



逢甲大學學生報告 ePaper

LED 創意之電子數位時鐘 設計與製作

作者：葉俊廷、張舜博、黃億欣、王筱薇、林采潔

系級：電機二甲

學號：D9736276、D9776502、D9776529、D9776559、D9776593

開課老師：何子儀 老師

課程名稱：微處理機系統實習

開課系所：電機系

開課學年：98 學年度 第 2 學期



摘要

8051為一個小型的微控制器，常被用來當作小型專題的控制器。時鐘為一般家庭中必備物品之一，為了使專題更生活化，於是本專題利用8051單晶片控制設計一個電子時鐘，並動手實作製成實品，其主要功能為時間顯示與設定。

在設計軟體中，運用計時器(Timer)與中斷(Interrupt)的概念，挑選合適的計時器模式去實行，使CPU在穩定的狀態下運作。當有中斷發生時，CPU跳至其相對的向量位址去執行，直至副程式結束回到主程式，CPU繼續穩定運作；在硬體方面，用了七段顯示器跟LED燈來顯示，由於有很多的LED燈，所以用解碼器(3-8 decoder)來協助控制，使8051不會因為接腳不足而無法控制LED燈的亮暗，而為了避免LED燒壞，使用電阻來限制電流。

關鍵字：微控制器、電子時鐘、LED

目次

摘要	i
目次	ii
第一章 製作原理	1
第二章 系統架構	8
第三章 系統功能	34
第四章 實驗結果與操作說明	35
第五章 結論與討論	39
第六章 隊員工作劃分	40
第七章 工作日誌	41
第八章 心得	44
附錄 C	47
附錄 D	48
參考文獻	54

第一章製作原理

1.1 計時/計數器

1.1.1 工作模式與計時/計數選擇

1. 由TMOD暫存器(表1.1)設定，TMOD說明下表1.2，其中低的4位元為計時/計數器0，高的4位元為計時/計數器1。

表1.1 TMOD 暫存器

TMOD:	GATE	C/ \bar{T}	M1	M0	GATE	C/ \bar{T}	M1	M0
-------	------	--------------	----	----	------	--------------	----	----

表1.2 TMOD 說明

符號	說明
GATE	GATE=1，當TCON的TRX指令被設為1時，且接腳 \overline{INTX} 為高電位，計時器/計數器才會動作 GATE=0，當TCON的TRX指令被設為1時，計時器/計數器就會動作
C/ \bar{T}	C/ \bar{T} =1，計數器模式 C/ \bar{T} =0，計時器模式
M1、 M2	選擇工作模式位元，模式設定如表1.2(a)

表1.2(a) 模式設定

M1	M2	模式
0	0	0
0	1	1
1	0	2
1	1	3

TCON暫存器說明，如表1.4

表1.3 TCON 暫存器

TCON:	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
--------------	------------	------------	------------	------------	------------	------------	------------	------------

表1.4 TCON 暫存器說明

符號	位址	說明
TF1	TCON.7	計時/計數器1的溢位旗標
TR1	TCON.6	計時/計數器1的啟動控制位元，TR1=1，工作；TR1=0，停止
TF0	TCON.5	計時/計數器0的溢位旗標
TR0	TCON.4	計時/計數器0的啟動控制位元，TR0=1，工作；TR0=0，停止
IE1	TCON.3	外部中斷1的負緣旗標
IT1	TCON.2	外部中斷1的觸發型式控制位元
IE0	TCON.1	外部中斷0的負緣旗標
IT0	TCON.0	外部中斷0的觸發型式控制位元

2.模式0(Mode 0)分析：如圖1.1

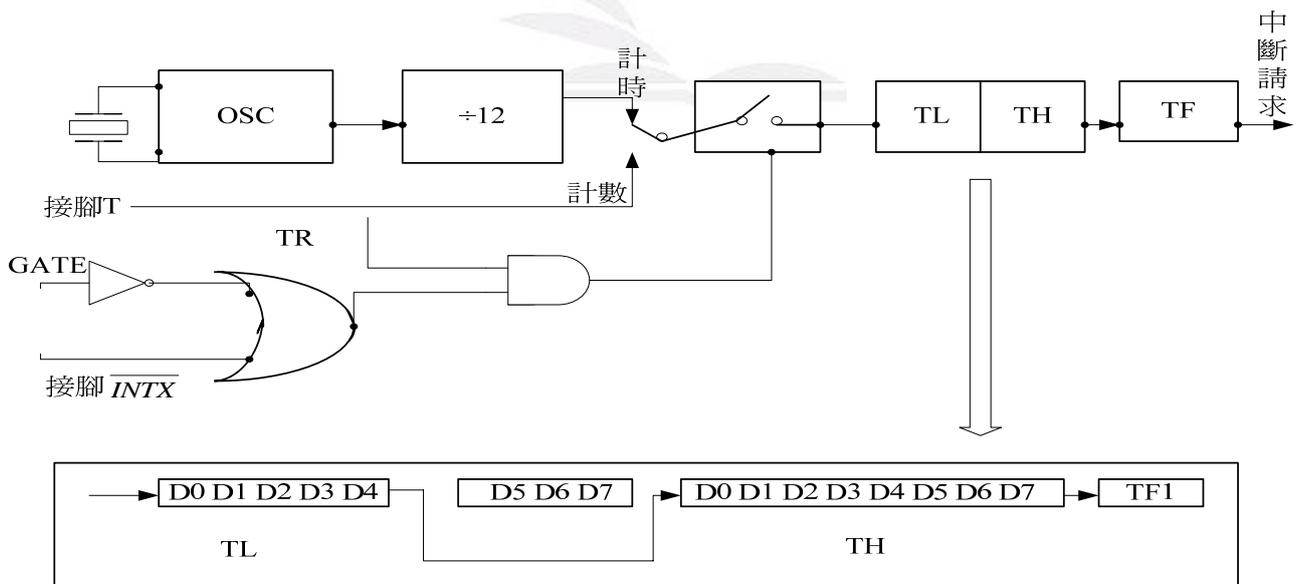


圖1.1 模式0

在模式0時，TL和TH組成13位元之向上計數器，其初值可以用MOV指令設定，當13個位元都為1時，再輸入一個脈波而使13個位元都為0，則此時會令計時/計數溢位旗標TF=1。

3.模式1(Mode 1)分析：如圖1.2

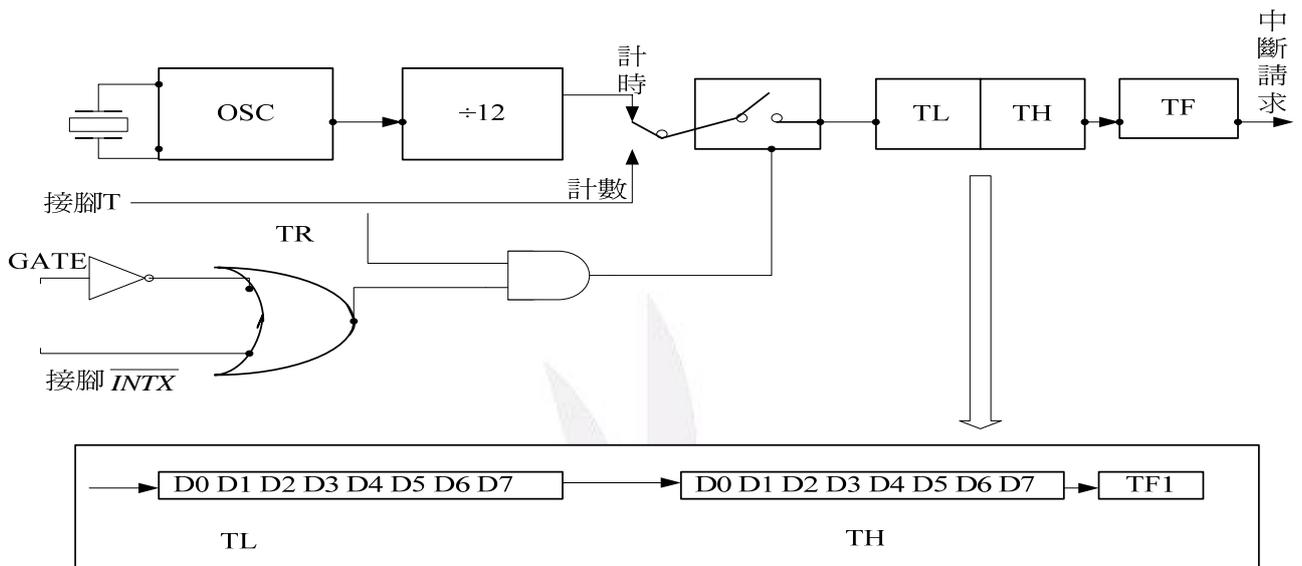


圖1.2 模式1

在模式1時，TL和TH組成16位元之向上計數器，其初值可以用MOV指令設定，當16個位元都為1時，再輸入一個脈波而使16個位元都為0，則此時會令計時/計數溢位旗標TF=1。

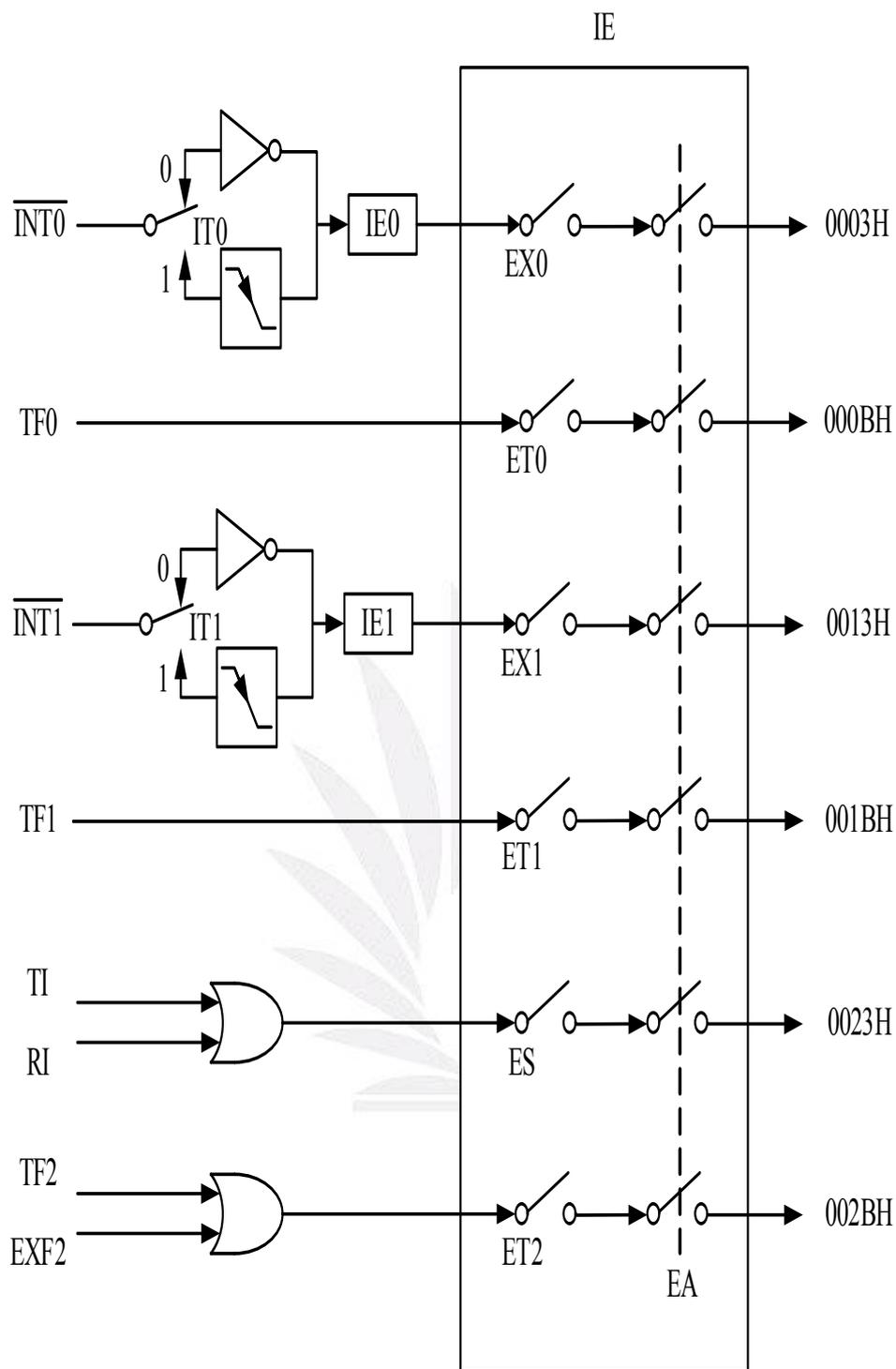


圖1.4 中斷來源及中斷相對副程式位址

2. 中斷致能暫存器IE，如表1.6

表1.5 IE暫存器

IE :	EA	—	ET2	ES	ET1	EX1	ET0	EX0
------	----	---	-----	----	-----	-----	-----	-----

表1.6 IE 暫存器說明

符號	位址	說明
EA	IE.7	EA=0，所有中斷除能 EA=1，由個別中斷位元控制是否致能
—	IE.6	
ET2	IE.5	ET2=1，計時/計數器2的中斷致能 ET2=0，計時/計數器2的中斷除能只有8052有
ES	IE.4	ES=1，串埠列中斷致能 ES=0，串埠列中斷除能
ET1	IE.3	ET1=1，計時/計數器1的中斷致能 ET1=0，計時/計數器1的中斷除能
EX1	IE.2	EX1=1，外部中斷1(接腳 $\overline{INT1}$)致能 EX1=0，外部中斷1(接腳 $\overline{INT1}$)除能
ET0	IE.1	ET0=1，計時/計數器0的中斷致能 ET0=0，計時/計數器0的中斷除能
EX0	IE.0	EX0=1，外部中斷0(接腳 $\overline{INT0}$)致能 EX0=0，外部中斷0(接腳 $\overline{INT0}$)除能

3.外部中斷：外部中斷接腳 $\overline{INT0}$ 及 $\overline{INT1}$ 可用計時/計數器控制暫存器 TCON 內的 IT0 與 IT1 位元規劃為觸發。IT0=1，則為負緣觸發型中斷，當 $\overline{INT0}$ 由高電位變低電位，會令 IE0 自動保持 1，直到 CPU 跳去執行相對應的中斷副程式才會自動將 IE0 清除為 0。IT0=0，為低準位動作， $\overline{INT0}$ 的低電位須維持至 CPU 去執行相對應的中斷副程式為止，否則中斷請求就會消失。

4.計時/計數器中斷：當計時/計數器產生溢位，溢位旗標 TF=1，直至 CPU 跳去執行相對應的中斷副程式，才會自動將 TF 清除為 0。

5.所有可以產生中斷請求的旗標位元，都可以用指令加以偵測、設定或清除。

第二章系統架構

2.1 硬體電路設計與製作

本專題設計之硬體電路用到元件(1)AT89S51(2)74LS138(3)74LS245(4)七段顯示器(5)LED 燈(6)排阻(7)簡易鍵盤(8)電阻(9)電容(10)按鈕(11)石英震盪器 12MHz (12)A1015GR。

2.1.1 硬體架構圖

圖 2.1 為本專題設計之硬體架構圖。

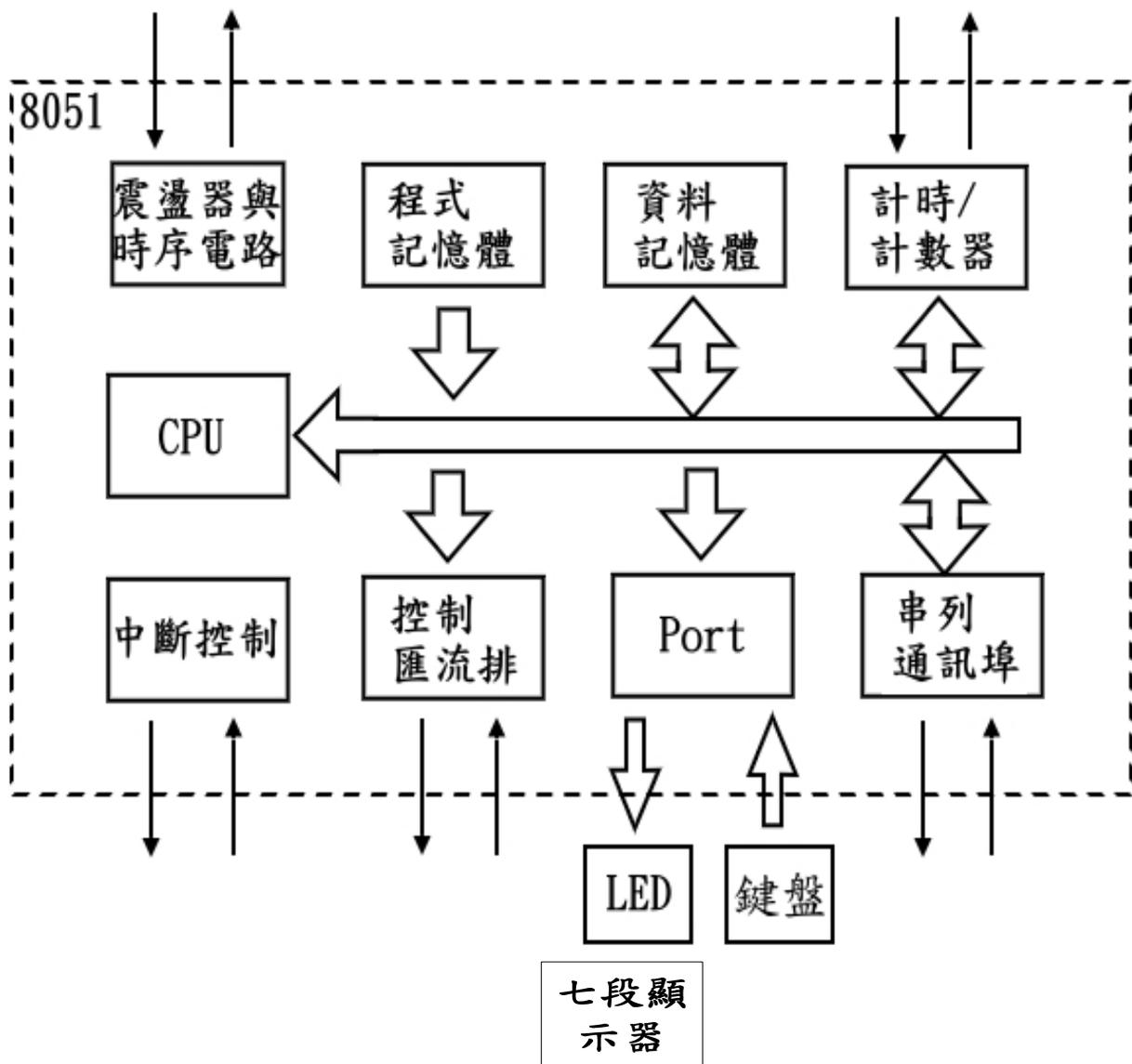


圖 2.1 系統架構圖

2.1.2 AT89S51

1. AT89S51 屬於 MCS-51 系列，而 MCS-51 系列單晶片是 INTEL 公司的產品，其中 8051 是 INTEL 公司於 1981 年生產製造的原始晶片，內部結構如表 2.1。

表 2.1 內部結構

內部結構	內部記憶體				輸入/輸出	計時/計數器
	RAM	ROM	EPROM	Flash Memory	I/O	16位元
89S51	128byte	0	0	4Kbyte	32腳	2個

2. 內部方塊圖(如圖 2.2)，說明如下

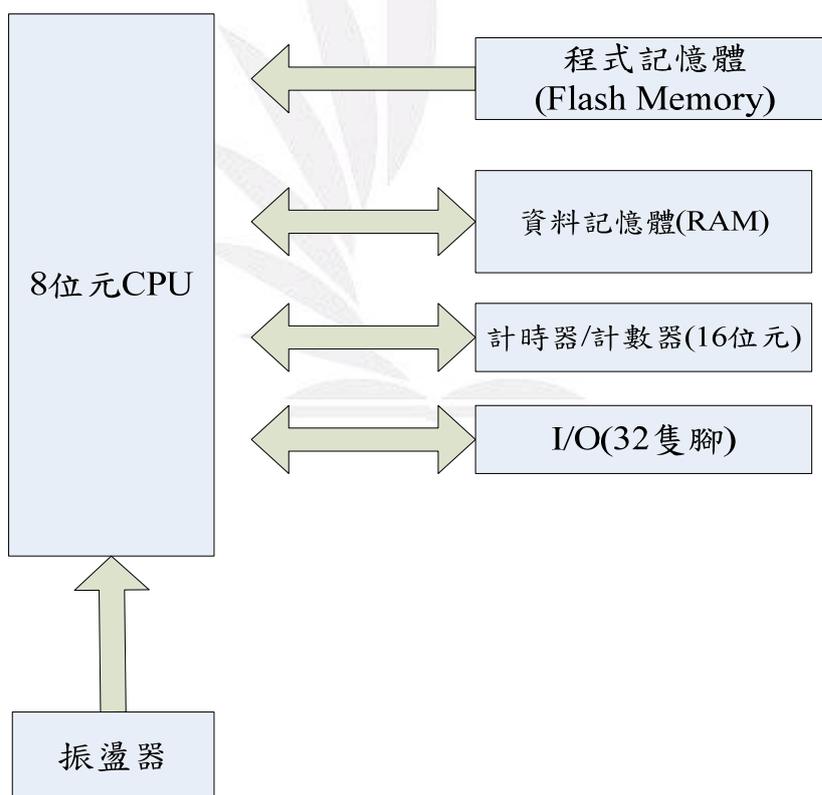


圖 2.2 微電腦控制方塊圖

- A.振盪器：外接一個石英晶體，可生整個系統所需時序脈波。
- B.CPU：用來執行指令、控制整個微電腦的運作。
- C.程式記憶體：ROM(Read-Only Memory)或Flash Memory，用來儲存程式及固定不變的常數。ROM或Flash Memory的特點是不會因電源切斷而消失。
- D.資料記憶體：RAM(Random Access Memory)，用來儲存程式執行中須加以改變的資料，可由CPU存取資料，但電源切斷時，資料消失。
- E.計時器/計數器：用指令設定16位元的計時器或作為16位元的計數器。
- F.I/O接腳：一共有32隻輸入/輸出接腳可被應用。

3.接腳圖(如圖2.3)，共有40隻接腳，說明如下

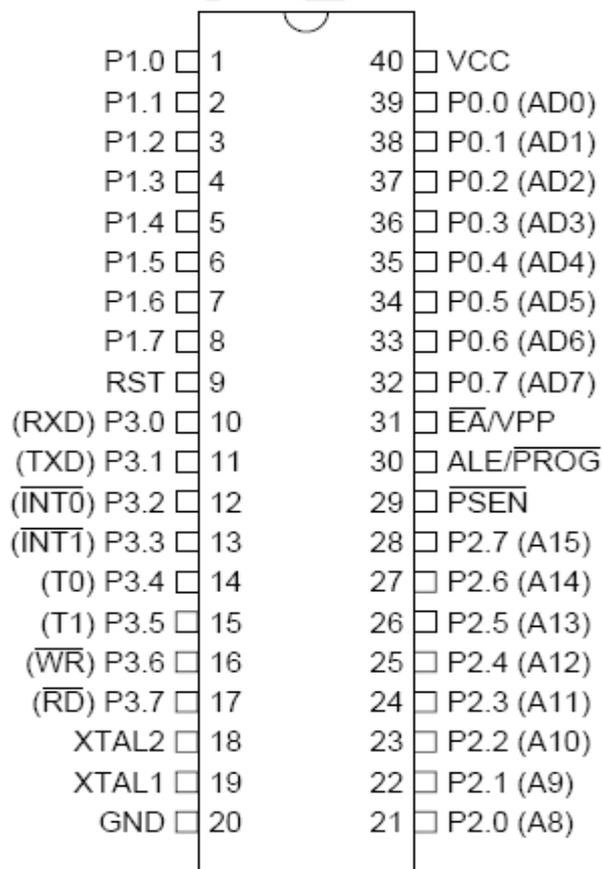


圖2.3 編號之接腳圖

- A.Vss或GND：編號第20隻接腳，電路的接地電位。
- B.Vcc：編號第40隻接腳，電路隻電源接腳，外接+5V。

C.XTAL1及XTAL2：編號第18~19隻接腳，用來外接石英震盪器(如圖2.4)。

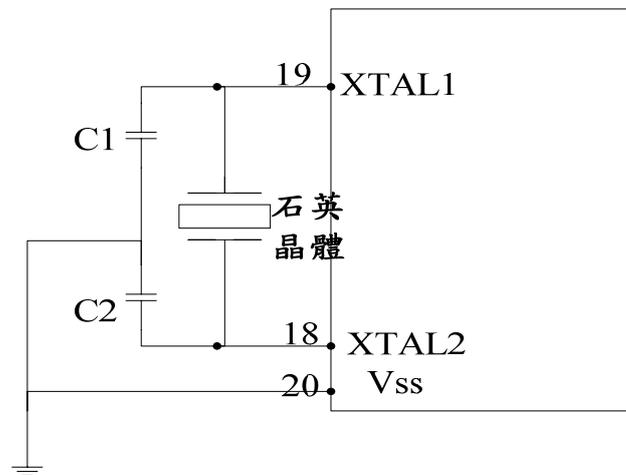


圖2.4 振盪電路

D.RST(REST)：編號第9隻接腳，重置輸入腳，當輸入為High時，則重置；有時會加按鈕，以方便重置設定(如圖2.5)。

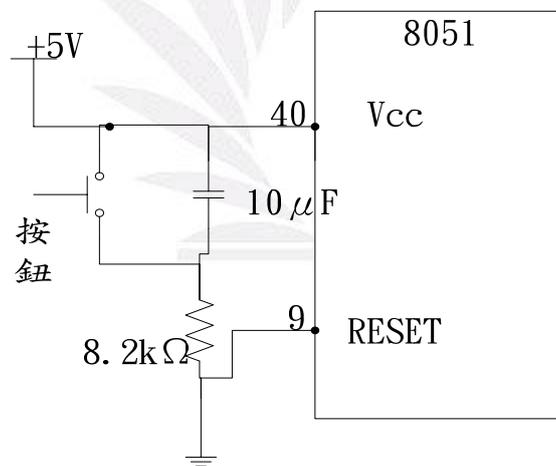


圖2.5 重置電路

E. \overline{EA} ：編號第31隻接腳，屬於輸入腳，在AT89S51必接Vcc，(但在MCS-51系列內，也有接GND的)，若接GND，則CPU被迫讀取外部程式記憶體。

F.P0.0~P0.7：編號第32~39支接腳，可為八位元輸入埠或輸出埠，統稱為Port0，簡稱為P0。可做為位址線或資料線，每隻接腳都屬於開汲極(如圖

2.6)，無內部提升電阻器(pullup resistor)，需要在接腳上外接提升電阻器。當使接腳為輸入腳用時，必須先將1寫入這隻腳。

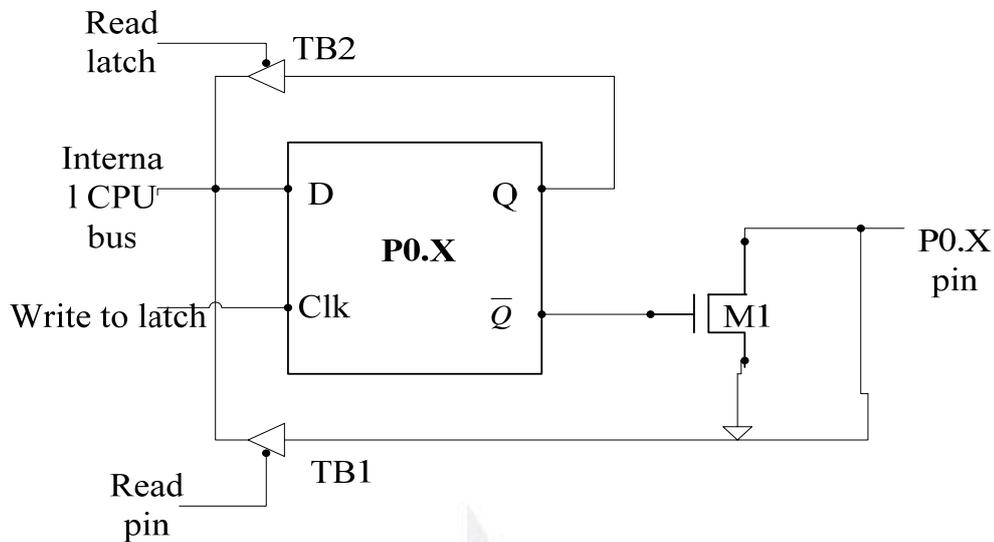


圖2.6 P0結構

G.P1.0~P1.7：編號第1~8隻接腳，可為八位元輸入埠或輸出埠，統稱為Port1，簡稱為P1。有內部提升電阻器。當使接腳為輸入腳用時，必須先將1寫入這隻腳。

H.P2.0~2.7：編號第21~28隻腳，可為八位元輸入埠或輸出埠，統稱為Port2，簡稱為P2。有內部提升電阻器。當使接腳為輸入腳用時，必須先將1寫入這隻腳。可跟Port 0一起做為位址線，用來輸出高位元。

I.P3.0~3.7：編號第10~17隻腳，可為八位元輸入埠或輸出埠，統稱為Port3，簡稱為P3。有內部提升電阻器。當使接腳為輸入腳用時，必須先將1寫入這隻腳。Port 3的接腳還有其他功能(如表2.2)。

表2.2 Port 3 接腳功能

接腳名稱	特殊功能
P3.0	RXD(串列埠輸入腳)
P3.1	TXD(串列埠輸出腳)
P3.2	$\overline{INT0}$ (外部中斷0的輸入腳)
P3.3	$\overline{INT1}$ (外部中斷1的輸入腳)
P3.4	T0(計數器0的輸入腳)
P3.5	T1(計數器1的輸入腳)
P3.6	\overline{WR}
P3.7	\overline{RD}

J.ALE：編號第30隻接腳，位址閃鎖致能(address latch enable)輸出腳。

K. \overline{PSEN} ：編號第29隻接腳，外部程式記憶體致能(program store enable)輸出腳。

4.內部結構

A.指令解碼器與控制單元：程式指令運算碼先從記憶體讀入指令暫存器中，加以解碼分析，再透過控制單元發出信號，使系統內部資料能作適當的傳送與運算。

B.算數邏輯單元(Arithmetic and Logic Unit, ALU)：負責執行算術運算及邏輯運算的部門，使用累積器(Accumulator；簡稱ACC或A)的暫存器。

C.程式計數器(Program Counter；PC)：會自動指出存放於記憶體中下一個等待執行指令的位址，以便CPU讀取，當CPU 每讀取一個指令碼後，PC內容會自動加1，如此CPU 即可依PC 內容值，逐一的提取並執行每一個指令碼。

D.程式記憶體：在AT89S51中，只有Flash Memory作為儲存程式的記憶體，

特點是不會因為電源切斷而使內容消失，可擴充外部記憶體。其中有幾個特殊位址，如表2.3。

表2.3 特殊位址

位址	影響原因	說明
0000H	重置	當重置信號發生後，CPU會自動重0000H開始執行程式，所以主程式第一個指令的位址必須放在0000H
0003H	外部中斷0	當CPU接受外不中斷0時，會自動跳到位址0003H執行中斷副程式
000BH	計時/計數器0中斷	當CPU接受計時/計數器0溢位而產生中斷時，會自動跳到位址000BH執行中斷副程式
0013H	外部中斷1	當CPU接受外不中斷0時，會自動跳到位址0013H執行中斷副程式
001BH	計時/計數器1中斷	當CPU接受計時/計數器1溢位而產生中斷時，會自動跳到位址001BH執行中斷副程式
0023H	串列埠中斷	當串列埠接收或傳送資料完畢時，會產生中斷時，CPU會自動跳到位址0023H執行中斷副程式

E. 資料記憶體：以RAM作為資料記憶體，其內容會隨電源切斷而消失，可外接記憶體。其位址說明如下：

- a. 一般用途暫存器：位址 00H~1FH，共 32 個 byte，每八個 byte 為一個 bank，所以共四個 Bank，Bank0：00H~07H，Bank1：08H~0FH，

Bank2：10H~17H，Bank03：18H~1FH，由 PSW 暫存器來選擇使用哪一個 Bank。

b.可位元定址區：位址在 20H~2FH，共 16 個 Byte，即 128bit，每一個位元可直接使用。

c.資料儲存區：位址 30H~7FH，共 80 個 Byte。

F. 特殊功能暫存器(Special Function Registers；SFR)：位址範圍是位於資料記憶體 80H~FFH 的地方，共 128 Bytes，如下表 2.4。

表 2.4 特殊功能暫存器位址

符號	名稱	位址
*ACC	累積器	0E0H
*B	B 暫存器	0F0H
*PSW	程式狀態字元	0D0H
SP	堆疊指標	81H
DPTR	資料指標(包含 DPL 和 DPH)	83H&82H
DPL	資料指標低位元組	82H
DPH	資料指標高位元組	83H
*P0	埠 0	80H
*P1	埠 1	90H
*P2	埠 2	0A0H
*P3	埠 3	0B0H
*IP	中斷優先次序控制	0B8H
*IE	中斷致能控制	0A8H
TMOD	計時/計數器模式控制	89H
*TCON	計時/計數器控制	88H
符號	名稱	位址

TH0	計時/計數器 0 高位元組	8CH
TL0	計時/計數器 0 低位元組	8AH
TH1	計時/計數器 1 高位元組	8DH
TL1	計時/計數器 1 低位元組	8BH
*SCON	串列埠控制	98H
SBUF	串列資料緩衝器	99H
PCON	電源控制	87H

*表示可位元定址

- a. 累積器 A(Accumulator)：簡稱 ACC 或 A，大部分運算都透過此暫存器，如儲存運算結果、資料傳輸、跳躍判斷等，功能最多
- b. B 暫存器：主要用來除法和乘法運算，乘法運算中用來存放結果的高位元組，除法運算中用來存放餘數，但也可做一般暫存器
- c. 程式狀態字元(Program Status Word；PSW)：(如表 2.5)

表 2.5 PSW 暫存器

CY	AC	F0	RS1	RS0	OV	—	P
----	----	----	-----	-----	----	---	---

- (1) 進位旗標(Carry；CY)：簡寫 C，位址 PSW.7 表示是否借位或進位
- (2) 輔助進位旗標(Auxiliary Carry；AC)：位址 PSW.6，表示第 3 位元是否進位或借位
- (3) 暫存器庫選擇：RS0 和 RS1，位址分別為 PSW.3、PSW.4，用於選擇 Bank(如表 2.6)

表 2.6 Bank 選擇位址

RS1	RS0	Bank	位址
0	0	0	00H~07H
0	1	1	08H~0FH
1	0	2	10H~17H
1	1	3	18H~1FH

(4) 溢位旗標(Overflow ; OV) : 表位元 7 有借位或進位

(5) 同位旗標(Parity ; P) : 當累積器的內容，若等於 1 的位元有奇數個，則 P=1，否則 P=0

d. 堆疊指標(Stack Pointer ; SP) : 當程式執行 CALL 或 PUSH 指令或中斷副程式時，SP 自動加 1，並將資料存在堆疊器，當程式執行 RET 或 RETI 或 POP 時，堆疊器資料被取出，SP 自動減 1

e. 資料指標暫存器(Data Pointer ; DPTR) : 由 DPH 和 DPL 兩個位元組組成，主要用於儲存 16 位元位址，做為存取資料的位址指標

2.1.3 74LS138 : 為一個廣泛應用的解碼器，有三個位址輸入 A、B、C，八個位址輸出 Y0~Y7，用 A、B、C 來控制 Y0~Y7 的輸出，此外還有三個致能輸入，分別為 G1、 $\overline{G2A}$ 、 $\overline{G2B}$ ，G1 為輸入高(High)致能， $\overline{G2A}$ 、 $\overline{G2B}$ 為輸入低(Low)致能。(圖 2.7 和 2.8 分別為其外觀圖與接腳圖，表 2.7 為其真值表)。

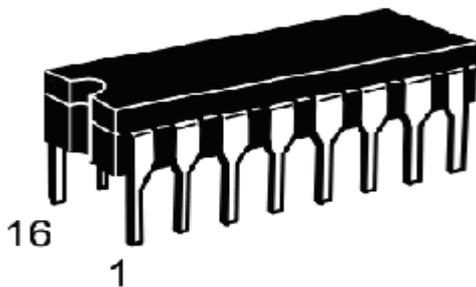


圖 2.7 外觀圖

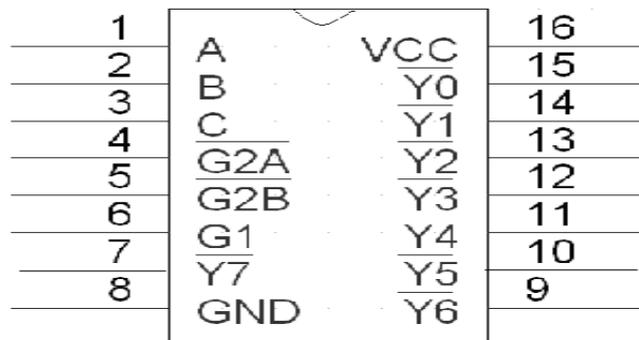


圖 2.8 接腳圖

表2.7 74LS138的真值表

Input						Output							
G1	G2A	G2B	A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

2.1.474LS245 (接腳圖如圖2.9)是一種8位元收發器，用資料線來使非同步資料聯繫，控制功能使外部同步要求降到最低。DIR控制使A資料傳送到B或B資料傳送到A， \overline{G} 為致能輸入，真值表如表2.8。

表2.8 74LS245真值表

\overline{G}	DIR	結果
0	1	B to A
0	1	A to B
1	X	各自獨立

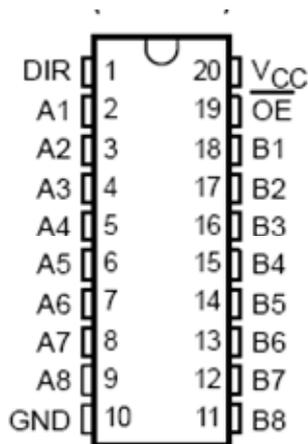


圖2.9 接腳圖

2.1.5 七段顯示器顯示模組有分為「共陽極」與「共陰極」，可依照使用需求不同而選擇其中一個使用。如選擇使用共陽極，則將COM點接至Vcc；若選擇共陰極，則COM點就接至接地。接腳如圖2.10。

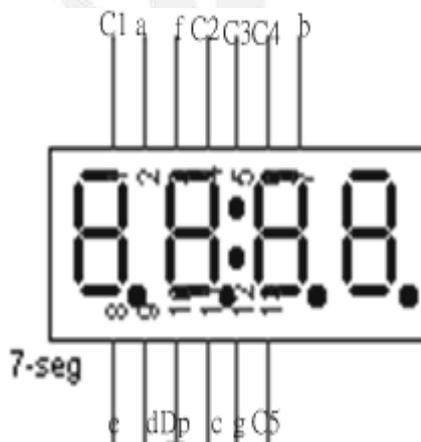


圖2.10 接腳圖

表2.9 字形表(共陽)

字形	Dp	g	f	e	d	c	b	a	字形碼
0	1	1	0	0	0	0	0	0	C0H
1	1	1	1	1	1	0	0	1	F9H
2	1	0	1	0	0	1	0	0	A4H
3	1	0	1	1	0	0	0	0	B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	92H
6	1	0	0	0	0	0	1	0	82H
7	1	1	1	1	1	0	0	0	F8H
8	1	0	0	0	0	0	0	0	80H
9	1	0	0	1	0	0	0	0	90H
A	1	0	0	0	1	0	0	0	88H
B	1	0	0	0	0	0	1	1	83H
C	1	1	0	0	0	1	1	0	C6H
D	1	0	1	0	0	0	0	1	A1H
E	1	0	0	0	0	1	1	0	86H
F	1	0	0	0	1	1	1	0	8EH

2.1.6 LED燈 發光二極體，特性跟二極體相似，長的那一端接正，外觀如圖

2.11，電路符號如圖2.12。

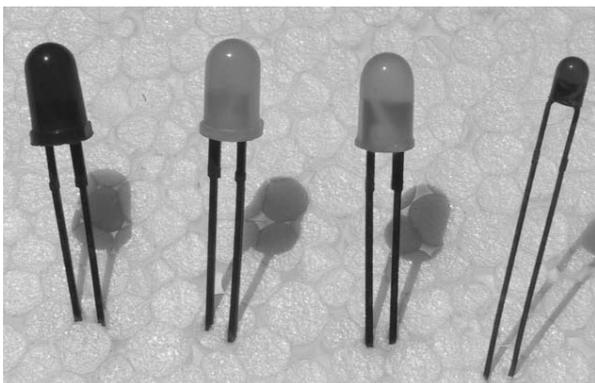


圖2.11 外觀圖

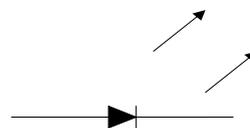


圖2.12 電路符號

2.1.7 排阻 簡單來講就是電阻並聯，有點的地方接Vcc(外型圖如圖2.13)。

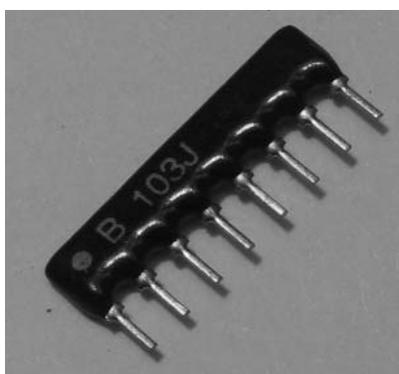


圖2.13 外型圖

2.1.8 簡易鍵盤 其電路內部結構如圖2.14，外觀如圖2.15。

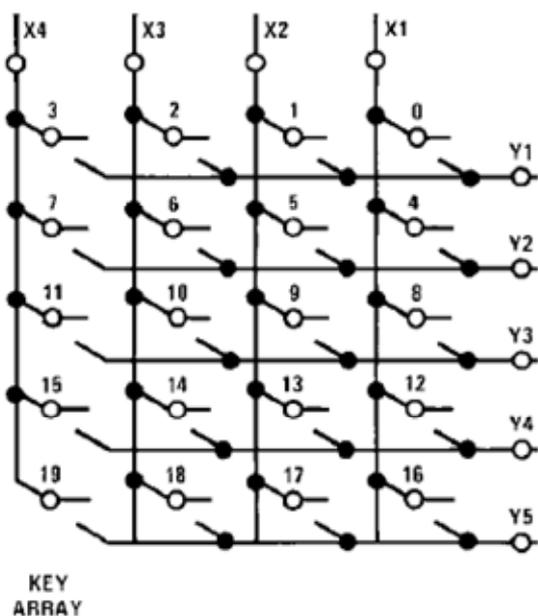


圖2.14 電路內部結構



圖2.15 外觀圖

2.1.9 電阻(Resistor)：

1.固定電阻：

A.直接標示型電阻(水泥電阻)(外觀如圖2.16)。

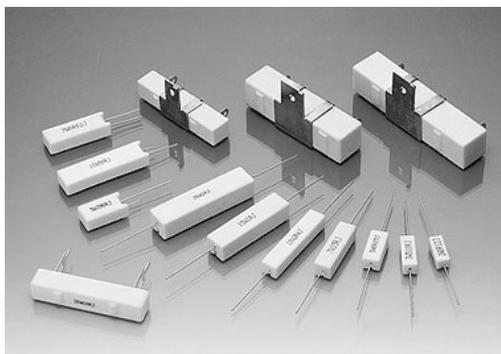


圖2.16 水泥電阻外觀圖

B.電阻色碼的讀取準則(外觀如圖2.17)如表2.10。

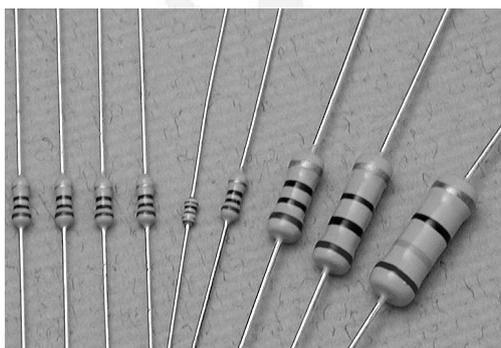


圖2.17 外觀圖

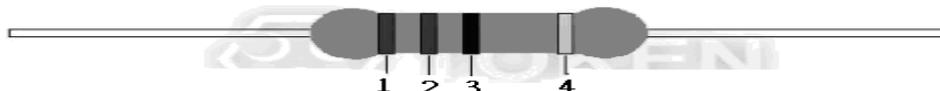


圖 2.18 色帶位置

表2.10 色碼表

顏色	第1色帶	第2色帶	第3色帶	第4色帶
	第1個數字	第2個數字	乘數	容許誤差(%)
黑	0	0	1	
棕	1	1	10	±1

紅	2	2	10^2	± 2
橙	3	3	10^3	
黃	4	4	10^4	
綠	5	5	10^5	± 0.5
藍	6	6	10^6	± 0.25
紫	7	7	10^7	± 0.1
灰	8	8	10^8	± 0.05
白	9	9	10^9	
金			1/10	± 5
銀			1/100	± 10
無				± 20

商用色碼電阻依照其誤差精度規格有其標準值，實驗常用者為額定功率：1/4W、1/2W。阻值誤差： $\pm 5\%$ 。

2. 可變電阻(Variable Resistor ; VR)：又稱電位器(Potentiometer)

A. 電路符號如圖2.19所示，電路中常見可變電阻符號及為典型應用，只使用兩個端子，其一為可調端，另一為固定端。需注意固定端使用時之方向與阻值變化兩者間之搭配。

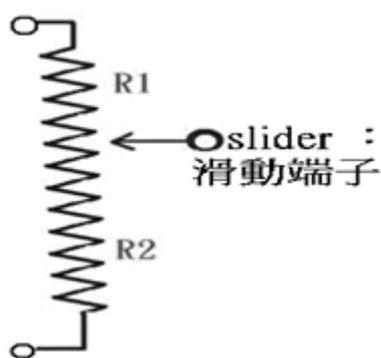


圖2.19 電路符號

B. 可變電阻常用型式：

a. A型：對數形，常用於音量控制。

b.B型：直線型，實驗中最常用，做信號振幅大小的調整。

c.M、N型：常用於左右通道的平衡調整。

2.1.10 電容器(Capacitor)：由兩個金屬導體之間夾絕緣介質而成。

1.電解電容器：(外觀圖如2.20)

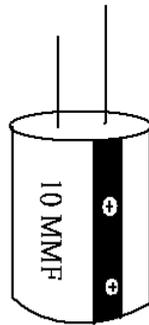


圖2.20 外觀圖

有極性電容，使用時需注意其極性，長腳：+，短腳：-，電容值大且都採直接標示。

2.陶瓷電容器(Ceramic-disk Capacitor)：(外型如圖2.21所示)

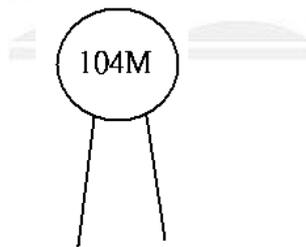


圖2.21 陶瓷電容外型圖

由於大多做成圓版形狀，所以又稱「圓版電容」，為無極性電容，高頻響應較佳，故常當作旁路電容使用，其電容值小(以pF為單位)，採間接標示，圖示電容值讀法：

$$\frac{1}{\alpha} \frac{0}{\beta} \frac{4}{\gamma} \frac{M}{\rho} \Rightarrow C=10 \times 10^4 \text{ pF} \pm 20\% = 0.1 \mu\text{F}$$

表2.11 代碼誤差對照表

代碼	F	G	H	J	K	L	M	N
誤差	±1%	±2%	±3%	±5%	±10%	±15%	±20%	±30%

電容值 $C=(\alpha \times 10 + \beta) \times 10^{\gamma} \pm \rho$ 百分誤差代碼pF。

3. 高值的無極性電解電容：NP電容器(如圖2.22)

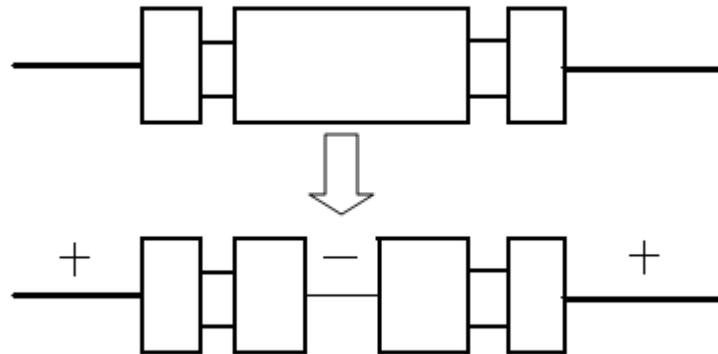


圖2.22 NP電容器

4. 電容的工作電壓(耐壓)

A. 多以均方根值標示：

a. V, WV(Working Voltage)：正常耐壓(連續動作電壓)。

b. TV(Test Voltage)：測試瞬間之耐壓(瞬間峰值電壓)。

B. 一般耐壓設計以20%~50%安全量即可。

C. 提高耐壓能力可藉由電容串聯完成。

2.1.11 按鈕如圖2.23所示

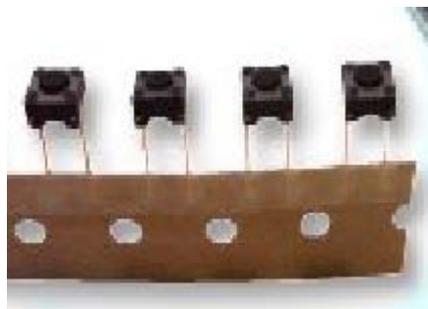


圖2.23 按鈕外觀圖

2.1.12 石英震盪器(12MHz)，在此專題中用來使8051內部有時脈週期

2.1.13 A1015GR 為電晶體 PNP，下圖圖 2.24 為其外型圖，其中 1：base 基極 2：collector 集極 3：Emitter 射極，圖 2.25 為電路符號

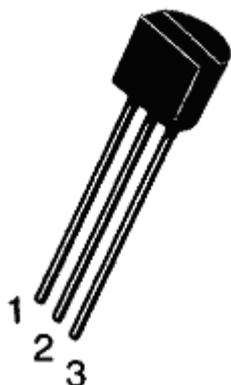


圖 2.24 外型圖

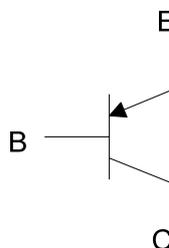


圖 2.25 電路符號

依不同接法有四種模式，如表 2.12。

表 2.12 模式圖

EBJ	BCJ	模式
順偏	逆偏	主動(active)
逆偏	順偏	非主動(inversed)
逆偏	逆偏	不導通 Cut-off
順偏	順偏	飽和(Saturation)

常用公式

$$1. i_c = I_s \times e^{V_{BE}/V_T}$$

$$2. i_B = \frac{i_c}{\beta} = \frac{I_s \times e^{V_{BE}/V_T}}{\beta}$$

$$3. i_E = i_c + i_B = \left(\frac{1 + \beta}{\beta} \right) \times I_s \times e^{V_{BE}/V_T}$$

$$4. \alpha = \frac{\beta}{1 + \beta}$$

2.1.14 電路圖

將電路圖分成幾個部份如下，完整電路請參照附錄 C

主體電路 如圖 2.26 ， Part 1 如圖 2.27 ， Part2 如圖 2.28 ， Part3 如圖 2.29

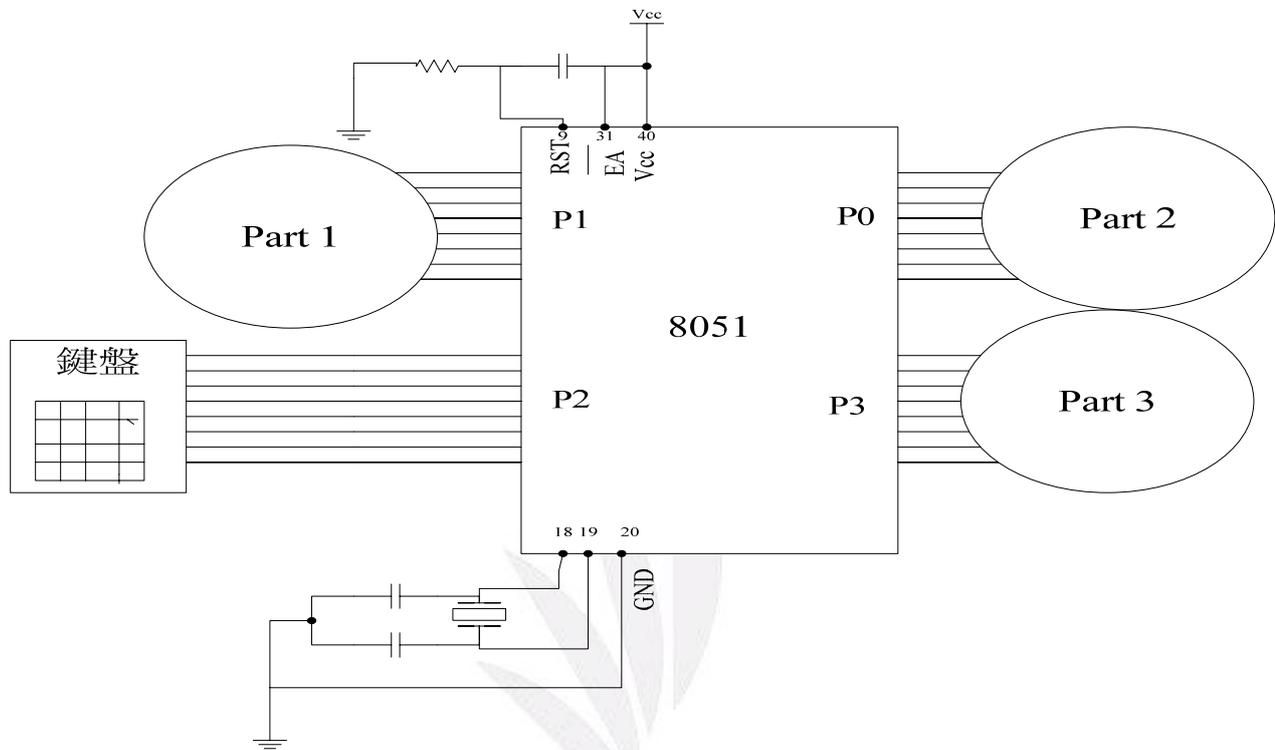
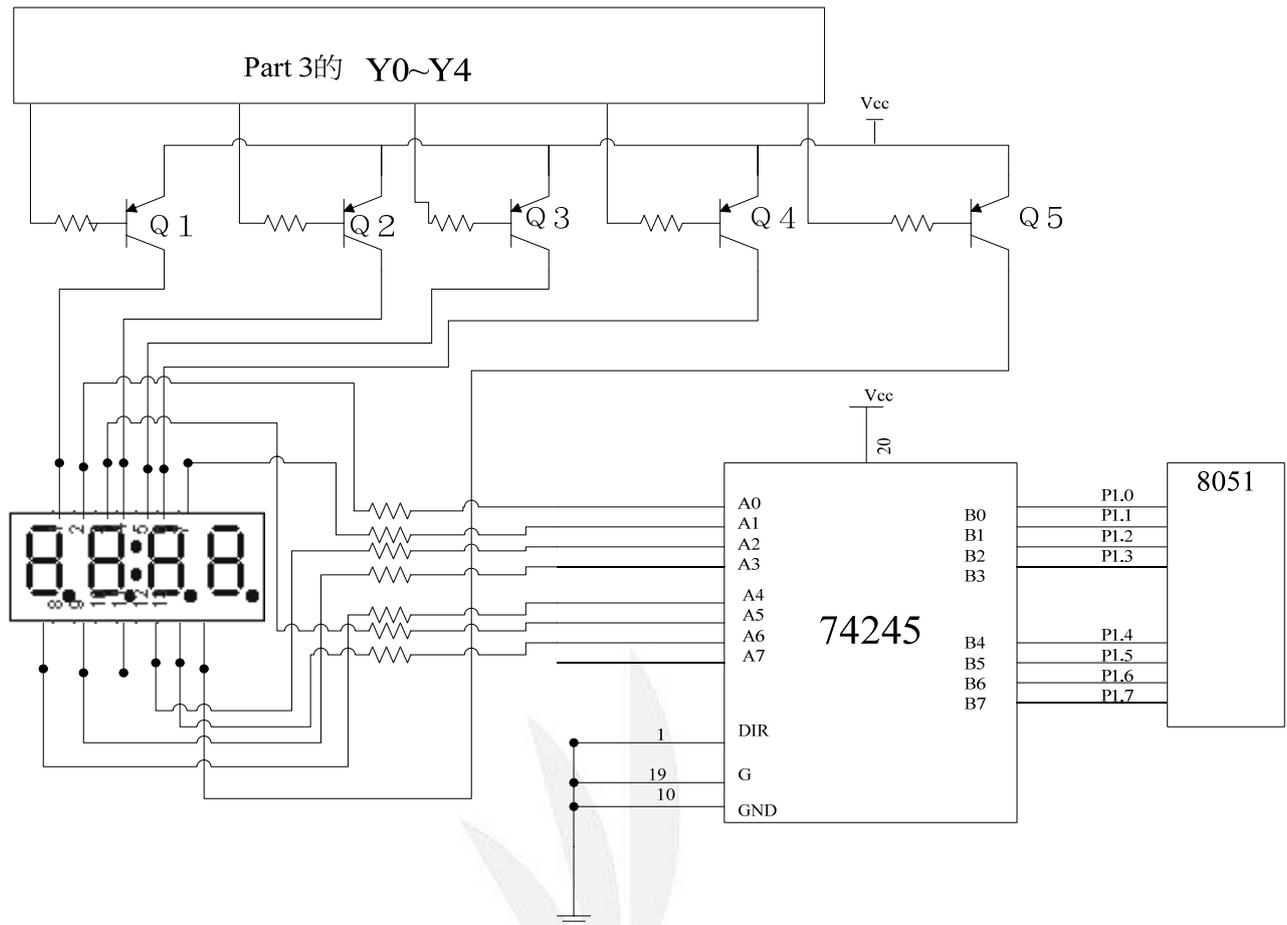


圖 2.26 主體電路



如圖 2.27 Part 1

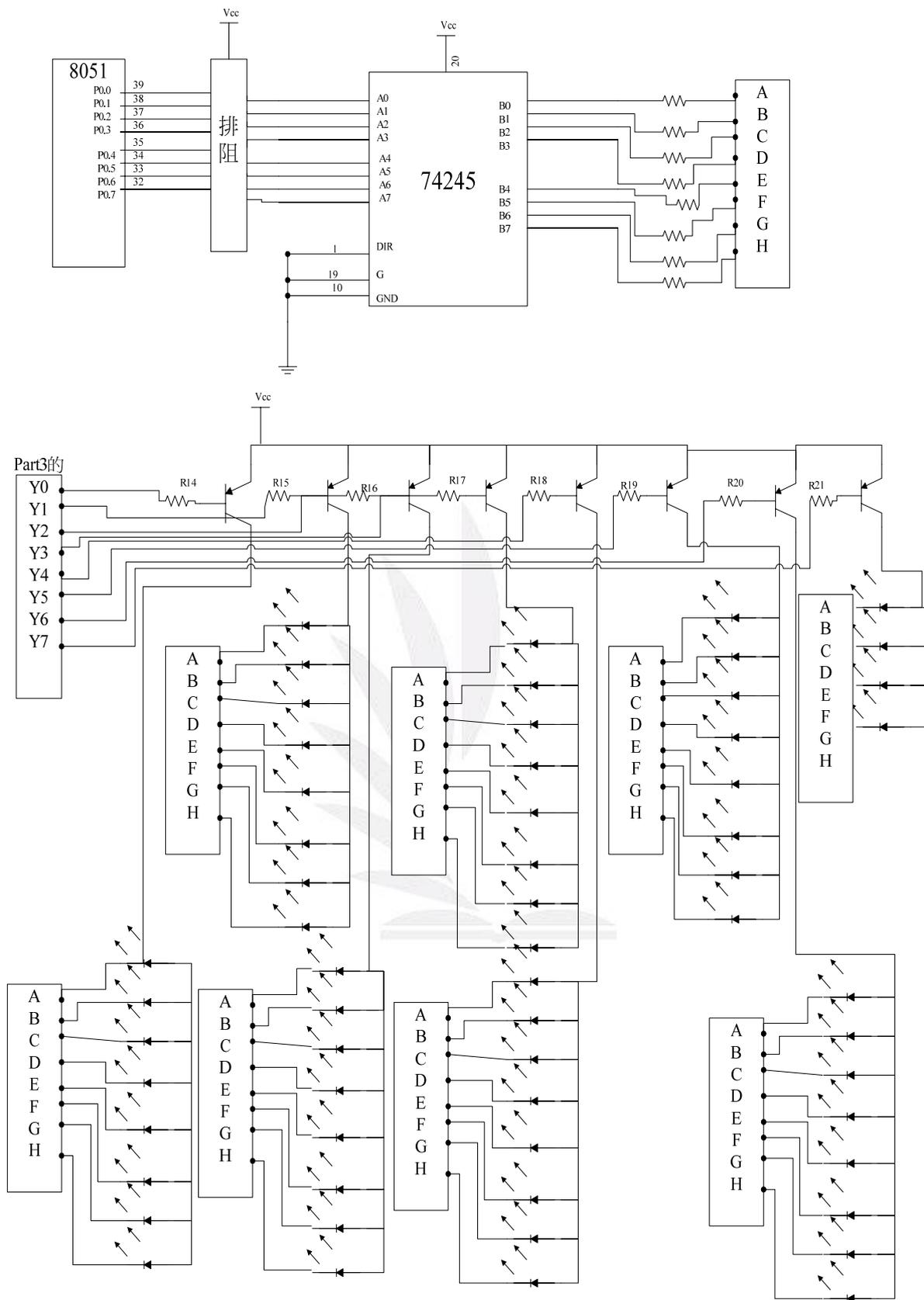


圖 2.28 Part2

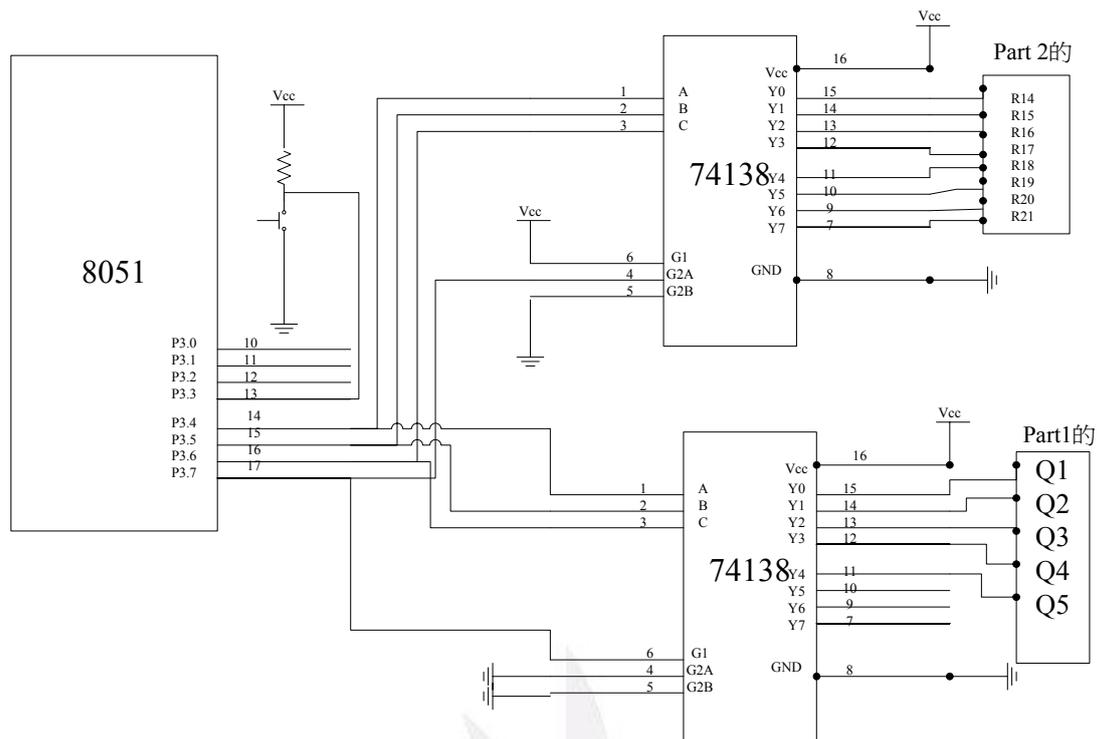


圖 2.29 Part3

2.2 軟體架構(流程圖)

程式碼請參照附錄A，程式碼List請參照附錄B組合語言表請參照附錄D

主程式如圖2.30，計時器中斷副程式如圖2.31，外部中斷副程式如圖2.32。

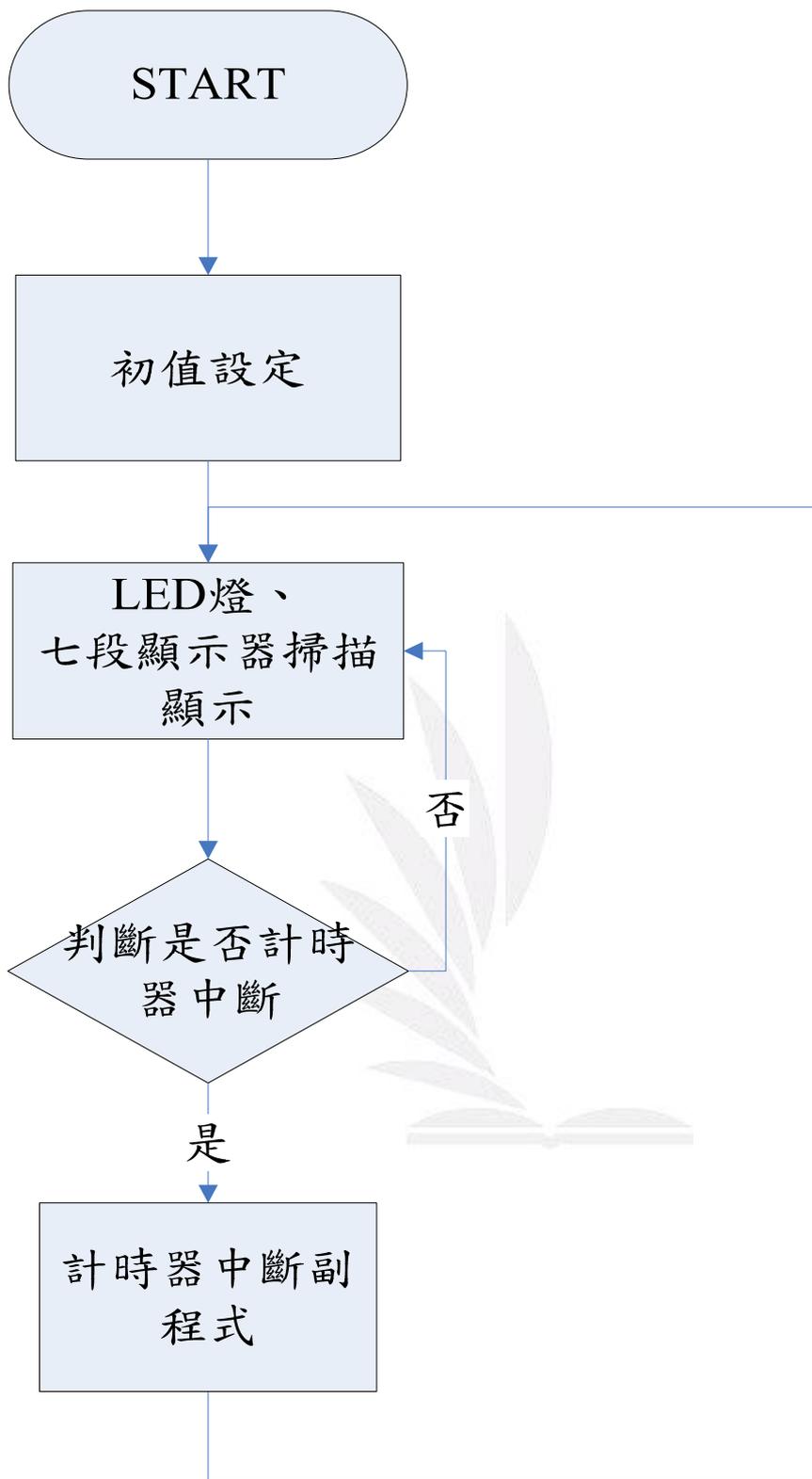


圖2.30 主程式

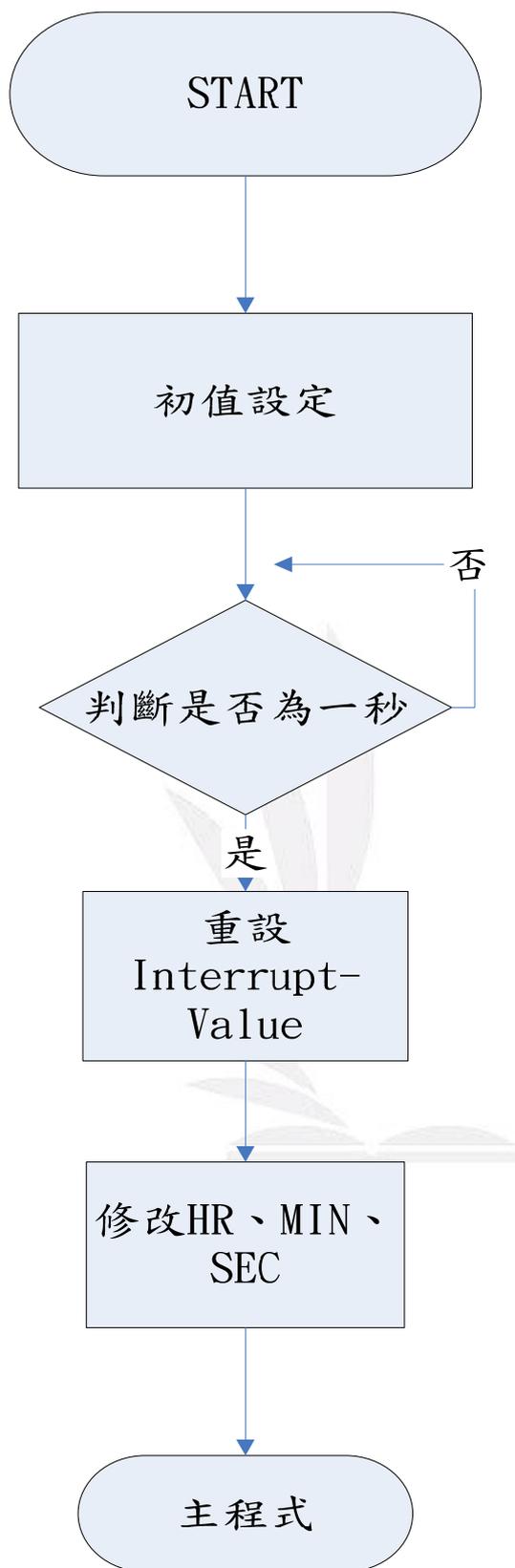


圖2.31 計時器中斷副程式

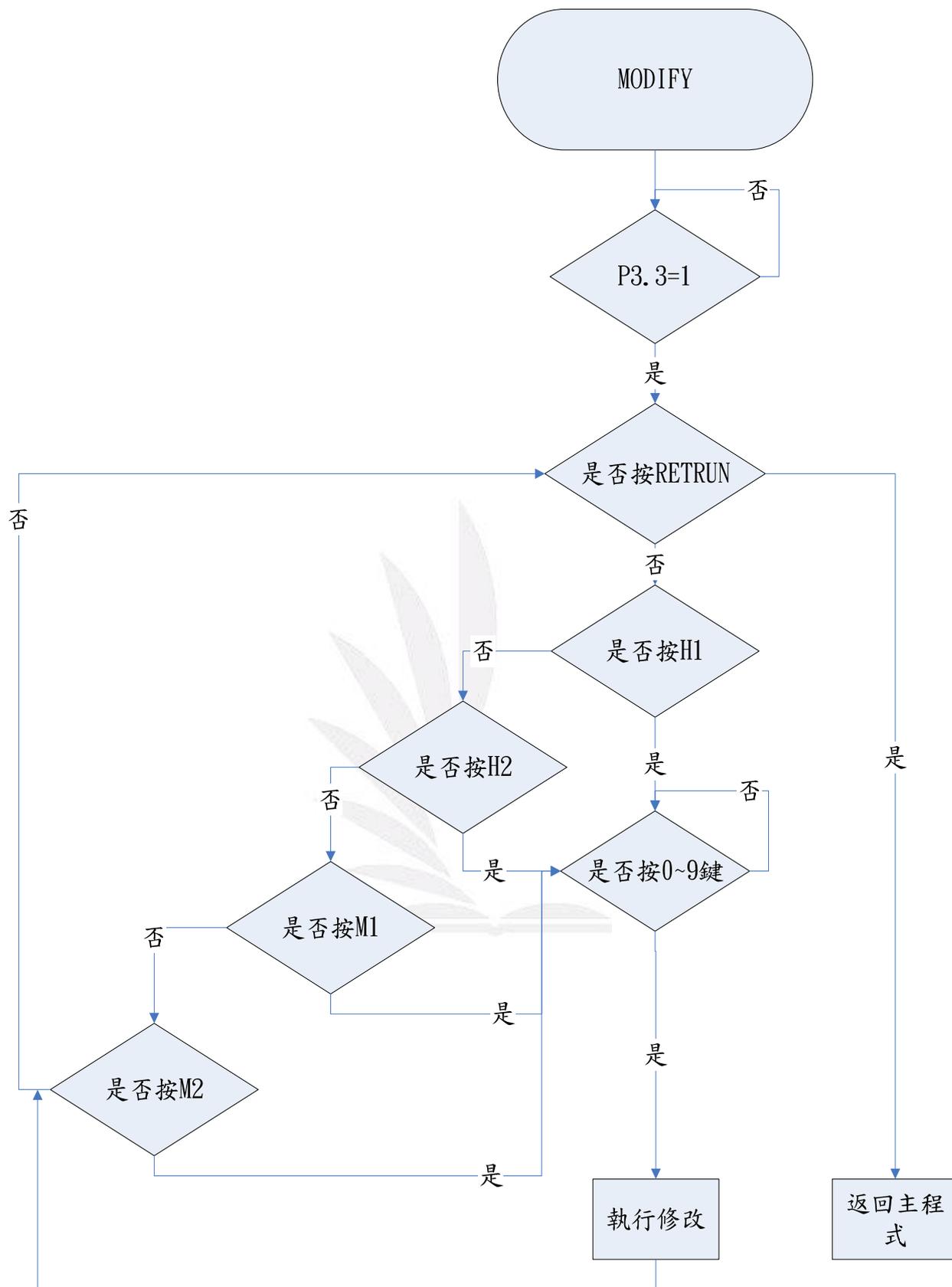


圖2.32 外部中斷副程式

第三章系統功能

本專題之主要功能如下：

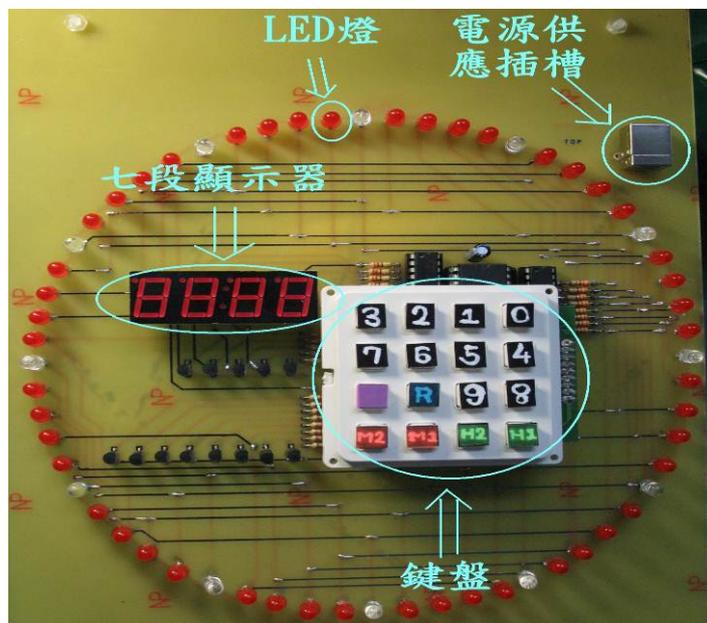
1. 時間顯示：用七段顯示器，顯示幾時幾分，並利用LED燈來顯示秒數，每一秒亮一個燈，同時七段顯示器上的兩點也會跟著閃爍，當時間一分鐘過後，LED燈一顆接著一顆滅，再過一分鐘後，LED燈又開始一顆一顆亮，一直輪循。
2. 時間設定：先利用按鈕跳至設定，接著按鍵盤上的數字鍵可更改數字，按H1、H2、M1、M2可跳至其相對位置進行修改，按R完成設定。



第四章實驗結果與操作說明

4-1 操作說明

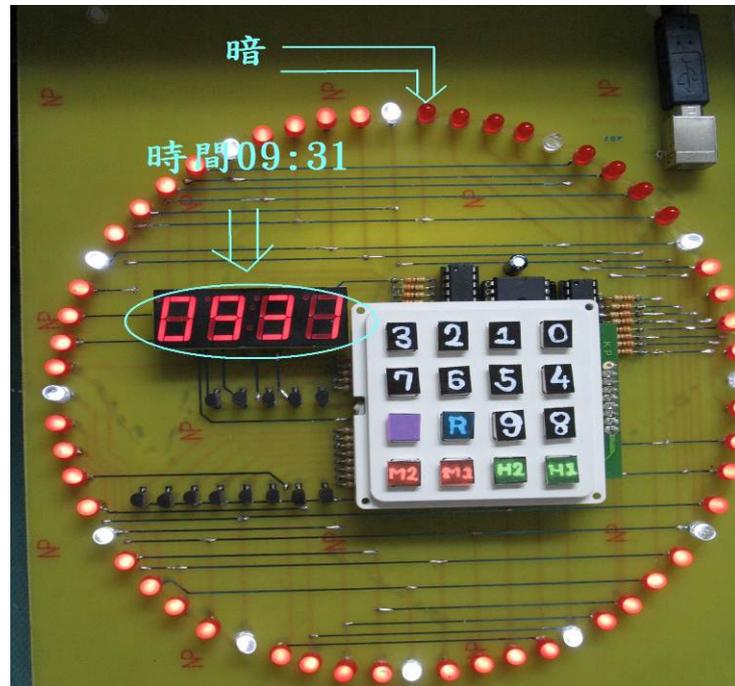
1. 元件位置介紹



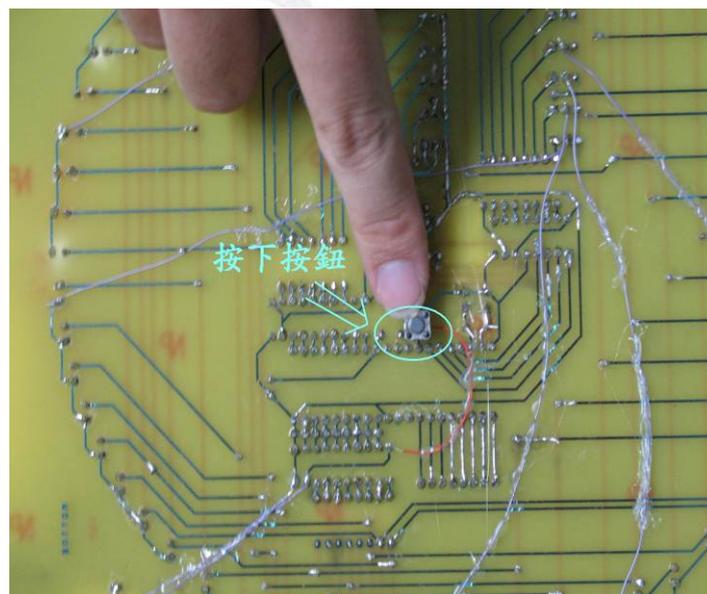
2. 接上電源，此時七段顯示器上會出現所設初值時間，LED 燈開始一顆接著一顆亮。



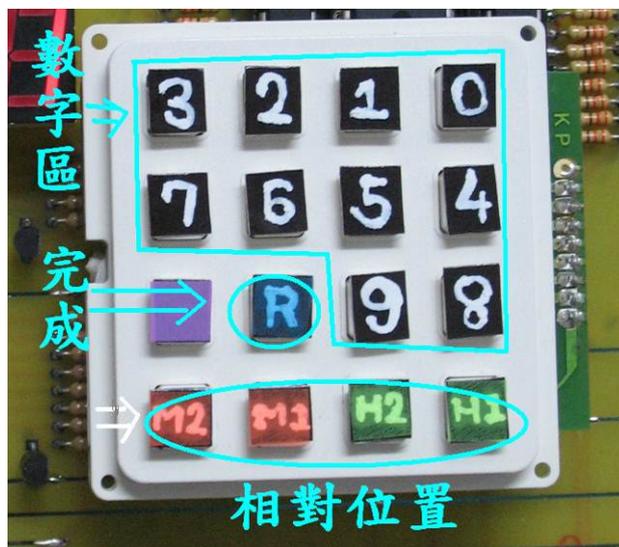
3. 一分鐘後一顆接一顆滅。



4. 按下設定按鈕後(在背後)，七段顯示器與 LED 燈停止，並可以開始設定時間。



- 5.按鍵盤上的數字鍵(如：1、2、3)調整所要時間，按 H1 鍵調整時數的十位數，H2 鍵調整時數的個位數，M1 鍵調整分鐘的十位數，M2 鍵調整分鐘的個位數。



- 6.按 R 鍵，設定完成，此時會看到七段顯示器上的時間為調整後的時間，且 LED 燈重新開始計時。
- 7.拔掉電源，此時七段顯示器與 LED 燈全滅。

4-2 實驗結果

- 1.LED 燈如預期所示，一顆接著一顆亮，並在 60 秒後，接著一顆接著一顆滅，一直輪循。
- 2.當 LED 燈當第 60 顆亮與第一顆滅的瞬間，七段顯示器的分鐘個位數跟著加 1。
- 3.當按設定按鈕後，七段顯示器停止且 LED 燈全滅。
- 4.按鍵盤上的數字(如：1、2、3)調整所要時間，七段顯示器上會出現相對應的數字，分別按 H1 鍵調整時數的十位數、H2 鍵調整時數的個位數、M1 鍵調整分鐘的十位數、M2 鍵調整分鐘的個位數，七段顯示器會在相對位置顯示修改過程。
- 5.按 R 鍵，設定完成，此時會看到七段顯示器上的時間為調整後的時間，且

LED 燈重新開始計時。

6. 拔掉電源後，七段顯示器與 LED 燈全滅。



第五章 結論與討論

5-1 改進方向

1. 在工作分配上可採取別組方式，2 2 1 或 2 3 的分組方式，這樣在做事時，有伴可以討論，不會自己埋頭苦幹。
2. 工作不必分的太細，這樣不會有人做的要死要活，有人卻閒著沒事做，而且工作沒細分，大家都可以學到每個部份。
3. 可多加LED燈的變化建表(雖然原先有建，但時間上來不及所以加進去，所以沒有放入)，使其變化看起來豐富些，不會貧乏無趣。
4. 可多加計時器功能或鬧鈴功能，使功能多一些。
5. 程式方面，由於基礎不好，一直修改，後來還出現位址亂跳的問題（這是一般不會注意到的事），所以寫程式時要注意位址的使用。
6. 在Layout時，要注意畫線的粗細，因為我們在第一塊板子線粗不夠，導致蝕刻後大部分的線都斷掉，耗費一堆時間在補線。

5-2 結論

在一番努力之後，終於把成品做出來，會覺得市面上所販售的時鐘，我們已經有能力可以製作，而且可以做得比市面上的更有創意，而且LED有一定亮度，半夜醒來時不必開燈也可知其時間，相當方便。

第六章隊員工作劃分

姓名	工作內容
葉俊廷	程式撰寫、焊接電路
張舜博	Layout 圖繪畫、洗板子
黃億欣	電路設計、接麵包版、焊接
王筱薇	專題本製作、協助 PPT 製作
林采潔	工作日誌記錄、PPT 製作、鑽孔



第七章 工作日誌

◎5/14：與老師討論時鐘基本功能，和預期完成之項目，因為和另一組題目相同，所以老師指導我們使用 LED 燈顯示時鐘功能，並加上自己的創意。

◎5/16：小組會議

1. 蒐集電子鐘相關資料。
2. 兩天後繳交自己的時鐘創意給組長。

◎5/18：

1. 更改原訂電子鐘預期功能—由原本的基本 LED 顯示改為 LED 配合馬達顯示（旋轉 LED 電子鐘），並採用老師的建議先往基本的電子鐘之程式和電路著手，再一步一步配合視覺暫留原理及馬達轉速進而完成專題。
2. 初步設計程式之大致流程及討論電路圖。

◎5/19：

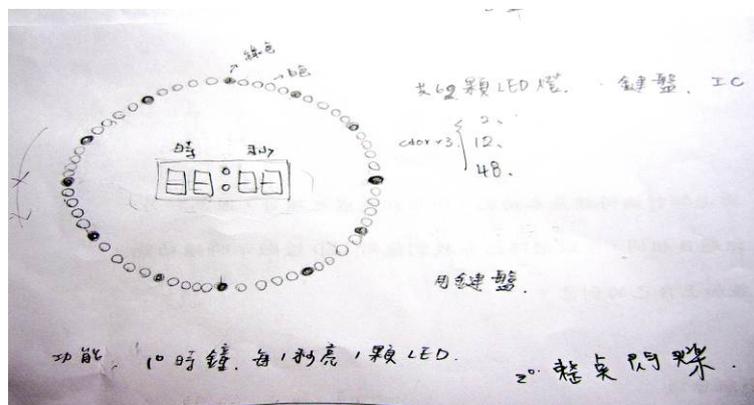
完成 OrCAD 電路繪製。

◎5/20：

修改上週老師說需要更改的投影片報告格式及內容。

◎5/22：

考量時間關係，所以不做旋轉 LED 電子鐘，改做利用 LED 燈來顯示秒針的基本電子鐘。

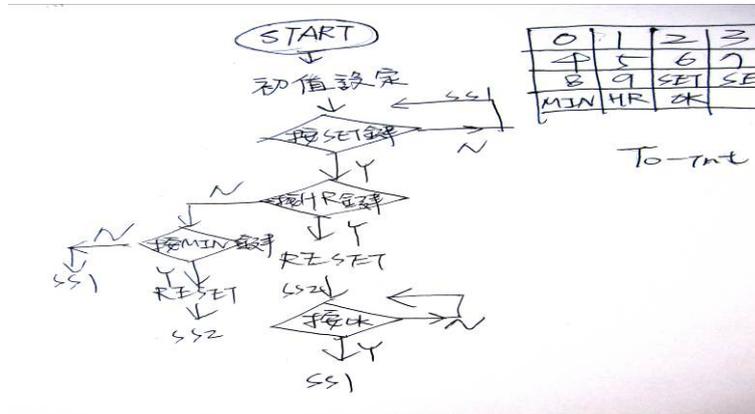


◎5/24：

重新討論程式的初步設計及完成流程圖。

◎5/25 :

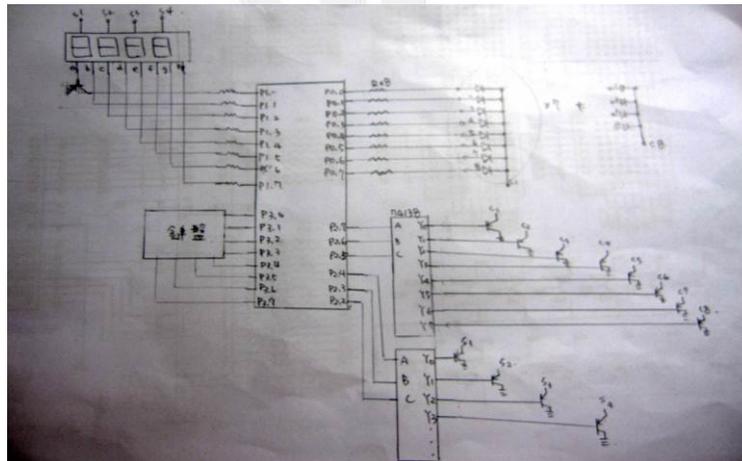
1.主程式流程圖完成。



2.討論如何繪製電路圖。

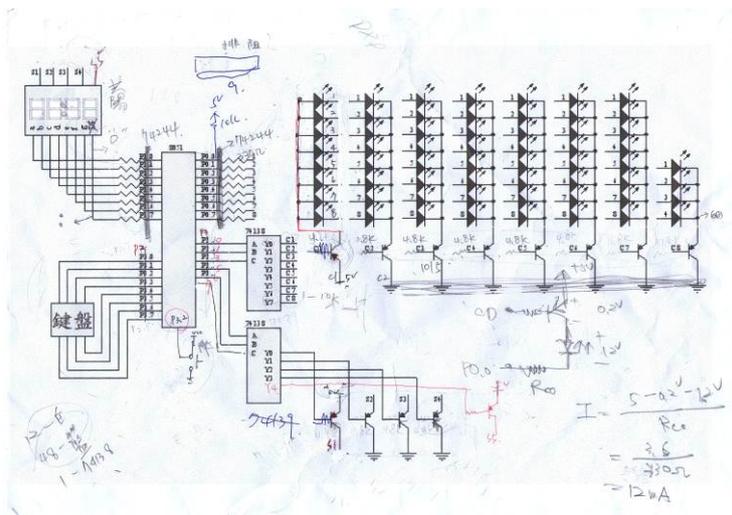
◎5/26 :

電路圖完成並且開始討論程式內容撰寫。



◎5/28 :

- 1.將電路圖拿給助教檢查，並且將些許錯誤改進，完成電路圖。
- 2.程式撰寫。



◎5/30.31 :

在麵包板上接上實際電路，測試電路是否有接錯。

◎6/02 :

完成 layout 圖。

◎6/04 :

撰寫程式。

◎6/05.06.07.08 :

1.洗電路板、鑽孔、銲接電路。

2.撰寫程式。

◎6/09 :

電路板測試、校正、裝飾電路板

第八章心得

葉俊廷 D9736276

在一年級的時候，就聽到很多學長說二年級做為處理機實習的專題的時候，會非常的辛苦!!當時的我並還不以為然，但到了現在才真的體會到當時他們所說的是甚麼意思...這個專題不只是讓我們學習如何從正課所學的微處理機的知識正確的應用到真正的實作上面、更是讓我們學到如何在一個小組中去分配好工作、去好好溝通、甚至是互相體諒!!我想這是我在這次專題中最深刻體會的...

而至於實作成品方面，我所負責的是程式撰寫的部分。還記得兩個禮拜前的我甚至是連組語的基本功都沒有，但經過老師以及所有住較耐心的教導之下，慢慢開始有了撰寫程式的概念。在撰寫的過程中，不乏碰到許許多多從未遭遇過的困難，起初總是會有想要放棄的念頭...但因為助教的耐心的教導，使我就在心裡告訴自己絕對不能放棄，不然就是對不起助教們了。也因為有了絕對不能放棄的理念，撰寫程式的能力也一步一步的進步，從剛開始連迴圈、中斷都不會寫的我，到現在我能把整個完整的時鐘寫出來...真的覺得自己進步的很多、也很感動終於能把這個困難重重的專題在期限之內完成!!

在這三個禮拜的努力中，所學到的真的是無法用言語來形容的，對老師以及助教們更是由衷的感謝，希望在以後的日子裡還能有機會在接觸到這些東西，因為真的是越做越有興趣呀~~!!

張舜博 D9776502

一個專題的製作真的是很不容易，從一開始的選題目到最後完成整個專題，中間會遇到非常多的問題，電路該如何去設計、LAYOUT 該怎麼去設定怎麼去LAY 才能夠完成、程式如何去跟電路配合、DEBUG.....等許許多多瑣碎的問題

要解決，而我們的過程中也真的遇到了這些問題。就我而言，光是 LAYOUT 就花了很久的時間，因為對於這軟體是第一次使用，因此什麼東西都得重頭學起，第一次 LAY 完了，卻發現線寬沒改、孔徑沒改，但因為圖有點複雜，改了又會碰到而造成短路，因此只好再重新做一次。不過有了經驗之後，第二次做起來就快了許多，也比較清楚該怎麼配置元件能使得自己更好牽線，這些都是由於第一次的失敗而學到的。所以我覺得失敗有時也不是一件壞事，重要的是能從失敗中學習到東西，使自己下次不要再重蹈覆轍。而製作專題的這段時間，因為有許多地方不懂而常常去請教助教們，真的是十分麻煩他們，假如沒有他們的幫忙，可能這份專題會難產吧！在此十分的謝謝助教們。也謝謝組員們彼此間的幫忙合作，或許之間會有摩擦及不愉快，但相信都是為了做好這份專題而產生的，從中也學到團隊合作的重要性，唯有齊心齊力，才能做好一件事，謝謝大家，大家都辛苦了。

黃億欣 D9776529

「一個成功的專題，背後一定有一位厲害的指導老師和一群盡心盡力的助教」。在整個實作過程中，從決定題目到完成電路板，這中遇到了許多難題，多虧老師和助教們的熱心指導和幫忙，我們才能在規定時間順利完成，也因為他們的專業讓我學到了不少課堂上尚未學到的學識和技能。

對一般人而言，也許是因為太過普遍，時鐘似乎只是個不起眼的生活用品。但經由這次的專題製作，我才了解完成一個電子時鐘的背後並不是想像中的容易、一個小小的單晶片竟然能夠使大家完成各式各樣的大小專題，微電腦這種東西真的很微妙。用自己製做的燒錄器燒錄程式，真的是很奇妙的喜悅感。

能夠認真投入這份專題製作，我覺得我真是幸運的人，整個歷程不僅讓我學到了相關的專業知識，焊接技術也大幅增進。經過這次製作專題的磨練，讓我成長了不少，其中的體會和感受真的是言語所無法形容的。

王筱薇 D9776559

從學期一開始就知要做專題，當時的心想一定很有趣，想說要做出個好玩或有趣的東西，可是當選題目時，組員不幸的猜拳猜輸，當時題目為 PSW，可是經過大家討論，覺得那不知是什麼東西，要全部重頭來會浪費很多時間，於是決定改題目，誰知在改完題目後，才發現跟別組衝到，由於很多東西早就規劃好，所以就只能繼續下去。中途做的時候，老師希望我們做的時鐘可以作旋轉，想說就專心作旋轉時鐘，後來發現我們會錯意，不但原先的沒進度，旋轉的也沒進度，整個大延遲，到期末整個就趕趕趕，還因為工作分配沒講好，導致大家的不愉快，我想大家應該是因為幾乎都一個禮拜都沒睡飽過，睡眠不足而導致脾氣暴躁，我個人是不太介意，畢竟他們所說的各有一套論點，誰對誰錯無法考究，把事情做好最重要。而在這次的分組做專題中，我發現溝通真的很重要，而且要適時的去相信組員，不要因為不信任而全部自己做，會容易導致兩頭空，當然也要適時的詢問組員，或許組員正在等待我們開口幫忙。

D9776593 林采潔

從這個專題中，我了解 1.許多元件的用途和 2.設計電路圖的邏輯方向。

剛開始著手這個專題的時候什麼都不太懂，到現在完成專題，當中不知道經歷多少的挫折、修改，一次一次的詢問老師跟助教，終於專題慢慢的完成，在快完成的時候，所有人都很感動，辛苦真的有值得！

就我們專題製作小組而言，大家都是很盡心盡力的想做好專題，所以造成有些溝通上的問題而造成不愉快，就像有時候想要協助其他組員的工作，但是有可能溝通上出了問題而導致有點小爭執，不過我們都知道大家的出發點都是為了能把專題做好，所以將心比心，在爭執過後反而感情更好，更能夠將專題做好。

尤其是我們組長葉俊廷跟副組長黃億欣，對於這個專題的付出，我這個小組元都五體投地，他們為了專題花的時間是我們的一倍，都只是為了我們的專題！跟在他們身邊我感到十分光榮也學習到許多東西。

在這次專題的結束，我想對於以後其他專題製作，我能更輕易上手，能將這次的經驗用在上面讓未來更不容易出錯。希望在專題的發表會上，我們能夠將這嘔心瀝血的專題內容好好呈現給各位！



附錄 B

算數運算指令

指令	說明	位元組	工作週期 (時脈數)
ADD A,Rn	暫存器累加至累加器	1	12
ADD A,direct	直接位元組加至累加器	2	12
ADD A,@Ri	間接位元組加至累加器	1	12
ADD A,#data	常數值加至累加器	2	12
ADDC A,Rn	與C一起將暫存器加至累加器	1	12
ADDC A,direct	與C一起將直接位元組加至累加器	2	12
ADDC A,@Ri	與C一起將間接位元組加至累加器	1	12
ADDC A,#data	與C一起將常數值加至累加器	2	12
SUBB A,Rn	累加器減暫存器再減C	1	12
SUBB A,direct	累加器減直接位元組再減C	2	12
SUBB A,@Ri	累加器減間接位元組再減C	1	12
SUBB A,#data	累加器減常數值再減C	2	12
INC A	累加器加一	1	12
INC Rn	暫存器加一	1	12
INC direct	直接位元組加一	2	12
INC @Ri	間接位元組加一	1	12
DEC A	累加器減一	1	12
DEC Rn	暫存器減一	1	12
DEC direct	直接位元組減一	2	12

DEC @Ri	間接位元組減一	1	12
INC DPTR	資料指標加一	1	24
MUL AB	A 乘以B	1	48
DIV AB	A 除以B	1	48
DA A	累加器作BCD 調整	1	12

邏輯運算指令

指令	說明	位元組	工作週期 (時脈數)
ANL A,Rn	暫存器AND 至累加器	1	12
ANL A,direct	直接位元組AND 至累加器	2	12
ANL A,@Ri	間接位元組AND 至累加器	1	12
ANL A,#data	常數值AND 累加器	2	12
ANL direct,A	累加器AND 至直接位元組	2	12
ANL direct,#data	常數AND 至直接位元組	3	24
ORL A,Rn	暫存器OR 至累加器	1	12
ORL A,direct	直接位元組OR 至累加器	2	12
ORL A,@R i	間接位元組OR 至累加器	1	12
ORL A,#data	常數值加OR 累加器	2	12
ORL direct,A	累加器OR 至直接位元組	2	12
ORL direct,#data	常數OR 至直接位元組	3	24
XRL A,Rn	暫存器XRL 至累加器	1	12
XRL A,direct	直接位元組XRL 至累加器	2	12
XRL A,@Ri	間接位元組XRL 至累加器	1	12

XRL A,#data	常數值加XRL 累加器	2	12
XRL direct,A	累加器XRL 至直接位元組	2	12
XRL direct,#data	常數XRL 至直接位元組	3	24
CLR A	清除累加器	1	12
CPL A	累加器反相	1	12
RL A	累加器向左旋轉	1	12
RLC A	累加器與C 一起向左旋轉	1	12
RR A	累加器向右旋轉	1	12
RRC A	累加器與C 一起向右旋轉	1	12
SWAP A	累加器的高低四位元交換	1	12

資料轉移指令

指令	說明	位元組	工作週期 (時脈數)
MOV A,Rn	暫存器內容移至累加器	1	12
MOV A,direct	直接位元組內容移至累加器	2	12
MOV A,@Ri	間接位元組內容移至累加器	1	12
MOV A,#data	常數值移至累加器	2	12
MOV Rn,A	累加器內容移至暫存器	1	12
MOV Rn,direct	直接位元組內容移至暫存器	2	24
MOV Rn,#data	常數值移至暫存器	2	12
MOV direct,A	累加器內容移至直接位元組	2	12
MOV direct,Rn	暫存器內容移至直接位元組	2	24

MOV direct,direct	直接位元組內容移至直接位元組	3	24
MOV direct,@Ri	間接位元組內容移至直接位元組	2	24
MOV direct,#data	常數移至直接位元組	3	24
MOV @Ri,A	累加器內容移至間接位元組	1	12
MOV @Ri,direct	直接位元組內容移至間接位元組	2	24
MOV @Ri,#data	常數移至間接位元組	2	12
MOV DPTR,#data 16	16 位元常數移至資料指標	3	24
MOVC A,@A+DPTR	程式記憶體的資料移入累加器	1	24
MOVC A,@A+PC	程式記憶體的資料移入累加器	1	24
MOVX A,@Ri	外部RAM的資料移入累加器 (8位元定址)	1	24
MOVX A,@DPTR	外部RAM的資料移入累加器 (16位元定址)	1	24
MOVX @Ri,A	累加器內容寫到外部RAM (8位元位址)	1	24
MOVX @DPTR,A	累加器內容寫到外部RAM (16位元位址)	1	24
PUSH direct	直接位元組內容放至堆疊區	2	24
POP direct	從堆疊區拿回資料至直接位元組	2	24
XCH A,Rn	累加器與暫存器的內容互換	1	12
XCH A,direct	累加器與直接位元組的內容互換	2	12
XCH A,@Ri	累加器與間接位元組的內容互換	1	12
XCHD A,@Ri	累加器與間接位元組的低四位元 互換	1	12

程式分控指令

指令	說明	位元組	工作週期 (時脈數)
ACALL addr	絕對式副程式呼叫	2	24
LCALL addr	遠程副程式呼叫	3	24
RET	從副程式返回	1	24
RE TI	從中斷副程式返回	1	24
AJMP addr	絕對式跳躍	2	24
LJMP addr	遠程跳躍	3	24
SJMP rel	短程跳躍	2	24
JMP @A+DPTR	間接跳躍	1	24
JZ rel	若 A=0 就跳躍	2	24
JNZ rel	若 A≠0 就跳躍	2	24
CJNE A,direct,rel	若累加器與直接位元組內容，不相等就跳躍	3	24
CJNE A,#data,rel	若累加器內容不等於 data 就跳躍	3	24
CJNE Rn,#data,rel	若暫存器內容不等於 data 就跳躍	3	24
CJNE @Ri,#data,rel	若間接位元組內容不等於 data 就跳躍	3	24
DJNZ Rn,rel	暫存器內容減一，若不等於零就跳躍	2	24
DJNZ direct,rel	直接位址內容減一，若不等於零就跳躍	3	24

NOP	沒動作	1	12
-----	-----	---	----

布林變數指令

指令	說明	位元組	工作週期 (時脈數)
CLR C	清除進位旗標	1	12
CLR bit	清除bit	2	12
SETB C	設定進位旗標	1	12
SETB bit	設定bit=1	2	12
CPL C	進位旗標反相	1	12
CPL bit	bit 反相	2	12
ANL C,bit	bit AND 至進位旗標	2	24
ANL C,/bit	bit 反相後再AND 至進位旗標	2	24
ORL C,bit	bit OR 至進位旗標	2	24
ORL C,/bit	bit 反相後OR 至進位旗標	2	24
MOV C,bit	bit 之狀態移至進位旗標	2	12
MOV bit,C	進位旗標之狀態移至bit	2	24
JC rel	若C=1 就跳躍	2	24
JNC rel	若C=0 就跳躍	2	24
JB bit,rel	若bit=1 就跳躍	3	24
JNB bit,rel	若bit=0 就跳躍	3	24
JBC bit,rel	若bit=1 就跳躍，且清除此位元	3	24

參考文獻

- [1] 蔡朝洋編譯，單晶片微電腦 8051/8951 原理與應用，全華圖書，2008 年 8 月，第十版
- [2] 蕭閔學編著，大學電子學實習 I-電子電路分析篇，台科大圖書，2008 年 9 月，初版
- [3] Muhammad Ali Mazid、Janice Gillispie Mazidi、Rolin D.McKinlay，The 8051 Microcontroller and Embedded Systems，Upper Saddle River、Pearson/Prentice Hall，2006 年，第二版.
- [4] 郭庭吉, 吳金戌編著，單晶片 8051 專題製作：使用 Keil uvision3 組合語言，文魁資訊，2008 年 8 月，初版

