



國立臺北科技大學

電子工程系碩士班  
碩士學位論文

設計與實作工廠即時監控資訊系統  
Design and Development of the Factory Real-Time  
Information Monitoring System

研究生：林清河

指導教授：黃士嘉 教授

中華民國一百零九年一月二日

## 摘要

論文名稱：設計與實作工廠即時監控資訊系統

頁數： 35

校所別：國立臺北科技大學 電子工程系 碩士班

畢業時間：一百零九學年度 第一學期

學位：碩士

研究生：林清河

指導教授：黃士嘉 教授

關鍵詞：工業 4.0、智慧化工廠、工具機、感應模組、即時監控

近年來工業 4.0 技術發展越趨成熟，生產模式從以前的大量生產、代工製造方式轉變為以即時需求、彈性調整及客製化生產模式的智慧化工廠概念邁進。為達到智慧化工廠的目標，首先須整合工廠內工具機的資訊，從而快速了解工廠內的運作情形。然而傳統工廠可能由於工具機機型老舊不具備資料傳輸的能力造成監控困難或是依靠人工方式記錄工廠的運作資料，成為智慧化工廠的瓶頸。

本論文實作一套「工廠即時監控資訊系統」提出無須仰賴工具機本身的傳輸方式，利用在工具機旁安裝感應模組並藉由微電腦控制器控制用以讀取工具機上的各項資訊並傳送到系統內記錄下來。工廠管理人員可以透過電腦或是智慧型裝置即時監控工廠內的運作情形，並可利用報表來了解歷史資料，為達到智慧化工廠建立夯實的基礎。

# ABSTRACT

Thesis Title : Design and Development of the Factory Real-Time Information Monitoring System

Page : 35

University : National Taipei University of Technology

Graduate School : Master Program, Department of Electronic Engineering

Date : January, 2021

Degree : Master

Graduate Student : Ching-Ho Lin

Advisor : Shih-Chia Huang, *Prof.*

Keywords : Industry 4.0 、 Smart Factory Model 、 Machine 、 Sensor 、 Real-Time

In the past few years, the development of Industry 4.0 technology has become more and more mature, and the mode of production has changed from mass production and OEM manufacturing to the smart factory model which is based on instant demand, flexible adjustment and customized production. In order to achieve the goal of smart factory, first of all, it is necessary to integrate the information gathered from machines in the factory to quickly understand its operational status. However, traditional factories may be difficult to monitor as machine models are too antiquated to transmit data, or rely on manual input of factory operation data, which becomes a bottleneck for smart factories.

This paper implements a "Factory Real-Time Information Monitoring System" and proposes a method which does not need to rely on data transmission by the machine tool itself.

We place a sensor on the machine and use the micro controller of the sensor to read various information from the machine and send it our system. Factory managers can use computers or smart devices to monitor the operating status from the factory in real time, and record all the information as reports. This allows them to understand historical data and establish a solid foundation for realizing the smart factory model.

## 誌 謝

本論文承蒙科技部為落實知識應用研究並結合民間企業需求，補助產學合作研究計畫案。感謝本次的合作企業：群亞電子股份有限公司，於合作的過程中分享與傳授實務上的經驗及應用，並且提供工廠場地及技術人員配合進行實驗研究。

時光飛逝轉眼間碩士求學日子即將邁入尾聲，在這近兩年半的學習時光中不僅學到許多專業知識，更重要的是透過不斷的閱讀論文及上台報告的訓練，學習到如何從中分析到有用的資訊並加以整合成報告，給聆聽的人能夠清楚理解我講的內容。

在此我要先感謝我的指導老師 黃士嘉老師，由於本身是在職生白天時間需要上班且時常會需要加班，因此只能利用晚上時間與老師討論而老師也能夠體諒並配合，老師也會分享本身的專業知識見解，讓我在這趟求學旅程受益匪淺、收穫良多，感謝老師的敦敦教誨。

感謝在這段求學期間的每位任課老師，能夠體諒學生因為白天工作繁忙使得上課期間比較晚進入教室，而交出的報告也會仔細閱讀並予以告知錯誤讓我們更加了解。除了傳授課本知識外也會分享本身的人生經驗給我們，讓我們除了學習專業知識外也多了許多人生體悟。

最後感謝公司同事支持體諒，讓我能夠有這個機會到學校進修，並時常詢問我工作量會不會很大會不會影響課業，令我可以在完成工作之餘課業也能夠掌握，而不會趕不上進度。

# 目 錄

摘 要.....	i
ABSTRACT.....	ii
誌 謝.....	iv
目 錄.....	v
表目錄.....	vii
圖目錄.....	viii
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機.....	2
1.3 研究目的.....	2
1.4 論文章節編排.....	3
第二章 相關研究.....	4
2.1 JSON.....	4
2.2 Spring MVC 框架.....	5
2.3 RWD.....	6
2.4 Android WebView.....	7
2.5 WeMos D1 WiFi.....	8
第三章 系統設計.....	9
3.1 系統架構.....	9
3.2 系統功能規劃.....	16
第四章 系統開發成果.....	20
4.1 工廠即時監控資訊系統.....	20

4.2	感應裝置設置 .....	24
4.3	工廠實驗 .....	26
4.4	即時監控傳輸實驗 .....	30
第五章	討論 .....	32
第六章	結論與未來方向 .....	33
6.1	結論 .....	33
6.2	未來方向 .....	33
參考文獻 .....		34

## 表目錄

表 2.1	WebView 常用方法 .....	7
表 3.1	感應模組列表 .....	10
表 3.2	資料表命名方式列表 .....	13
表 4.1	實驗結果 .....	31
表 5.1	傳統工廠與工廠即時監控資訊系統做法比較表 .....	32



## 圖目錄

圖 1.1	工業革命發展史 .....	1
圖 2.1	JSON 示意圖 .....	4
圖 2.2	Spring MVC 框架 .....	5
圖 2.3	使用 RWD 裝置顯示示意圖 .....	6
圖 2.4	使用 Android WebView 架構圖 .....	7
圖 2.5	WeMos D1 WiFi 架構圖 .....	8
圖 3.1	系統架構圖 .....	9
圖 3.2	控制層資料傳輸流程圖 .....	11
圖 3.3	監控層流程圖 .....	12
圖 3.4	資料庫關聯圖 .....	14
圖 3.5	即時監控資訊系統架構圖 .....	15
圖 3.6	系統功能規劃架構圖 .....	16
圖 3.7	儀表板資料流程圖 .....	17
圖 3.8	調整感應裝置流程圖 .....	18
圖 4.1	主畫面 .....	20
圖 4.2	調整感應裝置畫面 .....	21
圖 4.3	警戒值設定畫面 .....	22
圖 4.4	溫濕度報表資料畫面 .....	23
圖 4.5	溫濕度及重量感應模組 .....	24
圖 4.6	溫濕度及火災警報模組 .....	25
圖 4.7	電源控制模組 .....	25
圖 4.8	裁切機實驗畫面 .....	26

圖 4.9	烤箱實驗畫面 .....	27
圖 4.10	洗床機實驗畫面 .....	28
圖 4.11	廠區任意位置實驗畫面 .....	29
圖 4.12	燈具測試區實驗畫面 .....	29
圖 4.13	連接多台感應裝置實驗畫面 .....	30
圖 4.14	感應裝置回應時間 .....	31

# 第一章 緒論

## 1.1 研究背景

由於工業 4.0[1][2]的興起，傳統工廠也正面臨轉型的問題，如圖 1.1 所示，綜觀工業發展史來看，第一次的工業革命為機械化，是利用水力與蒸氣的力量當作動力來取代原有靠著人力、畜力、水利等方式，達到機械化生產；而第二次工業革命則是電氣化，發展出利用電力的方式達到大量生產的目標；第三次工業革命為資訊化，使用電子裝置及資訊科技取代人為操作機器的方式達到數位化自動生產，增強製造過程中的精準度及自動化流程；工業 4.0 又稱為生產力 4.0，也就是第四次工業化革命，其概念是升級生產製造能力由自動化朝向智慧化的方式邁進，藉由收集工廠內各個工具機的資料並將其透過分析、統計後，顯示工廠內各工具機的即時生產狀態，依據現有狀況找出最有效率方式，改變傳統單一生產模式實現靈活生產製作的目標，以達到智慧化工廠[3]的目的。



圖 1.1 工業革命發展史

## 1.2 研究動機

為了朝向智慧化工廠邁進，首先要可以獲取工廠內工具機設備的各項資訊，現今新出廠的工具機設備都已具備資料傳輸的能力以及能夠透過無線網路將資料分享出去，只要透過統一通訊標準的方式即可收集工具機的資訊。然而傳統工廠的問題在於仍有許多工具機設備因其機型老舊不具備資料傳輸的能力或是此前工具機設備缺乏系統性的規劃及整合，無統一的通訊標準造成收集工具機設備的資訊困難且工具機的造價昂貴汰換不易，因此在資料收集及統計還是得依靠工廠管理人員親力親為，但資料的收集依靠人工效率非常低、準確力也不高。另一個問題在於，工具機設備由於時常需要全天候運轉，但可能會因為過熱或是濕度過高造成散熱困難使得工具機意外停止運作，若無法即時發現將影響工廠產能，工廠可能採用人力巡邏的方式定時檢視工具機的運作情形，然而傳統工具機本身幾乎都沒有顯示溫濕度的螢幕，因此只能透過手摸進行檢測，往往因為主觀認知或是一時的疏忽而判斷有誤，造成工廠管理人員要進行決策時的資訊有落差而判斷錯誤。

## 1.3 研究目的

本論文實作一套「工廠即時監控資訊系統」，提出工廠管理人員可以使用電腦或是智慧型裝置即可監控工廠內工具機設備運作的狀況，而無須汰換現有的工具機設備，也不需要依靠人力的方式收集資料，只要在工具機旁邊安裝感應裝置即可透過無線網路將資料傳輸到系統中，透過該系統可解決下述的問題：

1. 自動收集及保存工具機資料而不依靠人力方式，增加準確率及效率。
2. 令傳統工具機增加資料傳輸能力。
3. 解決缺乏遠端即時監控方式。

## 1.4 論文章節編排

本論文將會分成六個章節，各章節內容與編排方式如下所示：

### 第一章：緒論

本論文的研究背景、研究動機、研究目的與論文章節編排。

### 第二章：相關研究

本論文所使用的相關技術背景介紹。

### 第三章：系統設計

本論文系統架構、流程及功能規劃。

### 第四章：系統開發成果

本論文系統成果展示、工廠實際操作及實驗結果。

### 第五章：討論

本論文與傳統工廠比較。

### 第六章：結論與未來方向

本論文結論及未來可研究方向。

## 第二章 相關研究

本章節旨在介紹基於本論文實作過程中所運用的相關技術，分為 JSON、Spring MVC 框架、RWD、Android WebView 及 WeMos D1 WiFi 五個部分。

### 2.1 JSON

JavaScript 物件表示法(JavaScript Object Notation, JSON)是種以純文字當作基底進行儲存和傳送的資料交換語言，能夠將複雜的資料轉換成易於閱讀的方式，就可以輕鬆的與其它系統進行資料交換，並且支援許多資料格式，最常用作網頁間系統的資料交換。

JSON 的變數名稱為字串必須用雙引號包括，資料內容可以是字串、數字、布林值、物件及陣列等等，變數之間以逗號隔開，如圖 2.1 以本系統實作的資料格式為例，Sensor 是一個物件裡面包含從感應裝置接收到的三種資料，第一個變數名稱為"humidity"，代表濕度裡面儲存字串資料，第二個變數名稱為"temp\_cal"，代表溫度裡面儲存數字資料，第三個變數名稱為"fire\_status"，代表火光感應狀態裡面儲存布林值。

```
{
  "Sensor" :
  {
    "humidity" : "70%",
    "temp_cal" : 30,
    "fire_status" : true
  }
}
```

圖 2.1 JSON 示意圖

## 2.2 Spring MVC框架

Spring MVC[4]是以 Java 程式語言作為開發的框架，是一種用在建構 Web 應用程式[5]的 MVC 架構，圖 2.2 為系統架構圖其原理為：

1. 當使用者透過瀏覽器發動請求，會包含使用者所請求的訊息、請求的網址及其它的資訊。
2. 當請求透過瀏覽器傳送到伺服器後，會先被前端控制器（DispatcherServlet）捕捉進行處理，根據請求包含的訊息及網址查詢註解或是 XML 設定檔找到對應的處理映射器（HandlerMapping）。
3. 前端控制器根據處理映射器的回覆查詢該請求要交給哪一個控制器（Controller）進行處理。
4. 前端控制器依據選定控制器後會將請求交給該控制器。
5. 控制器根據訊息的內容找到適合的業務邏輯和方法進行處理並將處理後的內容轉換成模型和視圖名稱回覆給前端控制器。
6. 前端控制器使用視圖解析器（ViewResolver），依據視圖名稱匹配具體的視圖物件。
7. 視圖(View)會在結合模型資料內容與匹配到的視圖物件進行渲染生成畫面。
8. 前端控制器最後會將渲染的畫面傳遞至瀏覽器呈現給使用者。

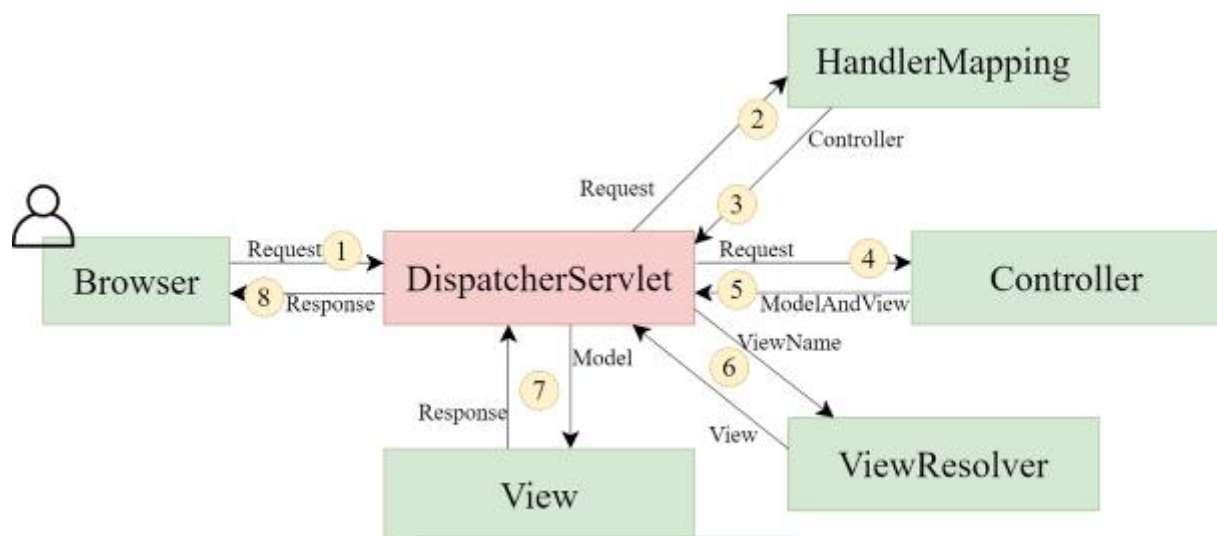


圖 2.2 Spring MVC 框架

## 2.3 RWD

響應式網頁設計（Responsive Web Design, RWD）[6]是一種網頁設計的技術，比起傳統網站沒有考慮到智慧型裝置的處理，當使用手機瀏覽傳統網頁時，會因為螢幕尺寸過小而造成排版亂掉或是文字及圖片過小，還需要特別放大、拖拉網頁才可以正常瀏覽網頁使得閱讀不易，大大增加網頁瀏覽的困難度。

當使用 RWD 後可以讓網頁在不同尺寸的螢幕或是裝置都能夠呈現適合比例的設計原則，減少使用者需要針對網頁進行額外的平移和縮放等行為，讓使用者可以一目瞭然的看到網頁增加瀏覽的便利性。如圖 2.3 所示，分別展示了使用 RWD 後於電腦螢幕、平板及手機等不同尺寸螢幕的情況下所顯示的頁面示意圖。在使用電腦螢幕觀看網頁時會依據瀏覽器的大小調整頁面使得頁面始終保持滿版置中並適當的調整網頁文字及圖片大小；當使用行動裝置如平板觀看網頁時，會依據螢幕尺寸大小作調整，例如原先在左列的功能選單會於畫面最上面呈現；當使用較小的裝置如手機觀看網頁時，會將頁面的內容重新排版，使得頁面上的文字及圖片都可以有最佳的觀看效果，而不用在左右滑動頁面，讓使用者有最佳的體驗[7]。

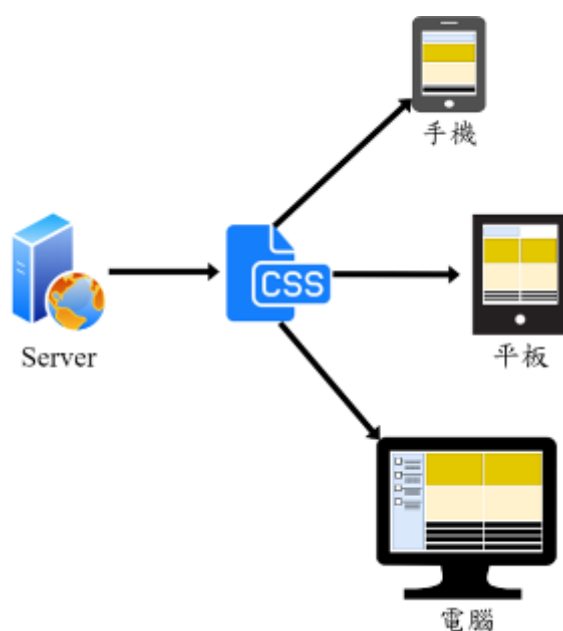


圖 2.3 使用 RWD 裝置顯示示意圖



## 2.4 Android WebView

WebView[8]是 Android 應用程式開發的一種元件，可以將其視為一個瀏覽器，利用該元件透過簡易的設定即可載入及顯示網頁並製作成應用程式。架構如圖 2.4 所示，當使用智慧型裝置打開應用程式後，應用程式會即時連線到設定好的網址，將網頁內容呈現在手機畫面上再搭配前述的 RWD 設計的網頁可以讓使用者很順暢的操作網頁。優點為只需建置一套 RWD 網頁系統，不用再額外設計手機應用程式，使網頁風格可以統一及降低開發成本且過程簡單快速，常用方法如表 2.1，只要設定幾種參數即可使用。

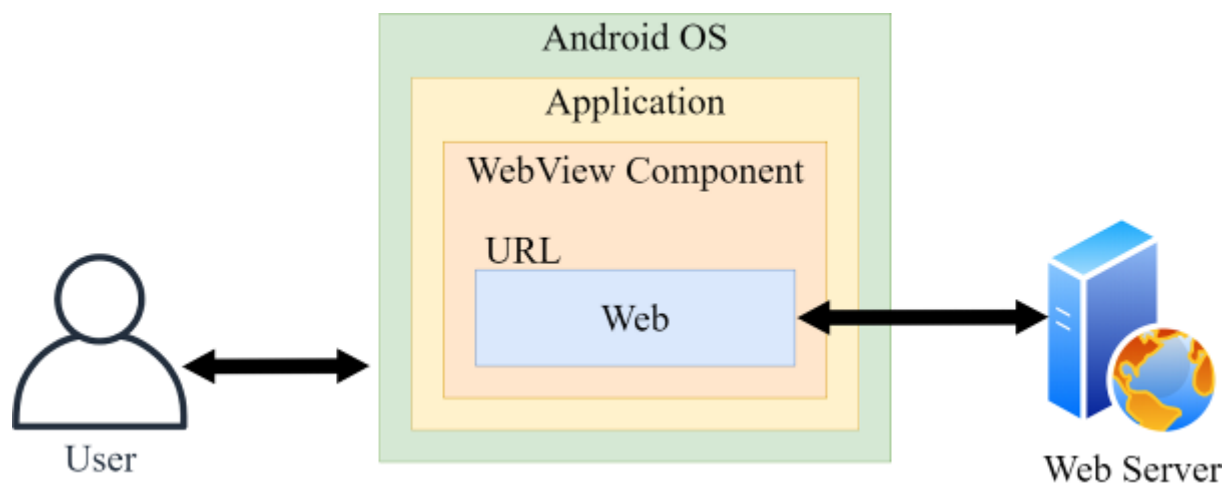


圖 2.4 使用 Android WebView 架構圖

表 2.1 WebView 常用方法

方法	說明
loadUrl(String url)	設定要連接的網址
canGoBack()	判斷網頁是否可以返回上一頁
goBack()	讓網頁返回上一頁
setJavaScriptEnabled(boolean flag)	設定是否啟用 javascript，通常設定為 true。RWD 網頁很多動作都是利用 javascript
setCacheMode(int mode)	設定網頁是否可以暫存
shouldOverrideUrlLoading()	載入網頁直接從應用程式的 WebView 顯示且不使用系統預設瀏覽器

## 2.5 WeMos D1 WiFi

Arduino[9]是基於開放原始碼的 I/O 微電腦控制器，其包含多種電路板並且使用類似於 Java、C 語言的程式語言，其具有高彈性、易使用，能簡單的透過感應模組收集周遭環境的變化及控制裝置等特性。

本論文使用 WeMos D1 WiFi[10]，其架構如圖 2.5 所示，屬於 Arduino 眾多板子中的一員，特點說明如下：

1. 有 11 個數位 I/O 接腳，其中 D3 與 D15、D4 與 D14、D5 與 D13、D6 與 D112、D7 與 D11 它們兩兩之間是互通。
2. 1 個類比輸入接腳，輸入範圍 0-3.3V。
3. 工作輸入電壓 6-24V。
4. 與其它電路板相比，其具有內建 WiFi 的優勢，不用額外連接 WiFi 模組。

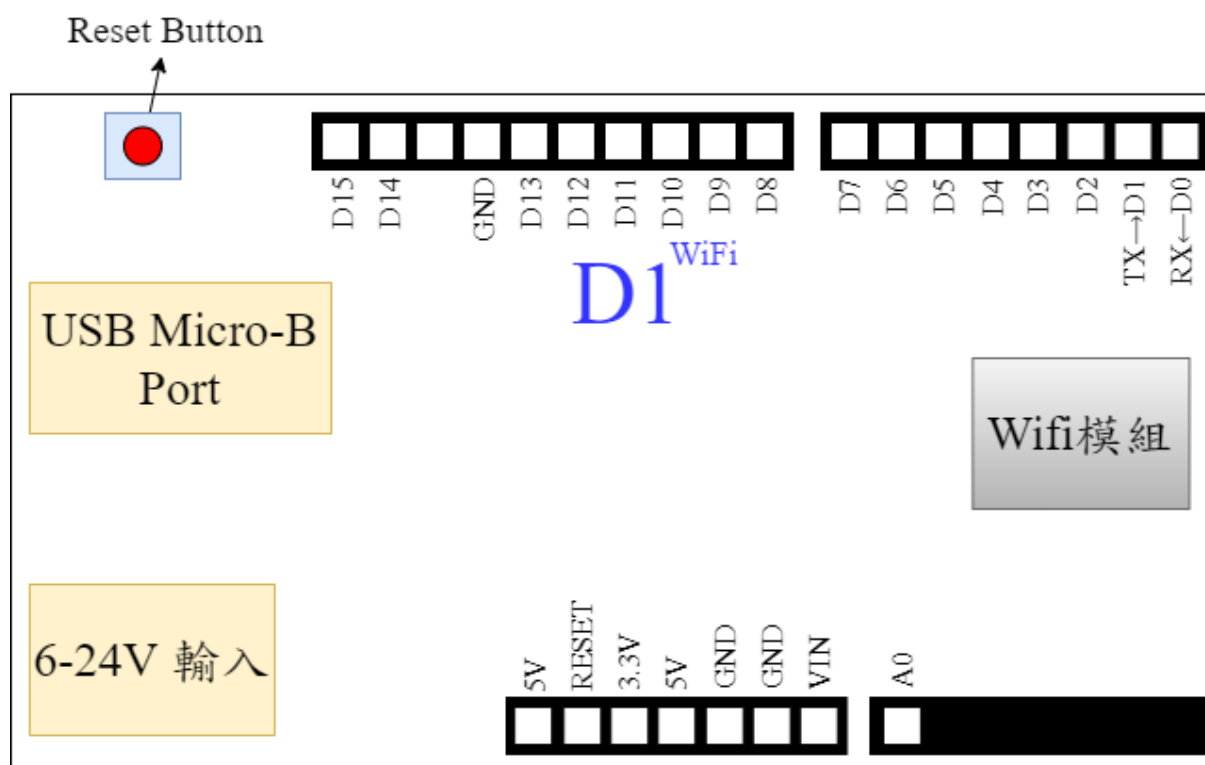


圖 2.5 WeMos D1 WiFi 架構圖

## 第三章 系統設計

### 3.1 系統架構

本論文為達到工廠遠端監控的能力，實作工廠即時監控資訊系統，系統架構如圖 3.1 所示共可分為控制層及監控層，於以下章節詳細介紹。

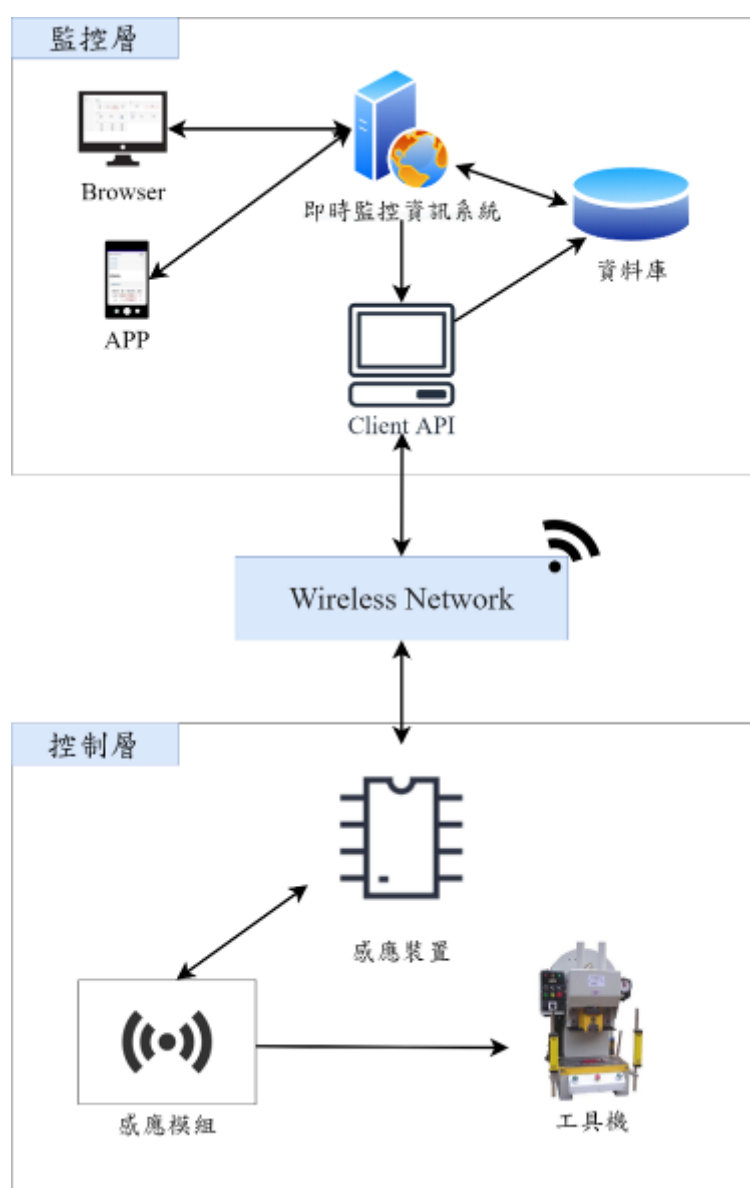


圖 3.1 系統架構圖

### 3.1.1 控制層

控制層主要作為與工具機溝通的媒介，利用微電腦控制器 WeMos D1 WiFi 控制各種感應模組運作來讀取工具機的資料，使得原先不具備資料傳輸能力的工具機也能將資料傳送出去。本次使用感應模組[11][12][13][14]相關資訊如表 3.1 所示，工廠管理人員可依照每一台工具機的特性選擇適合的感應模組安裝在工具機旁或是可以放置在工廠內適合的環境隨時監控整個工廠的狀況。

表 3.1 感應模組列表

感應模組名稱	參數	說明
溫濕度模組	溫度	測量範圍：0~50℃ 精準度：±2℃
	濕度	測量範圍：20~90% 精準度：±5%
重量模組	重量	範圍：0~500(g)
火光感應模組	火光反應	通電：有火光反應 沒通電：無火光反應
一氧化碳感應模組	一氧化碳	通電：一氧化碳濃度超過警戒 沒通電：一氧化碳濃度正常
繼電器	控制通電	打開：通電 關閉：沒通電

在控制層中能夠有多個微電腦控制器 WeMos D1 WiFi 同時運作，依照工廠管理人員的規劃進行增減。每個 WeMos D1 WiFi 都會設定一個唯一的 IP 用以辨識管理，工廠管理人員可以在每個工具機安裝多個感應模組。

本論文使用 Arduino IDE[15]當作開發工具並利用 Arduino 語法撰寫程式驅動 WeMos D1 WiFi 控制感應模組，並透過 HTTP 接收來自監控層 Client API 的請求，並透過 JSON 當作傳輸格式將讀取到的感應模組資料回傳給 Client API。

如圖 3.2 所示，當監控層的 Client API 發動請求後會依據 IP 找到對應的 WeMos D1 WiFi，而當 WeMos D1 WiFi 接收到請求後會依據請求的內容讀取感應模組。若成功讀取，則將讀取到的資料轉成對應的 JSON 格式回傳給 Client API；若感應模組有問題造成讀取失敗或是控制有問題，則會將失敗訊息轉換成 JSON 格式回傳給 Client API。

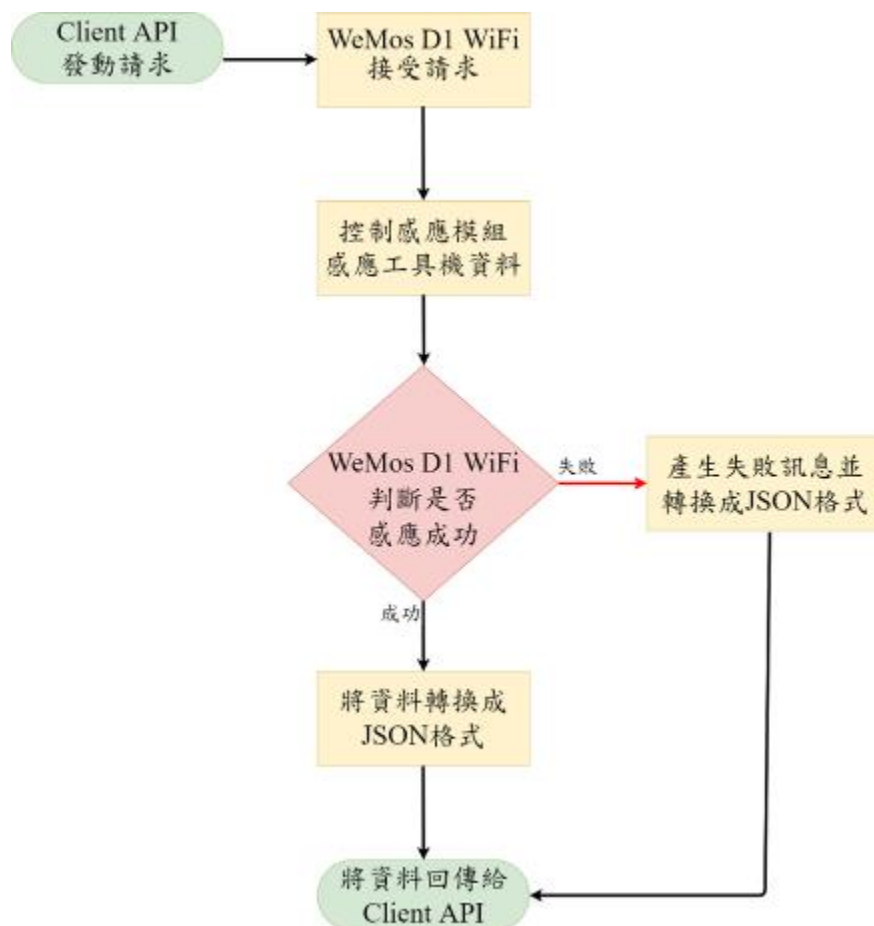


圖 3.2 控制層資料傳輸流程圖

### 3.1.2 監控層

監控層主要由即時監控資訊系統定時呼叫 Client API 後再與控制層請求資料，當 Client API 接收到來自控制層的資料後，會將工具機的相關資料直接寫入資料庫中。而即時監控資訊系統會定時從資料庫讀取資料保持最新狀況，以下將針對 Client API、資料庫及即時監控資訊系統進行說明。

## ■ Client API

Client API[16]是採用 Java 語言進行開發，利用 eclipse[17]工具當作程式開發環境，並將其設置在 Apache Tomcat[18]伺服器上運行。每經過一段時間即時監控資訊系統就會呼叫 Client API 向控制層發動請求，流程如圖 3.3 所示。當 Client API 接收到來自即時監控資訊系統的請求後，會將請求內容轉換成 JSON 格式後透過 HTTP 的方式向控制層請求資料，Client API 接受到來自控制層回應的資料後，會判斷資料是否有誤，若資料格式正常，則直接轉換成 SQL 並寫入資料庫對應的感應模組資料表中；若資料格式有誤或是其他因素造成問題，則會將錯誤訊息轉換成 SQL 寫入資料庫中的紀錄檔。

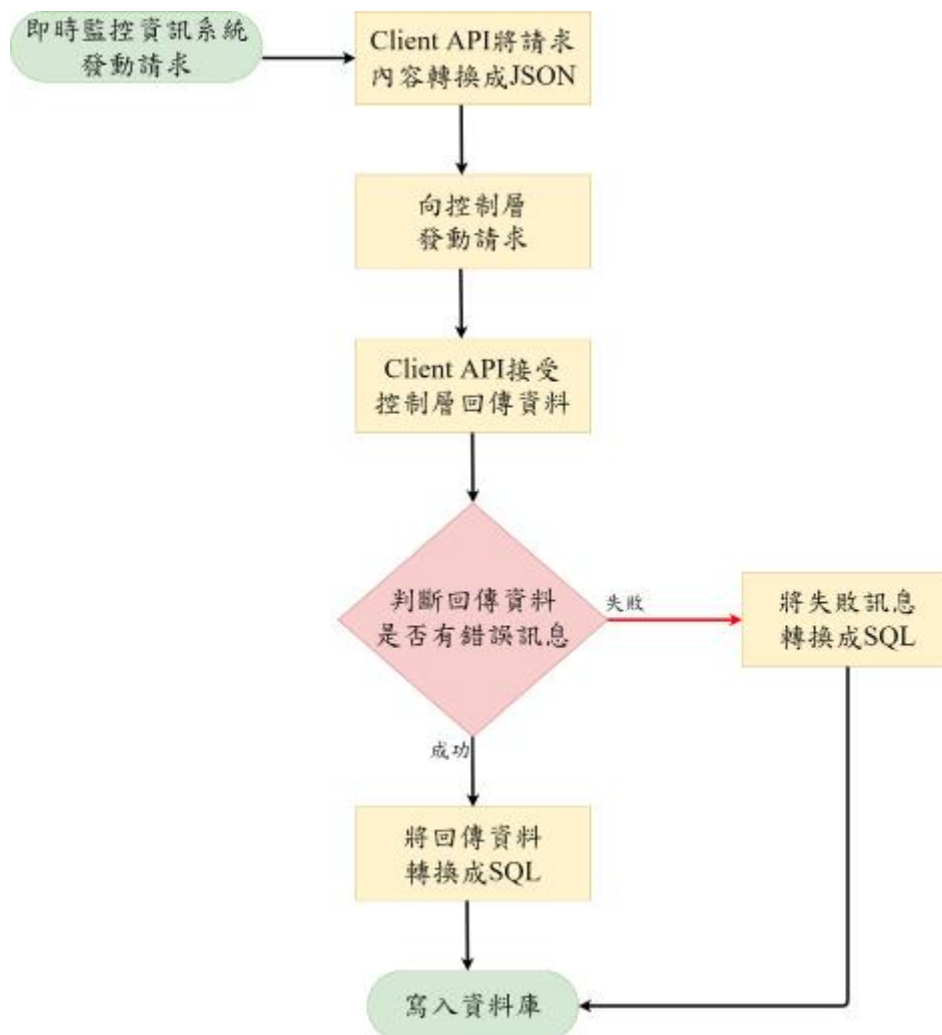


圖 3.3 監控層流程圖

## ■ 資料庫

本系統資料庫採用 MySQL[19]，它是一個關聯式資料庫管理系統，其具有開源、體積小及速度快等特性，許多中小型網站都採用 MySQL 當作資料庫使用。此次資料庫設計為了識別方便將資料庫的資料表區分為三大部分：感應裝置資訊、各類感應模組及系統使用說明。如表 3.2 所示，每個資料表都依據其特性將其歸類到專屬的類別，資料表命名方式：以"模組\_"加"功用"。

表 3.2 資料表命名方式列表

模組	說明	範例
Mod_	感應裝置資訊	Mod_Main 感應裝置主檔
Sen_	各類感應模組	Sen_Dht11 溫濕度模組
Sys_	系統使用	Sys_Resp_Status 回傳狀態代碼檔

如圖 3.4 所示為本系統的資料庫關聯圖，主要架構說明如下：

1. 感應裝置主檔 (Mod\_Main)：主要內容對應到控制層的感應裝置，每一筆資料都對應到一個 WeMos D1 WiFi，裡面記錄著唯一識別的 IP Address、裝置名稱以及是否啟用等資訊。
2. 感應模組 (Mod\_Sen)：功用為記錄目前所有的感應模組，依照不同工廠的特性可以設定不同的感應模組。
3. 模組參數資料 (Mod\_ParmData)：是用來記錄感應模組的各項參數資料，一個感應模組中可能包含多項參數可以使用，像是溫濕度模組就可以讀取到溫度及濕度參數，因此其與感應模組是 Many-to-One 的關係，裡面記錄著對應的感應模組 id、參數代號、參數名稱及該參數的警戒值。
4. 感應裝置使用模組 (Mod\_Main\_Sen\_R)：主要是用來記錄感應裝置主檔與感應模組之間的關係，由於在工廠中可能會有多個感應裝置而一個感應裝置又會同時控制多



個不同的感應模組，且同一個感應模組可能有多個感應裝置同時使用，因此它們之間是 Many-to-Many 的關係，將採用另一個資料表來記錄它們的關係。

5. 感應紀錄 (Mod\_Resp\_Log)：用以記錄每次 Client API 發動請求後從控制層接收到的資料紀錄，裡面包含感應裝置 id、感應模組 id 及回傳狀態，若從控制層感應的資料是正常，就會記錄正常狀態；若從控制層感應的資料有誤或是無法連接，就會記錄異常狀態及相關錯誤訊息。
6. Sen 開頭相關資料表：用以記錄每次從控制層取得的感應模組資料的相關資訊，依不同的感應模組記錄在不同的資料表上，像是 Sen\_Dht11 記錄溫濕度模組相關的資料包含感應裝置 id、溫度、濕度及感應時間；Sen\_Hx711 包含感應裝置 id、重量及感應時間。

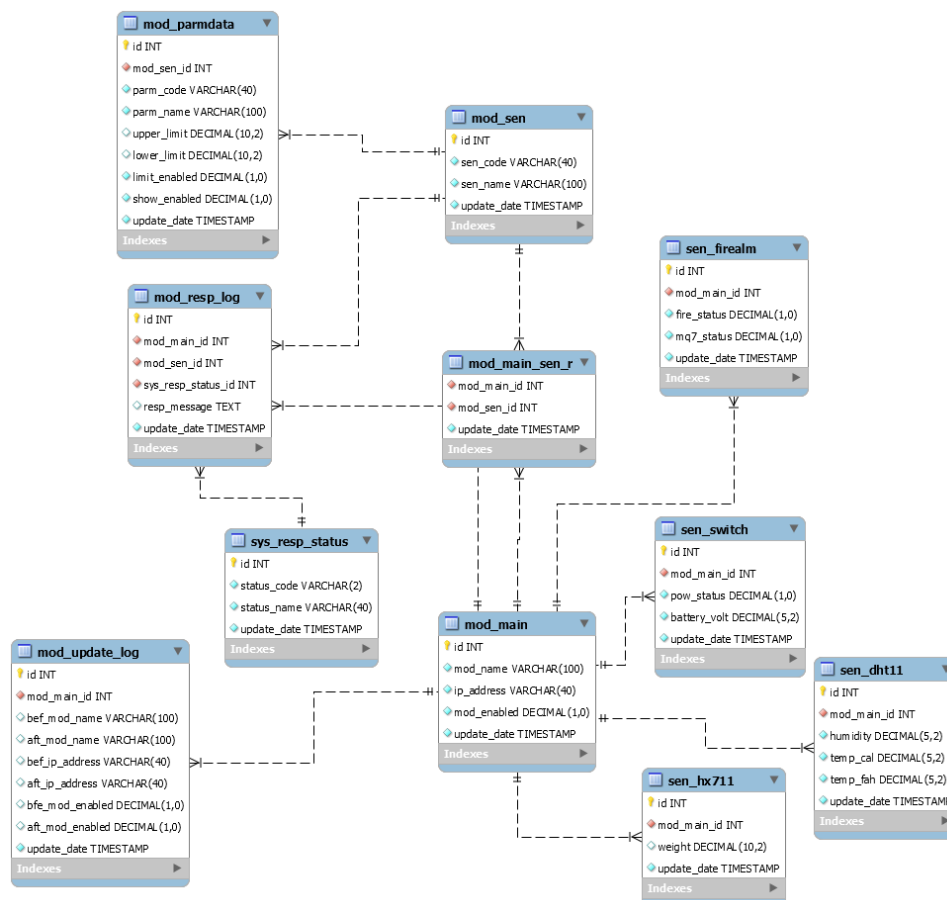


圖 3.4 資料庫關聯圖



## ■ 即時監控資訊系統

本論文設計的即時監控資訊系統，系統架構如圖 3.5 所示。系統部屬在 Apache Tomcat 上，利用 Java 程式語言開發，Maven 管理系統套件；資料庫層使用 Hibernate[20] 用以將關聯式資料庫中的資料表映射至 Java 中以物件的方式呈現；邏輯層則是用前面章節介紹的 Spring MVC 框架串起顯示層與資料層；顯示層利用 SiteMesh[21] 當作網頁整合框架並使用 jQuery[22] 產生網頁的各種互動式效果，搭配採用 Bootstrap[23] 來設計 RWD 模式；而智慧型裝置以 Android 開發其應用程式採用前面章節介紹的 WebView 設計並連接網頁，讓網頁在呈現上不管是電腦、手機或平板上皆可看到一致的效果，工廠管理人員在使用上可以依據當下情況使用不同裝置連線系統。

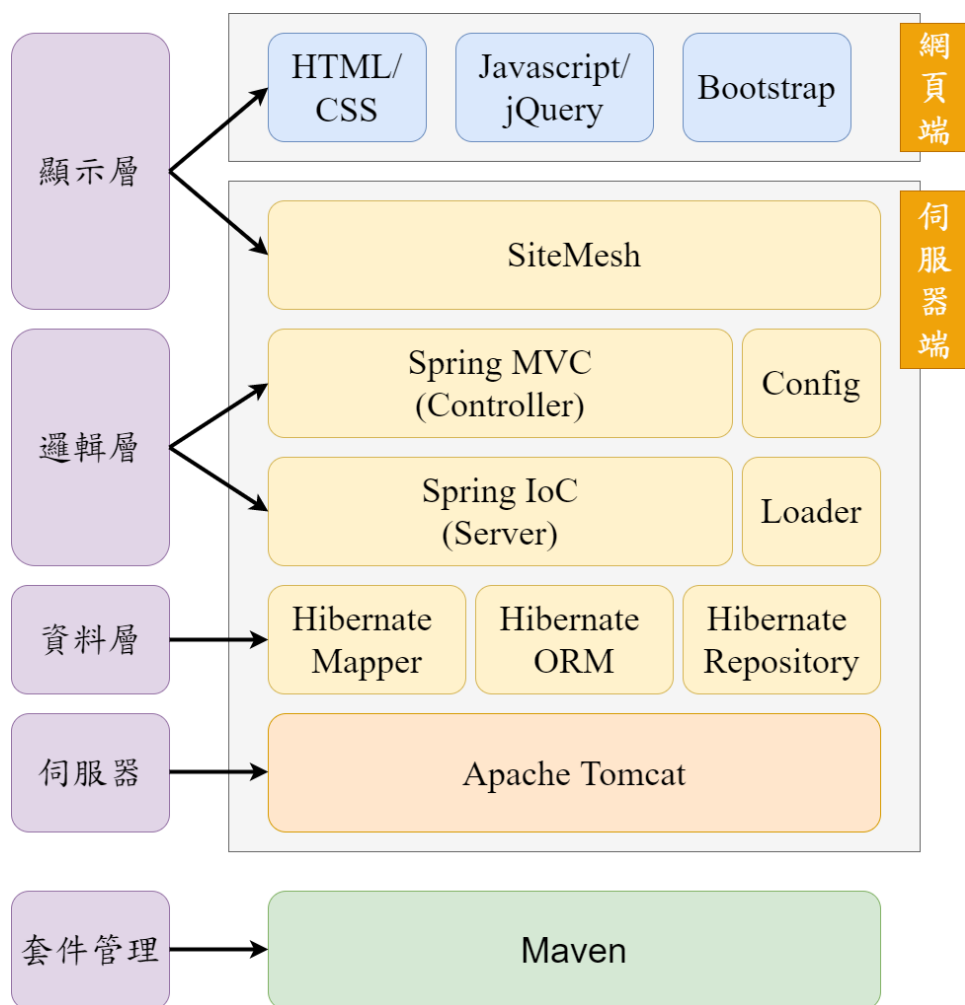


圖 3.5 即時監控資訊系統架構圖

## 3.2 系統功能規劃

本論文為提供工廠管理人員操作使用的工廠即時監控資訊系統，功能規劃如圖 3.6 所示，總共可以分為四大部分：儀表板、感應裝置、感應模組及報表資料，將於以下章節說明。

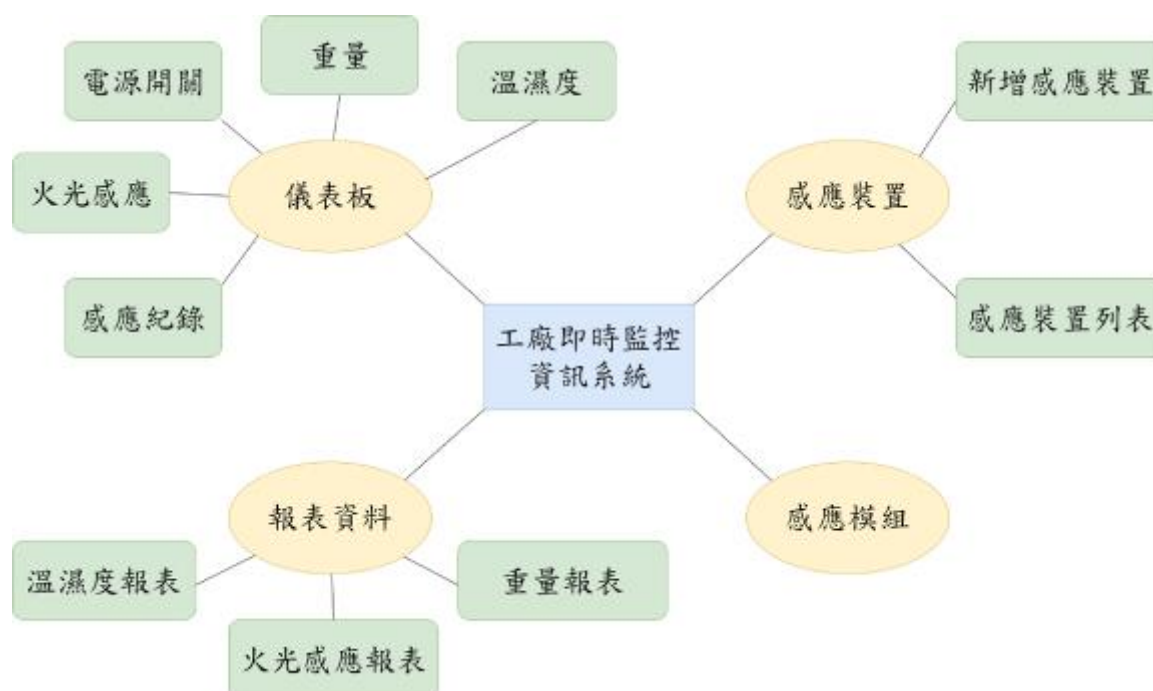


圖 3.6 系統功能規劃架構圖

### 3.2.1 儀表板功能

主要是呈現當下工廠的運作情況，工廠管理人員可以透過該頁面清楚的知道每個工具機運作的情形以及有設置感應裝置的廠區位置狀況。頁面會呈現目前有連接的感應模組資訊，往後若工廠管理人員有需要還可以再增加不同的感應模組上去。另外還會顯示感應裝置目前運作的狀況，讓工廠管理人員可以即時知道工具機是否運作正常，若發現異常即可前往確認，用以增加工具機的運作效率。

儀表板頁面會依據感應模組設定的上下限警戒值，即時於頁面做出對應的變化，流

程如圖 3.7 所示。

1. 系統每經過一段時間從資料庫讀取感應裝置對應的感應模組最新資料。
2. 判斷該筆資料是否超過或低於感應模組設定的上下限警戒值。
3. 若超過或低於代表為異常狀況會於頁面中將該數值以特別顏色顯示，提醒工廠管理人員注意。
4. 若沒有判斷出超過或低於警戒值則於儀表板頁面正常顯示。

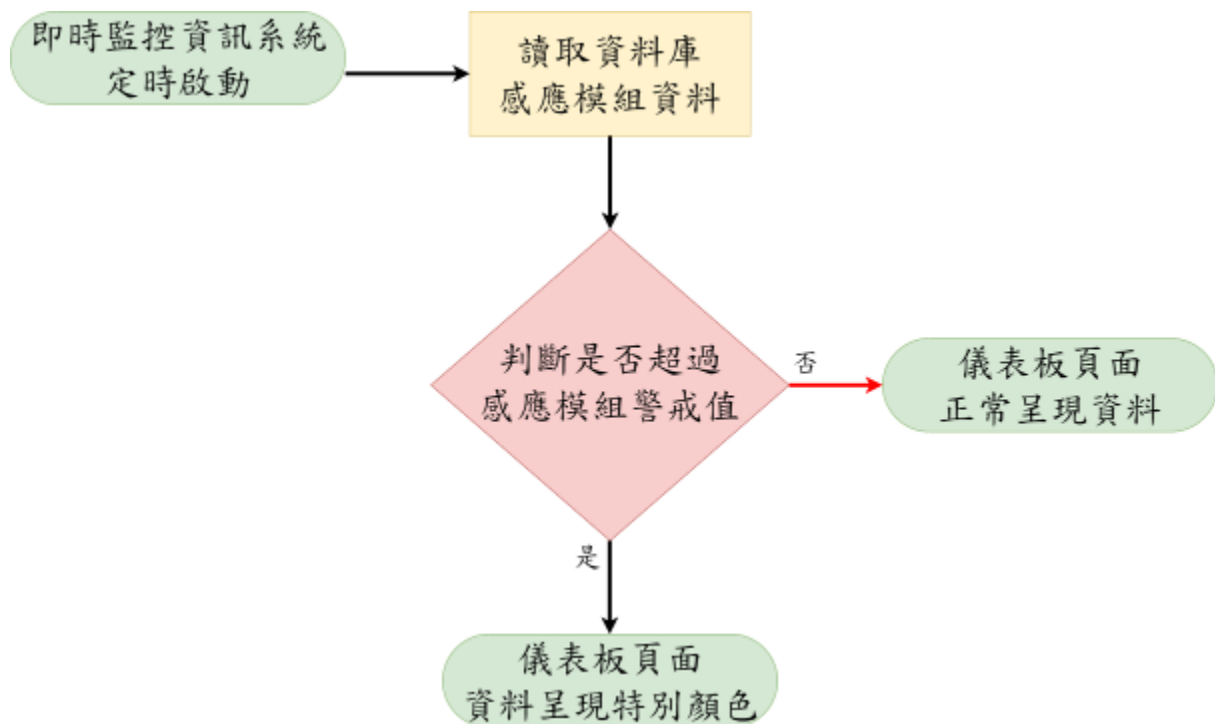


圖 3.7 儀表板資料流程圖

### 3.2.2 感應裝置功能

該功能可分為新增感應裝置及感應裝置列表功能，主要是讓工廠管理人員依照工廠的特性新增或修改感應裝置及設置對應的感應模組，其操作流程如圖 3.8 所示。

1. 工廠管理人員判斷要新增感應裝置或是修改舊有的感應裝置。
2. 若是新增則到新增感應裝置功能頁面：輸入感應裝置 IP、名稱、設定所對應的感應模組及是否啟用。
3. 若是修改舊有感應裝置內容：先到感應裝置列表功能中找到要修改的感應裝置，點選後輸入要修改的內容。
4. 新增或修改完成後，系統會檢核欄位是否正常以及 IP 是否重複，若檢核有誤，則顯示錯誤訊息提醒工廠管理人員；若檢核無誤則將感應裝置資料寫入資料庫。

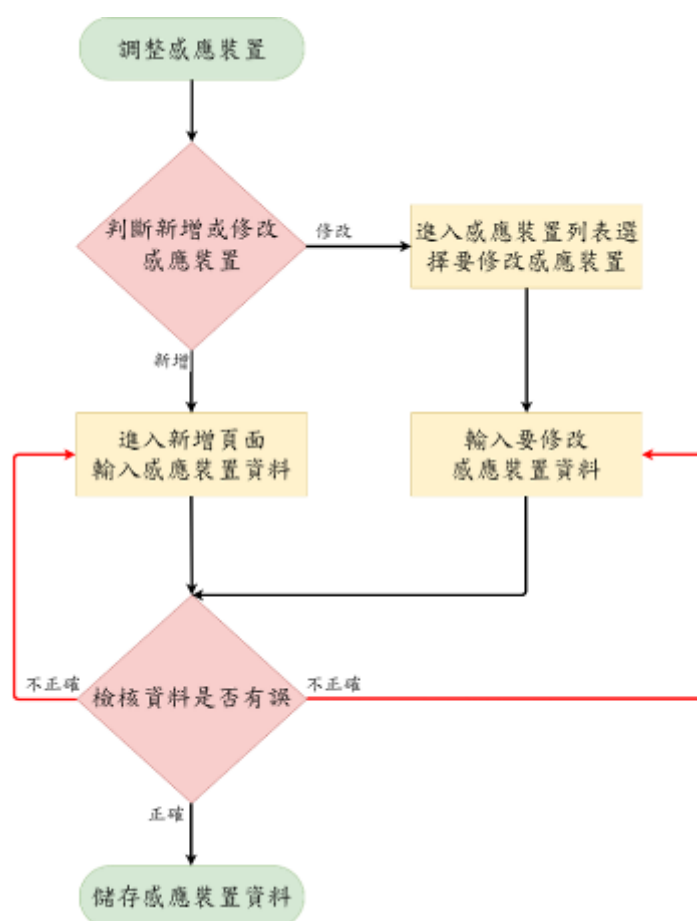


圖 3.8 調整感應裝置流程圖

### 3.2.3 感應模組功能

該功能主要是用來設定感應模組的警戒值，工廠管理人員可以依照工廠的需求自由調整每個感應模組的警戒值，系統會顯示該感應模組的參數資料供工廠管理人員作調整，像是溫濕度模組就可以調整溫度及濕度的上下限數值。

### 3.2.4 報表資料功能

報表資料功能主要提供工廠管理人員了解工廠過往運作情形，可分為溫濕度報表、重量報表及火災感應報表，可以透過該功能清楚的知道即時監控資訊系統收集的感應模組歷史資料變化如何。若工廠管理人員無法時刻監控系統，也可以透過報表功能了解每一台工具機的運作情況並加以分析找出可能的問題。

## 第四章 系統開發成果

本論文目的在於開發與建置「工廠即時監控資訊系統」，讓工廠管理人員可以透過電腦或是智慧型裝置連接系統網頁即可遠端即時監控工廠內的工具機運作情形。

### 4.1 工廠即時監控資訊系統

#### 4.1.1 主畫面

主畫面如圖 4.1 所示，呈現當前工廠內所安裝的感應裝置感應到的各項資料，工廠管理人員可以透過該功能即時了解工廠內的運作情形，達到即時遠端監控的目的。

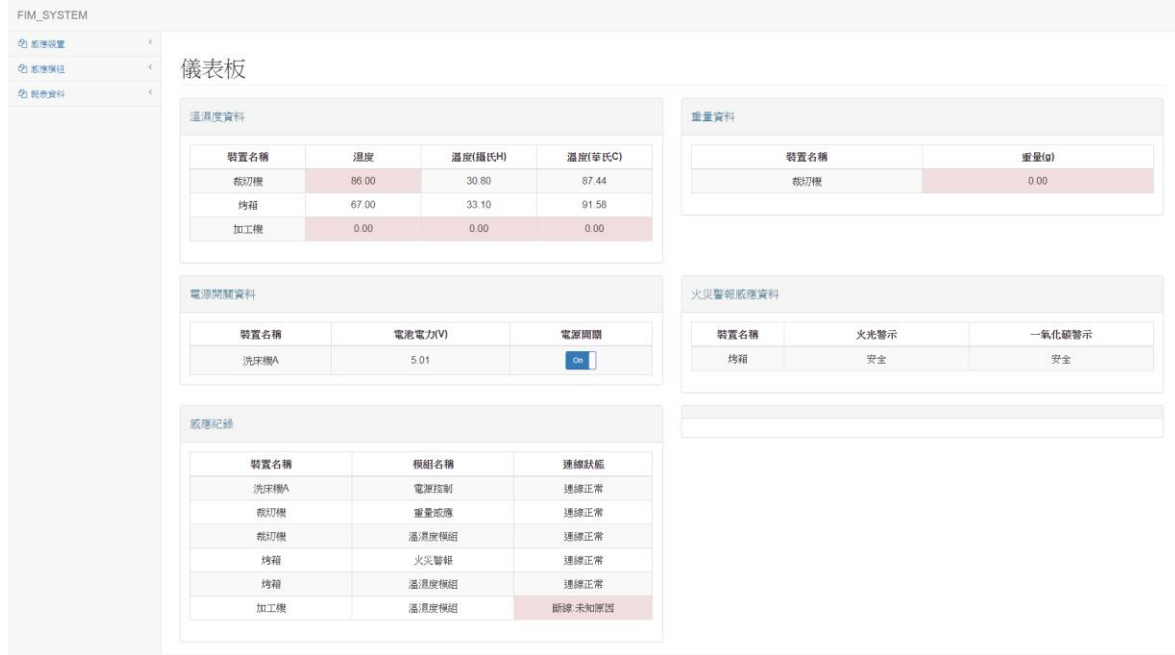


圖 4.1 主畫面

1. 溫濕度資料：呈現當前有連線的感應裝置並顯示目前的濕度及溫度資料，其中裁切機的濕度欄位底色有異常，是由於濕度已超過設定的警戒值用以提醒管理人員。

2. 重量資料：顯示當前連線到裁切機旁感應裝置收集到的重量資料。
3. 電源開關資料：顯示當前利用電源開關模組連接的工具機，上頭會顯示當前工具機是否啟動且工廠管理人員也可以透過電源開關按鈕遠端操作工具機啟動或關閉。
4. 火災警報感應資料：用以顯示當前工具機是否有火災發生，當出現火光反應或是一氧化碳超出警示值，畫面會即時呈現讓工廠管理人員可以快速反應處理問題。
5. 感應紀錄：用以顯示目前有連接的感應裝置連線狀況，當有感應裝置連線異常會於畫面上即時顯示通知工廠管理人員前往處理，如加工機顯示斷線。

### 4.1.2 新增及修改感應裝置

圖 4.2 (a) 顯示當前所有感應裝置列表並顯示相關資訊，若想要修改 IP 位置、調整感應模組或是調整是否啟用可以點選修改進入修改畫面，如圖 4.2 (b) 所示；若想要增加新的感應裝置可以點選左列的新增感應裝置功能。

FIM\_SYSTEM

感應裝置

新增感應裝置

感應裝置列表

感應模組

報表資料

感應裝置列表

Show 10 entries

Search:

編號	裝置名稱	ip 地址	是否啟用	感應模組	
1	洗床機A	192.168.43.102	啟用	電源控制	修改
2	裁切機	192.168.43.100	啟用	重量感應, 溫濕度模組	修改
3	烤箱	192.168.43.101	啟用	火災警報, 溫濕度模組	修改
4	加工機	192.168.50.222	啟用	溫濕度模組	修改

Showing 1 to 4 of 4 entries

Previous 1 Next

(a)

FIM\_SYSTEM

感應裝置

感應模組

報表資料

修改感應裝置

名稱 裁切機

ip 地址 192.168.43.100

是否啟用 On

感應模組 ☒ 溫濕度模組 ☒ 重量感應 ☐ 電源控制 ☐ 火災警報

儲存

(b)

圖 4.2 調整感應裝置畫面

### 4.1.3 感應模組警戒值設定

在圖 4.3 (a) 中顯示當前系統所可以選擇的感應模組，並且能針對每個感應模組設定各個參數的警戒值。點選感應模組左邊的加號會顯示當前的參數如圖 4.3 (b) 所示。當要修改警戒值時直接點選數字就會跳出小視窗讓管理人員直接修改，如圖 4.3 (c) 所示，不會再導到其它頁面修改操作上較便利。另外最右邊欄位可以針對該參數設定是否啟用警示，若取消，則該參數超過警戒值於主頁面不會改變顏色，給予較好操作彈性。

FIM\_SYSTEM

系統設置  
系統保護  
**模組警戒值**  
系統資料

模組警戒值

	模組名稱	模組代號
+	溫濕度模組	dht11
+	重量感應	hx711
+	電源控制	switch
+	火災警報	fireAim

(a)

FIM\_SYSTEM

系統設置  
系統保護  
系統資料

模組警戒值

		模組名稱	模組代號		
		溫濕度模組	dht11		
參數名稱	參數代號	上限警戒值	下限警戒值	是否啟用警示	
濕度	humidity	80	25	<input checked="" type="checkbox"/>	
溫度(攝氏C)	temp_cel	40	20	<input checked="" type="checkbox"/>	
溫度(華氏C)	temp_fah	100	0	<input checked="" type="checkbox"/>	
+	重量感應		hx711		
+	電源控制		switch		
+	火災警報		fireAim		

(b)

FIM\_SYSTEM

系統設置  
系統保護  
系統資料

模組警戒值

		模組名稱	模組代號		
		溫濕度	dht11		
參數名稱	參數代號	上限警戒值	下限警戒值	是否啟用警示	
濕度	humidity	70	25	<input checked="" type="checkbox"/>	
溫度(攝氏C)	temp_cel	40	20	<input checked="" type="checkbox"/>	
溫度(華氏C)	temp_fah	100	0	<input checked="" type="checkbox"/>	
+	重量感應		hx711		
+	電源控制		switch		
+	火災警報		fireAim		

(c)

圖 4.3 警戒值設定畫面



## 4.1.4 報表資料使用

報表資料功能會依據設定的感應模組產生功能，如圖 4.4 (a) 以溫濕度報表為例。

一進入該功能會顯示要查詢的工具機及區間，可以利用小日曆點選要查詢的期間如圖 4.4 (b)，按下查詢即會以折線圖顯示該區間的溫濕度變化量，如圖 4.4 (c) 所示。

(a)

(b)



(c)

圖 4.4 溫濕度報表資料畫面

## 4.2 感應裝置設置

本論文共設置三種不同的感應裝置來控制感應模組讀取工具機上的資料，進行成果驗證。

### 4.2.1 溫濕度及重量感應模組

如圖 4.5 (a) 利用接線盒將感應裝置放置在裡面，而溫濕度及重量感應模組則放置在接線盒外頭用以感應工具機資料並利用杜邦線連接到裡面的感應裝置，行動電源固定在接線盒下方用以驅動感應裝置運作。圖 4.5 (b) 為接線盒內部構造圖，若感應裝置有問題方便工廠管理人員進行維護。



(a)



(b)

圖 4.5 溫濕度及重量感應模組

### 4.2.2 溫濕度及火災警報模組

圖 4.6 的感應模組為溫濕度感應、火光感應及一氧化碳檢測，設置方式也是放在接線盒裡面並於下方固定行動電源用以驅動。

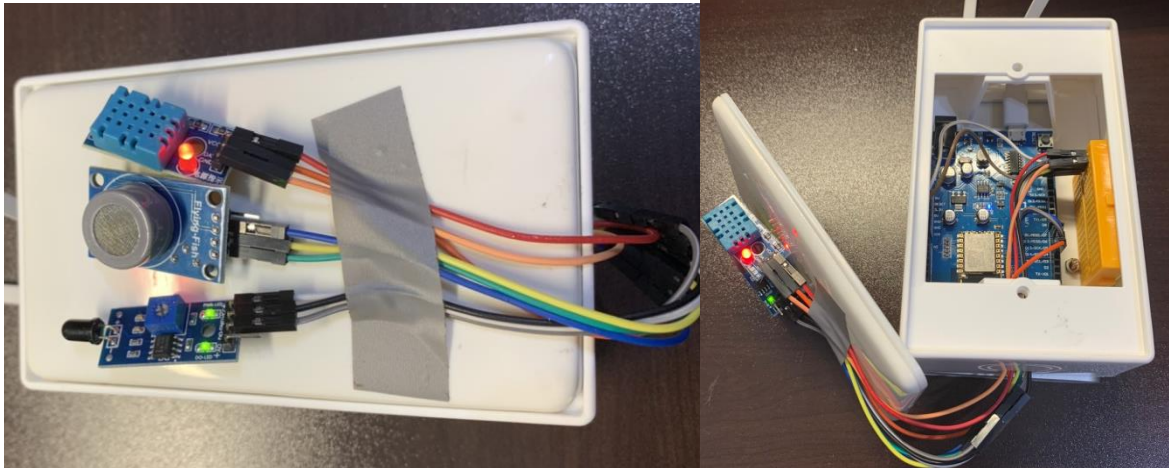


圖 4.6 溫濕度及火災警報模組

### 4.2.3 電源控制模組

圖 4.7 主要是用來控制工具機的電源開關，連接一條電源線插頭並提供插座供工具機插上並於接線盒外側邊固定電池用以驅動感應裝置控制電源。



圖 4.7 電源控制模組

## 4.3 工廠實驗

於工廠中進行模擬實驗，在工廠內依據不同的工具機及環境設置不同的感應裝置並透過系統觀察是否可以讀取到資料。

### 4.3.1 裁切機

在裁切機旁放置溫濕度及重量感應模組，工廠管理人員可以透過手機即時監控，如圖4.8 (a) 所示，用以監控裁切機運作時的溫濕度情形及隨時記錄裁切完的物品重量；如圖4.8 (b) 當裁切機在運作結束後，可以將裁切完物品放置重量感應模組上，可自動記錄物品重量。

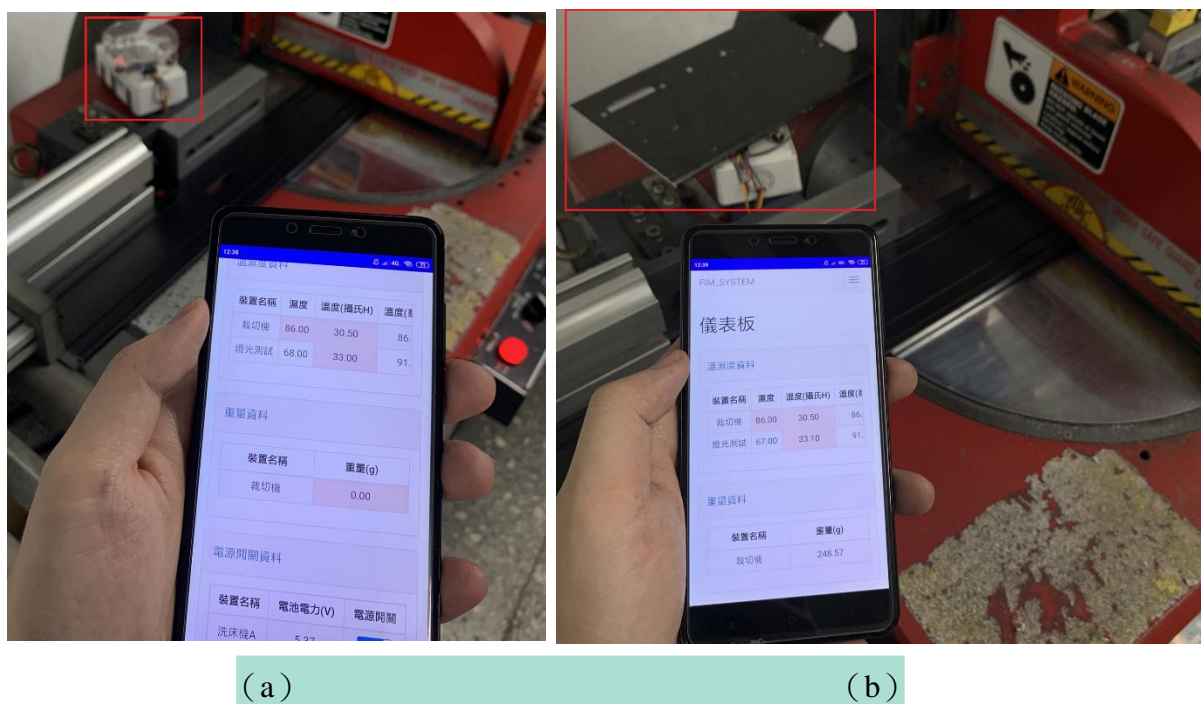
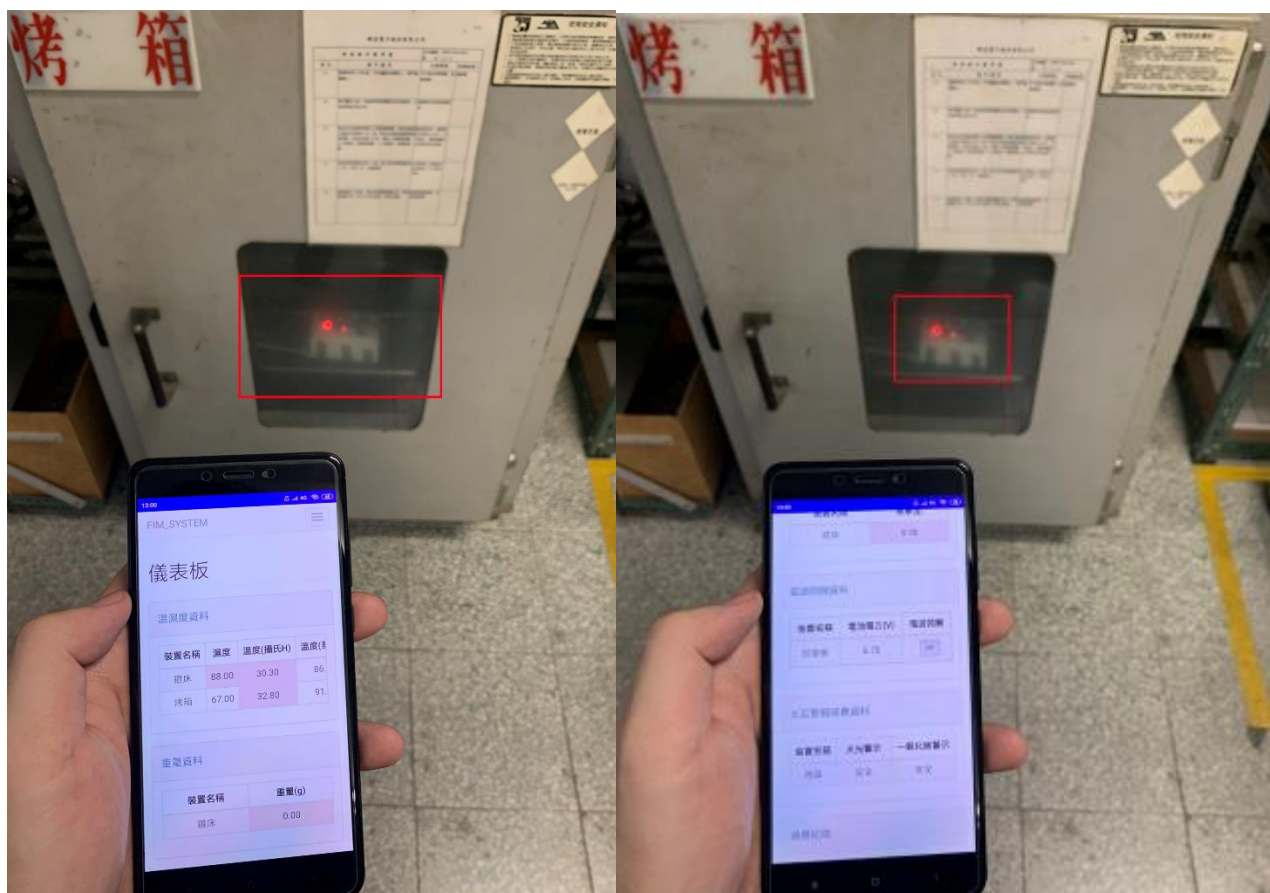


圖 4.8 裁切機實驗畫面



### 4.3.2 烤箱

由於烤箱是以天然氣驅動，因此於其內部放置溫濕度、火光及一氧化碳感應模組用以監控。如圖4.9(a)當烤箱在運作的過程中工廠管理人員可以隨時監控溫濕度變化量；圖4.9(b)中也可以監控是否有火光及一氧化碳是否有外洩的情況；當溫度超過警戒值儀表板畫面會將該欄位以特別顏色顯示，令工廠管理人員可以一目瞭然看到，並前往工具機旁查看是否有問題。



(a)

(b)

圖 4.9 烤箱實驗畫面

### 4.3.3 洗床機

洗床機旁放置電源控制模組並將洗床機的電源線接上該模組的插座，而電源控制模組的電源線再接上電源用以控制洗床機。工廠管理人員可以隨時利用即時監控系統控制洗床機是否運作。如圖4.10 (a) 所示，目前洗床機為打開狀態；當按下電源開關按鈕後洗床機即會關機，電源開關也會顯示Off，如圖4.10 (b) 所示。



圖 4.10 洗床機實驗畫面

#### 4.3.4 廠區任意位置

除了將感應裝置放在工具機旁，還可以擺放在廠區任意位置。如圖4.11所示，原先該位置已有設置LED螢幕供管理人員了解廠區溫濕度。在此放置溫濕度感應模組進行監控，讓管理人員即便在看不到LED螢幕的情況，也能透過系統監控工廠內的溫濕度。



圖 4.11 廠區任意位置實驗畫面

如圖4.12該區為燈具測試位置，用以實驗燈具運作情形及壽命，實驗期間往往耗時許久，因此可以於該處擺放溫濕度及火災警報，用以監控實驗期間內的溫濕度及監控火災反應，使得管理人員不用密集巡邏，只要透過系統即可隨時觀察運作情形。



圖 4.12 燈具測試區實驗畫面

## 4.4 即時監控傳輸實驗

為了因應當有大量感應裝置連接時能達到即時監控的目的，將實驗當隨著感應裝置的增加回應時間的變化量。

### 4.4.1 實驗方式

即時監控資訊系統及Client API架設在Tomcat Server上，感應裝置只連接溫濕度感應模組，並連到同一個無線網路。實驗當感應裝置連接分別為10、20、50台情況下，測試同時從即時監控資訊系統發起請求到感應裝置回傳資料再加上Client API接收資料後寫入資料庫所需要的總時間，分別測試10、20、50及100次，每次間隔5秒計算其平均回應時間，圖4.13為連接多台感應裝置實驗畫面。

儀表板

溫濕度資料			
裝置名稱	濕度	溫度(攝氏H)	溫度(華氏C)
test01	65.00	28.30	82.94
test02	65.00	28.30	82.94
test03	65.00	28.30	82.94
test04	59.00	29.10	84.38
test05	59.00	29.10	84.38
test06	65.00	28.30	82.94
test07	65.00	28.30	82.94
test08	59.00	29.10	84.38
test09	65.00	28.30	82.94
test10	59.00	29.10	84.38
test11	59.00	29.10	84.38
test12	65.00	28.30	82.94
test13	65.00	28.30	82.94
test14	59.00	29.10	84.38
test15	65.00	28.30	82.94
test16	59.00	29.10	84.38
test17	59.00	29.10	84.38
test18	65.00	28.30	82.94
test19	65.00	28.30	82.94
test20	59.00	29.10	84.38
test21	59.00	29.10	84.38
test22	65.00	28.30	82.94
test23	65.00	28.30	82.94
test24	59.00	29.10	84.38
test25	65.00	28.30	82.94
test26	59.00	29.20	84.56
test27	59.00	29.20	84.56
test28	65.00	28.50	83.30
test29	65.00	28.50	83.30
test30	59.00	29.20	84.56
test31	59.00	29.20	84.56
test32	65.00	28.50	83.30
test33	65.00	28.50	83.30
test34	59.00	29.20	84.56
test35	65.00	28.60	83.48
test36	59.00	29.20	84.56
test37	59.00	29.20	84.56
test38	65.00	28.60	83.48
test39	65.00	28.60	83.48
test40	59.00	29.20	84.56
test41	59.00	29.20	84.56
test42	65.00	28.60	83.48
test43	65.00	28.40	83.12
test44	59.00	29.10	84.38
test45	65.00	28.40	83.12
test46	59.00	29.10	84.38
test47	59.00	29.10	84.38
test48	65.00	28.40	83.12
test49	65.00	28.40	83.12
test50	59.00	29.10	84.38

圖 4.13 連接多台感應裝置實驗畫面



## 4.4.2 實驗結果

本次實驗結果如表4.1所示，在一開始實驗只有連接5台感應裝置的情況下，平均回應時間落在0.3秒~0.4秒之間；連接10台感應裝置後回應時間提升兩倍落在0.58秒~0.89秒之間，時間比起5台的情況下是翻倍增加；連接20台感應裝置後，回應時間落在1.1秒~1.6秒之間已經突破到1秒以上，在可接受範圍內；最後連接50台感應裝置情況下，回應時間攀升到2.9秒~4.2秒之間，尚在每次請求間隔5秒內。

如圖4.14為感應裝置回應時間比較，隨著連接感應裝置的數量增加回應時間會以等比例的形式增加。

表 4.1 實驗結果

感應裝置數量 測試次數	5台感應裝置	10台感應裝置	20台感應裝置	50台感應裝置
執行10次平均時間	0.401s	0.85s	1.549s	3.331s
執行20次平均時間	0.316s	0.818s	1.616s	3.317s
執行50次平均時間	0.499s	0.895s	1.275s	2.922s
執行100次平均時間	0.351s	0.585s	1.179s	4.221s

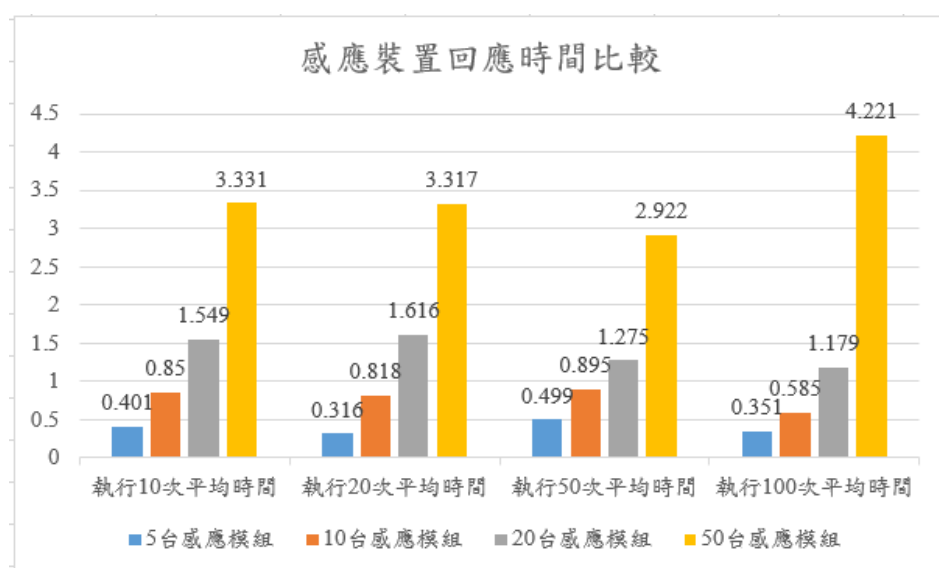


圖 4.14 感應裝置回應時間

## 第五章 討論

表5.1為原先傳統工廠運作方式及使用工廠即時監控資訊系統後的比較表，以下就兩種做法進行說明。

### ■ 傳統工廠

1. 若工具機本身無LED螢幕可以顯示資料，只能由工廠管理人員定期巡邏廠區並依靠經驗判斷工具機的運作情形並將之記錄下來。
2. 有些工具機其本身有LED螢幕可以顯示當前運作情形或是廠區內有設置LED螢幕可以查看當前工廠內的溫濕度情形，但可能會因廠區較大或是分佈不同樓層使得無法隨時監控整個廠區的狀況。

### ■ 工廠即時監控系統

1. 利用在工具機外設置感應裝置或廠區各處放置感應裝置用以自動收集整個廠區的運作情形，與採用人工巡邏及檢測的方式相比較為客觀，準確度也大幅度增加。
2. 感應裝置使用無線網路將資料傳送到系統後，並儲存在資料庫中進行保存。
3. 使用電腦或是智慧型裝置可以隨時監控廠區內的情形，其資料收集時間在連接50台感應裝置的情況下仍可以在5秒內回應，比起傳統工廠利用人力定時巡邏的方式更加有效率及快速。

表 5.1 傳統工廠與工廠即時監控資訊系統做法比較表

	傳統工廠	工廠即時監控資訊系統
收集及保存資料	人工方式檢測及記錄工具機運作情形	利用外接感應裝置自動收集資料並儲存在資料庫中
資料傳輸	人工作業	利用無線網路傳輸資料
遠端即時監控	無資料呈現方式或是僅依靠 LED 螢幕顯示	1. 透過電腦或智慧型裝置監控 2. 連接 50 台感應裝置下回應時間仍在 5 秒內

## 第六章 結論與未來方向

### 6.1 結論

本論文設計與實作一套「工廠即時監控資訊系統」，利用在工具機及廠區四周設置感應裝置自動收集各項資料，並採用無線網路傳送到系統內透過資料庫進行保存。用以改善傳統工廠因工具機老舊或是此前無完整的規劃，造成仍採用人工作業的方式收集及記錄廠區的運作情形。

除此之外本系統更可以透過電腦或是智慧型裝置即時監控工廠內的運作情形，令工廠管理人員即便不在現場，也可以即時了解整個工廠的運作情況。系統也提供報表查詢作業可以顯示工廠內過往的運作情形，工廠管理人員能夠依照歷史資料作分析，可以更為精準的進行決策判斷。依照過往資料搭配目前的即時運作狀態調整工廠產能找出最有效率的方式，實現靈活生產的目標，為智慧化工廠目的作準備。

### 6.2 未來方向

本論文所提出的工廠即時資訊監控系統已可改善許多傳統工廠遇到的問題，但仍有許多可以加強與優化的地方，未來可以朝向以下幾個方向努力：

1. 本論文目前只實作幾種感應模組，未來可以依照不同工廠的需求持續增加感應模組種類。
2. 以中小型工廠規模來說目前連接 50 台感應裝置情況下，回應時間在 5 秒內是夠用。未來若持續增加感應裝置數量其回應時間也會持續增加，若是超出可等待時間，就要優化回應時間。
3. 資訊科技發展帶來方便隨之而來便是資訊安全的隱憂，使得工廠內的資料可能被人更換或竊取。未來可以透過使用更加安全的傳輸協定或是利用識別碼辨別感應裝置的合法性增加工廠資料傳輸的安全性。

## 參考文獻

- [1] J. Schlechtendahl, M. Keinert, F. Kretschmer, A. Lechler, and A. Verl, “Making Existing Production Systems Industry 4.0-Ready,” *Production Engineering*, vol. 9, no. 1, pp. 143-148, 2015.
- [2] F. Griffiths and M. Ooi, “The Fourth Industrial Revolution - Industry 4.0 and IoT,” *IEEE Instrumentation & Measurement Magazine*, vol. 78, pp. 218-235, Feb. 2019.
- [3] D. Raposo, A. Rodrigues, S. Sinche, J. S. Silva, and F. Boavida, “Industrial IoT Monitoring: Technologies and Architecture Proposal,” *Sensors*, vol. 18, no. 10, pp. 3658, Oct. 2018.
- [4] Dandan Zhang, Zhiqiang Wei, and Yongquan Yang, “Research on Lightweight MVC Framework based on Spring MVC and Mybatis,” in *Proc. Sixth International Symposium on Computational Intelligence and Design (ISCID) 2013*, Oct. 2013, ISBN 978-1-4799-0906-3.
- [5] Leff Avraham, Rayfield, and James T, “Web-Application Development Using the Model/View/Controller design Pattern,” in *Proc IEEE Enterprise Distributed Object Computing Conference*, pp. 118-127, Sept. 2001.
- [6] Ethan Marcotte, *Responsive Web Design*, A List Apart, May 2010.
- [7] Amy Schade, *Responsive Web Design (RWD) and User Experience*, Nielson Norman Group, May 2014.
- [8] “WebView”[Online]. Available: <https://developer.android.com/reference/android/webkit/WebView>. [Accessed: 6-Aug-2020]
- [9] “Arduino”[Online]. Available: <https://blog.jmaker.com.tw/arduino-tutorials-1/>. [Accessed: 6-Aug-2020]

- [10] “WeMos D1 WiFi”[Online]. Available: <https://www.instructables.com/id/Programming-the-WeMos-Using-Arduino-SoftwareIDE/>. [Accessed: 6-Aug-2020]
- [11] “dht11”[Online]. Available: <https://iscixin.github.io/freakhq/DHT11/>. [Accessed: 6-Aug-2020]
- [12] “Hx711”[Online]. Available: [https://blog.jmaker.com.tw/hx711\\_set/](https://blog.jmaker.com.tw/hx711_set/). [Accessed: 6-Aug-2020]
- [13] “FileAlame”[Online]. Available: <https://circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing>. [Accessed: 6-Aug-2020]
- [14] “Mq-7”[Online]. Available: <http://www.learningaboutelectronics.com/Articles/MQ-7-carbon-monoxide-sensor-circuit-with-arduino.php>. [Accessed: 6-Aug-2020]
- [15] “Arduino ide”[Online]. Available: <https://www.arduino.cc/>. [Accessed: 6-Aug-2020]
- [16] “Api”[Online]. Available: <https://pydoing.blogspot.com/2011/05/java-understanding-of-java-api.html>. [Accessed: 6-Aug-2020]
- [17] “Eclipse”[Online]. Available: <https://www.eclipse.org/>. [Accessed: 6-Aug-2020]
- [18] “Apache Tomcat”[Online]. Available: <http://tomcat.apache.org/>. [Accessed: 6-Aug-2020]
- [19] “Mysql”[Online]. Available: <http://www.mysql.com/documentation>. [Accessed: 6-Aug-2020]
- [20] “Hibernate”[Online]. Available: [https://www.tutorialspoint.com/hibernate/hibernate\\_architecture.htm](https://www.tutorialspoint.com/hibernate/hibernate_architecture.htm). [Accessed: 6-Aug-2020]
- [21] “SiteMesh”[Online]. Available: <https://sites.google.com/site/wintestsite1/my-blog/sitemesh-an-introduction>. [Accessed: 6-Aug-2020]
- [22] “jQuery”[Online]. Available: <https://jquery.com/>. [Accessed: 6-Aug-2020]
- [23] “Bootstrap”[Online]. Available: <https://getbootstrap.com/>. [Accessed: 6-Aug-2020]