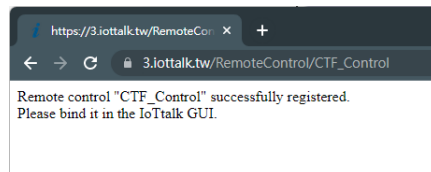


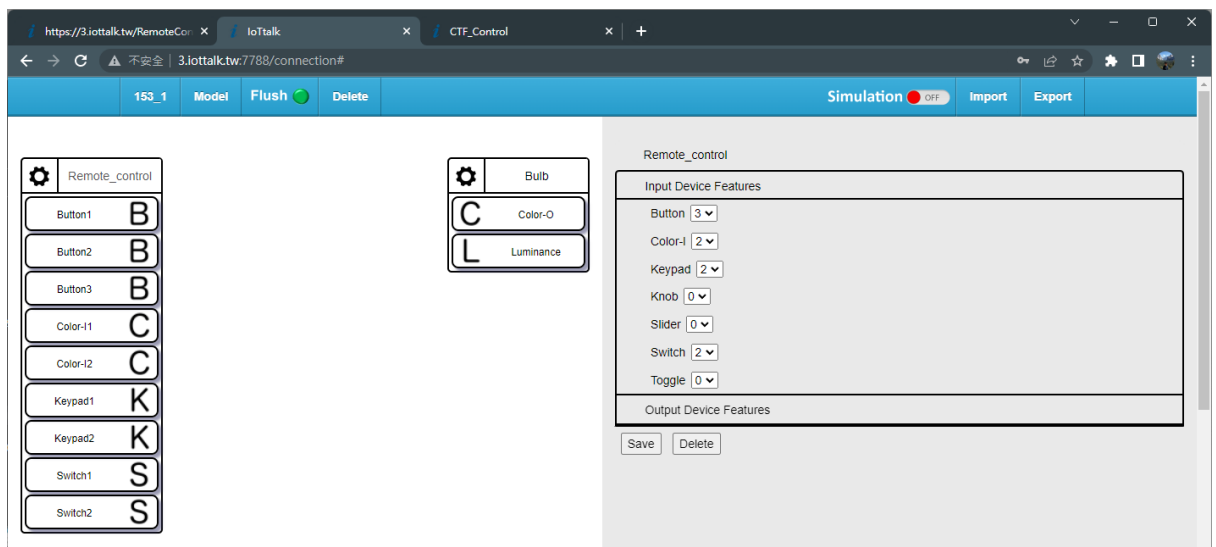
Part1 測試與實驗

1. 透過 3.iottalk.tw/RemoteControl/<device name> 建立自己個 Remote Control
我的名字陳琮方，故取作為 **CTF_Control**



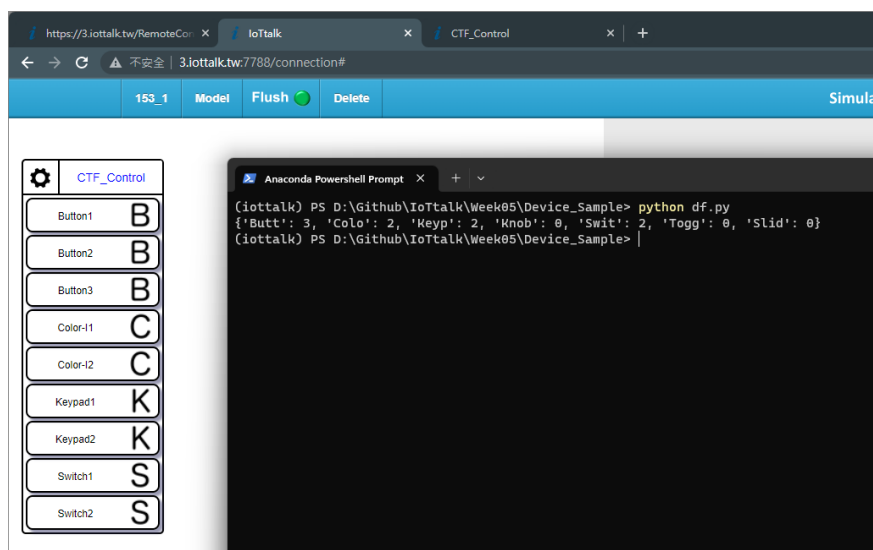
2. 建立專案，並新增 Remote Control & Bulb

Input Device Feature 設為幾個，我們的目標網頁就希望會呈現多少個控制，所以我這邊先設為 3, 2, 2, 2

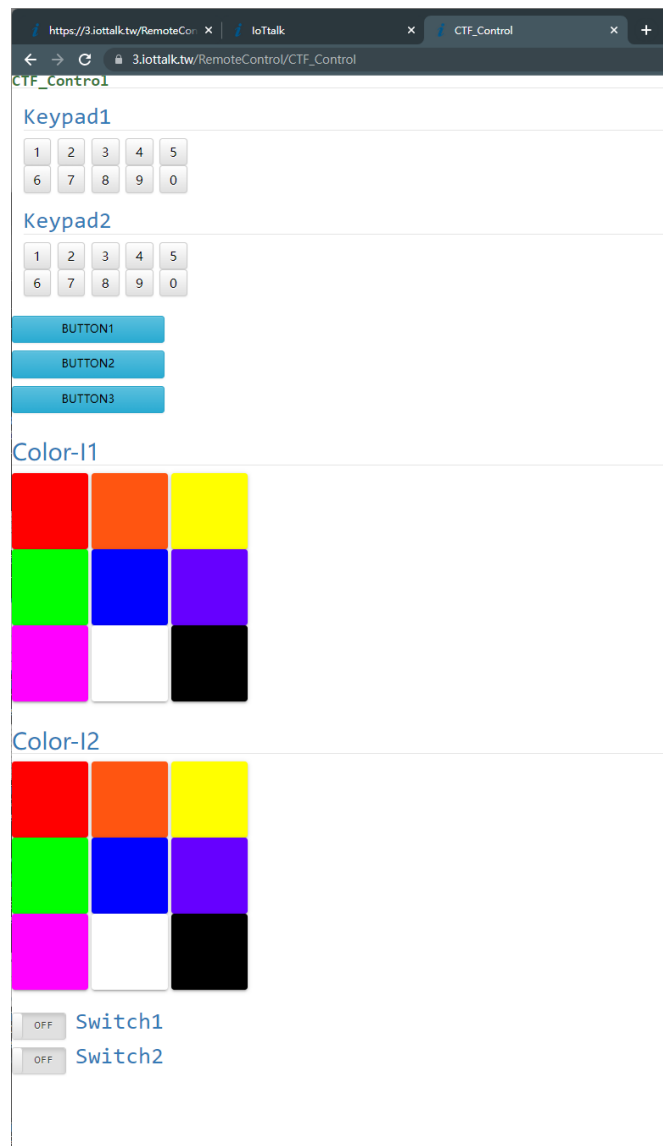


3. 將剛剛的 **CTF_Control** 加入設定，測試看看現在的狀況

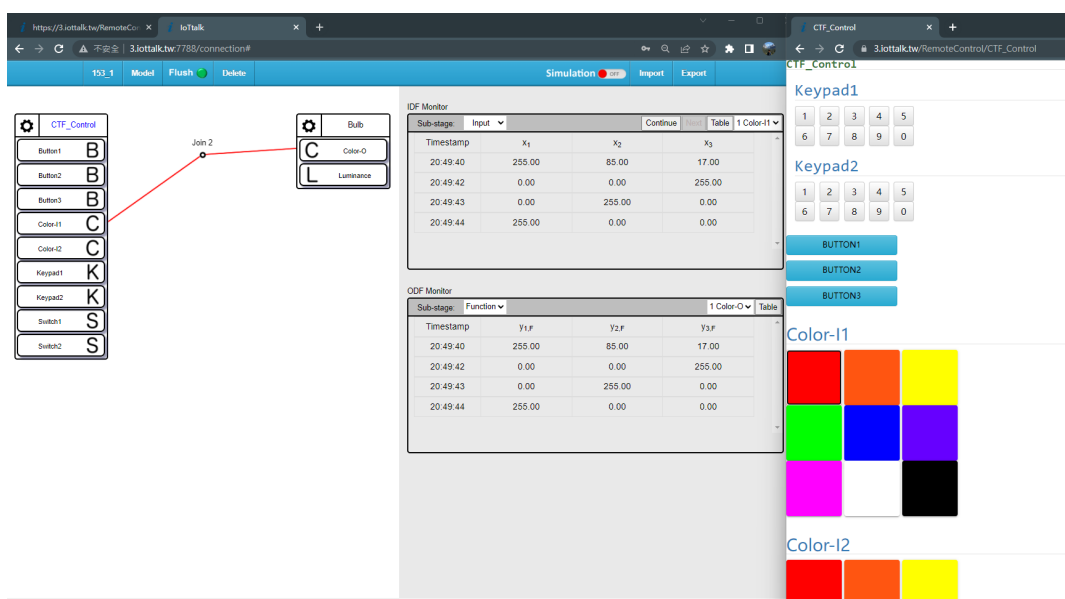
- 更改並測試 df.py 程式碼：



- 連接 https://3.iottalk.tw/RemoteControl/CTF_Control



- 增加 Join，功能正常



Part2 Flask

4. 編寫 Flask 的網頁路徑進入點 (於 app.py)

```
@app.route('/RemoteControl/<device_id>/', methods=['GET'])
def remote_control(device_id):
```

並改成自己寫的 *remote_control.html*, 並將剛剛 *df.py* 的程式碼拿來用, 將各開關的數量傳給 flask 中的 *render_template*, 產生對應的數量

```
Ctl_0 = csmapi.pull(device_id, '__Ctl_0__')
if Ctl_0 != []:
    selected_df_flags = Ctl_0[0][1][1]['cmd_params'][0]
    df_list = profile['df_list']
    df_dict = {'Butt': 0, 'Colo': 0, 'Keyp': 0, 'Knob': 0, 'Swit': 0, 'Togg':0, 'Slid':0}
    for index, element in list(enumerate(selected_df_flags)):
        if element == '1':
            df_dict[df_list[index][:4]] += 1

print('Device: ', df_dict)

return make_response(render_template('remote_control.html', device_id=device_id, df_dict=df_dict))
```

5. 編寫 remote_control.html (switch.html 改的) 新增 Color-I, Keypad, Button

```
{% if df_dict['Butt'] != 0 %}
<legend class="text-primary h2">Button</legend>
{% for i in range(df_dict['Butt']) %}
    <button onclick="csmapi.push('{{device_id}}', '{{i+1}}', '{{i+1}}'" style="width:200px;" >Button{{i+1}}</button>
    <br>
{% endfor %}
<br>
{% endif %}
```

這裡要注意的點是上課說過的, 編號是 1 開始, 不是 0, 所以要 **$\{i + 1\}$**

截圖只放 Button, 其他以此類推

6. Demo Video

<https://youtu.be/EdPfUhHCMhM>

7. 心得

這次比之前都要再複雜許多, 也將 flask 練習了一遍, 更加清楚 IoTalk 平台的能力, 也對 html, flask, 以及 csmapi 有更多的理解。