

# 智慧雲端多車管理系統

專題編號：109-CSIE-S019

執行期限：108 年第 1 學期至 109 年第 1 學期

指導教授：陳彥霖教授

專題參與人員： 106590008 邱奕凱

106590011 歐陽緒

106590013 羅柏凱

## 一、摘要

本專題以工業 4.0 為基礎，利用雲端平台的架構，架設倉儲管理自走車的監管系統，負責人員可以透過此系統監視自走車的當前狀況、設定任務以及透過大數據分析了解車輛之狀況，並希望可以將蒐集到的資料進行分析及整理，結合人工智慧達成更加自動化的系統。

透過 MEAN (MongoDB, Express.js, Angrular.js, Node.js) 來架設雲端平台，並將此平台架設於 Amazon Web Services (AWS)，以及加入排程設定、路徑規劃使自走車能以最佳路線去完成目標任務。

利用物聯網 (IoT) 的概念，定時接收倉儲中的自走車之資訊，並顯示於網頁中使負責人員可以方便的對車輛分配任務及觀看車輛狀況，並在未來希望可以透過歷史資料分析之結果去增減車輛配置、查看車輛工作效率是否下降以及自動規劃路徑、排程和執行任務車輛，以提高整體稼動率。

**關鍵詞：**物聯網(IoT)、路徑規劃、網宇實體系統(CPS)、智慧工廠

## 二、緣由與目標

因應工業 4.0 的興起，我們想製作一個用於工廠自走車管控的系統，並與實驗室的學長溝通了解後，加入了與東元公司製作倉儲管理系統的計畫，協助製作一個自走車的監管系統。希望可以透過此專題解決工廠中的痛點，並達到以下目標：

- 整廠整線
- 提高稼動率
- 工業 4.0

## 三、系統架構

此系統主要是將地端的設備資料透過雲端傳送到雲端，再將資料存入資料庫後處理後呈現給使用者，而使用者也可以利用排程設定的功能設定路徑的資料等，如下圖所示。

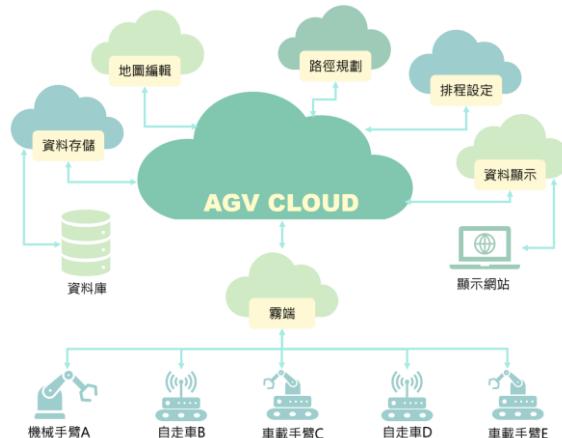


圖 1 系統架構

## 四、使用技術及方法

### (一) MEAN[1][2]

#### 1 MongoDB

一種以 document 導向的 NoSQL 資料庫系統，由於工廠內裝置包含信號塔、AGV 車及 UWB 車等，可以利用多型的特性及概念來儲存資訊。

#### 2 Express.js

利用 RESTful API for CRUD 以及 Express.js 提供的架構開發較好維護的 Web application。

#### 3 Angrular.js

強調於應用 MVC 模式下以及雙向資料綁定的優勢來達到前端撰寫的方便性。

#### 4 Node.js

適用於中小型團隊開發，且效能比傳統的 PHP 快上許多，且擁有功能強大的 npm 管理套件包來提升專案的擴展性。

## (二) Scalable Vector Graphics , SVG

可縮放向量圖形，這種圖形格式最大的特色是不被解析度限制可以自由地縮放，並且可利用自身所包含的座標系統精準的對照出相對位置。

## (三) 路徑規劃

運用 Dijkstra 演算法與 A\*演算法：主要利用於任務中的路徑規劃；廣度優先搜尋演算法：利用於自動排程中搜尋可執行任務之設備。

## (四) 排程設定

我們主要關注於三種問題，設定與檢查：在設定任務時會檢查設定自料是否正確及防範衝突的發生；時間排程：透過時間排程的設定與 cron 套件的使用來達到定時輪詢的功能；並行處理：當任務在實行是因為是線性結構，因此會導致排程中子任務與子任務的循環等待。

## 五、 專題成果

### (一) 使用者儀錶板

使用者可以在此畫面(圖 2)檢視車輛及設備在地圖中的位置、檢視設備的即時狀況及任務進行狀況。



圖 2 使用者儀錶板

### (二) 地圖編輯器

使用者可以在此頁面依照需求對不同的地圖進行編輯，如新增站點或障礙物，讓之後的任務設定更加方便及完善。

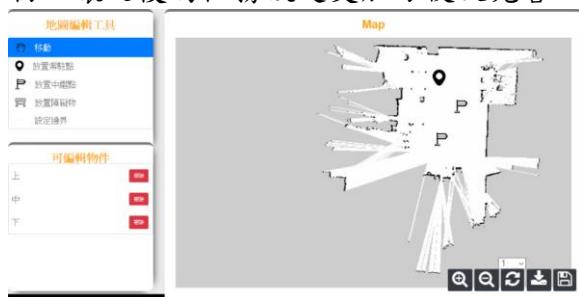


圖 3 地圖編輯頁

## (三) 任務設定

使用者可以透過任務設定的功能對車輛進行任務的設定，設定完任務後使用者可以使用設定完的排程，透過 API 對地端的車輛下達指令。



圖 4 任務設定



AGV 照片

## (四) 成果

右側 QRcode 為實驗室的學長使用我們所開發的平台所錄製的成果影片。



## 六、 後續發展可能性

### (一) 排程設定自動化

現階段我們所設定的排程是以車輛為主來做設定的，但這樣無法有效的提高工廠的稼動率，也會有許多因設備規格導致的問題，所以我們希望透過設備自動化排程的方式，讓使用者以原料為主設定排程，透過廣度優先搜尋的方式找尋距離最近且符合規定的裝置，來解決此問題。

### (二) 排程並行處理

我們的排程目前是以線性的架構執行，因此所有的排程與子任務都只能逐一執行，這會導致若其他任務與此任務並無關聯的話也會陷入等待導致無法如期執行。因此未來希望可以使用其他結構的排程演算法來避免這種狀況發生。

## 七、 參考文獻

- [1] 姜琇森, 蕭國倫, 許瑋苓, 陳怡君, 何彥儒, 黃豐凱, 一次搞懂 JavaScript 全端技術 jQuery、Node.js、Express、MongoDB，加魅數位，2019
- [2] Refsnes Data, “W3schools”, Internet: [www.w3schools.com](http://www.w3schools.com), 1999-2020
- [3] 謙景裕, 李明哲, 蔡維峻, 陳永源. “A Multiple Purpose Path Routing Algorithm” in NCS, 2009