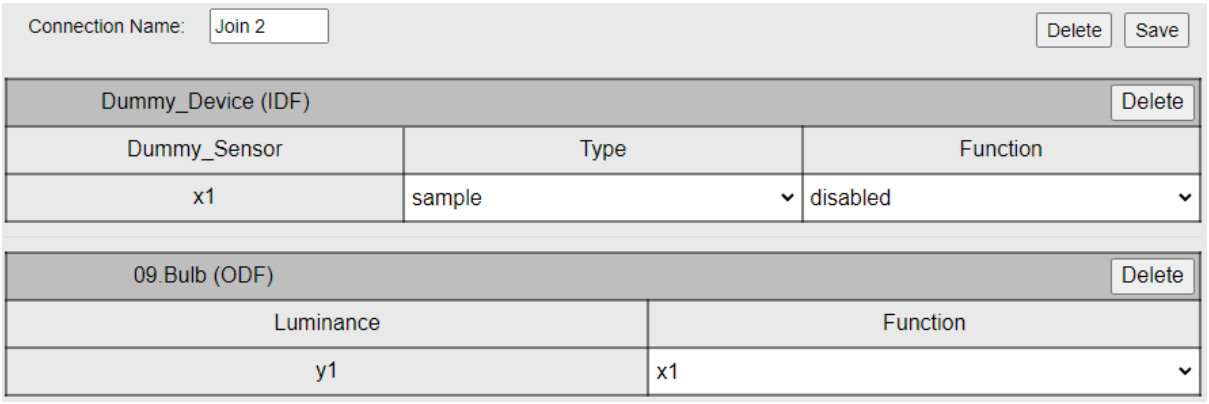
4. 加入 Dummy Device



5. 架構圖



6. 都調成 Disable



7. 修改 DAI2.py 程式碼

   將 d_name 改成自己的名字 CTFang

```
Reg_addr = mac_addr   # Note that the mac_addr generated in DAN.py always be the same cause using UUID !
DAN.profile['dm_name'] = 'Dummy_Device'   # you can change this but should also add the DM in server
DAN.profile['df_list'] = ['Dummy_Sensor', 'Dummy_Control']   # Check IoTtalk to see what IDF/ODF the DM has
DAN.profile['d_name']  = "CTF." + str(random.randint(100,999 )) + "_" + DAN.profile['dm_name'] # None
DAN.device_registration_with_retry(ServerURL, Reg_addr)
```

8. Demo

   https://youtu.be/lBA10oOMJfs

9. 心得

   這次實驗讓我更了解 auto scale，也嘗試到了用 python code 去控制 IoTtalk 上的 Device，體驗到這個平台的厲害。

Part2:

1. 下載 Buld 程式碼 https://github.com/IoTtalk/Bulb
2. 修改 js/ida.js 檔案，讓他連到正確的 Server

```
$(function () {
    csmapi.set_endpoint ('http://3.iottalk.tw:9999');
    var profile = {
        'dm_name': 'Bulb',
        'idf_list': [],
        'odf_list': [Luminance, Color_0],
    }
```

3. 修改 js/ida.js 的 RGB Function 值，讓他為不一樣的顏色

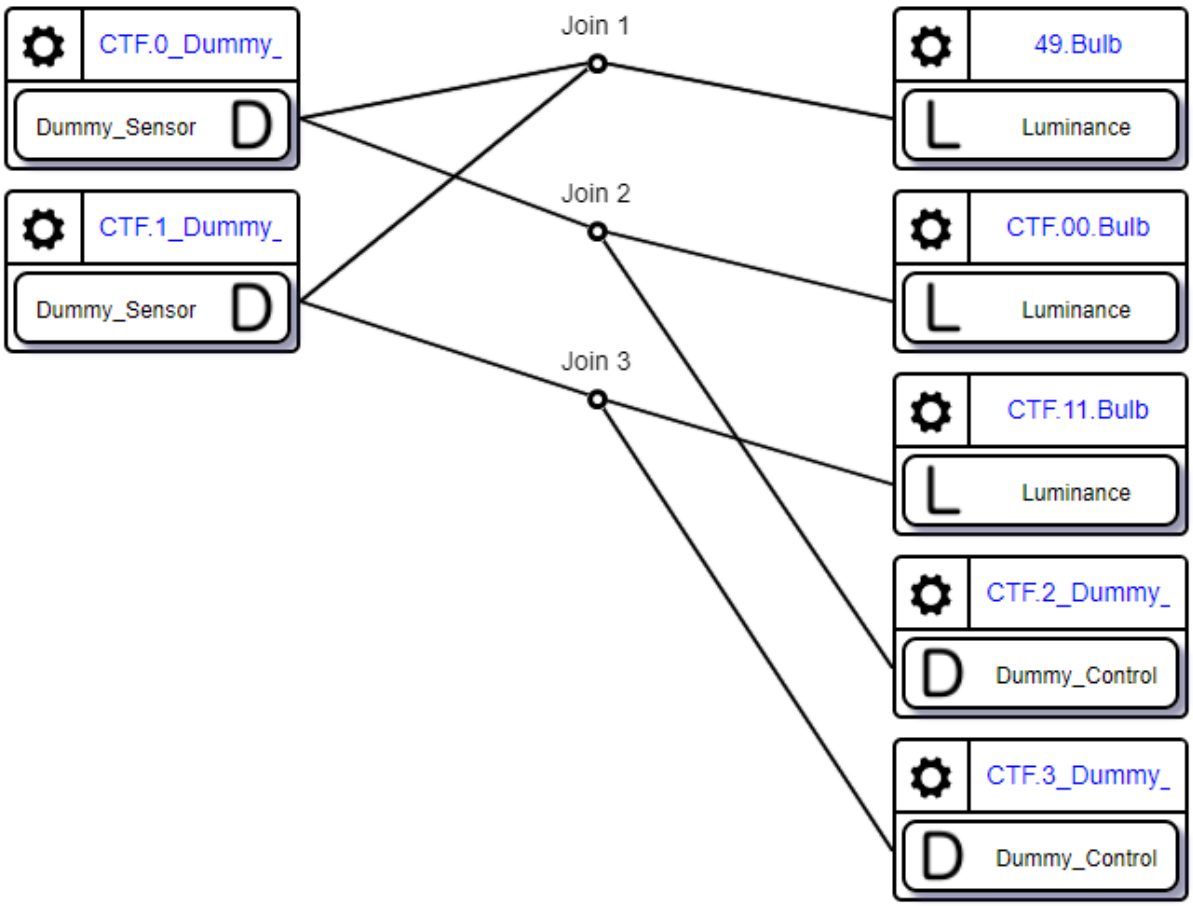
```
var r = 0;
var g = 255;
var b = 255;
var lum = 100;
```

```
var r = 255 ;
var g = 0;
var b = 255;
var lum = 100;
```

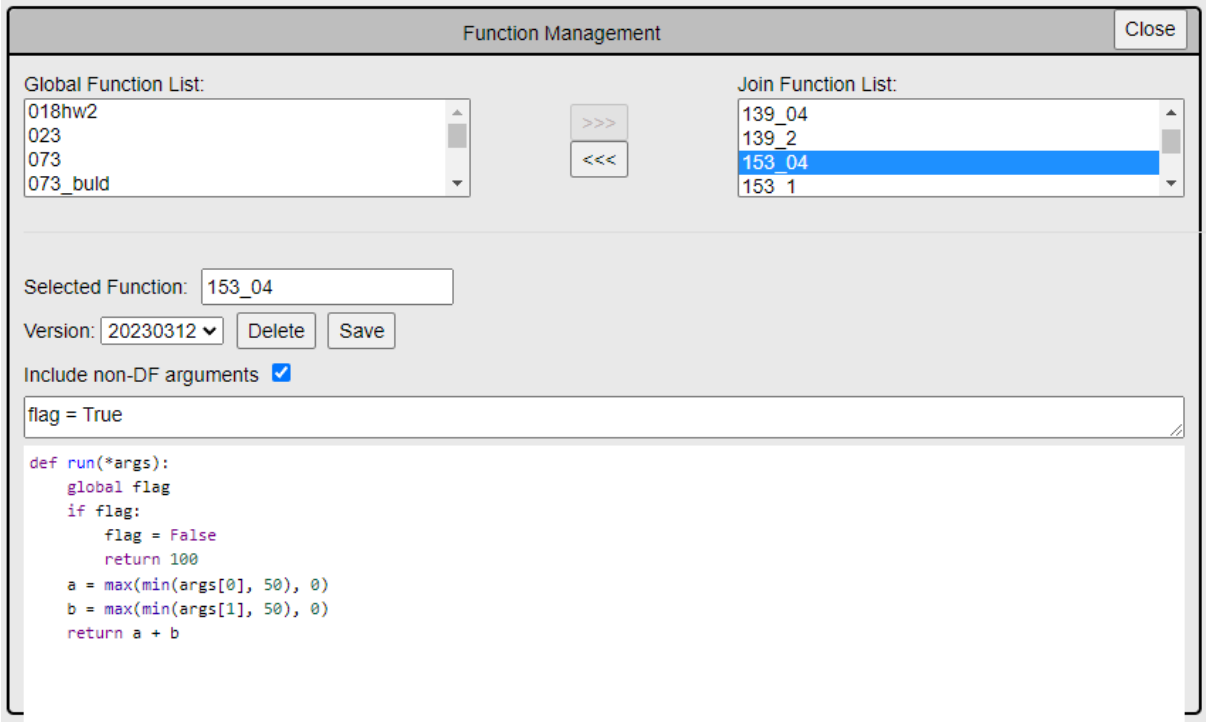
4. 修改 js/dan.js 裡面的名字，為燈泡的名字

```
function register (endpoint, profile, callback) {
    profile['d_name'] = 'CTF.00.' + profile['dm_name'];
    _profile = profile;
    csmapi.set_endpoint(endpoint);
```

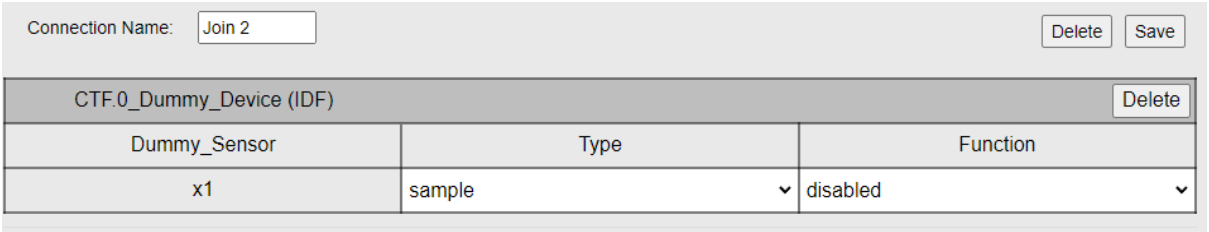
5. 架構圖



6. Join Function，並使用全域變數 flag



7. Join2 & Join3 都設為 Disable



8. Demo

   https://youtu.be/m04ZmNBoXMs

9. 心得

   這次作業明顯比上兩次複雜許多，不僅是要花很多時間才能理解要做甚麼，也要花一些時間才能找到自己設定上的問題，但透過這次，可以自訂義裝置透過此平台連結，這個平台的發展性又更大了!!

   (因為截圖不好看出變化，所以以影片代替規定數量的截圖!)