

Arduino 程式設計專題製作報告

題目：自動鎖

作者：臺北市私立復興實驗高級中學

十一年信班 14 趙貞洵 16 劉蕃熙

指導老師：康宗良

一、專題簡介：

自動鎖，指的是當門被關起時會自動鎖上，當輸入密碼正確時，則會將門打開。以此為基礎，為了減少使用的按鈕與材料量，我們決定簡化一般輸入 4 碼 0~9 的數字密碼，改成 0 和 1 兩個數字，但為了兼顧安全與方便，密碼增為 8 位，這樣就可以產生 256 種可能的密碼。

這次我們利用在生活科技課學習到的電子元件與寫法，以及資訊課學到的 c++ 語法，使用 Arduino WeMos D1 R1 開發板，製作出簡易自動鎖。此次專題也是我們首次結合軟硬體，自行發想、分析、實作、測試並完成製作自動鎖。對我們來說是非常特別且難得的學習經驗。

二、專題製作過程

(一)問題分析

【存取正確密碼與輸入密碼】

利用兩個大小為 8 的一維矩陣 `right_ans[8]`, `my_ans[8]`，分別紀錄正確密碼與輸入密碼。

【記錄輸入的位數】

利用一個 int 型態的變數 `idx`，一開始 `idx=0`，當有按鈕被按下時，紀錄 `my_ans[idx]` 的值，並將 `idx+1`，並於每次輸入完完整密碼後將 `idx` 歸為 0。

【控制門鎖開關】

當門要鎖著時，伺服馬達轉為 90 度鎖住門；當門要打開時，伺服馬達轉為 180 度打開門鎖。

(二)電子元件介紹

1. LOLIN (WeMos) D1 WiFi Arduino UNO 開發板(圖一)：這個開發板的微控制器是使用 ESP-8266EX，開發板共有 11 個數位 I/O 腳位和 1 個類比輸入腳位，板載 5V 1A 開關電源。與 Arduino 兼容，使用 Arduino IDE 編程。



【圖一：LOLIN (WeMos) D1 WiFi Arduino UNO 開發板】

2. 伺服馬達(圖二)：「伺服馬達」或稱「伺服舵機」，其內部是由直流馬達、減速齒輪、位置感測器、電路板組合而成。我們可以透過傳遞訊號的方式來控制伺服馬達轉到指定角度（可以旋轉 180 度），並且固定住，跟一般直流馬達不太一樣。這項特色能滿足很多動作需求，可以用在機器人控制、遙控車方向控制、飛機模型等等。
3. 微動開關(圖三)：微動開關是最常見的 Arduino 輸入裝置，輸入的內容分為按下與放開兩種情況。微動開關有四個針腳，分為兩組兩兩各自連通，當微動開關被按下時，左右的針腳就會導通。
4. 蜂鳴器(圖四)：蜂鳴器是一種電聲元件，主要用來發出警示聲、提示生或是回饋音等等。本報告使用的電磁式蜂鳴器主要是由震盪器、電磁線圈、磁鐵、震動魔片以及外殼組成。
5. RGB Led(圖五)：RGB Led 是一個可以發出紅色(Red)、綠色(Green)、藍色(Blue)的 Led 燈，並且可以分別設定紅、綠、藍三個顏色的亮度，混合出各種不同顏色的 Led 燈光。
6. 電阻(圖六)：電阻是以用電阻材料製成的、有一定結構形式、能在電路中起限制電流通過作用的二端電子元件，可以用於控制某一部份電路的電壓和電流比例，限制流經某一段電路的電流。



【圖二：伺服馬達】



【圖三：按鈕】



【圖四：蜂鳴器】

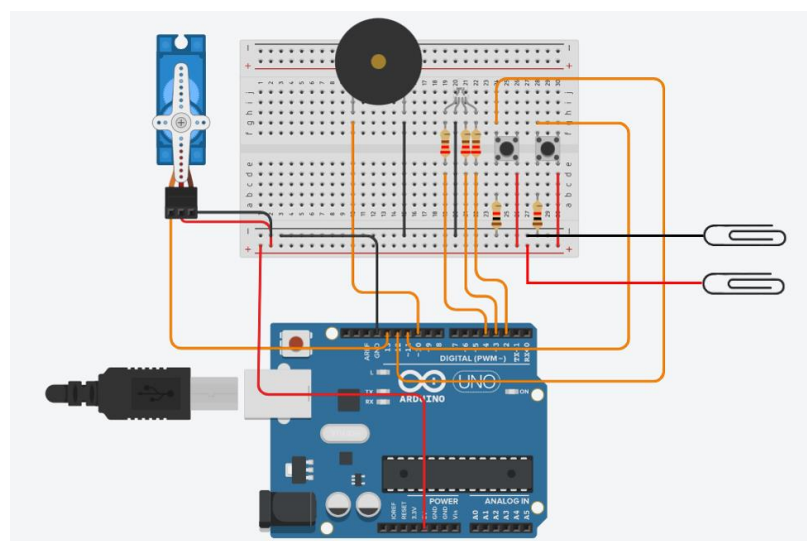


【圖五：RGB Led】



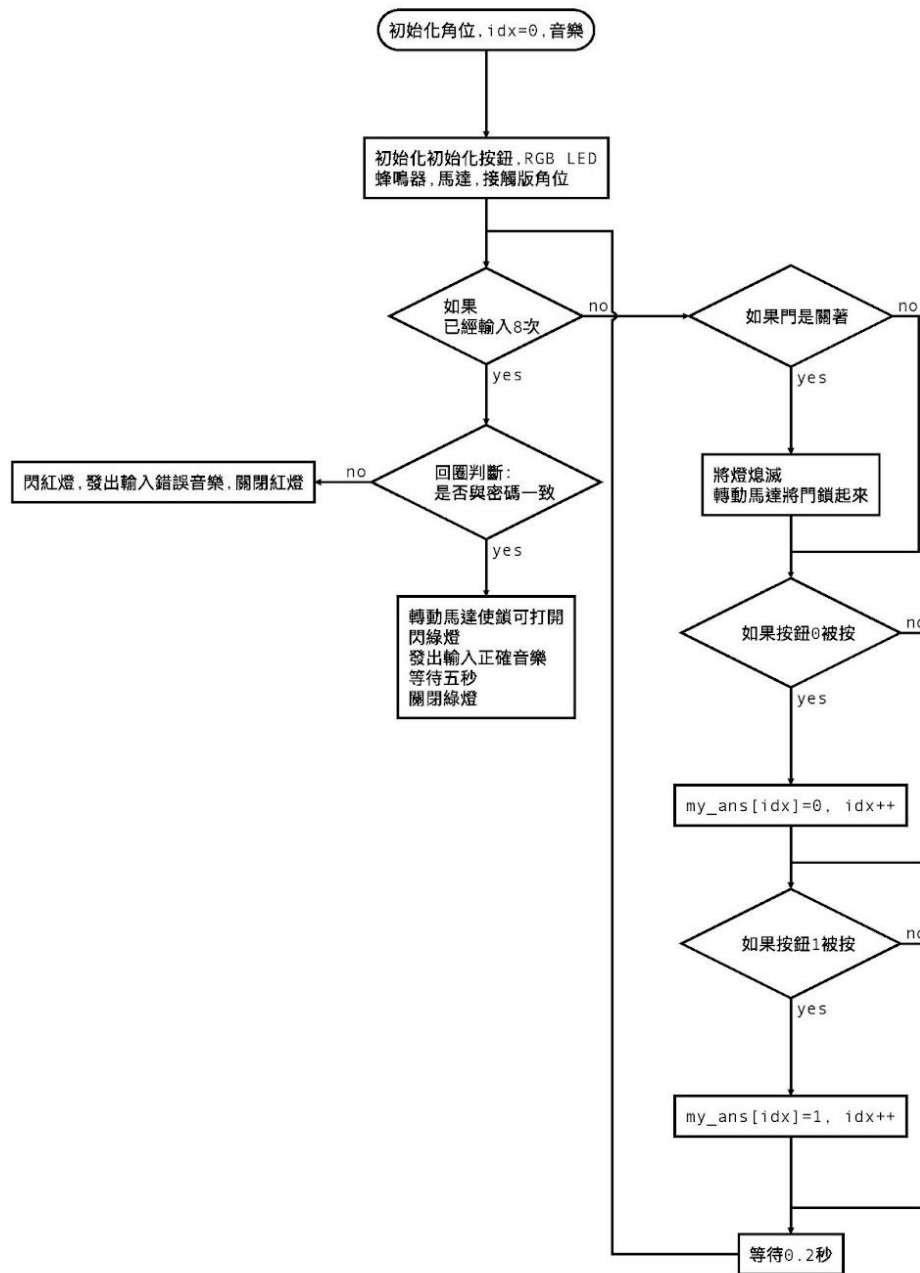
【圖六：電阻】

(三) 電路圖



【圖七】

(四)程式設計流程圖



【圖八：流程圖】

此流程圖利用 hackmd.io 中內建的 UML 語法製作

code:<https://hackmd.io/y7xmjDFgTtqo7eKmKYWqZQ?both>

(五) 程式截圖與說明

```
1 #include<Servo.h>
2
3 int but0=13, but1=12, R=4, G=5, B=16, buzzer=15, M=14, door=0;
4 int idx=0;
5 bool right_ans[8]={0,1,1,0,1,1,1,0}, my_ans[8];
6 Servo m;
7
8 void rgb(int r, int g, int b){
9     analogWrite(R,r);
10    analogWrite(G,g);
11    analogWrite(B,b);
12 }
13
14 void Yes(){
15     m.write(1510);
16     rgb(0,255,0);
17     for(int i=0; i<2; i++){
18         tone(buzzer,349,300);
19         delay(300);
20     }
21     delay(5000);
22     rgb(0,0,0);
23 }
24
25 void No(){
26     rgb(255,0,0);
27     for(int i=0; i<3; i++){
28         tone(buzzer,349,2000);
29         delay(500);
30     }
31     rgb(0,0,0);
32 }
33
34 void check(){
35     bool b=0;
36     for(int i=0; i<8; i++){
37         if(my_ans[i]!=right_ans[i]){
38             b=1;
39             break;
40         }
41     }
42     if(b==1)
43         No();
44     else
45         Yes();
46 }
```

```

47
48 void setup() {
49     pinMode(but0, INPUT);
50     pinMode(but1, INPUT);
51     pinMode(R, OUTPUT);
52     pinMode(G, OUTPUT);
53     pinMode(B, OUTPUT);
54     pinMode(buzzer, OUTPUT);
55     m.attach(M, 500, 2500);
56     pinMode(door, INPUT);
57 }
58
59 void loop() {
60     if(idx==8) {
61         check();
62         idx=0;
63     }
64     if(digitalRead(door)==0) { //鎖起來
65         rgb(0, 0, 0);
66         m.write(510);
67     }
68     if(digitalRead(but0)==1) {
69         my_ans[idx]=0;
70         idx++;
71     }
72     if(digitalRead(but1)==1) {
73         my_ans[idx]=1;
74         idx++;
75     }
76     delay(200);
77 }

```

說明：

第 1~6 行：初始化角位、輸入密碼次數初始化為 0、設置密碼

第 8~12 行：RGB Led 變色 function rgb(r, g, b) 分別代表三種顏色的亮度

第 14~23 行：當輸入密碼正確時執行：馬達轉動 90 度使門鎖打開、閃綠燈、發出輸入正確音樂，等待五秒以免使用者未及時打開、熄滅綠燈

第 25~32 行：當輸入密碼錯誤時執行：閃紅燈、發出輸入錯誤音樂、熄滅紅燈

第 34~46 行：當密碼輸入 8 次 (idx=8) 時執行：

如果有與正確密碼不符者，呼叫 No() 函式

如果全部符合呼叫 Yes() 函式

第 48~57 行：宣告與定義蜂鳴器、伺服馬達、RGB LED 燈為輸出腳位，按鈕、接觸線為輸入腳位

第 59~77 行：重複執行：

第 60~63 行：密碼輸入 8 次 (idx=8)：呼叫 check() 函式

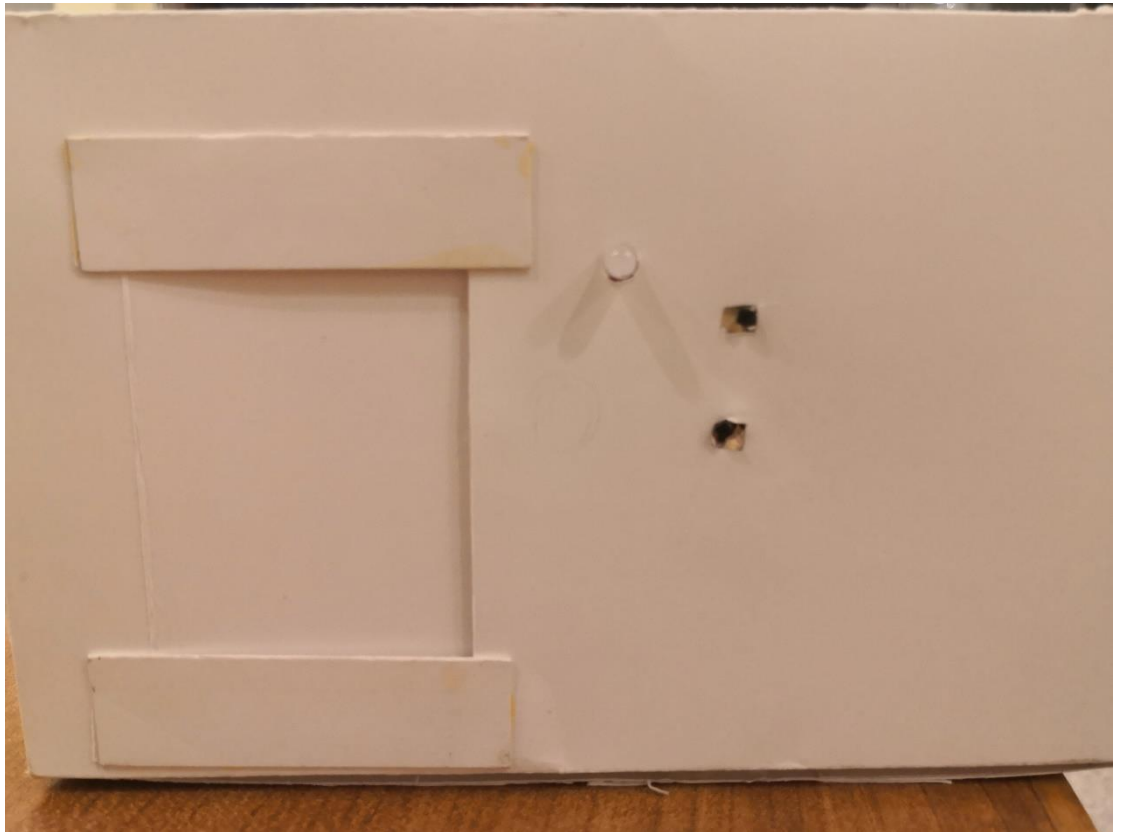
第 64~67 行：如果門是關上的：將鎖鎖起來、REG Led 熄滅

第 68~75 行：分別執行按下按鈕 0 或 1 時，紀錄於 my_ans[]
並使 idx++

第 76 行：等待 200ms 以免按鈕跳動產生的問題

四、操作說明與測試結果

(一) 成品截圖

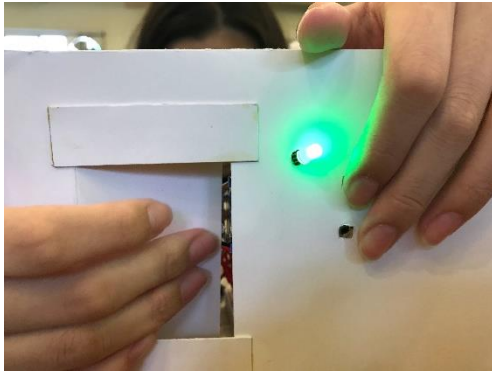


【圖九：成品外觀】

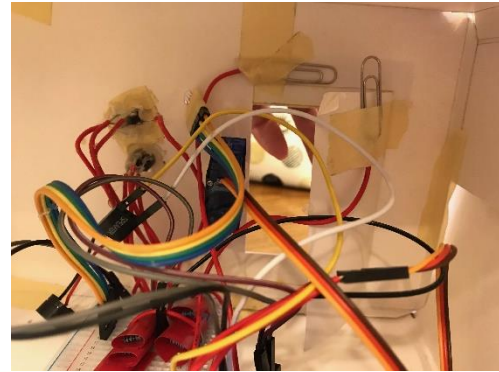
(二) 操作說明

當門被關上，兩個迴紋針會接觸並且導電，伺服馬達向右轉 90 度將門卡住，呈上鎖狀態，RGB Led 熄滅。

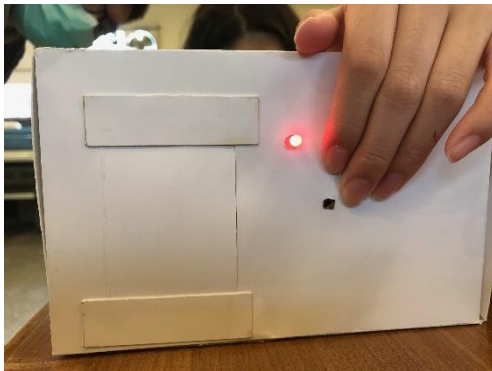
此時用按鈕輸入密碼，按上面的按鈕表示輸入 0，按下面的按鈕表示輸入 1，共需輸入 8 碼。當密碼正確，RGB Led 亮綠燈，蜂鳴器發出輸入正確的音樂，伺服馬達向左轉 90 度，呈解鎖狀態，在 5 秒內可以開啟（如圖十、圖十一），超過 5 秒如果門仍為關閉狀態將再次自動鎖上；當密碼錯誤 RGB Led 亮紅燈，蜂鳴器發出輸入錯誤的音樂，伺服馬達角度維持不動，門維持上鎖狀態（如圖十二、圖十三）。



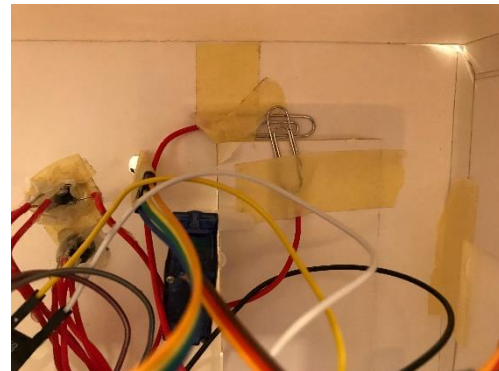
【圖十】



【圖十一】



【圖十二】



【圖十三】

五、專題結果與討論

(一) 討論與改進

1. 程式應於將成分段寫，並在每段完成時分別測試，以減少 debug 時間。分段方式可以以 function 或電子元件為單位。
2. 外觀製作須提早考慮到，以免電路接法無法直接適用於外觀，後期更改會浪費很多不必要的時間又容易出錯。
3. 未來希望可以利用時間內（如以 5 秒為單位）按的 0,1 按鈕由二進位轉成十進位，以增加安全性。目前密碼可能為 $2^8=256$ ，如改成 10 進位 4 碼可將可能增加到 10000，且無需增加額外電子元件，但可能會使使用方便性下降（使用者須按按鈕的下數增為 2 倍）希望未來改進時可以做出兼顧安全、方便、成本的電子鎖版本。

(二) 心得與反思

本次專題報告不管是在硬體方面或軟體方面，都讓我們擁有很大的突破。這是我們從第一堂課到現在第一次以自己的創意和課堂中習得的知識為根本，成功做出一個可用的小裝置。一開始我們完全想不出有什麼合適的製作內容，便參考了許多網站、影片，但內容不是過於初階、只需要用到一兩個元件和幾行程式，就是太複雜、使用的元件我們還沒有接觸過。最後，我們從擁有的元件功能思考，才想到「自動門鎖」這個主題。

有鑑於先前幾堂課接電路的經驗，我們決定先利用 Tinkercad 將電路圖畫好，不只是由於可以設定電線的顏色所以比較容易檢查、修改也較方便，更是因為直接接實體電路實在太容易出錯了。

比起接好實體再對照七彩的杜邦線、腳位畫電路圖，看著經過簡化、整理的電路圖接線真的容易許多。即便如此，我們的電路板看起來仍然有點混亂，而且測試的過程還是因為有一條線接錯，導致程式運行不成功，將程式碼檢查了好幾次後，才發現是不小心把按鈕的兩邊都接到負極了。

而在程式撰寫方面，可能是因為比起電路圖熟悉許多，因此也順利不少，但中間也發生了一些問題。最大的錯誤是我們以為第 64 行的 `digitalRead(door)==0` 時是代表門被打開，電路不聯通，但實際上是指兩邊沒有電壓差，因此是接通狀態。

另外，我們這一次是在畫完流程圖後直接開始寫程式，全部寫完才一起測試、debug，但我們發現這是不太好的方式。首先，這樣有錯誤時會無法分辨是哪裡出錯的，也會導致程式碼因為 debug 的原因變的很亂、不易讀，而產生更多錯。因此應該要寫完一個 function 或某種元件後就測試一次，這樣可以大幅度減少除錯時間。

最後測試時，我們的裝置一開始還可以正常運作，沒想到後來一直故障，我們決定測試是那些元件出了問題，於是撰寫程式個別讓每一個元件運作，沒想到全部都沒有反應。當時我們以為可能是 Arduino 板壞了，於是等到上課時向老師反應，沒想到是因為我們一開始接電路時，由於可以使用的角位不足接完所有元件，而將門上的迴紋針插到 0 號角位。這一個角位如果接到 GND，會導致 Arduino 板重啟，因此不會運行程式。

在這次的專題報告中，我們增加了對電子元件的熟悉度並清楚其程式撰寫方式和電路接法，以及分析目標並實作完成的能力，更重要的是增進我們對於軟硬體結合的統合能力，這是很難得而重要的學習。希望在未來可以改善上述問題，做出更安全、更精細的自動鎖。

我的其他學習歷程：<https://felicitytomato.github.io/MLP/>