

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



智慧型停車場

學生姓名： 楊 正 琳

陳 科 甫

鄭 為 仁

指導老師： 葉 忠 賢 老師

中 華 民 國 104 年 05 月

誌 謝

在這高職學校三年吸收到不少專業知識與技能，每當碰到困難不懂，主任、老師們總會為大家解惑，帶給大家學習上大大助益。使我們獲益良多，心中的喜悅是一定有的，對自我的肯定又打了一劑強心針，常每日捫心自問"今天學習進步了多少?到底自己能否獨當一面?"。

此次專題我們學習到組員們的溝通與相互幫忙合作重要性，有大家的幫助才能使專題更完整的呈現。實作中用了不少的材料、工具，讓我們認識零件與它的特性，這樣在製作的過程中，可以更加了解這個專題。

感謝老師協助建議，電路的設計擺放幫我們相當多的忙，還好有你協助為我們減少困難阻礙，使我們更有信心向前邁進。

我們深感，你的教學理念在培養學生上透過專業能力的養成，形成才德兼備、思考敏銳、具有宏觀視野的專業人才

組長楊正琳、組員陳科甫、鄭為仁等 謹上 2015/05

智慧型停車場

摘 要

隨著科技的快速發展，時間的流逝，至從觀太陽、擺鐘到現在的高科技，人類不斷研究，不斷創新紀錄。我們將整合電子、電機、機械等方面的知識，製作一組完整的自動立體停車場包含紅外線感測電路、LCD電路、馬達電路、CPU 電路、等四大電路部分，配合電源、模型、直流馬達、步進馬達等硬體以及作為可自動感應、自動尋找車位、顯示車位之邏輯判斷的程式流程軟體，因此本專題的製作可說是涉及了多方面的知識領域。

其中的核心-CPU，我們選擇了較為熟悉的8051 單晶片，不僅僅是因為在系上大三的課堂中曾學習其理論及實驗應用，更因為它的硬體架構及周邊設備完整、指令集功能強大、程式可複寫功能等種種優勢，正符合我們的需求，而利用程式的模組化，副程式的應用，使程式如堆積木般的組合起來，更容易閱讀及進行修改。

自動停車場的動作原理是藉由LCD的顯示、鍵盤的掃描偵測是否按下按鍵，經由8051 單晶片，由程式進行邏輯判斷後，對直流馬達電路下指令做轉動動作。

關鍵詞：霍爾 IC、8051 單晶片、89C51 單晶片 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	1
四、製作預期成效	2
貳、理論探討	3
參、專題製作	20
一、設備及器材	20
二、製作方法與步驟	20
三、專題製作	22
肆、製作成果	27
伍、結論與建議	30
一、結論	30
二、建議	30
參考文獻	31
附錄一 智慧型停車場程式碼	32

表目錄

表 2-1-1 DDRAM 顯示位置對應圖	3
表 2-1-2 LCD 模組接腳說明	4
表 2-1-3 暫存器接腳與其作用	6
表 2-1-4 清除顯示 Clear Display 圖	7
表 2-1-5 游標歸位 Curse Home 指令圖	7
表 2-1-6 游標歸位 Curse Home 指令圖(2).....	8
表 2-1-7 游標移位控制指令圖	8
表 2-1-8 DD RAM 位址設定指令圖	9
表 2-1-9 旗標 BF 即位址計數器 AC 的內容。指令圖	9
表 2-1-10 CG RAM 或 DD RAM 中指令圖	10
表 2-1-11 CG RAM 或 DD RAM 中讀取資料指令	10
表 2-1-12 LCD 控制指令	10
表 2-2-1 馬達控制 IC	12
表 3-1-1 專題製作使用儀器（軟體）設備一覽	20
表 3-3-1 專題製作計畫書	22
表 3-3-2 智慧型停車場材料表	25

圖目錄

圖 1-3-1 製作流程圖	2
圖 2-1-1 LCD 模組結構	3
圖 2-1-2 LCD 模組接腳圖	4
圖 2-1-3 LCD 模組寫入時序圖	6
圖 2-1-4 LCD 模組讀出時序圖	6
圖 2-1-5 LCD 初始化步驟	11
圖 2-2-1 馬達控制電路圖	12
圖 2-2-2 步進馬達線圈	13
圖 2-2-3 步進馬達特性	14
圖 2-2-4 步進馬達運轉領域	15
圖 2-2-5 步進馬達激磁驅動電路	16
圖 2-2-6 步進馬達激磁(步級角，轉矩小).....	16
圖 2-2-7 步進馬達激磁(步級角，轉矩大).....	17
圖 2-2-7 步進馬達激磁(步級角的 1/2).....	17
圖 2-3-1 霍爾 IC 腳位圖	18
圖 2-3-2 霍爾 IC 配置圖	18
圖 2-3-3 霍爾 IC 配置圖(2).....	19
圖 3-2-1 製作流程圖	21
圖 3-3-1 智慧型停車場之完整電路圖	23
圖 3-3-2 運作流程圖	24
圖 4-1-1 電路板焊接過程	27
圖 4-1-2 檢測電路板	27
圖 4-1-3 電路板製作成品	27
圖 4-1-4 智慧停車場製作成品	27
圖 4-1-5 智慧型停車場報告製作	27
圖 4-1-6 智慧型停車場簡報製作	27
圖 4-1-7 智慧型停車場完整圖	28

圖 4-1-8 升降台完整介面圖	29
------------------------	----

壹、前言

一、製作動機

現在的人口逐漸增加，人們出門都使用機車、汽車作為代步工具，而人們到達目的地時都是使用路邊停車，但是路邊停車會使道路狹窄而造成人們的不方便，不但會使原有道路受阻，而且因為數量不足而導致違規停車之情況，故有智慧型立體停車場的興建，才是首要發展的趨勢，這樣台灣人口們的車輛可以減少交通事故發生。

二、製作目的

本專題研究是希望減少因違規停車而被罰錢，造成道路上行人通行不便，所以本專題的目的可以減少地平面的空間而做上下層的設計，一般停車場二樓層以上都以環繞式行駛到達樓層，若有升降台的功能可以讓在繁忙中的人們在停車場立即尋找到空有的車位，且利用升降台來到達指定樓層，省時又省力。

三、製作架構

(一) 專題製作流程

採用 89C51 作為硬件核心，採用 Flash ROM，內部具有 4KB ROM 存儲空間，能於 3V 的超低壓工作，而且與 MCS-51 系列單晶片完全兼容，但是運用於電路設計中時由於不具備 ISP 在線燒錄技術，當在對電路進行測試時，由於程式的錯誤修改或對程式的新增功能需要燒入程式時，對晶片的多次拔插會對晶片造成一定的損壞。

顯示的部分由 LCD 來顯示，由於 LCD 具備耗電低、電路簡單容易，因此我們選用它。缺點是觀察不易，字體過小。馬達控制晶片採用 L293 晶片，L293 晶片是一種高性能的控制晶片，精度高，控制容易。

(二) 製作流程圖

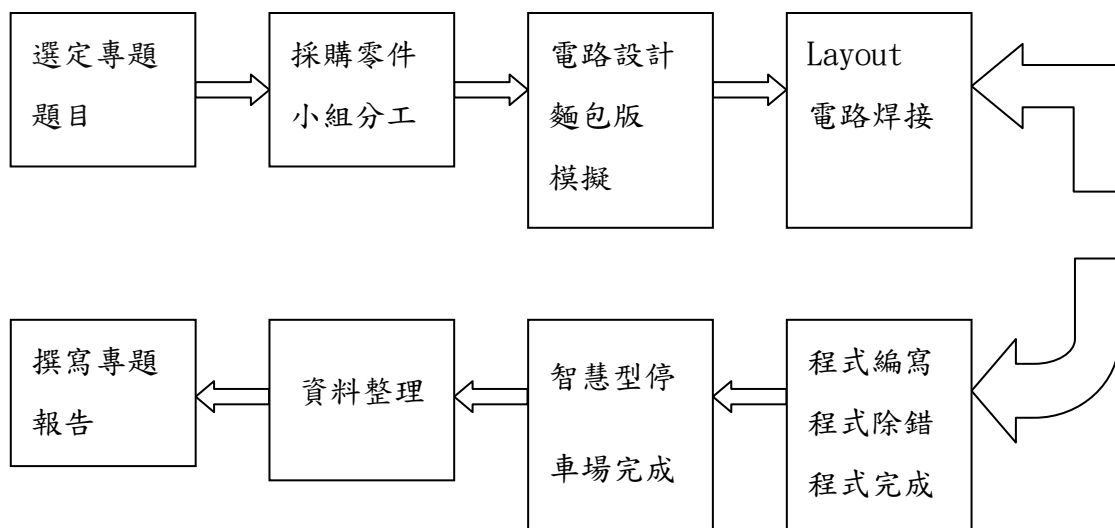


圖 1-3-1 製作流程圖

四、製作預期成效

我們這一個小組雖然是第一次一起進行專題製作...。雖然不知道能否如期完成，為此為這專題初步定義成效如下...

- (一) LCD 顯示 1 樓 2 樓共有 12 個空車位碼，當車子進入停車場時升降台上升。
- (二) 車子停到第 1 個車位，升降台下降，等待車子進入，第二台車子進入停車場尋找第二個車位等待車子停到車位，升降台下降。
- (三) 等待第三台車子進入停車場，升降台上升停到第三個車位((依此類推))。
- (四) LCD 顯示，當第 1 層車位全部停滿，選定第二層。
- (五) 當車子要離開停車場，選定車位的按鍵，升降台必對應到想要離開的車子所在的停車場位置，等待車子進入轉盤後，升降台下降，車子離開停車場。

貳、理論探討

一、LCD 顯示

液晶顯示器，因為工作所需的電流非常低，因此被廣泛使用於高級電器產品及儀表的顯示裝置上。目前市面上的 LCD，大都已經包裝成模組以方便控制，可分文字型及繪圖形 2 種。文字型 LCD 有 16X1，16X2，20X1，20X2，40X1，40X2 數種。

LCD 模組的結構：

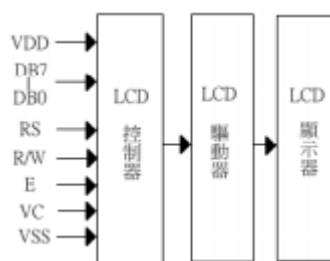


圖2-1-1 LCD模組結構

模組是由控制器，驅動器，顯示器三部份所組成

DDRAM 位址與可供顯示位置的對應圖

表2-1-1 DDRAM顯示位置對應圖

	0	1	2	3		12	13	14	15
第一行	80	81	82	83	。	8C	8D	8E	8F
第二行	C0	C1	C2	C3		CC	CD	CE	CF

16 字 X 2 行 32 字

至於 CPU 如何將資料送給 LCD 顯示的方法，是將欲顯示的字元的 ASCII 碼寫到 LCD 內部的 DDRAM，LCD 就會將這個自在其對應的位置顯示出來。例如，若想在 LCD 的左上角處(位址 80h)，顯示英文字B。就將英文字B的 ASCII 碼(42H)，送到LCD DDRAM 的 80H 位址。

例：

```
MOV      A, #80H           ;設定要顯示的位置為第一行第一字
ACALL    W_INS              ;呼叫副程式，將指令傳至LCD
MOV      A, #"B"           ;顯示值為"B"，也可改成MOV A, #42H
ACALL    W_DATA             ;呼叫副程式，將資料傳至LCD
```

\雖然 LCD 的接腳因製造廠商的不同而排列方式有所不同，不過都是編號由 1 至 14，的 14 根接腳。

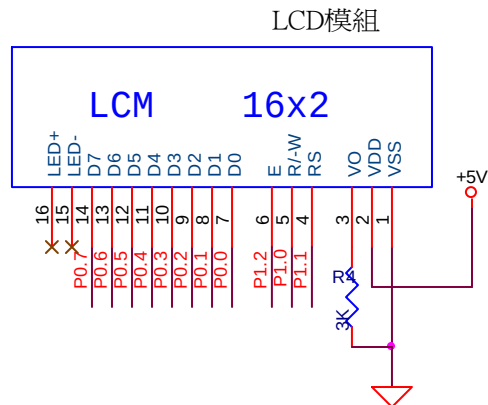


圖 2-1-2 LCD 模組接腳圖

表2-1-2 LCD模組接腳說明

NO	Symbol	Function
1	VSS	接地
2	VDD	主電源 5V
3	VO	顯示明暗對比控制腳 CONTRAST ADJ
4	RS	暫存器選擇信號(Register) RS=0 為資料，RS=1 為指令暫存器
5	R/W	R/W=1 表示從 LCD 模組讀取資料 R/W=0 表示將資料寫入 LCD 模組
6	E	LCD 模組的致能端
7	DB 0	Data Bit 0
8	DB 1	Data Bit 1
9	DB 2	Data Bit 2
10	DB 3	Data Bit 3
11	DB 4	Data Bit 4
12	DB 5	Data Bit 5
13	DB 6	Data Bit 6
14	DB 7	Data Bit 7 但是也拿來傳送 BF

1-2 LCD內部的旗號與暫存器

(一)忙碌旗號 Busy Flag(BF)

BF 旗號是 LCD 用來告訴 CPU 他內部是否再忙碌的一個旗號，若 BF=1，

表示LCD 正在處理內部的工作，因此此時不可以寫資料給 LCD。當BF=0，表示 CPU 可以寫資料給LCD。當接腳 RS=0 且 R/W=1 時，忙碌旗標的接腳會由 DB7 輸出。

;-----

; CHECK LCD BUSY

;-----

CHECK: PUSH A

BUSY: CLR RS

SETB RW

SETB EN

MOV A, P0

CLR EN

JB ACC.7, BUSY ; 忙碌旗標的接腳會由DB7 輸出

ACALL DELAY

POP A

RET

(二)暫存器

LCD 模組內只有 2 個 8 位元暫存器，稱為指令暫存器(Instruction Register IR)，和資料暫存器(Data Register, DR)，他們都是 8 位元暫存器，由 RS 腳來選用。

指令暫存器IR用來接收單晶片送來的命令，例如清除顯示，或功能設定等等...資料暫存器DR則用來接收單晶片要寫到DDRAM(共80Byte)或 CGRAM(共64Byte)的資料緩衝區。當單晶片寫到 DR暫存器之後。LCD內部的控制電路會將資料自動寫到 DD RAM 或 CG RAM 中，而位址是由 LCD內部的位址計數器(Address Counter AC)所指定。而單晶片要讀取資料時，需先將欲讀取的位址放入 IR 暫存器中，LCD就會將其內容放入DR中，然後單晶片就可以去讀取 DR 的資料。

表2-1-3 暫存器接腳與其作用

接 腳		作 用
RS	R/W	
0	0	把指令碼或位址寫入指令暫存器 IR,並執行指令或設定一個要存入/取出資料的位址.
0	1	讀取忙碌旗標 BF 或位址計數器 AC 的內容, DB=7 忙碌旗標 BF 的內容. DB6~DB0=位址計數器 AC 的內容
1	0	把資料寫入資料暫存器 DR 內部會自動執行 DR->DD RAM 或 DR->CG RAM
1	1	從資料暫存器 DR 讀取資料 ¹⁰ 內部會自動執行 DR<-DD RAM 或 DR<-CG RAM

(三)。位址計數器 AC

位址計數器是用來指定欲存取的 DD RAM和 CG RAM 的位址。位址設定指令將位址寫入 IR 暫存器之後，LCD 內部控制電路會將 IR 暫存器的內容送至 AC。當資料存取之後，AC 便會自動加一(I/D=1)或減一(I/D=0)。

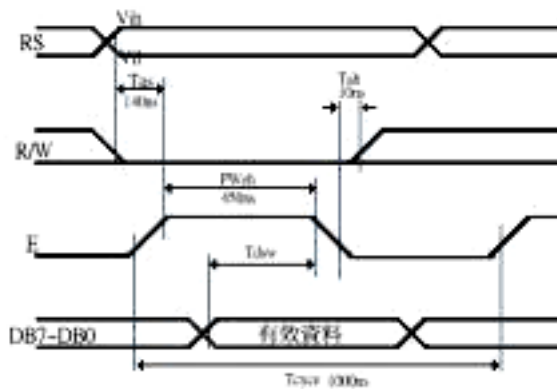


圖 2-1-3LCD 模組寫入時序

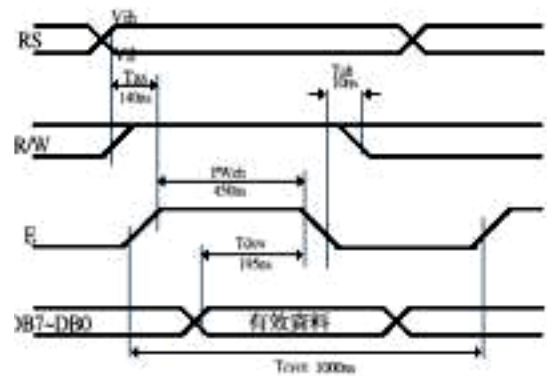


圖 2-1-4 LCD 模組讀出時序圖

(四)。字元產生器 Character Generat ROM (CG ROM)

LCD 內部有一個存放字型的 ROM，它裡面存著 192 個 5x7 點矩陣的字型。這些字型，由存放在 DD RAM 中的 ASCII 碼叫出來顯示。例如：'A'的字型碼為 01000001(即 41H)

(五)。自創字型產生器

LCD 模組除了提供標準的字型 CG ROM 供人使用之外，另外還提供一塊 64 位元組的 CG RAM 空間供人從放 user 自己設計的字，一個字要 8 個位元組

(5x7 點矩陣)，因此 $64/8=8$ ，做多可放 8 個新字元。要顯示自己的字元時，需先將 5X 點矩陣圖形放入 CG RAM 中。而要叫出自元時，其字型碼為 00H~07H。

1-3。LCD模組控制命令

LCD 模組可以接受 CPU 送至 IR 暫存器的命令，並加以執行。其指令共有 11 道，除了 Clear Display 和 Cursor Home 為 1。64ms 之外，其餘都為 40ns 指令。

(一)清除顯示 Clear Display

表2-1-4 清除顯示 Clear Display圖

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
0	0	0	0	0	0	0	0	0	1

動作：將 DDR 資料全部填入空白碼 20H，並將游標移到左上角原點位置

;-----

; CLEAR LCD

;-----

CLRLCD: MOV A, #01H

ACALL W_INS RET

(二)游標歸位 Curse Home指令

表2-1-5 游標歸位 Curse Home指令圖

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
0	0	0	0	0	0	0	0	1	*

動作：DD RAM的資料保持不變，僅將游標移至左上角原點處，及第一行的第一個字，DD RAM 的位址計數器設為 00H。

(三)顯示 ON/OFF 控制(Display ON/OFF)指令

動作：D：顯示器控制位元，D=0 關 D=1，開

C：游標顯示控制位元，C=0 不顯示 C=1 顯示

B：游標閃爍 Blink 控制位元，B=0 不閃爍，B=1 閃爍

表2-1-6 游標歸位 Curse Home指令圖(2)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
0	0	0	0	0	0	1	D	C	B

我們設定為 MOV A, #0CH ;DISPLAY ON/OFF

ACALL W_INS

(四)游標移位控制指令

表2-1-7 游標移位控制指令圖

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

動作: 這個指令可以在不做任何讀寫資料的情況下,移動游標或整個顯示幕向右或向左移.

S/C	R/L	動作說明
0	0	游標左移,AC 值減
0	1	游標右移,AC 值加
1	1	整個顯示幕向左移(所有資料和游標)
1	1	整個顯示幕向右移(所有資料和游標)

(五)功能設定 Function Set 指令

動作： DL：設定介面的位元寬度，DL=1 時為 8 位元，若 DL=0 時為 4 位元。

N：設定顯示器的行數，N=0 時，1 行顯示 N=1 時，2 行顯示。

F：字型 Font 設定。

(六) CG RAM 位址設定指令

動作：設定 CG RAM 的位址，由 A5~A0 之六個位元位址 00H~3FH 位址共 64 個位元組，當寫入本命令之後，接個輸入的資料將寫到 CGRAM 中。

(七)DD RAM 位址設定指令

表2-1-8 DD RAM 位址設定指令圖

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
0	0	1	A6	A5	A4	A3	A2	A1	A0

動作：設定 DD RAM 的位址，由 A6~A0 之七個位元定址，接著寫入的資料將寫入 DD RAM 中。

(八)讀取忙碌旗標 BF 即位址計數器 AC 的內容。指令

表2-1-9 旗標 BF 即位址計數器 AC 的內容。指令圖

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
0	1	BF	A6	A5	A4	A3	A2	A1	A0

動作：若讀出的 BF=1，表示 LCD 模組正忙著內部的工作，因此，無法接受CPU 的命令，直到 BF=0 才可以。在讀 BF 的同時，也會讀到位址計數器的值(DB0~DB6)，這個讀出的位址，可能是 CG RAM 的位址或是 DD RAM 的位址，端看前面設定。

(九)寫資料到 CG RAM 或 DD RAM 中指令

表2-1-10 CG RAM 或 DD RAM 中指令圖

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

動作：寫入 8 位元的資料(D0~D7)到 DD RAM或 CG RAM，至於是寫到那一個，是看前一次式設定 DD RAM或是 CG RAM 的位址。

;-----

; WRITE DATA TO LCD

;-----

W_DATA: PUSH A

ACALL CHECK

SETB RS

CLR RW

SETB EN

MOV P0, A

CLR EN

POP A

RET

(十)自 CG RAM 或 DD RAM 中讀取資料指令

動作：寫入 8 位元的資料(D0~D7)到 DD RAM或 CG RAM，至於是寫到那一個，則是看前一次式設定 DD RAM 或是 CG RAM 的位址

表2-1-11 CG RAM 或 DD RAM 中讀取資料指令圖

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB3	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

表 2-1-12 LCD 控制指令

命 令 (Instruction)	指 令 碼	功 能 說 明
清除顯示器	0 0 0 0 0 0 0 0 0 0 1	將所有顯示的資料清除，且游標回到原點（位址 00H）
游標歸位	0 0 0 0 0 0 0 0 0 1 *	游標回到原點，但 DD RAM 的內容不變
進入模式設定	0 0 0 0 0 0 0 1 I/D S	設定游標移動方向，或讀／寫資料後是否要移標位址是否閃爍（B）
顯示器 ON/OFF 控制	0 0 0 0 0 0 1 D C B	控制所有顯示器（D），或游標（C）的 ON/OFF，和游標位址是否閃爍（B）
顯示器或游標移動方式設定	0 0 0 0 0 1 S/C R/L * *	在不需改變 DD RAM 內容的情況下，移動游標和顯示器資料
功能設定	0 0 0 0 1 DL N F * *	設定界面匯流排寬度（DL）顯示器行數（N），和字形（F）
CG RAM 位址設定	0 0 0 1 AC5 AC4 AC3 AC2 AC1 AC0	設定 CG RAM 位址，以開始接收或傳送 CG RAM 的資料
DD RAM 位址設定	0 0 1 AD6 AD5 AD4 AD3 AD2 AD1 AD0	設定 DD RAM 位址，以開始接收或傳送 DD RAM 的資料
忙碌旗號 / 位址讀取	0 1 BF A6 A5 A4 A3 A2 A1 A0	讀取 LCD 內部的 BF 旗號和位址計數器內容
資料寫到 CG RAM 或 DD RAM	1 0 D7 D6 D5 D4 D3 D2 D1 D0	將資料寫到 DD RAM 或 CG RAM
從 CG RAM 或 DD RAM 讀出資料	1 1 D7 D6 D5 D4 D3 D2 D1 D0	從 CG RAM 或 DD RAM 讀取資料
*:Don't care bit I/D=1:遞加 I/D=0:遞減 S=1:顯示器移動 S=0:顯示器不移動 D=1:要顯示 D=0:不顯示 BF=1:LCD 內部正忙碌 BF=0:LCD 準備好接受資料 C=1:游標 ON C=0:游標 OFF B=1:閃爍 B=0:不閃爍 S/C=1:顯示器移動 S/C=0:游標要移動 F=1:5×10 點矩陣 F=0:5×7 點矩陣 R/L=1:向右移動 R/L=0:向左移動 DL=1:8 位元 DL=0:4 位元 N=1:二行 N=0:一行		

1-4.LCD初始化副程式

LCD 模組在送電後，使用之前，單晶片必須要先去規劃 LCD 的各項功能，或工作模式，LCD 才能正常工作。這個動作稱為 LCD 的初始化。例：功能設定命令

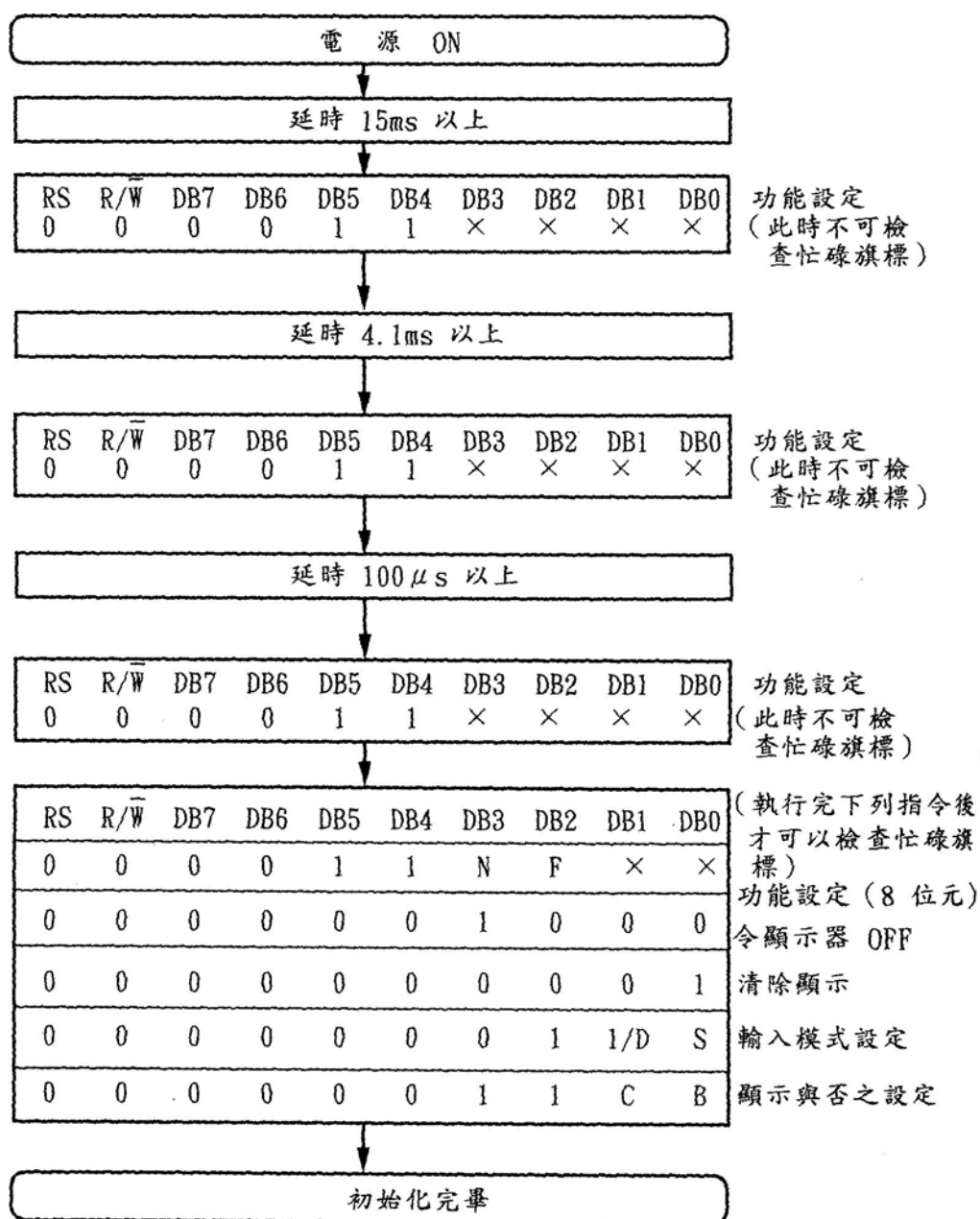


圖 2-1-5 LCD 初始化步驟

二.馬達控制

(一)控制電路

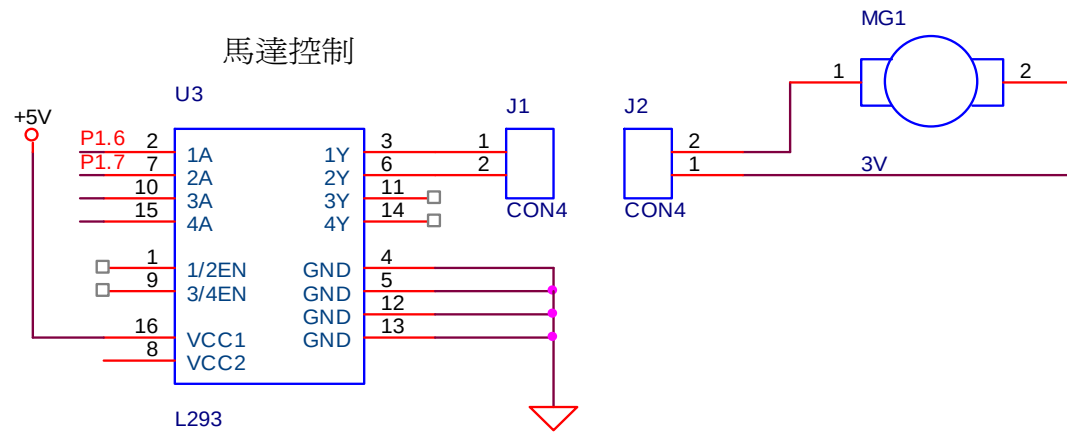


圖 2-2-1 馬達控制電路圖

L293 為馬達控制 IC：

1A 接到 8051 之 P1。6 腳

2A 接到 8051 之 P1。7 腳

控制電梯上升、下降的方法如下表所示

表 2-2-1 馬達控制 IC

執行< 上升 >對策	
馬 達 前 轉	CLR P1。6
	SETB P1。7
執行< 下降 >對策	
馬 達 反 轉	CLR P1。7
	SETB P1。6

(二)步進馬達

步進馬達簡介

相關知識：

步進馬達又稱為步級馬達、脈波馬達。由於步進馬達的特性和一般的交流馬達、直流馬達完全不同，並不是一加上電源就會轉動，因此自成一族，以下說明步進馬達的特性、規格、用法等。

1. 步進馬達的特點

- (1) 旋轉的角度和輸入的脈波數成正比，假設馬達的規格為 0.9° ，即給一脈波轉 0.9° ，400 個脈波轉一圈 360° ，因此用開迴路控制可達高精度角度及高精度定位的要求。
 - (2) 啟動、停止、正反轉的應答性良好，控制容易。
 - (3) 每一步級的角度誤差小，且沒有累積誤差。
 - (4) 在可控制的範圍內，轉速和脈波的頻率成正比，所以變速範圍廣。
 - (5) 靜止時，步進馬達有很高的保持轉矩，可保持在停止位置，不需使用煞車器即不會自由轉動。
 - (6) 在起低速有很高的轉矩。
- 可靠性高，不需保養，整個系統的價格低廉。

2. 步進馬達的用途

- (1) 硬式磁碟機→磁頭定位
- (2) 軟式磁碟機→磁頭定位
- (3) 印表機→紙張傳送，印字頭驅動、色帶驅動。
- (4) 傳真機→紙張傳送
- (5) 影印機→紙張傳送、鏡頭驅動。
- (6) 紙帶閱讀機→紙張傳送
- (7) XY 工作檯→XY 定位

3. 步進馬達的種類

目前所用的步進馬達，以線圈的相數來分，有 2 相步進馬達及 5 相步進馬達：

- (1) 2 相步進馬達是目前使用量最多的步進馬達，基本步級角度有 1.8° 及 0.9° 兩種。內部有兩組線圈，如下圖

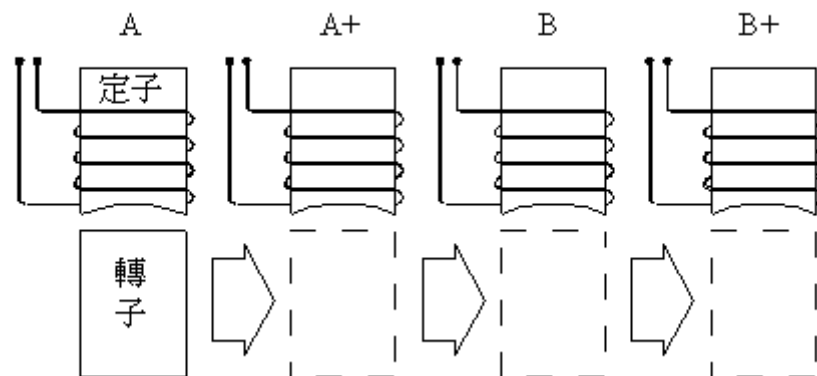


圖 2-2-2 步進馬達線圈

- (2) 5 相步進馬達具有較高的解析度，基本步級角有 0.72° 及 0.36° 兩種。內部有五組線圈。
- (3) 步進馬達依轉子的材料可分三大類：
 1. VR 型步進馬達：可變磁阻型步進馬達，轉子以軟鐵加工而成，步級

角通常為 15 度。

2.PM 型步進馬達：永久磁鐵型步進馬達，轉子是用永久磁鐵製成，步級角有 18 度、15 度、11.25 度、7.5 度多種。

3.HB 型步進馬達：複合型步進馬達，轉子是在永久磁鐵上包以多齒的軟鐵製成，步級角可小於 1.8 度。

4.步進馬達的每一個”線圈組”稱為一”相”。

4.步進馬達的特性

步進馬達的特性可用一些專用的術語加以描述，茲一一說明於下：

(1)速度-轉矩特性曲線下圖所示為步進馬達的速度與轉矩的關係，是選用步進馬達時最常用的圖表。

註：圖中的脫出轉矩、引入轉矩、最大應答週波數...。等會因所用激磁方式、驅動電路的不同而有所差異。

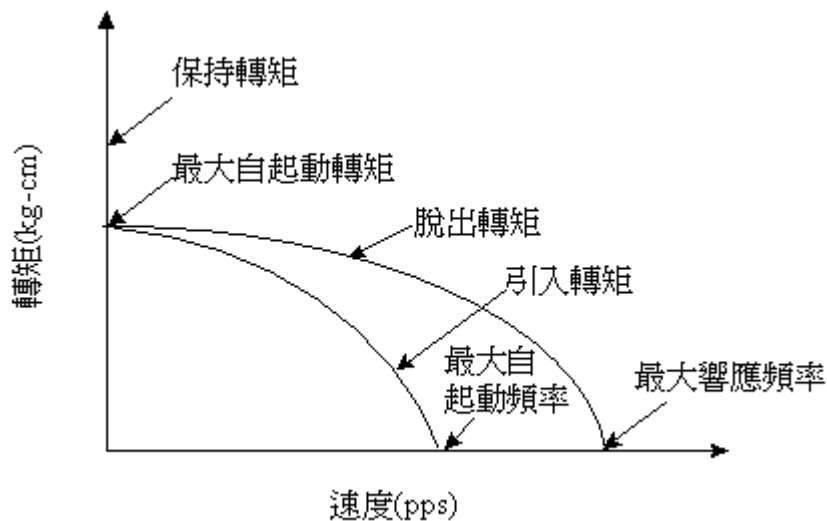


圖 2-2-3 步進馬達特性

(2)保持轉矩

2 相步進馬達採用 2 相激磁，或 5 相步進馬達採用 5 相激磁，各相都通過額定電流而令轉子靜止不動所產生的最大轉矩，稱為激磁最大靜止轉矩。

(3)無激磁保持轉矩

PM 型步進馬達及 HB 型步進馬達的轉子都使用永久磁鐵，所以在各相線圈都沒有通過電流時，還能產生將轉子保持在現有位置時的轉矩，稱為磁保持轉矩。

(4)引入轉矩

這是步進馬達能夠與輸入的脈波信號同步啟動、停止的最大轉矩，負荷大於引入轉矩時，步進馬達無法瞬時啟動，必須先做低速啟動，然後才逐漸提高轉速。

脫出轉矩

步進馬達以某固定的脈波頻率在運轉，輸出軸的負荷逐漸加重，直到失步前的轉矩。當超過脫出轉矩的負荷加於步進馬達時，步進馬達將產生失步的現象而停止轉動。

(1)自啟動領域

指步進馬達在“無負荷”時，能夠與輸入的脈波信號同步而瞬間啟動、停止、正反轉的可能領域。

當加上負荷時，自啟動領域會向左側縮小。

(2)運轉領域

這是步進馬達的高速領域。步進馬達欲在此領域運轉，則輸入的脈波信號頻率必須做緩慢下降之操作，如下圖

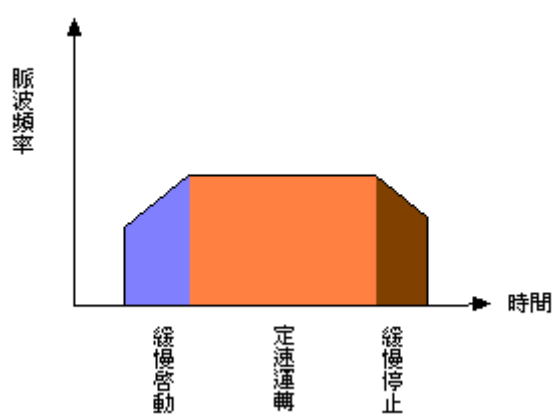


圖 2-2-4 步進馬達運轉領域

在運轉領域步進馬達無法做瞬時啟動、停止、正反轉的操作。如欲在運轉領域驅動馬達，首先須由自啟動領域啟動，再逐漸把脈波的速度加快，此稱為“緩慢啟動”，而欲停止(或反轉)時必須先將脈波的速度逐漸下降到自啟動領域內再停止，此稱為“緩慢停止”。

(3)最大啟動週波數

步進馬達在無負荷時能夠隨入的脈波信號同步而瞬時啟動的最大週波數(最高頻率)。在負荷的慣性大時，最大啟動週波數會變低。

(4)最大應答週波數

這是步進馬達能夠與輸入脈波信號同步運轉的最高週波數(最高頻率)。但是此最大應答週波數會因所有驅動電路的不同再產生差異。

(5)步級角

在輸入一個脈波信號時，步進馬達所旋轉的角度，稱為步級角。步級角會因激磁方式而有所不同，例如基本步級角 1.8° 的 2 相步進馬達，採用 1 相激磁或 2 相激磁時步級角為 1.8° ，採用 1-2 相激磁時步級角變成 0.9° 。

(6)脈波率 PPS

PPS 是步進馬達速度的表示單位。PPS 就是在 1 秒所輸入的脈波數。由於不同的步進馬達控制系統的每一個脈波所產生的步級角不同，所以在同樣的 PPS 時步進馬達的轉速(每分鐘的轉速 rpm)也會有所不同。

(7)共振

步進馬達在某特定的速度領域運轉時，振動會變大而運轉不順暢，產生此現象的速度領域稱為共振領域。每一型步進馬達的共振領域不盡相同，一般的 2 相步進馬達共振領域大約在 100PPS~200PPS 之間。

5.步進馬達的激磁方式

激磁就是令線圈通過電流。我們採用的 2 相步進馬達的驅動電路如下圖所示，有下列三種激磁方式

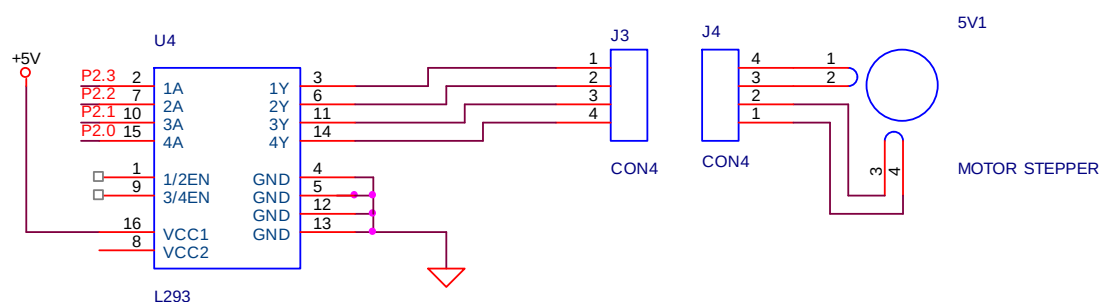


圖 2-2-5 步進馬達激磁驅動電路

(1)相激磁

每次令一個線圈通過電流。步級角等於基本步級角，消耗電力小，角精確度良好，但轉矩小、振動較大。激磁的順序如圖所示

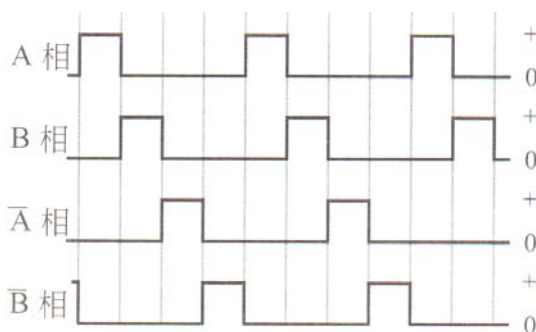


圖 2-2-6 步進馬達激磁(步級角，轉矩小)

(2)相激磁

每次令兩個線圈通過電流。步級角等於基本步級角，轉矩大、振動小，是目前使用最多的激磁方式。激磁的順序如圖所示。

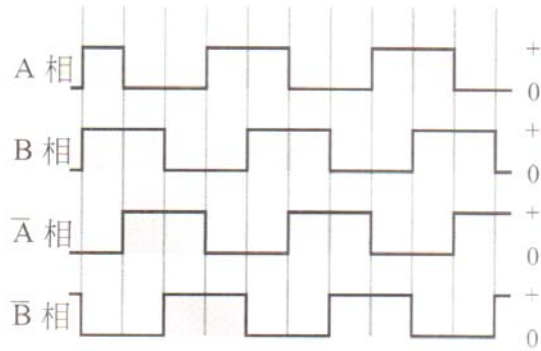


圖 2-2-7 步進馬達激磁(步級角，轉矩大)

(3) 1-2 相激磁

1-2 相激磁又稱為半步激磁，採用 1 相和 2 相輪流激磁，每一步級角等於基本步級角的 $1/2$ ，因此解析度提高一倍，且運轉平滑，和 2 相激磁的方式同樣受到廣泛的使用。激磁的順序如圖所示。

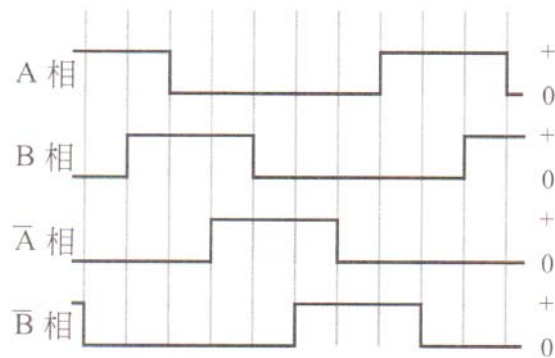


圖 2-2-7 步進馬達激磁(步級角的 $1/2$)

三、霍爾 IC

霍爾 IC

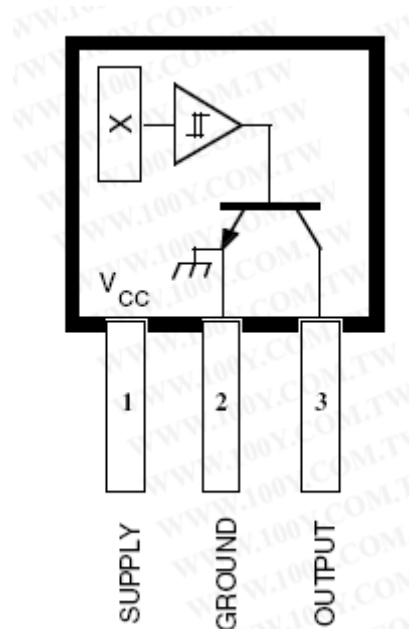


圖 2-3-1 霍爾 IC 腳位圖

型號為 3144

第 1 腳接 5V，第 2 腳接地，第 3 腳有磁場時為 LOW，無磁場時為 HIGH，因此磁鐵與 3144 的配置如下圖所示，則可用 3144 的 HIGH、LOW 來判斷目前升降台的位置以及轉盤的定位點。

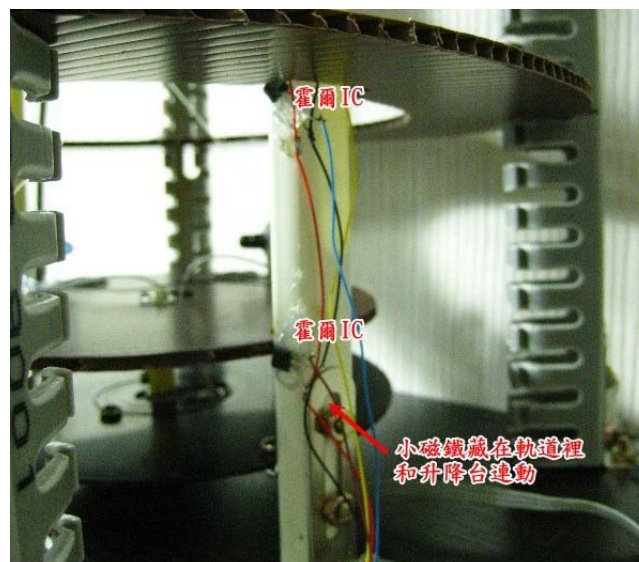


圖 2-3-2 霍爾 IC 配置圖

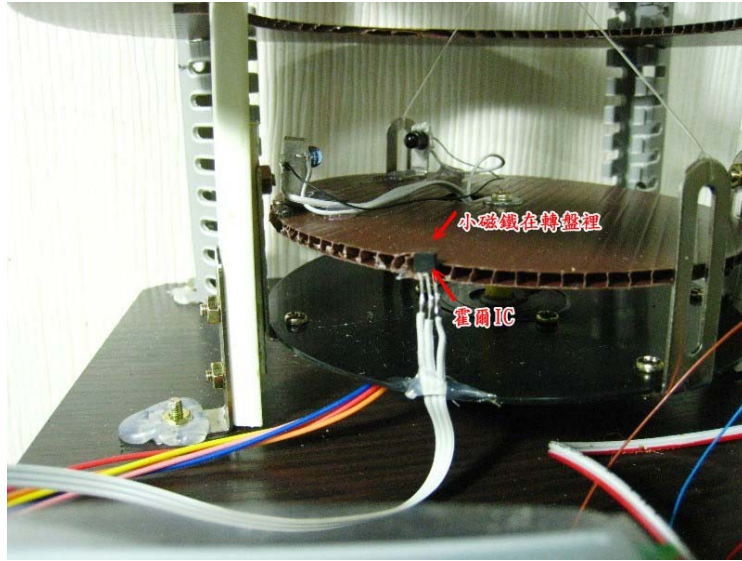


圖 2-3-3 霍爾 IC 配置圖(2)

1. 霍爾集成電路的使用電壓範圍較寬，但實用時電壓宜低不宜高，一般在 4.5-6V 為宜，過高的電源電壓會引起電路的溫升而使電路工作不穩定。
2. 開關型霍爾 IC 驅動負載時，其負載電流應小於霍爾 IC 的負載能力。為了使霍爾的輸出電壓幅度大，一般其輸出端加接較大阻值的負載電阻。
3. 驅動與霍爾 IC 不同的電平的負載時最好加接隔離與緩沖級，可利用光電耦合器或加三極管驅動級。
4. 大多數霍爾 IC 的磁感應距離在 5-10 毫米，須在實用時加以控制，並安置的發信磁鋼應與霍爾 IC 的感應點正對，減小磁路磁阻，使發信與檢測可靠準確。
5. 為了增強開關或線性霍爾 IC 的磁感應靈敏度，使用時亦可利用小磁鋼增強磁偏置或加大發信磁鋼的面積。

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

儀器(軟體)設備名稱	應用說明
個人電腦	專題報告 電路圖製作及進行專題成品電路測試
數位相機	拍攝小組合作過程 專題功能使用及記錄整個專題製作流程
雷射印表機	列印專題資料 圖片及專題報告成果
三用電錶	測量零件有無損壞及專題電路板各信號量
IC萬用燒錄器	利用燒錄器將程式燒錄製89C51單晶片 中
電源供應器	提供專題成品所需之電源
MicroSoft Office Word	專題報告 製作過程的撰寫
MicroSoft Office Power Point	進行口頭報告 製作及專題成品告報呈現
Keil-C	單晶片組合語言城市之編輯 燒錄軟體
Protel 99SE	繪製專題電路之線路圖

二、製作方法及步驟

本專題研究採用的是行動研究法，主要是由循環的研究歷程所構成，包括準備、實驗教學、電路資料分析及報告撰寫等階段。本研究之製作方法與步驟。

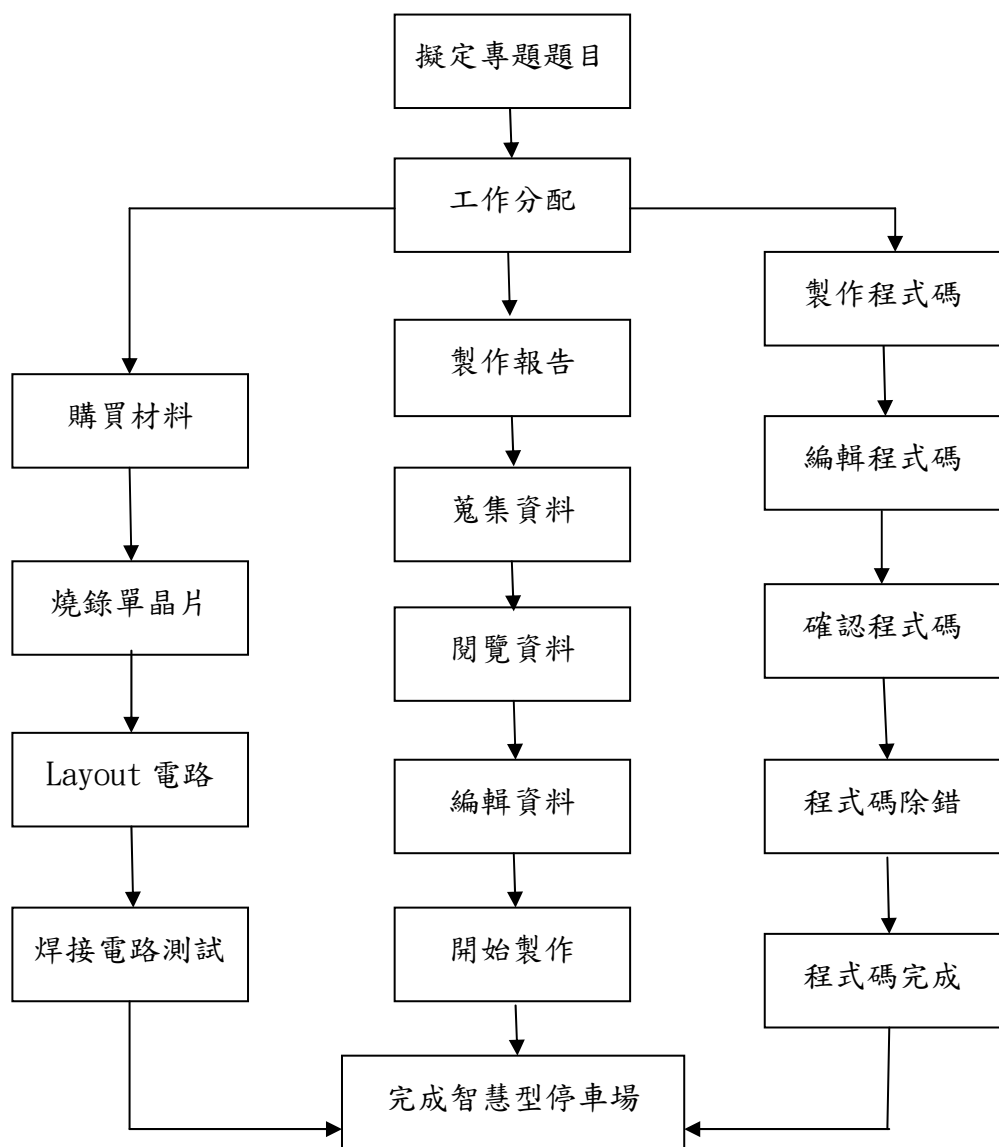


圖3-1-2製作流程圖

三、專題製作

表 3-3-1 專題製作計畫書

專題型別		☐ 個人型專題 ☒ 團隊型專題	
專題性質		利用8051單晶片來驅動升降台及感測	
科別/年級		資訊 科 三 年級。	
專題 名稱	中文名稱	智慧型停車場	
	英文名稱	Smart Parking	
專題內容簡述		本專題選擇熟悉的 8051 單晶片，製作一組完整的智慧型	
		停車場包含紅外線感測電路、LCD電路、馬達電路、CPU	
		電路作為可自動感應、自動尋找車位，智慧型停車場的動	
		作原理是藉由LCD的顯示、鍵盤的掃描偵測是否按下按	
		鍵，經由8051 單晶片，由程式進行邏輯判斷後，對直流	
		馬達電路下指令做轉動動作。	
指導老師姓名		葉忠賢 老師	
參與同學姓名		楊正琳（資訊科3-1）	陳科甫（資訊科3-1）
		鄭為仁（資訊科3-1）	
專題執行日期		103年 09 月 01 日至 104 年 05 月 31 日	

(二)智慧停車場動作之流程圖

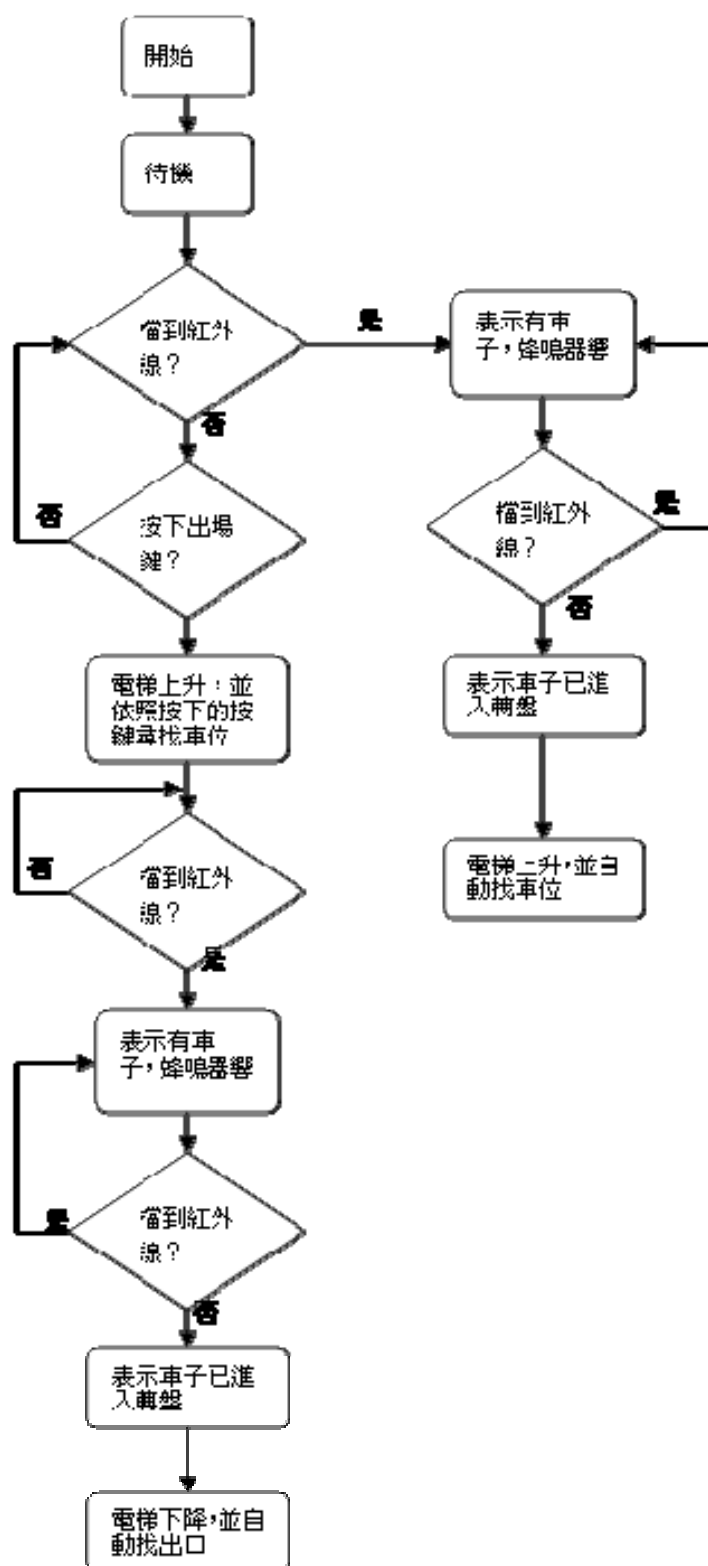


圖3-3-2運作流程圖

表3-3-2 智慧型停車場材料表

項目	數量	編號	規格	
1	1	BZ1	BUZZER	蜂鳴器
2	2	C1,C6	104	陶瓷電容
3	2	C3,C2	30PF	陶瓷電容
4	1	C4	10uF	電解電容
5	1	C5	220uF/16V	電解電容
6	1	D1	紅外線發射LED	
7	1	D2	紅外線接收LED	
8	1	J1	POWER	變壓器插座
9	1	MG1	3V	直流馬達
10	1	Q1	A1015	電晶體PNP
11	1	R1	1K	電阻
12	1	R2	3K	電阻
13	1	R3	300	電阻
14	1	R4	10K	電阻
15	12	SW1,SW2,SW3,SW4,SW5,SW6, SW7,SW8,SW9,SW10,SW11, SW12	KEY	
16	1	SW13	電源開關	
17	1	U1	LCM	16X2 LCD
18	1	U2	89S51	單晶片
19	2	U4,U3	L293	馬達驅動IC
20	1	U5	3144 轉盤定位	霍爾IC
21	1	U6	3144 最下層	霍爾IC
22	1	U7	3144 第1層	霍爾IC
23	1	U8	3144 第2層	霍爾IC
24	1	U9	78M05	5V穩壓IC
25	1	Y1	12MHZ	振盪器
26	1	5V1	MOTOR STEPPER	步進馬達

(三)專題製作成員工作分配及進度安排

專題製作成員： 楊正琳、陳科甫、鄭為仁

前置作業進度： 資料收集，分析討論，決定題目，共約六週

預計完成再需時間：約四個月(16 週)

進度 A。 硬體製作 (陳科甫、楊正琳)

1.研究電路圖： 二週

- 2.搜尋，購買材料：二週
- 3.焊接電路板：二週
- 4.檢查焊線及零件焊接是否正確：二週
- 5.整合軟體 Debug：五週
- 6.整合測試：三週

進度 B。軟體製作 (鄭為仁、楊正琳)

- 1.研究程式資料：三週
- 2.Key in 程式碼：三週
- 3.程式碼加註解：三週
- 4.檢查程式碼及註解是否正確：一週
- 5.編譯及燒錄 8051 程式：一週
- 6.整合硬體 Debug：二週
- 7.整合測試：三週

進度 C。報告書編撰 (陳科甫)

- 1.研究電路圖及程式資料：四週
- 2.搜尋參考技術文件：二週
- 3.報告書相關內容打字：四週
- 4.整合編輯相關文件：三週
- 5.檢查報告書內容是否正確：二週
- 6.報告書列印編冊：一週

肆、製作成果

我們小組由決定題目，製作模擬電路、繪製設計電路圖，進而完成焊接製作整個電路；這整個流程，我們小組都用數位相機及相關電腦設備將之紀錄下來，經將這些資料整理過後，我們將之呈現在我們的專題報告之中，如下所示：



圖4-1-1 電路板焊接過程

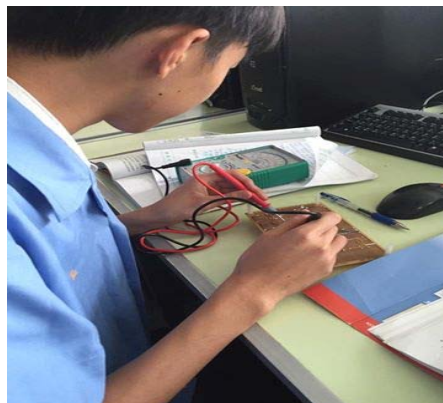


圖4-1-2 檢測電路板



圖4-1-3 電路板製作成品

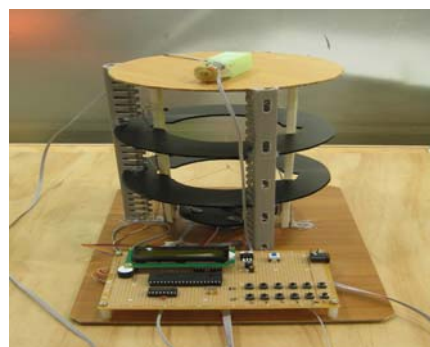


圖4-1-4 智慧停車場製作成品

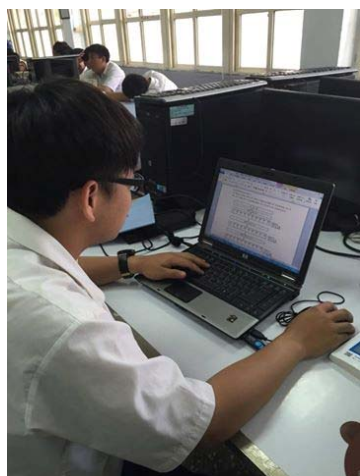


圖4-1-5 智慧型停車場報告製作

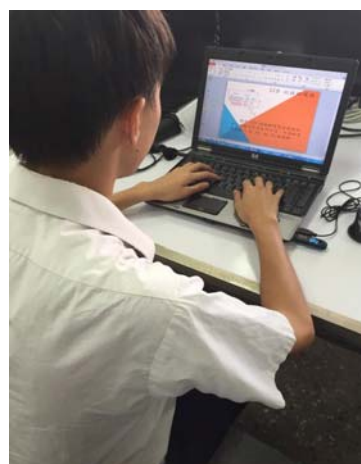


圖4-1-6 智慧型停車場簡報製作

(二操作方法

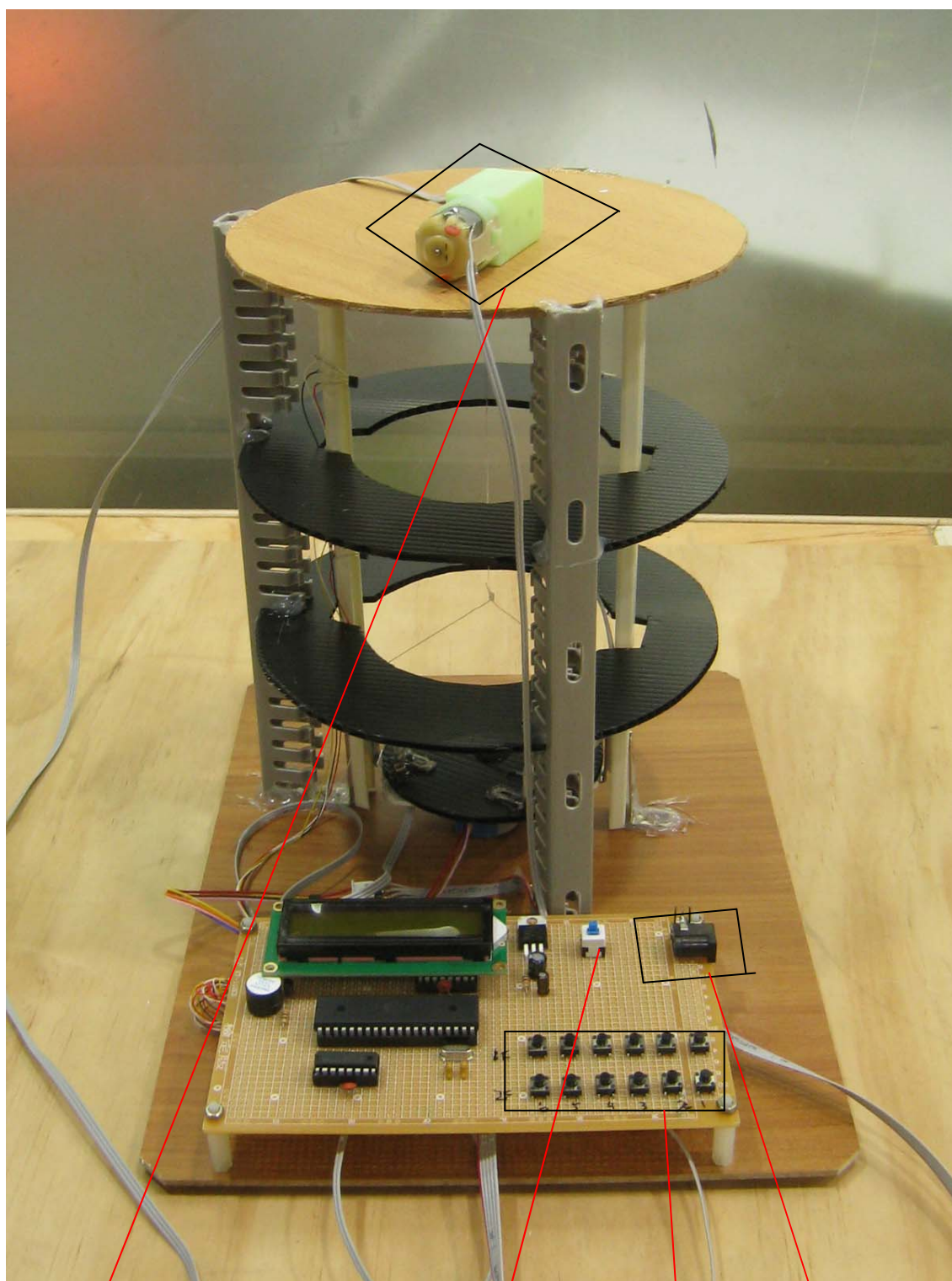


圖4-1-7 智慧型停車場完整圖

直流馬達

開關

車位按鍵

變壓器插座

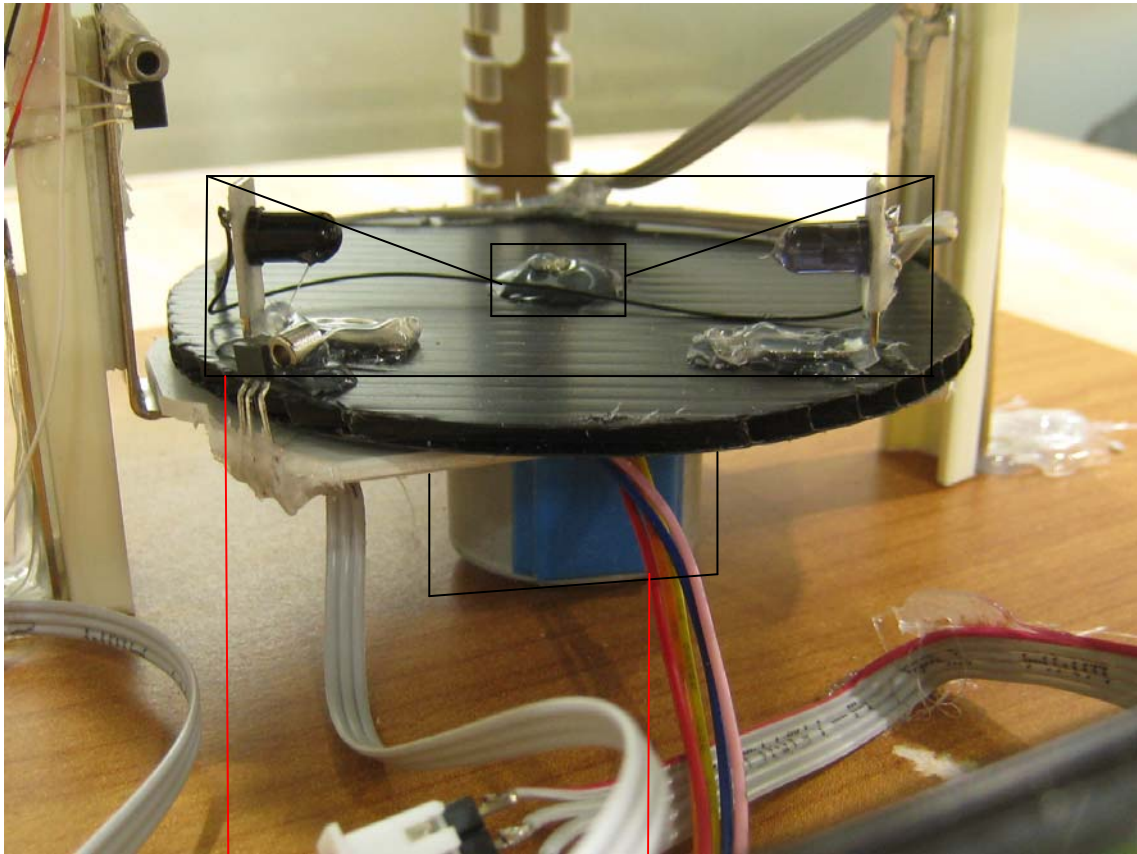


圖4-1-8 升降台完整介面圖

感測器

轉盤驅動馬達

1. 按下開關，LCD面板顯示一樓二樓共有12個車位當車子進入停車場時，升降台上升，轉盤會自動尋找第一個車位。
2. 到達第一個車位子時，升降台下降，等待車子進入，第二台車子進入停車場尋找第二個車位等待車子停到車位，升降台下降。
3. 依此類推，LCD顯示，當第1層車位全部停滿，選定第二層。
4. 當車子要離開停車場，選定車位的按鍵，升降台必對應到想要離開的車子所在的停車場位置，等待車子進入轉盤後，升降台下降，車子離開停車場。

伍、結論與建議

一、結論

本專題我們將針對我們小組對專題製作的整個學習過程，做一最後完整的彙總及記錄，以期作為未來學弟妹們日後學習之參考。

- (一) 透過此次專題製作學習的方式能幫助我們提升對課程的學習興趣。
- (二) 透過此次專題製作學習的方式更能幫助我們獲得知識的建構及整合，且亦可以幫助我們提升其創造思考的能力。
- (三) 小組同學認為專題製作學習為一主動探究的學習，學習中強調學習者必須負起主動探究學習的責任。
- (四) 專題製作學習鼓勵小組成員分工和合作學習的精神。
- (五) 小組同學認同資訊科技的知識在專題製作學習過程中，扮演著重要的角色，因其對電路製作、資料呈現及成果報告製作是很有幫助的。
- (六) 專題製作學習可以培養我們學習者具備問題解決、研究、反省、團體合作及應用資訊科技等多項能力。
- (七) 小組同學認為專題製作學習的階段中，會遇到不同的困難及問題，但看到自己的成品時，會很有成就感。
- (八) 整體而言，我們小組同學認為專題製作學習是一有價值的學習方式，因其確實可以增進自己資訊科技的能力及其技能。

二、建議

此次專題我們小組覺得還能調整升降台的上升及下降速度，可以更省時更有效率。

我們這次製作的智慧型停車場發現團隊合作跟任務分配是很重要的，還有希望以後藉由這個專題可以使違規停車的情形減少也可以使道路的空間變得更寬敞降低車禍的發生，我們小組同心協力的把這個專題製作完成，雖然中間遇到很多問題，但是我們還是平安順利的做完了，這要感謝所有組員的努力與合作，非常感謝你們！

參考文獻

- [1] 黃宏彥、余文俊、楊國輝編著，感測器原理與應用電路實習，高立圖書有限公司，1999。
- [2] 蔡朝陽，電工實習（四），全華科技圖書股份有限公司，中華民國八十三年五月。
- [3] 鄧錦城編著，8051單晶片實作寶典，益眾資訊有限公司，2000。
- [4] 陳明榮，2010，單晶片 8051 KEIL C 實作入門，台灣：松崗
- [5] 楊明豐，2011，8051 入門：輕鬆學，基隆：基峰
- [6] 王振裕、林純民，2012，微電腦控制：8051/8052專題實習(C語言)，台灣：全華圖書
- [7] 林明德，2014，專題製作-電子電路篇，台灣：台科大
- [8] 蔡朝洋、蔡承佑，2010，單晶片微電腦8051/8951原理與應用，台灣，全華圖書
- [9] 鄭鋒、王巧芝、劉瑞圖、高學輝，2012，單晶片微電腦系統開發應用，台灣：上奇資訊
- [10] 江戶川，2014，快學89S51-C，台灣：新文京

附錄一智慧型停車場程式碼

EN REG P1。2
RW REG P1。1
RS REG P1。0

;腳位說明：P2。6 電梯一樓感測，P2。7 電梯二樓感測，P2。5 紅外線，P2。4 轉盤中間感測，

;，P2。0~P2。3 步進馬達，P1。6~P1。7 直流馬達，P1。5 電梯到底感測

CNT EQU 5FH

;

ORG 00H ;設定程式開頭位置

AJMP START

;-----

; MAIN

;-----

START :

ACALL INIT ;呼叫 LCD 初始化副程式

MOV 4AH, #0 ;4AH=0 以下程式會用到

MOV 4BH, #0 ;4BH=0 以下程式會用到

CALL CENTER ;轉盤轉至中間位置

ACALL ON ;顯示功能表副程式

LOOP :

JNB P2.5, LOOP1 ;偵測出入口之紅外線

CLR P1.3 ;蜂鳴器叫

JB P2.5, \$;等待車子完成進入

SETB P1.3 ;蜂鳴器停

ACALL DELAY ;延時

ACALL DELAY ;延時

ACALL DELAY ;延時

CALL JUDGE ;尋找空車位

LOOP1: CALL SCAN ;掃描 12 顆按鍵

JMP LOOP

JUDGE:

MOV A, 30H ;30H=0 時停到第 1 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, J2 ;不是 0 跳 J2 表示已停車子

MOV 3CH, #0 ;轉盤角度值

MOV 30H, #1 ;1 表已有停車

CALL FINDPARK ;轉盤轉至空車位

MOV A, #85H ;設定顯示在 LCD 第 1 行第 6 字上

ACALL W_INS ;給 LCD 指令

MOV A, #01111111B ;顯示符號: <- (ASCII 碼=01111111)

ACALL W_DATA ;給 LCD 資料

RET

J2:

MOV A, 31H ;31H=0 時停到第 2 個位子 (30H~3FH 初始值都為 0)

```

CJNE    A, #0, J3          ;不是 0 跳 J3 表示已停車子
MOV     3CH, #25           ;轉盤角度值 60 度
MOV     31H, #1           ;1 表已有停車
CALL    FINDPARK          ;轉盤轉至空車位
        MOV     A, #87H    ;設定顯示在 LCD 第 1 行第 8 字上 以下依此類推
        ACALL   W_INS
MOV     A, #01111111B
ACALL   W_DATA
RET

```

J3 :

```

MOV     A, 32H            ;32H=0 時停到第 3 個位子 (30H~3FH 初始值都為 0)
CJNE    A, #0, J4          ;不是 0 跳 J4 表示已停車子
MOV     3CH, #50           ;轉盤角度值 120 度
MOV     32H, #1           ;1 表已有停車
CALL    FINDPARK          ;轉盤轉至空車位
        MOV     A, #89H    ;同上段
        ACALL   W_INS
MOV     A, #01111111B
ACALL   W_DATA
RET

```

J4 :

```

MOV     A, 33H            ;33H=0 時停到第 4 個位子 (30H~3FH 初始值都為 0)
CJNE    A, #0, J5          ;不是 0 跳 J5 表示已停車子
MOV     3CH, #75           ;轉盤角度值 180 度
MOV     33H, #1           ;1 表已有停車
CALL    FINDPARK          ;轉盤轉至空車位
        MOV     A, #8BH    ;同上段
        ACALL   W_INS
MOV     A, #01111111B
ACALL   W_DATA

```

RET

J5 :

MOV A, 34H ;34H=0 時停到第 5 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, J6 ;不是 0 跳 J6 表示已停車子

MOV 3CH, #100 ;轉盤角度值 240 度

MOV 34H, #1 ;1 表已有停車

CALL FINDPARK ;轉盤轉至空車位

MOV A, #8DH ;同上段

ACALL W_INS

MOV A, #01111111B

ACALL W_DATA

RET

J6 :

MOV A, 35H ;35H=0 時停到第 6 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, JJ1 ;不是 0 跳 JJ1 表示已停車子

MOV 3CH, #125 ;轉盤角度值 300 度

MOV 35H, #1 ;1 表已有停車

CALL FINDPARK ;轉盤轉至空車位

MOV A, #8FH ;同上段

ACALL W_INS

MOV A, #01111111B

ACALL W_DATA

RET

JJ1 :

MOV A, 36H ;36H=0 時停到第 7 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, JJ2 ;不是 0 跳 JJ2 表示已停車子

MOV 3CH, #0 ;轉盤角度值

MOV 36H, #1 ;1 表已有停車

CALL FINDPARK2 ;轉盤轉至空車位

MOV A, #C5H ;同上段

```

        ACALL    W_INS
MOV     A, #01111111B
ACALL   W_DATA
RET

```

JJ2 :

```

MOV     A, 37H      ;37H=0 時停到第 8 個位子 (30H~3FH 初始值都為 0)
CJNE    A, #0, JJ3  ;不是 0 跳 JJ3 表示已停車子
MOV     3CH, #25     ;轉盤角度值 60 度
MOV     37H, #1      ;1 表已有停車
CALL     FINDPARK2 ;轉盤轉至空車位
        MOV      A, #C7H      ;同上段
        ACALL     W_INS
MOV     A, #01111111B
ACALL   W_DATA
RET

```

JJ3 :

```

MOV     A, 38H      ;38H=0 時停到第 9 個位子 (30H~3FH 初始值都為 0)
CJNE    A, #0, JJ4  ;不是 0 跳 JJ4 表示已停車子
MOV     3CH, #50     ;轉盤角度值 120 度
MOV     38H, #1      ;1 表已有停車
CALL     FINDPARK2 ;轉盤轉至空車位
        MOV      A, #C9H      ;同上段
        ACALL     W_INS
MOV     A, #01111111B
ACALL   W_DATA
RET

```

JJ4 :

```

MOV     A, 39H      ;39H=0 時停到第 10 個位子 (30H~3FH 初始值都為 0)
CJNE    A, #0, JJ5  ;不是 0 跳 JJ5 表示已停車子
MOV     3CH, #75     ;轉盤角度值 180 度

```

```

MOV    39H, #1      ;1 表已有停車
CALL   FINDPARK2 ;轉盤轉至空車位
      MOV    A, #CBH      ;同上段
      ACALL  W_INS
MOV    A, #01111111B
ACALL  W_DATA
RET

```

JJ5 :

```

MOV    A, 3AH      ;3AH=0 時停到第 11 個位子 (30H~3FH 初始值都為 0)
CJNE   A, #0, JJ6 ;不是 0 跳 JJ6 表示已停車子
MOV    3CH, #100    ;轉盤角度值 240 度
MOV    3AH, #1      ;1 表已有停車
CALL   FINDPARK2 ;轉盤轉至空車位
      MOV    A, #CDH      ;同上段
      ACALL  W_INS
MOV    A, #01111111B
ACALL  W_DATA
RET

```

JJ6 :

```

MOV    A, 3BH      ;3BH=0 時停到第 12 個位子 (30H~3FH 初始值都為 0)
CJNE   A, #0, JOK ;不是 0 跳 JOK 表示已停車子
MOV    3CH, #125    ;轉盤角度值 300 度
MOV    3BH, #1      ;1 表已有停車
CALL   FINDPARK2 ;轉盤轉至空車位
      MOV    A, #CFH      ;同上段
      ACALL  W_INS
MOV    A, #01111111B
ACALL  W_DATA

```

JOK : RET

FINDPARK :

```
MOV    A, 3CH      ;角度值->A
MOV    R2, A        ;A->R2
CJNE   A, #0, FI1   ;A=0?(角度0時不用轉) 是0程式往下，否則跳FI1
JMP FIOKK           ;跳 FIOK
```

FI1 :

```
CALL   LEFT        ;轉盤開始轉動，轉一次約4度
DJNZ   R2, FI1      ;假設R2=75 就轉75次的4度=300度
SETB   P2.3
        SETB        P2.2
        SETB        P2.1
        SETB        P2.0
```

FIOKK : CLR P1.6

```
SETB   P1.7        ;電梯往上
```

FIOK :

```
JB     P2.7, FIOK2   ;電梯到二樓?
CLR P1.6             ;電梯停止
CLR P1.7
JNB P2.5, $          ;偵測出入口之紅外線
CLR P1.3             ;蜂鳴器叫
ACALL  DELAY         ;延時
ACALL  DELAY         ;延時
ACALL  DELAY         ;延時
JB     P2.5, $        ;等待車子完全通過
SETB   P1.3          ;蜂鳴器停
ACALL  DELAY         ;延時
ACALL  DELAY         ;延時
```

```

ACALL    DELAY          ;延時
CALL     CENTER         ;轉盤回原點
RET

```

FIOK2 : JMP FIOK

FINDPARK2 :

```

MOV      A , 3CH
MOV      R2 , A
CJNE     A , #0 , FI12
JMP FIOKKK2

```

FI12 :

```

CALL     LEFT
DJNZ     R2 , FI12
SETB     P2 . 3
        SETB     P2 . 2
        SETB     P2 . 1
        SETB     P2 . 0

```

FIOKKK2 :

```

CLR P1 . 6
SETB     P1 . 7          ;電梯往上

```

FIOKK2 :

```

JB  P2 . 6 , FIOK22      ;電梯到一樓？
CLR P1 . 6              ;電梯停止
CLR P1 . 7
JNB P2 . 5 , $           ;偵測出入口之紅外線
CLR P1 . 3
ACALL    DELAY

```

```

ACALL    DELAY
ACALL    DELAY
JB   P2.5, $           ;等待車子完全通過
SETB    P1.3
ACALL    DELAY
ACALL    DELAY
ACALL    DELAY
CALL     CENTER
RET
FIOK22 :    JMP FIOKK2
;=====
; SCAN KEY           ;參閱 6-3§SCAN 程式說明
;=====
SCAN :    MOV        R4, #0           ;清除 R4，R4 存放掃描按鍵值
          MOV        R7, #2           ;R7 設定為 2，掃描 2 次
          CLR P3.0           ;按鍵第一排開始掃描
          SETB    P3.1
COL :
          JB        P3.7, S1          ;無按鍵(1)~S1，有按鍵(0)往下執行
          JNB        P3.7, $           ;消除彈跳誤動作，即按鍵(0)等待放開按鍵
(1)往下執行
          ACALL    DELAY
          ACALL    DELAY
          ACALL    DELAY
          SJMP     S7
S1 :      INC        R4
          JB        P3.6, S2
          JNB        P3.6, $
          ACALL    DELAY

```



```

        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7
S2 :    INC      R4
        JB       P3 ◦ 5 , S3
        JNB      P3 ◦ 5 , $
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7

S3 :    INC      R4
        JB       P3 ◦ 4 , S4
        JNB      P3 ◦ 4 , $
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7
S4 :    INC      R4
        JB       P3 ◦ 3 , S5
        JNB      P3 ◦ 3 , $
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7
S5 :    INC      R4
        JB       P3 ◦ 2 , S6
        JNB      P3 ◦ 2 , $
        ACALL    DELAY
        ACALL    DELAY

```

```

        ACALL    DELAY
        SJMP     S7

S6:      INC     R4
        CPL P3.0
        CPL P3.1
        DJNZ     R7, COL
        RET

S7:      CJNE    R4, #0, S8
        MOV     30H, #0      ;0 表無停車
        MOV     3CH, #0      ;轉盤角度值
        CALL    FINDPARK    ;轉盤轉至空車位
        MOV     A, #85H
        ACALL   W_INS
        MOV     A, #' '
        ACALL   W_DATA
        RET

S8:      CJNE    R4, #1, S9
        MOV     31H, #0      ;0 表無停車
        MOV     3CH, #15     ;轉盤角度值
        CALL    FINDPARK    ;轉盤轉至空車位
        MOV     A, #87H
        ACALL   W_INS
        MOV     A, #' '
        ACALL   W_DATA
        RET

S9:      CJNE    R4, #2, SA
        MOV     32H, #0      ;0 表無停車

```

```

MOV    3CH, #30      ;轉盤角度值
CALL   FINDPARK      ;轉盤轉至空車位
      MOV    A, #89H
      ACALL  W_INS
MOV    A, #'
ACALL  W_DATA
RET

SA:    CJNE   R4, #3, SB
      MOV    33H, #0    ;0 表無停車
      MOV    3CH, #45    ;轉盤角度值
      CALL   FINDPARK    ;轉盤轉至空車位
      MOV    A, #8BH
      ACALL  W_INS
      MOV    A, #'
      ACALL  W_DATA
      RET

SB:    CJNE   R4, #4, SC
      MOV    34H, #0    ;0 表無停車
      MOV    3CH, #60    ;轉盤角度值
      CALL   FINDPARK    ;轉盤轉至空車位
      MOV    A, #8DH
      ACALL  W_INS
      MOV    A, #'
      ACALL  W_DATA
      RET

SC:    CJNE   R4, #5, SD
      MOV    35H, #0    ;0 表無停車
      MOV    3CH, #75    ;轉盤角度值
      CALL   FINDPARK    ;轉盤轉至空車位
      MOV    A, #8FH

```

```

        ACALL    W_INS
MOV     A, #' '
        ACALL    W_DATA
        RET
SD:     CJNE     R4, #6, SE
        MOV      36H, #0      ;0 表無停車
        MOV      3CH, #0      ;轉盤角度值
        CALL     FINDPARK2 ;轉盤轉至空車位
        MOV      A, #C5H
        ACALL    W_INS
        MOV      A, #' '
        ACALL    W_DATA
        RET
SE:     CJNE     R4, #7, SF
        MOV      37H, #0      ;0 表無停車
        MOV      3CH, #15     ;轉盤角度值
        CALL     FINDPARK2 ;轉盤轉至空車位
        MOV      A, #C7H
        ACALL    W_INS
        MOV      A, #' '
        ACALL    W_DATA
        RET
SF:     CJNE     R4, #8, SG
        MOV      38H, #0      ;0 表無停車
        MOV      3CH, #30     ;轉盤角度值
        CALL     FINDPARK2 ;轉盤轉至空車位
        MOV      A, #C9H
        ACALL    W_INS
        MOV      A, #' '
        ACALL    W_DATA

```

```

RET
SG :    CJNE    R4 , #9 , SH
        MOV     39H , #0      ;0 表無停車
        MOV     3CH , #45     ;轉盤角度值
        CALL    FINDPARK2 ;轉盤轉至空車位
        MOV     A , #CBH
        ACALL   W_INS
        MOV     A , #' '
        ACALL   W_DATA
        RET
SH :    CJNE    R4 , #10 , SI
        MOV     3AH , #0      ;0 表無停車
        MOV     3CH , #60     ;轉盤角度值
        CALL    FINDPARK2 ;轉盤轉至空車位
        MOV     A , #CDH
        ACALL   W_INS
        MOV     A , #' '
        ACALL   W_DATA
        RET
SI :    CJNE    R4 , #11 , SOK
        MOV     3BH , #0      ;0 表無停車
        MOV     3CH , #75     ;轉盤角度值
        CALL    FINDPARK2 ;轉盤轉至空車位
        MOV     A , #CFH
        ACALL   W_INS
        MOV     A , #' '
        ACALL   W_DATA
        RET
SOK :

```

```

    RET
;-----
; POWER ON
;-----
ON :    PUSH    A
        MOV     A , #80H
        ACALL   W_INS
        MOV     CNT , #16
        MOV     DPTR , #MSG1
        ACALL   W_STR
        MOV     A , #C0H
        ACALL   W_INS
        MOV     CNT , #16
        MOV     DPTR , #MSG2
        ACALL   W_STR
    POP A
    RET

```

CENTER : ;轉盤定位至中間位置

```

    JNB P1.5 , CE2 ;電梯是否在最底
    SETB    P1.6
    CLRP1.7      ;電梯往下
    JMP CE3
CE2 :    CLRP1.6
        CLRP1.7      ;電梯停
        MOV     4AH , #1
CE3 :
        CLR     P2.0

```

```

    SETB    P2.1
    SETB    P2.2
    SETB    P2.3
JNB P2.4, CBACK ;轉盤中間之磁簧開關
    ACALL   DELAYB
    SETB    P2.0
    CLR     P2.1
SETB    P2.2
    SETB    P2.3
JNB P2.4, CBACK
    ACALL   DELAYB
    SETB    P2.0
    SETB    P2.1
CLR     P2.2
    SETB    P2.3
JNB P2.4, CBACK
    ACALL   DELAYB
JMP CENTER
CBACK :   MOV    4BH, #1
    SETB    P2.0
    SETB    P2.1
    SETB    P2.2
    SETB    P2.3

```

MOV A, 4AH ;4AH、4BH 要皆為 1，表示電梯到底且轉盤在中間才
能返回

```
ANL A, 4BH
CJNE A, #1, CB2
MOV 4AH, #0
MOV 4BH, #0
RET
```

CB2 :

```
JMP CENTER
```

LEFT :

```
CLR P2.3
SETB P2.2
SETB P2.1
SETB P2.0

ACALL DELAYB
SETB P2.3
CLR P2.2
SETB P2.1
SETB P2.0

ACALL DELAYB
SETB P2.3
SETB P2.2
CLR P2.1
SETB P2.0
```


ACALL	DELAYB
SETB	P2 ◦ 3
SETB	P2 ◦ 2
SETB	P2 ◦ 1
CLR	P2 ◦ 0

ACALL	DELAYB
CLR	P2 ◦ 3
SETB	P2 ◦ 2
SETB	P2 ◦ 1
SETB	P2 ◦ 0

ACALL	DELAYB
SETB	P2 ◦ 3
CLR	P2 ◦ 2
SETB	P2 ◦ 1
SETB	P2 ◦ 0

ACALL	DELAYB
SETB	P2 ◦ 3
SETB	P2 ◦ 2
CLR	P2 ◦ 1
SETB	P2 ◦ 0

ACALL	DELAYB
SETB	P2 ◦ 3
SETB	P2 ◦ 2
SETB	P2 ◦ 1
CLR	P2 ◦ 0

```

    ACALL    DELAYB
CLR        P2 ◦ 3
    SETB     P2 ◦ 2
    SETB     P2 ◦ 1
    SETB     P2 ◦ 0

```

```

    ACALL    DELAYB
    SETB     P2 ◦ 3
    CLR      P2 ◦ 2
SETB       P2 ◦ 1
    SETB     P2 ◦ 0

```

```

    ACALL    DELAYB
    SETB     P2 ◦ 3
    SETB     P2 ◦ 2
CLR        P2 ◦ 1
    SETB     P2 ◦ 0

```

```

    ACALL    DELAYB
    SETB     P2 ◦ 3
    SETB     P2 ◦ 2
    SETB     P2 ◦ 1
    CLR      P2 ◦ 0

```

```

    ACALL    DELAYB
RET

```

```

;-----
; INIT LCD
;-----

INIT :   MOV     R0 , #12
        MOV     R1 , #30H
        CLR     A
NEXT0 :  MOV     @R1 , A
        INC     R1
        DJNZ    R0 , NEXT0
        MOV     A , #38H
        ACALL   W_INS
        MOV     A , #38H
        ACALL   W_INS
        MOV     A , #38H
        ACALL   W_INS
        MOV     A , #38H
        ACALL   W_INS
        MOV     A , #08H ;FUNCTION SET
        ACALL   W_INS
        MOV     A , #01H ;CLEAR DISPLAY
        ACALL   W_INS
        MOV     A , #06H ;INPUT MODE SET
        ACALL   W_INS
        MOV     A , #0CH ;DISPLAY ON/OFF
        ACALL   W_INS
        RET

;-----
; CLEAR LCD
;-----

CLRLCD : MOV     A , #01H

```

```

    ACALL    W_INS
    RET

```

```

;-----

```

```

; WRITE STRING TO LCD

```

```

;-----

```

```

W_STR :   PUSH    A
          MOV     R5 , #0
STR :     MOV     A , R5
          MOVC    A , @A+DPTR
          ACALL   W_DATA
          INC     R5
          DJNZ    CNT , STR
          POP     A
          RET

```

```

;-----

```

```

; WRITE INS TO LCD

```

```

;-----

```

```

W_INS :   PUSH    A
          ACALL   CHECK
          CLR     RS
          CLR     RW
          SETB    EN
          MOV     P0 , A
          CLR     EN
          POP     A
          RET

```

```

;-----

```

```

; WRITE DATA TO LCD

```

```

;-----
W_DATA :  PUSH    A
          ACALL   CHECK
          SETB    RS
          CLR     RW
          SETB    EN
          MOV     P0 , A
          CLR     EN
          POP     A
          RET

;-----
; CHECK LC D BUSY
;-----
CHECK :   PUSH    A
BUSY :    CLR     RS
          SETB    RW
          SETB    EN
          MOV     A , P0
          CLR     EN
          JB      ACC.7 , BUSY
          ACALL   DELAY
          POP     A
          RET

;-----
; DELAY
;-----
DELAY :   MOV     R6 , #20
D1 :      MOV     R7 , #150
          DJNZ    R7 , $
          DJNZ    R6 , D1

```

```

                RET

;  =====
;  ==延時副程式=====
;  =====

DELAY1S:  MOV  R5 , #15
DL0:  MOV R6 , #250
DL1:  MOV R7 , #200
DL2:  DJNZ R7 , DL2
      DJNZ R6 , DL1
      DJNZ R5 , DL0
      RET

;=====
;=====延時 10mS=====
;=====

DELAYB:  MOV R6 , #25
DLB0:
      MOV R7 , #200
      DJNZ R7 , $
      DJNZ R6 , DLB0
      RET

;----- --
; MESSAGE
;----- --

MSG1:  DB      'F1 : 1 2 3 4 5 6 '
MSG2:  DB      'F2 : 1 2 3 4 5 6 '

                END

                EN      REG      P1 ◦ 2

RW      REG      P1 ◦ 1

```

RS REG P1。0

;腳位說明：P2。6 電梯一樓感測，P2。7 電梯二樓感測，P2。5 紅外線，P2。4 轉盤中間感測，

;，P2。0~P2。3 步進馬達，P1。6~P1。7 直流馬達，P1。5 電梯到底感測

CNT EQU 5FH

;

ORG 00H ;設定程式開頭位置

AJMP START

;-----

; MAIN

;-----

START :

ACALL INIT ;呼叫 LCD 初始化副程式

MOV 4AH, #0 ;4AH=0 以下程式會用到

MOV 4BH, #0 ;4BH=0 以下程式會用到

CALL CENTER ;轉盤轉至中間位置

ACALL ON ;顯示功能表副程式

LOOP :

JNB P2。5, LOOP1 ;偵測出入口之紅外線

CLRP1。3 ;蜂鳴器叫

JB P2。5, \$; 等待車子完成進入

SETB P1。3 ;蜂鳴器停

ACALL DELAY ;延時

ACALL DELAY ;延時

ACALL DELAY ;延時
CALL JUDGE ;尋找空車位

LOOP1 : CALL SCAN ;掃描 12 顆按鍵

JMP LOOP

JUDGE :

MOV A, 30H ;30H=0 時停到第 1 個位子 (30H~3FH 初始值都為 0)
CJNE A, #0, J2 ;不是 0 跳 J2 表示已停車子
MOV 3CH, #0 ;轉盤角度值
MOV 30H, #1 ;1 表已有停車
CALL FINDPARK ;轉盤轉至空車位
MOV A, #85H ;設定顯示在 LCD 第 1 行第 6 字上
ACALL W_INS ;給 LCD 指令
MOV A, #01111111B ;顯示符號：<- (ASCII 碼=01111111)
ACALL W_DATA ;給 LCD 資料
RET

J2 :

MOV A, 31H ;31H=0 時停到第 2 個位子 (30H~3FH 初始值都為 0)
CJNE A, #0, J3 ;不是 0 跳 J3 表示已停車子
MOV 3CH, #25 ;轉盤角度值 60 度
MOV 31H, #1 ;1 表已有停車
CALL FINDPARK ;轉盤轉至空車位
MOV A, #87H ;設定顯示在 LCD 第 1 行第 8 字上 以下依此類推
ACALL W_INS

MOV A, #01111111B

ACALL W_DATA

RET

J3 :

MOV A, 32H ;32H=0 時停到第 3 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, J4 ;不是 0 跳 J4 表示已停車子

MOV 3CH, #50 ;轉盤角度值 120 度

MOV 32H, #1 ;1 表已有停車

CALL FINDPARK ;轉盤轉至空車位

MOV A, #89H ;同上段

ACALL W_INS

MOV A, #01111111B

ACALL W_DATA

RET

J4 :

MOV A, 33H ;33H=0 時停到第 4 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, J5 ;不是 0 跳 J5 表示已停車子

MOV 3CH, #75 ;轉盤角度值 180 度

MOV 33H, #1 ;1 表已有停車

CALL FINDPARK ;轉盤轉至空車位

MOV A, #8BH ;同上段

ACALL W_INS

MOV A, #01111111B

ACALL W_DATA

RET

J5 :

MOV A, 34H ;34H=0 時停到第 5 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, J6 ;不是 0 跳 J6 表示已停車子

MOV 3CH, #100 ;轉盤角度值 240 度

MOV 34H, #1 ;1 表已有停車

```

CALL  FINDPARK ;轉盤轉至空車位
      MOV      A , #8DH      ;同上段
      ACALL    W_INS
MOV    A , #01111111B
ACALL  W_DATA
RET

```

J6 :

```

MOV    A , 35H      ;35H=0 時停到第 6 個位子 (30H~3FH 初始值都為 0)
CJNE   A , #0 , JJ1 ;不是 0 跳 JJ1 表示已停車子
MOV    3CH , #125    ;轉盤角度值 300 度
MOV    35H , #1      ;1 表已有停車
CALL   FINDPARK ;轉盤轉至空車位
      MOV      A , #8FH      ;同上段
      ACALL    W_INS
MOV    A , #01111111B
ACALL  W_DATA
RET

```

JJ1 :

```

MOV    A , 36H      ;36H=0 時停到第 7 個位子 (30H~3FH 初始值都為 0)
CJNE   A , #0 , JJ2 ;不是 0 跳 JJ2 表示已停車子
MOV    3CH , #0      ;轉盤角度值
MOV    36H , #1      ;1 表已有停車
CALL   FINDPARK2 ;轉盤轉至空車位
      MOV      A , #C5H      ;同上段
      ACALL    W_INS
MOV    A , #01111111B
ACALL  W_DATA
RET

```

JJ2 :

```

MOV    A , 37H      ;37H=0 時停到第 8 個位子 (30H~3FH 初始值都為 0)

```

```

CJNE    A, #0, JJ3    ;不是 0 跳 JJ3 表示已停車子
MOV     3CH, #25      ;轉盤角度值 60 度
MOV     37H, #1       ;1 表已有停車
CALL    FINDPARK2 ;轉盤轉至空車位
        MOV     A, #C7H      ;同上段
        ACALL   W_INS
MOV     A, #01111111B
ACALL   W_DATA
RET

```

JJ3 :

```

MOV     A, 38H      ;38H=0 時停到第 9 個位子 (30H~3FH 初始值都為 0)
CJNE    A, #0, JJ4    ;不是 0 跳 JJ4 表示已停車子
MOV     3CH, #50      ;轉盤角度值 120 度
MOV     38H, #1       ;1 表已有停車
CALL    FINDPARK2 ;轉盤轉至空車位
        MOV     A, #C9H      ;同上段
        ACALL   W_INS
MOV     A, #01111111B
ACALL   W_DATA
RET

```

JJ4 :

```

MOV     A, 39H      ;39H=0 時停到第 10 個位子 (30H~3FH 初始值都為 0)
CJNE    A, #0, JJ5    ;不是 0 跳 JJ5 表示已停車子
MOV     3CH, #75      ;轉盤角度值 180 度
MOV     39H, #1       ;1 表已有停車
CALL    FINDPARK2 ;轉盤轉至空車位
        MOV     A, #CBH      ;同上段
        ACALL   W_INS
MOV     A, #01111111B
ACALL   W_DATA

```

RET

JJ5 :

MOV A, 3AH ;3AH=0 時停到第 11 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, JJ6 ;不是 0 跳 JJ6 表示已停車子

MOV 3CH, #100 ;轉盤角度值 240 度

MOV 3AH, #1 ;1 表已有停車

CALL FINDPARK2 ;轉盤轉至空車位

MOV A, #CDH ;同上段

ACALL W_INS

MOV A, #01111111B

ACALL W_DATA

RET

JJ6 :

MOV A, 3BH ;3BH=0 時停到第 12 個位子 (30H~3FH 初始值都為 0)

CJNE A, #0, JOK ;不是 0 跳 JOK 表示已停車子

MOV 3CH, #125 ;轉盤角度值 300 度

MOV 3BH, #1 ;1 表已有停車

CALL FINDPARK2 ;轉盤轉至空車位

MOV A, #CFH ;同上段

ACALL W_INS

MOV A, #01111111B

ACALL W_DATA

JOK : RET

FINDPARK :

MOV A, 3CH ;角度值->A

MOV R2, A ;A->R2

CJNE A, #0, FI1 ;A=0 ? (角度 0 時不用轉) 是 0 程式往下，否則跳 FI1

JMP FIOKK ;跳 FIOK

FII :

CALL LEFT ;轉盤開始轉動，轉一次約 4 度

DJNZ R2 , FII ;假設 R2=75 就轉 75 次的 4 度 =300 度

SETB P2 . 3

SETB P2 . 2

SETB P2 . 1

SETB P2 . 0

FIOKK : CLR P1 . 6

SETB P1 . 7 ;電梯往上

FIOK :

JB P2 . 7 , FIOK2 ;電梯到二樓？

CLR P1 . 6 ;電梯停止

CLR P1 . 7

JNB P2 . 5 , \$;偵測出入口之紅外線

CLR P1 . 3 ;蜂鳴器叫

ACALL DELAY ;延時

ACALL DELAY ;延時

ACALL DELAY ;延時

JB P2 . 5 , \$;等待車子完全通過

SETB P1 . 3 ;蜂鳴器停

ACALL DELAY ;延時

ACALL DELAY ;延時

ACALL DELAY ;延時

CALL CENTER ;轉盤回原點

RET

FIOK2 : JMP FIOK

FINDPARK2 :

```

MOV    A , 3CH
MOV    R2 , A
CJNE   A , #0 , FI12
JMP FIOKKK2

```

FI12 :

```

CALL   LEFT
DJNZ   R2 , FI12
SETB   P2 . 3
        SETB   P2 . 2
        SETB   P2 . 1
        SETB   P2 . 0

```

FIOKKK2 :

```

CLRP1 . 6
SETB   P1 . 7      ;電梯往上

```

FIOKK2 :

```

JB  P2 . 6 , FIOK22  ;電梯到一樓？
CLRP1 . 6      ;電梯停止
CLRP1 . 7
JNB P2 . 5 , $      ;偵測出入口之紅外線
CLRP1 . 3
ACALL  DELAY
ACALL  DELAY
ACALL  DELAY
JB  P2 . 5 , $      ;等待車子完全通過
SETB   P1 . 3
ACALL  DELAY
ACALL  DELAY

```

```

        ACALL    DELAY
        CALL     CENTER
        RET
FIOK22 :    JMP FIOKK2
;=====
; SCAN KEY          ;參閱 6-3§SCAN 程式說明
;=====
SCAN :    MOV     R4 , #0          ;清除 R4 , R4 存放掃描按鍵值
          MOV     R7 , #2          ;R7 設定為 2 , 掃描 2 次
          CLR P3 . 0              ;按鍵第一排開始掃描
          SETB    P3 . 1
COL :
          JB      P3 . 7 , S1      ;無按鍵(1)~S1 , 有按鍵(0)往下執行
          JNB     P3 . 7 , $        ;消除彈跳誤動作 , 即按鍵(0)等待放開按鍵
(1)往下執行
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
          SJMP    S7
S1 :    INC      R4
          JB      P3 . 6 , S2
          JNB     P3 . 6 , $
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
          SJMP    S7
S2 :    INC      R4
          JB      P3 . 5 , S3
          JNB     P3 . 5 , $

```

```

        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7

S3 :    INC      R4
        JB       P3 ◦ 4 , S4
        JNB      P3 ◦ 4 , $
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7

S4 :    INC      R4
        JB       P3 ◦ 3 , S5
        JNB      P3 ◦ 3 , $
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7

S5 :    INC      R4
        JB       P3 ◦ 2 , S6
        JNB      P3 ◦ 2 , $
        ACALL    DELAY
        ACALL    DELAY
        ACALL    DELAY
        SJMP     S7

S6 :    INC      R4
        CPL P3 ◦ 0
        CPL P3 ◦ 1

```



```
DJNZ    R7 , COL
RET
```

```
S7 :    CJNE    R4 , #0 , S8
        MOV     30H , #0      ;0 表無停車
        MOV     3CH , #0      ;轉盤角度值
        CALL    FINDPARK      ;轉盤轉至空車位
        MOV     A , #85H
        ACALL   W_INS
        MOV     A , #' '
        ACALL   W_DATA
        RET
```

```
S8 :    CJNE    R4 , #1 , S9
        MOV     31H , #0      ;0 表無停車
        MOV     3CH , #15      ;轉盤角度值
        CALL    FINDPARK      ;轉盤轉至空車位
        MOV     A , #87H
        ACALL   W_INS
        MOV     A , #' '
        ACALL   W_DATA
        RET
```

```
S9 :    CJNE    R4 , #2 , SA
        MOV     32H , #0      ;0 表無停車
        MOV     3CH , #30      ;轉盤角度值
        CALL    FINDPARK      ;轉盤轉至空車位
        MOV     A , #89H
        ACALL   W_INS
        MOV     A , #' '
        ACALL   W_DATA
```

```

RET
SA:    CJNE    R4, #3, SB
        MOV     33H, #0      ;0 表無停車
        MOV     3CH, #45     ;轉盤角度值
        CALL    FINDPARK    ;轉盤轉至空車位
        MOV     A, #8BH
        ACALL   W_INS
        MOV     A, #' '
        ACALL   W_DATA
        RET
SB:    CJNE    R4, #4, SC
        MOV     34H, #0      ;0 表無停車
        MOV     3CH, #60     ;轉盤角度值
        CALL    FINDPARK    ;轉盤轉至空車位
        MOV     A, #8DH
        ACALL   W_INS
        MOV     A, #' '
        ACALL   W_DATA
        RET
SC:    CJNE    R4, #5, SD
        MOV     35H, #0      ;0 表無停車
        MOV     3CH, #75     ;轉盤角度值
        CALL    FINDPARK    ;轉盤轉至空車位
        MOV     A, #8FH
        ACALL   W_INS
        MOV     A, #' '
        ACALL   W_DATA
        RET
SD:    CJNE    R4, #6, SE
        MOV     36H, #0      ;0 表無停車

```

```

MOV    3CH, #0      ;轉盤角度值
CALL   FINDPARK2 ;轉盤轉至空車位
      MOV    A, #C5H
      ACALL  W_INS
MOV    A, #'
ACALL  W_DATA
RET
SE:    CJNE   R4, #7, SF
      MOV    37H, #0      ;0 表無停車
      MOV    3CH, #15      ;轉盤角度值
      CALL   FINDPARK2 ;轉盤轉至空車位
      MOV    A, #C7H
      ACALL  W_INS
MOV    A, #'
ACALL  W_DATA
RET
SF:    CJNE   R4, #8, SG
      MOV    38H, #0      ;0 表無停車
      MOV    3CH, #30      ;轉盤角度值
      CALL   FINDPARK2 ;轉盤轉至空車位
      MOV    A, #C9H
      ACALL  W_INS
MOV    A, #'
ACALL  W_DATA
RET
SG:    CJNE   R4, #9, SH
      MOV    39H, #0      ;0 表無停車
      MOV    3CH, #45      ;轉盤角度值
      CALL   FINDPARK2 ;轉盤轉至空車位
      MOV    A, #CBH

```

```

        ACALL    W_INS
MOV     A, #' '
        ACALL    W_DATA
        RET
SH:     CJNE     R4, #10, SI
        MOV      3AH, #0      ;0 表無停車
        MOV      3CH, #60      ;轉盤角度值
        CALL     FINDPARK2 ;轉盤轉至空車位
        MOV      A, #CDH
        ACALL    W_INS
        MOV      A, #' '
        ACALL    W_DATA
        RET
SI:     CJNE     R4, #11, SOK
        MOV      3BH, #0      ;0 表無停車
        MOV      3CH, #75      ;轉盤角度值
        CALL     FINDPARK2 ;轉盤轉至空車位
        MOV      A, #CFH
        ACALL    W_INS
        MOV      A, #' '
        ACALL    W_DATA
        RET

SOK:
        RET
;-----
; POWER ON
;-----
ON:     PUSH     A
        MOV      A, #80H

```



```

        CLR      P2.1
SETB    P2.2
        SETB     P2.3
JNB P2.4, CBACK
        ACALL    DELAYB
        SETB     P2.0
        SETB     P2.1
CLR      P2.2
        SETB     P2.3
JNB P2.4, CBACK
        ACALL    DELAYB
        SETB     P2.0
        SETB     P2.1
        SETB     P2.2
        CLR      P2.3
JNB P2.4, CBACK
        ACALL    DELAYB
JMP CENTER
CBACK:   MOV      4BH, #1
        SETB     P2.0
        SETB     P2.1
        SETB     P2.2
        SETB     P2.3

```

MOV A, 4AH ;4AH、4BH 要皆為 1，表示電梯到底且轉盤在中間才能返回

```

ANL     A, 4BH
CJNE    A, #1, CB2
MOV      4AH, #0
MOV      4BH, #0

```

RET

CB2 :

JMP CENTER

LEFT :

CLR P2 ◦ 3

SETB P2 ◦ 2

SETB P2 ◦ 1

SETB P2 ◦ 0

ACALL DELAYB

SETB P2 ◦ 3

CLR P2 ◦ 2

SETB P2 ◦ 1

SETB P2 ◦ 0

ACALL DELAYB

SETB P2 ◦ 3

SETB P2 ◦ 2

CLR P2 ◦ 1

SETB P2 ◦ 0

ACALL DELAYB

SETB P2 ◦ 3

SETB P2 ◦ 2

SETB P2 ◦ 1

CLR P2 ◦ 0

ACALL	DELAYB
CLR	P2 ◦ 3
SETB	P2 ◦ 2
SETB	P2 ◦ 1
SETB	P2 ◦ 0

ACALL	DELAYB
SETB	P2 ◦ 3
CLR	P2 ◦ 2
SETB	P2 ◦ 1
SETB	P2 ◦ 0

ACALL	DELAYB
SETB	P2 ◦ 3
SETB	P2 ◦ 2
CLR	P2 ◦ 1
SETB	P2 ◦ 0

ACALL	DELAYB
SETB	P2 ◦ 3
SETB	P2 ◦ 2
SETB	P2 ◦ 1
CLR	P2 ◦ 0

ACALL	DELAYB
CLR	P2 ◦ 3
SETB	P2 ◦ 2
SETB	P2 ◦ 1
SETB	P2 ◦ 0


```

        ACALL    DELAYB
        SETB     P2 ◦ 3
        CLR      P2 ◦ 2
SETB     P2 ◦ 1
        SETB     P2 ◦ 0

```

```

        ACALL    DELAYB
        SETB     P2 ◦ 3
        SETB     P2 ◦ 2
CLR      P2 ◦ 1
        SETB     P2 ◦ 0

```

```

        ACALL    DELAYB
        SETB     P2 ◦ 3
        SETB     P2 ◦ 2
        SETB     P2 ◦ 1
        CLR      P2 ◦ 0

```

```

        ACALL    DELAYB
RET

```

```

;-----

```

```

; INIT  LCD

```

```

;-----

```

```

INIT :   MOV     R0 , #12
         MOV     R1 , #30H
         CLR     A

```

```

NEXT0 :  MOV      @R1 , A
          INC      R1
          DJNZ     R0 , NEXT0
          MOV      A , #38H
          ACALL    W_INS
          MOV      A , #38H
          ACALL    W_INS
          MOV      A , #38H
          ACALL    W_INS
          MOV      A , #38H
          ACALL    W_INS
          MOV      A , #08H ;FUNCTION SET
          ACALL    W_INS
          MOV      A , #01H ;CLEAR DISPLAY
          ACALL    W_INS
          MOV      A , #06H ;INPUT MODE SET
          ACALL    W_INS
          MOV      A , #0CH ;DISPLAY ON/OFF
          ACALL    W_INS
          RET

```

```

;-----

```

```

; CLEAR LCD

```

```

;-----

```

```

CLRLCD : MOV      A , #01H
          ACALL    W_INS
          RET

```

```

;-----

```

```

; WRITE STRING TO LCD

```

```

;-----
W_STR :   PUSH    A
          MOV      R5 , #0
STR :     MOV      A , R5
          MOVC     A , @A+DPTR
          ACALL    W_DATA
          INC      R5
          DJNZ     CNT , STR
          POP      A
          RET

```

```

;-----
; WRITE INS TO LCD

```

```

;-----
W_INS :   PUSH    A
          ACALL    CHECK
          CLR      RS
          CLR      RW
          SETB     EN
          MOV      P0 , A
          CLR      EN
          POP      A
          RET

```

```

;-----
; WRITE DATA TO LCD

```

```

;-----
W_DATA :  PUSH    A
          ACALL    CHECK
          SETB     RS
          CLR      RW
          SETB     EN

```

```

        MOV     P0 , A
        CLR     EN
        POP     A
        RET

;-----
; CHECK LC D BUSY
;-----
CHECK :   PUSH     A
BUSY :    CLR      RS
        SETB     RW
        SETB     EN
        MOV      A , P0
        CLR      EN
        JB       ACC.7 , BUSY
        ACALL    DELAY
        POP      A
        RET

;-----
; DELAY
;-----
DELAY :   MOV      R6 , #20
D1 :      MOV      R7 , #150
        DJNZ     R7 , $
        DJNZ     R6 , D1
        RET

;  =====
;  ==延時副程式==
;  =====
DELAY1S : MOV     R5 , #15
DL0 :     MOV     R6 , #250

```

DL1 : MOV R7 , #200

DL2 : DJNZ R7 , DL2

 DJNZ R6 , DL1

 DJNZ R5 , DL0

 RET

;=====

;=====延時 10mS=====

;=====

DELAYB : MOV R6 , #25

DLB0 :

 MOV R7 , #200

 DJNZ R7 , \$

 DJNZ R6 , DLB0

 RET

;----- - -

; MESSAGE

;----- - -

MSG1 : DB 'F1 : 1 2 3 4 5 6 '

MSG2 : DB 'F2 : 1 2 3 4 5 6 '

END