

停車場車位顯示系統

當車輛進入賣場停車場時，常因無法即時掌握空位而繞行尋找，造成時間浪費與行車不便。為了解決此問題，開發停車場車位顯示系統，透過感測與網頁整合技術，實現即時監控與車位預約功能。本系統使用 Arduino 配合光敏電阻偵測車位狀況，搭配 LED 燈模擬現場車位指示；資料由 Python 程式串接，透過 MySQL 整合至後端資料庫，並透過 Apache 與 Ngrok 將網頁顯示於前端。使用者可透過網頁登入帳號查看即時車位資訊，並進行線上預約。成功預約後，系統將鎖定車位 10 分鐘，並控制現場燈號，若使用者逾時未到或取消，系統將自動釋放車位。本系統具備遠端監控、低成本、易擴充等優勢，可應用於賣場、商辦或中小型停車場。

2.軟體操作流程圖



圖 4 ngrok 圖 5 tinkercad

```
graph TD; Start([開始網頁]) --> Login[登入網頁]; Login --> Select[選擇想預約的車位]; Select --> Confirm[預約成功]; Confirm --> Wait[計時十秒]; Wait --> Cancel{是否 按下取消預約}; Cancel -- YES --> CancelAction[取消預約]; Cancel -- NO --> CancelAction; CancelAction --> Start; Wait --> Timeout{是否十分鐘已到}; Timeout -- YES --> CancelAction; Timeout -- NO --> Confirm;
```

The flowchart illustrates the online reservation process. It begins with '開始網頁' (Start website), leading to '登入網頁' (Login website), then '選擇想預約的車位' (Select the seat you want to reserve). This leads to '預約成功' (Reservation successful), followed by '計時十秒' (Timing 10 seconds). A decision point '是否 按下取消預約' (Whether the cancel reservation button is pressed) follows. If 'YES', it leads to '取消預約' (Cancel reservation), which loops back to the start. If 'NO', it leads to another decision point '是否十分鐘已到' (Whether 10 minutes have passed). If 'YES', it leads to '取消預約'. If 'NO', it loops back to '預約成功'.

圖 6 軟體操作流程圖

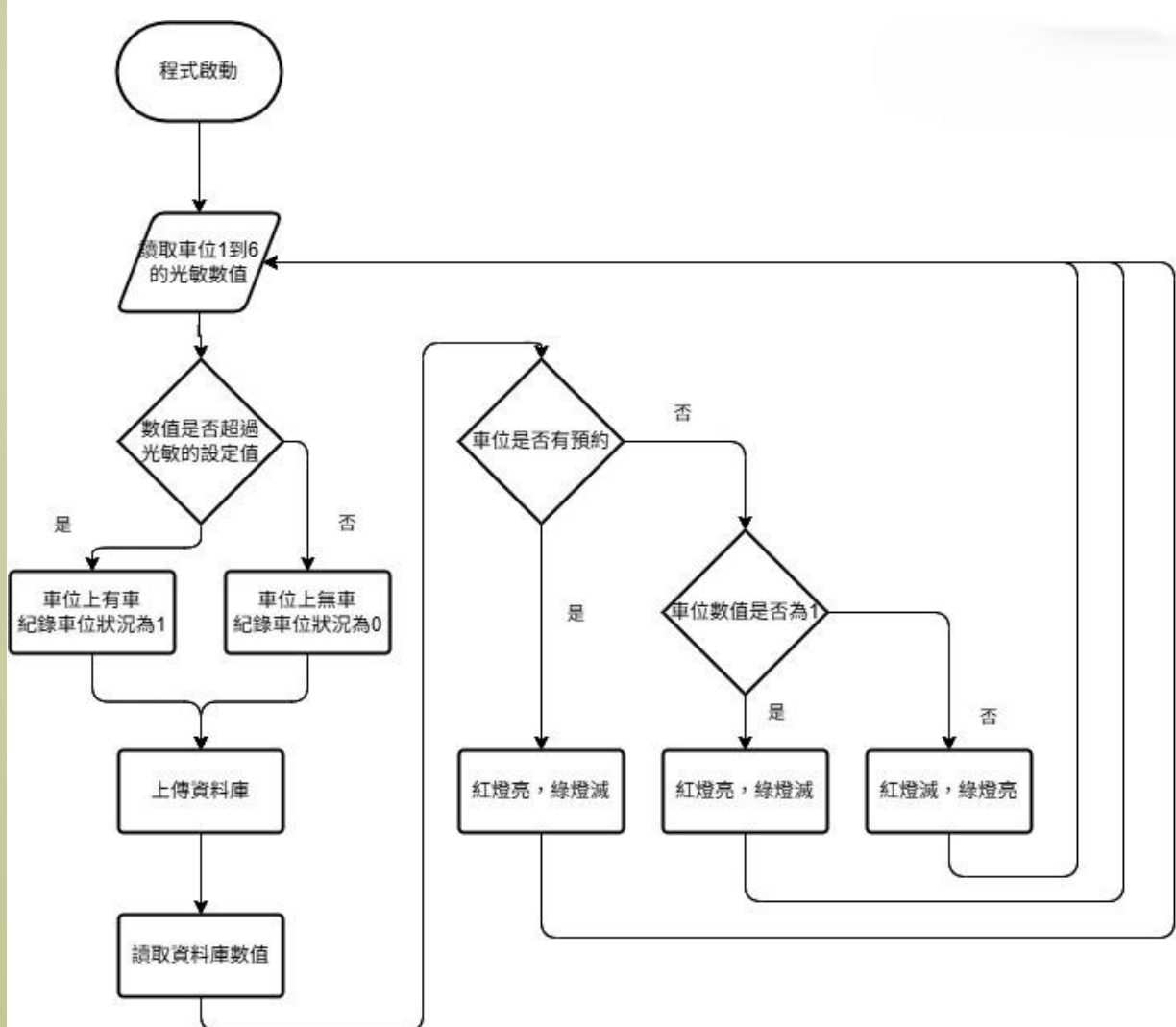


圖 7 Arduino 程式流程圖

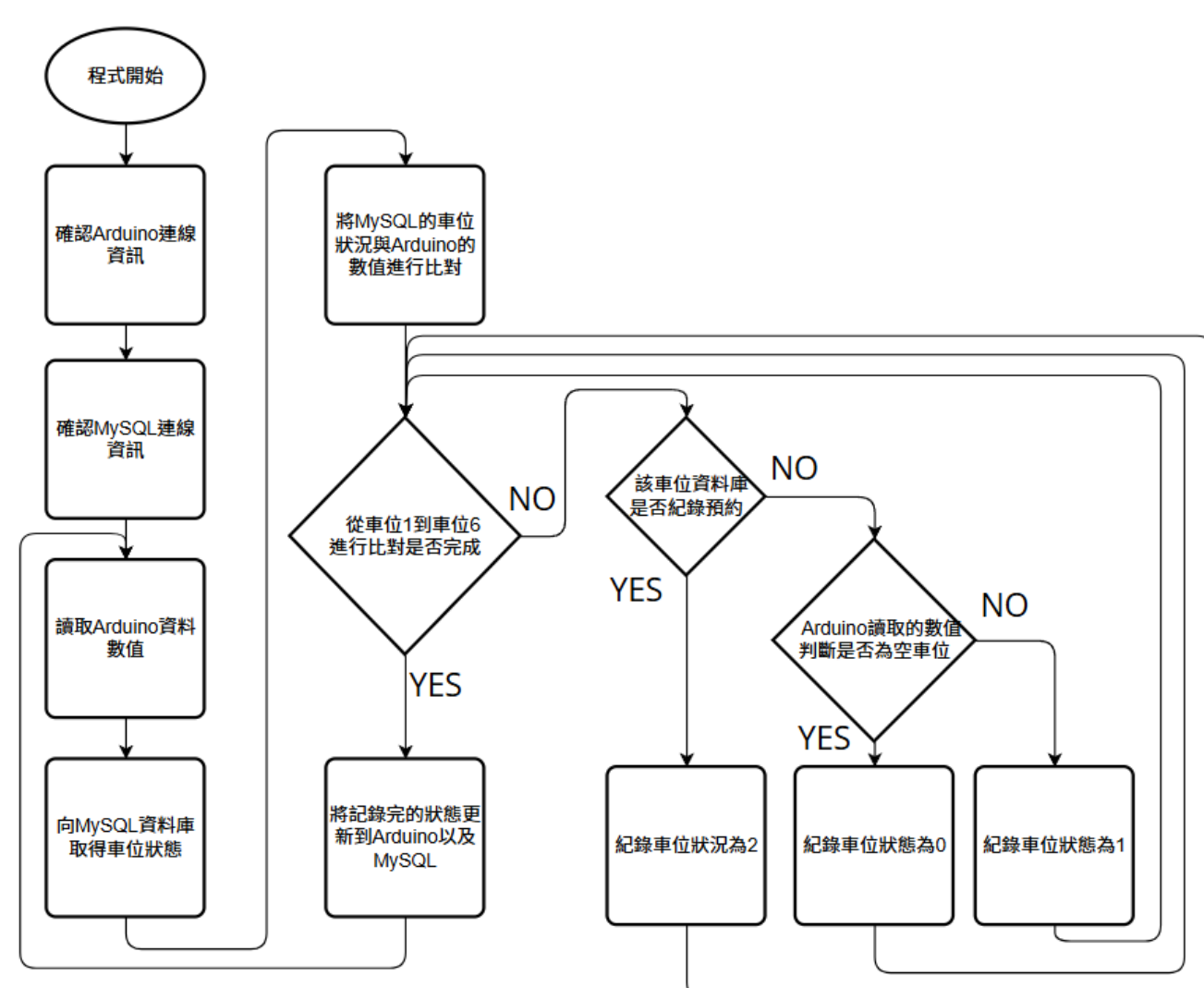


圖 8 資料庫Python程式流程圖

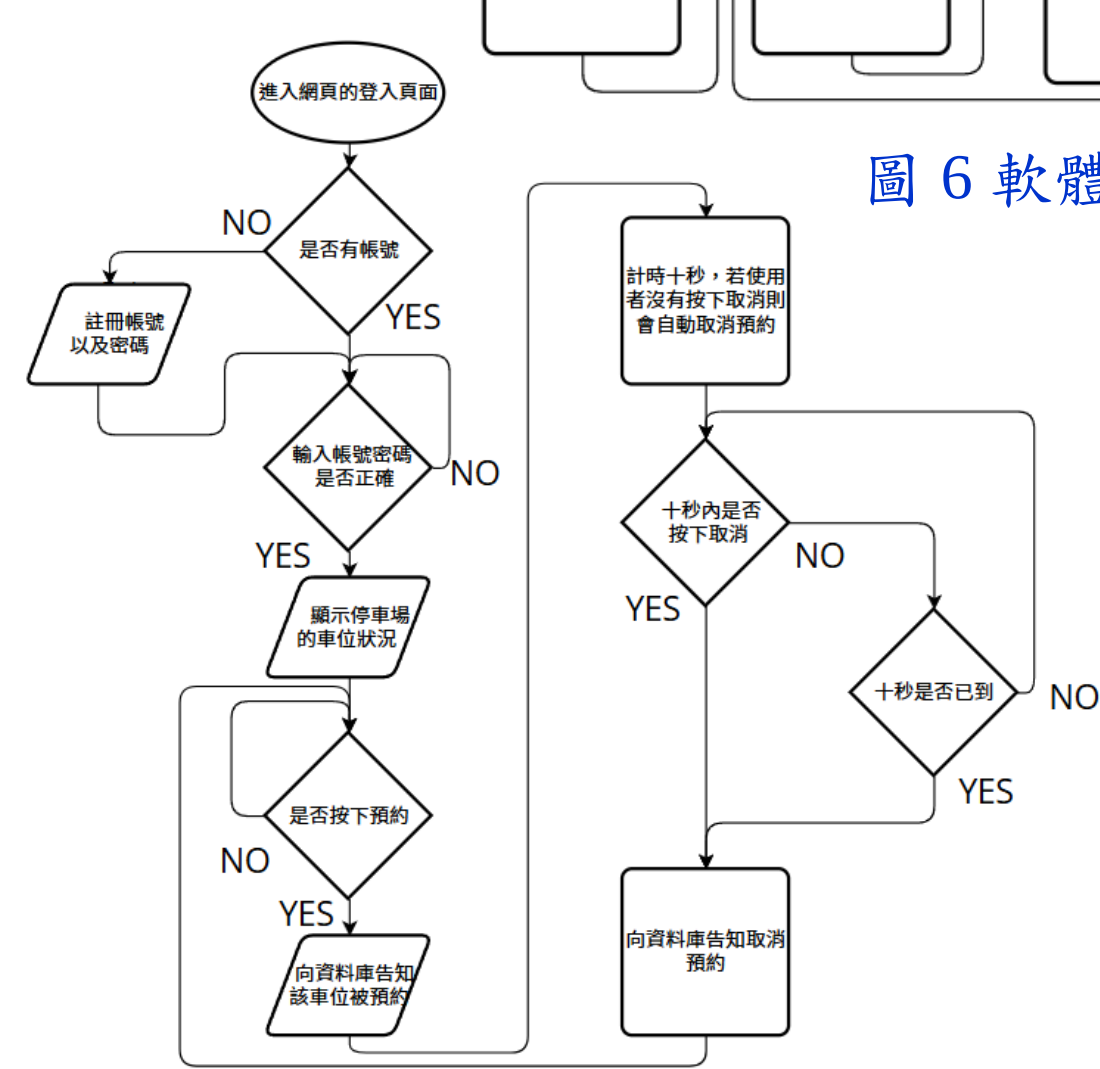


圖 9 網頁程式流程圖

5.重要程式碼

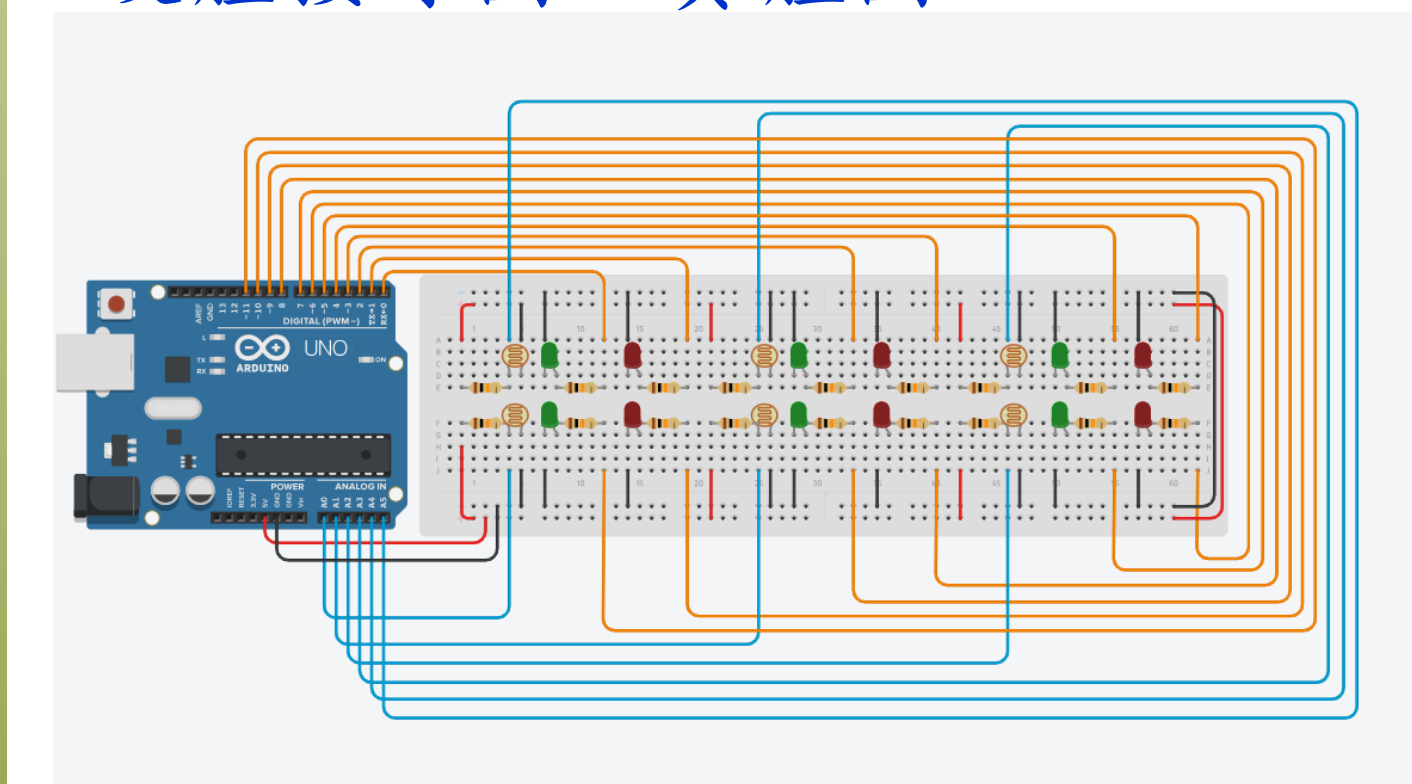


圖 10 硬體接線圖



圖 13 Arduino程式碼

登入

使用者名稱

密碼

登入

還沒有帳號？[點此註冊](#)

使用者名稱

密碼

註冊

圖 15 登入頁面

圖 16 註冊頁面

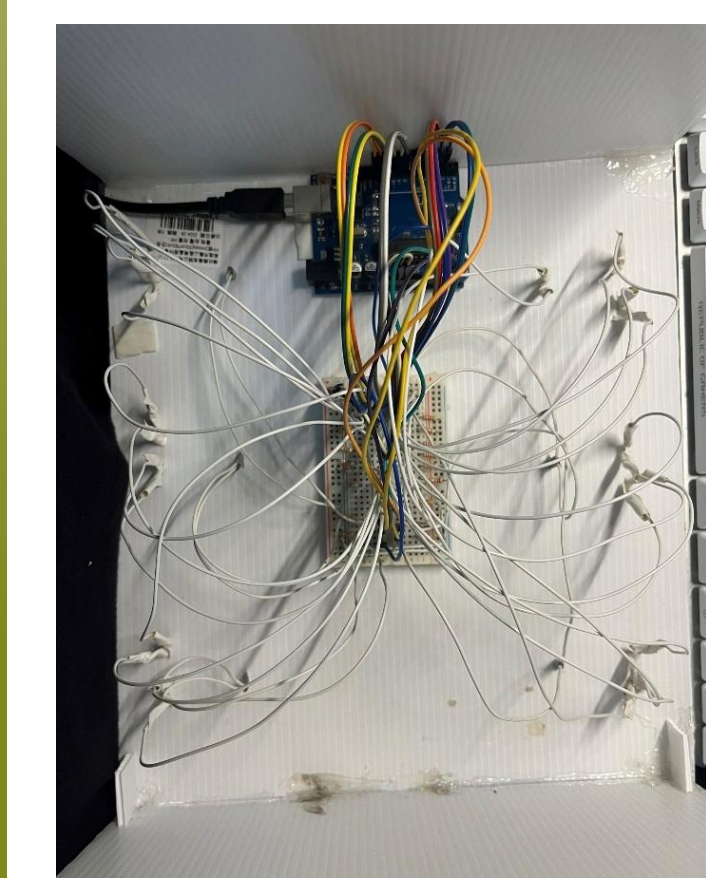


圖 11 硬體模型背面

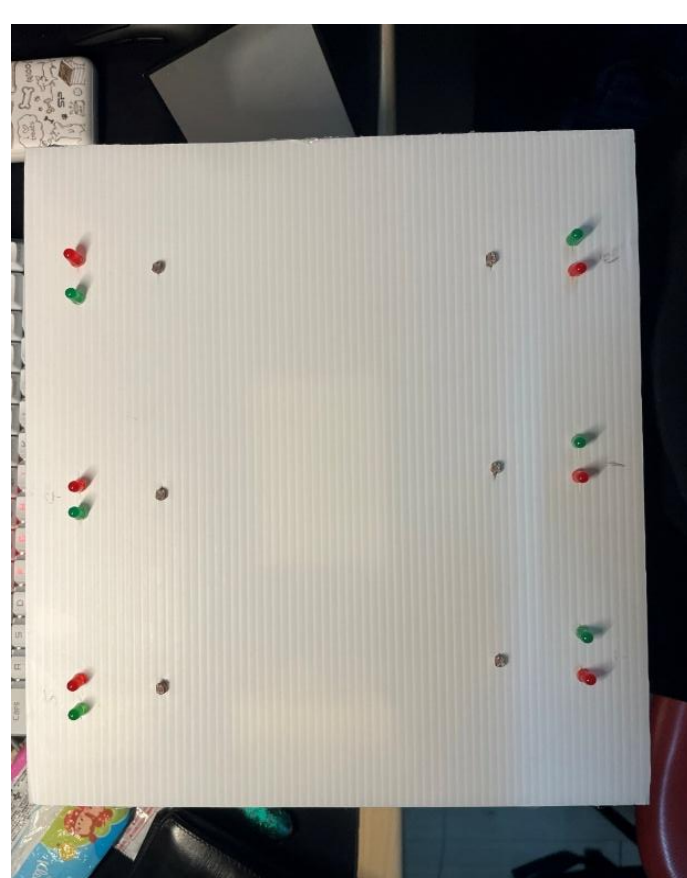


圖 12 硬體模型正面

```

18 while True:
19     # 讀取 Arduino 傳來的數據
20     if ser.in_waiting > 0:
21         data = ser.readline().decode().strip("\n\r") # 讀取執行待發
22         if data:
23             slot_status = data.split(",") # 解析數據
24             print("Arduino 傳來數據: (slot_status)")
25
26             # MySQL 新增格的設置
27             cursor.execute("SELECT status FROM parking_slots")
28             updated_status = [int(row[0]) for row in cursor.fetchall()] # 讀取舊值 int
29
30             print(f"當前 MySQL 狀態: {updated_status}")
31
32             # 更新 MySQL
33             for i, status in enumerate(slot_status):
34                 status = int(status) # 讀取舊值 int
35                 if updated_status[i] != 2:
36                     cursor.execute("UPDATE parking_slots SET status = %s WHERE id = %s", (status, i+1))
37
38             db.commit()
39
40             # 發送 MySQL 新的狀態回 Arduino
41             send_data = "%,".join(map(str, updated_status)) + "\n" # 讀取格式字符串
42             ser.write(send_data.encode())
43             print(f"發送回 Arduino: {send_data}")
44
45             db.commit() # 提交資料插入
46             time.sleep(0.5)

```

圖 14 Python程式碼



圖 17 已預約頁面

圖 18 已有車位被停頁面

