**中 國 文 化 大 學**

**資 訊 工 程 學 系**

**資 訊 系 統 專 題**

**無人商店**

**學 生：陳 宥 睿**

**黃 姵 瑄**

**邱 郁 涵**

**指導教授：洪 敏 雄**

**中 華 民 國 106年 5 月**

**無人商店**

**專題學生：陳宥睿、黃姵瑄、邱郁涵**

**指導教授：洪敏雄 博士**

**中國文化大學 資訊工程學系**

**摘要**

隨著資訊與網路科技的快速發展，電子商務迅速崛起，消費型態轉變，使得消費管道變得更加多元，實體通路變得不再是主流，舊的商業模式不再適用，零售物流業將面臨新的改革。我們的專題想要研究新的零售模式——無人商店。無人商店的風潮襲捲全球，新革命性的零售商店所依靠進技術分為三大類：AI人工智慧、RFID無線射頻、自動販賣機。身為中、美兩大電商龍頭，亞馬遜與阿里巴巴皆已投入無人商店的市場，如亞馬遜的「Amazon Go」、阿里巴巴的「淘咖啡」，而兩大電商所掌握的關鍵技術，如何偵測、蒐集、及分析消費者移動軌跡及消費行為為我們專題研究方向，研究各店商所掌握的優勢技術，且找出我們自身能力所及的領域來進行簡單的實作。

**關鍵詞**：物聯網、RFID、新營運模式

**指導教授\_\_\_\_\_\_\_\_\_\_(簽名)**

**Unmaned Store**

**Student: You-Rui Chen, Pei-Xuan Huang, Yu-Han Chiu**

**Advisor: Prof. Min-Xiong Hong**

**Department of Computer Science and Information Engineering**

**Chinese Culture University**

# Abstract

With the rapid development of information and network technology, the rapid rise of e-commerce and the shift in consumption patterns have made the consumption pipeline more diversified, the physical channel has become less mainstream, and the old business model is no longer applicable. Retail logistics industry will face new reforms, and our topic is to study new retail models — unmanned stores. The trend of unmanned stores is sweeping the world, and the new revolutionary retail stores rely on technology to fall into three categories: AI artificial intelligence, RFID radios, and vending machines. As the two major e-commerce leaders in China and the United States, Amazon and Alibaba have invested in the market of unmanned stores, such as Amazon's "Amazon Go" and Alibaba's "Tao Coffee", and the key technologies mastered by the two e-commerce companies — How to detect, collect, and analyze consumer movement trajectories and consumer behaviors as our special research direction, Study the superior technologies mastered by each store and find out the areas in which we can work to make simple implementation.

**Keywords:** Cloud Computing, Internet of Things, RFID, New Business Model

# 研究動機與目的

## 研究動機

由於課程安排，課堂偶而會有業師來進行演講。在某一次的演講中，業師提到無人商店系統 「Amazon GO」，引起我們的興趣，想要深入探討這個系統，了解它是如何運作。因此，我們也想要做出一套屬於自己的無人商店系統。

## 研究目的

本研究透過日前所學及所知技術來達成目標，為了達到上述之目標，將使用三大軟印體如下：

1. 使用RFID

透過無線電訊號識別特定目標並讀寫相關數據，並將數據內容紀錄在我們的資料庫

1. APP開發

使用 Android Studio開發APP，讓使用者在我們的APP進行查帳、付款等目的

1. 網站開發

使用ASP.NET的MVC架構進行網站開發，動態連接資料庫，顯示使用者所購買的品項且記錄是否結帳之紀錄。

# 文獻探討

基於第1章所述之研究動機與目的，我們對目前已有之網路技術進行研究，而目前當代的軟硬體解釋名詞如下：

## 物聯網

物聯網是傳統電信、網際網路等資訊乘載體，讓所有能行使獨立功能的裝置設備實現互相聯通的網路，需要依賴大量的技術得以成形，例如將裝置連線到網際網路應用程式介面（API）技術、大數據管理技術、預測分析技術、AI和機器學習技術、雲端以及無線射頻（RFID）技術…等。

透過物聯網，讓服務端對裝置、機器、及人員進行控管，也可以對家庭設備和汽機車進行遙控、搜尋位置、防止物品被盜等，類似自動化操控系統。並透過蒐集上述的數據，聚合成大數據，透過大量的數據分析，整合成有用的資訊，比如：如何設計車禍率低的道路、未來的都市該怎麼更新發展、及災害預測、犯罪防治、流行病控管…等。

物聯網的最終理想是讓現實世界走向數位化，讓分散的資訊聚合，搭建物與物的橋樑。物聯網主要應用領域包括：運輸物流領域、工業製造、健康醫療等。我們的專題主要使用到物聯網技術之一的RFID來進行實作。

## RFID

無線射頻辨識系統（Radio Frequency Identification，簡稱RFID）是一種「非接觸式」的自動識別技術。RFID由兩種裝置組成，電子標籤（Tag）和讀取裝置（Reader）。工作原理如下，將電子標籤內嵌在商品中，透過無線電波的發射，將所需要資料傳到讀取器，利用射頻訊號通過空間耦合，交變磁場或電磁場原理，實現無接觸傳遞，在由後端應用系統，進行資訊判讀、運用，而達到辨識目的。

RFID的功能類似IC晶片卡或智慧卡，但記憶體容量更大，外觀尺寸也可以縮小許多，不僅可以用來取代固定式或接觸式識別標籤、商品條碼（Bar code）。更可以將RFID與電腦、資料庫系統、防火牆…等技術作結合，進而應用於倉儲管理、門禁管制，追蹤控制、交通運輸監控、生產自動化、電子票證、自動化識別…等各領域中。因此，RFID相關技術被廣泛應用於各領域已是指日可待。

RFID的電子標籤可分為主動式及被動式電子標籤，主動式的電子標籤較貴，因此我們採用被動式電子標籤，然而電子標籤又有分為可讀寫及唯讀的性質，我們採用可讀寫電子標籤作為追蹤管理，藉由讀取器傳送指令來改寫或清除更新標籤內容（Tag），所以讀取器也算是寫卡機。

## Android

Android作業系統採用Linux核心，處理手機硬體相關的工作，一般情況下，Android開發員不需了解Linux相關技術。Android開發應用程式使用Java語言，只需使用Android提供的API就可控制裝置的硬體設備。一個Android應用程式除了有Java程式碼，還須依照Android應用程式框架規定撰寫設定檔及提供應用程式所需要的資源。

Android應用程式採用Java程式設計語言，Java是一種發展20年以上的程式設計技術，他擁有完整的開發技術與眾多的開發人員，擁有許多其他程式設計技術沒有的優點。Android在發表後，採用最多Java開發人員使用的Eclipse當開發工具，搭配Android為Eclipse設計的Android Development Tools（ADT）。隨著Android版本不斷更新，為了提供更完整與靈活的開發方式，Android應用程式開發工具從原來的Eclipse改為「IntelliJ IDEA」，之後又整合所有的Android開發套件，在2014年12月發表「Android Studio v1.0」。

雖然現在仍可以使用Eclipse搭配ADT的開發環境，但Eclipse的ADT已不再維護，而且Android也在發表Android Studio以後，建議開發人員使用Android Studio來開發應用程式，所以我們的專題也將使用Android Studio作為我們的APP開發。

## ASP.NET MVC

MVC是一種設計樣式，由三大部分組成，分別為Model、View和Controller。Model負責商業邏輯及資料面，View處理UI介面，Controller接受Request請求、指揮協調Model和View及回應結果。這樣的好處是讓關注點分離（SoC, Separation of concerns），降低系統複雜度，提高理解性，讓系統維護及擴展更為便利。

所謂的SoC關注點分離是一種設計原則，將一個電腦程式分為不同的部分或區塊。而每一個部分都有一個關注點，每個關注點內部程式或功能只包含其關心部分，具體化實現SoC的程式稱為模組化系統。

ASP.NET MVC是微軟實踐MVC理論而推出的Web框架，將概念上的MVC樣式轉變成具體可行的框架，針對Model、View和Controller提供實作的功能以及周邊配套與輔助機制。簡而言之，MVC實設計模式，而ASP.NET是支持MVC設計的框架。

# 研究內容、方法與工作項目

## 軟體發展設計

本研究透過國內外無人商店架構之參考，架構出新一套的無人商店系統。

### 簡介

本研究透過國內外學術文獻為例，開發出一套自己的無人商店系統。分為目的、計畫範圍、RFID系統、APP系統、網頁系統。

1. 目的

本研究以實作出一個無人商店為目標

1. 計畫範圍

本研究分為三大部分，RFID、手機APP、網頁系統，由讀取RFID，判斷資料為何，再由手機去登入網頁端去做結帳。

### 系統概述

本文旨在說明「無人商店」之系統目標、系統範圍、系統架構、軟體架構。

1. 系統目標

想要實現更舒適的超商購物環境，不用再一一排隊給人員結帳，只需透過感應等設備就能結帳完畢。讓生活更加便利及友善。

1. 系統範圍

本文件旨在說明本系統範圍，包含系統名稱、系統功能說明、系統架構、系統建構項目需求概述及軟、硬體需求環境。

1. 系統名稱

無人商店

1. 系統功能說明

RFID 端說明

電子標籤（Tag）被讀取端（Reader）讀取，並回傳資料庫找出資料訊息，然後再回傳在網頁端顯示。

手機APP端說明

1. 會員登入

提供使用者登入。

1. 使用介紹

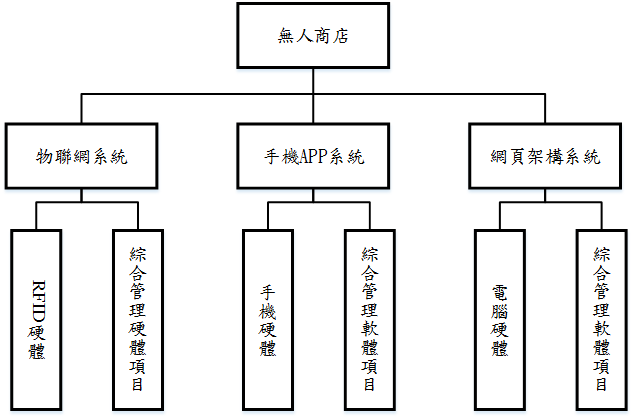
提供使用者簡易的教學，如何透過手機APP對已購買的商品做檢視、查詢等服務。

網頁端說明

在屏幕顯示購買清單項目，供使用者瀏覽或供管理員新增修改刪除項目。

1. 系統架構

本系統分三個子系統開發，分別有手機用戶端APP、物聯網系統、網頁架構系統，三個子系統相互連接形成本系統，如圖3.2。



1. 1. 軟體系統架構

本節將說明手機APP系統、網頁架構系統及物聯網系統所需之驅動、開發環境與相對應之功能。

### 手機APP系統

本節將說明手機APP系統之開發環境、APP所含的功能。

1. 開發環境

本節將闡述本研究之APP系統使用的開發環境與軟體。

1. Android Studio

使用Android Studio軟體，從外觀設計到功能開發都使用這套軟體，如圖 3.3。

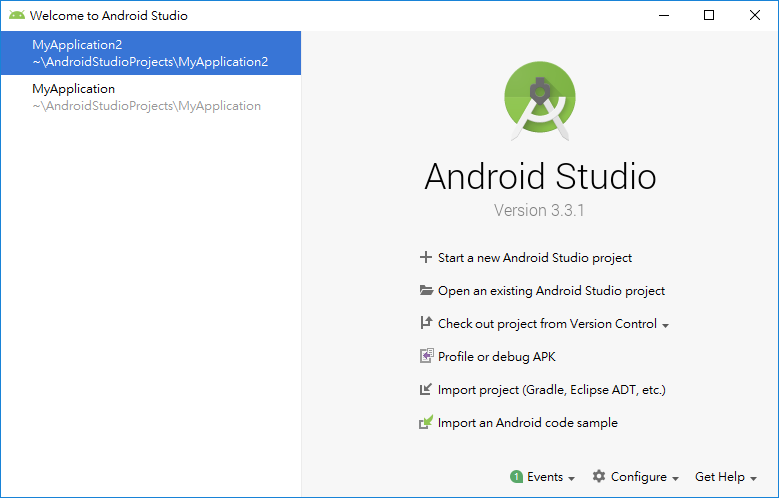


圖 .3 Photoshop介紹

1. APP功能

以下將說明本APP內所有功能與相對應的頁面配置。

1. 登入系統

當使用者輸入帳號密碼時即可登入至網頁系統使用該系統。

1. 查看交易紀錄

使用者可以透過查詢，查看消費的歷史紀錄。

* 1. 硬體系統架構

本節將說明實驗設備、物聯網系統架構規劃，整個系統主要包含感知層、網路層與應用層。

1. **系統硬體設備**

本研究所使用之硬體及實驗設備如下:

* 1. 一組樹莓派( Raspberry pi 3 model B+ )
  2. 一台路由器
  3. 一組影像感測器
  4. 一組煙霧感測器
  5. 一組智慧插座
  6. 一隻Android手機

除了上述設備以外，還有整個實驗模型的配線、待測試的裝置部分尚未決定。

1. **物聯網系統架構**

本節將說明本研究使用之物聯網系統部分架構、部分採用之通訊協定與整體概念。

感知層包含了，各式各樣的感測器、樹苺派主機與行動終端。

網路層包含了，以路由器與樹苺派主機建立之外部網絡伺服器與手機終端使用的行動通訊網路。

應用層包含了，本研究之整套APP結合人工智慧與物聯網開發之E能宅-智慧家庭系統。

。

1. 物聯網系統架構

透過區域劃分，方便管理所有裝設之設備，採用的設備多數採用ZigBee協定，透過內部網路連線至樹苺派主機，在使用路由器將樹苺派指定為固定IP後架設外網伺服器，至此行動用戶端將可透過外網無視空間限制，即時掌握家中環境，如圖3.30。

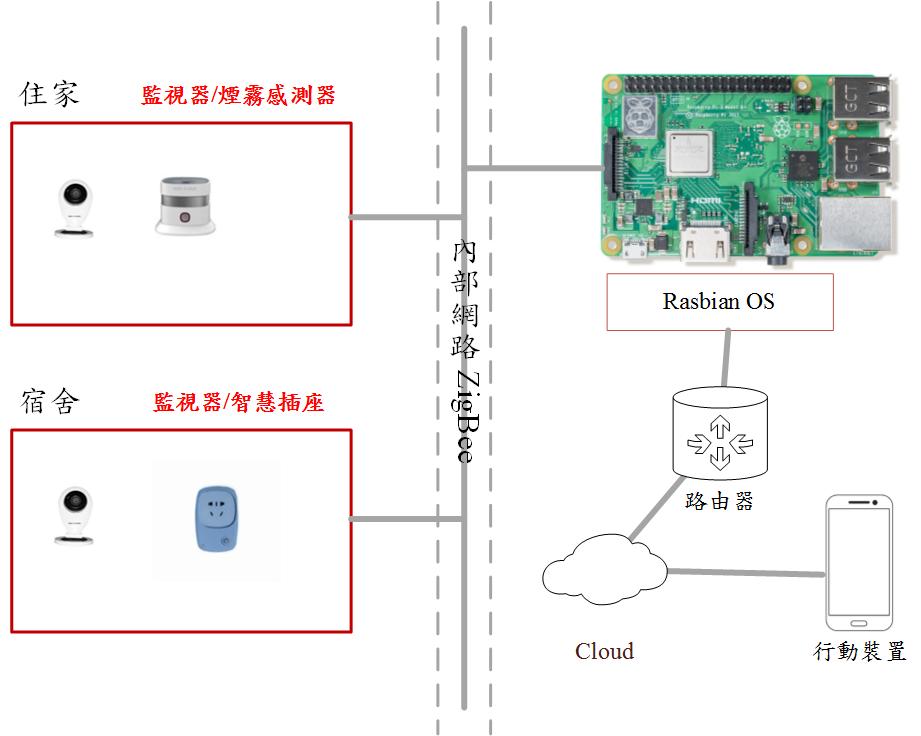


圖 .30物聯網系統概念示意圖

# 初步實驗結果與人力配置

* 1. 研究進度

本節將說明當前研究進度，主要分為APP系統、網頁系統與物聯網系統三大部分，以下將說明各項工作進度與成果呈現。

### 手機APP部分

目前已確定使用Android Studio與JAVA撰寫當開發環境，初步的GUI構思與架構已經擬定完成，下一個目標會全心投入程式碼解決製作上的問題，這部分還需要量時間解決，也會去涉獵關於美編的知識將我們的GUI作的更完美。

1. APP GUI設計

以下為我們大致上的GUI 框架，目前是以VISIO先做出大致上的框架模式，確認功能也按鈕位置等等，日後會再以PhotoShop重製，並套入Flow所提供的模組，並且呈現出一些動態效果。



圖 .1 APP登入頁面

此圖為我們的登入頁面，未來會稍微美化頁面。我們主要是以簡潔功能介面為主，讓使用者方便使用此系統。如圖4.1。



圖 .2 APP主控制頁面

此圖為我們系統首頁，一進畫面可以清楚看到是否為正確用戶名稱以及帳號，方便用戶確認身分。也顯示用儲值金額、回饋點數。結帳需使用右上角的QR code進行付款，如圖4.2。



圖 .3 APP新增區域頁面

此圖為我們的新增區域頁面，再使用者點及新增以後，會來到這個畫面，並讓使用者自定區域名稱，如圖4.3。



圖 .4 APP區域配置頁面

此圖為我們的區域配置頁面，此頁面功能較多，主要功能為語音控制，使用者在此頁面典籍下方語音控制按鈕，便可輕鬆控制該區域中的所有設備，左側放大鏡則是用於查看設備當前狀況，右側垃圾桶為刪除設備按鈕如，圖4.4。



圖 .5 APP裝置設定頁面

此圖為我們的裝置設定頁案，該頁面用於讓使用者能夠得知設備當前狀況，對於不同設備會有不同的資料介面，此處以溫溼度感測器為例子，會顯示對應設備的詳細資訊如，圖4.5。



圖 .6 APP選單頁面

此圖為我們的選單頁面，在首頁頁面時由左向右滑出便會出現該選單，主要呈現出該APP所有設定內容如，圖4.6。

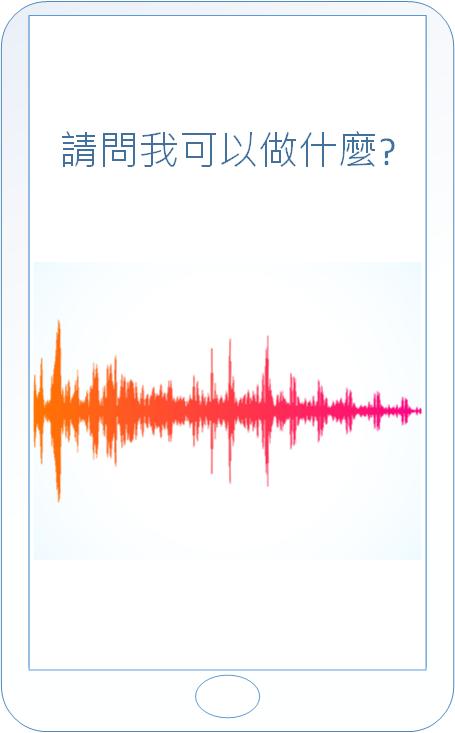


圖 .7 語音辨識頁面

此圖為我們的語音辨識介面，再開啟語音辨識功能時，就會跳到這個頁面，用於聽取使用者指令，圖中波浪會透過Flow給予他波浪效果，如圖4.7。

### 人工智慧部分

當前研究進度為了使Wit.ai提供的套件得以運用，在此先建立了可套用的程式平台Node.js，並試著在此系統中建立起測試用的環境，在下個階段會試著在此程式中運行語音辨識的流程，並與手機APP系統建立起連結，關於這部分的資料還需繼續研究，才可使AI系統更為完整，以下將說明node環境設置流程。

1. node環境設置流程

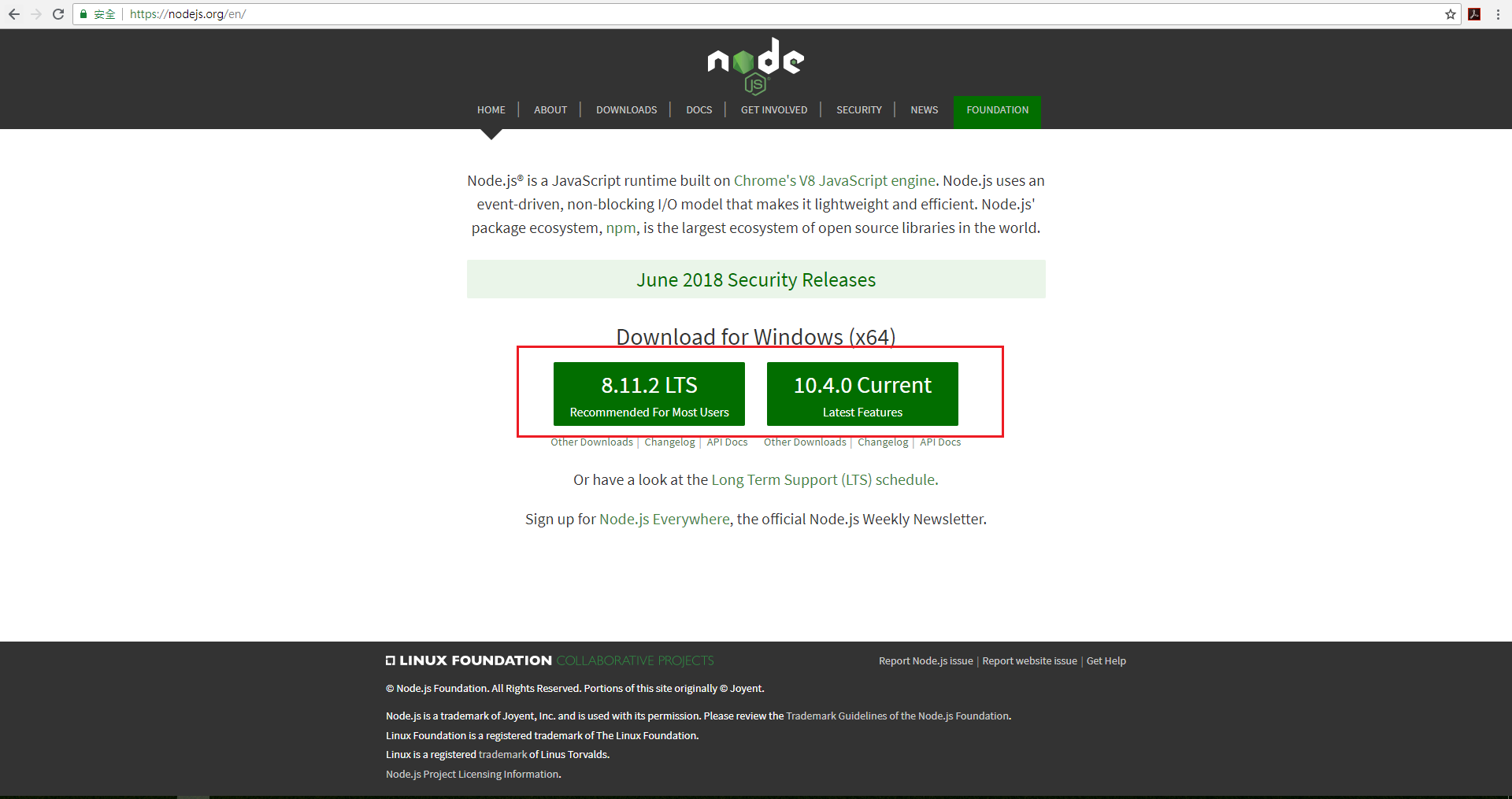


圖 .8 node環境設置圖

步驟1. 前往Node.js網站(https://nodejs.org/en/)並在紅色框框處中選擇你要安裝的版本號，如圖4.8。

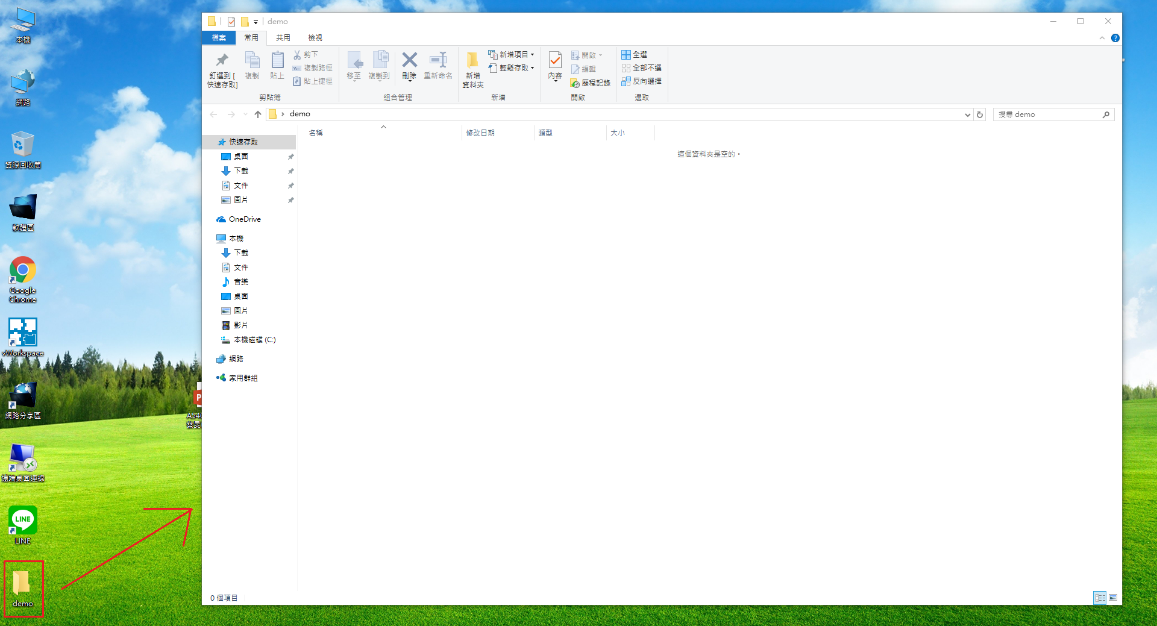


圖 .9 node環境設置圖

步驟2. 在想要的路徑中創建專案資料夾，這裡則是在桌面上建立紅色框框內的新資料夾，名稱叫demo，如圖4.9。

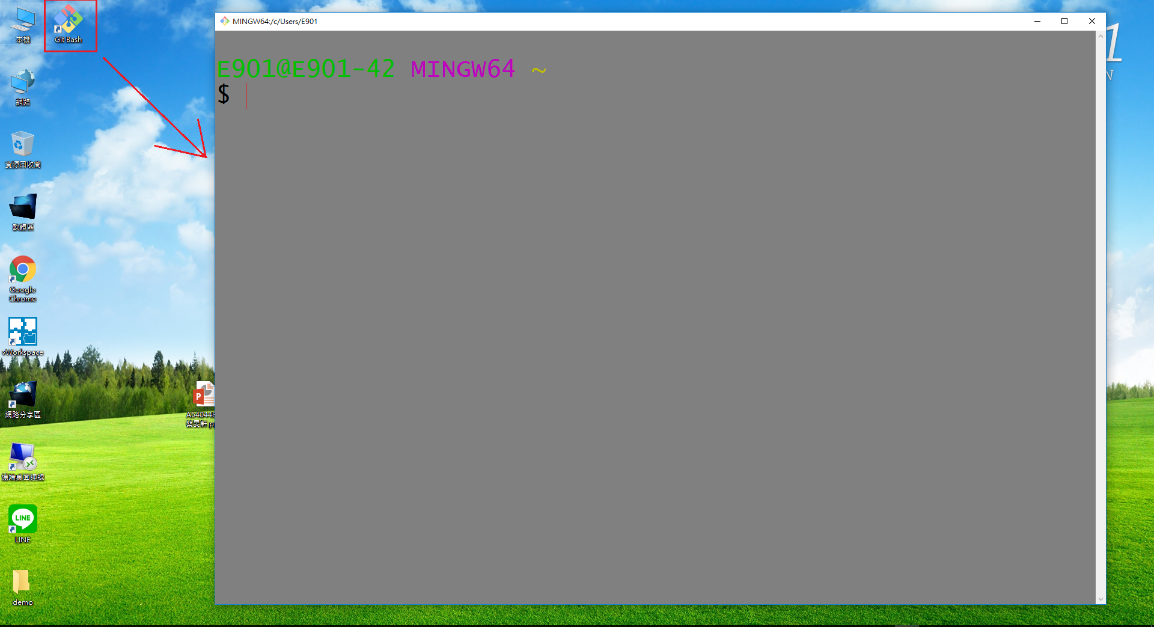


圖 .10 node環境設置圖

步驟3. 開啟Git Bash，如圖4.10。

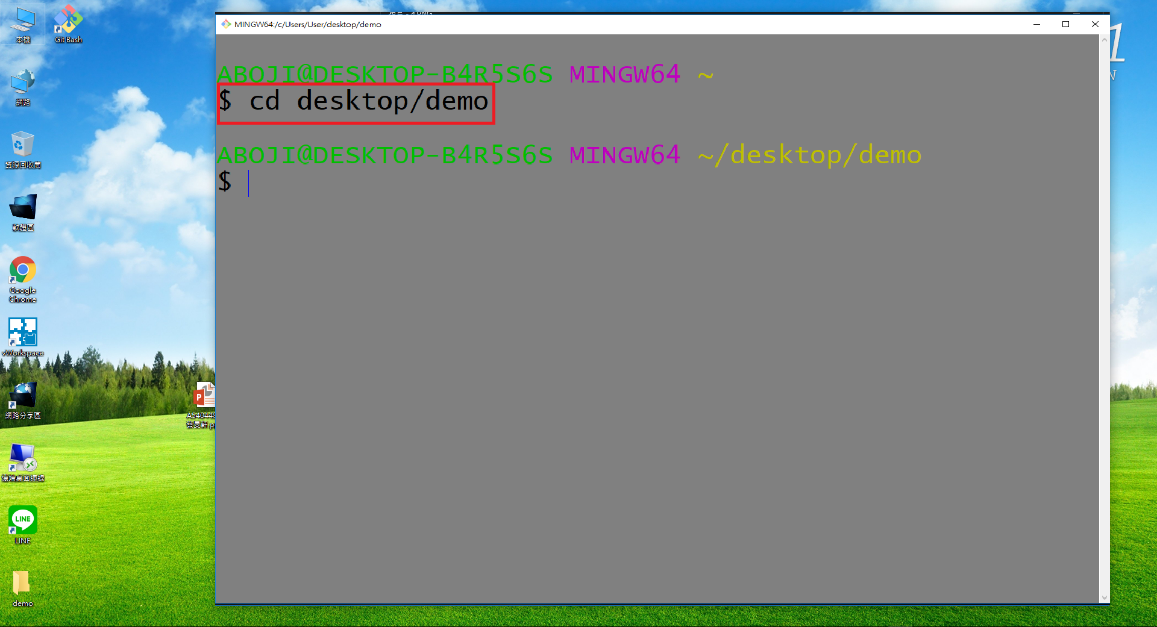


圖 .11 node環境設置圖

步驟4. 在Git Bash中輸入紅色框框內的程式碼後，點擊Enter鍵，即可移至紅色框框中所指示的路徑，而這個路徑即為你的專案資料夾所在的路徑，如圖4.11。

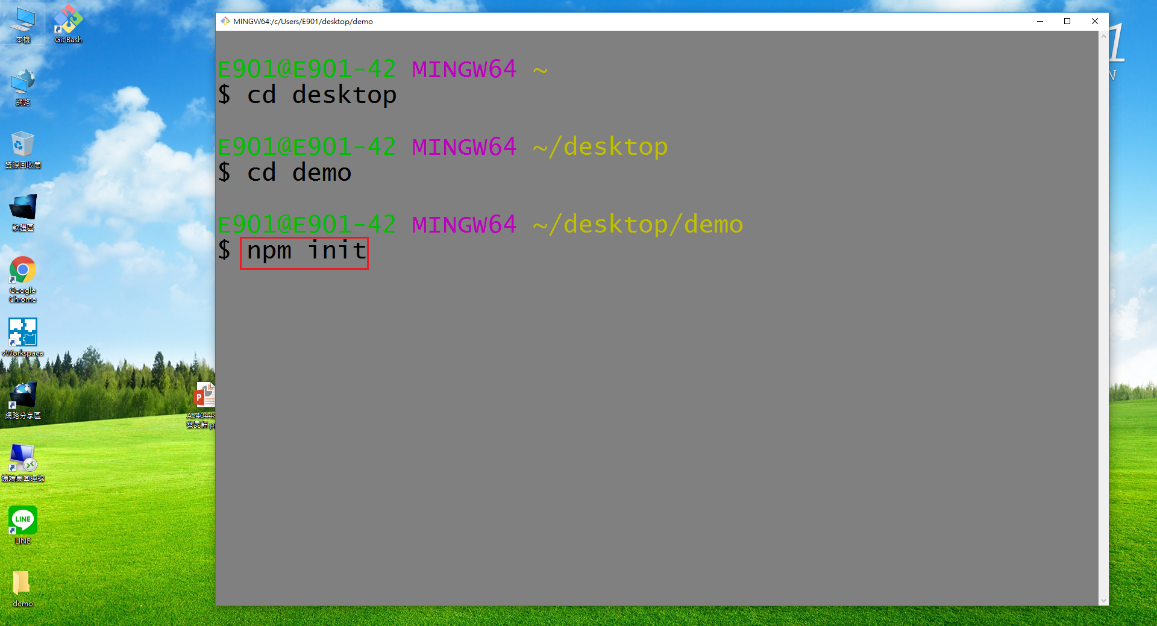


圖 .12 node環境設置圖

步驟5. 在Git Bash上輸入紅色框框中的程式碼後，點擊Enter鍵，即可開始環境的初始化，如圖4.12。

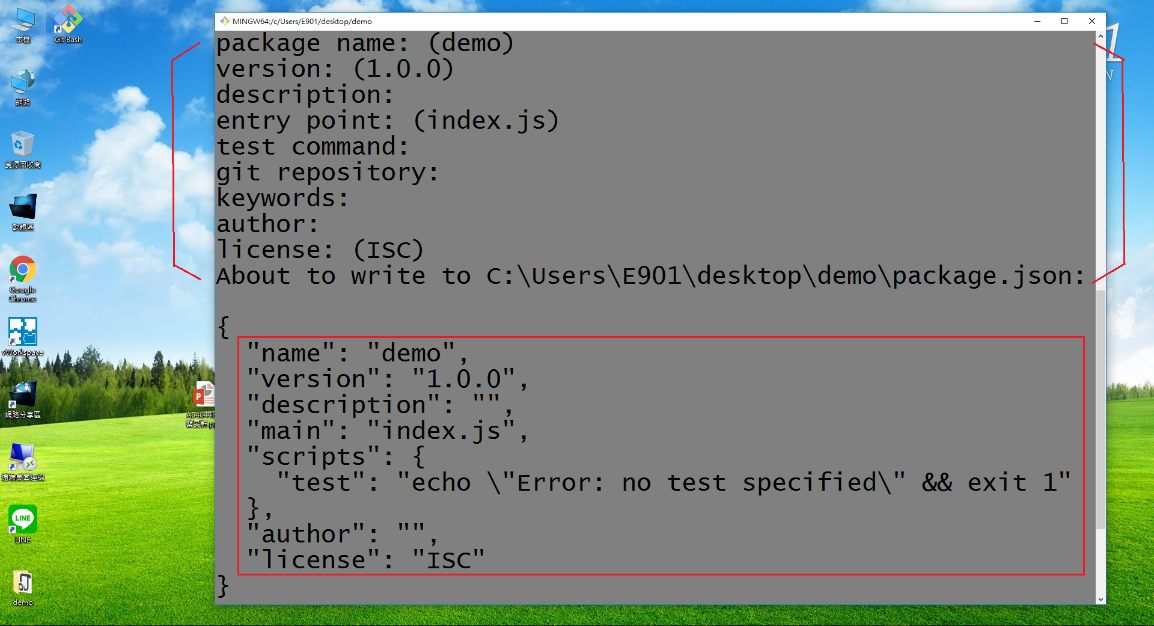


圖 .13 node環境設置圖

步驟6. 在紅色括弧中輸入所要求的資料後，點擊Enter鍵，將在紅色框框的位置顯示出剛剛輸入的資料，確認無誤後，點擊Enter鍵，即可完成環境的初始化，如圖4.13。

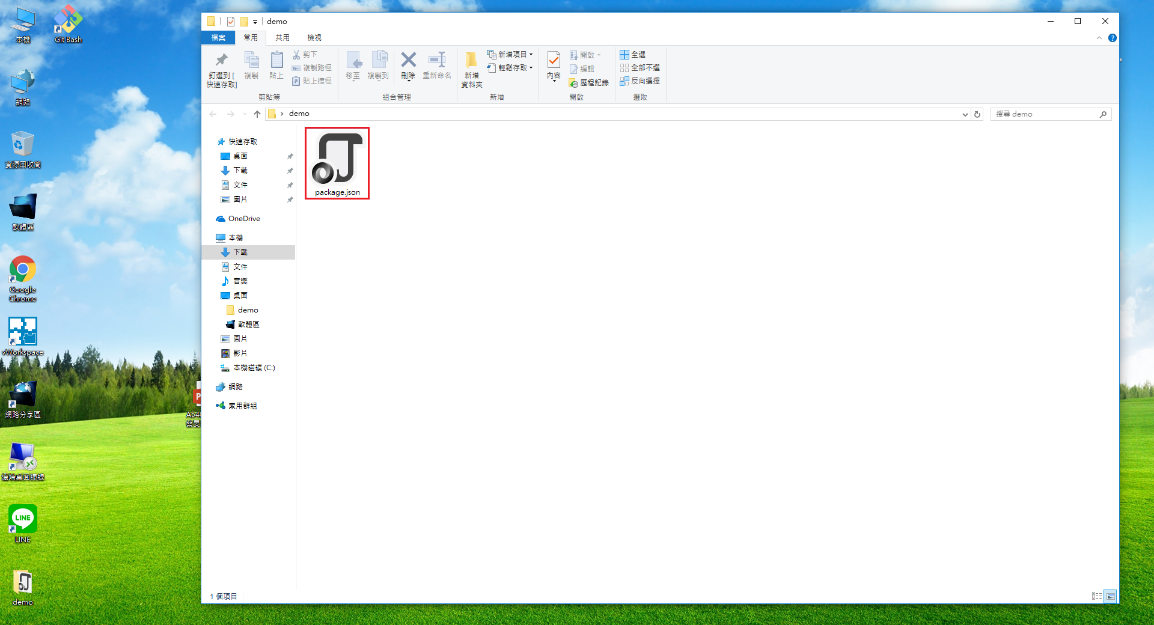


圖 .14 node環境設置圖

步驟7. 完成後，確認專案資料夾內是否建立起紅色框框中的package.json檔，這個檔案將會顯示出已經安裝的套件，如圖4.14。

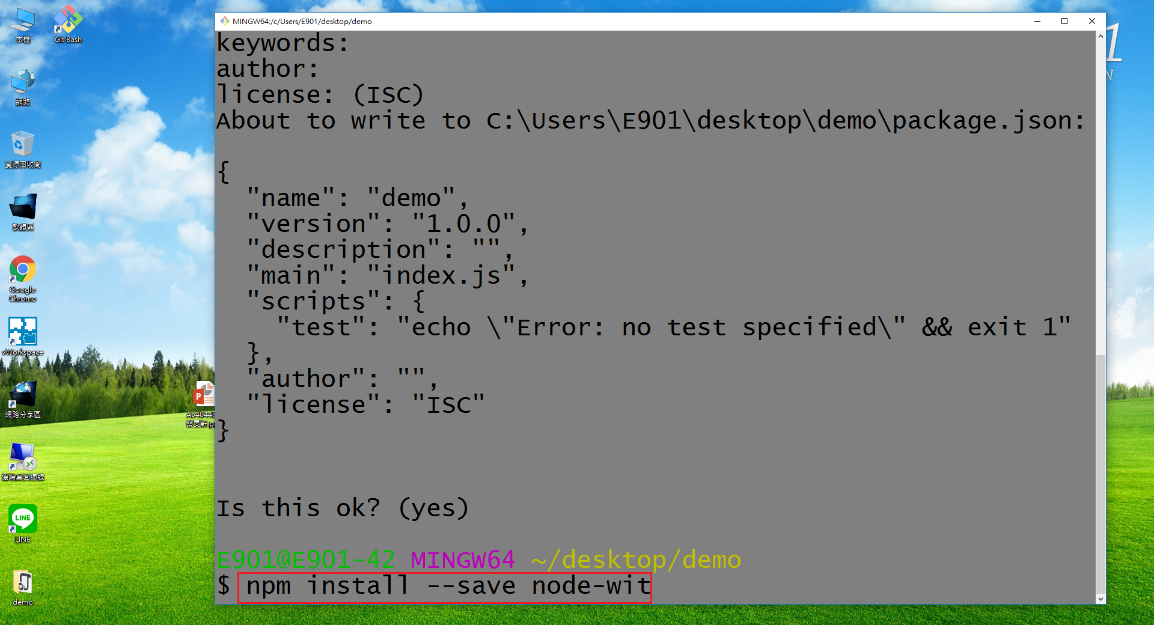


圖 .15 node環境設置圖

步驟8. 開啟Git Bash，輸入紅色框框中的程式碼後，點擊Enter鍵，即可將需要的套件加入至package.json，如圖4.15。

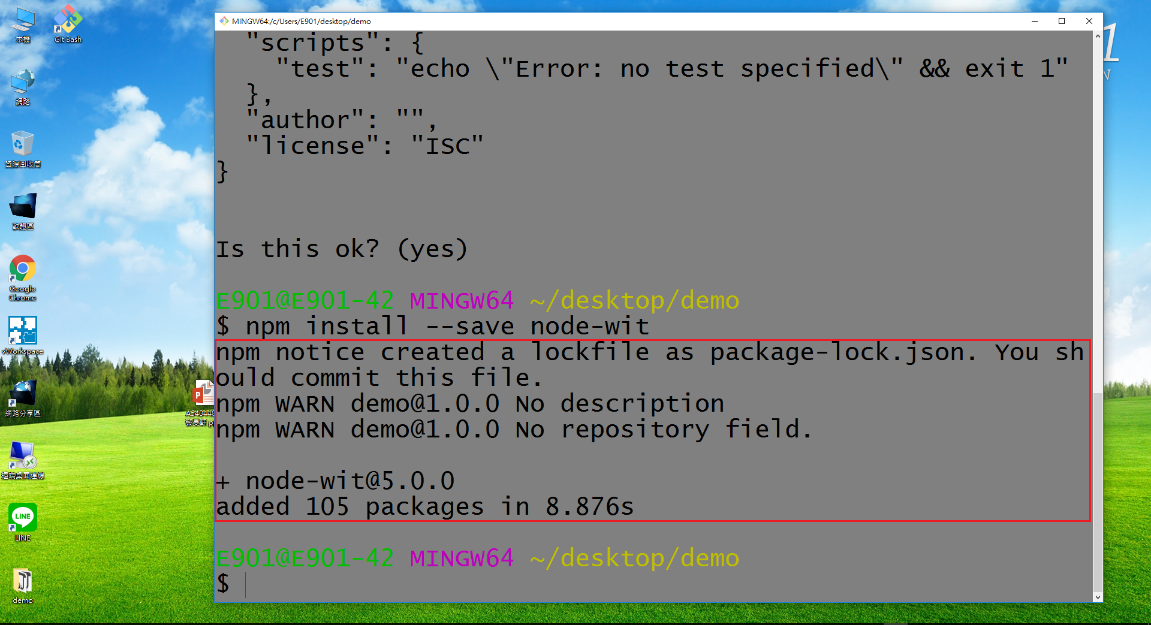


圖 .16 node環境設置圖

步驟9. 若成功加入，則會在Git Bash中顯示出紅色框框內的程式碼，如圖4.16。

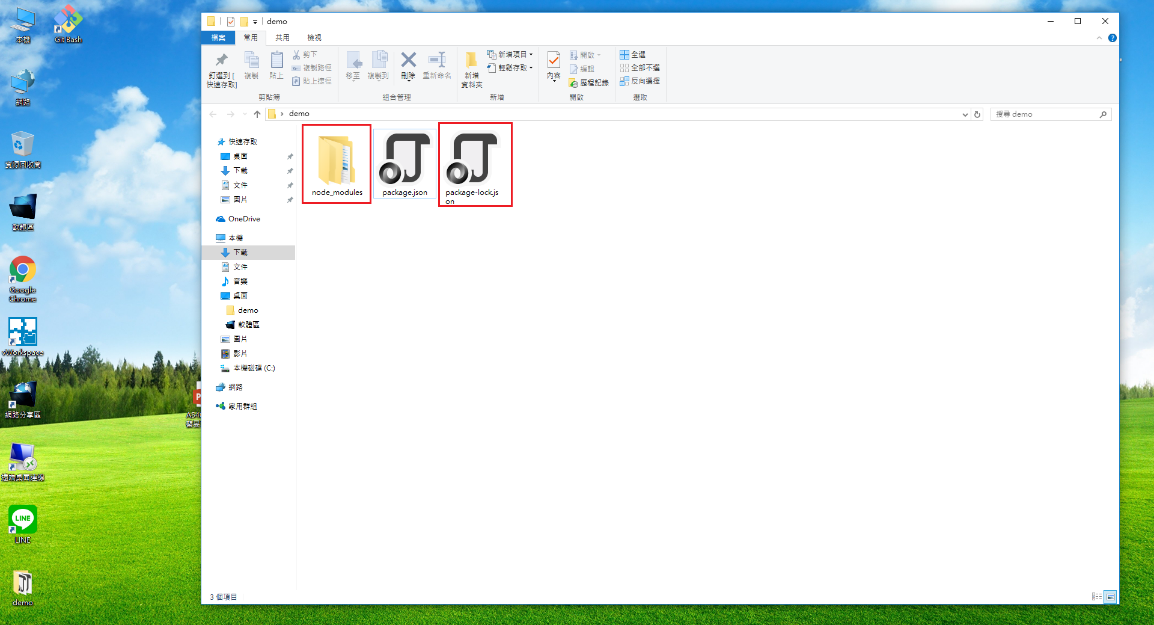


圖 .17 node環境設置圖

步驟10. 成功加入後，專案資料夾內則會建立起紅色框框中的資料夾與json檔，如圖4.17。

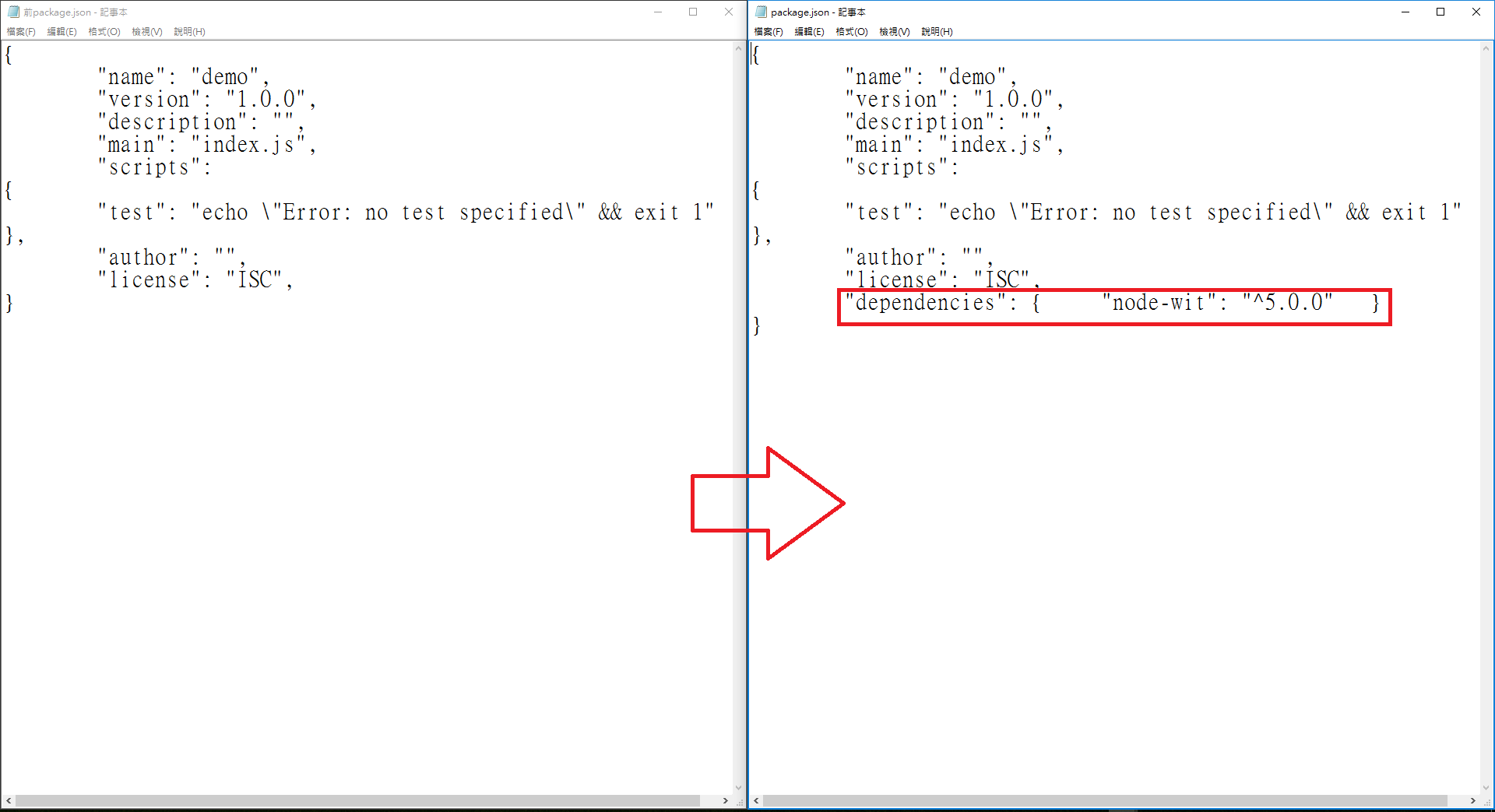


圖 .18 node環境設置圖

步驟11. 加入套件後，專案資料夾內的package.json檔中的內容將會從左邊的程式碼變為右邊，而兩者之間的差異在於多了紅色框框中的程式碼，如圖4.18。

### 物聯網部分

當前研究已確定使用Raspberry pi 3 model B+作為核心主機，並採用Raspbian作為OS系統，現已完成樹苺派系統建置、各種感測器的選用及採買與物聯網部分實驗模型初步設計。

因現階段查到的物聯網資料多數都是透過內網的方式建立連線與管理，但此方法與本研究之核心方向不相符，因此未來預計先去確立外網伺服器具體架設之技術，成功架立後再回過頭連接感測器等設備，以下將展示現已完成之樹苺派系統建置。

1. 樹苺派系統建置

透過Win 32 Disk Manager 燒錄最新版本Raspbian到Micro SD卡後，開啟Terminal進行系統初始設定，在輸入指令令軟體更新至最新版本，而後加入中文輸入法等調整，如圖4.19。

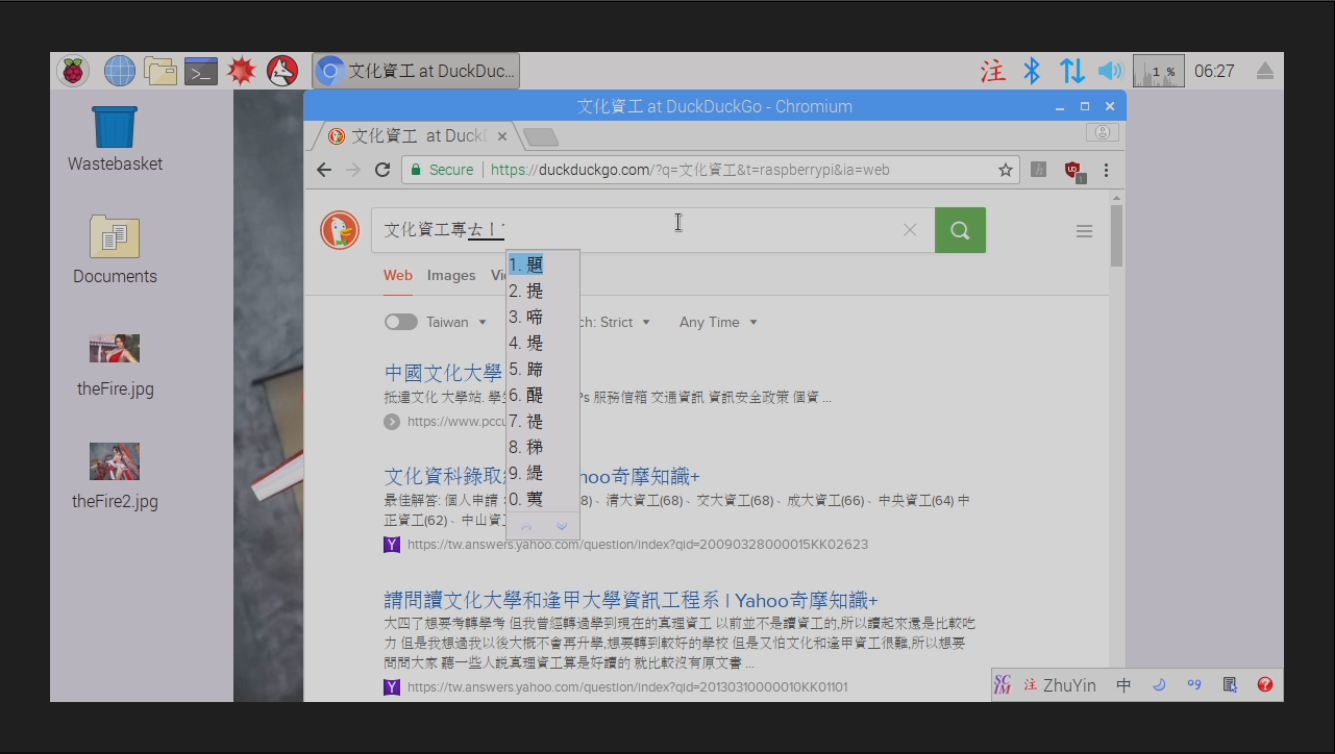


圖 .19 Raspberry pi運行截圖

* 1. 研究人員分配

由一般居家環境正常使用之概念架構，選擇以居住家庭及辦公室為個案，發展出可行的居家管理APP。

參與本研究之工作人員及預定之工作內容如表4.1所示。

**表 4.1參與本項研究之人員及工作內容**

|  |  |
| --- | --- |
| 姓名 | 工作內容 |
| 吳梓境  紀俊丞 | 介面規劃設計、程式設計、程式開發督導 |
| 吳承諺 | 樹梅派硬體架構之設計、各種感測器規劃  專題統整規劃、物聯網系統設計 |
| 蔡旻軒 | 人工智慧應用、WIT.AI 程式設計與實行 |

# 預期成果

由一般居家環境正常使用之概念架構，選擇以居住家庭及辦公室為個案，發展出可行的居家管理APP。

本研究規劃案，預期可以得到下列之成果：

1. 便利性

以往家庭中，所以電器都有獨立開關，不論按鍵式或是遙控式，希望透過我們所設計的APP來給予使用者們更多的方便性，將所有電器的控制方式移植至手機上，甚至能夠透過語音操控，以獲取更大的便利性。

1. 等待時間的減少

以往任何電器設備都幾乎都有一些等待時間，如冷氣開了需要一段時間房間溫度才會明顯下降，又或者是將熱水煮開的這段時間等等，希望能夠透過APP的遠端以及定時控制，有效的減少使用者在使用這些電器產品時的等待時間，可以在預定時間開啟某項產品，例如：能將開水設定在早晨幾點鐘自動煮水，起床後能夠直接使用，不須再等待燒開水的這段時間。

1. 弱勢族群

我們APP希望能夠做到完全透過手機控制所有產品，對於行動不方便的族群，不用再請他人幫忙拿取遙控器之類的，又或者是簡單的按鍵式開關都能夠透過手機實行，我們也將寫入語音方面的功能，對於任何族群都將達到方面性的提升。

# 參考文獻

1. 鄭永楨,“International Journal of Science and Engineering”,台南大學,vol.5, no.1,pp.17-22
2. 王路敬,“電腦編成技巧與維護”,中國信息產業商會,vol.18, pp.44-46, 2017
3. 李國淼,“信息通信”,湖北通信服務公司,vol.8, pp.115-116, 2015
4. 張欣,“科技資訊”,北京國際科技服務中心,vol.15, pp.38-40, 2016
5. “智能感測器”,2018, May 18, In Wikipedia the free encyclopedia, Retrieved June12,2018,https://zh.wikipedia.org/wiki/%E6%99%BA%E8%83%BD%E6%84%9F%E6%B8%AC%E5%99%A8
6. “物聯網”,2018, June 11, In Wikipedia the free encyclopedia, Retrieved June 13,2018,<https://zh.wikipedia.org/wiki/%E7%89%A9%E8%81%94%E7%BD%91>
7. “語音識別”,2018, May 17, In Wikipedia the free encyclopedia, Retrieved June 13,2018,<https://zh.wikipedia.org/wiki/%E8%AF%AD%E9%9F%B3%E8%AF%86%E5%88%AB>
8. “Android studio”,2014, Dec 12, In Wikipedia the free encyclopedia, Retrieved June 13,2018,<https://www.ithome.com.tw/news/92892>
9. “樹莓派”,2018, June 6, In Wikipedia the free encyclopedia, Retrieved June 13,2018https://zh.wikipedia.org/wiki/%E6%A0%91%E8%8E%93%E6%B4%BE