

執行環境：

建議 Python3.8 以上並安裝 websockets 套件

→ pip install websockets

檔案說明:

robot_ws_client.py	主要 Library 檔案，使用 WebSocket 與 Robot 通訊 功能： 1. 發送指令讓 Robot 移動/停止 2. 接收 Robot 狀態，根據狀態可以處理自訂的行為
ws_client_sample.py	使用 robot_ws_client.py 的範例，互動式介面
ws_client_sample2.py	使用 robot_ws_client.py 的範例，測試巡迴移動停靠 8 個站點，每站停留 3 秒，重複 5 次

robot_ws_client.py 說明:

def connect(self, ip="localhost", port=8000)	連線到指定 IP 與 Port 的 WebSocket 伺服器，並啟動訊息監聽。 * 必須確保 Robot 與執行 python 的電腦在同一個 Wifi 網段下，Robot 的 IP 可以從工程模式查看 * 工程模式進入法: 設定 → 關於 → 製造商名稱 → 左側會出現工程模式的 Tab
def disconnect(self)	中斷 WebSocket 連線
def send_delivery_command(self, table_name)	讓 Robot 前往指定桌號 (可設定多個餐桌用 "," 分隔, ex: "A01,A02,A03")
send_continue(self)	讓 Robot 繼續動作，當 robot 抵達餐桌時，執行 continue 可讓 robot 前往下一桌 (需設定多餐桌)
def send_return(self)	讓 Robot 返回出餐點 (起始開機點)
def send_stop(self)	讓 Robot 停止
def set_message_handler(self, callback)	監聽 Robot 的狀態，參數 callback 為 json dict 格式，兩個 key 分別為 status 與 target status 代表 robot 當前狀態, target 則是對應狀態的目標桌號

Robot 狀態:

set_message_handler 回傳內容中 status 的狀態

delivering	送餐到 target 桌
arrived	抵達 target 餐桌
blocking	無法正常移動到 target 桌
return	Robot 正在返回起始點 (當完成送餐或是出現 blocking, 則 robot 會返回)
straight	Robot 正在往前行走
backward	Robot 正在往後行走
turn_right	Robot 正在向右轉
turn_left	Robot 正在向左轉