

# 正修科技大學

109 學年度學生專題報告

資訊工程系-甲班

進修部四技

亂碼電子鎖

校內指導老師：李克宇

學生姓名：黃玉鑫 90618101

王維安 90618106

陳慶煌 90618109

## 摘 要

在現今我們人手一支手機的時代加上從不離身，讓我們想到其實鑰匙可以淘汰掉了，網路應用的發達及科技的進步，所帶來了便利性及安全性的提升。

我們當初想到現今一般的鎖其實上網查看如何使用工具來開鎖其實很容易就可以解開了，雖說現今有使用紅外線來開啟鐵捲門的鎖，但如果是有心人士只要會接收紅外線訊號其實這也會被破解，於是我們想到使用物聯網(Arduino)所產生的亂碼上傳到伺服器，在由手機 App 擷取伺服器上的資料(全程需要網路)，App 上顯示的密碼來輸入數字鍵來解開電子鎖，這樣的裝置是比前面所說的裝置還要更安全!

**關鍵字：**物聯網、亂碼、App、電子鎖

# 目 錄

摘要.....	I
圖目錄 .....	III
第壹章、簡介 .....	1
第貳章、系統功能 .....	2
2-1 伺服端 .....	2
2-2 客戶端 .....	2
第參章、硬體配備介紹 .....	5
3-1 Arduino UNO.....	5
3-2 NodeMCU.....	5
3-3 伺服馬達 .....	6
3-4 4*4 鍵盤 .....	7
3-5 LCD1602 .....	7
第肆章、硬體電路分析 .....	8
第伍章、軟體說明 .....	11
5-1 Arduino IDE.....	11
5-2 Android Studio .....	14
5-3 Adafruit IO.....	16
執行情形 .....	18
未來研究方向 .....	24
討論與製作感言 .....	25
致謝.....	26
參考文獻 .....	27

## 圖目錄

第壹章、簡介 .....	1
第貳章、系統功能 .....	2
圖 2-1 系統功能架構.....	4
第參章、硬體配備介紹 .....	5
圖 3-1 Arduino UNO.....	5
圖 3-2 NodeMCU.....	6
圖 3-3 伺服馬達.....	6
圖 3-4 4*4 鍵盤 .....	7
圖 3-5 4*4 鍵盤電路.....	7
圖 3-6 LCD1602.....	7
第肆章、硬體電路分析 .....	8
圖 4-1 硬體架構流程圖 .....	8
圖 4-2 電路圖 .....	9
第伍章、軟體說明 .....	11
圖 5-1 傳送資料流程.....	12
圖 5-2 NodeMCU產生AP流程 .....	13
圖 5-3 App內容(1).....	14
圖 5-4 App內容(2).....	14
圖 5-5 App內容流程(1).....	15
圖 5-6 App內容流程(2).....	15
圖 5-7 Adafruit IO資料 .....	17
圖 5-8 伺服器流程.....	17

執行情形 .....	18
圖 6-1 選擇清單按鍵說明 .....	18
圖 6-2 功能表功能.....	18
圖 6-3 Set WiFi 進去畫面 .....	19
圖 6-4 手機 App 連線傳送資料過程 .....	19
圖 6-5 App 傳送完以後重新連線 .....	20
圖 6-6 input Password 進去畫面 .....	20
圖 6-7 使用者正在觀看 App 密碼 .....	21
圖 6-8 使用者關閉鎖 .....	21
圖 6-9 使用者正在輸入密碼.....	22
圖 6-10 使用者輸入正確密碼.....	22
圖 6-11 密碼鎖開啟.....	23
圖 6-12 手機被警告.....	23

## 第壹章 簡介

現今人手一支手機加上從不離身，於是我們想到讓身上的鑰匙轉換成手機 App，App 顯示密碼讓使用者可以輸入家裡的電子門鎖，密碼會隨著時間而變更，這樣讓使用者身上不用再多帶鑰匙這一種重物，安全性更是提升。

這樣的想法是來自於現在很多電腦(手機)都會要求二次認證，像是電腦登入 Google 就會有二次認證訊息傳到你的手機，再由手機顯示的認證碼輸入到電腦上進行登入，於是這個專題就這樣出來了。

## 第貳章 系統功能

系統功能分為伺服端與客戶端，伺服端由硬體控制加上伺服器，硬體電源來自行充電源，客戶端由手機 App 當主要功能和硬體操作，兩者全程都需要網路。

### 2 - 1 伺服端

由 Arduino UNO 和 NodeMCU 來組成，Arduino 裝置控制和產生資料，NodeMCU 來傳送資料，裝置方面有可觀看的 LCD 螢幕還有可輸入的鍵盤，可讓客戶端操作裝置使用功能，裝置有兩種功能，(1).Arduino 產生資料(含密碼)傳給 NodeMCU，密碼每 30 秒更換一次，(2).當客戶端按下裝置的設定 WiFi 此功能 NodeMCU 會發送 AP 給手機 App 使用，此時手機必須連線到 NodeMCU 所發出的 AP(Access Point)，搭配 App 輸入 WiFi 網域名稱及密碼連線 Push 到 NodeMCU 來做更改，NodeMCU 接收到以後會用從 App Push 過來的資料來重新跟網路與伺服器連線，若這組網域名稱、密碼不能連上會使用上一組連線過的網域名稱、密碼來重新連線(EEPROM 存取、讀出)，相反如果連線上了該組會存入 EEPROM。

### 2 - 2 客戶端

客戶端可以使用裝置的鍵盤和顯示文字的 LCD，當裝置上電時會出現功能表，功能表供客戶端選擇 2 種功能，(1).設定 WiFi 網域

名稱及密碼，(2).密碼輸入以致讓裝置開門，當輸入錯誤超過 3 次 Arduino 會彙整資料傳送一個狀態給伺服器，此狀態如果讓手機 App 讀取到會發送警告給客戶端觀看有人正在嘗試開啟門鎖，兩者功能都需要搭配手機 App 來使用。App 對應裝置上的 2 種功能，(1).當使用者按下裝置設定 WiFi，此時 NodeMCU 會發出 AP(Access Point)供手機連線，手機連線以後使用 App 來輸入 NodeMCU 連接 WiFi 的網域名稱及密碼來做更改連線，(2).App 會自動從伺服器上抓取密碼下來供客戶端觀看，每 30 秒更換一次密碼(同步更換)，再由 App 上的密碼來輸入硬體上的鍵盤，鍵盤功能可關閉門鎖也可輸入，輸入的數字會顯示在 LCD 螢幕上，輸入錯一個數字可倒退刪一個數字也可以全部刪除，當密碼正確 LCD 會顯示 password correct 即開啟門鎖，不正確即顯示 password Error 門鎖不動，當使用者連續輸入超過 3 次，NodeMCU 發的資料裡會有一個狀態做更改然後上傳到伺服器，此時手機讀取資料的時候會順便讀取此狀態，讀取到以後 App 會發送一個警告給手機讓客戶端知道有人正在嘗試開啟門鎖。



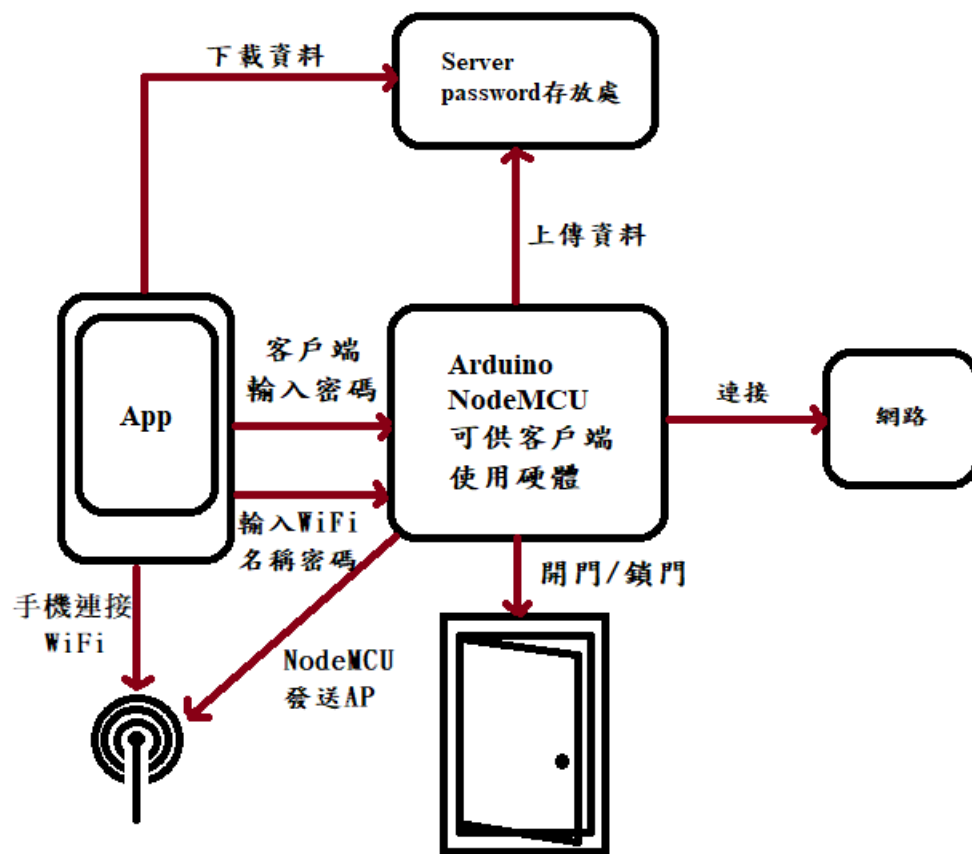


圖 2-1 系統功能架構

## 第參章 硬體配備介紹

### 3 – 1 Arduino

Arduino 電路板設計使用各種微處理器和控制器。這些電路板配有一組數字和類比 I/O 引腳，可以連接各種擴充板或麵包板（封鎖板）和其他電路。這些電路板具有串列埠，包括某些型號上的通用串列匯流（USB），也用於從個人電腦載入程式。微控制器通常使用 C/C++ 程式語言。除了使用傳統的編譯工具鏈外，Arduino 專案還提供了一個基於 Processing 語言專案的整合式開發環境。

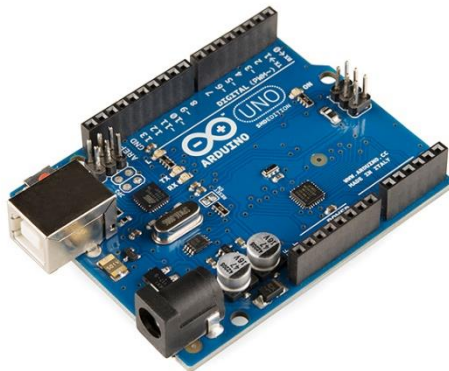


圖 3-1 Arduino UNO

### 3 – 2 NodeMCU

NodeMCU 是一個開源的物聯網平台。它使用 Lua 手稿語言編程。該平台基於 eLua 開源專案，底層使用 ESP8266 sdk 0.9.5 版本。該平台使用了很多開源專案，例如 lua-cjson, spiffs. NodeMCU 包含

了可以執行在 esp8266 Wi-Fi SoC 晶片之上的韌體, 以及基於 ESP-12 模組的硬體。



圖 3-2 NodeMCU

### 3 - 3 伺服馬達

伺服電機是一種帶有輸出軸的小型設備。通過向伺服機構發送編碼信號，可以將該軸定位到特定的角度位置。只要編碼信號存在於輸入線上，

伺服就會保持軸的角位置。如果編碼信

號改變，則軸的角位置改變。在實踐中，伺服系統用於無線電控制的飛機中以定位控制表面，如電梯和方向舵。它們還用於無線電遙控車，Puppet，當然還有機器人。



圖 3-3 伺服馬達

### 3-4 4\*4 鍵盤

4\*4 鍵盤是一種可以像電腦鍵盤一樣按下按鍵發送訊號。行與列之交集點為一按鍵點，最多可控制 16 個按鍵如圖 3-4 所示。將接腳接上 Arduino Uno 以掃描方式掃描其行列線，偵測行列的變化即可得知現階段按下哪顆按鍵。



圖 3-4 4\*4 鍵盤

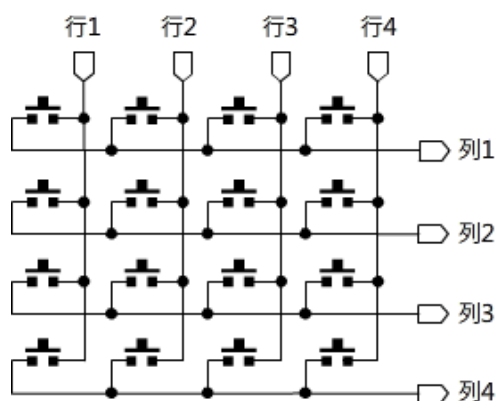


圖 3-5 4\*4 鍵盤電路

### 3-5 LCD1602

16x2 液晶顯示字符模塊，可用於顯示字母、數字、字符等。 I2C LCD1602 液晶模塊可以顯示 2 行，每行 16 個字符。對於初學者來說，不必為繁瑣複雜液晶驅動電路連線而煩惱，



圖 3-6 LCD1602

LCD 擴展板將電路簡化，使用相關文檔中的 Library，只需使用幾行簡單控制代碼便能完成 LCD 控制顯示的功能。背面的電位器還能提供你調節液晶顯示器對比度的功能。

## 第肆章 硬體電路分析

硬體主要為 Arduino UNO 和 NodeMCU 來當核心，電源部分由行充電源供應，2 塊以 RX、TX 來傳送資料，NodeMCU 主要是負責產生資料傳送給 Arduino 和伺服器，則 Arduino 就是負責控制硬體裝置方面的問題，裝置有 2 種功能，一種是判斷輸入密碼是否正確，另一種是更換 NodeMCU 連接 WiFi 的名稱及密碼。

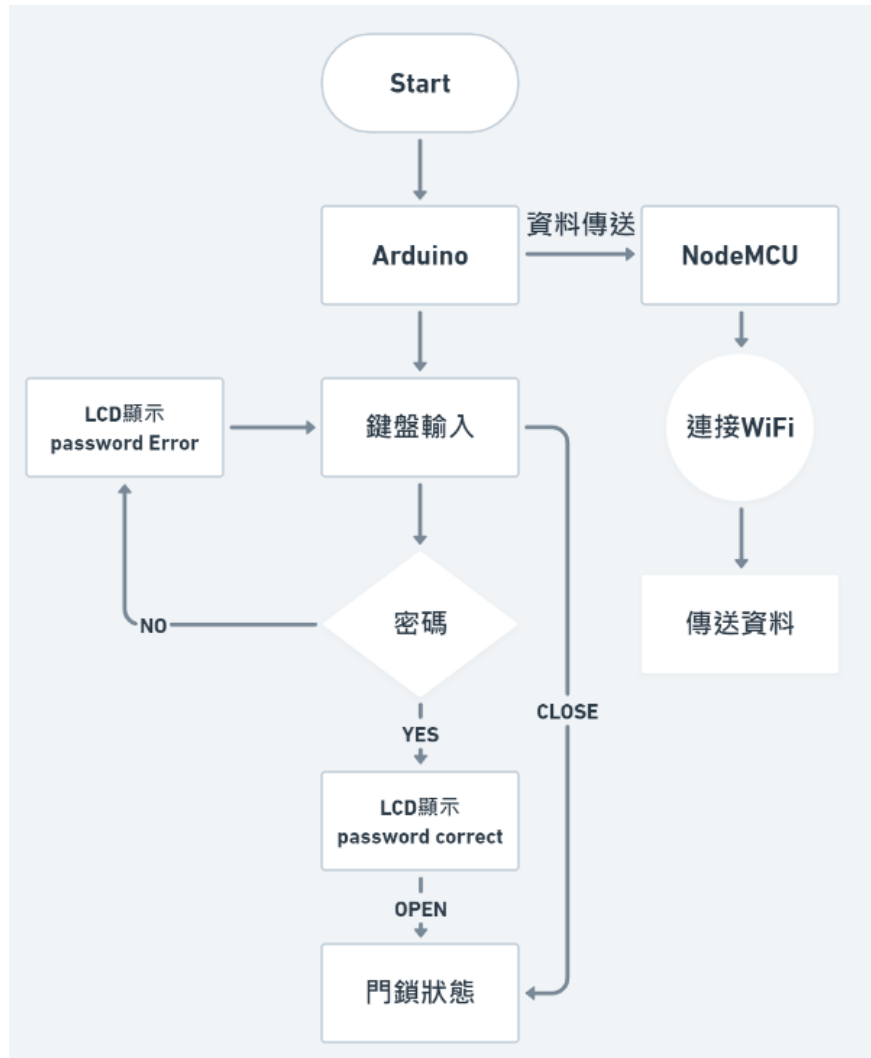


圖 4-1 硬體架構流程圖

由圖 4-1 可知 Arduino 彙整資料給 NodeMCU，NodeMCU 再連接 WiFi 在發送資料給伺服器，鍵盤可以切換功能，一個是設定 WiFi，一個是輸入密碼，輸入密碼功能鍵盤可以讓門鎖關閉也可以輸入密碼，密碼輸入錯誤 LCD 顯示 password Error 訊息門鎖狀態不動，當連續輸入錯誤 3 次會發送一個狀態給 NodeMCU，密碼輸入正確 LCD 顯示 password correct 門鎖開啟。

### 伺服馬達

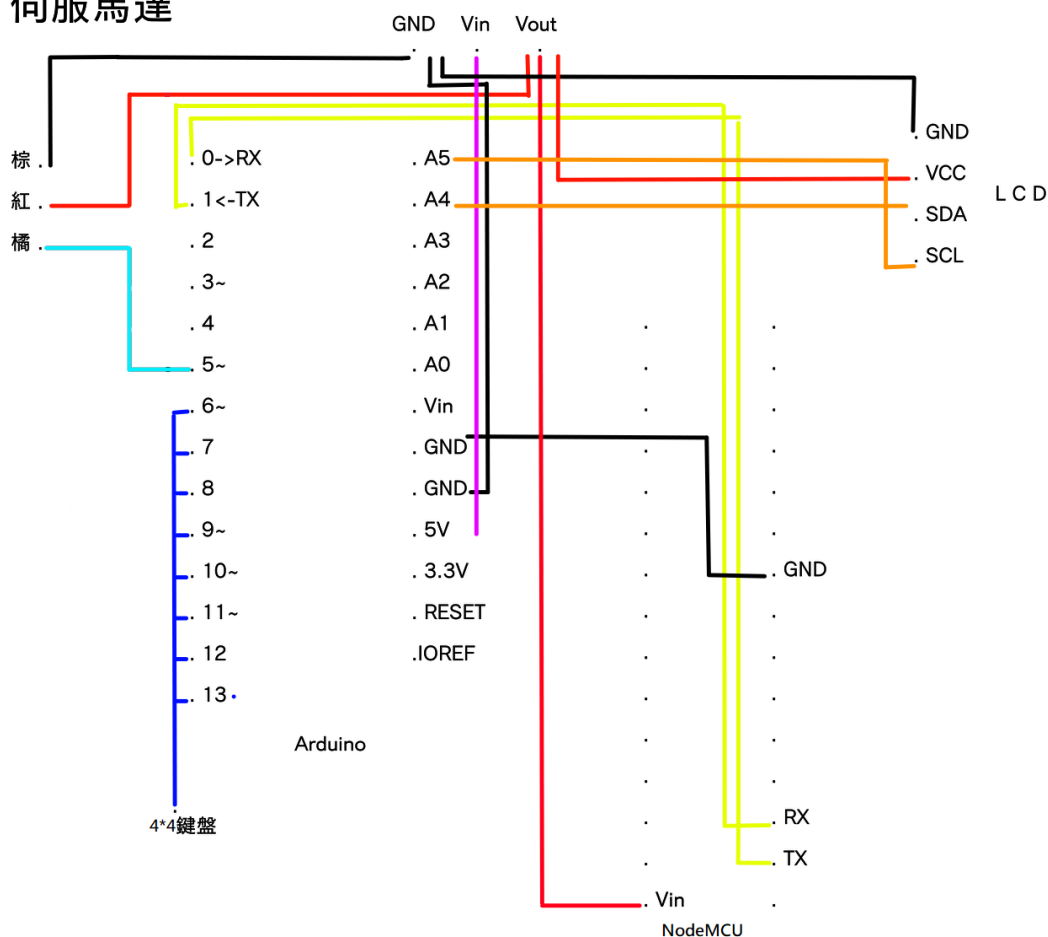


圖 4-2 電路圖

圖 4-2 為此硬體電路圖，Arduino RX TX 與 NodeMCU RX TX 交叉接以便傳送資料接地要相接，Arduino 5 pin 接伺服馬達，6~13 pin 接 4\*4 鍵盤，A5 A4 接 LCD1602，剩下的接地與 GND Vin Vout (5v) 整理到同一區塊，主要電源為行動電源。

## 第伍章 軟體說明

### 5 – 1 Arduino IDE

編輯程式採用 Arduino IDE 來編輯 Arduino 與 NodeMCU，編寫主要用 C 語言來編譯程式。

按鍵輸入密碼、LCD 顯示輸入的密碼錯誤、正確、開/關門鎖與彙整資料全是由 Arduino 所包，彙整的資料傳給 NodeMCU，資料包括密碼、目前鎖的狀態和嘗試輸入超過錯誤 3 次密碼都會傳狀態給 NodeMCU，要控制這些硬體需裝上面所介紹的硬體 Library。

NodeMCU 用於連接 WiFi、產生 AP(Access Point)，WiFi 連接於手機網路或者區域 WiFi，通過網路來把 Arduino 產生的資料傳到伺服器 (Adafruit IO)，如果要使用 Arduino IDE 編輯 NodeMCU 的話，必須要安裝 NodeMCU 的開發版程式在 Arduino IDE，還有 Adafruit IO(伺服器)、ESP8266 WiFi 的 Library，接下來把 NodeMCU 產生的資料 push 到伺服器上，至於 NodeMCU 產生的 AP 必須由客戶端按下裝置的功能表設定 WiFi 功能後才會發出 AP，此時手機必須就要連上 NodeMCU 所發出的 AP，再使用手機上的 App 來更改 NodeMCU 連線的 WiFi 名稱及密碼，手機傳送成功 NodeMCU 會發送 push OK 來當做傳送成功的意思，此時 NodeMCU 會使用這組 WiFi 名稱及密碼來重新連線，如果連線不上會連線上一組連線成功



的 WiFi(EEPROM 存取、讀出)，相反的連線成功會存入 EEPROM

裡。(AP 發出的名稱：Lockpass 密碼：12345678)

以下拆分為兩種功能來做介紹，傳送資料與發送 AP 接收 App 資料的流程。

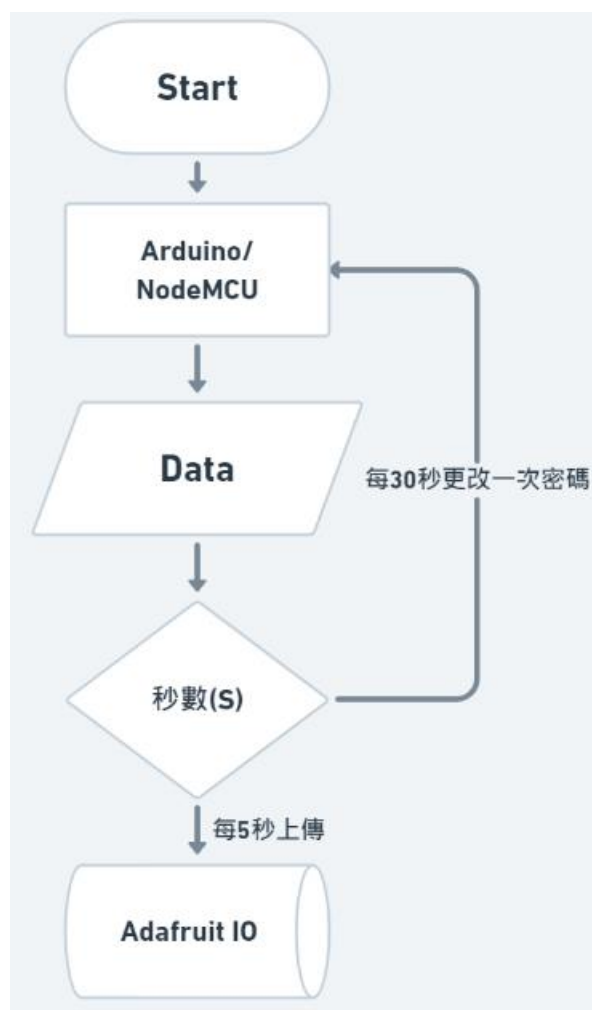


圖 5-1 傳送資料流程

由圖 5-1 可知道從 Arduino 所產生的資料傳給 NodeMCU，

NodeMCU 每 5 秒會上傳一次伺服器資料，每 30 秒 NodeMCU 會重

新取得一個密碼包裝成資料給伺服器，上傳資料協定為 MQTT。

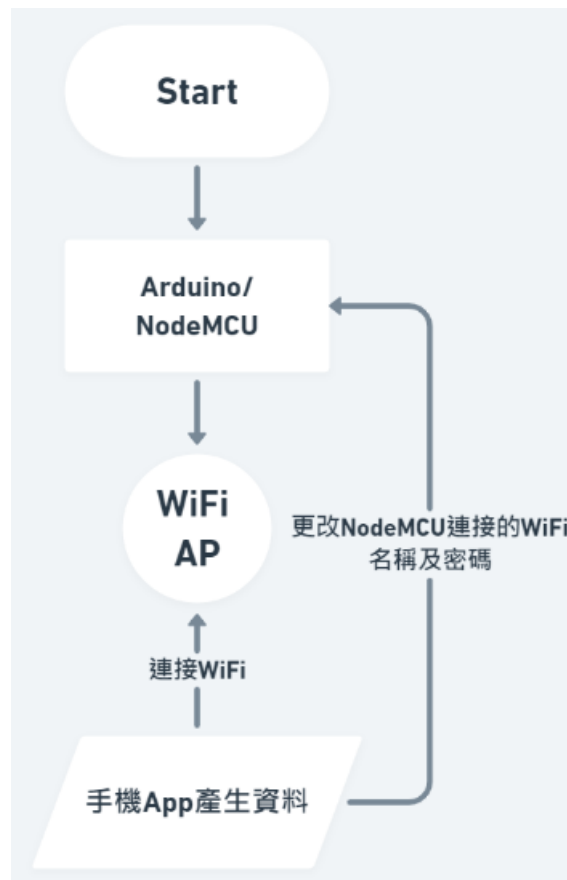


圖 5-2 NodeMCU 產生 AP 流程

由圖 5-2 可知當 NodeMCU 產生 AP，此時手機必須連上 NodeMCU 所發出的 AP，連接上以後客戶端使用者必須使用 App 來連接 NodeMCU，此時協定為 TCP 手機使用 Socket 傳送，為 NodeMCU 更改 WiFi 的名稱及密碼，但是如果傳送的這組名稱及密碼無法讓 NodeMCU 做連線，NodeMCU 會重新讀取上一次成功連線的 WiFi(EEPROM)。(AP 發出的名稱：Lockpass 密碼：12345678)

## 5 – 2 Android Studio

使用了 Android Studio 來製作手機 App，語言使用 Java 來編寫，App 主要有兩種功能，一種用於來取得 NodeMCU 上傳伺服器的資料，App 在從伺服器上擷取資料下來解析以便客戶端觀看，另一種是客戶端必須使用裝置上的功能表使用設定 WiFi 功能才能使用，當按下裝置此功能裝置會發出 AP 來給手機做連接，連接上以後可發送資料給裝置，以變更裝置連接 WiFi 的名稱密碼。

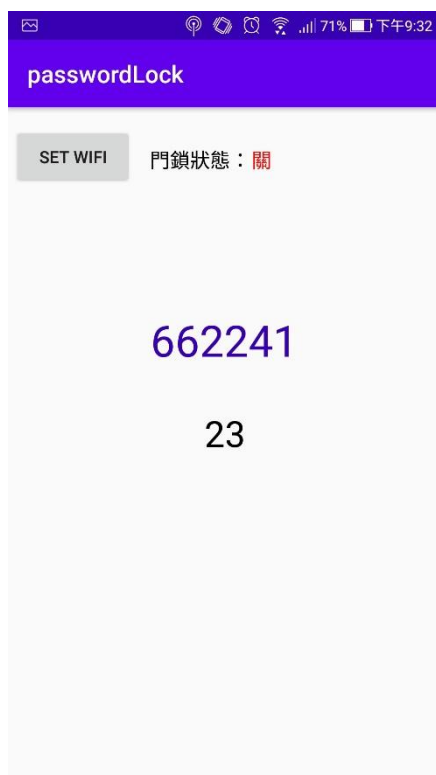


圖 5-3 App 內容(1)



圖 5-4 App 內容(2)

由圖 5-3 可知左上角按鈕為切換功能，中上為門鎖狀態，中間為開啟裝置的密碼以及更換秒數。

圖 5-4 為更改裝置連線 WiFi 網域名稱及密碼，手機必須連上裝置所發出的 AP 才能使用此功能，右上角按鈕為與裝置連線(此連線協定為 TCP 手機使用 Socket 做連線)，當按下按鈕如果當下手機連接於裝置 AP 下方文字會顯示已連線，相反的未連接會顯示連線失敗，中間兩個視窗是為裝置更改 WiFi 的視窗是可輸入的，中下的按鈕是當右上角顯示已連線才可按，當輸入完中間兩個視窗會把這些資料傳送給裝置，當裝置接收到會回傳給手機 App Push ok 在下方文字，相反的右上角顯示未連線或是連線失敗就會顯示傳送失敗。

(AP 發出的名稱：Lockpass 密碼：12345678)

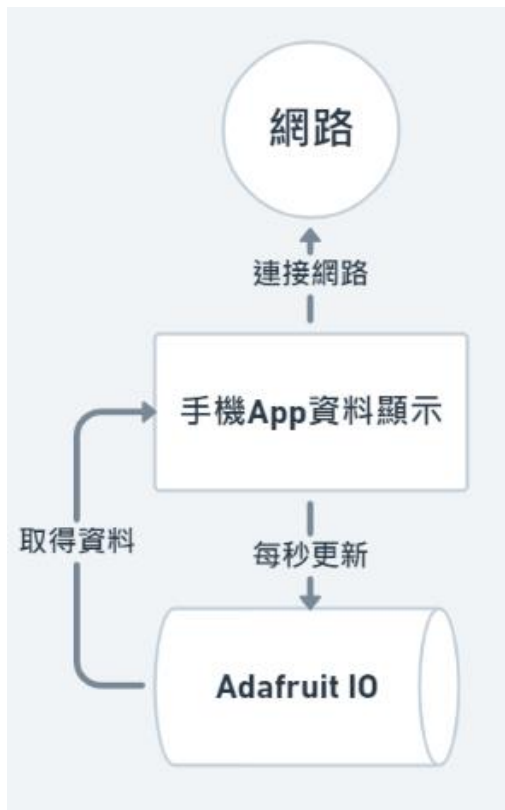


圖 5-5 App 內容流程(1)

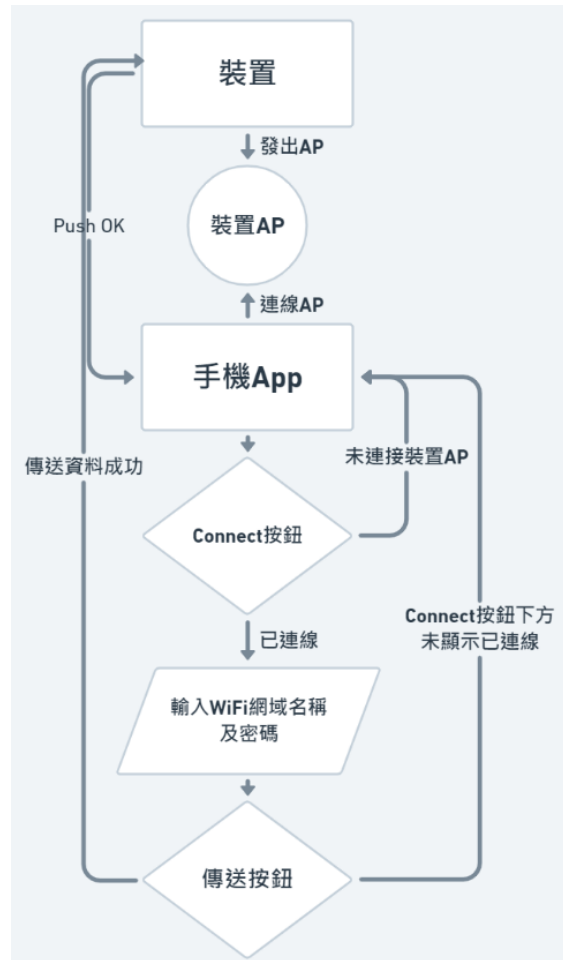


圖 5-6 App 內容流程(2)

圖 5-5、5-6 為 App 內容流程圖，由圖 5-5 可知道每一秒都在更新 App 的資料為客戶端觀看，圖 5-6 是手機必須連上裝置所發出的 AP 才能繼續下方流程，當手機連接裝置 AP 時就可以按下 Connect 按鈕再來輸入所要更換連線的 WiFi 名稱及密碼按出傳送，裝置如果接收到資料會回傳 Push ok 給手機 App，已完成此流程步驟。

### 5 – 3 Adafruit IO

Adafruit IO 是個免費使用 MQTT 雲端運算服務的網站，我們利

用它來當作存放密碼的伺服器，由 NodeMCU 傳來的資料儲存於這  
把這些資訊內容轉換成網頁(圖 5-7)，可讓 App 來擷取網頁內容。

Created at	Value
2020/11/12 7:52:46PM	334021,25,n,o
2020/11/12 7:52:41PM	334021,30,n,o
2020/11/12 7:52:36PM	154107,05,n,o
2020/11/12 7:52:30PM	154107,10,n,o
2020/11/12 7:52:25PM	154107,15,n,o
2020/11/12 7:52:20PM	154107,20,n,o
2020/11/12 7:52:15PM	154107,25,n,o
2020/11/12 7:52:10PM	154107,30,n,o
2020/11/12 7:52:05PM	035368,05,n,o
2020/11/12 7:52:00PM	035368,10,n,o

圖 5-7 Adafruit IO 資料

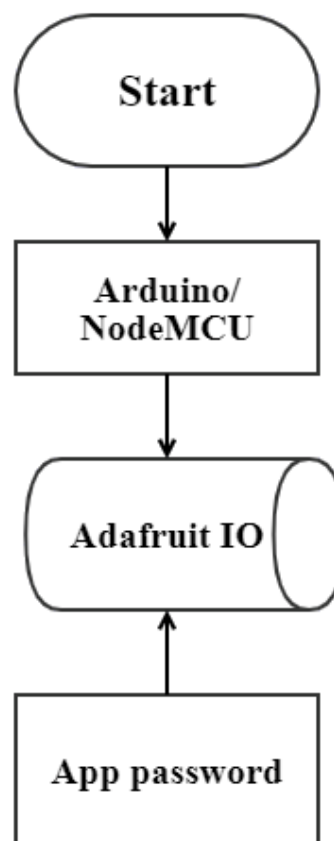


圖 5-8 伺服器流程

由圖 5-8 可知 NodeMCU 上傳資料給 Adafruit IO 儲存，再由 App 下載 Adafruit IO 資料來更新密碼。

## 執行情形



圖 6-1 選擇清單按鍵說明



圖 6-2 功能表功能



圖 6-3 Set WiFi 進去畫面(可按 D 強制跳出)

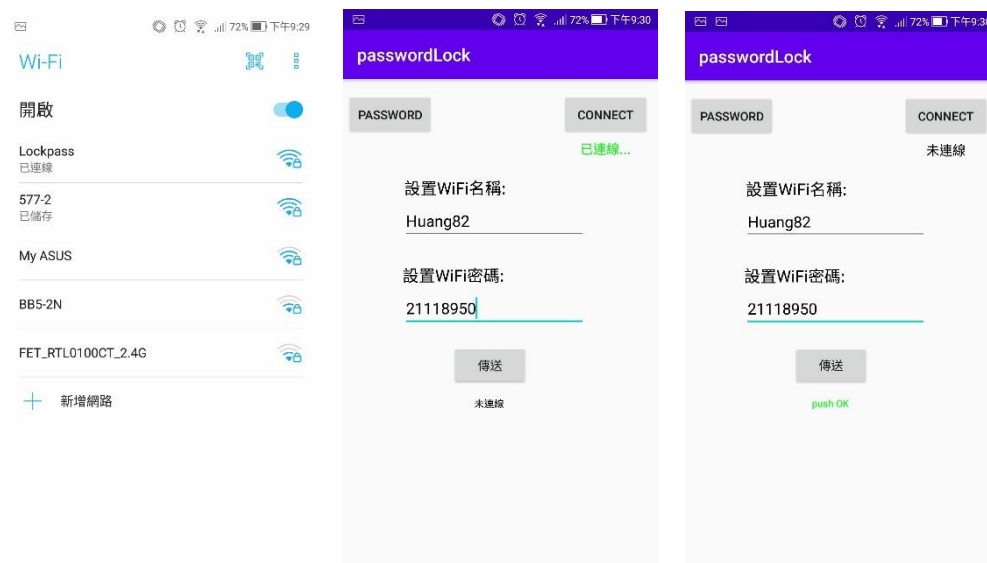


圖 6-4 手機 App 連線傳送資料過程

(AP 發出的名稱：Lockpass 密碼：12345678)





圖 6-5 App 傳送完以後重新連線(可按 D 強制跳出)



圖 6-6 input Password 進去畫面

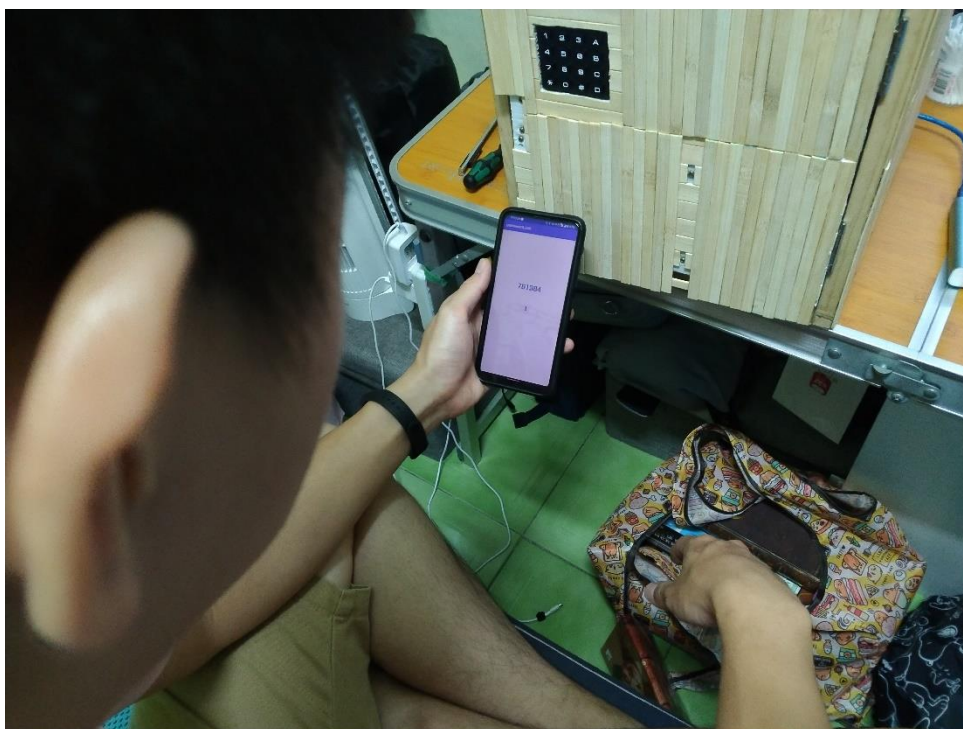


圖 6-7 使用者正在觀看 App 密碼

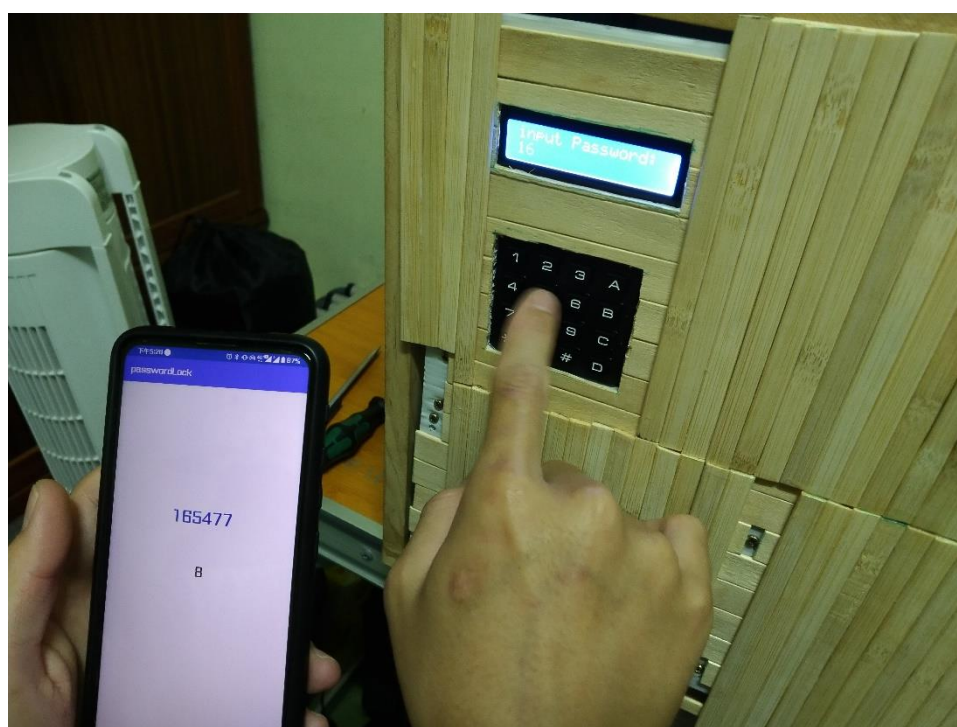


圖 6-8 使用者關閉鎖





圖 6-9 使用者正在輸入密碼

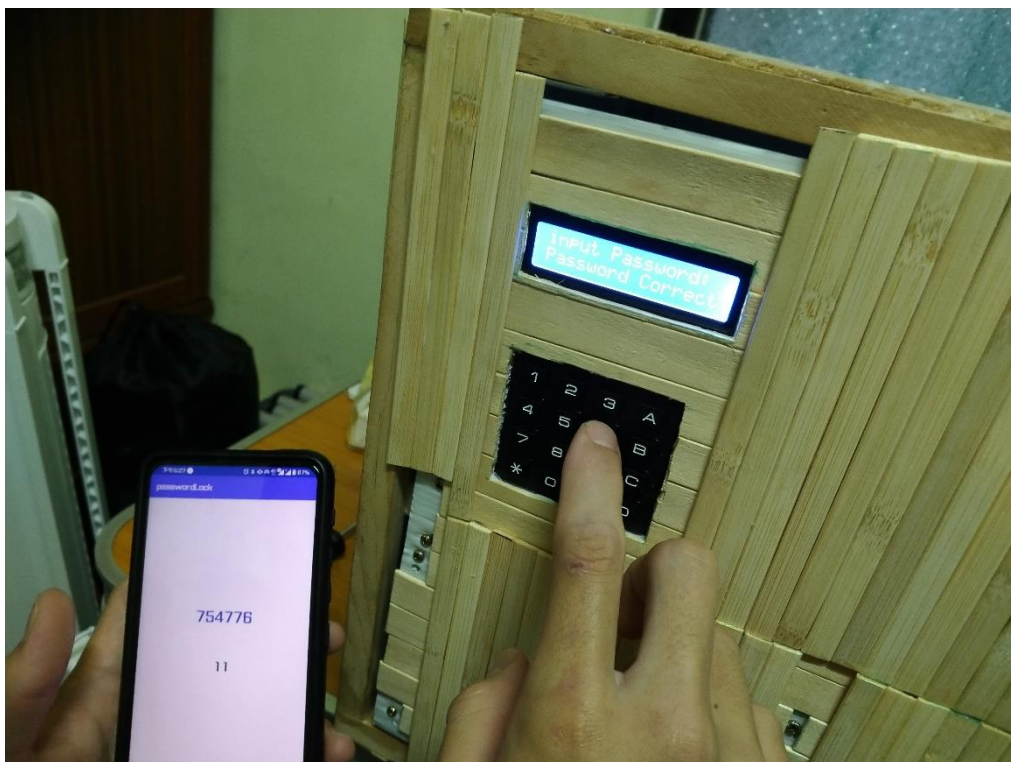


圖 6-10 使用者輸入正確密碼

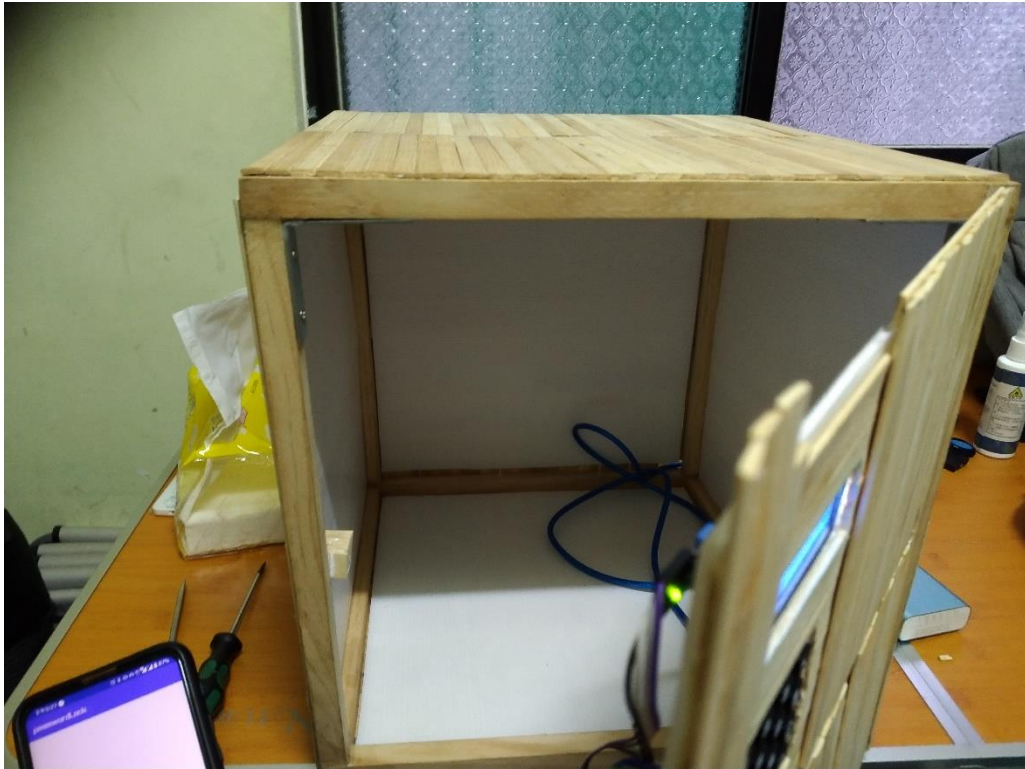


圖 6-11 密碼鎖開啟

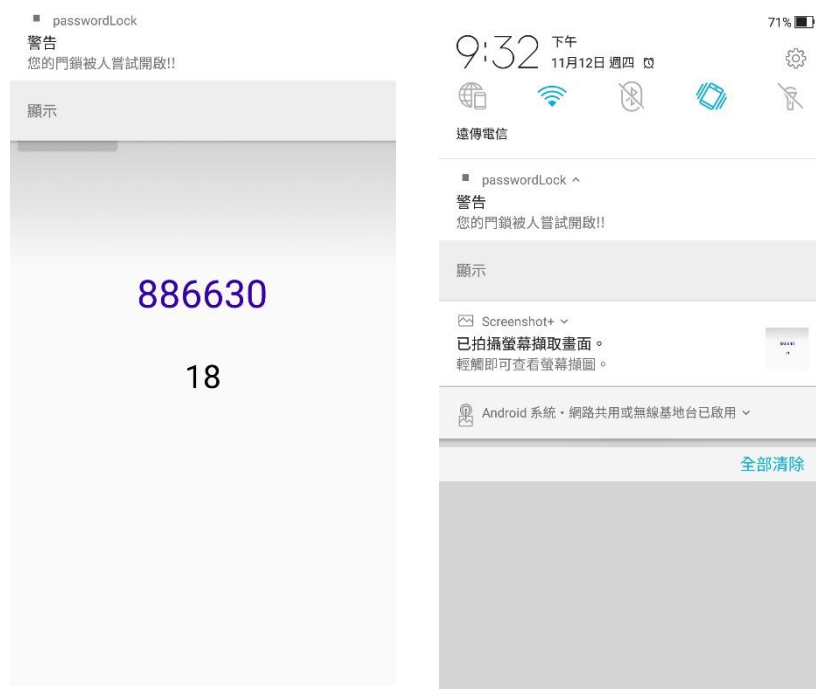


圖 6-12 手機被警告(輸入錯誤 3 次會被警告)

## 未來研究方向

製作出亂碼電子鎖是為了取代現在的一般門鎖，為了普及化需要一間公司來管理，當客戶安裝此門鎖需申辦帳密下載 App，App 用於接收從公司資料庫下載下來的資料進行解鎖，每一個帳密門鎖密碼都是獨立的。密碼可以優化成 2 維/3 維或是 QR Code 門鎖增加個掃描器這樣一來就不用用手按數字鍵，還有停電門鎖該怎麼處置、網路與網路穩定性也是一個很大問題。

## 討論與製作感言

專題製作真的會使人各方面進步，當時剛想出決定專題要做亂碼電子鎖，那時其實大約流程都訂好了，但實際下去做卻每一個流程都會出問題，像是原本伺服器是用 google apps script 來儲存至 google Excel，但它每次 push 上去的等待時間實在太久而換成 Adaruit IO，還有 App 原本是用 App Inventor 2(AI2)下去做的，但待機一段時間會跑出連線已斷的一大串錯誤，AI2 又是讓人方便做 App 的網頁所以很多功能不如直接用程式下去打，這些問題也只是其中 2 個問題還有很多沒有敘說，但出現問題也間接地讓我們學到新的程式語言、工具，也讓我們學到如何分配工作，再把每個人分配到的工作結合起來，這對我們以後工作絕對有幫助!

## 致謝

在製作期間非常感謝李克宇老師為我們的引導，讓我們在短短的時間內學到不同的程式語言、工具，也要感謝學校的各個老師的教導，雖然在這段期間常常遇到一推問題，常常上網爬文、詢問到非常的晚，但每一個問題解決促使我們更成長一步，希望剩下畢業前的時間更能充實增進我們的實力！

## 參考文獻

- [1] 陳柏孝，應用物聯網技術於提升保險櫃的安全性，義守大學電子工程學系碩士班碩士論文, 2017.
- [2] Wiki 維基百科
- [3] 趙英傑，超圖解 Arduino 互動設計入門
- [4] 趙英傑，超圖解物聯網 IoT 實作入門：使用 JavaScript／Node.JS／Arduino／Raspberry Pi／ESP8266／Espruino
- [5] 孫宏明，Android 程式設計入門、應用到精通（第五版）
- [6] Adafruit IO API, <https://io.adafruit.com/api/docs/#adafruit-io-http-api>
- [7] 洪維恩，Java 7 教學手冊 第五版(附光碟)



成品照片

