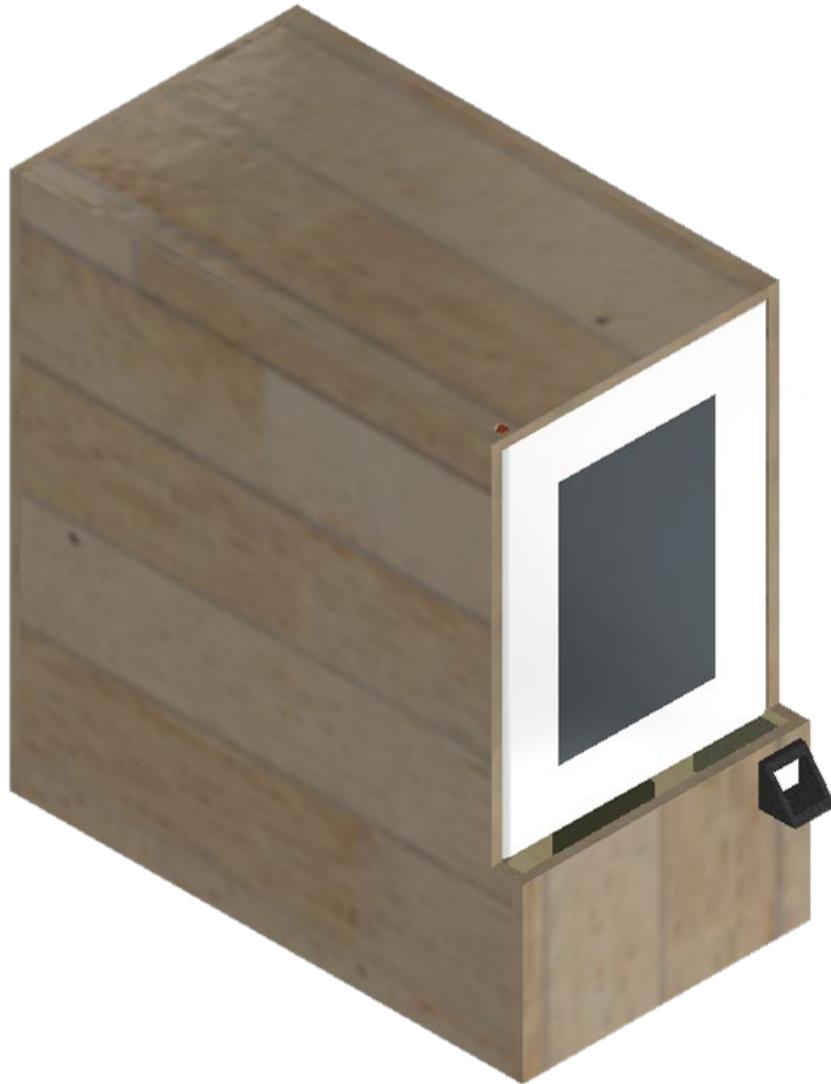


臺北市立大安高級工業職業學校專題製作競賽  
「專題組」作品說明書



群別：電機與電子群

作品名稱：禁機的巨人

關鍵詞：時間管理、自主讀書、Flask伺服器

# 目錄

壹、摘要 .....	1
貳、研究動機 .....	1
參、主題與課程之相關性或教學單元之說明 .....	2
一、硬體製作 .....	2
二、程式撰寫 .....	2
三、電路雕刻 .....	2
四、成品外觀 .....	3
肆、研究方法 .....	3
一、研究流程 .....	3
(一)、研究步驟 .....	3
(二)、操作步驟 .....	4
二、使用材料及工具 .....	7
(一)、零件介紹 .....	7
(二)、機構原理 .....	10
(三)、軟體介紹 .....	11
伍、研究結果 .....	15
一、硬體結構 .....	15
(一)、轉軸 .....	16
(二)、門 .....	16
二、軟體通訊架構 .....	16
(一)、Flask 伺服器及 API .....	17
(二)、Serial .....	17
三、成果展示 .....	17
(一)、APP 介面 .....	17
(二)、網頁介面 .....	17
陸、討論 .....	20
一、接腳不足 .....	20
二、硬體限制 .....	20
三、通訊處理 .....	20
柒、結論 .....	20
捌、參考資料及其他 .....	22
一、書籍資料 .....	22
二、網路資料 .....	22

## 表目錄

表 1	時間分配表 .....	3
表 2	Esp8266 NodeMCU 規格.....	7
表 3	Mega2560 規格.....	7
表 4	N20 直流減速馬達規格.....	8
表 5	5V 1A 無線充電發射模組規格.....	8
表 6	LRS-50-12 規格.....	9
表 7	USART-HMI 規格.....	9
表 8	UVC 254NM T5 規格 .....	9
表 9	LY-03 規格.....	9
表 10	ACS712 規格 .....	10
表 11	ESP32-CAM 規格 .....	10
表 12	AS608 規格.....	10

## 圖目錄

圖 1	Arduino 程式撰寫 .....	2
圖 2	Visual Studio 網頁前端設計 .....	2
圖 3	電路板成品 .....	2
圖 4	手機箱成品 .....	3
圖 5	研究步驟 .....	4
圖 6	遠端解鎖流程 .....	4
圖 7	定時解鎖流程 .....	5
圖 8	本地解鎖流程 (1) .....	6
圖 9	本地解鎖流程 (2) .....	6
圖 10	Esp8266 NodeMCU .....	7
圖 11	Mega2560 .....	7
圖 12	N20 減速馬達 .....	8
圖 13	復位開關 .....	8
圖 14	無線充電發射模組 .....	8
圖 15	LRS-50-12 .....	9
圖 16	UVC 254NM .....	9
圖 17	USART-HMI .....	9
圖 18	LY-03 .....	9
圖 19	ACS712 .....	10
圖 20	ESP32-CAM .....	10
圖 21	AS608 .....	10
圖 22	Autodesk inventor logo .....	11
圖 23	Autodesk inventor 3D 圖繪製介面 .....	11
圖 24	Arduino logo .....	12
圖 25	Arduino 程式撰寫 .....	12
圖 26	Arduino 程式撰寫與接線測試 .....	12
圖 27	Altium Desiner logo .....	13
圖 28	成品電路圖 .....	13
圖 29	成品雕刻圖 .....	13
圖 30	Flask logo .....	14
圖 31	App Inventor logo .....	14
圖 32	Fritzing logo .....	14
圖 33	Fritzing 接線圖 .....	15
圖 34	Inventor 爆炸圖 .....	15
圖 35	聯軸器 .....	16
圖 36	門板 .....	16
圖 37	通訊關係圖 .....	16
圖 38	APP 主頁 .....	17
圖 39	APP 讀書計畫 .....	17
圖 40	APP 問空畫面 .....	17
圖 41	網頁左右側頁面 .....	18
圖 42	網頁主頁 .....	18

圖 43	網頁讀書計畫.....	19
圖 44	網頁鏡頭監控.....	19

# 【禁機的巨人】

## 壹、摘要

為了使孩子能有效運用時間來學習，本專題設計出一款自主使用亦可交由家長管理，兩者功能兼備的手機箱。我們不僅使用了網頁進行遠端監控、也設計 APP 讓管理者可以在手邊沒有電腦的情況下，方便地進行遠端操控。此外，手機箱內部加上了無線充電、電流檢測器及 UVC 消毒燈管，可偵測手機放置時間以統計讀書時長，並保護使用者可以遠離病毒的威脅。本專題讓使用者可以藉由這個手機箱，達到時間妥善利用以及高效學習的目的。

在控制端，我們使用網頁、APP 進行控制，透過 NodeMCU 連接使用 Python 撰寫的 Flask 伺服器，並使用 API 來進行網頁與手機箱之間的溝通，進行遠端解鎖、定時解鎖的控制。手機箱內加裝了 UVC 消毒燈管進行手機消毒、APP 設定讀書計畫並顯示於 HMI 上，以及藉由無線充電，監測手機收納於箱子的時長。

## 貳、研究動機

在 2018 年底，世界衛生組織(WHO)正式宣佈將網路遊戲成癮(Gaming disorder)納入精神疾病，主要準則為「失控造成生活失能」，即個人對於上網玩遊戲失去控制力，甚而影響他的日常生活功能，比如學校課業及工作任務等，即使個人感知到已產生負面影響，仍無可自拔。我們想要做出一個能夠協助家長控管小孩手機使用時間的裝置，且能培養小孩自我規畫及自主學習的能力。

手機成癮日益嚴重的現在，學生因手機濫用而嚴重影響日常生活、學習成績，而家長卻沒有一種效益高的管理方式。若強硬禁止手機的使用，會使孩子與家長的關係產生裂痕，且家長在上班途中，無法時時刻刻關注孩子目前使用手機的狀況，那是否能有解決方法？

為了改善這樣的狀況，我們希望以「遠端控制系統」和「讀書計畫表」的小箱子來達成目標，減輕家長的負擔，同時記錄孩子的讀書狀況，進而培養孩子自我管理的能力。透過程式設計與網路溝通撰寫，使手機 APP 及網頁能遠端控制手機箱門的開合，由家長決定手機箱門的開關，包括遠端解鎖和定時解鎖。當家長在家時，能使用指紋解鎖解開起箱門，也能收集手機的放入時間，方便孩子自我檢討。手機箱也擁有自我規畫讀書時間的功能，讓使用者能自行設計適合自己的讀書計畫，提醒使用者何時要讀書，增進讀書的積極度。

## 參、主題與課程之相關性或教學單元之說明

### 一、硬體製作

高二電子學實習課中，學習到 3D 列印機的使用，讓我們能將門的構想實現，箱子主體則使用 5mm 木板組成，利用雷射切割機，切割出所需大小並進行組裝。

### 二、程式撰寫

我們使用 Arduino IDE 為此次專題硬體撰寫控制程式，並使用 Mega2560、NodeMCU 作為硬體平台，NodeMCU 為此次專題的網路通訊器，最後透過 Flask 進行溝通，串聯起 APP、伺服器及硬體端的通訊，達到在外能對箱體進行控制。



```
16 #include <esp8266.h>
17 #include <ESPClient.h>
18 #include <ESP8266WiFi.h>
19 #include <ESP8266WebServer.h>
20 #include <ESP8266HTTPClient.h>
21 #include <ESP8266HTTPUpdateEngine.h>
22 #include <ESP8266WiFiMulti.h>
23 #include <ESP8266WiFiClient.h>
24 #include <ESP8266WiFiGeneric.h>
25 #include <ESP8266WiFiSoftAP.h>
26 #include <ESP8266WiFiSoftAPModes.h>
27 #include <ESP8266WiFiSoftAPMode.h>
28 #include <ESP8266WiFiSoftAPMode.h>
29 #include <ESP8266WiFiSoftAPMode.h>
30 #include <ESP8266WiFiSoftAPMode.h>
31 #include <ESP8266WiFiSoftAPMode.h>
32 #include <ESP8266WiFiSoftAPMode.h>
33 #include <ESP8266WiFiSoftAPMode.h>
34 #include <ESP8266WiFiSoftAPMode.h>
35 #include <ESP8266WiFiSoftAPMode.h>
36 #include <ESP8266WiFiSoftAPMode.h>
37 #include <ESP8266WiFiSoftAPMode.h>
38 #include <ESP8266WiFiSoftAPMode.h>
39 #include <ESP8266WiFiSoftAPMode.h>
40 #include <ESP8266WiFiSoftAPMode.h>
41 #include <ESP8266WiFiSoftAPMode.h>
42 #include <ESP8266WiFiSoftAPMode.h>
43 #include <ESP8266WiFiSoftAPMode.h>
44 #include <ESP8266WiFiSoftAPMode.h>
45 #include <ESP8266WiFiSoftAPMode.h>
46 #include <ESP8266WiFiSoftAPMode.h>
47 #include <ESP8266WiFiSoftAPMode.h>
48 #include <ESP8266WiFiSoftAPMode.h>
49 #include <ESP8266WiFiSoftAPMode.h>
50 #include <ESP8266WiFiSoftAPMode.h>
51 #include <ESP8266WiFiSoftAPMode.h>
52 #include <ESP8266WiFiSoftAPMode.h>
53 #include <ESP8266WiFiSoftAPMode.h>
54 #include <ESP8266WiFiSoftAPMode.h>
55 #include <ESP8266WiFiSoftAPMode.h>
56 #include <ESP8266WiFiSoftAPMode.h>
57 #include <ESP8266WiFiSoftAPMode.h>
58 #include <ESP8266WiFiSoftAPMode.h>
59 #include <ESP8266WiFiSoftAPMode.h>
60 #include <ESP8266WiFiSoftAPMode.h>
61 #include <ESP8266WiFiSoftAPMode.h>
62 #include <ESP8266WiFiSoftAPMode.h>
63 #include <ESP8266WiFiSoftAPMode.h>
64 #include <ESP8266WiFiSoftAPMode.h>
65 #include <ESP8266WiFiSoftAPMode.h>
66 #include <ESP8266WiFiSoftAPMode.h>
67 #include <ESP8266WiFiSoftAPMode.h>
68 #include <ESP8266WiFiSoftAPMode.h>
69 #include <ESP8266WiFiSoftAPMode.h>
70 #include <ESP8266WiFiSoftAPMode.h>
71 #include <ESP8266WiFiSoftAPMode.h>
72 #include <ESP8266WiFiSoftAPMode.h>
73 #include <ESP8266WiFiSoftAPMode.h>
74 #include <ESP8266WiFiSoftAPMode.h>
75 #include <ESP8266WiFiSoftAPMode.h>
76 #include <ESP8266WiFiSoftAPMode.h>
77 #include <ESP8266WiFiSoftAPMode.h>
78 #include <ESP8266WiFiSoftAPMode.h>
79 #include <ESP8266WiFiSoftAPMode.h>
80 #include <ESP8266WiFiSoftAPMode.h>
81 #include <ESP8266WiFiSoftAPMode.h>
82 #include <ESP8266WiFiSoftAPMode.h>
83 #include <ESP8266WiFiSoftAPMode.h>
84 #include <ESP8266WiFiSoftAPMode.h>
85 #include <ESP8266WiFiSoftAPMode.h>
86 #include <ESP8266WiFiSoftAPMode.h>
87 #include <ESP8266WiFiSoftAPMode.h>
88 #include <ESP8266WiFiSoftAPMode.h>
89 #include <ESP8266WiFiSoftAPMode.h>
90 #include <ESP8266WiFiSoftAPMode.h>
91 #include <ESP8266WiFiSoftAPMode.h>
92 #include <ESP8266WiFiSoftAPMode.h>
93 #include <ESP8266WiFiSoftAPMode.h>
94 #include <ESP8266WiFiSoftAPMode.h>
95 #include <ESP8266WiFiSoftAPMode.h>
96 #include <ESP8266WiFiSoftAPMode.h>
97 #include <ESP8266WiFiSoftAPMode.h>
98 #include <ESP8266WiFiSoftAPMode.h>
99 #include <ESP8266WiFiSoftAPMode.h>
100 #include <ESP8266WiFiSoftAPMode.h>
```

圖 1 Arduino 程式撰寫

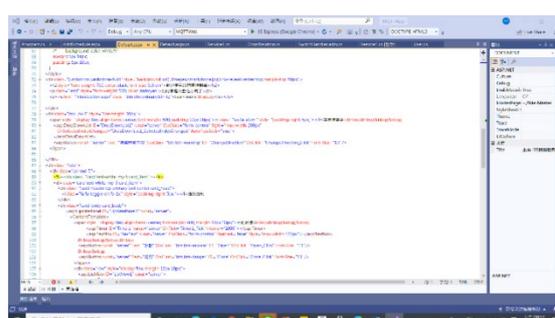


圖 2 Visual Studio 網頁前端設計

### 三、電路雕刻

由於線材繁複，為了節省空間，決定利用高二電子學實習學習過的 Altium Designer，自製電路板以節省線材與空間。

首先在 Altium Designer 中，繪製出所需的電路後，透過自動佈線，完成 PCB 電路板，利用電路板雕刻機刻出所需的電路板，最後進行焊接，如圖 3，不只有效減少電路面積，更能降低線路錯誤的可能性。

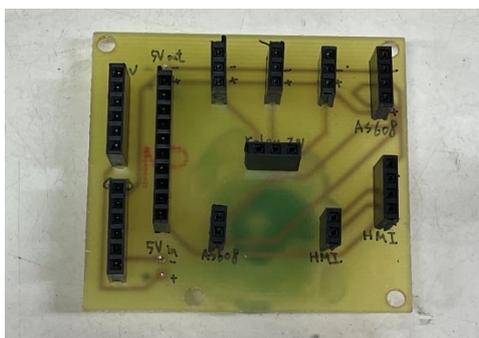


圖 3 電路板成品

#### 四、成品外觀

本次設計的手機箱是使用木材與 3D 列印製作而成，為了方便收納與放置，手機箱尺寸依照衛生紙盒的大小進行設計，尺寸大約為兩個衛生紙盒，可以輕鬆放置在書桌上，或是家中的任何一個小角落，如圖 4。



圖 4 手機箱成品

#### 肆、研究方法

##### 一、研究流程

##### (一)、研究步驟

七月初決定專題題目後，便著手開始蒐集資料、討論手機箱體的功能與大致結構。構想決定後，分工合作進行機構、程式的製作，在使用者立場新增視覺體驗優良的 HMI，以 HMI 介面帶給使用者最佳的顯示頁面，時間分配如表 1，研究步驟如圖 5。

表 1 時間分配表

	7月	8月	9月	10月	11月	12月	1月
1.資料蒐集							
2.元件採購							
3.外殼設計							
4.程式撰寫							
5.通訊連接							
6.電路製作							
7.成品測試							

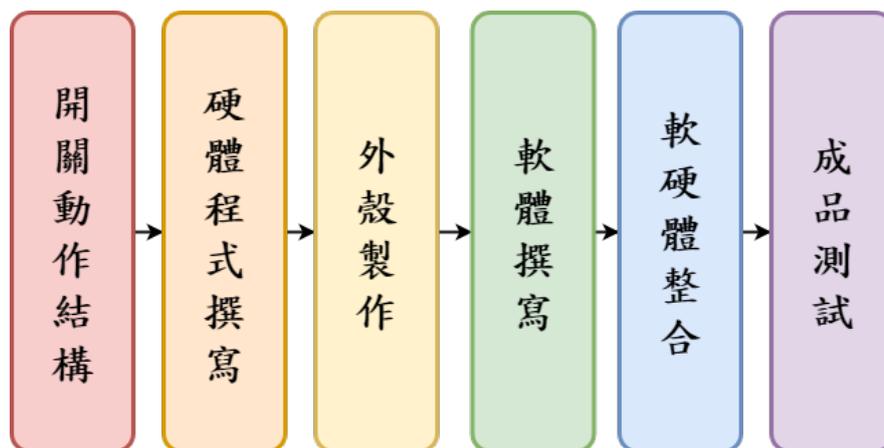


圖 5 研究步驟

## (二)、操作步驟

因為本次專題分別由許多平台構成，故流程圖皆以子流程分述呈現。

### 1、解鎖功能

#### (1) 遠端解鎖流程

開啟 APP，按下解鎖按鈕，顯示解鎖成功圖示後，將數據更新至資料庫，箱子偵測到資料庫數據更新後即自動開門，如圖 6。

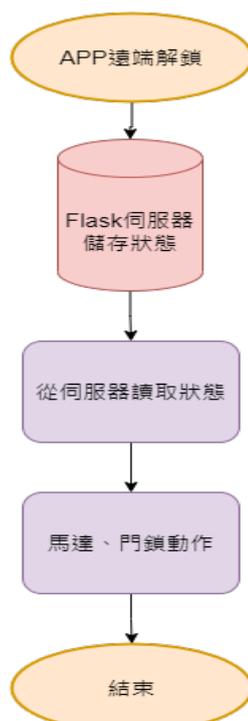


圖 6 APP 遠端解鎖

#### (2) 定時解鎖流程

開啟 APP，按下定時解鎖按鈕，設定解鎖時間的頁面便會跳

出，設定時間後按下確認，便顯示時間設定成功圖示，將數據更新至資料庫，箱子偵測到資料庫數據更新後，到達設定時間即自動開門，如圖 7。

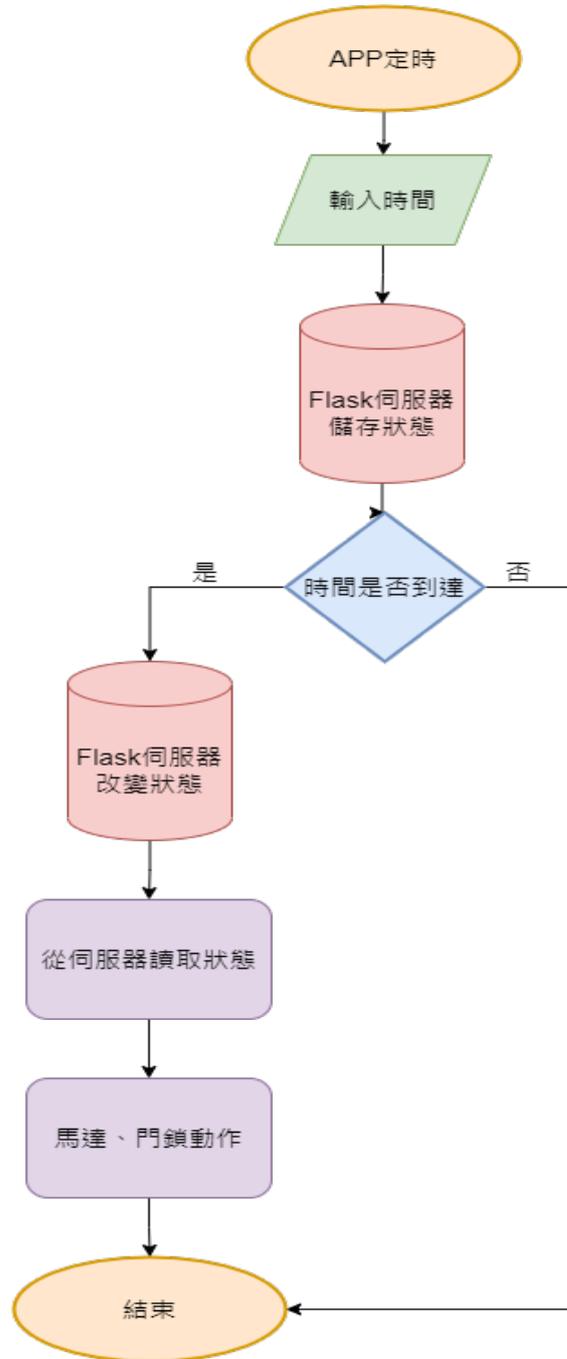


圖 7 APP 定時解鎖

### (3) 本地解鎖流程

啟動 APP，按下註冊指紋按鈕，跳出註冊模式開啟成功圖示，將數據更新至資料庫上，箱子偵測到資料庫數據更新後，啟動指紋模組註冊模式，將手指靠上進行註冊。註冊完成後即可使用儲存的指紋進行本地解鎖。

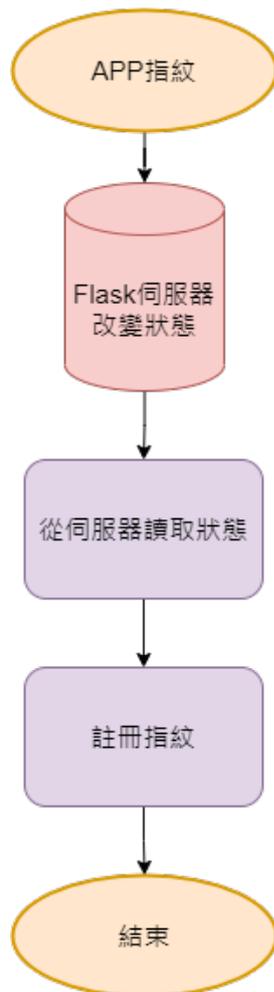


圖 8 註冊指紋

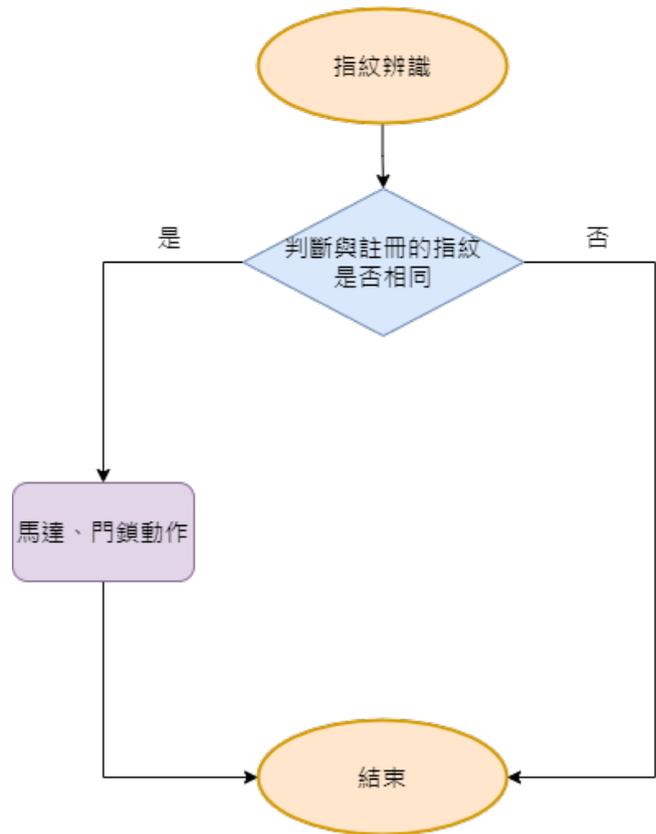


圖 9 指紋辨識

## 2、其他功能

### (1) 更新讀書計畫流程

啟動 APP，輸入當天的讀書計畫，按下上傳讀書計畫按鈕，跳出科目設定成功圖示，將科目儲存至伺服器數據庫中，HMI 即會根據時間顯示先前設定之不同時段的學習科目。

### (2) 智能消毒流程

將手機放入箱子中，並將門關上，即會自動開啟紫外光消毒功能。

### (3) 箱體狀態監控流程

檢流計會根據無線充電模組的輸出電流傳送不同的狀態到伺服器數據庫。APP 則根據數據庫的資料來顯示不同的狀態。以達到確認箱內物品是否為手機，還可以透過 APP 連線箱內鏡頭查看箱內狀態。

## 二、使用材料及工具

### (一)、零件介紹

#### 1、Esp8266 NodeMCU

NodeMCU 是一個開源的 IOT 專案，可使用在多種類型的單晶片開發流程，例如 lua-cjson、spiffs 等等，當然也包含了這塊 Esp8266 NodeMCU 如圖 10，其規格如表 2。

Esp8266 NodeMCU 不只提供充足的外接接腳，還能透過自帶的 micro USB 孔進行 Arduino IDE 程式的編寫，方便操作和優良的擴充性是我們選擇這塊開發版的主要原因。

表 2 Esp8266 NodeMCU 規格

工作電壓	DC 5V(4.5~10V)
數位接腳	10 個
類比接腳	10 個
WIFI 標準	802.11 b/g/n
工作模式	STA/AP/STA+AP
重量	7g



圖 10 Esp8266 NodeMCU

#### 2、Mega2560

「Mega2560」是 Atmel 公司的 MegaAVR 系列中，極具代表性的一項產品。它擁有 256KB 的閃存記憶體，進行程式的編寫和儲存，是目前 Arduino 程式編譯中最泛用的一顆單晶片，程式撰寫容易入門，且擁有大量網路資源及模組，對初學者來說，是一片非常友善的開發板。Mega2560 之規格與外型如表 3、圖 11：

表 3 Mega2560 規格

核心處理器	Mega2560
I/O 數	54
程式記憶體大小	256KB
程式記憶體類型	閃存
電壓 - 電源	DC 7V ~ 12V
時脈速度	16 Mhz



圖 11 Mega2560

#### 3、GA12-N20 金屬微型減速馬達

GA12-N20 為構造簡易且體積非常小的直流減速馬達，如圖 12，規格如表 4，內部含直流電積及減速齒輪組，擁有 0.5kg/cm 的轉舉，足以帶動門軸：

表 3 N20 微型金屬減速馬達規格

產品尺寸	12*9*28mmm
重量	10g
無載速度	25rpm
產品轉矩	0.3 kg/cm
工作電壓	DC 3V
工作電流	0.15A



圖 12 N20 微型金屬減速馬達

#### 4、復位開關(微動開關)

復位開關用於確定門轉到目標位置，並加以控制機構運行如圖 13。

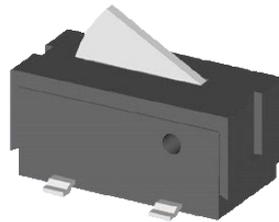


圖 13 復位開關

#### 4、5V1A 無線充電發射裝置

無線充電發射模組放置在隔板上，將支援無線充電的裝置放在上面，就能提供使用者無線充電功能，如圖 14，其規格如表 5。

表 5 5V 1A 無線充電發射模組規格

PCB 尺寸	3.2×3.2 mm
線圈直徑	5 cm
重量	2 g
工作電壓	DC 5 V
最大輸出	5 W
傳輸距離	2~8 mm



圖 14 5V 1A 無線充電發射模組

### 5、電源供應器(LRS-50-12)

電源供應器提供 12V 電源，12V 用以驅動電磁鎖、主控板最大可輸出 50 瓦特，其規格、外型如表 6、圖 1。

表 6 LRS-50-12 規格

輸入電壓	85~264VAC
輸出電壓	DC12V
額定功率	50.4W
頻率範圍	47~63Hz



圖 15 LRS-50-12

### 6、USART HMI 智能串口屏(5 吋 TJC3224T050\_011R)

HMI 用於箱體上顯示讀書計畫、當前時間，以提醒使用者保握時間，其規格、外型如表 7、圖 16。

表 7 USART HMI 規格

亮度	300nt
解析度	480*800
可視角度	135 度
對比度	1：500
響應速度	0.1s



圖 16 USART HMI

### 7、UVC 燈管(Philips UVC 254NM T5 4W)

紫外線消毒燈對任何細菌或病毒都有效，集中高強度紫外線在短時間內即可殺菌，如圖 17，其規格如表 8。

表 8 UVC 254NM T5 規格

工作電壓	AC 110V
適用功率	4W
波長	253.7nm
管徑	T5，15mm



圖 17 UVC 254NM

### 8、電磁鎖(LY-03)

電磁鎖依靠電力產生磁力，使鎖頭縮回，我們藉由繼電器來進行控制，如圖 18，其規格如表 9。

表 9 LY-03 規格

產品尺寸	22*28*55mmm
工作電壓	DC 12V
額定電流	0.3A



圖 18 LY-03

## 9、電流感測計 (ACS712)

ACS712 是基於霍爾檢測的原理，具有體積小的優勢，我們利用它偵測無線充電電流，以偵測手機放入時間，其規格、外型如表 10、圖 19。

表 10 ACS712 規格

產品尺寸	31*13mm
供電電壓	DC 5V
最大電流	20A



圖 19 ACS712

## 10、ESP32-CAM

ESP32-CAM 採用低功耗雙核 32 位 CPU，支持 OV2640、OV7670 鏡頭，我們利用它來顯示箱體內部當前狀態，以辨別箱內物品是否為手機，其規格、外型如表 11、圖 20。

表 11 ESP32-CAM 規格

產品尺寸	27*40.5*4.5mm
CPU	ESP32
ROM	520KByte
通用 IO	9
供電電壓	5V
額定電流	2A



圖 20 ESP32-CAM

## 11、指紋模組 (AS608)

AS608 指紋模組集成光路和指紋處理部分的一體化指紋處理模塊，具有體積小、功耗低、接口簡單的特點，乾溼手指適用性良好，指紋搜索速度快。其規格、外型如表 12、圖 21。

表 12 AS608 規格

產品尺寸	23.4*20.3*48.1mm
供電電壓	DC 3.3V
供電電流	<60mA
窗口面積	15.3*18.2mm



圖 21 AS608

## (二)、機構原理

利用「NodeMCU」與「Mega2560」偵測伺服器狀態，並控制馬達的旋轉，當按下解鎖後，門鎖縮入，馬達帶動門軸轉動，就可以輕鬆推動門。

### (三)、軟體介紹

#### 1、Autodesk Inventor

Autodesk Inventor 圖 22，是一款能夠藉由曲線以及曲面建構 2D 草圖及 3D 物件的繪圖軟體，廣泛應用於工業設計、美術設計、玩具及建築相關等領域。內建曲面工具可以精確地製作動畫、工程圖、分析評估以及生產用的模型。繪製的物件 搭配 3D 列印機，能夠快速地將使用者的構思實體化，非常便捷。Autodesk Inventor，繪製 3D 模型，最後由 3D 列印機印製出成品圖 23。



圖 22 Autodesk Inventor logo

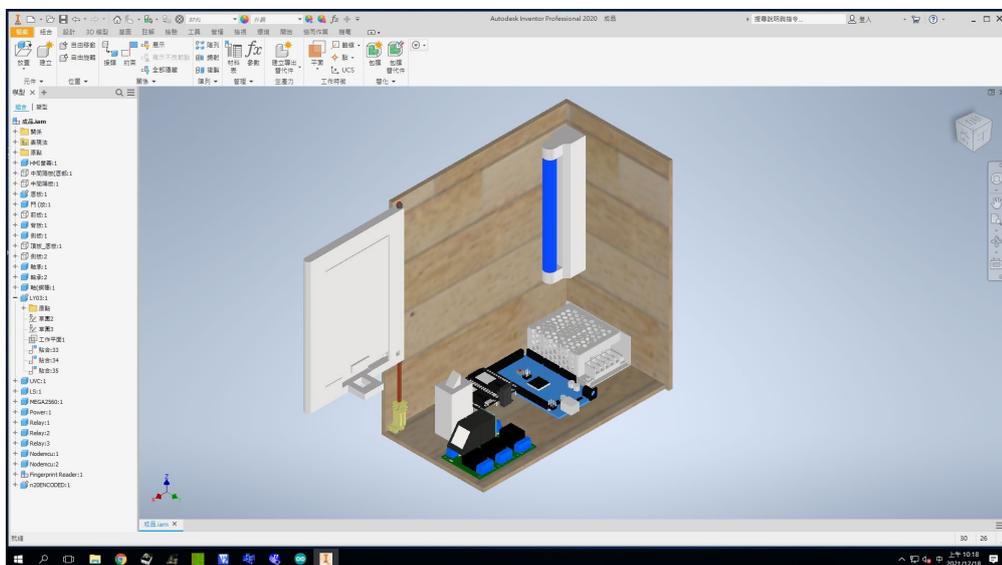


圖 23 Autodesk Inventor 3D 圖繪製介面

#### 2、Arduino IDE

Arduino IDE 是一個開放原始碼的硬體程式語言編寫軟體，它兼具類似 java、C 等後端伺服器語言的開發環境，且擁有許多已模組化的套件與函式庫，提供初學者使用。由於它在控制單晶片的方便性，可以輕鬆連結硬體套件及通訊系統，所以我們選擇 Arduino IDE 作為硬體程式編寫的軟體。



圖 24 Arduino logo

```
mega | Arduino 1.8.9
檔案 編輯 硬體 工具 說明
mega
delay(200);
digitalWrite(24, HIGH);
delay(400);
digitalWrite(24, LOW);
delay(1000);
digitalWrite(22, LOW);
update_data("get_data");
delay(5000);
update_data("get_data");
}
//指紋辨識
if (fp_status == 0) {
  if (getFingerprintIDez() == 1) {
    Serial.println("success");
    digitalWrite(26, LOW);
    digitalWrite(22, HIGH);
    delay(200);
    digitalWrite(24, HIGH);
    delay(400);
    digitalWrite(24, LOW);
    delay(1000);
    digitalWrite(22, LOW);
  }
}
//指紋註冊
if (fp_status == 1) {
  update_data("get_data");
  while (getFingerprintEnroll() != true);
  delay(15000);
}
```

圖 25 Arduino 程式撰寫



圖 26 Arduino 程式撰寫與接線測試

### 3、Altium Designer

Altium Designer 是由 Protel 的軟體開發商：Altium 公司推出的電子電路開發系統，這套軟體能進行原理圖設計、電路仿真模擬、PCB 繪製、電路板自動佈線、信號完整性分析和設計輸出等技術，為使用者提供了一個簡便的電子電路設計方式，輕鬆進行繁雜的電路板設計，熟練使用這一軟體將使電路設計的質量及效率大大提升。



圖 27 Altium Desiner logo

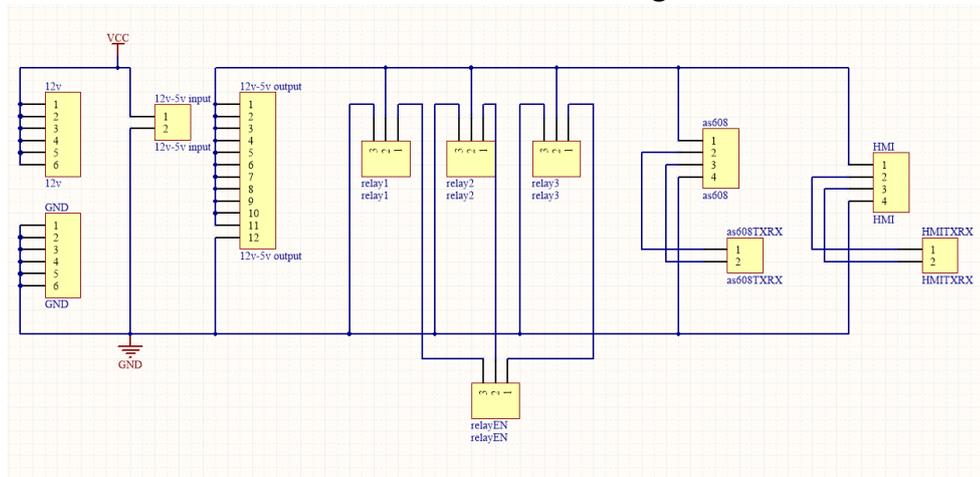


圖 28 成品電路圖

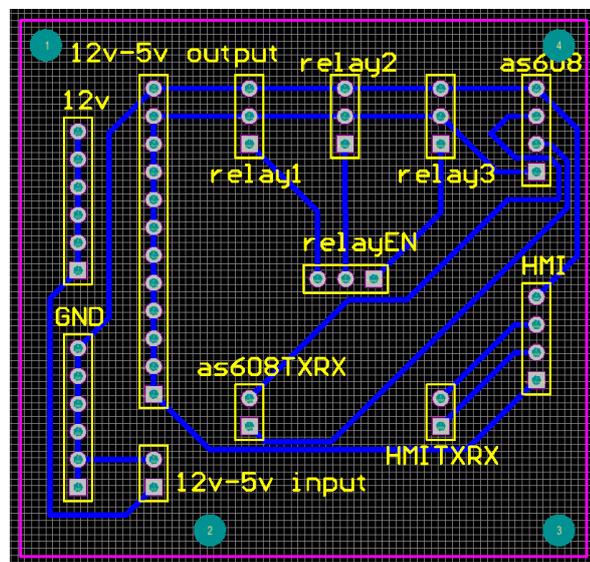


圖 29 成品雕刻圖

#### 4、Flask

Flask 是一個使用 Python 編寫的輕量級 Web 應用框架。基於 Werkzeug WSGI 工具箱和 Jinja2 模板引擎，並使用 BSD 授權。

Flask 被稱為「微框架」，因為它使用簡單的核心，用擴充增加其他功能。Flask 沒有預設使用的資料庫、表單驗證工具。然而，Flask 保留了擴增的彈性，可以用 Flask-extension 加入這些功能：ORM、表單驗證工具、檔案上傳、各種開放式身分驗證技術。



圖 30 Flask logo

#### 5、App Inventor

Android 應用開發者 (App Inventor) 是一款卡通圖形界面的 Android 智慧型手機應用程式開發軟體，它可以讓任何熟悉或不熟悉程序設計的人來創造基於 Android 作業系統的應用軟體。它使用圖形化界面，用戶可以拖放圖形對象來創造一個運行在安卓系統上的應用，它就可以在許多手機設備上運行。



圖 31 App Inventor logo

#### 6、Fritzing

Fritzing 是一款用於設計 PCB 印刷電路板的軟體，簡單好用的操作介面，以及資源豐富的元件庫，讓即使是沒有電機背景的使用者，也可以輕鬆繪製出理想中的電路圖，且如果是預設零件庫中不包含的元件，在網路上也可以找到許許多多的零件圖，用以匯入資料庫。



圖 32 Fritzing logo

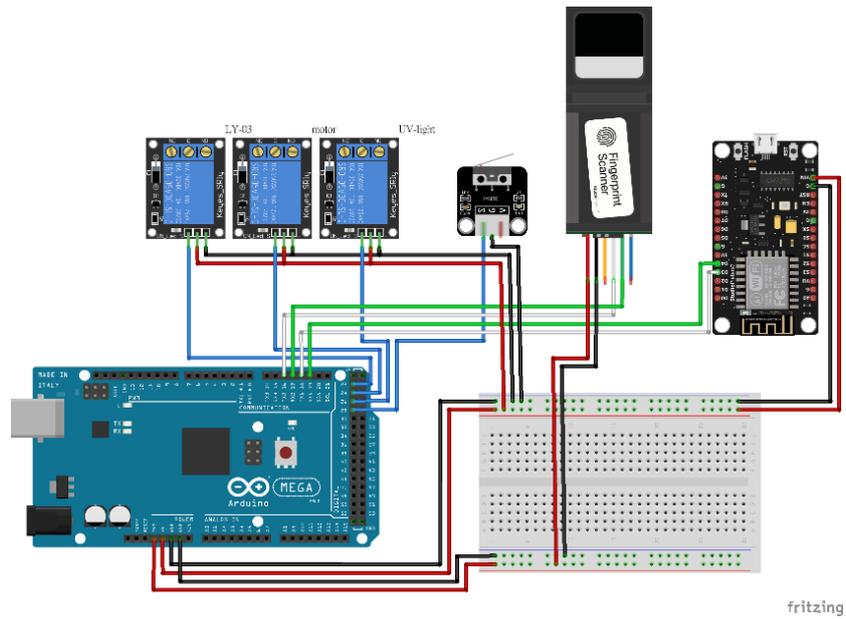


圖 33 Fritzing 成品接線圖

## 伍、研究結果

本專題結構可以分成硬體結構與軟體通訊架構，以下所示。

### 一、硬體結構

本次專題結構，如圖 34

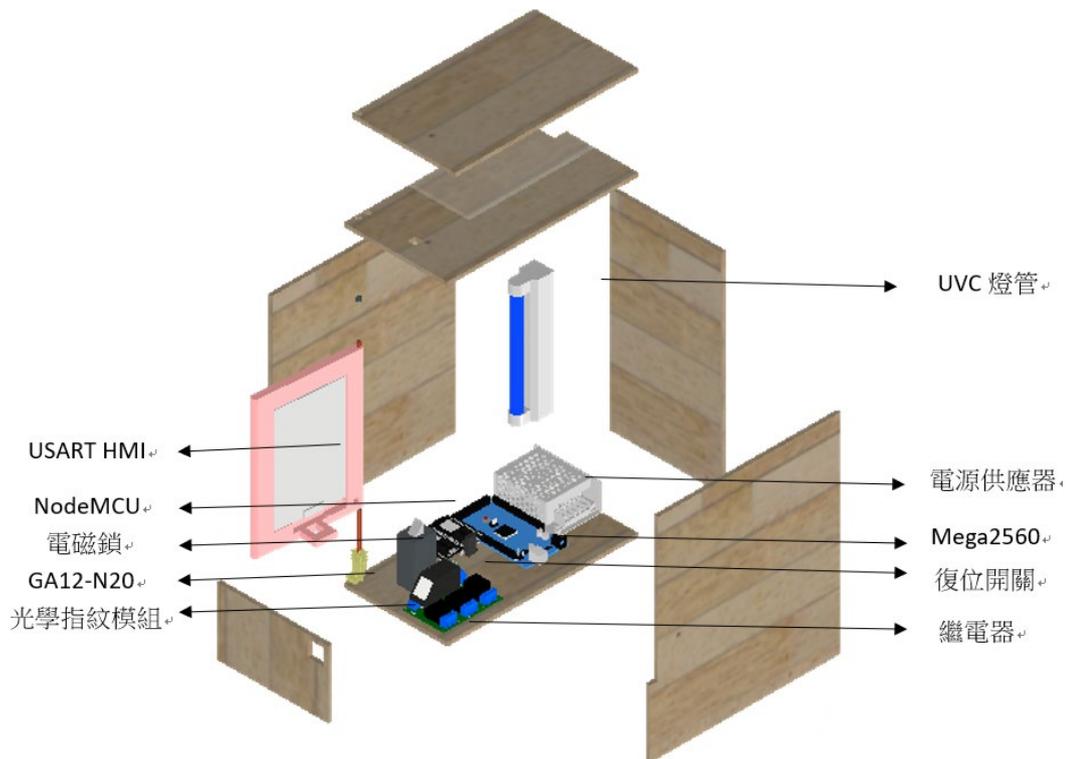


圖 34 Inventor 爆炸圖

### (一)、轉軸

透過 3D 列印製作的聯軸器(圖 35)，將 N20 減速馬達裝置上去，即可利用聯軸器將馬達的轉軸與門軸連結。



圖 35 聯軸器

### (二)、門

透過 3D 列印製作的門(圖 36)，做出 HMI 上的開口、電磁鎖的鎖頭，再藉由門軸轉動門板，進行開門。



圖 36 門板

## 二、軟體通訊架構

網路是現代人日常生活中不可或缺的一部分，網頁更是重中之重，無論是手機、平板、電腦，都可以使用瀏覽器上網，所以我們選擇網頁及 APP 作為使用者控制端的平台，透過 Flask 伺服器與 NodeMCU 進行通訊。本專題通訊關係，如圖 37。

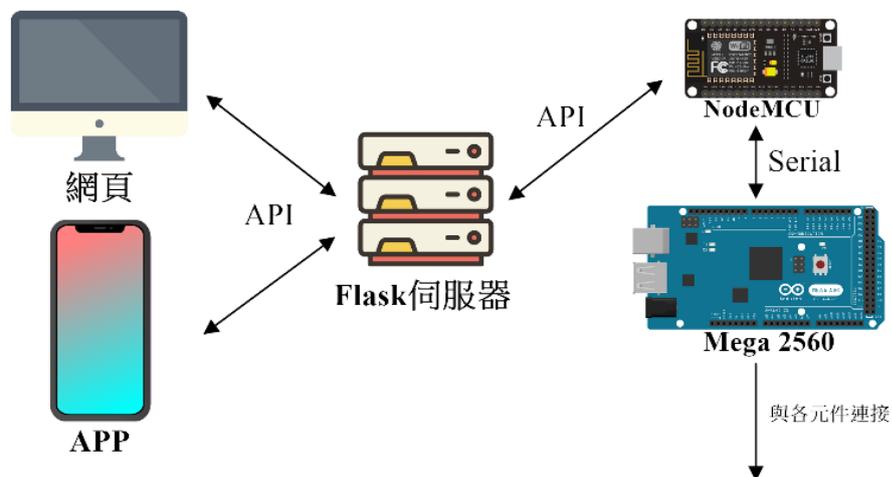


圖 37 通訊關係圖

### (一)、Flask 伺服器及 API

我們使用 Python 撰寫基於 Flask 框架的伺服器，並架設於家中。利用端口轉發使網頁及 APP 能從廣域網路連接到家中的區域網路，實現真正的遠端控制。

在 API 連接的部分，我們使用 HTTP GET 與伺服器溝通，並以 Json 資料格式傳送資料。傳送資料方面，使用更改 API 網址內容的方法上傳資料至伺服器。接收資料則是訪問 API 即會返回 Json 資料，供 APP、網頁及 NodeMCU 使用。

### (二)、Serial

Serial 為 Arduino 元件互相溝通的方式之一，藉由 Tx、Rx，使 Mega2560 與 NodeMCU 進行溝通，使 Mega2560 能透過 NodeMCU 與伺服器進行資料交換。

## 三、成果展示

### (一)、APP 介面

APP 分為三部分，下方為切換頁面，在主頁有遠端解鎖、定時解鎖及註冊指紋讀書計畫頁面，可更改每個時段的讀書計畫並上傳，於 HMI 顯示，鏡頭監控頁面，可以顯示手機箱內當前狀態，以確認箱內物品是否為手機，如圖。



圖 38 APP 主頁

圖 39 APP 讀書計畫

圖 40 APP 監控畫面

### (二)、網頁介面

網頁分三部分，左邊為切換頁面，右邊為組員介紹及專題簡介，上方為檢流計所偵測手機放入時間柱狀圖，以了解讀書效率，主頁可遠端解鎖及註冊指紋，讀書計畫頁面可更改每個

時段的讀書計畫並上傳，於 HMI 顯示，鏡頭監控頁面，顯示手機箱內當前狀態，以確認箱內物品是否為手機



圖 41 網頁左右側頁面



圖 42 網頁主頁



圖 43 網頁讀書計畫

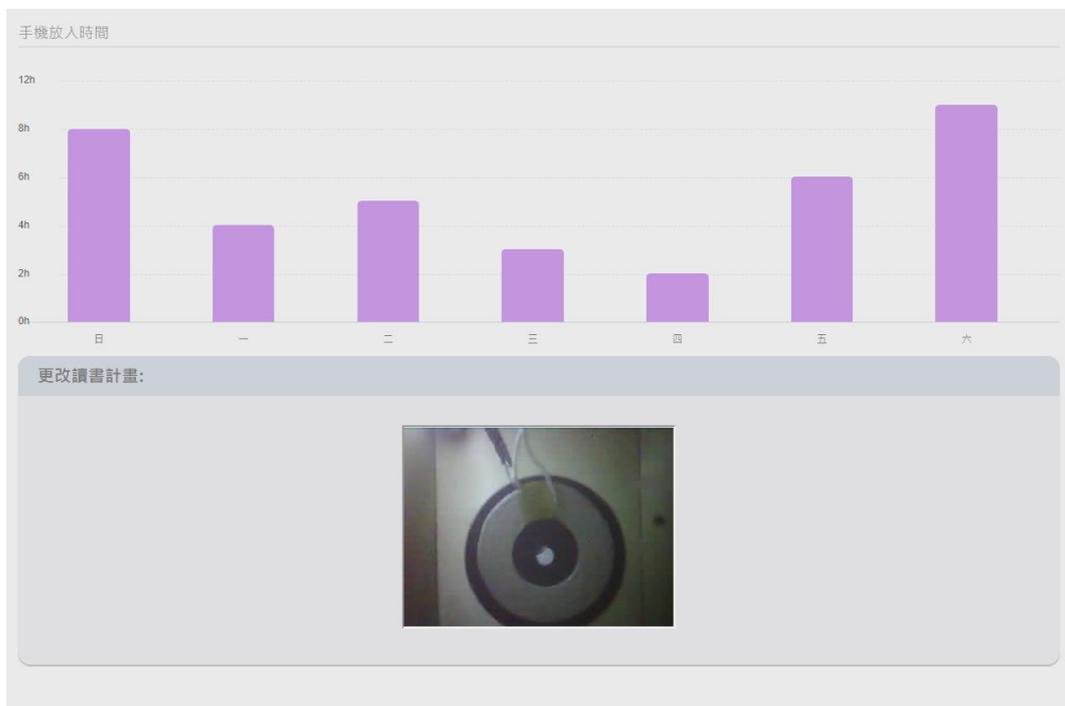


圖 44 網頁鏡頭監控

## 陸、討論

### 一、接腳不足

一開始，我們選用 NodeMCU 作為主控板，但因為元件數量過多，所需的接腳數也跟著上升，導致 NodeMCU 所提供的接腳數量不夠多，所以我們改選用 Mega2560 作為主控，網路功能則使用 Esp01 與 Mega2560 進行連接。

### 二、硬體限制

我們在硬體上使用了「Esp01」這塊板，但其在使用 AT 指令韌體時，Mega2560 從與其通訊到連接、回傳伺服器回應的過程，所耗費的時間不是很理想。

所以我們改為使用 Mega2560 與 NodeMCU 進行搭配，並將其二者的通訊方式改為使用序列埠溝通，使得與連線到伺服器以及後續的流程得以從主程式中獨立出來，以改善主程式的執行速度。

### 三、通訊處理

原本我們是使用 ThingSpeak 伺服器進行資料更新，但經過多次測試下來，我們發現其伺服器資料更新有著速率上的限制（15 秒一筆），這導致我們在進行撰寫程式的過程中，需要將更新資料的延遲時間考慮進我們的程式中，大幅降低程式在執行上的效率。

所以我們在後續，將 ThingSpeak 改為 Flask（web 應用框架）進行使用，透過其自行架設的伺服器，用以改善、提升伺服器回應的速率，讓整體程式得以用更高效的方式進行運作。

## 柒、結論

在經過組員間的討論，以及使用場合的種種因素下，我們最後決定使用兩個衛生紙盒作為我們專題結構的大小，大大減少了占用的空間。自動門的部分採用聯軸器及上下各一的軸承，使用 N20 馬達去帶動門的旋轉，進而達到自動開門的動作。而因應疫情的原因，我們在箱體內部也加入了 UVC 紫外光燈管搭配復位開關進行手機消毒，在確認箱門確實關上後，觸發復位開關才會將紫外燈開啟進行消毒，進而保護使用者的眼睛。而在門上我們使用了嵌入式的 HMI，將讀書計畫顯示在 HMI 上，提醒使用者當下該讀什麼科目。而為了預防被管理者將展示機亦或者備機放入箱子內，我們使用了 ESP32cam 和檢流計進行監控。

而我們遠端操控介面是使用 App Inventor 進行製作，但考慮到可能有 iOS 系統的使用者，所以我們另外寫了網頁，使 iOS 系統使用者即使無法使用 APP，也可以透過網頁去進行遠端操控。但我們的作品仍有許多

方面需要進行改進，包括備用電源、警報音提示。因為我們所使用的 LY-03 是通電縮入型，所以一旦停電，就必須等到復電後才能進行手機取出的動作。

我們的專題從零開始，從無到有，其中每一位組員的用心付出，不管是其中負責程式撰寫，又或是負責結構設計，都能互相學習討論，就像是一片片的拼圖。缺一不可，缺少一片就不完美，我們各自去學習專題所需的各項專業技能，漸漸拼湊出最後的成品，完美靈活的去運用我們所學習的專業能力。組員之間也都願意體諒與包容，學習團隊合作的真正精神，同心協力成為每次破關斬將的力量。在製作專題的過程中找尋不同領域專業人士的協助，能提供我們更多的想法與意見，來改變專題有所不足的地方；也學習當面對問題時，該如何解決問題，一旦上網查詢，便能快速的找到問題的解答；從網路搜尋資料的技能也提升很多，發現網路能提供的資源應有盡有，擴充專題的完整度。專題能將我們所學應用在實體上，並且轉換成對生活有幫助的技能，都是我們進步的動力。我們的專題除了能「禁機」，又能「晉級」，我們只會越來越好，去學習各式各樣的能力，成為真正的巨人。

影片原始畫質請參考網址：<https://youtu.be/D59Dxo0DUh4>

## 捌、參考資料及其他

### 一、書本資料

1. 黃穎豐、陳明鈺(2019)。Autodesk Inventor 2018 特訓教材基礎篇。全華圖書股份有限公司。
2. 張榮洲、張宥凱(2020)。電子電路及 Arduino 應用。全華圖書股份有限公司。
3. 趙英傑(2016)。超圖解物聯網 IoT 實作入門。旗標科技股份有限公司。

### 二、網路資料

1. Arduino 筆記(89)：ESP32CAM 初次設定並透過網頁瀏覽影像。取自：<https://atceiling.blogspot.com/2020/11/arduino89esp32cam.html>
2. ESP8266-NodeMCU 硬件參考。取自：  
<http://www.taichi-maker.com/homepage/reference-index/arduino-hardware-refrence/nodemcu/>
3. AS608 Fingerprint Sensor Module With Arduino。取自：  
<https://randomnerdtutorials.com/fingerprint-sensor-module-with-arduino/>
4. Flask：基礎網頁製作。取自：  
<https://ithelp.ithome.com.tw/articles/10222132>
5. USART HMI 智慧串列屏介紹。取自：  
<https://www.itread01.com/content/1548685447.html>
6. HMI 程式書寫方法。取自：  
<http://wiki.tjc1688.com/doku.php?id=6.%E6%8C%87%E4%BB%A4%E9%9B%86:2.%E4%B9%A6%E5%86%99%E8%AF%AD%E6%B3%95>