

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：電機電子群

科別：冷凍空調科

作品名稱：智慧紀錄電子鎖

關鍵字：遠端監控、電子鎖、拍照上傳



組員：黃言禎、楊少杰、陳柏羽、劉其蒙
指導老師：歐鎮寬老師

目錄

壹、摘要	1
貳、研究動機	1
一、傳統鑰匙的麻煩	1
二、紀錄功能	1
三、多元的開鎖方式	1
四、即時監控功能	1
參、主題與課程之相關性	2
一、智慧居家實習	2
二、電子學電路實習	2
三、電工實習	2
肆、研究過程及方法	2
一、研究過程	2
二、研究方法	3
伍、使用材料	3
一、nodemcu-32s	3
二、SSD1306 OLED0.96 吋	4
三、4x3 薄膜鍵盤	4
四、RFID-RC522	5
五、PCF8574	5
六、有源蜂鳴器	5
七、esp32cam	6
八、繼電器	6
九、電磁鐵	6
十、HC-SR04	7
十一、330Ω 電阻* 3 黃綠紅 LED 燈各* 1	7
十二、飛機木片	7
陸、研究結果	8
一、流程圖	8
二、實際展示	9
(一)、模型部分	9
(二)、門鎖部分	10
(三)、鏡頭部分	14
柒、問題與討論	16
一、電力	16
二、ESP32 晶片問題	18
三、薄膜鍵盤接腳過多	18
四、拍照傳 LINE 會有時差	18

五、自動門.....	18
捌、結論	18
玖、未來展望	18
拾、參考資料及其他	19

圖目錄

圖 1、研究過程.....	2
圖 2、nodemcu-32s[1]	3
圖 3、SSD1306 OLED0.96 吋[2]	4
圖 4、4x3 薄膜鍵盤[3]	4
圖 5、RFID-RC522[4]	5
圖 6、PCF8574[5].....	5
圖 7、有源蜂鳴器[6].....	5
圖 8、esp32cam[7]	6
圖 9、繼電器模組[8]	6
圖 10、電磁鐵.....	6
圖 11、HC-SR04[9]	7
圖 12、330Ω 電阻* 3 黃綠紅 LED 燈各* 1	7
圖 13、飛機木片[10].....	7
圖 14、esp32 流程圖.....	8
圖 15、esp32cam 流程圖	8
圖 16、模型正面.....	9
圖 17、模型俯視.....	9
圖 18、模型背面.....	10
圖 19、偵測 RFID.....	10
圖 20、密碼輸入正確的結果.....	12
圖 21、密碼輸入錯誤的結果.....	12
圖 22、具有正確 UID 的磁卡感應結果.....	12
圖 23、具有錯誤 UID 的磁卡感應結果.....	13
圖 24、使用 APP 當中的 ON 來開門	13
圖 25、按下 APP 的 ON 後的結果	13
圖 26、當閃光燈連續閃爍就代表已成功連上 WiFi	14
圖 27、跳出新視窗後即可查看即時影像.....	16
圖 28、汽車 12V 電瓶.....	16
圖 29、12V/5V 變壓器	17
圖 30、應變停電時所設計的電路圖.....	17

壹、摘要

為了應對生活忙碌導致人會忘記帶鑰匙，或是鑰匙突然不見的問題，現代問題用現代手段來解決，乾脆把傳統的鑰匙鎖換成科技化的鎖，它不需要實體傳統鑰匙控制門鎖而是用 RFID、密碼鎖還有利用手機 APP 透過近幾年的趨勢——物聯網來控制門鎖，而且我們的智慧門鎖還有紀錄的功能，在門上加裝超音波感應來觸發鏡頭拍照然後上傳到 LINE 來通知使用者，並且鏡頭也支援即時監控，只要打開手機 APP 或是輸入相對應的網址即可查看。我們運用較為簡單的 Arduino 來寫我們所有的程式，APP 的程式則是用 MIT App Inventor。

貳、研究動機

一、傳統鑰匙的麻煩

傳統鑰匙的麻煩在於會被忘記帶，而且它是金屬製成所以笨重，它是唯一一個開鎖方式，所以一旦忘記帶鑰匙或是不見的話就回不了家了。

二、紀錄功能

可以運用在私人或是公共場所，控管人員的流動並且可以記錄進出入的時間，還能拍照留存。

三、多元的開鎖方式

隨著科技日益發達，傳統門鎖逐漸被科技化，不再被侷限於只有一把鑰匙才能開門，而是有不同的方式甚至能透過手機 APP 來控制。

四、即時監控功能

科技化的時代是讓人們變得更便利，我們不再需要親自到大門前觀看門前是否有人或是出了什麼狀況，而是可以透過手機 APP 或網頁顯示出的即時監控畫面。

參、主題與課程之相關性

一、智慧居家實習

我們的專題運用了在高三上的「智慧居家實習」所學到的知識，就是運用 Arduino 來撰寫我們所需要的程式，Arduino 是相對較入門且易於理解的軟體，我們利用上課所學到的語法、範例來成功做出我們的智慧紀錄電子鎖。

二、電子學電路實習

元件之間的連接則是運用了高二的「電子學電路實習」所學到的知識，如何在麵包板上接線、電子零件的測量都是在實習課學到的，而且在這個實習當中學到的知識有助於我們理解元件的功能。

三、電工實習

在高一的「電工實習」有學到一些配線的相關知識還有如何設計電路，因此我們才有辦法設計電路並實作出來，解決了在本專題遇到的問題——當遇到停電的問題時怎麼辦？

肆、研究過程及方法

一、研究過程

我們有紀錄一份專題日誌，詳細的過程可參考附件一

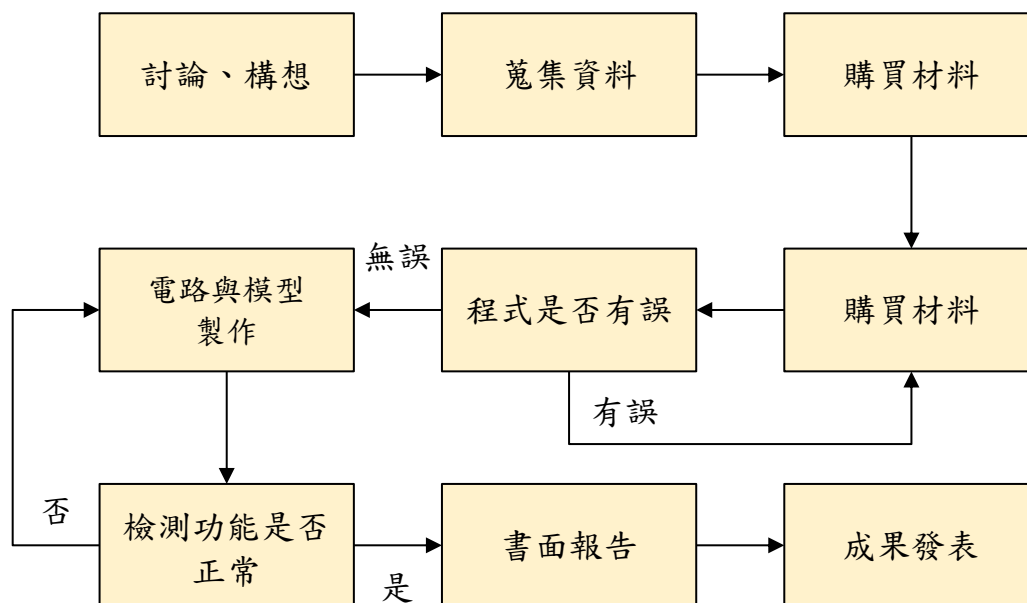


圖 1、研究過程

二、研究方法

為了要做出我們所構想的功能，像是我們要有密碼開鎖、RFID 開鎖還有手機開鎖的功能，我們首先會上網找看有沒有類似的範例或教學，參考它們的程式碼並用 Arduino 加以修改，透過 Arduino 上傳至晶片然後在麵包板上測試元件是否可以正常運作之後，我們就把程式整合在一起如附件二，程式總共有 770 行，是因為我們把所有元件都整合在一起，而整合在一起又會有邏輯的問題，所以我們不斷的上網查詢相關的資料、上國外的論壇、相關書籍或是詢問老師，來解決程式的問題。

伍、使用材料

一、nodemcu-32s (以下皆以 esp32 簡稱)

ESP32 是一系列低成本，低功耗的單晶片微控制器，整合了 WiFi 和雙模藍牙，腳位更是豐富多元，而我們所使用的 nodemcu-32s 就是 ESP32 家族的一員。



圖 2 、nodemcu-32s[\[1\]](#)

二、SSD1306 OLED0.96 吋

OLED 的優點在於顯示效果非常漂亮，還能顯示圖片，尺寸雖然不大，但完全不影響閱讀效果，我們選用了這個顯示屏來顯示門的狀態或是晶片的狀態。

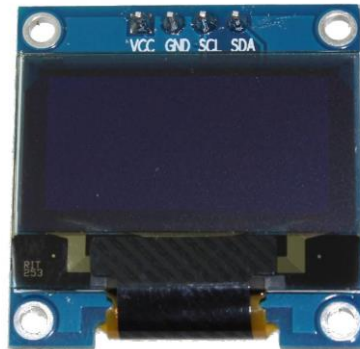


圖 3、SSD1306 OLED0.96 吋[\[2\]](#)

三、4x3 薄膜鍵盤

Arduino 操作輸入的方法還不少，最簡單的當然是按鈕相關的元件了。但如果輸入變得複雜，像是要輸入密碼之類的，可能就要考慮使用鍵盤模組了，選用 4x3 薄膜鍵盤主要是因為它簡單使用。



圖 4、4x3 薄膜鍵盤[\[3\]](#)

四、RFID-RC522

RFID 是許多人日常中不可或缺的裝置，舉凡大樓門禁、電梯權限控制，很多企業也用 RFID 來進行出勤管理。最常用到的 RFID 模組就是 RC522 了，我們用最簡潔的程式，來讀取卡片的 UID，並且和已儲存過的 UID 進行比對。然後再配合 OLED，讓它變成可以實用的系統。



圖 5、RFID-RC522[4]

五、PCF8574

它被廣泛運用在對 I/O 資源緊缺的元件上，而我們使用的薄膜鍵盤接腳過多再加上我們的 esp32 接腳有限，所以需要這個模組來做轉接使接腳的負擔減輕。



圖 6、PCF8574[5]

六、有源蜂鳴器

有源蜂鳴器是內建了一組固定的頻率，只要接通電源，就會發出固定的音調，我們使用它來幫助使用者能夠知道門是否有動作。



圖 7、有源蜂鳴器[6]

七、esp32cam

esp32cam 是 Arduino 當中相對較入門的鏡頭，它連接於一塊小型的 ESP32 板子，雖然腳位不多但仍然擁有許多功能，能夠滿足我們需要的監控跟拍照上傳 LINE 的功能。



圖 8、esp32cam[7]

八、繼電器

繼電器常被運用在實際電路的控制，也易於控制只要給予特定腳位電壓即可動作，這個繼電器模組能夠讀取數位信號，能夠當作開關來控制電磁鐵的吸磁。

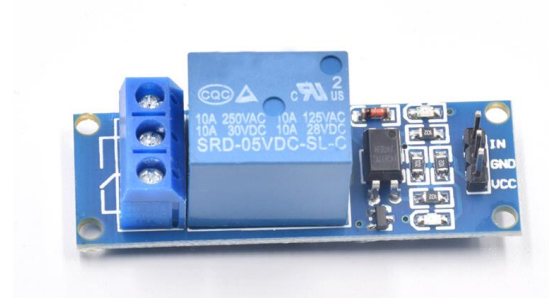


圖 9、繼電器模組[8]

九、電磁鐵

因為我們選擇製作模型來呈現我們的專題，因此我們選擇運用這個小型電磁鐵來當作電磁鎖，規格是 5V 且具有 8 公斤的吸力。



圖 10、電磁鐵

十、HC-SR04

這是超音波模組，它會送出 8 個 40khz 的方波，如果前方有障礙物，信號就會返回，模組收到信號後，再利用返回的時間，去計算該障礙的距離，所以運用這個功能就能夠控制鏡頭的拍照。



圖 11、HC-SR04[9]

十一、330Ω 電阻* 3 黃綠紅 LED 燈各* 1

為了使門動作時，能夠讓使用者更清楚門的狀態所以追加了指示燈功能，我們使用 LED 燈並透過 esp32 來控制其在門動作的時候有不同的效果。

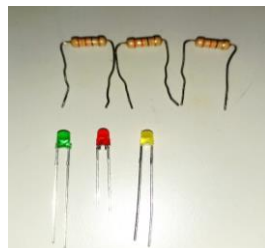


圖 12、330Ω 電阻* 3 黃綠紅 LED 燈各* 1

十二、飛機木片

製作模型用，比起常見的木板它更輕巧且易於組裝。



圖 13、飛機木片[10]

陸、研究結果

一、流程圖

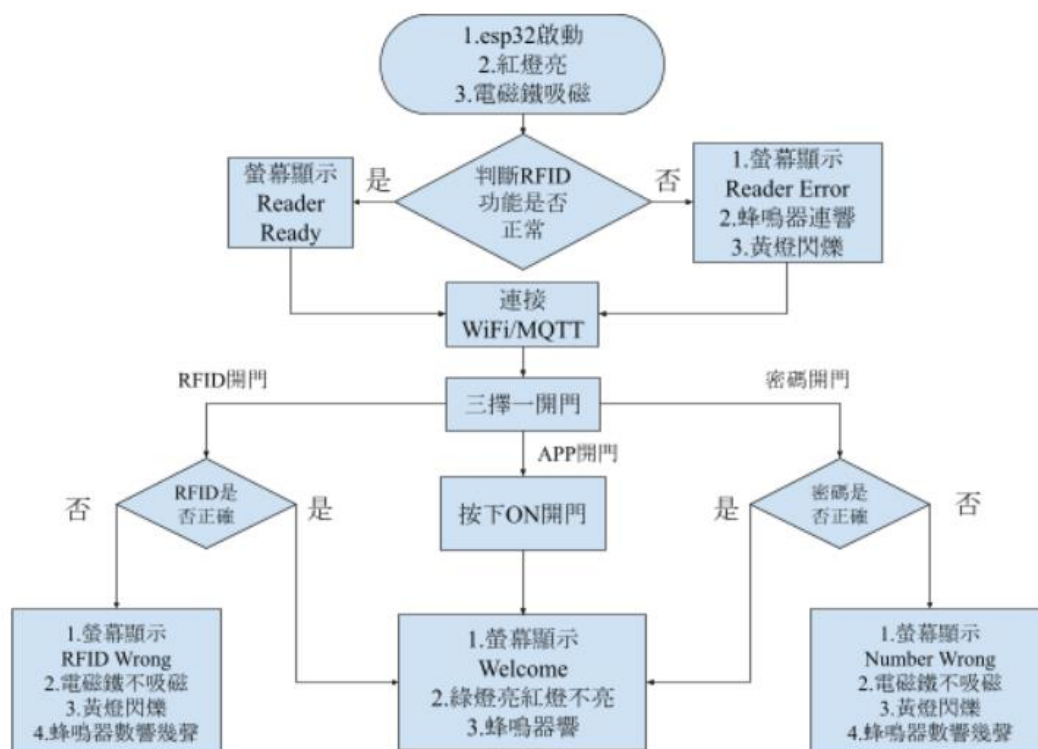


圖 14、esp32 流程圖

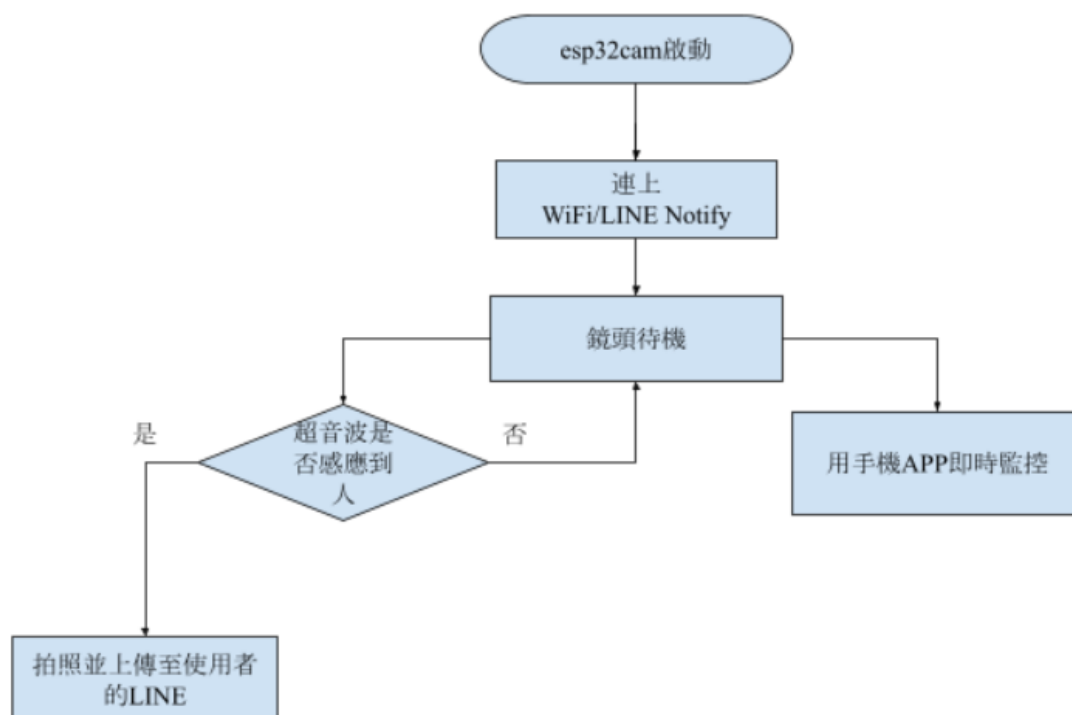


圖 15、esp32cam 流程圖

二、實際展示

(一)、模型部分

1、模型正面



圖 16、模型正面

2、模型俯視(屋頂是可以拿起來的)

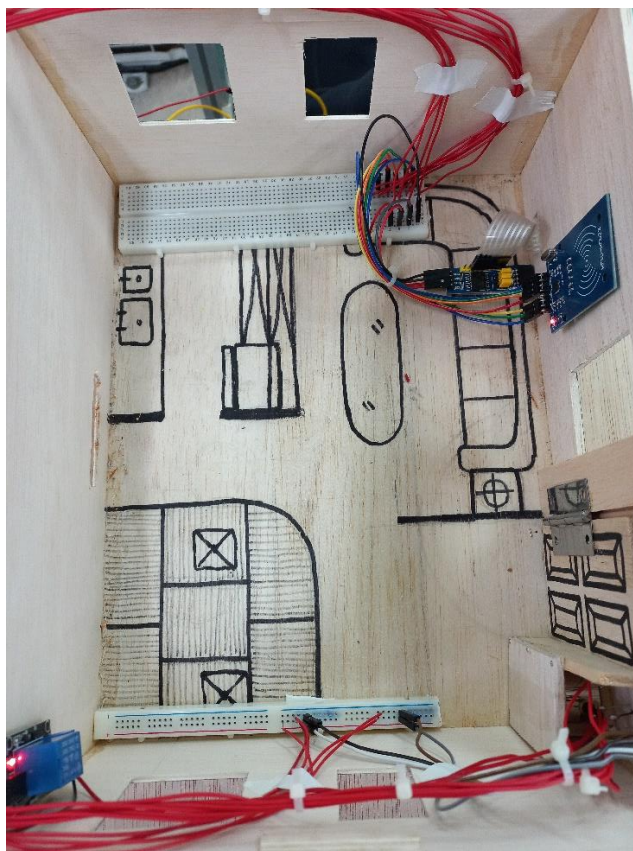


圖 17、模型俯視

3、模型背面

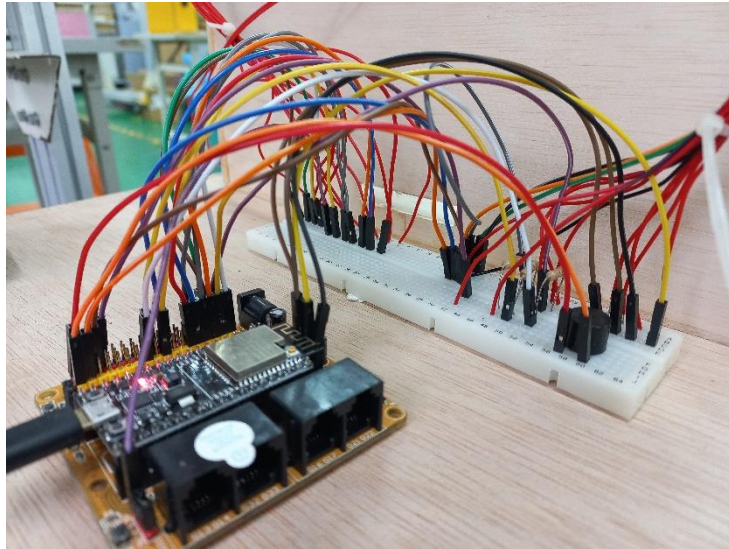


圖 18、模型背面

(二)、門鎖部分

1、偵測 RFID 功能是否正常。

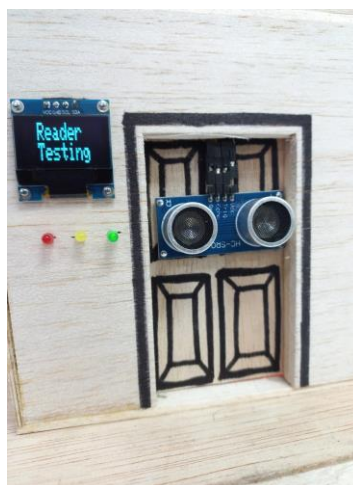


圖 19、偵測 RFID

2、偵測功能正常顯示 Ready，不正常則顯示 Error。

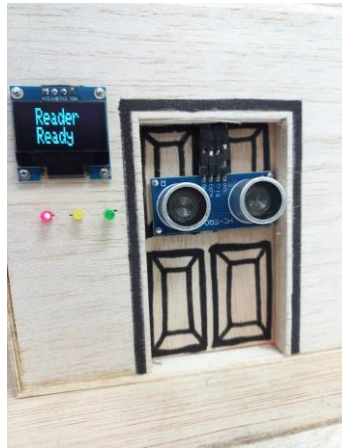


圖 20、RFID Ready

3、等待晶片連上 WiFi/MQTT。



圖 21、晶片連接 WIFI/MQTT

4、三擇一開門 - 密碼鎖

輸入一串設定的密碼後電磁鐵不吸磁即可打開門，同時螢幕顯示 Welcome、LED 綠燈亮還有蜂鳴器響。

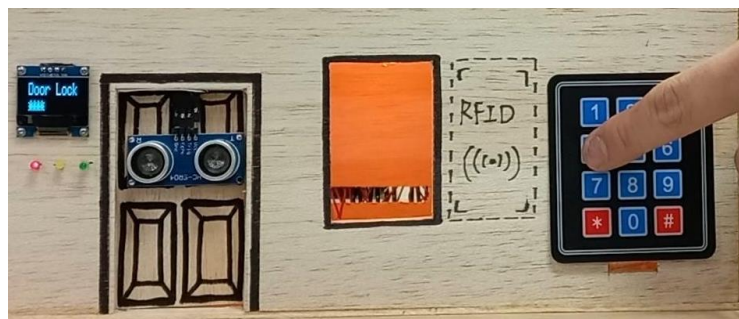


圖 22、輸入一串設定的密碼



圖 20、密碼輸入正確的結果

若是密碼輸入錯誤，電磁鎖依舊吸磁且螢幕顯 Number Wrong、LED 黃燈閃爍還有蜂鳴器連響數聲。



圖 21、密碼輸入錯誤的結果

5、三擇一開門 - RFID

使用一張有正確 UID 的磁卡到感應區感應，正確的話電磁鎖不吸磁可以開門，且螢幕顯示 Welcome、LED 綠燈亮還有蜂響。

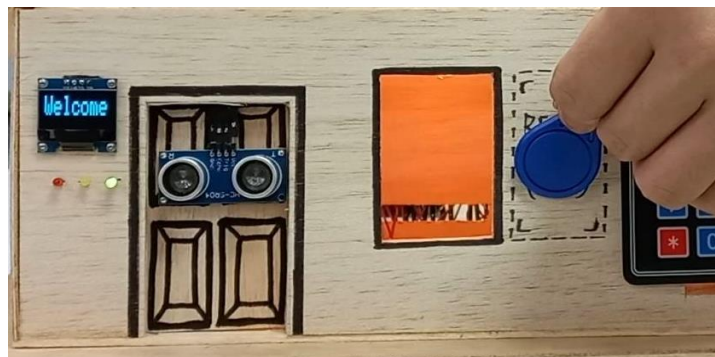


圖 22、具有正確 UID 的磁卡感應結果

若是具有錯誤 UID 的磁卡，電磁鎖依舊吸磁且螢幕顯 RFID Wrong、LED 黃燈閃爍還有蜂鳴器連響數聲。



圖 23、具有錯誤 UID 的磁卡感應結果

6、三擇一開門 - 手機 APP

使用我們設計的 APP，按下 APP 當中的 ON 後，電磁鎖不吸磁可以開門，且螢幕顯示 Welcome、LED 綠燈亮還有蜂鳴器響。



圖 24、使用 APP 當中的 ON 來開門



圖 25、按下 APP 的 ON 後的結果

(三)、鏡頭部分

- 1、等待鏡頭連接上 WiFi，當閃光燈連續閃爍就代表已成功連上。

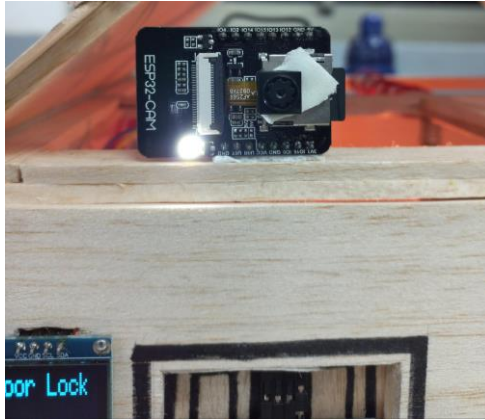


圖 26、當閃光燈連續閃爍就代表已成功連上 WiFi

- 2、連上 WiFi 後，超音波開始偵測 20 公分內是否有人，有的話就會拍照並上傳到使用者的 LINE 若沒有則超音波維持偵測狀態。



圖 30、超音波控制鏡頭拍照



圖 31、拍照後上傳至使用者的 LINE 裡

3、連上 WiFi 後，打開我們的 APP 並點擊裡面的監控，等待跳出一個新視窗即可查看即時的影像。



圖 32、點擊 APP 裡的監控



圖 27、跳出新視窗後即可查看即時影像

柒、問題與討論

一、電力

考慮到現實若是發生停電問題，我們設計了具有備用電源的電路圖，並選擇利用汽車 12V 的電瓶當備用電源再經過 12V/5V 的變壓器提供給我們的主晶片，如此一來停電時就能有電源使用了，解決方案及使用設備如下。

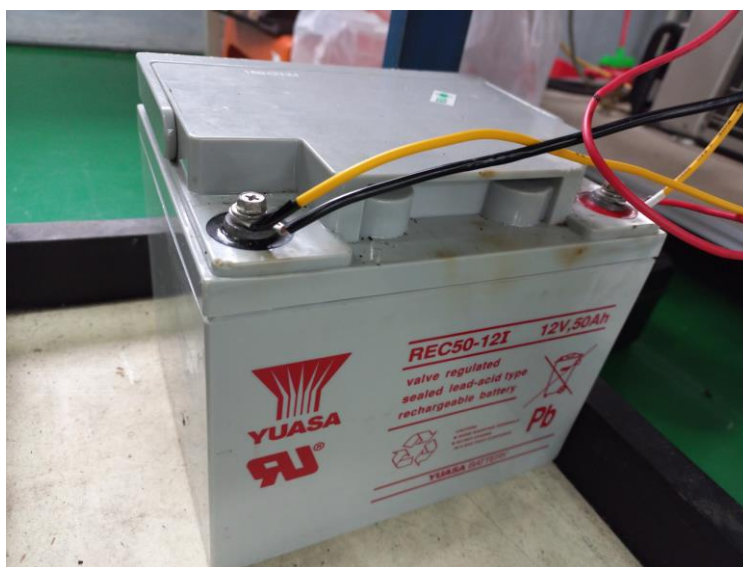


圖 28、汽車 12V 電瓶



圖 29、12V/5V 變壓器

如圖 30，平時正常供電的時候因為 Vcc 有電所以繼電器的 no 與 c 點導通 nc 與 c 點不通，平時的汽車電瓶是充電的狀態；當主電源停電的時候 Vcc 沒有電所以繼電器換成 nc 與 c 點導通 no 與 c 點不通，所以變成了汽車電瓶作為電源提供給我們的電子鎖與鏡頭，如此一來就算遇到停電的問題，電子鎖與鏡頭依然能正常使用。

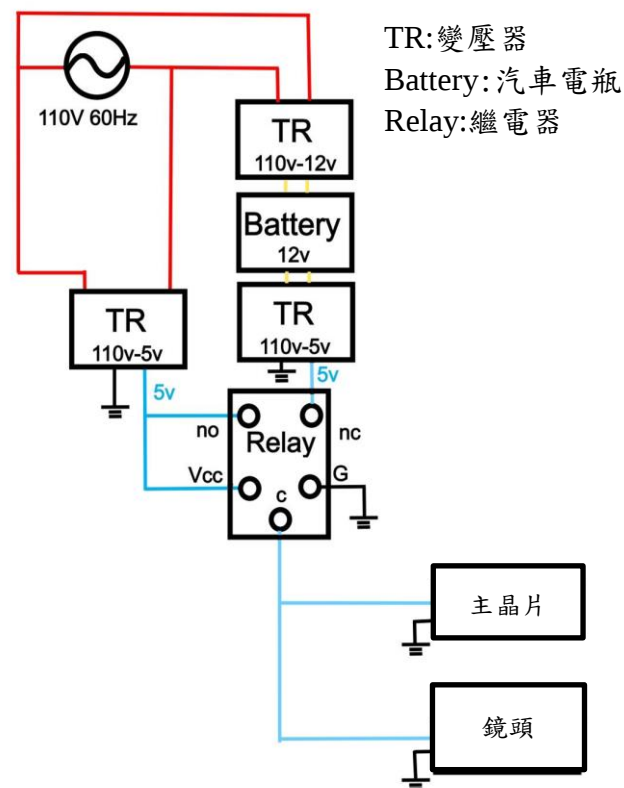


圖 30、應變停電時所設計的電路圖

二、ESP32 晶片問題

我們原本使用的晶片是 ESP-12K，但在我們後來要追加額外的零件時，發現了不明原因而導致它不能支援我們的追加零件，因此我們晶片換成了 nodemcu-32s。

三、薄膜鍵盤接腳過多

因為板子的接腳數量有限所以我們另外找了 PCF8574(參照圖 6) 來做轉接頭。

四、拍照傳 LINE 會有時差

原因出在於我們所用的 esp32cam 性能較差，還有使用的超音波過於靈敏，當鏡頭拍照存入緩存要上傳至 LINE 時，上傳的會是上一次所存的照片而不是當下所拍的照片，我們嘗試了許多程式的改寫，但依然不能解決。

五、自動門

原本我們有打算用伺服馬達來控制門，但是後來發現運用在模型上很困難、門的軌跡不順暢而且自動門的開關並不人性化，這些原因使我們捨棄了自動門的設計。

捌、結論

經過這一次的專題研究，我們成功做出有別於傳統的門鎖，有著密碼鎖、RFID 還有物聯網來控制。傳統鑰匙鎖只有唯一一種開鎖方式就是鑰匙開鎖；科技鎖則是利用密碼、RFID 還有物聯網控制門鎖，比起傳統鑰匙科技鎖的「鑰匙」更加的全面且方便，透過這個科技鎖不僅在生活上更便利，再加上拍照傳 LINE 的功能更是增加了使用者的安全性。

玖、未來展望

近年來物聯網被大量的運用在生活周遭，但它的發展尚未完全，一直以來安全性都是物聯網被嚴重質疑的一點，而且一旦網路出現狀況就會導致設備不能用的情形發生，所以我們期待在未來物聯網的功能可以在門鎖的使用上更普及、更完全。

拾、參考資料及其他

- [1] nodemcu-32s
https://www.waveshare.com/media/catalog/product/cache/1/image/800x800/9df78eab33525d08d6e5fb8d27136e95/n/o/nodemcu-32s-1_3.jpg
- [2] SSD1306 OLED 0.96 吋
<https://sc04.alicdn.com/kf/H89aca35e4d7f4189a23010d80d403e5fc.jpg>
- [3] 4x3 薄膜鍵盤
<https://www.nyerekatech.com/wp-content/uploads/2020/03/14-3-Membrane-Switch-Matrix-Keypad.jpg>
- [4] RFID-RC522
https://img.ruten.com.tw/s2/b/90/86/21931331400838_176.jpg
- [5] PCF8574
https://www.taiwaniot.com.tw/wp-content/uploads/2017/01/71jDLEIwcdL_SL1500.jpg
- [6] 有源蜂鳴器
<https://ae01.alicdn.com/kf/HTB1U3LnSFXXXXc5XpXXq6xXFXXXX/Free-shipping-100pcs-LOT-Active-Buzzer-Alarm-5v-Buzzer-5v-Sounder-speaker-Buzzer-Want-good-quality.jpg>
- [7] esp32cam
<https://www.cnx-software.com/wp-content/uploads/2018/09/ESP32-CAM.jpg>
- [8] 繼電器模組
<https://blog.jmaker.com.tw/content/images/2020/04/relay-6.png>
- [9] HC-SR04
https://www.taiwansensor.com.tw/wp-content/uploads/2018/03/NSE-1043-1_4-%E8%A4%87%E8%A3%BD-600x600.jpg
- [10] 飛機木片
https://uploads.52go.com.tw/10/10776207/15267918325_default.jpg
- [11] MQTT 教學（一）：認識 MQTT - 超圖解系列圖書
- [12] IoT 物聯網應用：使用 ESP32 開發版與 Arduino C 程式語言(作者：尤濬哲/出版社：台科大)
- [13] 傑森創工-教學網 <https://blog.jmaker.com.tw/>

附件一

群 科			專題組 創意組	參賽人數	4
作品名稱			智慧紀錄電子鎖		
年	月	日	進度	紀錄	工作分配
110	10	16	1.購買材料 2.確認材料功能 3.esp32探索	除了繼電器、電源變壓器外材料都買齊，且經過esp32的控制功能都正常，也有進行初步程式設計。	程式相關 - 言禎 資料蒐集、材料相關 - 柏羽 討論、發想和採買 - 其蒙 少杰 言禎 柏羽
110	11	2	1.物聯網和app程式完成 2.最後電路的討論	經過不斷的嘗試和努力下，終於把物聯網和app的arduino程式完成，也一起討論了最後的電路。	程式相關 - 言禎 柏羽 資料蒐集、材料相關 - 柏羽 討論 - 其蒙 少杰 言禎 柏羽
110	11	9	1.arduino拍照程式+傳line功能 2.準備電路的設備 3.模型構思	esp32cam的arduino程式比想像中困難，可能要花一些時間摸索和測試。電路我們準備了汽車電瓶，還有汽車上變壓的插座，也開始模型構思。	程式相關 - 言禎 柏羽 電路準備 - 其蒙 少杰 討論 - 其蒙 少杰 言禎 柏羽
110	11	16	1.esp32cam照片傳line的程式 2.製作模型所需的木板	我們找了老師協助後照片傳line的功能依然不能作用，但感覺快可以了需要更多的時間，另外用來做模型的木板已經有了。	程式相關 - 言禎 柏羽 木材提供 - 少杰 討論 - 其蒙 少杰 言禎 柏羽
110	11	23	1.arduino程式竣工 2.手機app即時監控功能	經歷重重難關但依然披荊斬棘完成了arduino的所有程式，另外app開始做即時監控的功能。	程式相關 - 言禎 柏羽 討論 - 其蒙 少杰 言禎 柏羽
110	11	30	1.構思模型製作和電路佈線 2.模型製作 3.汽車電瓶短接意外	這週大家都忙於考試所以僅有討論跟做模型。因為汽車電瓶沒放置好的緣故，而導致連接於其上的電線短路，所幸線路內有保險絲才沒有任何人受傷，電瓶也沒事功能正常，僅有電線燒毀。	討論 - 其蒙 少杰 言禎 柏羽 模型 - 其蒙 少杰 言禎 柏羽
110	12	7	1.添加伺服馬達、OLED螢幕、蜂鳴器和有LED燈*3 2.模型重新構想與製作	這週我們大刀闊斧添加了新的設備來增加專題精緻度。原本所做的模型不符合預期所以我們買了飛機木片，打算重新做另一個較好的成品。	討論 - 其蒙 少杰 言禎 柏羽 採買 - 柏羽
110	12	14	1.寫上週添加設備的程式arduino 2.程式無法正常運行 3.主晶片esp12k換成node32s	這週在寫程式的時候發現了伺服馬達不能控制，所以換了同系列的新主晶片，還遇到了不明原因的bug導致程式無法正常運行。	程式相關 - 言禎 柏羽 討論 - 其蒙 少杰 言禎 柏羽
110	12	21	1.程式無法正常運行已解決 2.模型2.0已開始製作	上週遇到的程式問題發現是原本寫好的rfid程式所導致的，我嘗試了不同的寫法就可以正常運作了。模型2.0也已開始製作中。	程式相關 - 言禎 柏羽 模型 - 其蒙 少杰 言禎 柏羽 討論 - 其蒙 少杰 言禎 柏羽
110	12	28	1.製作PPT和海報 2.拍攝影片	模型電路都已經整合再一起並且功能都正常可以運行，現在則著重在書面的資料、PPT和海報，還有拍攝實際展示的影片。	PPT - 其蒙 少杰 言禎 柏羽 影片 - 其蒙 少杰 言禎 柏羽 海報 - 其蒙 少杰 言禎 柏羽
111	1	4	1.成果發表	今天印證了我們所投入的時間、心力和金錢都沒有白費，我們在競賽當中得到第四名我已經心滿意足，從無到有，我在做專題所學到的、所領受的都遠超乎我所求所想，令我深刻的莫過於「我們」每個人都願意付出、願意合作的心思意念。	PPT - 其蒙 少杰 言禎 柏羽 報告 - 其蒙 少杰 言禎 柏羽

```

/*
 * Created by ArduinoGetStarted.com
 *
 * This example code is in the public domain
 *
 * Tutorial page: https://arduinogetstarted.com/tutorials/arduino-rfid-nfc-relay
 */

#include <SPI.h>
#include <MFRC522.h>
#include <Arduino.h>
#include <Wire.h>
#include <Keypad_I2C.h>
#include <Keypad.h>
#include <WiFi.h>;
#include <PubSubClient.h>
#define RELAY_PIN 13 // the Arduino pin connects to relay
#include <Adafruit_SSD1306.h>
#define I2CADDR 0x20

WiFiClient client;

int buttonState = 0;

PubSubClient mqtt_client(client);
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels
char *reference;

#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
TwoWire *jwire = &Wire;

char* ssid = "Q=CV";
char* ssidpassword = "34262672";

MFRC522 mfrc522(5, 15);

String tagUID = "4A A1 F9 80"; // String to store UID of tag.
String tagUID2 = "15 0C 5A 28";

#define Password_Length 5

const int ROWS = 4;
const int COLS = 3;

char hexaKeys[ROWS][COLS] = {

```

```

    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'};

byte rowPins[ROWS] = {0, 1, 2, 3};
byte colPins[COLS] = {4, 5, 6};

char Data[Password_Length];
char Master[Password_Length] = "1234";
byte data_count = 0;
char customKey;

Keypad_I2C kpd(makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS,I2CADDR,PCF8574, jwire );

void setup() {
  Serial.begin(115200);
  pinMode(12, INPUT_PULLUP);
  SPI.begin(); // init SPI bus
  mfrc522.PCD_Init();
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN,HIGH);
  pinMode(4, OUTPUT);// 蜂鳴器
  pinMode(32,OUTPUT);//錯誤黃燈
  jwire->begin( );
  kpd.begin( );
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
  {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;)
      ; // Don't proceed, loop forever
  }
  display.display();
  delay(500); // Pause for 2 seconds

  // Clear the display and show it.
  display.clearDisplay();
  display.display();
  //測試 RFID 設備是否正常
  byte v = mfrc522.PCD_ReadRegister(mfrc522.VersionReg);
  if((v == 0x00) || (v == 0xFF)){
    reference = " Reader Error ";
    showText1();
    digitalWrite(32,HIGH);
    digitalWrite(4,HIGH);
    delay(200);
    digitalWrite(32,LOW);
    digitalWrite(4,LOW);
  }
}

```

```

    delay(200);
    digitalWrite(32,HIGH);
    digitalWrite(4,HIGH);
    delay(200);
    digitalWrite(32,LOW);
    digitalWrite(4,LOW);
    delay(200);
    digitalWrite(32,HIGH);
    digitalWrite(4,HIGH);
    delay(200);
    digitalWrite(32,LOW);
    digitalWrite(4,LOW);
}else{
    reference = " Reader Ready ";
    showText1();
    delay(1500);
    display.clearDisplay();
}

//mqtt
Serial.begin(115200);
Serial.print("連線中");
Serial.println(ssid);
WiFi.begin(ssid,ssidpassword);
while(WiFi.status() != WL_CONNECTED) {
    delay(200);
    Serial.print(".");
    digitalWrite(32,HIGH);
    display.clearDisplay();
    display.setCursor(0, 0);
    display.setTextSize(2); // Draw 3X-scale text
    display.print("Connectingto WIFI... ");
    display.display();
}

display.clearDisplay();
display.setCursor(0, 0);
display.setTextSize(2); // Draw 3X-scale text
display.print(" Success ");
display.display();
delay(500);
display.clearDisplay();
showText2();

Serial.println(WiFi.localIP());
digitalWrite(32,LOW);
mqtt_client.setServer("broker.emqx.io",1883);
while (!mqtt_client.connect("")) {
    Serial.print(".");
    delay(200);
}
Serial.println("Connected !");

```

```

mqtt_client.subscribe("esp32/door");
mqtt_client.setCallback(callback);
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN,HIGH);

}

void loop() {

  customKey = kpd.getKey();
  buttonState = digitalRead(12); //讀取按鍵的狀態

  if(buttonState == LOW){
    display.clearDisplay();
    display.setCursor(0, 0);
    display.setTextSize(3);
    display.print("Welcome");//如果按鍵按了
    display.display();
    Serial.println("Open");

    digitalWrite(RELAY_PIN, LOW);
    digitalWrite(4,HIGH);
    delay(3000);
    digitalWrite(RELAY_PIN, HIGH);
    digitalWrite(4,LOW);
    display.clearDisplay();
    showText2();
  }
  if (customKey)
  {
    digitalWrite(4,HIGH);
    Data[data_count] = customKey;
    display.setCursor(data_count * 8, 20);
    display.print(" * ");
    display.display();
    data_count++;
    delay(100);
    digitalWrite(4,LOW);
  }

  if (data_count == Password_Length - 1)
  {
    display.clearDisplay();
    display.setCursor(0, 0);

    if (!strcmp(Data, Master))
    {
      display.setTextSize(3); // Draw 3X-scale text
      display.print("Welcome");
      display.display();
      Serial.println("Open");
    }
  }
}

```

```

    digitalWrite(RELAY_PIN, LOW);
    digitalWrite(4,HIGH);
    delay(3000);
    digitalWrite(RELAY_PIN, HIGH);
    digitalWrite(4,LOW);
    display.clearDisplay();
    showText2();

}
else
{
    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.print(" Number Wrong ");
    display.display();

    digitalWrite(32,HIGH);
    digitalWrite(4,HIGH);
    delay(500);
    digitalWrite(32,LOW);
    digitalWrite(4,LOW);
    delay(500);
    digitalWrite(32,HIGH);
    digitalWrite(4,HIGH);
    delay(500);
    digitalWrite(32,LOW);
    digitalWrite(4,LOW);

    display.clearDisplay();
    showText2();
}
clearData();
}

// Look for new cards
if ( mfrc522.PICC_IsNewCardPresent() ) {

// Select one of the cards
if ( mfrc522.PICC_ReadCardSerial() ) {

//Reading from the card
String tag = "";
for (byte i = 0; i < mfrc522.uid.size; i++)
{
tag.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
tag.concat(String(mfrc522.uid.uidByte[i], HEX));
}
tag.toUpperCase();

//Checking the card
if (tag.substring(1) == tagUID or tag.substring(1) == tagUID2) //change here the
UID of the card/cards that you want to give access

```

```

{
    display.clearDisplay();
    display.setCursor(0, 0);
    Serial.println("Authorized Tag");
    display.setTextSize(3); // Draw 3X-scale text
    display.print("Welcome");
    display.display();

    digitalWrite(RELAY_PIN, LOW);
    digitalWrite(4,HIGH);
    delay(3000);
    digitalWrite(RELAY_PIN, HIGH);
    digitalWrite(4,LOW);
    display.clearDisplay();
    showText2();
}

else
{
    Serial.print("Unauthorized Tag with UID:");

    for (int i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
    }
    Serial.println();
    display.clearDisplay();
    display.setCursor(0, 0);
    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.print(" RFID Wrong ");
    display.display();

    digitalWrite(32,HIGH);
    digitalWrite(4,HIGH);
    delay(500);
    digitalWrite(32,LOW);
    digitalWrite(4,LOW);
    delay(500);
    digitalWrite(32,HIGH);
    digitalWrite(4,HIGH);
    delay(500);
    digitalWrite(32,LOW);
    digitalWrite(4,LOW);

    display.clearDisplay();
    showText2();
}

mfrc522.PICC_HaltA(); // halt PICC
mfrc522.PCD_StopCrypto1(); // stop encryption on PCD
}
}

mqtt_client.loop();

```

```

}

//mqtt
String getTopicData(String myTopic, String inTopic, byte* payload ,unsigned int
length){
    String TopicData;
    if(inTopic == myTopic){
        for (int i = 0; i < length; i++) {
            TopicData += (char)payload[i];}
        return TopicData;
    }else{
        return "";
    }
}

void callback(char* topic, byte* payload, unsigned int length){
    Serial.println((getTopicData("esp32/door", topic, payload,length)));
    if (getTopicData("esp32/door", topic, payload,length) == "ON") {

        display.clearDisplay();
        display.setCursor(0, 0);
        display.setTextSize(3); // Draw 3X-scale text
        display.print("Welcome");
        display.display();
        Serial.println("Open");
        digitalWrite(RELAY_PIN, LOW);
        digitalWrite(4,HIGH);
        delay(3000);
        digitalWrite(RELAY_PIN, HIGH);
        digitalWrite(4,LOW);
        display.clearDisplay();
        showText2();

    }
}

void showText1(){
    display.setCursor(0, 0);
    display.setTextSize(2); // Draw normal-scale text
    display.setTextColor(SSD1306_WHITE);
    display.print(" Reader  Testing ");
    display.display();
    delay(500);
    display.clearDisplay();
    display.setCursor(0, 0);
    display.setTextSize(2);
    display.print(reference);
    display.display();
}

```

```
void showText2(){
    char *reference;
    display.setCursor(0, 0);
    display.setTextSize(2); // Draw normal-scale text
    display.setTextColor(SSD1306_WHITE);
    display.print(" Door Lock ");
    display.display();
}

void clearData()
{
    while (data_count != 0)
    {
        Data[data_count--] = 0;
    }
    return;
}
```

```

#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include "esp_http_server.h"

#define PART_BOUNDARY "1234567890000000000000987654321"

//ESP32-CAM 安信可模組腳位設定
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

//輸入 Wi-Fi 帳密
char* ssid = "Q=CV"; //Wi-Fi 帳號
char* password = "34262672"; //Wi-Fi 密碼
String Linetoken =
"pTZus0CfywBKKVcQA3VfkKfAQL8Mbd2UeRuMc7qORpZ"; //Line Notify 權杖
int trigPin = 13; //請將 Trig 接 GPIO15
int echoPin = 15;

void setup()
{
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //關閉電源不穩就重開
    機的設定

    Serial.begin(115200);
    Serial.setDebugOutput(true); //開啟診斷輸出

```

```

Serial.println();

camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//      Ensure ESP32 Wrover Module or other board with PSRAM is selected
//      Partial images will be transmitted if image exceeds buffer size
//
// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
//      for larger pre-allocated frame buffer.
if(psramFound()){ //是否有 PSRAM(Pseudo SRAM)記憶體 IC
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

//視訊初始化
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    ESP.restart();
}

//可自訂視訊框架預設大小(解析度大小)
sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {

```

```

s->set_vflip(s, 1); // flip it back
s->set_brightness(s, 1); // up the brightness just a bit
s->set_saturation(s, -2); // lower the saturation
s->set_vflip(s, 1); //垂直翻轉
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_SVGA); //解析度 SVGA(800x600),
VGA(640x480), CIF(400x296), QVGA(320x240), HQVGA(240x176),
QQVGA(160x120), QXGA(2048x1564 for OV3660)

//s->set_hmirror(s, 1); //水平鏡像

//閃光燈(GPIO4)
ledcAttachPin(4, 4);
ledcSetup(4, 5000, 8);

WiFi.mode(WIFI_STA);

for (int i=0;i<2;i++) {
  WiFi.begin(ssid, password); //執行網路連線

  delay(1000);
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(ssid);

  long int StartTime=millis();
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    if ((StartTime+5000) < millis()) break; //等待 10 秒連線
  }

  if (WiFi.status() == WL_CONNECTED) { //若連線成功
    Serial.println("");
    Serial.print("Camera Stream Ready! Go to: http://");
    Serial.println(WiFi.localIP());
    Serial.println(WiFi.localIP());
    Serial.println("");

    for (int i=0;i<5;i++) { //若連上 WIFI 設定閃光燈快速閃爍
      ledcWrite(4,10);
      delay(200);
      ledcWrite(4,0);
      delay(200);
    }

    break;
  }
}
}

```

```

if (WiFi.status() != WL_CONNECTED) { //若連線失敗
  ESP.restart(); //重啟電源
}

//閃光燈(GPIO4)
ledcAttachPin(4, 4);
ledcSetup(4, 5000, 8);

//測試傳送影像至 Line Notify，一小時最多上傳 50 張照片。
sendCapturedImage2LineNotify(Linetoken);
Serial.println();
startCameraServer();
pinMode(trigPin, OUTPUT);    //Define inputs and outputs
pinMode(echoPin, INPUT);
}

void loop()
{ digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);  // 給 Trig 高電位，持續 10 微秒
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long cm = pulseIn(echoPin, HIGH); // 收到高電位時的時間
  cm = (cm / 2) / 29.1;    // 將時間換算成距離 cm 或 inch
  Serial.println(cm);
  if (cm <= 30) {
    Serial.println("starting to Line");
    sendCapturedImage2LineNotify(Linetoken);
    delay(1000);
  }
  delay(5000); //You could only send up to 50 images to Line Notify in one hour.
}

String sendCapturedImage2LineNotify(String Token) {
  const char* myDomain = "notify-api.line.me";
  String getAll="", getBody = "";

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
  }

  Serial.println("Connect to " + String(myDomain));
  WiFiClientSecure client_tcp;
  client_tcp.setInsecure(); //run version 1.0.5 or above

  if (client_tcp.connect(myDomain, 443)) {

```

```

Serial.println("Connection successful");

String message = "有人進出門口";
String head = "--Taiwan\r\nContent-Disposition: form-data; name=\"message\";\r\n\r\n" + message + "\r\n--Taiwan\r\nContent-Disposition: form-data; name=\"imageFile\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
String tail = "\r\n--Taiwan--\r\n";

uint16_t imageLen = fb->len;
uint16_t extraLen = head.length() + tail.length();
uint16_t totalLen = imageLen + extraLen;

client_tcp.println("POST /api/notify HTTP/1.1");
client_tcp.println("Connection: close");
client_tcp.println("Host: notify-api.line.me");
client_tcp.println("Authorization: Bearer " + Token);
client_tcp.println("Content-Length: " + String(totalLen));
client_tcp.println("Content-Type: multipart/form-data; boundary=Taiwan");
client_tcp.println();
client_tcp.print(head);

uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
for (size_t n=0;n<fbLen;n=n+2048) {
    if (n+2048<fbLen) {
        client_tcp.write(fbBuf, 2048);
        fbBuf += 2048;
    }
    else if (fbLen%2048>0) {
        size_t remainder = fbLen%2048;
        client_tcp.write(fbBuf, remainder);
    }
}

client_tcp.print(tail);
esp_camera_fb_return(fb);

int waitTime = 10000; // timeout 10 seconds
long startTime = millis();
boolean state = false;

while ((startTime + waitTime) > millis())
{
    Serial.print(".");
    delay(100);
    while (client_tcp.available()) {
        char c = client_tcp.read();
        if (state==true) getBody += String(c);
        if (c == '\n') {
            if (getAll.length()==0) state=true;
            getAll = "";
        } else if (c != '\r')

```

```

        getAll += String(c);
        startTime = millis();
    }
    if (getBody.length() > 0) break;
}
client_tcp.stop();
Serial.println(getBody);
}
else {
    getBody = "Connected to " + String(myDomain) + " failed.";
    Serial.println("Connected to " + String(myDomain) + " failed.");
}
return getBody;
}

static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-
replace;boundary=" PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-
Length: %u\r\n\r\n";

httpd_handle_t camera_httpd = NULL;
httpd_handle_t stream_httpd = NULL;

static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<html>
<head>
  <title>ESP32-CAM Video Stream</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body { font-family: Arial; text-align: center; margin: 0px auto; padding-top: 30px; }
    table { margin-left: auto; margin-right: auto; }
    td { padding: 8 px; }
    .button {
      background-color: #2f4468;
      border: none;
      color: white;
      padding: 20px 40px;
      text-align: center;
      text-decoration: none;
      display: inline-block;
      font-size: 20px;
      margin: 6px 3px;
      cursor: pointer;
      -webkit-touch-callout: none;
      -webkit-user-select: none;
      -khtml-user-select: none;
      -moz-user-select: none;
      -ms-user-select: none;
      user-select: none;
      -webkit-tap-highlight-color: rgba(0,0,0,0);

```

```

    }
    img { width: auto ;
        max-width: 100% ;
        height: auto ;
    }
</style>
</head>
<body>
<h1>ESP32-CAM Video Stream</h1>
<img src="" id="photo" >
<div align="center">
    <button class="button" onclick="startStream('start');">START</button>
    <button class="button" onclick="startStream('stop');">STOP</button>
</div>
<script>
function startStream(x) {
    if(x=='start'){
        document.getElementById("photo").src = window.location.href.slice(0, -1) +
":81/stream";
    }
    else if(x=='stop'){
        document.getElementById("photo").src = "";
    }
}
window.onload = document.getElementById("photo").src =
window.location.href.slice(0, -1) + ":81/stream";
</script>
</body>
</html>
)rawliteral";

static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }
    while(true){
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println("Camera capture failed");
            res = ESP_FAIL;

```

```

    } else {
        if(fb->width > 400){
            if(fb->format != PIXFORMAT_JPEG){
                bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
                esp_camera_fb_return(fb);
                fb = NULL;
                if(!jpeg_converted){
                    Serial.println("JPEG compression failed");
                    res = ESP_FAIL;
                }
            } else {
                _jpg_buf_len = fb->len;
                _jpg_buf = fb->buf;
            }
        }
    }
    if(res == ESP_OK){
        size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
        res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
    }
    if(res == ESP_OK){
        res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
    }
    if(res == ESP_OK){
        res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
        strlen(_STREAM_BOUNDARY));
    }
    if(fb){
        esp_camera_fb_return(fb);
        fb = NULL;
        _jpg_buf = NULL;
    } else if(_jpg_buf){
        free(_jpg_buf);
        _jpg_buf = NULL;
    }
    if(res != ESP_OK){
        break;
    }
    //Serial.printf("MJPG: %uB\n", (uint32_t)(_jpg_buf_len));
}
return res;
}

void startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    config.server_port = 80;
    httpd_uri_t index_uri = {
        .uri      = "/",
        .method    = HTTP_GET,
        .handler   = index_handler,
        .user_ctx  = NULL
    };
    httpd_uri_t stream_uri = {

```

```
.uri    = "/stream",
.method = HTTP_GET,
.handler = stream_handler,
.user_ctx = NULL
};
if (httpd_start(&camera_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(camera_httpd, &index_uri);
}
config.server_port += 1;
config.ctrl_port += 1;
if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &stream_uri);
}
}
```