Using dummy devices to create a creative application.

作業要求

(1) You have to use the dummy device.

(2) You have to upload the Python files.

(3) Upload the related screenshots, for example, the screenshot of the IoTtalk page.

(4) Short description of your project.

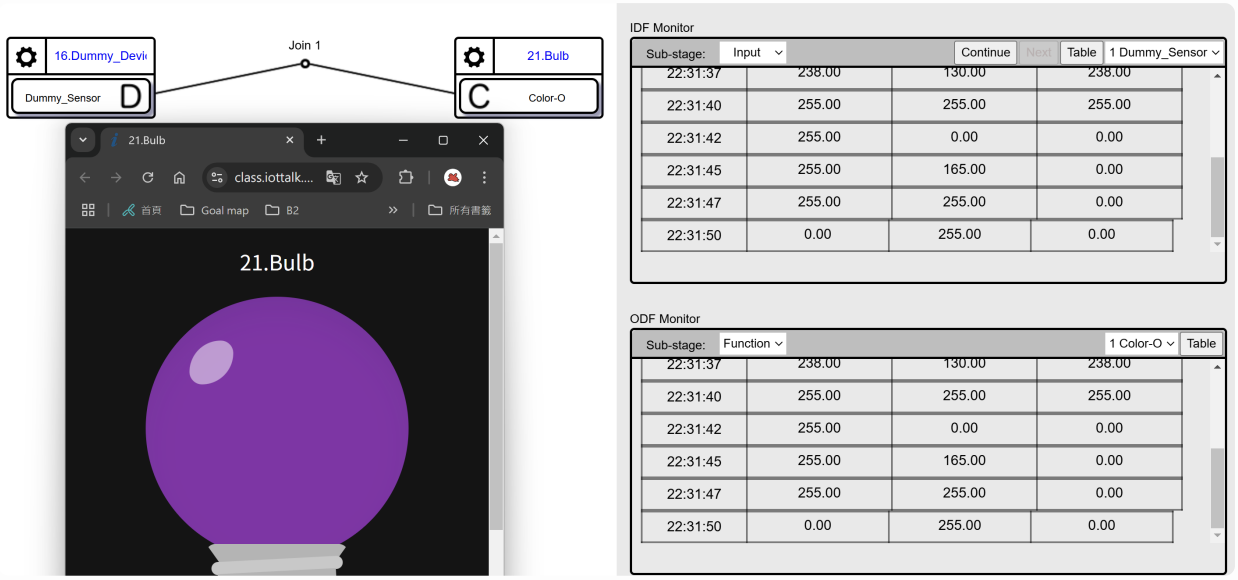
(5) Creativity will account for part of the score.

作業說明

> **利用虛擬燈泡模擬廣告燈顏色紅、橙、黃、綠、藍、靛、紫、白**

1. 定義顏色清單：使用 colors 列表，儲存紅、橙、黃、綠、藍、靛、紫、白的 RGB 值。
2. 計數器 color_index：用來指向目前要回傳的顏色。每次執行 `Dummy_Sensor()` 後，計數器加一，並使用 % 運算符來循環選擇顏色。
3. 延遲1.5秒：在每次回傳顏色前，執行 `time.sleep(2)` ，讓程式每隔1.5秒回傳一個新的顏色。

iot page



程式碼

```
import random
import pyautogui
from PIL import ImageGrab
import time

# 定義顏色的 RGB 值
colors = [
    (255, 0, 0),       # 紅
    (255, 165, 0),     # 橙
    (255, 255, 0),     # 黃
    (0, 255, 0),       # 綠
    (0, 0, 255),       # 藍
    (75, 0, 130),      # 靛
    (238, 130, 238),   # 紫
    (255, 255, 255)    # 白
]

ServerURL = 'https://class.iottalk.tw' #For example: 'https://DomainName'
MQTT_broker = 'iot.iottalk.tw' # MQTT Broker address, for example: 'DomainName'
or None = no MQTT support
MQTT_port = 8883
MQTT_encryption = True
MQTT_User = 'iottalk'
MQTT_PW = 'iottalk2023'
device_model = 'Dummy_Device'
IDF_list = ['Dummy_Sensor']
ODF_list = ['Dummy_Control']
device_id = None #if None, device_id = MAC address
device_name = None
exec_interval = 1 # IDF/ODF interval

# 初始化計數器
color_index = 0

def Dummy_Sensor():
    global color_index
```

```
    # 取得當前顏色
    r, g, b = colors[color_index]
    print(f"Returning Color: (R: {r}, G: {g}, B: {b})")

    # 更新 index 並模 8 以輪流選擇顏色
    color_index = (color_index + 1) % len(colors)

    # 每隔兩秒返回新的顏色
    time.sleep(1.5)
    return r, g, b

def Dummy_Control(data:list):
    print(data[0])

def on_register(r):
    print(f'Device name: {r["d_name"]}')
```