

高雄市高英高級工商職業學校
Kao Ying Industrial Commercial Vocational High School

專題製作報告



無線密碼捲門控制器

學生姓名：黃 郁 翔

陳 展 明

黃 韋 誠

指導老師：葉 忠 賢 老師

中 華 民 國 105 年 05 月

誌 謝

感謝葉忠賢老師，再這高中的三年裡，不止教導我們在電腦上的技術，也教導了我們許許多多的做人處事道理，方向來邁進，更謝謝老師的指導，讓我們可以重新發現自己長久以來的不足與缺點，並且在學習的過程中，與老師師生之間的感情更加深厚，這樣的感受，對我們小組而言，更是珍貴如珠，在我們不會的電路及沒有見過的零件老師都加以詳細告訴我們其中的功能，例如某些零件的腳位亦或者是程式碼上面出現的錯誤老師都會耐心的告訴我們其中的錯誤道理，沒有排解出這一些困難這一個專題想必很難完成所以非常的感謝老師對我們的加以重視希望在未來的專題中還可以繼續幫我們找出困難排除困難。

黃郁翔、陳展明、黃韋誠 謹上 2016/05

無線密碼捲門控制器

摘 要

隨著科技的進步，技術不斷地創新下，從以前的手拉式鐵捲門變成了現在擁有可以遠端控制的功能外，還可以有紅外線偵測做為安全性的防護，在不斷得探討下，我們將整合電子、電機、機械等方面的知識，製作一組完整的無線遠端控制器，包含了 RS232 電路、CPU 電路等各大電路部分，配合電源、以及硬體等，因此本次的專題可以說是涉及了多方面的知識資訊領域。

關鍵詞：RS232 電路、8051 單晶片、紅外線收發 LED 等等

目 錄

誌謝	I
摘要	II
目錄	III
表目錄	IV
圖目錄	V
壹、前言	1
一、製作動機	1
二、製作目的	1
三、製作架構	2
四、製作預期成效	3
貳、理論探討	4
參、專題製作	14
一、設備及器材	14
二、製作方法與步驟	15
三、專題製作	16
肆、製作成果	24
伍、結論與建議	25
一、結論	25
二、建議	25
參考文獻	26
附錄一 無線密碼捲門控制器程式碼	34

表目錄

表 2-2-1 93C66 規格圖	5
表 2-2-2 93C66 為 11 位元指令	6
表 3-1-1 專題製作使用儀器(軟體)設備一覽表	14
表 3-2-1 專題製作計畫書	16
表 3-3-2 無線密碼捲門控制器材料表	17

圖目錄

圖 1-3-1 專題製作流程圖	2
圖 2-1-1 8051 單晶片	4
圖 2-1-2 8051 接腳圖	4
圖 2-2-1 93C66 接腳圖	5
圖 2-2-2 93C66 系統方塊圖	5
圖 2-3-1 藍芽模組	7
圖 2-3-2 藍芽接腳圖	7
圖 2-4-1 串列傳輸示意圖	8
圖 2-4-2 SCON：串列埠控制暫存器(SERIAL PORT CONTROL REGISTER).....	10
圖 2-4-3 PCON：電源控制暫存器(POWER CONTROLREGISTER)	10
圖 2-5-1 馬達驅動圖	11
圖 2-6-1 遮斷感測電路	13
圖 2-6-2 紅外線接收發射器	13
圖 3-2-1 專題製作方法流程圖	15
圖 3-2-2 8051 主電路	18
圖 3-2-3 藍芽介面	18
圖 3-2-4 蜂鳴器電路	18
圖 3-2-5 5V 電源電路.....	19
圖 3-2-6 LED 電路.....	19
圖 3-2-7 紅外線電路	19
圖 3-2-8 LCD 電路	20
圖 3-2-9 馬達電路	20
圖 3-2-10 93C66 電路圖	20
圖 4-1-1 專題製作過程(一).....	21
圖 4-1-2 專題製作過程(二).....	21
圖 4-1-3 無線密碼捲門控制器	21
圖 4-1-4 成品測試功能	21

圖 4-1-5 藍芽操控圖圖	21
圖 4-1-6 完成無線密碼捲門控制器	21
圖 4-1-7 文書討論與製作	22
圖 4-1-8 專題程式製作	22
圖 4-2-1 專題成品說明	23
圖 4-2-2 APP 操作畫面	24
圖 4-2-3 捲門上升	24
圖 4-2-4 捲門下降	24

壹、前言

一、製作動機

鐵捲門壓傷人的意外層出不窮！鐵捲門就可以區分為好幾種，像早期使用的手動鐵捲門或大門的橫式鐵門；以及社區大樓、停車場，大多遙控使用的「電動鐵捲門」，其實就怕人車太靠近發生危險，但這些意外就時常發生在我們身邊，鐵捲門的死角造成了人們的視覺偏差，因此造成了許多小動物被壓在鐵捲門底下無法移動，而造成了下半身的殘廢，為了避免再發生這些憾事，紅外線密碼捲門是個不可缺少的東西。

二、製作目的

在現今的社會中，鐵捲門也越來越多的出現，常見的都可以區分好幾種，鐵捲門壓傷的事情可說是十分常見，但安全防護做得好，不用害怕有死角，紅外線感應是為了避免壓傷人車跟動物，鐵捲門是大多數人家中都有裝的，這是為了保護使用人的安全。

在團隊中家裡有用鐵捲門的也是非常多的，正因如此，激發了我們的居家安全，搭配紅外線的偵測，以簡便的方式呈現出安全，遠離意外的傷害。

三、製作架構

(一) 專題製作流

首先我們小組集合討論出專題的方向後，大家在一起去網路上找尋關於相關的專題資料，在討論好題目後確定購買的材料，首先在麵包板上面先進行第一步的模擬測試，在測試成功後再著手進行成品的製作與研究，並寫下相關的報告跟資料，成品完成後，在進行檢查是否有出錯，途中若發現錯誤再與老師進行討論進而改變，也避免因為人為的疏失，而造成多餘的損失，我們自己多準備一套備用的材料，小心使得萬年船，若是不幸失誤，還有第二套可以進行使用，更可為自己的荷包省下一些意外的花費下，也可以不用在那兩頭跑的花時間去買材料。

(二) 製作流程圖

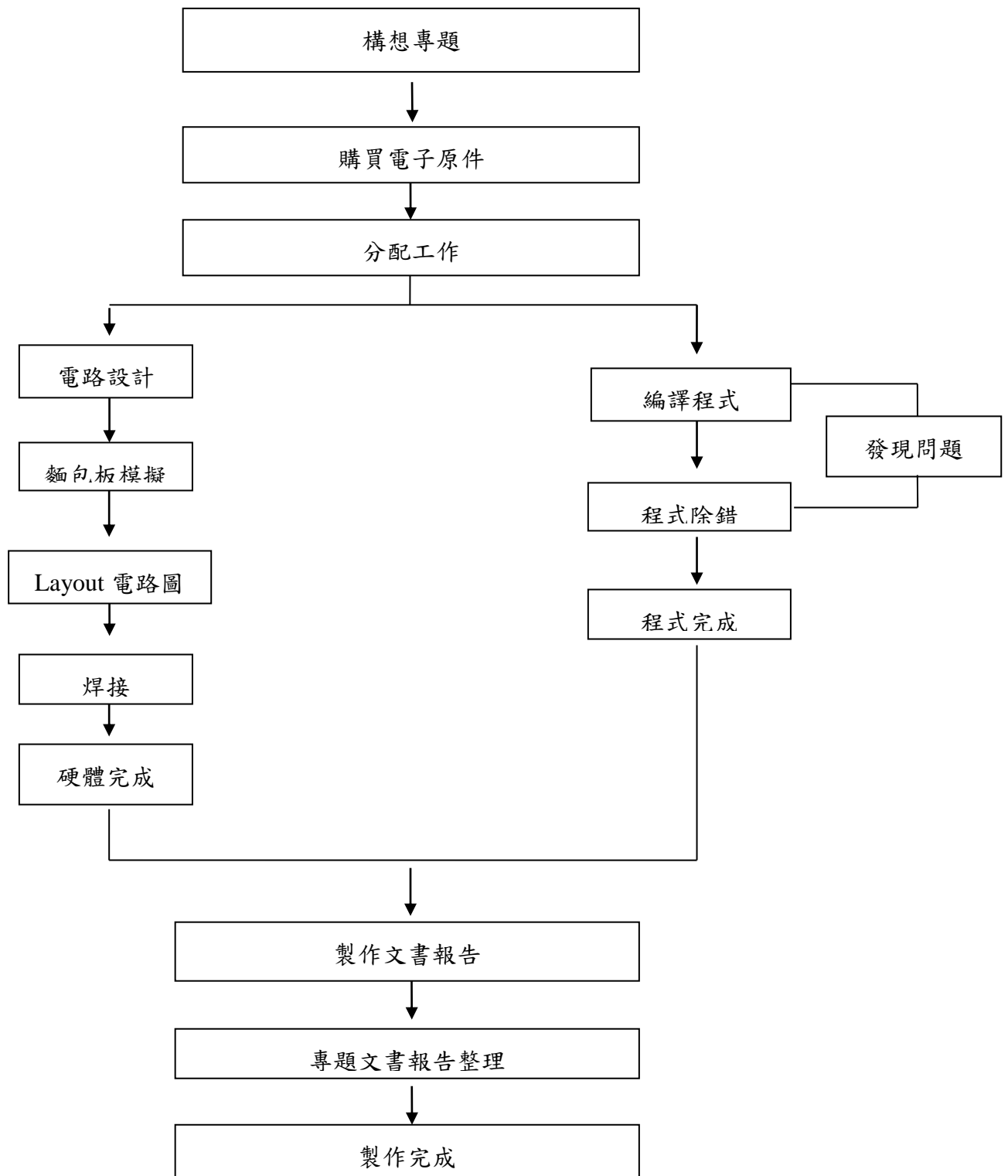


圖 1-3-1 專題製作流程圖

四、製作預期成效

小組們大家一開始都很迷失，完全都沒有概念，都很害怕會不會成功，影響到畢業，但經過同學、老師、以及網路上許許多多的資料幫助下，雖然我們這一個小組是第一次一起進行專題的製作，但在大家辛苦的付出後，我們也希望也能拿到一個漂亮的成績，所以將我們此次的專題預期成效定義如下所示：

使用者可透過 Android 手機程式介面將控制訊號透過藍芽傳輸到 8051，8051 接收到控制訊號並且作處理。

當接收到訊號時的處理方式如下：

- (1)將接收到的資料和 93C66 內儲存的密碼作比對。
- (2)辨識密碼正確時，綠色 LED 亮，錯誤時，紅色 LED 亮。
- (3)當密碼正確時，可接收手機的控制門訊號，訊號為開、停、關鐵捲門。
- (4)在藍芽控制紅外線密碼捲門下降時，若底下有異物使紅外線感測到時，將停止下降並發出警告聲響。

貳、理論探討

一、單晶片微電腦

石英振盪晶體提供 8051 工作所需時脈，系統重置是任何微處理機系統執行的第一步，使整顆控制晶片回到預先設定的硬體狀態下。8051單晶片的系統重置是由 RESET 接腳來做控制，當此接腳送入高電位超過24個振盪週期後，8051即進入晶片內部重置的狀態下，而且一直在此狀態下等待，直到RESET接腳轉為低電位後，才檢查EA接腳是高電位或低電位，若為高電位則執行晶片內部的程式碼，若為低電位便會執行外部的程式。

而8051在系統重置時，將其內部的一些重要暫存器設定為某些特定的值，對8051而言，在晶片內部的RESET接腳接有一史密特觸發電路，以及一個電阻到地端，在外部接上一個10uF的電容加上電阻，便可達成RESET的重置電路。如果要使程式碼重新執行時，可以按下外接的按鈕開關，將RESET接腳接至+5V電源，使系統重置而重新執行程式。

8051/8052 單晶片微電腦的內部結構主要包括 7 個部份:如下

- 1.中央處理單元(CPU)。
- 2.內部程式記憶體(ROM)。
- 3.內部資料記憶體(RAM)。
- 4.振盪與時序電路。
- 5.I/O埠。
- 6.計時/計數器。
- 7.中斷控制電路。



圖2-1-1 8051單晶片

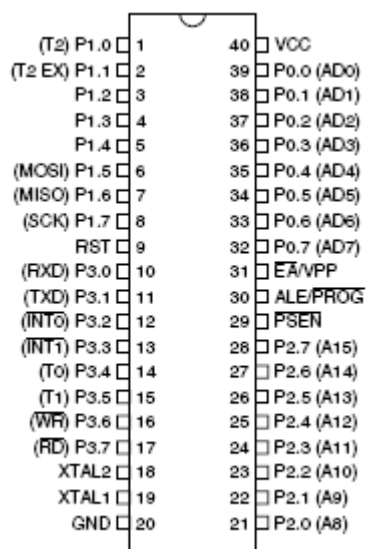


圖2-1-2 8051接腳圖

二、EEPROM 93C66 簡介

專題做的是電子密碼鎖，所以必須儲存密碼，並且在電源消失後仍要能紀錄，若記錄在 8051 中，因為 8051 的資料記憶體都是用 RAM，電源一關掉資料就會 lose 掉，所以必須再外部加上一顆 ROM，且為了方便性，使用電子消除式的 EEPROM，也是採用 ATMEL 的 93C66。

表 2-2-1 93C66 規格圖

	記憶體容量	接腳	指令格式
93C66	4Kbit(256X16)	8	11

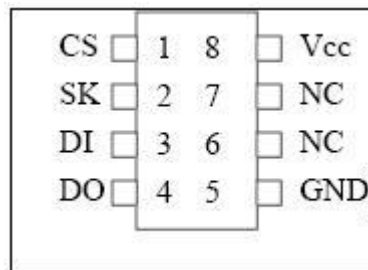


圖2-2-1 93C66 接腳圖

Vcc: 電源接腳 CS: 晶片選擇接腳此接腳為高電位(cs=1)時，才可正確讀寫 SK: 串列資料時脈接腳 DI: 串列資料輸入接腳 DO: 串列資料輸出接腳 GND: 接地腳 NC: 空接腳

93C66 系統方塊圖

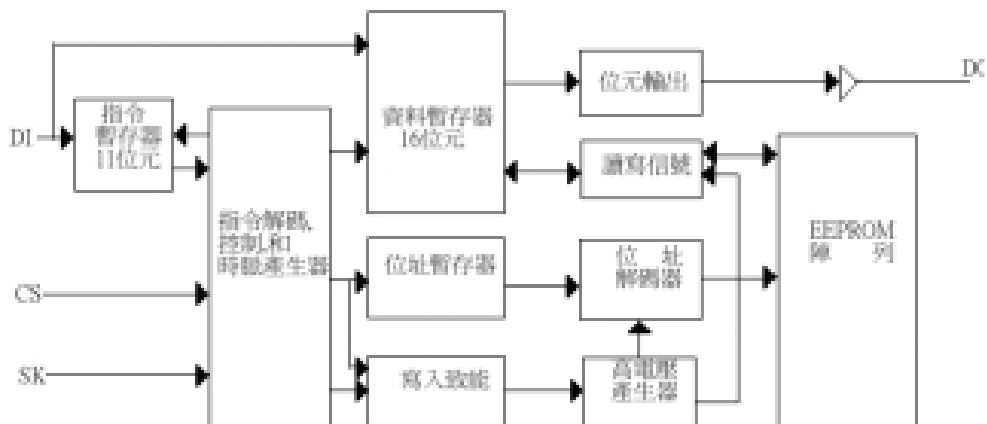


圖 2-2-2 93C66 系統方塊圖

93C66 指令集

表2-2-2 93C66 為 11 位元指令

Instruction	Start bit	OP code	Address	Input Data
Read	1	10	A7~A0	
WEN	1	00	11xxxxxx	
WRITE	1	01	A7~A0	D15~D0
WR-ALL	1	00	01xxxxxx	D15~D0
WDS	1	00	00xxxxxx	
ERASE	1	11	A7~A0	
REAL	1	00	10xxxxxx	

指令說明：

1. 讀取指令

當下達讀取指令後，位址經由解碼器選擇到相對的記憶體暫存器，其內容將被放入16位元串列一位暫存器中，在每個SK的正緣信號，資料依序每次1個bit由D0接腳輸出，且由最高位元D15先開始。

2. 寫入致能指令

當資料要寫入EEPROM之前，必須先下達寫入致能，否則寫入無效，一旦下達寫入致能指令之後，及變成寫入狀態，此時可用寫入指令將資料寫入EEPROM中，此狀態一直維持至下達寫入除能指令或電源消失為止。

3. 寫入指令

寫入指令會將16bit的資料寫入指定位址中，當資料輸入完成之後，必須使CS為低電位(CS=0)置少250ns，在使CS拉回高電位(CS=1)，然後檢查D0腳，當D0=0時，表示寫入動作未完成(Busy狀態)，當D0=1時，表示寫入動作完成(Ready)，寫入動作完成之後，必須下達寫入除能指令，以確保資料的安全性。

4. 全部寫入指令

當下達全部寫入指令後，位於其後的16位元資料，會填入所有的記憶體暫存器中，即其內容皆相同，當資料輸入完成之後，必須使CS為低電位，置少250ns，在使CS拉回高電位(CS=1)，然後檢查D0腳，當D0=0時，表示寫入動作未完成(Busy狀態)，當D0=1時，表示寫入動作完成(Ready)。

5. 寫入除能指令

當下達寫入除能指令之後，將禁止一切嘗試將資料寫入 EEPROM 的動作，以保持其內部資料的完整性，不會被破壞。

6. 清除指令

如果要清除 EEPROM 記憶體暫存器中某一位址的資料時，可以使用清除指令，當下達清除指令完成之後，必須使 CS 為低電位(CS=0)置少 250ns，在使 CS 拉回高電位(CS=1)，然後檢查 D0 腳，當 D0=0 時，表示寫入動作未完成(Busy 狀態)，當 D0=1 時，表示寫入動作完成(Ready)。

7. 全部清除指令

當下達全清除指令後，所有的記憶體暫存內容皆變高電位，此時必須使CS為低電位(CS=0)置少250ns，在使CS拉回高電位(CS=1)，然後檢查 D0 腳，當 D0=0 時，表示寫入動作未完成(Busy 狀態)，當 D0=1 時，表示寫入動作完成(Ready)。

三、藍芽模組

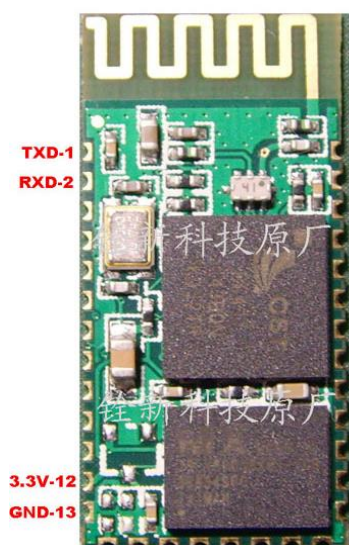


圖 2-3-1 藍芽模組

NO	PIN NAME	NO	PIN NAME
1	UART-TX	18	SPI-MISO
2	UART-RX	19	SPI-CLK
3	UART-CTS	20	USB D+
4	UART-RTS	21	GND
5	PCM-CLK	22	GND
6	PCM-OUT	23	PIO(0)
7	PCM-IN	24	PIO(1)
8	PCM-SYNC	25	PIO(2)
9	AIO(0)	26	PIO(3)
10	AIO(1)	27	PIO(4)
11	RESET	28	PIO(5)
12	3.3V	29	PIO(6)
13	GND	30	PIO(7)
14	GND	31	PIO(8)
15	USB D-	32	PIO(9)
16	SPI-CSB	33	PIO(10)
17	SPI-MOSI	34	PIO(11)

圖 2-3-2 藍芽接腳圖

使用方法：首次和手機連接時需配對，以後就隨連隨用。模組通電後模組上的指示燈是快閃，也就是配對準備狀態，需在30秒內配對，如果30秒內沒配對，就變為慢閃進入等待狀態。配對密碼：1234 如果30秒內沒配對，再配對時需重開電源。藍牙模組和電腦連接後就是那種非常快的閃了，如果有資料傳送，就是長亮，連接成功後，使用第11腳、13腳與電腦端作無線RS232傳輸。

四、RS232 串列介面處理模組

串列傳輸

串列傳輸為CPU與周邊裝置或CPU與CPU間的資料傳輸方法之一，最簡單的串列傳輸只需兩條傳輸線，使用時的方式每次傳輸一個位元的資料，所以具有傳輸線少的優點，並且容易防止雜訊干擾，適合較遠距離的資料傳輸。然而，由於資料傳輸一次僅送一個位元，因此傳輸資料的速度慢是其缺點。

串列傳輸的結構雖然簡單，但也由於太簡略所以產生許多問題，必須藉由傳輸協定來解決。然而，一個完整的傳輸協定包括從硬體到軟體，相當複雜。其中最基本的一種非同步式串列介面（Universal Asynchronous Receiver Transmitter，簡稱UART）常被用於一般的串列傳輸應用中。

串列傳輸在傳送一個位元組時，必須要傳送8次，而UART的串列傳輸方式是在傳送8個位元資料之前加上一個起始位元，並在傳送8個位元資料之後加上一個停止位元，於是原先傳送一個位元組要傳送8次就增為10次。以下是UART串列傳輸的示意圖，傳輸時間順序由左至右：

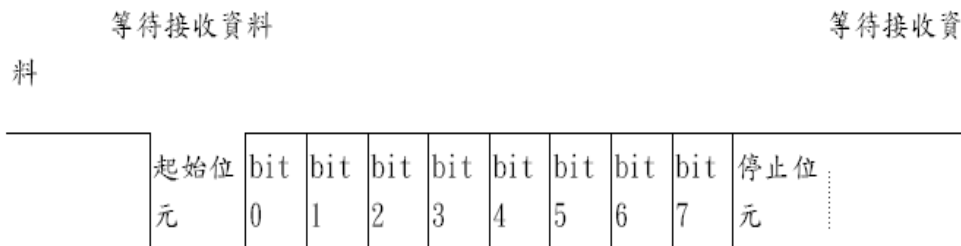


圖 2-4-1 串列傳輸示意圖

在UART的傳輸結構中，起始位元固定為0，停止位元固定為1，所以接收端的動作是一直不斷的檢查傳輸線的狀態。當傳輸線上的信號一直為1就表示沒有資料傳送；當傳輸線上的信號由1變為0，即表示有資料將傳送，接收端就會開始準備接收8個位元資料，直到傳送完8個位元資料，傳送端最後會送出停止位元，並使傳輸線的信號保持為1，以等待下一次的資料傳輸。經由增加起始位元與停止位元方式，雖然會使串列傳輸效率更降低，但可解決位元資料傳輸的起始與停止之問題。另一串列傳輸協定為傳輸速度，通常以鮑率(Baud Rate)，即每秒傳輸的位元數來衡量，一般UART常使用的鮑率有1200、2400、4800、9600及19200等。兩種裝置在進行串列傳輸時，必須決定以何種鮑率來進行資料傳輸，當兩種裝置使用同一鮑率才能確保資料傳輸正確無誤。

UART 的結構

8051單晶片的串列埠是一組全雙工的UART，即就是8051的UART可以在同一時間進行串列資料的傳送與接收。8051單晶片使用P3.0接腳做為串列傳輸的接收端(RXD)，P3.1接腳做為串列傳輸的輸出端(TXD)，並利用特殊功能暫存器(Special Function Register，簡稱SFR)中的串列埠緩衝器(Serial Port Buffer，簡稱SBUF)執行串列傳輸的工作。當串列傳輸工作設定完成之後，傳送端會存入一筆資料到SBUF中，並藉以引發資料傳送的動作；當串列傳輸工作設定完成之後，接收端會將接收資料放入SBUF中。但在8051單晶片的UART結構中，接收資料端以及傳送資料端實際使用上的暫存器並不是同一個，只不過它們是因為均相對應到相同的定址位址，所以因此在傳送或接收資料的時候，8051單晶片會自動選擇使用不同的暫存器，所以8051的串列埠可以同時進行資料的傳送與接收。

8051單晶片進行串列資料傳輸時，串列埠具有輸入緩衝的功能，即當串列埠接收到一筆資料後，會把資料存放至SBUF中，然後繼續接收資料，並在接收或等待接收下一筆資料的過程中處理SBUF中的資料。因此，串列埠可以持續不斷的接收資料，而不必在接收一筆資料後等待該資料完全處理完畢後才能進行下一筆資料的接收。但在第二筆資料被UART接收完畢前，第一筆資料須被完全處理完畢由程式讀入，否則會產生資料流失的問題。

UART 相關暫存器

在SFR記憶體中以及UART相關的暫存器能將其區分為兩個，分別是串列埠控制暫存器(Serial Port Control register，簡稱為SCON)以及電源控制暫存器(Power Control register，簡稱為PCON)。以下是為此二暫存器的結構圖：

SCON：串列埠控制暫存器 (SERIAL PORT CONTROL REGISTER)
位址：98H

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0 SCON. 7 串列埠模式選擇。
SM1 SCON. 6 串列埠模式選擇。
SM2 SCON. 5 在串列埠為模式 2 和 3 時，致能多處理器通信的功能。在模式 2 或 3，如果 SM2=1，則當接收到的第 9 資料位元為 0 時，RI 不動作。在模式 1 時，若 SM2=1，當接收到的停止位元不正確時，RI 也不動作，在模式 0 時，SM2 必須為 0。
REN SCON. 4 由軟體去設定或清除，以決定是否接收串列輸入資料，(REN=1 接收)。
TB8 SCON. 3 在模式 2 或 3 時，為傳送資料時的可規劃資料位元，由軟體控制。
RB8 SCON. 2 在模式 2 或 3 時，接收的可規劃資料位元放在這個位元裡。在模式 1 時，如果 SM2=0，RB8 為接收到的停止位元；在模式 0 時，RB8 沒有用。
TI SCON. 1 傳送中斷旗號，在模式 0 時，在第 8 位元結束時，硬體會將它設為 1，其他模式時，是在停止位元的開始時設定為 1。此位元必須由軟體清除。
RI SCON. 0 接收中斷旗號，在模式 0 時，在第 8 位元結束時，硬體會將它設為 1，其他模式時，在停止位元的一半的時候由硬體設定(參考 SM2)此位元必須由軟體清除。

圖 2-4-2 SCON：串列埠控制暫存器(SERIAL PORT CONTROL REGISTER)

PCON：電源控制暫存器 (POWER CONTROL REGISTER)
位址：(87H)

SMOD	—	—	—	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

SMOD 雙倍速率位元，當串列埠工作於模式 1、2 或 3 時，如使用 Timer1 作速率產生器，且 SMOD=1，則速率為雙倍。
— 保留將來使用
— 保留將來使用
— 保留將來使用
CF1 一般用途
CF0 一般用途
PD 電源下降位元，80C51BH 時，設定此位元為” 1” 就進入電源下降模式 (僅 CHMOS 可以)。
IDL IDLE 模式位元，80C51BH 時，設定此位元為” 1” 就進入 IDLE 模式 (僅 CHMOS 可以)。
若同時寫 1 至 PD 和 IDL 時，PD 優先

圖 2-4-3 PCON：電源控制暫存器(POWER CONTROL REGISTER)

UART 串列埠的四種工作模式

在SCON結構圖中可以得知SCON位元是由模式選擇位元，可規劃資料位元及旗標位元所組成。而PCON結構圖中可知只有SMOD位元與串列埠傳輸速度有關，其他位元則是用於省電模式的設定。

五、馬達和驅動電路

我們選用9V的直流馬達來拉重物，一般直流馬達有兩個線端，一為正端，一為負端，只要在兩線端輸入額定的直流電壓，便可使馬達轉動，若輸入電壓極性正接，則可使馬達轉動，若輸入電壓極性反接，則可使馬達逆轉。

驅動電路部份：經過各種不同規格的馬達測試，以及驅動電路的修改，目前已經可以利用驅動電路穩定的使馬達正常的動作。

如圖所示，L293馬達驅動IC之1A、2A接至8051之P2.0及P2.1，8051給P2.0高電位，P2.1低電位，則L293之1Y輸出9V，2Y零伏，馬達右轉。

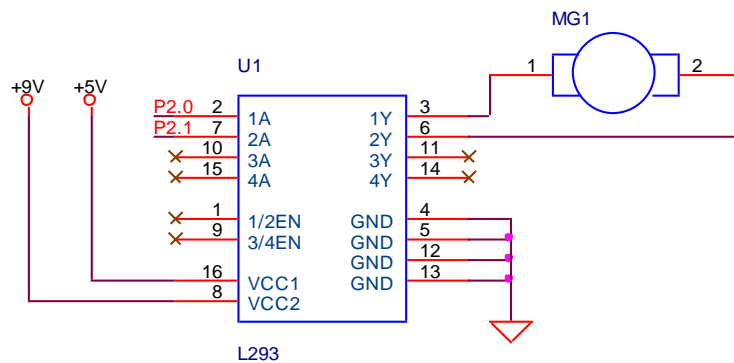


圖2-5-1 馬達驅動圖

六、紅外線防夾

紅外線是在可視紅光光譜之外的不可視光，就因為紅外線也是光的一種，所以它也同樣具有光的特性，它無法穿越不透光的物體。並非因為我們看不到紅外線，就表示它不存在，實際上，在我們生活的四週即充斥著紅外線光，它可能是從電燈發出，也可能太陽光發出，使用者並不需要使用執照即可以使用紅外線。

低速紅外線（Slow IR）應用在電視遙控器上已有相當長的一段時間了，其他像是錄影機、音響等遙控器也是；電視遙控器將特定的訊號編碼，然後透過紅外線通訊技術將編碼送出（通常你可以看到遙控器的訊號燈亮了一下），而設置在電視上的紅外線接收器收到編碼之後，將其進行解碼而得到

原來的訊號；就像是電視收到的訊號為加大音量，在收到訊號後即進行加大音量的動作。

低速紅外線是指其傳輸速率在每秒115.2Kbits者而言，它適用於傳送簡短的訊息、文字或是檔案。有低速紅外線也有高速紅外線（Fast IR），它是指傳輸速率在每秒1或是4Mbits者而言，其他更高傳輸速率則仍在發展中。對於網路解決方案而言，高速紅外線可以說是其基礎，包括檔案傳輸、區域網路連結甚至是多媒體傳輸。

如果將其運用在網路上呢？紅外線網路適用於例如教室的環境，或是小型、封閉的區域。對於講究資訊保全的人而言，紅外線網路或許是一個不錯的選擇，因其無法穿透牆壁傳輸，位在建築物之外的人將不可能直接截取到散佈在建築物內的紅外線訊號。但相對地，這也構成了缺點；紅外線傳輸極容易受到牆壁的阻礙。另一方面，紅外線也是一種低成本的無線傳輸形式。

視線紅外線（Line-of-Sight）以有如雷射線般的直線形態傳輸資料，因為其動作是以直線前進，如果傳輸的路途中沒有任何的障礙物，則資料的傳輸可以說相當快速且具有效率的，因為紅外線也是光的一種，所以它前進的速度是光速。

但是就像太陽光一樣，紅外線不能穿透牆壁或是大型物體，所以，在資料收發的兩端必須相互對準才能進行通訊，這對行動通訊可能常常移動位置的情形而言，是非常不便的，且容易受到下雨、下雪或是霧氣的干擾。

相對於視線紅外線直線前進的運動形態，散射式紅外線（Diffused）就是一種非直接傳輸的方式。散射式紅外線在一定區域中藉著天花板、牆壁及其他物體之表面，以擴散反射的方式達成傳送資料的目的，而接收端則對收集到的訊號進行解譯。

以散射式紅外線進行傳輸的最大好處在於，收送的兩端不必如視線紅外線要正確地對準，但是就因為紅外線是散射形式，所以傳輸範圍及效能將受到更大的限制。

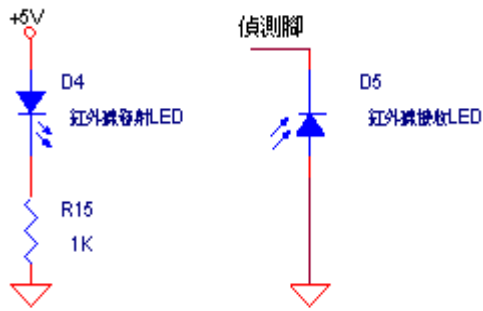


圖2-6-1 遮斷感測電路

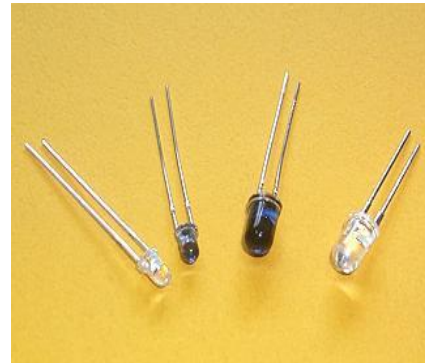


圖2-6-2 紅外線接收發射器

可以將電路看成是距離約10公分的光遮斷器，原理如下，光遮斷器有兩端，一端為紅外線發射端，另一端為接收端，發射及接收會使接收端的電晶體C與E導通。如下圖所示，發光二極體發射紅外線光使NPN電晶體導通時，8051 I/O腳為0V，當有遮蔽物時，紅外線光無法使電晶體導通，這時8051 I/O腳為5V，如此一來便可判斷有無遮蔽物。

上、下門檔為磁簧開關當作感測器，磁簧開關就是有磁力時會有感應，有感應時，磁簧開關裡的鐵片會被磁力吸住而導通，因此就像是開關一樣。

當按下上開鍵時，馬達轉動，門往上捲，上捲時8051要知道門要捲到什麼時後停，所以在軌道的上方用一個上門檔（磁簧開關），並且在門片下方門片裡放入小磁鐵，當門上捲，小磁鐵到達上門檔時，上磁簧開關裡面的鐵片就被吸住，而使P1.2導通到GND，也就是P1.2由1->0來感測門是否到達最上方的位置，這樣上門檔就可以感測門的位置了，如此可以避免自動門超過範圍，下門檔也是同樣的方式。

參、專題製作

一、設備及材料

表 3-1-1 專題製作使用儀器(軟體)設備一覽表

儀器(軟體)設備名稱	應用說明
個人電腦	製作報告、撰寫程式、找尋資料
麵包版	模擬電路
手機	測試藍芽無線傳輸
三用電錶	測試零件
電源供應器	供應所需的電源
MicroSoft Office Word	製作專題書面報告
MicroSoft Office Power Point	製作專題報告簡報
PC版	完成電路
鑽孔機	製作模型
熱熔膠槍	製作模型

二、製作方法和製作步驟

此專題的研究方法是採用身體力行法，主要是由不斷的研究探討，其中包括了準備、實驗、以及電路的分析跟文書報告的處理等此。專題研究步驟。

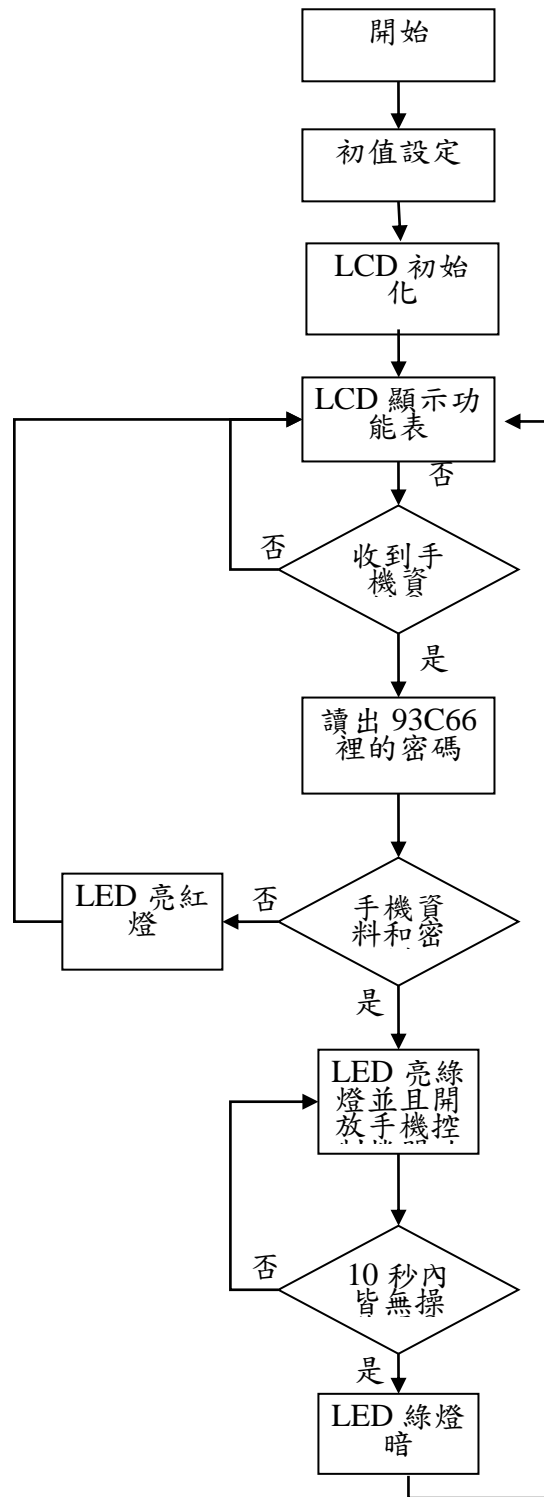


圖 3-2-1 專題製作方法流程圖

三、專題製作

表3-2-1 專題製作計畫書

專題型別	<input type="checkbox"/> 個人型專題 <input checked="" type="checkbox"/> 團隊型專題	
專題性質	單晶片控制研究	
科別／年級	資訊科三年級	
專題名稱	中文名稱	無線密碼捲門控制器
	英文名稱	Bluetooth infrared control password roll-up door
專題內容簡述	<p>利用 89C51 單晶片學習了控制硬體理解單晶片的運用、還有使用 89C51 設計一個紅外線的專題，我們選擇了運用紅外線作為感測的方式，讓無形的紅外線接觸到人體後，啟動感測開關防止壓傷事件、讓此應用於生活當中、使生活能更為方便，也使自己能增廣知識。</p>	
指導老師姓名	葉忠賢 老師	
參與同學姓名	黃郁翔組長	陳展明組員
	黃韋誠組員	
專題執行日期	104 年 09 月 1 日至 105 年 5 月 31 日	

表 3-2-2 無線密碼捲門控制器材料表

項目	數量	編號	規格	
1	2	C1,C6	104	陶瓷電容
2	2	C3,C2	30PF	陶瓷電容
3	1	C4	10uF/25V	電解電容
4	1	C5	220uF/16V	電解電容
5	1	D1	LED(綠)	
6	1	D2	LED(紅)	
7	1	D3	紅外線發射 LED	
8	1	D4	紅外線接收 LED	
9	1	J1	藍芽模組 HC05	
10	1	J2	POWER	電源
11	1	LS1	BUZZER	蜂鳴器
12	1	MG1	3V	
13	1	Q1	A1015	電晶體 PNP
14	1	R1	10K	電阻
15	1	R2	3K	電阻
16	3	R3,R4,R5	300	電阻
17	1	R6	1K	電阻
18	1	SW1	上磁簧開關	
19	1	SW2	下磁簧開關	
20	1	U1	AT89S51	單晶片
21	1	U2	LCM	16X2 LCD
22	1	U3	93C66	記憶 IC
23	1	U4	L293	馬達驅動 IC
24	1	U5	78M05	5V 穩壓 IC
25	1	Y1	11.0592MHZ	振盪器

全部電路圖

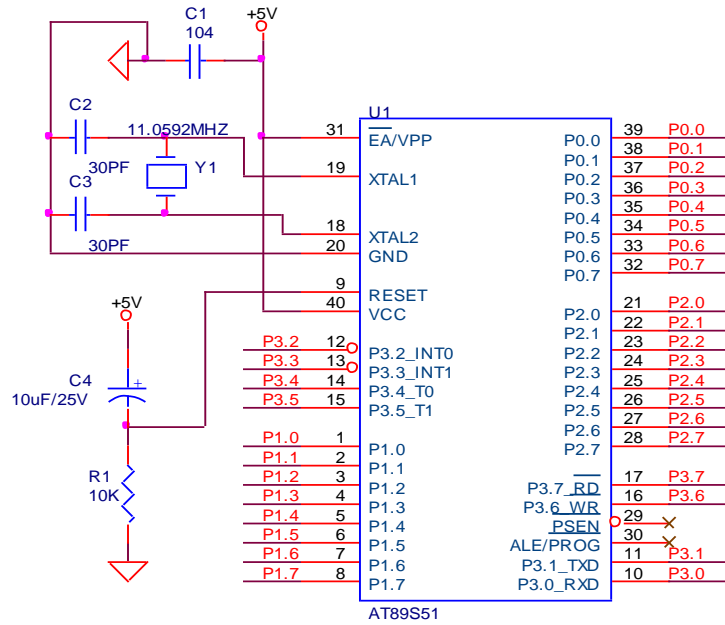


圖 3-2-2 8051 主電路

藍芽模組HC05

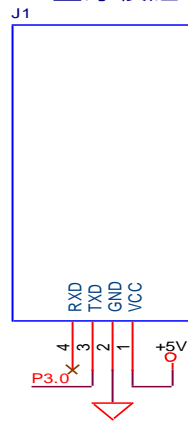


圖 3-2-3 藍芽介面

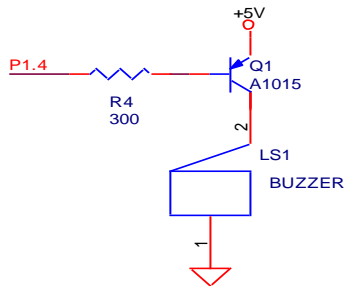


圖 3-2-4 蜂鳴器電路

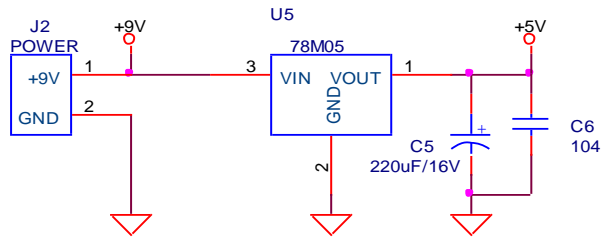


圖 3-2-5 5V 電源電路

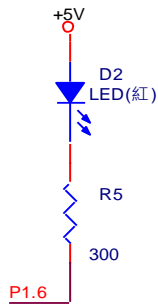
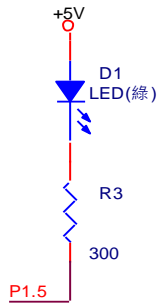


圖 3-2-6 LED 電路

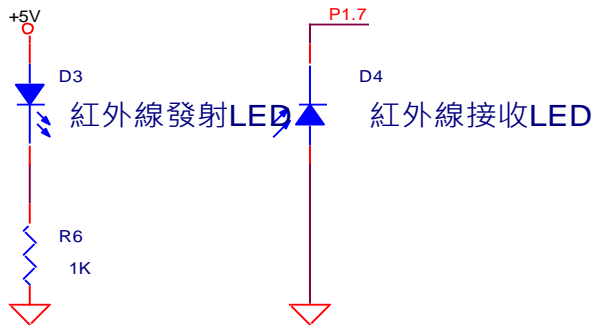


圖 3-2-7 紅外線電路

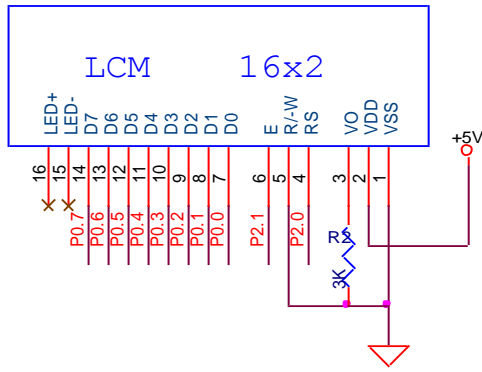


圖 3-2-8 LCD 電路

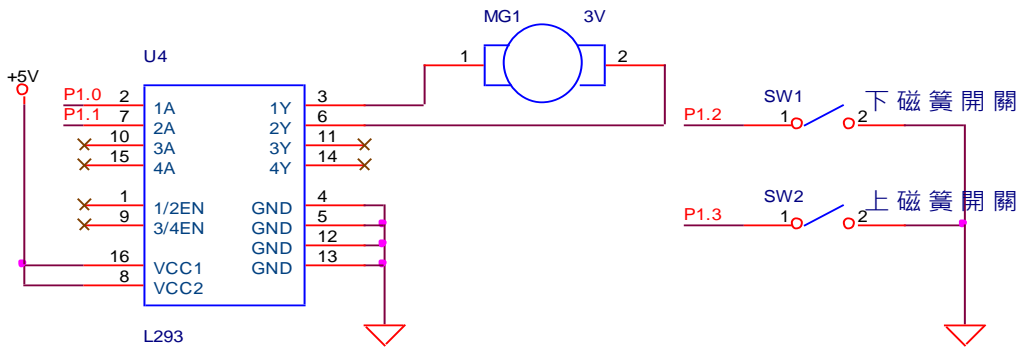


圖 3-2-9 馬達電路

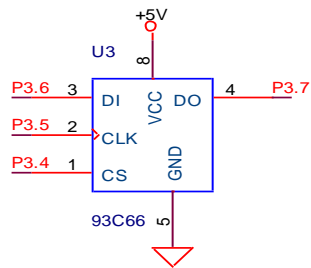


圖 3-2-10 93C66 電路圖

肆、製作成果

(一) 專題製作過程

我目前為止，我們所製作的電路實作部分，面對我們所遭遇的困難以及問題，都已克服了我們的下一步驟即是將我們開始製作外面包裝的成品製作，將之接到我們的，由於有牽涉到木工配置的問題，所以待我們將之完成後，會再以照片及實體的方式慢慢呈現之。我們小組決定題目光控手控自動窗簾，完成焊接且製作整個完整電路，這整個流程中，我們小組都用相機及相關電腦設備將之拍照下來，經將這些資料完整處理過後，我們呈現在這個專題報告之中，如下所示：

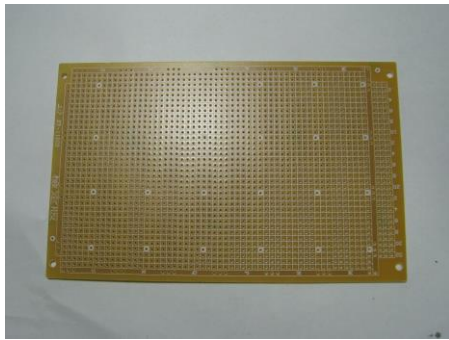


圖4-1-1 專題製作過程(一)

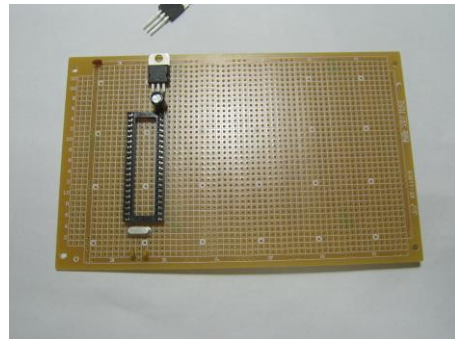


圖4-1-2 專題製作過程(二)



圖4-1-3 無線密碼捲門控制器



圖4-1-4 成品測試功能



圖4-1-5 藍芽操控圖



圖4-1-6 完成無線密碼捲門控制器



圖 4-1-7 文書討論與製作



圖 4-1-8 專題程式製作

(二) 操作方式

打開專題電路板後，接上 DC 12V 變壓器供電給主控板，在進行手機藍芽的連線，在輸入完正確密碼時便會亮起綠色 LED 同時也會響起音效並且在 LCD 顯示器上顯示出正確，但若輸入錯誤密碼時則會亮起紅色 LED 響起音效也在 LCD 上顯示出錯誤，在輸入正確密碼後可以在藍芽程式內可進行密碼的更改，若忘記所設定的密碼時可以按下 reset 鍵密碼將變回原本的 1234。

在進行完密碼認證後，便可以正常執行捲門的上升以及下降功能，在操作上升到頂時便會自動停止，在下降時若紅外線 LED 被遮蔽時，捲門便會停止下降並且同時地發出警告聲響已做為提醒使用者捲門下紅外線收發器被遮蔽。

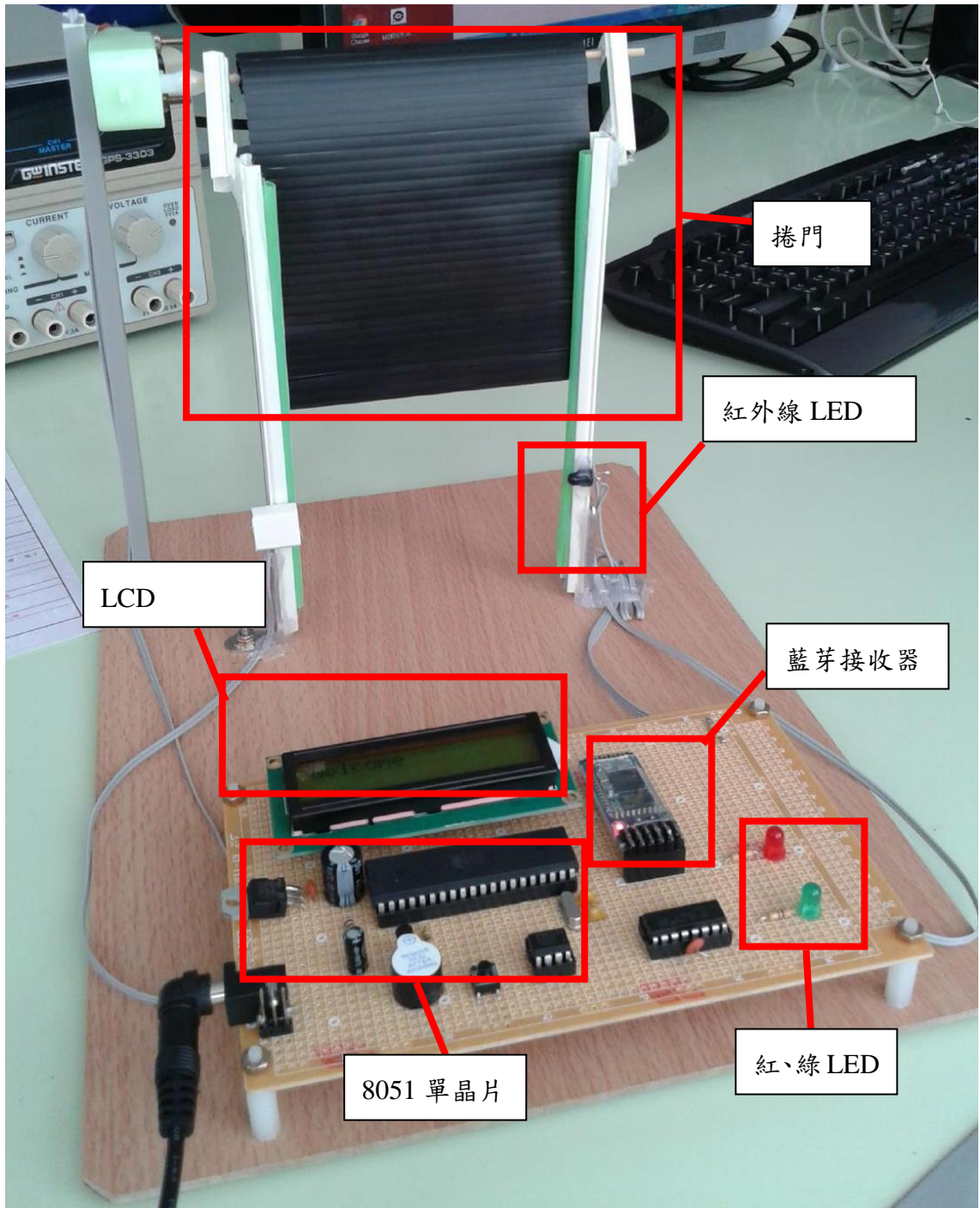


圖 4-2-1 專題成品說明

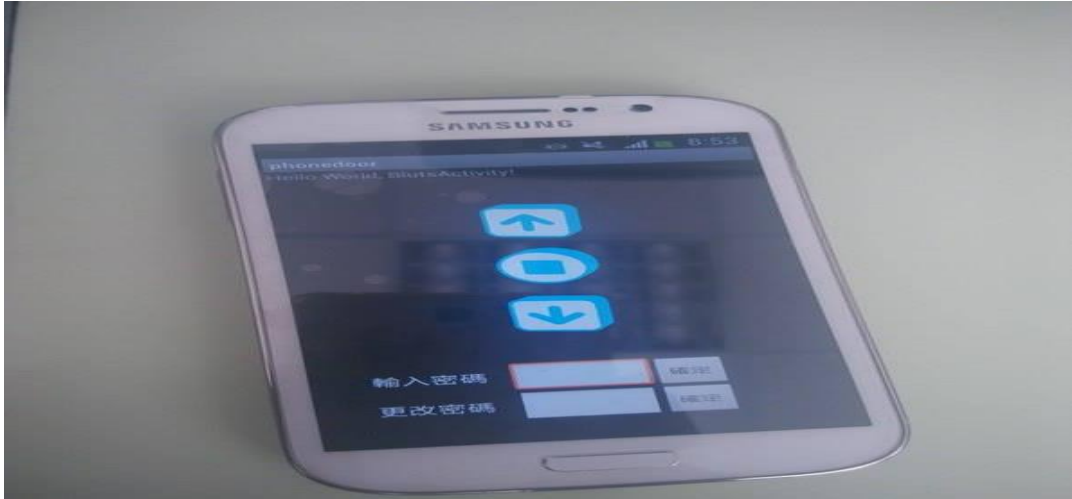


圖 4-2-2 APP 操作畫面

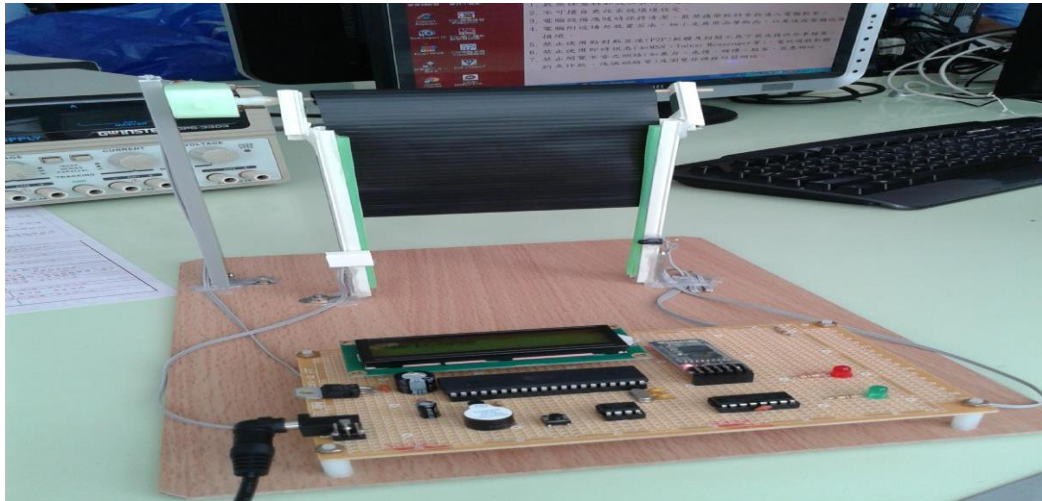


圖 4-2-3 捲門上升



圖 4-2-4 捲門下降

伍、結論與建議

一、結論

研究發現與心得感想起初發現這個題目時，產生了點興趣，便開始著手進行這個作品，我們最先面臨的第一個大問題就是遮斷感測電路緊接而來的問題便是磁性感測電路的使用方法在翻閱書籍還有網路上搜尋了先關資料後，解決了這些問題，不過呢，還是在電路上消耗了一段時間，接著困難又朝我們而來，決定以程式控制馬達轉動時間，但是程式設計的基礎不甚理想，而也是一項艱辛的挑戰，程式碼又都是英文的，翻閱書籍時常常因為看不懂英文而搞不清楚程式實質上的功用，雖然無法完全理解程式碼，但是在找尋資料的方法上卻學到很多，對於往後製作報告是相當大的幫助。未來研究方向這個作品雖然已經無線化但是還不是最完美的，我們可以改善捲動的捲動已達到更好的使用方式。

二、建議

將此遮斷感測電路看成是距離約10公分的光遮斷器，其原理如下，光遮斷器有兩端，一端為紅外線發射端，一端為接收端，發射及接收會使接收端的電晶體C與E導通。如下圖所示，發光二極體發射紅外線光使NPN電晶體導通時，8051 I/O腳為0V，當有遮蔽物時，紅外線光無法使電晶體導通，這時8051 I/O腳為5V，如此一來便可判斷有無遮蔽物。

再將磁簧開關當作感測器，磁簧開關就是有磁力時會有感應，有感應時，磁簧開關裡的鐵片會被磁力吸住而導通，因此就像是開關一樣。

參 考 文 獻

- (1) 吳朗，感測與轉換原理元件與應用，全欣資訊圖書。
- (2) 蔡朝洋，單晶片微電腦8051原理與應用，全華。
- (3) 陳明榮，單晶片8051實作入門，文魁。
- (4) 文淵閣工作室/編著，手機應用程式設計超簡單：App Inventor 2初學特訓班，基峰出版社。
- (5) 王安邦，MIT App Inventor 2 易學易用 開發Android應用程式，上奇資訊出版社。
- (6) 王安邦，中文版 MIT App Inventor 2：易學易用 開發Android應用程式，上奇資訊出版社。
- (7) 黃嘉輝，8051單晶片原理與應用。
- (8) 陳明榮，單晶片8051 KEIL C實作入門第二版。
- (9) 楊明豐，8051入門 C語言輕鬆學。
- (10) 陳明榮，單晶片8051 KEIL C實作入門第三版。

附錄一 無線密碼捲門控制器程式

```

#include <REG51.H> //Keil library
#include <math.h> //Keil library
#include <stdio.h> //Keil library
#include <INTRINS.H> //Keil library
#define uchar unsigned char
#define uint unsigned int
#define DataPort P0 //LCD1602數據埠
#define READ 0x0600 //讀取：xxxxx110,A7-A0
#define WR_EN 0x04c0 //寫入致能xxxxx100,11xxxxxx
#define WRITE 0x0500 //寫入xxxxx101,A7-A0
#define WR_ALL 0x0440 //全部寫入xxxxx100,01xxxxxx
#define WR_DS 0x0400 //寫入除能xxxxx100,00xxxxxx
#define ERASE 0x0700 //清除xxxxx111,A7-A0
#define ER_ALL 0x0480 //全部清除xxxxx100,10xxxxxx
#define ShiftNo 5 //移位次數
#define SIZE 2 //存取93C66字元組數目
sbit RS=P2^0; //LCD1602命令埠
sbit EN=P2^1; //LCD1602命令埠
sbit RST=P2^2; //LCD1602命令埠
sbit DO=P3^7; //93C66串列輸出腳
sbit DI=P3^6; //93C66串列輸入腳
sbit SK=P3^5; //93C66串列脈波輸入腳
sbit CS=P3^4; //93C66致能輸入腳
sbit MOT1=P1^0; //定義腳位
sbit MOT2=P1^1; //定義腳位
sbit sup=P1^3; //定義腳位
sbit sdo=P1^2; //定義腳位
sbit bz=P1^4; //定義腳位
sbit ledg=P1^5; //定義腳位
sbit ledr=P1^6; //定義腳位
sbit IRR=P1^7; //定義腳位
int BP1[]={0,0,0,0};
int BP2[]={0,0,0,0,0,0,0,0};
typedef unsigned char BYTE;
typedef unsigned short WORD;
code char string1[]={"Welcome "}; //LCD 要顯示的字串
code char string2[]={" "}; //LCD 要顯示的字串
code char string3[]={" CODE: "}; //LCD 要顯示的字串
code char string4[]={" NEW CODE: "}; //LCD 要顯示的字串
code char string5[]={" ERROR "}; //LCD 要顯示的字串
code char string6[]={" DATA RESET "}; //LCD 要顯示的字串
code char string7[]={" Correct "}; //LCD 要顯示的字串
uint wi=0,bi=0,dc=0;
int bb,ec=0,tc=0;
void delayms(unsigned int time)
{
    unsigned int n;
    while(time>0) //while作到time=0為止
    {
        n=120;
        while(n>0) n--; //while作到n=0為止
        time--; //time減1
    }
}

```

```

void Out93c66cmd(int command)           //輸出11位元命令至93C66
{
    int i;
    for(i=0;i<11;i++)
    {
if(command & 0x8000)
        DI=1;           //輸出資料1
        else
        DI=0;           //輸出資料0
        SK=1;           //產生串列脈波
        SK=0;
        command<<=1;   //下一位元
    }
} // Out93c66cmd

void CheckBusy93c66(void)               //檢查忙碌旗標
{
    CS=0;           //產生負脈波
    CS=1;
    while(DO==0);   //93C66忙碌中(DO=0)?
} // CheckBusy93c66

void read_93c66(int rd_addr)            //讀取93C66字元資料
{
    int i;
    int rd_cmd;
    int rd_word;
    CS=1;           //致能93C66
    rd_cmd=(READ|rd_addr)<<5; //輸出11位元命令
    Out93c66cmd(rd_cmd);
    for(i=0;i<16;i++) //自93C66讀取16位元字元資料
    {
        SK=1;           //產生串列脈波
        SK=0;
        rd_word<<=1;
        if(DO==1)
        rd_word|=0x0001; //儲存1位元資料
    }
    CS=0;           //除能93C66
    bb=rd_word;
} // read_93c66

void write_93c66(int wr_addr,int wr_word) //寫入字元資料至93C66
{
    int i;
    int wr_cmd;
    CS=1;           //致能
    wr_cmd=(ERASE|wr_addr)<<5; //輸出清除命令至93C66
    Out93c66cmd(wr_cmd); //輸出11位元命令
    CheckBusy93c66(); //檢查忙碌旗標
    CS=0;           //結束清除
命令
    CS=1;
    wr_cmd=(WR_EN<<5); //輸出寫入致能至
93C66
    Out93c66cmd(wr_cmd); //輸出11位元命令
    CS=0;           //結束寫入
致能
    CS=1;

```

```

        wr_cmd=(WRITE|wr_addr)<<5;    //輸出寫入命令至93C66
        Out93c66cmd(wr_cmd);          //輸出11位元命令
        for(i=0;i<16;i++)             //寫入字元資料至93C66
        {
            if(wr_word & 0x8000)
                DI=1;
            else
                DI=0;
        }
SK=1;                                //產生串列脈波
        SK=0;
        wr_word<<=1;                  //下一位元
    }
    CheckBusy93c66();                //檢查忙碌旗標
    wr_cmd=(WR_DS<<5);               //輸出寫入除能至93C66
    Out93c66cmd(wr_cmd);
    CheckBusy93c66();                //檢查忙碌旗標
    CS=0;                             //除能93C66
} // read_93c66

```

```

void wrins(char instruction)
{

```

```

    RS=0;        //lcd控制腳位=0
    EN=0;        //lcd控制腳位=0
    delaysms(1);
    EN=1;        //lcd控制腳位=1
    P0=instruction; //將要設定的指令傳到P0 P0連到LCD的DB0~DB7
    delaysms(1);
    EN=0;        //lcd控制腳位=1
    delaysms(8);
    P0=0Xff;

```

```

}

```

```

void position(char line,column) //設定LCD要顯示的位置
{
    unsigned char instruction;
    line--;
    column--;
    instruction=0x80+(0x40*line + column); //第1行 第1字到最後1字為 80 81 ~8F
                                           //第2行 第1字到最後1字為 C0 C1
~CF
    wrins(instruction); //將位置資料傳到LCD
}

```

```

//對LCM寫入一個字元的函式
void wrdata(char d)
{

```

```

    RS=1;        //lcd控制腳位=0
    EN=0;        //lcd控制腳位=0
    delaysms(1);
    EN=1;        //lcd控制腳位=1
    P0=d;        //將要顯示的字傳到P0 P0連到LCD的DB0~DB7

```

```

    delaysms(1);
    EN=0;          //lcd控制腳位=0
    delaysms(1);
    P0=0Xff;
}

    //對LCM寫入一個字"串"的函式
void display(char *string)
{
    char k=0;
    while(string[k] !=0x00)    //while 當string[k]不為0x00時
    {
        wrdata(string[k]);    //對LCM寫入一個字元的函式
        k++;                  //陣列位置加1
    }
}

//LCM初始設定
void init_lcd()
{
    delaysms(30);
    wrins(0x38);    //LCM初始設定
    wrins(0x38);    //LCM初始設定
    wrins(0x38);    //LCM初始設定
    wrins(0x38);    //LCM初始設定
    wrins(0x08);    //LCM初始設定
    wrins(0x01);    //LCM初始設定
    wrins(0x06);    //LCM初始設定
    wrins(0x0c);    //LCM初始設定
}

void lcd_on()          //LCM主畫面顯示
{
    position(1,1);    //LCM從第1行第1字開始
    display(string1); //顯示string1陣列裡的字串
    position(2,1);    //LCM從第2行第1字開始
    display(string2); //顯示string2陣列裡的字串
}

void save()            //93c66存入密碼
{
    int ik;
    for(ik=0;ik<4;ik++)
    {
        write_93c66(ik,BP2[ik+5]); //存入位置ik指定 內容BP2陣例5~8
        delaysms(10);
    }
}

void readm()          //93c66讀出密碼
{
    int ik;
    for(ik=0;ik<4;ik++)
    {
        read_93c66(ik); //讀出位置ik指定 暫存到BP1陣例0~3
    }
}

```

```

        BP1[ik]=bb;
        delaysms(10);
    }
}

//*****
//主程序*****
//*****
void main()
{
    delaysms(100);        //延時100ms
    TMOD=0x20;           //令計時器1工作於模式2
    TH1=253;             //設定計數值
    TL1=253;             //設定計數值
    TR1=1;               //啟動計時器1
//設定串列埠的模式，並令串列埠中斷致能
    SCON=0x70;           //設定串列埠為模式1
    RI=0;                 //開始接收資料
    TI=0;                 //準備發射資料
    ES=1;                 //把串列埠中斷 致能
    EA=1;                 //把中斷的總開關 致能
    delaysms(100);       //延時100ms
    init_lcd();           //LCM初始設定
    delaysms(500);
    lcd_on();             //lcd顯示主畫面
    readm();              //讀93C66所存的密碼
    readm();              //讀93C66所存的密碼
    while(1)              //主程式迴圈
    {
        if(BP2[0]=='C' && wi==5)    //收到手機傳來C+4位密碼時
        {
            wi=0;
            bz=0;                    //蜂鳴器響1聲
            delaysms(50);
            bz=1;
            position(1,1);           //LCM從第1行第1字開始
            display(string3);        //顯示string3陣列裡的字串
            position(1,7);           //LCM從第1行第7字開始
            wrdata(BP2[1]);wrdata(BP2[2]);wrdata(BP2[3]);wrdata(BP2[4]);    // 密碼
            顯示在LCD上
            if(BP2[1]==BP1[0] && BP2[2]==BP1[1] && BP2[3]==BP1[2] &&
            BP2[4]==BP1[3])//比對密碼
            {ledg=0;                //綠燈亮
                ledr=1;             //紅燈暗
                ec=1;
                position(2,1);      //LCM從第2行第1字開始
                display(string7);    //顯示string7陣列裡的字串
                delaysms(1000);
            }
            else
            {
                ledg=1;             //綠燈暗
                ledr=0;             //紅燈亮
                position(2,1);      //LCM從第2行第1字開始
                display(string5);    //顯示string5陣列裡的字串
                delaysms(1500);
                ledg=1;             //紅、綠燈暗
                ledr=1;
            }
        }
    }
}

```

```

    }
    lcd_on();
}

else if(BP2[0]=='D' && wi==9) //收到手機傳來D+密碼+新密碼時
{
    wi=0;
    bz=0; //蜂鳴器響1聲
    delayms(50);
    bz=1;
    if(BP2[1]==BP1[0] && BP2[2]==BP1[1] && BP2[3]==BP1[2] &&
BP2[4]==BP1[3])//比對密碼
    {
        position(1,1); //LCM從第1行第1字開始
        display(string4); //顯示string4陣列裡的字串
        position(1,11); //LCM從第1行第11字開始
        wrdata(BP2[5]);wrdata(BP2[6]);wrdata(BP2[7]);wrdata(BP2[8]);//顯示新密
碼
        save(); //93C66存入新密碼
    }
    delayms(1500);
    readm(); //讀93C66所存的密碼
    lcd_on(); //LCD顯示主畫面
}

else if(BP2[0]=='u' && wi==1) //收到手機傳來u時
{
    wi=0;
    if(ec==1)
    {
        bz=0; //蜂鳴器響1聲
        delayms(50);
        bz=1;
        MOT1=1;MOT2=0; //馬達轉動 門上開
        tc=0;
        dc=1;
    }
}

else if(BP2[0]=='s' && wi==1) //收到手機傳來s時
{
    wi=0;
    if(ec==1)
    {
        bz=0; //蜂鳴器響1聲
        delayms(50);
        bz=1;
        MOT1=0;MOT2=0; //馬達停
        tc=0;
        dc=0;
    }
}

else if(BP2[0]=='d' && wi==1) //收到手機傳來d時
{
    wi=0;
    if(ec==1)
    {
        bz=0; //蜂鳴器響1聲

```



```

    delayms(50);
    bz=1;
    MOT1=0;MOT2=1;           //馬達轉動 門下關
    tc=0;
    dc=2;
    }
}

if(sup==0 && dc==1)         //上磁簧開關有感測時 且門正在往上時
{
    dc=0;
    MOT1=0;MOT2=0;         //馬達停
}

if(sdo==0 && dc==2)         //下磁簧開關有感測時 且門正在往下時
{
    dc=0;
    MOT1=0;MOT2=0;         //馬達停
}

if(IRR==1 && dc==2)         //紅外線有感測時 且門正在往下時
{
    dc=0;
    MOT1=0;MOT2=0;         //馬達停
    while(IRR==1)          //等待紅外線結束感測
    {
        bz=0;              //蜂鳴器響1聲
        delayms(100);
        bz=1;
        delayms(200);
    }
}

if(tc>2000)                 //綠燈亮超過2000時
{
    ec=0;tc=0;             //停止開放控制門
    wi=0;
    ledg=1;
    ledr=1;
}
else if(ec==1)             //當綠燈亮時 tc計時值加1
{
    tc++;
}
delayms(5);

if(RST==0)                 //重置鍵
{
    while(RST==0);
    BP2[5]='0';BP2[6]='0';BP2[7]='0';BP2[8]='0';
    save();
    position(2,1);         //LCM從第2行第1字開始
    display(string6);      //顯示string6陣列裡的字串
    delayms(1500);
    lcd_on();
}
}

```

```

    }
}

void scon_int (void) interrupt 4
{
    if(RI==1)                //若有接收到手機或RFID傳來資料，則：
    {
        RI=0;                //把RI清除為0
        BP2[wi]=SBUF;
        wi++;

    }
    else TI=0 ;

}
}

```