

微電腦控制技術 本位訓練教材
MCS-51 LED 點陣陣顯示控制
專題製作

編號：PEN-MCC0507

編著者：胡茂林

審稿者：杜日富

主辦單位：行政院勞工委員會職業訓練局

研製單位：中華人民職業訓練研究發展中心

印製日期：九十一年十一月

單元 PEN-MCC0507 學習指引

當你要學習本單元之前，您必須先熟悉下列知識：

(1) 瞭解 MCS-51 指令的控制與應用 (PEN-MCC0504)

如果你尚無法勝任以上所列知識，請你先將本教材單元放回原位，並回去學習上列須先熟悉的知識單元，或請教你的老師。

如果你能夠勝任以上所列知識，請翻到下一頁進行學習。

引言

本單元為 MCS-51 學習單元中的最後綜合學習單元。為了讓學員能利用本綜合學習單元的學習效果，本單元採用了工業電子甲級技術士檢定題目的第一題，做為本專題製作學習單元的題目，仔細研讀並實作，一定可得到相對的成果。並可具有報考工業電子甲級技術的部分能力。

定義

TTL 界面電氣特性：電晶體-電晶體邏輯電路晶片 TTL(Transistor-Transistor Logic)為一種以電晶體電路為輸入，也以電晶體電路為輸出的邏輯積體電路。其電源電壓固定為 5 伏特。相較於 CMOS 晶片，則 TTL 晶片的功率消耗遠高於 CMOS 晶片，但 TTL 晶片運作速度比較快。

對於 TTL 晶片，因使用的電源電壓是統一為 5 伏特，所以各項電氣信號準位也有統一的標準：

V_{oh} (輸出端輸出高準位的電壓)：2.4V 以上到 5.0V 以下

V_{ih} (被輸入端視為高準位的電壓)：2.0V 以上到 5.0V 以下

V_{ol} (輸出端輸出低準位的電壓)：0V 以上到 0.4V 以下

V_{il} (被輸入端視為低準位的電壓)：0V 以上到 0.8V 以下

故可容許 0.4V 的雜訊干擾

I_{oh} (輸出端輸出高準位的流出電流)：最少 400 μA

I_{ih} (輸入端輸入高準位的吸入電流)：最多 40 μA

I_{ol} (輸出端輸出低準位的吸入電流)：最少 16mA

I_{il} (輸入端輸入低準位的流出電流)：最多 1.6mA

故標準 TTL 的扇出數(Fan Out: 一個輸出閘所能推動輸入閘的數目)為 10 個。

學習目標

- 一、學員能在 30 分鐘內，依據本專題材料表中所列之材料限制下，繪出可以符合本專題所需之功能需求且符合電氣特性之電路圖。
- 二、學員能在 5 分鐘之內，將所指定的三個 8×8 點矩陣字型轉譯成字形碼。
- 三、學員能在 30 分鐘內，在不參考任何資料下，繪出所給之矩陣式鍵盤電路的掃描程式流程圖。

學習活動

本單元之學習活動，主要是在學習如何利用 MCS-51 單晶片建構出點矩陣 LED 顯示掃描電路，掃描的軟體定時、遮沒及資料結構有關的軟體應用，鍵盤矩陣的電路及掃描、消除彈跳的軟體設計應用。其中包含顯示掃描計時中斷的應用，顯示遮沒的必要之原因，字形碼的取得、顯示功能暨顯示緩衝區建立的關連性，鍵盤的掃描方法及彈跳的產生及消除方法。

首先你應先詳細閱讀本單元第一個學習目標，瞭解字形的建構方法，及點矩陣 LED 顯示器的結構及控制方法。然後再準備第二學習目標所需之工具與材料，並依顯目要求設計出基本電路。再進入第三個學習目標將軟體程式設計完成，直到測試成功：

參考書籍：八十六年度數位甲級技術士技能檢定術科試題，職訓局。

參考網站：<http://elec.nihs.edu.tw>

本單元的第一個學習目標是

學員能在 30 分鐘內，依據本專題材料表中所列之材料限制下，繪出可以符合本專題所需之功能需求，且符合電氣特性之電路圖。

壹、數位電子甲級技術士技能檢定術科試題使用說明：

- 一、本試題係配合檢定前公佈試題原則命製，主辦單位應於考生繳款後儘速將全份試題寄交應檢人員以憑準備。
- 二、術科測驗時間八小時，評分時間二小時。
- 三、主辦單位須按應檢人數，備妥各試題所需材料及規格表或資料手冊（每十人增加備份材料一份）。
- 四、主辦單位請依場地設備表備妥各項機具設備、儀表等，供應檢人員使用。
- 五、主辦單位須裝置完成一符合試題說明及動作要求之電路板，以供測試用。
- 六、本檢定題目包括四題，每一場次術科測驗時，由每一應檢人員自行抽定一題實施檢定。
- 七、應檢人員可於檢定完畢待監評人員評分後，將場地所供給之材料領走，若有異議須當場提出，離場後概不負責。
- 八、檢定當天評分完畢時，凡未領走材料者，主辦單位不負保管責任。

貳、數位電子甲級技術士技能檢定 術科應檢須知（發受檢人、評審員、檢定單位）

- 一、考生應按試題之參考資料進行電路設計、裝配、測試及調整，並設計程式以達成試題說明之動作要求。檢定時間為八小時，其工作要點如下：
 - （一）考生領取材料後，應依據功能要求，利用所供給材料自行設計電路，並裝配、焊接或繞接於檢定 PC 板上。
 - （二）利用所裝配完成之電路，自行設計程式，以達成所要求功能，並燒錄在單晶片中。
 - （三）組裝工作請參考各圖說由受檢人自行完成。
- 二、注意事項：
 - （一）應檢當天請攜帶身份證明文件及術科測驗通知單入場，凡無身份證明文件（或當天考試結束前未備妥）以資確認者，一律取消應檢資格。
 - （二）檢定開始十五分鐘內未入場應檢者視為放棄，取消應檢資格，如欲領取檢定材料，請於本場次檢定結束前，攜帶證件向承辦單位填單領取材料，逾期不予受理。
 - （三）試題及自備工具表所列工具請自行攜帶，檢定單位不另提供，凡未自備手工具或試題而必須向考場借取者，一律按規定扣分。
 - （四）檢定開始三十分鐘內，受檢人員應自行檢查所需使用之材料有否缺漏、或是否良好，如有問題應立即報告監評人員處理，否則一律視為受檢人員疏忽，按規定扣分。

- (五) 同一元件僅可更換二次（以損壞元件交換），每更換一元件均列入扣分。
 - (六) 通電檢驗若發生短路現象（無熔絲開關跳脫、或插座保險絲熔斷者），即應停止工作，不得重修，並以不及格論。
 - (七) 受檢人員不得接受他人協助、或協助他人（如動手、講話、動作提示、軟體拷貝……），一經發現視為作弊，雙方均以不及格論。
 - (八) 受檢人員未經監評人員允許私自離場、或離場不歸者，以不及格論。
 - (九) 凡故意損壞公物、設備，除應負賠償責任外，一律取消應檢資格。
 - (十) 受檢人員於檢定完畢後、或離場前，應作適當之現場清理工作，否則按規定扣分。
 - (十一) 凡有未盡事宜，於現場說明。
- 三、應考時間共計八小時，午餐由受檢人員自行準備。

參、裝配原則：

一、裝配說明：

- (一) 元件裝置於電路板時，由低至高依序安裝。
- (二) 電阻器安裝於電路板時，色碼之讀法必須由左而右，由上而下方式安裝。
- (三) 元件標示之數據必須以方便目視、及閱讀為原則。
- (四) 元件裝配應與電路板貼密，唯 1W 以上電阻器或電晶體、整流器……等，必須有 5~8 mm 高度。
- (五) IC 須使用 IC 座，不可直接焊於電路板上。
- (六) 線材與連接器端子焊接時，重疊區佔端子長 1/2 以上，鉚錫覆蓋重疊區至少 2/3 以上，且線材之 PVC 皮至端子的間距應小於 2 mm。

二、焊接說明：

- (一) 焊接可採用先焊接後剪腳，餘長不得超過 0.5 mm；或先剪腳再焊接，唯 IC 座接腳不得剪除。
- (二) 元件接腳不得彎曲延伸至銅箔圓孔邊緣以外。
- (三) 焊錫應佈滿銅箔面之元件接腳圓點內。
- (四) 焊接時焊錫量應適中不得有冷焊、針孔、氣泡……。
- (五) 焊接面可使用裸銅線或絕緣線，以整齊美觀為原則。

肆、應檢人員自備工具表：

項目	名稱	規格	單位	數量	備註
1	尖嘴鉗	5"或6"	支	1	
2	斜口鉗	5"或6"	支	1	
3	起子組	十字、一字	組	1	
4	電表	3 1/2 數位式或一般指針式	具	1	
5	文具	原子筆、尺(15公分以上)等	組	1	
6	電烙鐵	30W/AC110V	支	1	
7	吸錫器		支	1	
8	積體電路拔插器	U型	支	1	
9	烙鐵架及海棉	組裝式	組	1	
10	繞線槍(筆)	可繞鍍銀線	支	1	

註：受檢時可自備個人電腦設備，規格可參考場地設備表。

伍、場地設備表（檢定人數 10 人）：

項次	名稱	規格	價格	單位	數量	備註
1	示波器	雙軌跡 20MHz 以上		台	10	
2	IC 燒錄器	可燒錄 89C51/8751/PIC16C5x		套	10	
3	列表機	80 行以上		台	13	
4	數位 IC 測試器	可測 TTL、CMOS		台	2	
5	電腦設備	至少含有： 1. 80486 或以上 2. 硬碟至少 120MB 容量 3. 軟碟 1.2M 或 1.44M 4. 單色或彩色螢幕 5. DOS V6.0 以上（具合法版權）		套	10	
6	電源供給器	0~±30V 3A、及+5V 3A		台	10	
7	繞線槍（筆）			支	5	
8	電晶體測試器	可測 β 值、及 I_{ceo}		台	2	
9	烙鐵架及海棉	組裝式		組	10	
10	積體電路整腳器	8~40P、適用 DIP 型		支	10	
11	工作桌椅	桌面 120×60 公分以上含 AC110V±10%電源		組	10	
12	場地	光線充足、日曬遮光		m ²	60	
13	空調設備	窗型或箱型冷氣，依場地大小設置				
14	組譯程式	含 8751/8951 與 PIC16C5 之 Compiler、Link 程式（須具合法版權）		套	10	
15	文書處理程式	DOS 之 Editor		套	10	
16	資料手冊	89C51、PIC16C57、TTL、CMOS、LCD 模組		冊	各 5	
17	列表紙	80 行		箱	6	
18	實作樣品	含 DEMO 程式		套	各 2	
19	模擬器	可模擬 PIC 16C5X 系列（含組譯程式）		套	4	
20	模擬器	可模擬（含組譯程式）		套	8	

※1~20 項次須備份一份

請翻至下一頁。

陸、試題

試題一：

- 一、 試題編號：117-890101。
- 二、 試題名稱：LED 點矩陣顯示控制電路。
- 三、 檢定時間：八小時。
- 四、 試題說明及動作要求：

(一) 試題說明：

1. 本電路是由單晶片 IC 89C51 掃描 4 個按鍵，並控制一個 8×8 LED 點矩陣顯示器。
2. 請依電路板元件佈置參考圖、及所供給材料自行設計電路，並裝配、焊接或繞接於多孔萬用電路板上。
3. 自行設計程式並燒錄在單晶片中，使所裝配完成之電路能達成下列的動作要求。
4. 單晶片各埠之動作模式，由監評委員於檢定當場指定。

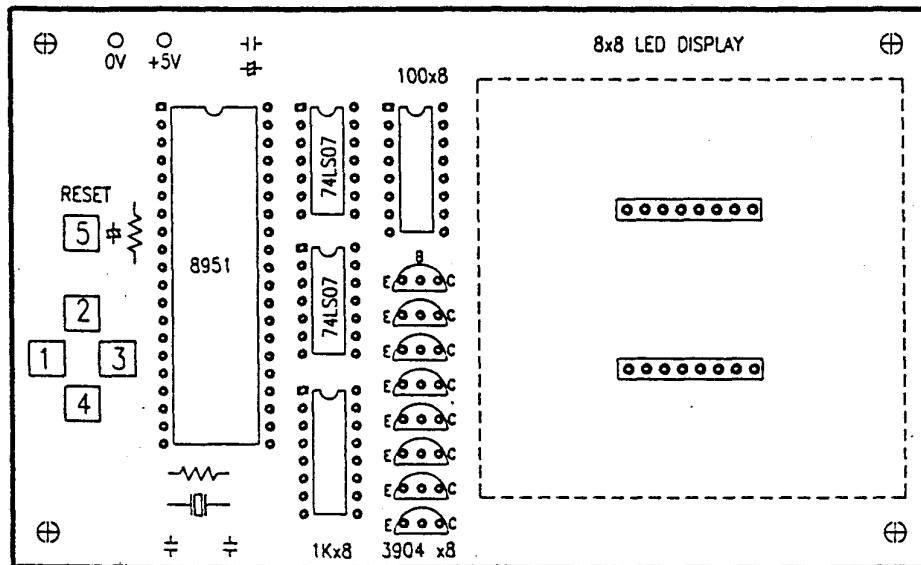
(二) 動作要求（實際之動作功能於檢定當場公佈，下列參考例，僅供練習參考使用，並非實際試題）：

1. 所設計之電路在電源 ON 時，以及在任意時刻按下重置(RESET)鍵時，具有能顯示評審於檢定當場所指定的圖形文字之功能。
參考例：全部 LED(8×8)同時 ON、OFF 閃爍 N 次，閃爍之間隔 T 秒。
2. 當按下“1”鍵後，顯示器顯示某一圖形文字、或外加某些功能（由監評委員於檢定當天指定）。
參考例：顯示英文字母“A”。
3. 當按下“2”鍵後，顯示器顯示某一圖形文字、或外加某些功能（由監評委員於檢定當天指定）。
參考例：將動作要求 B 所顯示之英文字母“A”由右向左移。
4. 當按下“3”鍵後，顯示器顯示某一圖形文字、或外加某些功能（由監評委員於檢定當天指定）
參考例：將動作要求 B 所顯示之英文字母“A”由下向上移。
5. 當按下“4”鍵後，顯示器顯示某一圖形文字、或外加某些功能（由監評委員於檢定當天指定）
參考例：顯示受檢人員本身之身份字號，每一字母顯示間隔 1 秒。

(三) 文件：

1. 繪製所設計之電路圖。
2. 繪製程式流程圖。
3. 列印程式及說明（中英文不拘）。

五、電路板元件佈置參考圖：



六、供給材料表：

項次	名稱	規格	單位	數量	備註
1	8×8 LED 顯示器	8×8 5m/m	個	1	
2	單晶片 IC	89C51 電子式清洗型	個	1	
3	緩衝 IC	74LS07	個	3	
4	電晶體	2N3904	個	10	
5	IC 座	14P 長腳型及短腳型	個	各 2	
6	排阻	100Ω 雙排型 16P、8 組	個	1	
7	排阻	1KΩ 雙排型 16P、8 組	個	1	
8	排腳座	單排 16P 長腳型	個	1	
9	石英晶體	11.0592MHz	個	1	
10	電解電容	22 μ F/25V， \pm 20%	個	6	
11	陶質電容	10PF/50V	個	2	
12	積層電容	0.1 μ F	個	4	
13	碳膜電阻	10KΩ 1/4， \pm 5%	個	2	
14	碳膜電阻	10MΩ 1/4， \pm 5%	個	2	
15	IC 座	40P 長腳型及短腳型	個	各 1	
16	IC 座	16P 長腳型及短腳型	個	各 2	
17	按鍵開關	常開型	個	5	
18	多孔萬用電路板	雙面 160 mm×100 mm	片	1	
19	銅柱	雙凹 3×15m/m	個	4	蚱螺絲
20	接針	14 PIN 長腳型	個	1	
21	30 號鍍銀 OK 線	三種顏色	公尺	各 2	
22	焊錫	H60/W0.8	公尺	1	
23	SVR	10KΩ	個	各 2	

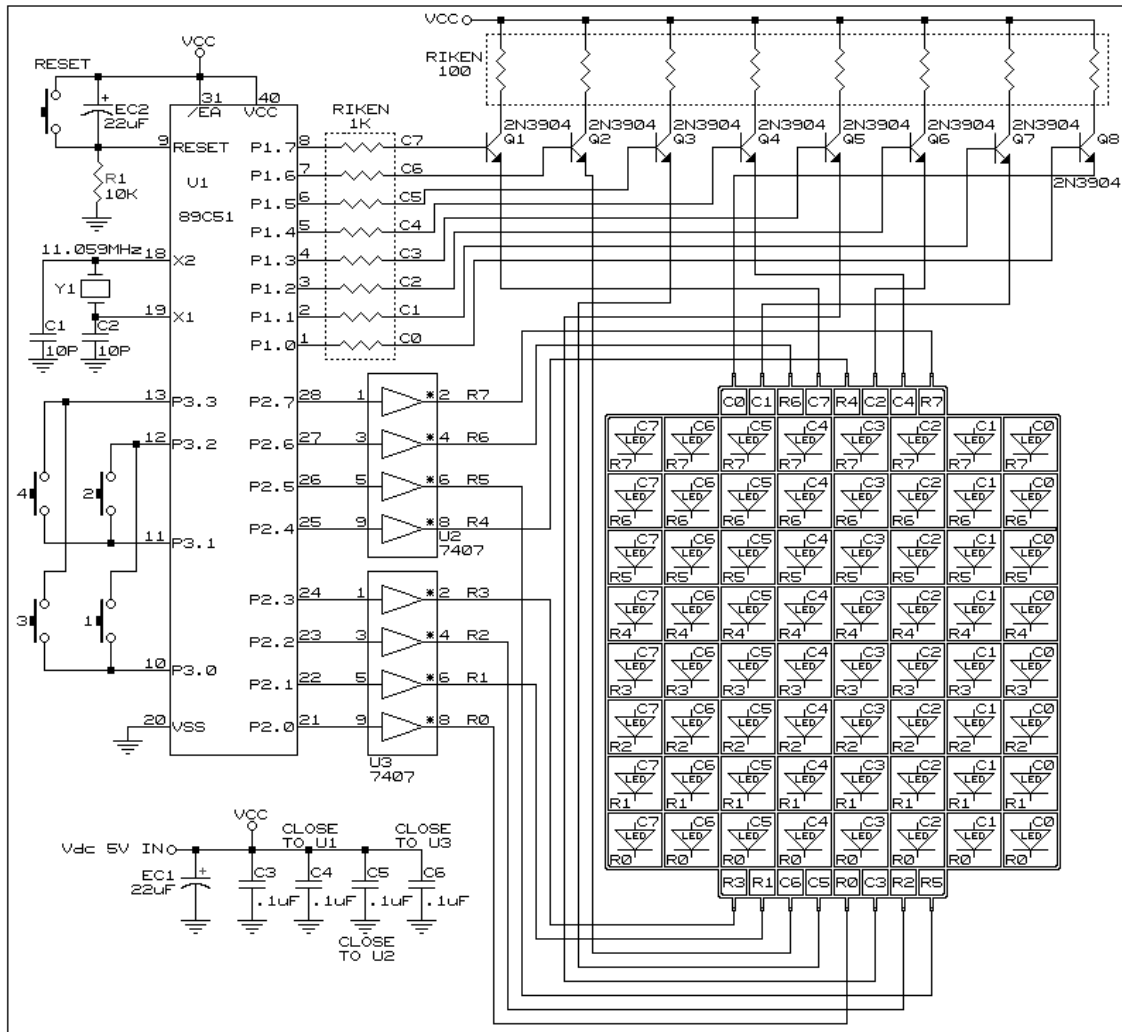
柒、評分表

姓名		抽籤編號 座位號碼		評審 結果	<input type="checkbox"/> 合格	
學科准考 證號碼		檢定日期	年 月 日		<input type="checkbox"/> 不合格	
術科測 驗通知 單號		領取測 驗材 料處 簽名				
項目	評 分 標 準	扣 分 標 準		配 分	實 分	備 註
		每處扣分	最高扣分			
一、依據應檢須知規定以不合格論者 <input type="checkbox"/>		列為左項之一者不予評分，並請考生在本欄簽名：				
二、未能於規定時間內完成硬體組裝者 <input type="checkbox"/>						
三、任一按鍵毫無功能或動作者以不合格論 <input type="checkbox"/>						
四、 硬 體 裝 配	1. 更換被動零件者	2	10	20 分		
	2. 更換主動零件(IC、晶體……等)者	10	20			
	3. 導線繞過電路板外緣而連接者	1	5			
	4. 電路板焊接面有零件者	1	5			
	5. 跳線有跨越零件者	1	5			
	6. 其他未按裝配規則裝配者	1	5			
五、 功 能 要 求	1. Reset 按鍵無法正常動作者(有 Bug 者)	20	20	40 分		
	2. 按鍵 1 無法正常動作者(有 Bug 者)	5	20			
	3. 按鍵 2 無法正常動作者(有 Bug 者)	5	20			
	4. 按鍵 3 無法正常動作者(有 Bug 者)	5	20			
	5. 按鍵 4 無法正常動作者(有 Bug 者)	5	20			
六、 工 作 安 全 與 習 慣	1. 使場地個人電腦中毒者	10	10	30 分		
	2. 未自備手工具或試題者	5	5			
	3. 工作桌面凌亂者	5	5			
	4. 不符合工作法暨安全程序要求者	5	10			
	5. 離場前未清理工作崗位者	5	10			
七、 文 件	1. 電路的設計	2	10	20 分		
	2. 流程圖的繪製	2	10			
	3. 程式的列印與演算片	2	10			
總 分			扣 分			
			得 分			
評 審 員 名						

註：凡因場地設備及材料規格不符致影響考生權益，概由承辦單位負責。

捌、電路硬體結構

- 一、 雖可使用繞線槍，但以焊接方式較為習慣者，還是選用焊接方式。
- 二、 試題要求常會當場以矩陣掃描方式來要求建構按鍵電路，故雖然同樣使用四支 I/O 接腳，但還是以矩陣掃描方式為之。
- 三、 所有功能動作要求均以參考例為之。假設受檢人員身份證字號為：R123456789
- 四、 電路圖如下所示：



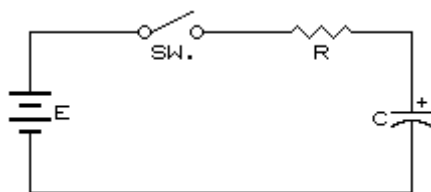
五、電路原理說明：

(一) 電源電路：電源採用單一 5 伏特直流電源即可，矩陣式 LED 顯示器之亮度，只要提供電流足夠，亮度即可達到需求，與電壓之高低，可以脫離絕對的關係，LED 的點亮在靜態時（非掃描方式）只要 1V，10mA 即可。故只要高於 1V 以上之電源，即可正常點亮 LED。

任何一片獨立的電路板，均要有一電源濾波器，以防止電源饋線的突波或雜訊干擾。供給材料表上有 $22\ \mu\text{F}$ 電容，故使用一只 $22\ \mu\text{F}$ 電容來消除低頻雜訊，但因其為由金屬膜所捲成的電解電容，具有電感性，對高頻雜訊無法產生消除效果，因此再並聯一只 $0.1\ \mu\text{F}$ 的陶質電容器，以供高頻雜訊的旁路。

為了使每只主動元件的積體電路能不受雜訊干擾，故在每一只 IC 的電源端，就近均並聯連接一只 $0.1\ \mu\text{F}$ 的旁路電容器。

(二) 重置電路：當電源供應上升尚未穩定之前，89C51 單晶片不可以開始執行程式，故在此電源供應之電壓穩定之前 89C51 必須處在重置的狀態中，因此，由一 RC 充電電路來達到此項要求，由 RC 串聯的暫態現象可知



$$E_R = E e^{-t/RC} \dots\dots ①$$

$$\therefore E_C = E - E_R$$

$$\therefore E_C = E \left(1 - e^{-\frac{t}{RC}} \right) \dots\dots ②$$

E_C ：電容器上的電壓

E_R ：電阻器上的電壓

E ：電源電壓

e ：自然對數

t ：SW. ON 的時間

RC ：電路常數 $R \times C$

其中當

$t = 1RC$ 則 $e^{-1} = 0.368$ 故

$E_R = 0.368E$ ， $E_C = 0.632E$

$t = 2RC$ 則 $e^{-2} = 0.135$ 故

$$E_R = 0.135E, E_C = 0.865E$$

$$t = 3RC \text{ 則 } e^{-3} = 0.049 \text{ 故}$$

$$E_R = 0.049E, E_C = 0.951E$$

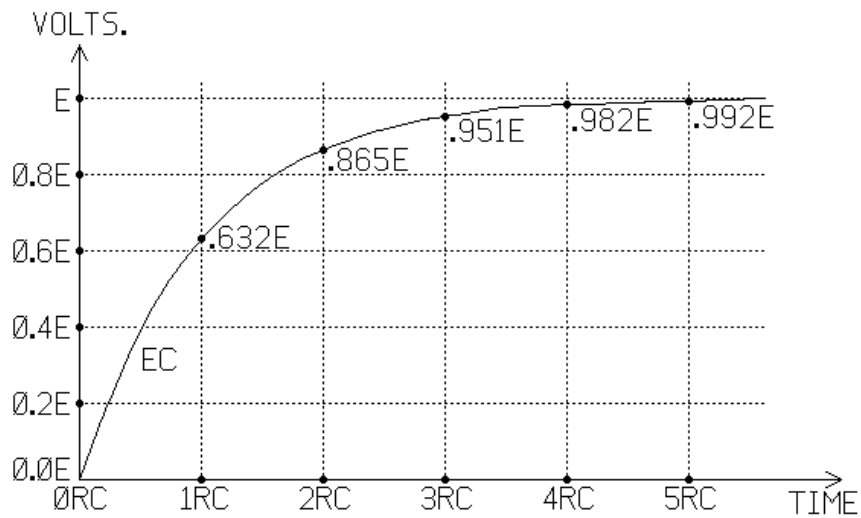
$$t = 4RC \text{ 則 } e^{-4} = 0.0183 \text{ 故}$$

$$E_R = 0.0183E, E_C = 0.982E$$

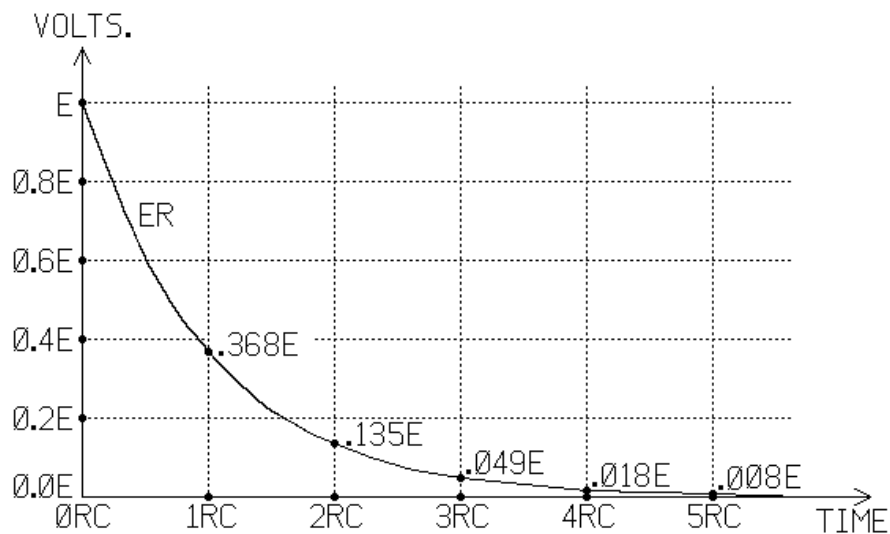
$$t = 5RC \text{ 則 } e^{-5} = 0.0067 \text{ 故}$$

$$E_R = 0.0067E, E_C = 0.993E$$

一般我們稱，充電時間達到 5 倍電路常數（或稱 RC 時間常數）即使電容器充滿，使 $E_C = E$ ， $E_R = 0$ ，如下圖所示：



電容器電壓暫態波形



電阻器電壓暫態波形

在 89C51 單晶片中，以 TTL 介面標準而言，達到 $V_{IH} = 2.0\text{ V}$ 以上就進入重置狀態，故須將重置腳接在 RC 電路的電阻對地兩端，使之在一供電時，重置腳得到高電壓而重置，此後再慢慢依 RC 電路常數下降。

在 89C51 晶片之重置端內部有一等效於 50K 到 300K 的電阻，為了不讓此偏移值大的電阻影響重置時間太大，因此採外加一只遠低於其最小值 50K 之電阻 10K 來完成電路常數的電阻值。如此可以確保不受其內部電阻因使用不同類的 89C51 產生極大的重置時間之變化。

因在 89C51 一旦進入重置後，依 TTL 介面電氣特性，要低於 0.8V 以下 (V_{IL}) 才視為 Low 而脫離重置。依公式① $E_r = Ee^{-t/RC}$ 中 $E_r = 0.8\text{ V}$ ， $E = 5\text{ V}$ ， $e = 2.71828$ ， $R = 10\text{ K}$ 為已知條件。其中因電源使用電源供應器，假設電源上升時間 0.1 秒，則因供給材料表中僅有 22 μF 電容，故使用 22 μF 的電容器為電路常數中的電容，如此，RC 時間常數（電路常數）約為 22 $\mu\text{F} \times 10\text{ K} = 0.22\text{ 秒}$ 。

由已知條件可得：

$$E_r = Ee^{-t/RC} \text{ 中 } 0.8\text{ V} = 5\text{ V} \times 2.71828^{-t / 0.22\text{s}}$$

$$\text{可得 } 0.16 = (2.71828)^{-t / 0.22\text{s}} \text{ 故}$$

$$\ln 0.16 = -t / 0.22\text{s}$$

$$\rightarrow -1.83258 = -t / 0.22\text{s}$$

$$\rightarrow t = 1.83258 \times 0.22\text{s} = 0.40317\text{ 秒}$$

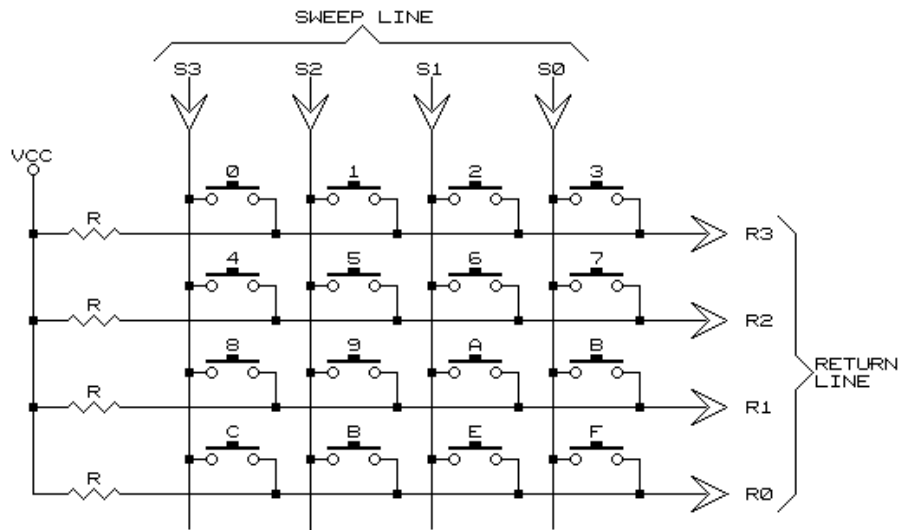
因此約 0.4 秒脫離重置，其值大於 0.1 秒，故符合需求。

事實上，因已知電源為 5V，而 0.8V 脫離重置，求得 $0.8\text{ V} \div 5\text{ V} = 0.16$ ，在 E_r 的電壓曲線圖上可得知 $E_r = 0.16E$ 大約是 1.8RC，而 $RC = 0.22\text{ 秒}$ ，故 $1.8RC = 0.396\text{ 秒}$ ，與 0.40317 秒差異不大。

在動作要求上有重置(RESET)鍵之要求，當按下重置鍵，要 89C51 進入重置狀態，故將一按鍵並接在重置電路的電容器兩端，即可在按下按鍵時，89C51 重置接腳得到重置所需之高電壓 5V，此時電容器中之電荷完全放電完畢，在按鍵放掉時電容器重新充電，須經 0.4 秒後才脫離重置狀態。

按鍵因非常脆弱，容易在將電容器電荷放電時的能量浪湧電流出現，而造成微小火花，以致使按鍵氧化而接觸不良，故通常應該在開關到電容器之任一接點先串一只 330 Ω 以下之電阻來阻止浪湧電流，但供給之材料表中沒有多餘的適用電阻，故省略掉，因非長期使用，省略掉此一阻尼電阻可以被接受，但在進行一般商品之設計時，這是不可省略的。

- (三) 時脈振盪電路：因材料表僅提供 11.0592MHz 的晶體振盪器及兩只 10Pf 的陶質電容，故僅能如此選用，依圖安裝，兩只 10Pf 用來減少振盪器之干擾，使振盪更穩定。
- (四) 鍵盤掃描電路：因為共要有四個按鍵，近年來，現場題目常要求要以矩陣掃描方式來建構鍵盤。故特別捨簡就繁，同樣使用四支 I/O 接腳，卻以矩陣掃描方式完成。其掃描的原理為：

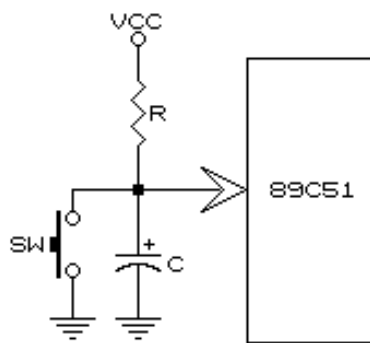


掃描線 S3、S2、S1、S0 為輸出，任何時間只有一個掃描線輸出 LOW，其餘均輸出 HIGH，每次的輸出狀態輸出後，即自做為輸入的返回線 R3、R2、R1、R0 讀取按鍵狀態。

1. 當讀取 R3~R0 的返回線全部是 HIGH 時，表示當時為 LOW 輸出的掃描線所接的四個按鍵，均無任何一個被按下。
例如：此時 S3~S0 為 0111，而 R3~R0 為 1111，表示按鍵 0、4、8、C 均沒被按下。
2. 當讀取 R3~R0 的返回線中有出現 LOW 時，則該返回線所對應的四個按鍵至少有一個按下，至於是那一個被按下，則看此時掃描線中那一條目前是 LOW，此均為 LOW 之交叉點的按鍵被按下。
例如：S3~S0 為 1101，而 R3~R0 為 1011，則 R2 返回線所對應的四個按鍵 4、5、6、7 中因 S1 為 LOW，故知“6”被按下。

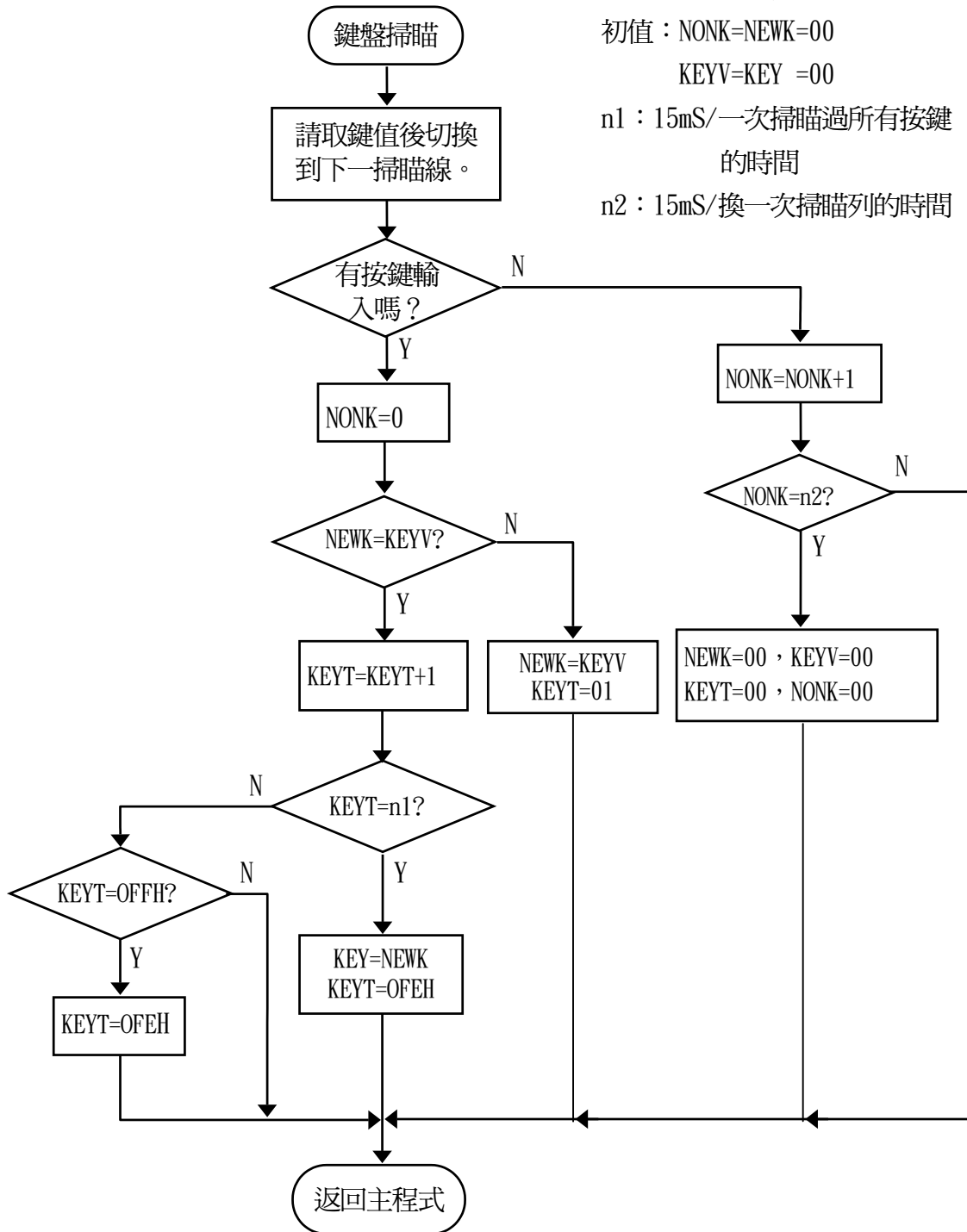
在任何按鍵被按下或放開時，均會有彈跳產生。彈跳就是在開關從開路情況下被按下到快要短路時會產生多次接觸與不接觸間的跳躍。當按鍵放開時也會出現同樣狀況。這種彈跳會使一次按鍵的動作被電路視為好幾次的按同一鍵。

由以上敘述可知，彈跳是需要消除的，一般彈跳的間隔不會大於 10ms，故 10ms 以內的接觸時間均被視為彈跳。消除彈跳有兩種方法，一為硬體彈跳消除，另一為軟體彈跳消除。硬體彈跳消除電路如下所示：

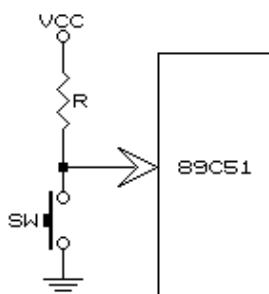


由一 RC 充電電路組成。平時 C 經 5 倍時間常數充滿電壓，在 SW 按下時，只要有第一次接觸，C 上之電壓即放電到低電位，在彈跳中，每次 SW 之彈開，C 上的電壓因必須經電阻來充電，因此需要有一段時間，設計上，只要控制 RC 時間常數，使電容器上的電壓充到被視為 HIGH 準位的時間大於 10ms 即可。

NONK：沒有按鍵的掃描次數
 NEWK：上一次按鍵值
 KEYV：此次掃描到的鍵值
 KEYT：同一按鍵被掃描次數
 KEY：已承認的按鍵值
 初值：NONK=NEWK=00
 KEYV=KEY =00
 n1：15mS/一次掃描過所有按鍵的時間
 n2：15mS/換一次掃描列的時間

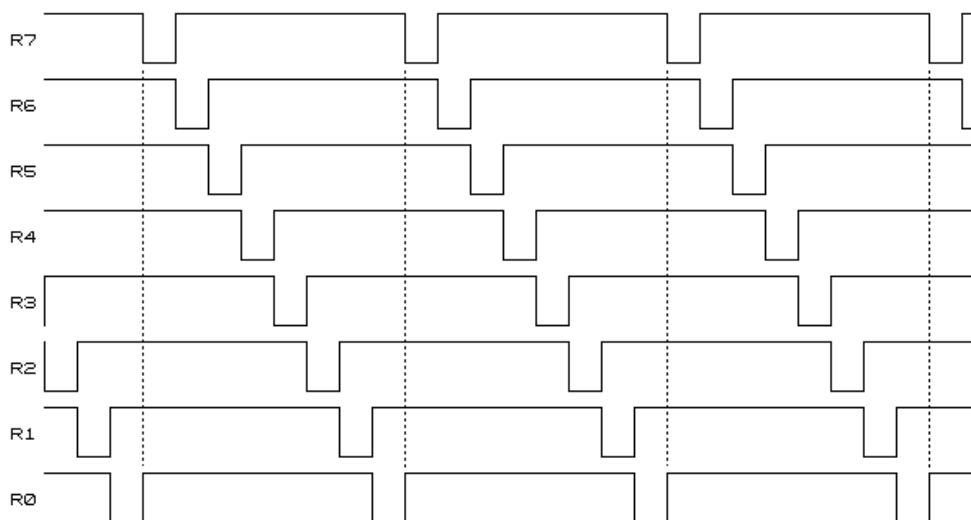


軟體的彈跳消除，則在讀取按鍵按下狀況持續 10ms 以上才認可該次按鍵，中途如有發現按鍵沒接觸好，則計時 10ms 的計時值立即歸零重新計時。按鍵放掉亦同。如此，則按鍵電路如下所示即可。



(五) 顯示驅動電路

1. 顯示器為 8×8 的點矩陣 LED，分有共陽及共陰兩種，其控制方式大同小異，學員只要學會一種，另一種即可輕易使用它。
2. 顯示器之接腳可分為“行”與“列”，接腳沒有依照順序，但幾乎每一家生產廠商其接腳安置均相同方式。
3. 本專題以行送資料，共八行，列送掃瞄線，任何時間每次只亮一列。
4. 因人的眼睛暫留時間是十六分之一秒，意即亮點只要每秒閃亮 16 次，即看不出它在閃爍。但這在是二分之一責任週期情形下而言，意即亮三十二分之一秒，暗三十二分之一秒，亮暗各佔一次閃爍時間的一半。本顯示器是每一列亮完要等另七列循環亮一次才再亮起，所以是八分之一責任週期。



因此：二分之一週期要每秒閃十六次，同理八分之一週期每秒就要閃六十四次。故每秒要掃描至少六十四次。

每一循環有八條掃描線要輪流，因此每切換一條掃描線的時間，應為：

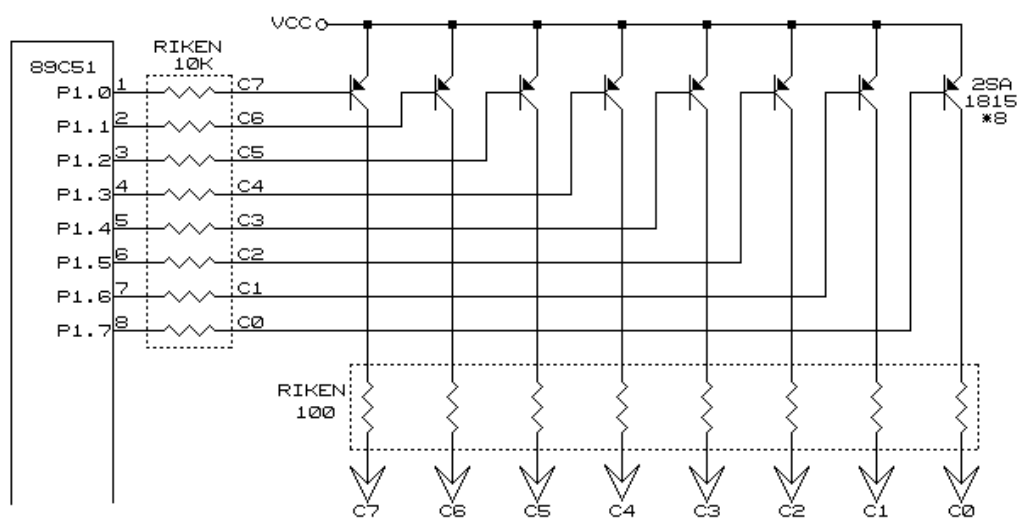
$$\frac{1}{64 \times 8} \text{秒} = \frac{1000\text{mS}}{512} = 1.953125\text{mS} \approx 2\text{mS}$$

5. 本專題設定由 89C51 PORT1 來送每一列顯示中的八位元資料（8 行）。因對資料而言，每次均只點亮一只 LED，而對掃描列而言，每次最多亮八只 LED，最少則沒有一顆亮，為確保亮度均勻，因此，LED 的限流電阻必須串在資料控制埠上。在材料表中最小的電阻只有 100Ω 的排阻，故選用此排阻在做限流電阻器。LED 長亮的耗電為 1V、10mA，但此 LED 是八分之一責任週期，故可以供應較大電流，以提高亮度，每只 LED 最大耗電電流約：

$$(5\text{V} - 1\text{V}) \div 100\Omega = 40\text{Ma}$$

為原長亮電流的四倍。

每一資料位元均經一只電晶體 2N3904 來驅動。事實上，在本電路結構上最好是使用 PNP 的電晶體，如下圖方式連接，以消除負回授作用，幸而此為數位電路，基極固定 0V 或 5V，因此不影響電路品質。但基極電阻不能太大。



電晶體的基極必須串接一電阻器以做偏壓控制，89C51 輸出 HIGH，表示要點點該 LED，此時最大輸出電流僅 $160\ \mu\text{A}$ （89C51 輸出埠除埠 0 以外每一 I/O 接腳僅能推動四個 TTL 的邏輯輸入閘），而 2N3904 電晶體的特性參數如下：

- (1) 25°C 時最大集極功率消耗350mW
 - (2) 最高工作頻率280MHz
 - (3) 最高 V_{cbo} 承受電壓30V
 - (4) 最高 V_{ceo} 承受電壓20V
 - (5) 最高 V_{ebo} 承受電壓 5V
 - (6) 最大集極電流 I_C 200mA
 - (7) 動態順向電流增益 h_{fe} 100
- （其餘無關的參數不再列出）

- (1) 當埠 0 輸出 LOW 時，電晶體截止，其集極電流為 0，故集極功率消耗為 $5V(V_{ce}) \times 0\text{mA}(I_C) = 0\text{W}$ 。

當埠 0 輸出 HIGH 時，電晶體導通，集極電流約 40mA，但 $V_{ec} = 0.2\text{V}$ ，故集極功率消耗為 $0.2\text{V} \times 40\text{mA} = 4\text{mW}$ 。

因電晶體只工作在截止及完全導通兩種狀況，故沒有其他功率消耗值，而 $4\text{mW} < 350\text{mW}$ 甚遠，故電晶體沒有承受不了功率之慮。

- (2) 可工作頻率高達 280MHz，遠高於資料切換頻率，故合用於本電路。
- (3) 電源才 5V，小於 V_{cbo} ， V_{ceo} ，而 V_{ebo} 有 R6 分壓，也小於 5V，故合於特性參數條件。
- (4) 在特性參數中沒標列直流放大倍數 β 值，但有動態順向電流增益 h_{fe} 值，其值略小於 β 值，故可使用 h_{fe} 值來代替 β 值做參考。所以 $\beta=100$ 。

已知最大 I_C 為 40mA，故最大 I_b 值為：

$$I_b = I_c \div \beta = 40\text{mA} \div 100 = 0.4\text{mA}$$

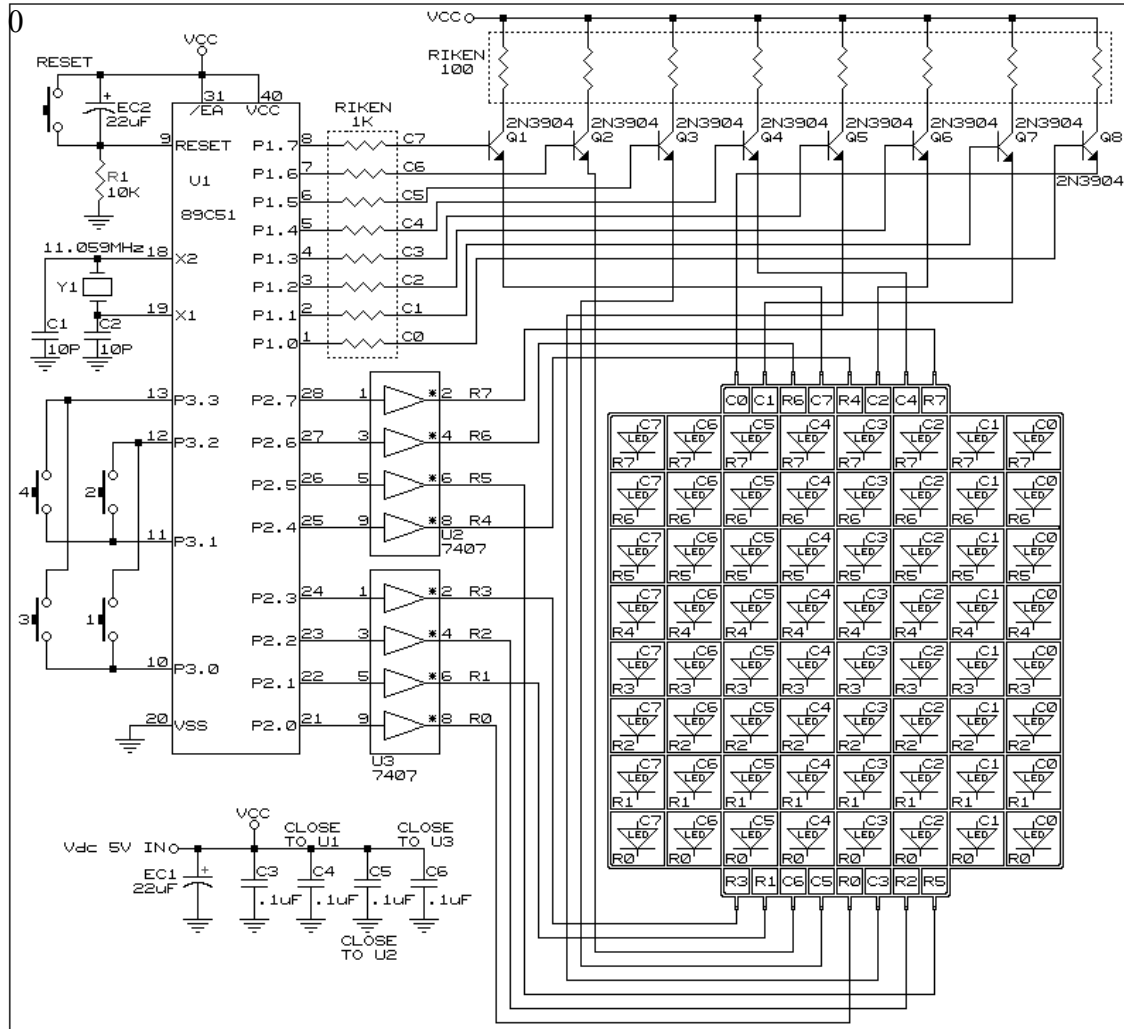
但 89C51 在 I/O 埠輸出 HIGH 時最大電流為 $160\ \mu\text{A}$ ，故選用這麼低的 β 值之電晶體，事實上並不恰當。但材料表只提供此編號電晶體，別無選擇，只好讓 89C51 I/O 埠輸出 HIGH 時的電壓下降一些（約 3.5V 左右），基極電阻只有 $1\text{K}\Omega$ 的排阻可以選用。

6. 本專題設定由 89C51 PORT2 來送列的下拉電源，經 7407 積體電路來驅動每一列，任何時間只有一列為 LOW，使得為 LOW 那一列的八只 LED 負端經 7407 的輸出端而接地，讓 LED 亮不亮由 PORT1 的輸出來決定。7407 為開路集極輸出，最高提升電壓可達 30V，這樣的電路結構，最符合所列材料供應限制。當然也可使用電晶體來控制「列」，用 7407 來控制 LED 正極的「行」。但這樣的結構，電路的功率消耗會大增。不切實際。
- (六) 程式記憶體的選用：因材料表無供應 ROM，而 89C51 單晶片的內部有 4K 位元組的可以電氣抹除的程式記憶體 ROM 區，因此選用內部程式記憶體，故/EA 接腳接高電位。4K 位元組的程式記憶體已足夠於安置本專題的功能要求之程式長度了。
- (七) 其他：在試題中，已無其他功能要求，也無多餘零件材料，故硬體電路至此完成。

學習評量一

試在 30 分鐘內，依據本專題材料表中所列之材料限制下，繪出可以符合本專題所需之功能要求，且符合電氣特性之電路圖。

學習評量一答案



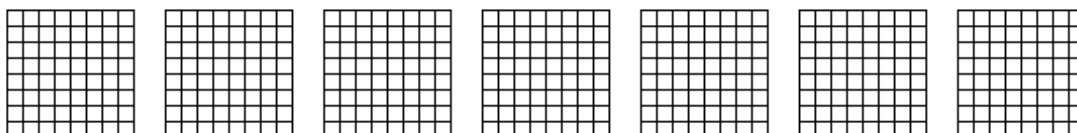
假如你的答案與上述之重點相似，請翻至下一頁。假如你的答案不與上述之重點相似，則請閱讀第 4 頁所列之參考書籍，或請翻至第 6 頁重新閱讀以便發現你的錯誤之處，並將第 25 頁上的錯誤改正，然後翻到下一頁。

本單元的第二個學習目標是

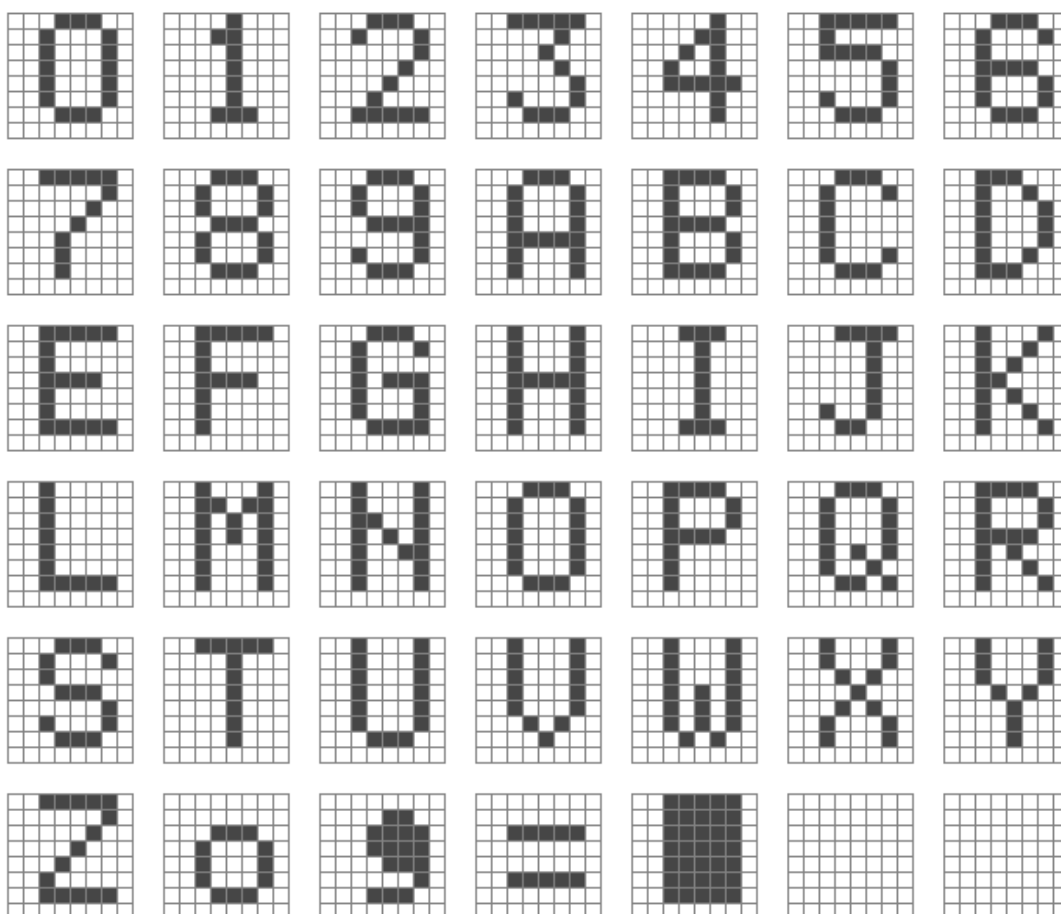
學員能在 5 分鐘之內，將所指定的三個 8×8 點矩陣字型轉譯成字形碼。

壹、依照試題的功能要求所述，可知道本顯示裝置要能夠顯示

- 一、0~9 的阿拉伯數字等字形。
- 二、A~Z 的英文字母等字形。
- 三、空白、句點、逗點、全亮等圖形。
- 四、首先，學員必須先在紙上作業，畫出 8×8 個小方格為一個單位字形區的表格，如下圖所示：

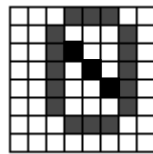


五、然後將字形依序塗入小方格內，如下圖所示：



六、再依硬體電路掃描方式，一是「行」輸出資料，「列」做掃描，或「列」輸出資料，「行」做掃描。來決定字形碼。

(一)「行」輸出資料，「列」做掃描(本專題控制方式)以字形「0」為例，依序自第七列(最上一列)到第0列其字形碼為：



列七：R7：00011100→1CH ；最上一列。

列六：R6：00100010→22H

列五：R5：00110010→32H

列四：R4：00101010→2AH

列三：R3：00100110→26H

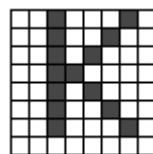
列二：R2：00100010→22H

列一：R1：00011100→1CH

列○：R0：00000000→00H ；最下一列。

故其字形碼為 1CH，22H，32H，2AH，26H，22H，1CH，00H

以字形「K」為例，依序自第七列到第0列，其字形碼為：



列七：R7：00100010→22H

列六：R6：00100100→24H

列五：R5：00101000→28H

列四：R4：00110000→30H

列三：R3：00101000→28H

列二：R2：00100100→24H

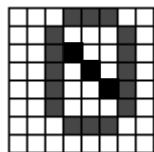
列一：R1：00100010→22H

列○：R0：00000000→00H

故其字形碼為 22H，24H，28H，30H，28H，24H，22H，00H

(二) 「列」為輸出資料，「行」做掃描者。

以字形「0」為例，依序自第七行（最左一行）到第 0 行，其字形碼分別為：



列七：C7：00000000→00H ；最左一行。

列六：C6：00000000→00H

列五：C5：01111100→7CH

列四：C4：10100010→A2H

列三：C3：10010010→92H

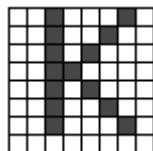
列二：C2：10001010→8AH

列一：C1：01111100→7CH

列○：C0：00000000→00H

故其字形碼為 00H，00H，7CH，A2H，92H，8AH，7CH，00H

以字形「K」為例，依序自第七行到第 0 行，字形碼分別為：



列七：C7：00000000→00H

列六：C6：00000000→00H

列五：C5：11111110→FEH

列四：C4：00010000→10H

列三：C3：00101000→28H

列二：C2：01000100→44H

列一：C1：10000010→82H

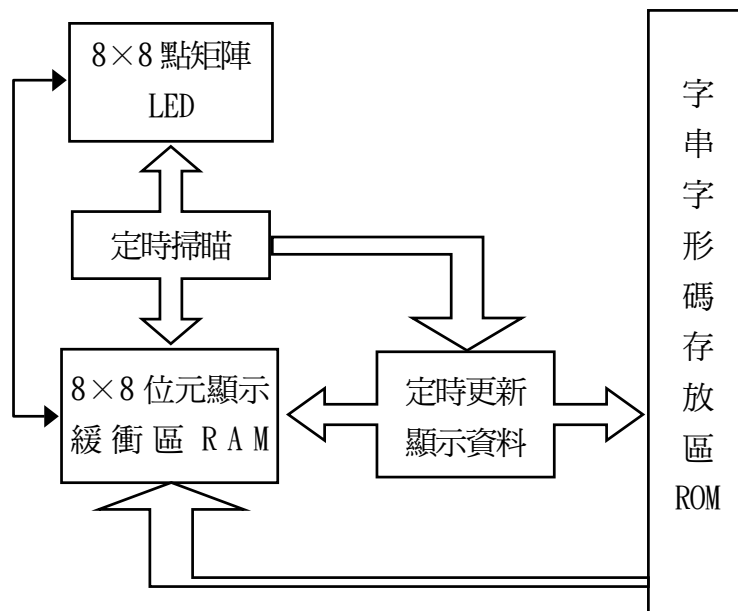
列○：C0：00000000→00H

故其字形碼為 00H，00H，FEH，10H，28H，44H，82H，00H

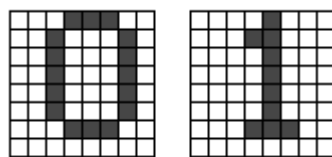
貳、字形緩衝區：

字串是一連串字形的串接，無論上移、下移、左移、右移，均有將字形碼依序按移動方向結合指標選取其中八個位元組來顯示，依所需的移動速度，在每顯示完 N 次的完整畫面後指標移動一個位置，即可產生畫面的移動，N 的大小就決定移動的速度。

- 一、 首先必須在 RAM 區建立一組八個位元組的顯示緩衝器，讓程式固定時間將本緩衝器的內容，依掃描規則掃描顯示到點矩陣 LED 上。
- 二、 將要顯示的字串之字形碼依資料結構存在程式記憶體中，使用一個指標在適當時機（畫面掃描了 N 次後）將移動的下一畫面（依移動的方向而定）之八位元字形，載入緩衝區中。

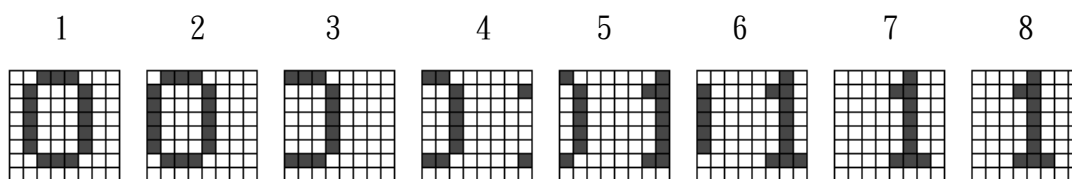


- 三、 在本專題的硬體電路中，以將字串左、右移動為困難度較高的部分，上下移動則純粹只要移動指標來將字形區往上或往下取八位元列緩衝區即可達成目的。但往左右移則不然。因掃描為列，資料為行，左、右移，則可能在畫面上出現某兩個字元的一部分，如下圖為「0、1」之向左移之可能出現畫面：



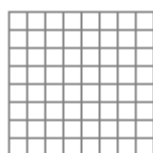
：已知字形

可能出現之左移字形畫面（由左至右依序出現）



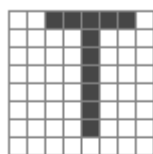
故須有一段字形處理之程式段來解決此一問題，以下描述其處理方法：

- (一) 假設字串為「T123456789」。
- (二) 假設字形碼被安置在 ROM 區的 0800H 為起點的位置，則該 ROM 區之字串字形碼分別如下：



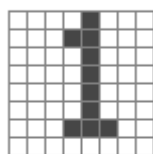
0800H

00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,
(加空白字元以做起始畫面)



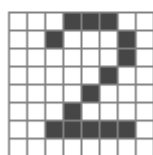
0808H

3EH, 08H, 08H, 08H, 08H, 08H, 08H, 00H



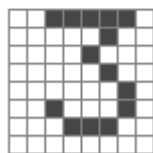
0810H

08H, 18H, 08H, 08H, 08H, 08H, 1CH, 00H



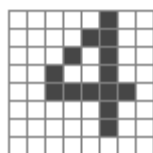
0818H

1CH, 22H, 02H, 04H, 08H, 10H, 3EH, 00H



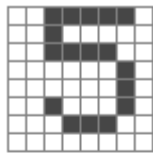
0820H

3EH, 04H, 08H, 04H, 02H, 22H, 1CH, 00H

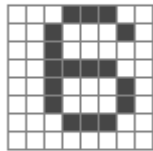


0828H

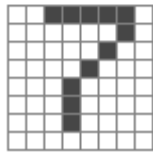
04H, 0CH, 14H, 24H, 3EH, 04H, 04H, 00H



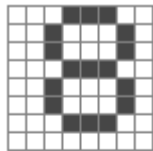
0830H 3EH, 20H, 3CH, 02H, 02H, 22H, 1CH, 00H



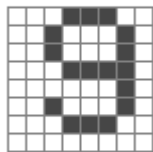
0238H 1CH, 22H, 20H, 3CH, 22H, 22H, 1CH, 00H



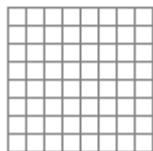
0240H 3EH, 02H, 04H, 08H, 10H, 10H, 10H, 00H



0248H 1CH, 22H, 22H, 1CH, 22H, 22H, 1CH, 00H



0250H 1CH, 22H, 22H, 1EH, 02H, 22H, 1CH, 00H



0258H 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,
(加一空白字元，使“9”可以全部移出)

(三) 假設以 30H 到 37H 為 RAM 區的顯示緩衝區。

(四) 其字形資料的運作如下所描述，即可達到字串向左移動的目的。雖然本試題例中的要求並無字串上、下、左、右移動，但很有可能應試時，出此試題要求，故在此說明之。

1. 在得知按下要字串往左移動的按鍵後，先執行下列字串左移的初始動作。

(1) 以暫存器 DPTR 做為字形區指標，指向程式記憶體的字串字形碼區起點 0800H，然後取前兩個字元的字形碼共 16 個位元組，分別存入顯示緩衝區起點 (30H) 起的 16 個元組的位址內。

```

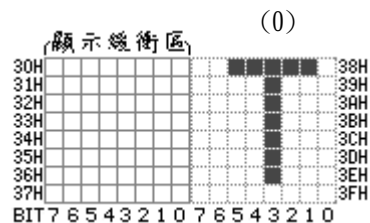
MOV    DPTR, #080H    ; 建議使用獨立暫存器庫
MOV    R0, #30H      ; R0 指向緩衝區起點
MOV    R1, #10H      ; 共要取 16 個位元組
GET1 : MOV    A, @DPTR    ; 取一個位元組
MOV    @R0, A        ; 存入緩衝區中
INC    DPTR          ; 指向下一個位元組
INC    R0            ; 指向下一個位元組
DJNZ   R1, GET1      ; 共要取 16 個位元組
; 到目前 DPTR 已經指向“1”的列 7 之字形碼位元組。

```

(2) 顯示掃描因要約每秒換掃描線 512 次，即約每秒要顯示 64 張完整的畫面，故固定約每 2ms 換一次掃描線。

假設字形左移的速度為每 0.2 秒左移一顯示行（1.6 秒移動一個字元），故掃描線每更換 100 次，則顯示緩衝區內容要更換一行。每一次的顯示緩衝區內容更換的方式說明暨圖示如下：

A. 共先載入兩個字元的字形碼到顯示及字形緩衝區。



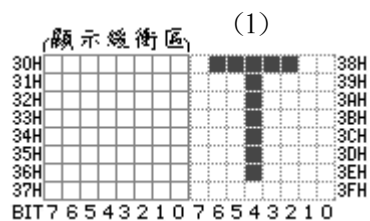
a. 先載入第一空白字元到顯示緩衝區 30H~37H 中。

b. 同時載入下一個字元到字形緩衝區 38H~3FH 中。

（已於(1) 項時完成）。

c. 目前顯示空白（顯示緩衝區內容）

B. 掃描線替換 100 次後（0.2 秒）所有緩衝區內容向左移一個位置。



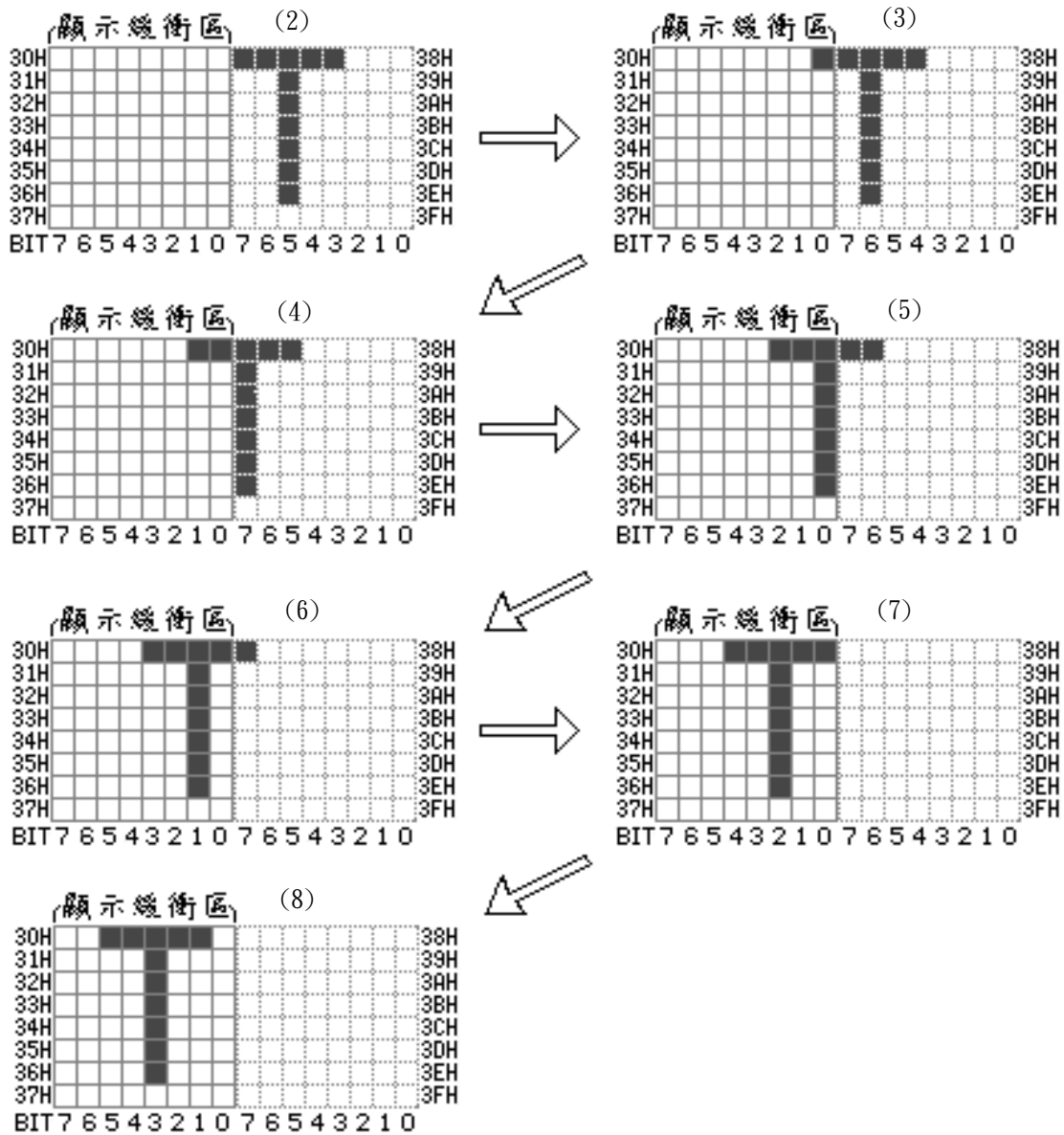
- a. 先將 38H 位元組串上 C 旗號左旋一個位元，再存回 38H 位址。
- b. 再將 30H 位元組串上 C 旗號左旋一個位元，再存回 30H 位址。
- c. 再將 39H 位元組串上 C 旗號左旋一個位元，再存回 39H 中。
- d. 再將 31H 位元組串上 C 旗號左旋一個位元。再存回 31H 中。
- e. 重複以上動作，但左旋的位元組一直往下，直到左旋完 37H 位元組，即可得到如上圖所示結果。

```

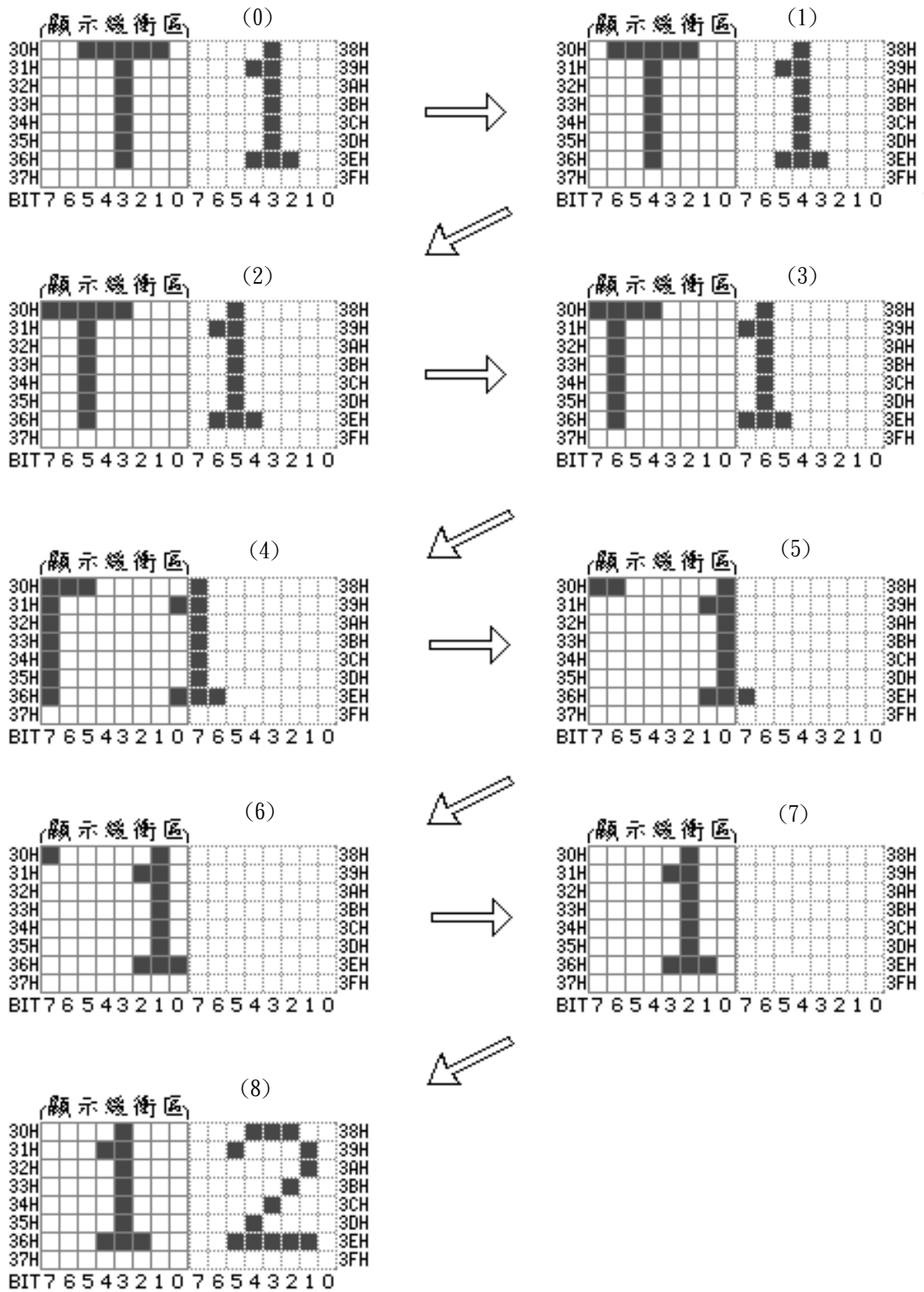
MOV    R0, #38H    ; R0 做指標
MOV    R1, #30H    ; R1 做指標
MOV    R2, #08H    ; R2 做計數器
MOV    A, @R0      ; 取資料
RLC    A           ; 左旋（串上 C 旗號）
LOOP:  MOV    @R0, A ; 存回原位
MOV    A, @R1      ; 取資料
RLC    A           ; 串上 C 做左旋動作
MOV    @R1, A      ; 存回原位
INC    R0          ; 指向下一個位元組
INC    R1          ; 指向下一個位元組
DJNZ   R2, LOOP    ; 共左旋八個回合才完畢。

```

- C. 再經約 0.2 秒的顯示掃描後，又再重複 B 項所有動作，直到重複 B 項動作八次，使“T”字形出現在顯示區之正確位置時，再取下一個字元“1”到字形緩衝區之中。如下列一系列圖形所示：



- 到左圖為止已經完整地左移一個字元了。再將字形區 ROM 中取下一個字元到字形緩衝區 (38H~3FH 中) 中。
- D. 下圖是在發覺已經完整地左移了一個字元後，再到字串中取下一個字元“1”存到 38H 到 3FH 字學形緩衝區中的情形。
 - E. 再重複 B 項動作八次 (每隔 0.2 秒一次) 又會將字形“T”左移掉，使“1”出現在顯示區正確位置。如下順序所示：

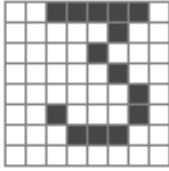


到“1”左移到顯示器之正確位置後，又必須將下一個字元“2”取到字形緩衝區中了。

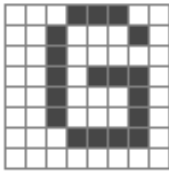
- F. 當取完最後一個字元「空白」後，就必須在空白移到顯示區正中央後又回頭取第一個字元“T”到字形緩衝區中，如此完成一組字串之左移動作。
- G. 如果要左移字串，只是取字之方向相反，顯示緩衝區設在 38H 到 3FH，而且 B 項的動作將左旋的動作改成右旋，並且先旋 30H，再旋 38H 等等之順序即可達到目的。
- H. 要上移或下移，只要自 ROM 區取資料到顯示緩衝區每次取八位元組，但每次的起點均加 1 以做上移或減 1 以做下移，再簡單不過了。

學習評量二

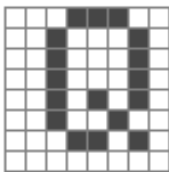
一、試在 5 分鐘內，將下列字形轉譯成字形碼，其中行為資料線列為掃瞄線。



列	7	6	5	4	3	2	1	0
碼								

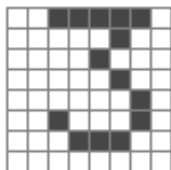


列	7	6	5	4	3	2	1	0
碼								

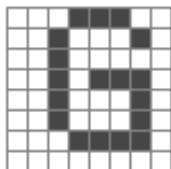


列	7	6	5	4	3	2	1	0
碼								

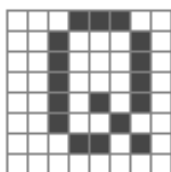
學習評量二答案



列	7	6	5	4	3	2	1	0
碼	3EH	04H	08H	04H	02H	22H	1CH	00H



列	7	6	5	4	3	2	1	0
碼	1CH	22H	20H	2EH	22H	22H	1EH	00H



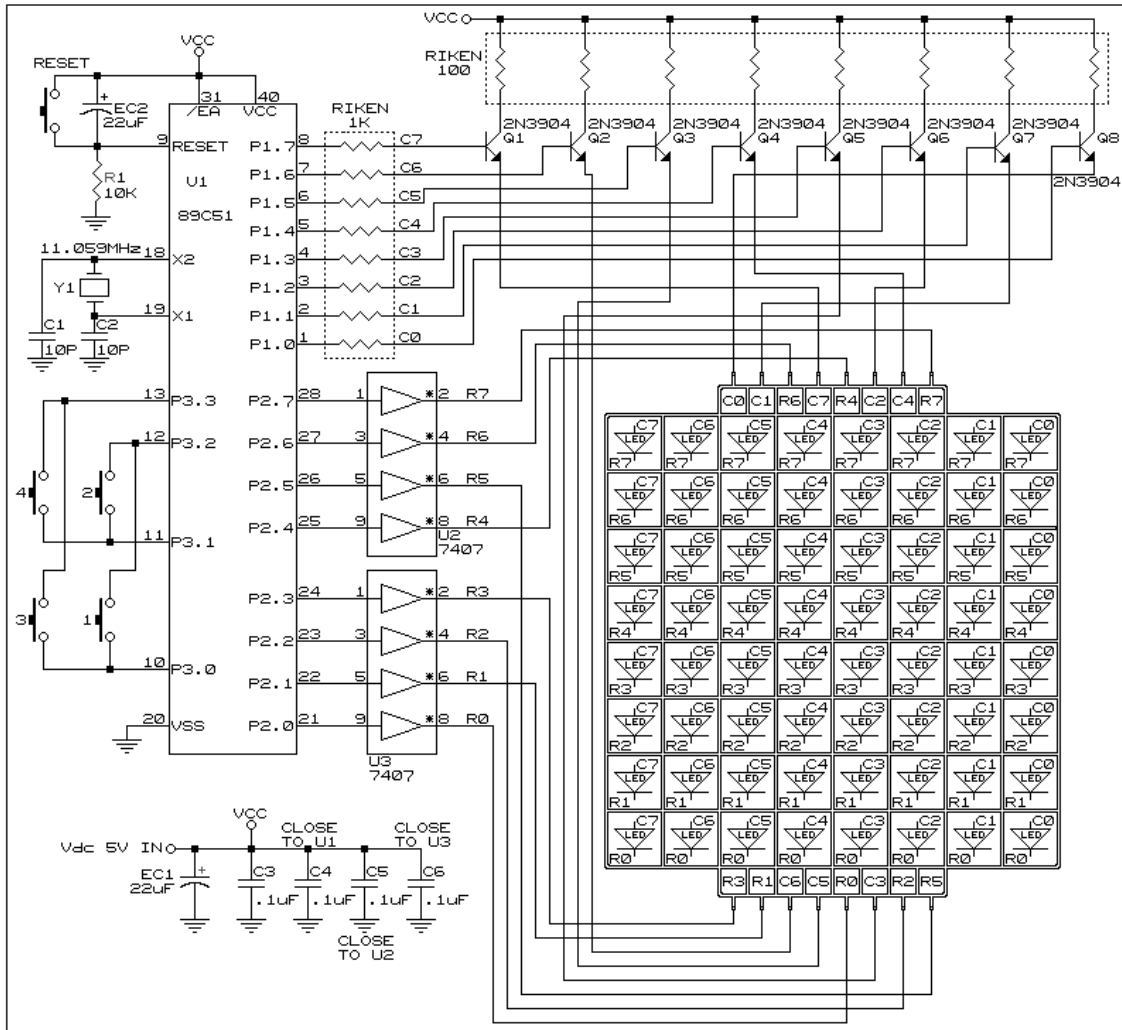
列	7	6	5	4	3	2	1	0
碼	1CH	22H	22H	22H	2AH	24H	1AH	00H

假如你的答案與上述之重點相似，請翻至下一頁。假如你的答案不與上述之重點相似，則請閱讀第 4 頁所列之參考書籍，或請翻至第 27 頁重新閱讀以便發現你的錯誤之處，並將第 39 頁上的錯誤改正，然後翻到下一頁。

本單元的第三個學習目標是

學員能在 30 分鐘內，在不參考任何資料下，繪出所給之矩陣式鍵盤電路的掃描程式流程圖。

壹、本專題的硬體電路參考圖如下圖所示：



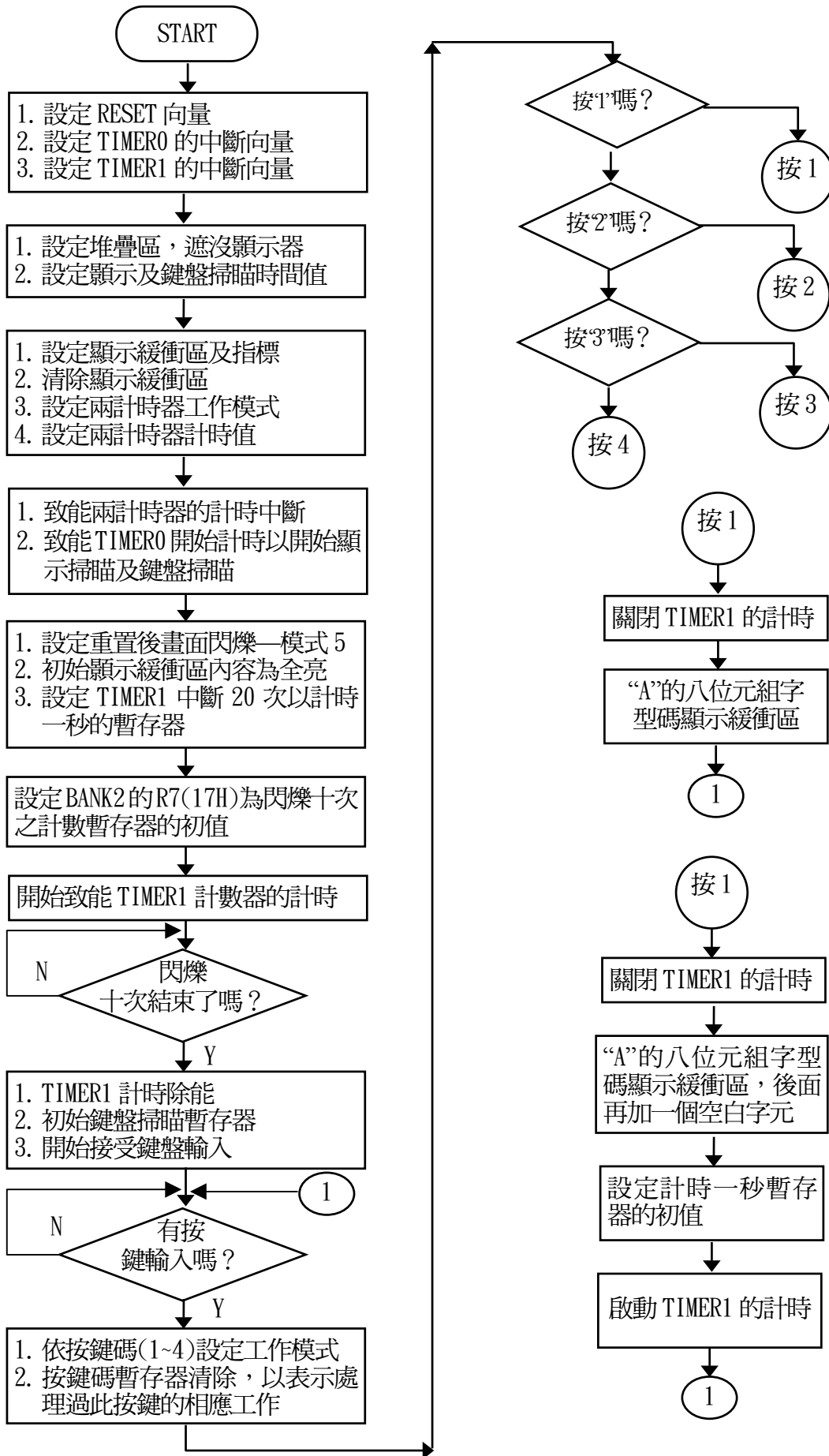
貳、本專題的功能要求如下：

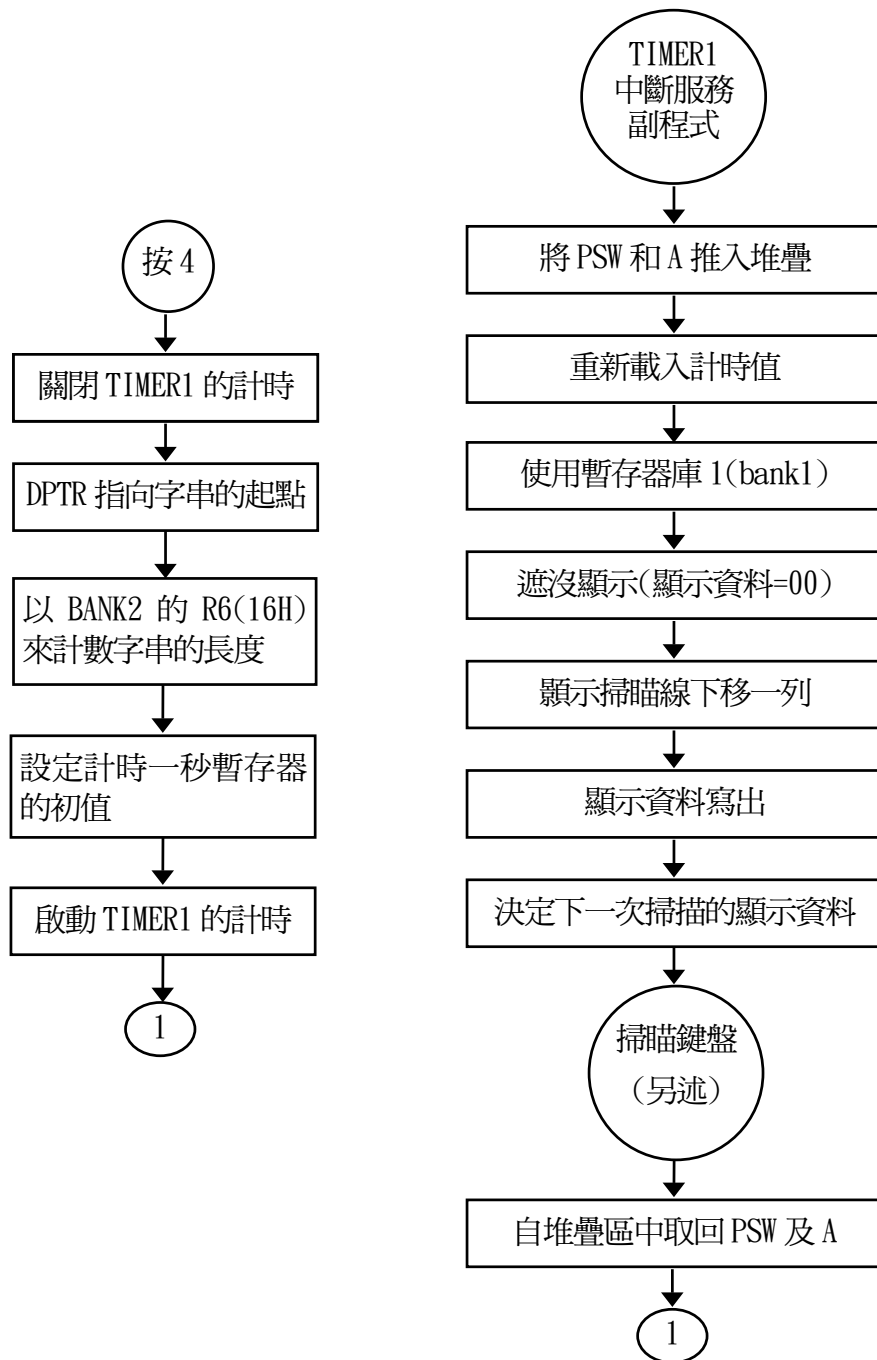
- 一、 電路在電源 ON 時，以及任意時刻按下重置(RESET)鍵時，會使全部 LED(8×8) 同時 ON、OFF 閃爍，閃爍間隔一秒，並且閃爍五次（亮、暗各五次）。
- 二、 在按下按鍵“1”時，畫面固定顯示英文字母“A”。
- 三、 在按下按鍵“2”時，畫面出現英文字母“A”，然後向左移動，每間隔一秒向左移動一行，直到八行全部移出之後，再由右方向左移入，而成循環顯示。
- 四、 在按下按鍵“3”時，畫面出現英文字母“A”，然後向上移動，每間隔一秒向上移動一列，直到八列全部移出之後，再由下方移入，而成循環顯示。
- 五、 在按下按鍵“4”時，畫面依序每間隔一秒顯示一個字元，而顯示字串“T123456789”，含 9 後面之空白字之共 11 個字元，循環顯示。

參、程式流程：

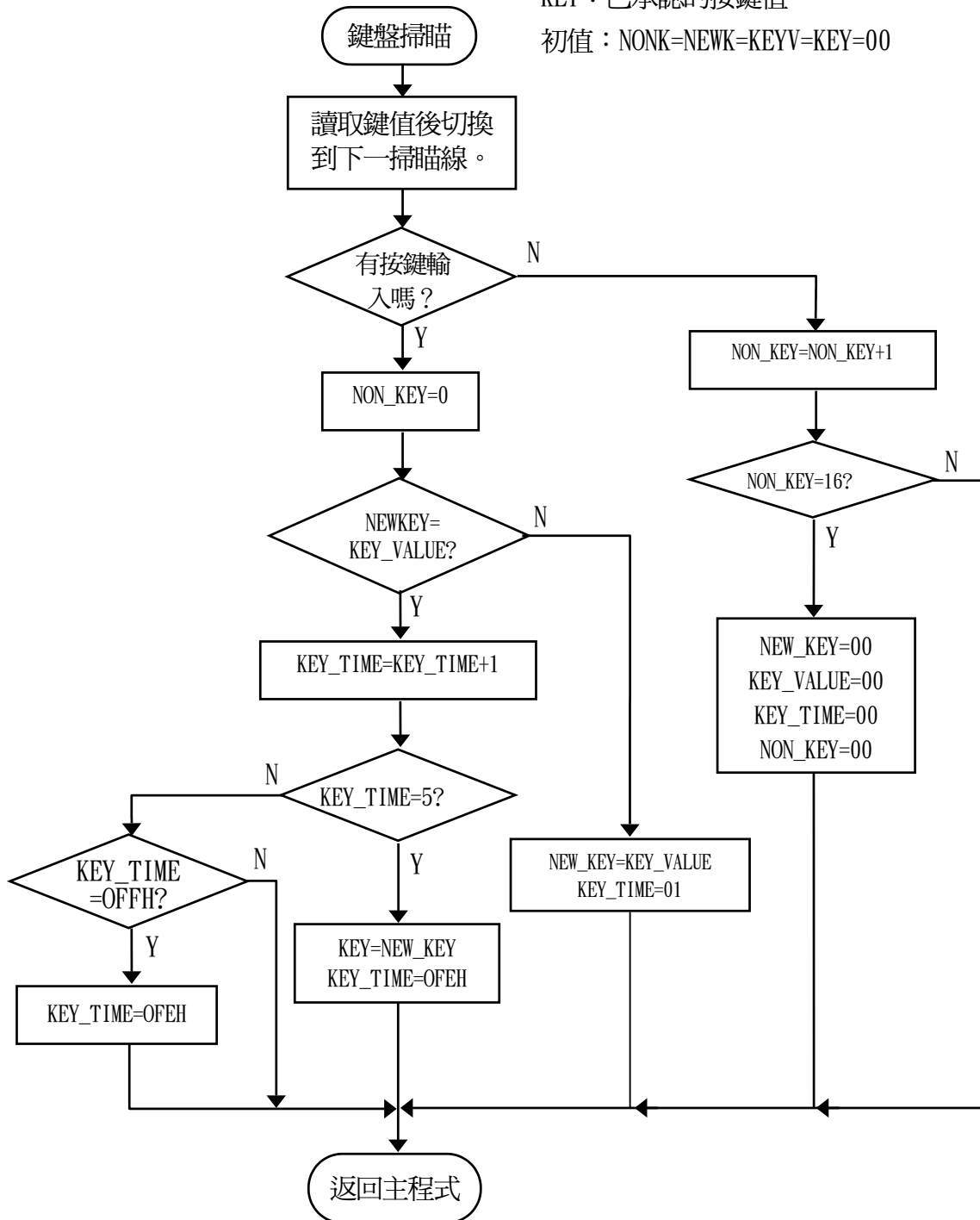
要達到以上所述之功能，有千百種不同的程式結構，但結果都相同，故本程式例僅供參考使用，當然它一定是可以正確滿足功能要求的。

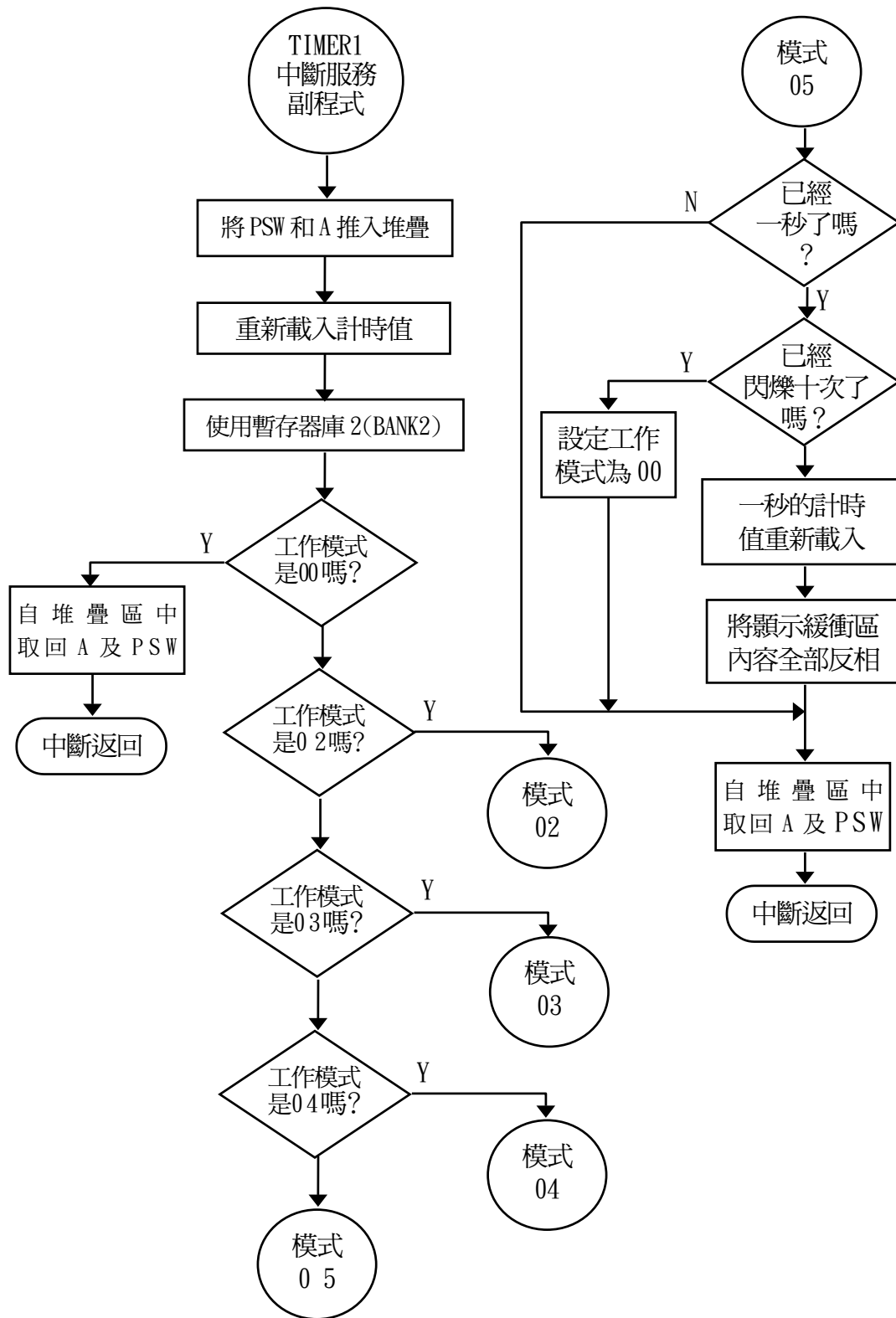
其程式流程圖如下：

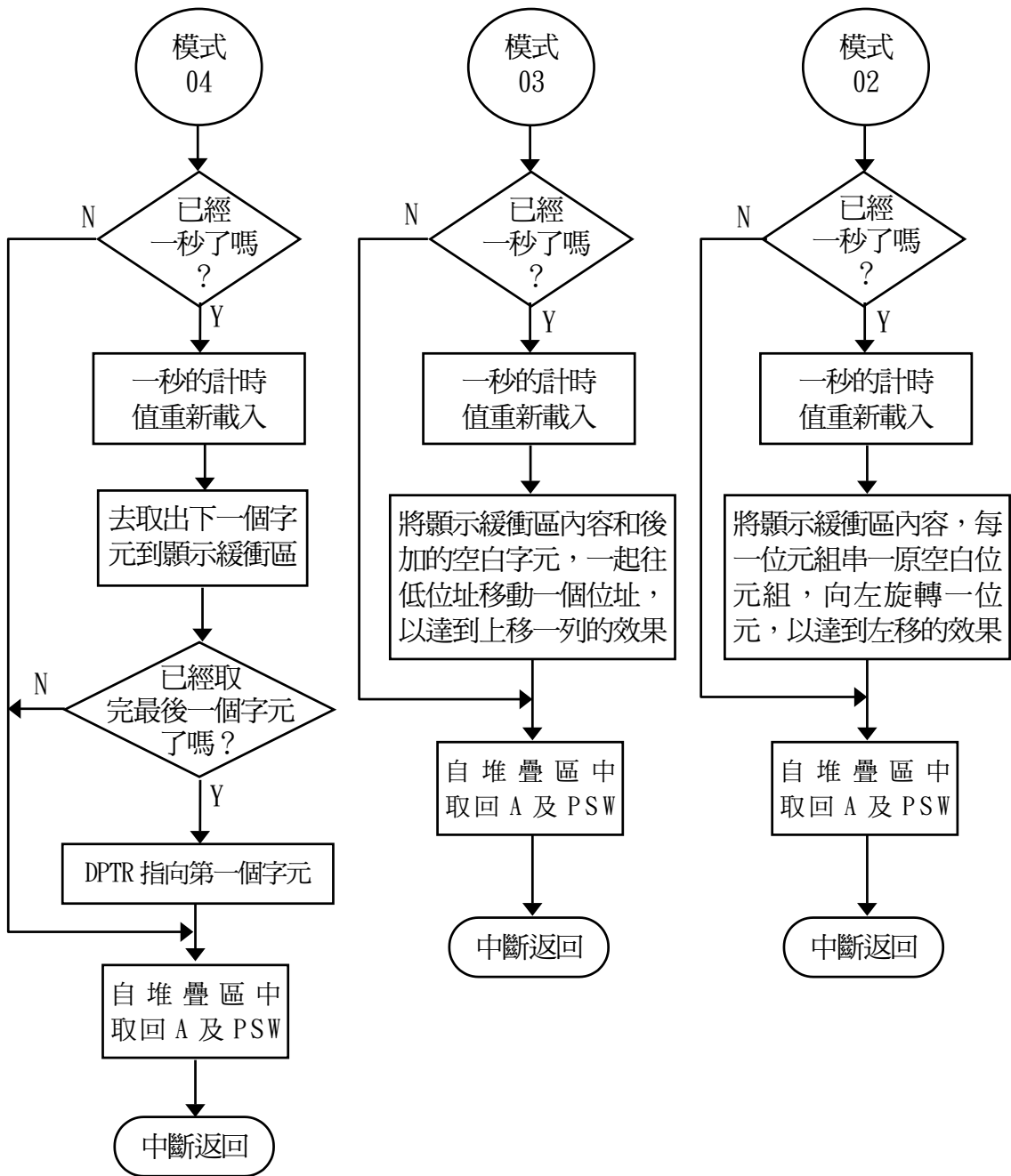




NONK_KEY：沒有按鍵的掃描次數
 NEWK_KEY：上一次按鍵值
 KEYV_VALUE：此次掃描到的鍵值
 KEYT_TIME：同一按鍵被掃描次數
 KEY：已承認的按鍵值
 初值：NONK=NEWK=KEYV=KEY=00







肆、程式本體

```

; ***** P M C 0 5 0 7 專 題 程 式 *****
COUNT0 EQU 30H ;計時計數器，記錄TIMER1時基的計時
COUNT1 EQU 31H ;備份COUNT0的值，讓COUNT0可以遞減改變
POINTER EQU 32H ;顯示掃描的資料位元組指標
WORK_MODE EQU 33H ;標示工模式
KEY EQU 34H ;記錄按鍵值
KEY_TIME EQU 35H ;做鍵盤的同一按鍵計次以消除彈跳用
KEY_VALUE EQU 36H ;記錄此次掃描的鍵值
NEW_KEY EQU 37H ;記錄上一次掃描的鍵值
NON_KEY EQU 38H ;記錄沒有按鍵的次數以消除放掉按鍵的彈跳

; ===== 程式起點 =====
ORG 0000 ;開機或重置的向量位址在0000H
JMP START ;跳到主程式的起點
ORG 0BH ;0號計時器的計數中斷向量位址在000BH
JMP TIMER0 ;跳到0號計時器的計數中斷服務副程式的起點
ORG 1BH ;1號計時器的計數中斷向量位址在001BH
JMP TIMER1 ;跳到1號計時器的計數中斷服務副程式的起點
ORG 30H ;主程式的起點可以自0030H開始
START: MOV SP, #5FH ;以60H到7FH共32個位元組當作堆疊區
MOV P1, #00 ;遮沒顯示
MOV P2, #0FFH ;先假設掃描到最後一列ROW 0
MOV P3, #0F7H ;埠3做按鍵輸入，先掃描P3.3的4和3鍵
MOV POINTER, #40H ;顯示緩衝器指標指向緩衝器起點
MOV A, #00 ;清除顯示緩衝區及顯示備分運作區
MOV R0, #40H ;緩衝區起點在40H

MOV R1, #10H ;共要清除16個位元組
STA1: MOV @R0, A
INC R0
DJNZ R1, STA1
MOV TMD, #11H ;兩計時器均十六位元計時
MOV TLO, #<(65536-2000)

```

```

MOV     TH0, #>(65536-2000)      ; 0 號計時器每 2mS 計數中斷一次
MOV     TL1, #<(65536-50000)
MOV     TH1, #>(65536-50000)    ; 1 號計時器每 50mS 計數中斷一次
MOV     IE, #8AH                 ; 致能 TIMER0 和 TIMER1 的中斷致能旗號
SETB   TR0                       ; 致能計時器 TIMER0 的計時

; ===== 讓顯示器全亮一秒全暗一秒共閃爍五次 =====
MOV     WORK_MODE, #05           ; 標示為第五種工作模式，讓 TIMER1 處理
MOV     A, #OFFH                 ; 全亮的顯示碼存入顯示緩衝區中
MOV     R0, #40H                 ; R0 指向緩衝區起點
MOV     R1, #08                  ; 共要寫入顯示緩衝區八個位元組
STA2 :  MOV     @R0, A
        INC     R0
        DJNZ   R1, STA2
MOV     COUNT0, #20              ; 每一秒鐘顯示緩衝區內容更換一次
MOV     COUNT1, #20              ; 永遠記著計時的初值
MOV     17H, #10                 ; TIMER1 使用 2 號暫存器庫，R7 做計數器閃爍次數
SETB   TR1                       ; 啟動 TIMER1 的計時
STA3 :  MVO    A, WORK_MODE       ; 查看是否已經閃爍結束
        JNZ   STA3                ; 閃爍結束 TIMER1 會清除 WORK_MODE
        CLR   TR1                  ; 關閉 TIMER1 的計時

; ===== 開始查看鍵盤 =====
STA4 :  MOV     KEY, #00           ; 清除按鍵
        MOV   KEY_TIME, #00        ; 清除按鍵計次
        MOV   KEY_VALUE, #00      ; 清除原鍵值
        MOV   NEW_KEY, #00        ; 清除新的按鍵碼
        MOV   NON_KEY, #00        ; 清除沒有掃瞄到按鍵的計次
STA5 :  MOV     A, KEY             ; 查看有無按鍵被掃瞄確定
        JZ    STA5                ; 沒有按鍵等到有按鍵
        MOV   WORK_MODE, A        ; 依照按鍵設定工作模式
        MOV   KEY, #00            ; 清除按鍵值，表示已經處理過
        CJNE A, #01, STA6         ; 查看是否按“1”鍵
        JMP   STA10               ; 跳去處理按“1”鍵的工作

```

```

STA6 :   CJNE   A, #02, STA7           ; 查看是否按“2”鍵
         JMP    STA20                   ; 跳去處理按“2”鍵的工作
STA7 :   CJNE   A, #03, STA8           ; 查看是否按“3”鍵
         JMP    STA20                   ; 跳去處理按“3”鍵和“2”鍵的初始工作相同
STA8 :   JMP    STA40                   ; 有按鍵，不是以上的鍵，那一定是按“4”
; =====處理按“1”鍵的工作，顯示英文字母“A”=====
STA10 :  CLR    TR1                     ; 計時器 TIMER1 停止計時
         MOV    DPTR, #TABLE_A         ; 以 DPTR 指向 A 的字型區
         MOV    R1, #08                 ; 取八個位元組的字型碼到顯示緩衝區中
         MOV    R0, #40H                ; R0 指向緩衝區起點
STA11 :  MOV    A, #00                  ; 沒有 MOV A, @DPTR 指令，所以 A 要為 00
         MOVC   A, @A+DPTR              ; 取字型碼
         MOV    @R0, A                  ; 字型碼存入顯示緩衝區
         INC    DPTR                    ; 來源指標加一已指向下一個位元組
         INC    R0                       ; 目的地指標加一已指向下一個存放的位址
         DJNZ   R1, STA11               ; 共要填入八位元組
         JMP    STA4                    ; 處理完畢，不必使用 TIMER1 故 TIMER1 維持不計時
; =====處理按“2”和按“3”鍵的工作，讓英文字母“A”左移或上移=====
STA20 :  CLR    TR1                     ; 計時器 TIMER1 暫時停止計時
         MOV    DPTR, #TABLE_A         ; 以 DPTR 指向 A 的字型區
         MOV    R1, #08                 ; 取八個位元組的字型碼到顯示緩衝區中
         MOV    R0, #40H                ; R0 指向緩衝區起點
STA21 :  MOV    A, #00                  ; 沒有 MOV A, @DPTR 指令，所以 A 要為 00
         MOVC   A, @A+DPTR              ; 取字型碼
         MOV    @R0, A                  ; 字型碼存入顯示緩衝區
         INC    DPTR                    ; 來源指標加一已指向下一個位元組
         INC    R0                       ; 目的地指標加一已指向下一個存放的位址
         DJNZ   R1, STA21               ; 共要填入八位元組
         MOV    R1, #08H                ; 要有八位元組的空白參予運作
         MOV    R0, #48H                ; 此運作區在 48H 到 4FH 間(其實本指令可以省)
         MOV    A, #00                  ; 清除此運作區
STA22 :  MOV    @R0, A
         INC    R0

```

```

DJNZ    R1, STA22
MOV     COUNT0, #20      ; 每一秒鐘顯示緩衝區內容左移一個位元
MOV     COUNT1, #20      ; 永遠記著計時的初值
SETB    TR1              ; 計時器 TIMER1 開始計時
JMP     STA4              ; 處理完畢，回去等到有按鍵輸入才改變

```

; =====處理按“4”鍵的工作，顯示身分證字號“T123456789”，每秒換一個字=====

```

STA40 :  CLR    TR1          ; 計時器 TIMER1 暫時停止計時
        MOV    DPTR, #STRING ; DPTR 指向字串起點
        MOV    16H, #10      ; 字串上一個空白字元共 11 個字元，已經取“T”
        ; 剩 10 個字元，以暫存器庫 2 的 R6 計數(位址為 16H)
        MOV    R0, #40H      ; R0 指向緩衝區起點
        MOV    R1, #08       ; 共要更新八個位元組
STA41 :  MOV    A, #00        ; 沒有 MOV A, @DPTR 指令，所以 A 要為 00
        MOVC   A, @A+DPTR    ; 取字型碼
        MOV    @R0, A        ; 字型碼存入顯示緩衝區
        INC    DPTR          ; 來源指標加一已指向下一個位元組
        INC    R0            ; 目的地指標加一已指向下一個存放的位址
        DJNZ   R1, STA41     ; 共要填入八位元組
        MOV    COUNT0, #20   ; 每一秒鐘顯示緩衝區內容左移一個位元
        MOV    COUNT1, #20   ; 永遠記著計時的初值
        SETB   TR1           ; 計時器 TIMER1 開始計時
        JMP    STA4          ; 處理完畢，回去等到有按鍵輸入才改變

```

; =====TIMER0 的中斷服務副程式=====

```

TIMER0  PUSH   PSW          ; 一進入中斷，要先將程式狀態語句推入堆疊
        PUSH   A            ; 在中斷服務副程式中會使用到 A，故先推入堆疊
        MOV    TL0, #<(65536-2000) ; 重新載入計時值
        MOV    TH0, #>(65536-2000)
        SETB   RS0          ; 使用暫存器庫 1
        MOV    P1, #00H     ; 遮沒顯示，以防止鬼影出現
        MOV    A, P2        ; 取出掃描列內容以向右旋轉來改到下一掃描列
        RR     A

```

```

MOV     P2, A
MOV     R0, POINTER
MOV     A, @R0           ; 取顯示資料
MOV     P1, A           ; 顯示出來
INC     R0               ; 指向下一列掃描的顯示資料位址
CJNE   R0, #48H, TIME1 ; 查看是否指向已超出顯示緩衝區
MOV     R0, #40H        ; 超出顯示緩衝區，指回起點
TIME1 : MOV     POINTER, R0
; ===== 掃描鍵盤 =====
MOV     A, P3           ; 讀取鍵盤
MOV     R2, A           ; 備份到 R2 中
XRL    A, #0CH         ; 將位元 3、2 反相，以指向下一鍵盤掃描列
ORL    A, #03          ; 位元 1、0 為掃描返回線，輸出 1 以做輸入並提升
MOV     P3, A           ; 將下一掃描線的切換確定後設定之
MOV     A, R2           ; 取回原讀入的鍵值
ANL    A, #03          ; 查看有無按鍵輸入，保留掃描返回線狀態
JZ     FINISH          ; 如果按下兩鍵則當做此次掃描無效
CJNE   A, #03, KEYIN1 ; 查看是否沒有按鍵
; ===== 沒有按按鍵的處理 =====
INC     NON_KEY         ; 沒有按鍵輸入的計次加一
MOV     A, NON_KEY     ; 查看無按鍵輸入計次是否已經為第 10 次了
CJNE   A, #10, FINISH ; 一開始為第一次，到第 10 次約 18mS 彈跳消除時間
MOV     KEY_TIME, #00 ; 清除按鍵計次
MOV     KEY_VALUE, #00 ; 清除原鍵值

```

```

MOV    NON_KEY, #00          ;清除新的按鍵碼
MOV    NON_KEY, #00          ;清除沒有掃描到按鍵的計次
JMP    FINISH

```

; ===== 一定有按鍵的處理 =====

```

JB     A. 0, KEYIN3          ;是否沒按“3”或“4”鍵？如果沒按，則位元 0 為 1
JB     P3. 2, KEYIN2         ;是按“3”或“4”，現在的 P3. 2=1 則原來 0，是按 4
MOV    KEY_VALUE, #3         ;此次掃描的鍵值為“3”
JMP    KEYIN5
MOV    KEY_VALUE, #4         ;此次掃描的鍵值為“4”
JMP    KEYIN5
JB     P3. 2, KEYIN4         ;A. 0=1，按 1 或 2，現在的 P3. 2=1 則原為 0，是按 2
MOV    KEY_VALUE, #1         ;此次掃描的鍵值為“1”
JMP    KEYIN5
MOV    KEY_VALUE, #2         ;此次掃描的鍵值為“2”
MOV    NON_KEY, #00          ;清除沒有掃描到按鍵的計次
MOV    A, NEW_KEY            ;取出上一次有按鍵輸入的按鍵值到 A 中
CJNE   A, KEY_VALUE, KEYIN6   ;查看是否與上一次按鍵相同
INC    KEY_TIME              ;同一按鍵計次加一
MOV    A, KEY_TIME           ;查看是否同一按鍵已經掃描五次了
CJNE   A, #05, KEYIN7         ;5 次約 18mS 的彈跳消除時間
MOV    A, KEY_TIME           ;按鍵成立，存入 KEY 中，以承認該按鍵
MOV    KEY, A
MOV    KEY_TIME, #0FEH       ;標示此按鍵已經承認過了
JMP    FINISH
MOV    KEY_TIME, #01         ;新按鍵出現，計次一
MOV    A, KEY_VALUE          ;備份新按鍵值
MOV    NEW_KEY, A
JMP    FINISH
CJNE   A, #OFFH, FINISH      ;查看是否本按鍵已經被承認過
DEC    KEY_TIME              ;回覆標示此按鍵已經承認過了的標示值
POP    A
POP    PSW

```



```

RETI
; ===== T I M E R 1 的中斷服務副程式 =====
PUSH   PSW           ; 一進入中斷，要先將程式狀態語句推入堆疊
PUSH   A             ; 中斷服務副程式中，會使用到 A，故先推入堆疊
MOV    TL1, #<(65536-50000) ; 重新載入計時值
MOV    TH1, #>(65536-50000)
SETB   RS1           ; 使用 2 號暫存器庫
MOV    A, WORK_MODE ; 查看是工作在哪一個工作模式中
CJNE   A, #00, TIMER2
POP    A
POP    PSW
RETI
CJNE   A, #02, TIMER3 ; 不可能有模式 1，因模式 1 不使用本計時器
JMP    MODE2
CJNE   A, #03, TIMER4
JMP    MODE3
CJNE   A, #04, TIMER5 ; 都不是以上工作模式，則一定是工作模式 5
JMP    MODE4
; ===== 重置後的工作——全亮、全暗交替閃爍各一秒共五次的處理 =====
DEC    COUNT0        ; 查看是否已經一秒了
MOV    A, COUNT0
INZ    MODE53         ; 尚未到一秒，返回主程式
DJNZ   R7, MODE51     ; 查看是否已經閃爍五次了
MOV    WORK_MODE, #00 ; 標示工作模式 1 已經做完
JMP    MODE53
MOV    A, COUNT1      ; 取出計時初值
MOV    COUNT1, A      ; 存回計時器中
MOV    R0, #40H       ; R0 指向顯示緩衝區的起點
MOV    R1, #08        ; 共要反相八個位元組的顯示資料
MOV    A, #OFFH       ; 將顯示緩衝區的內容全部反相
XRL   A, @R0          ; 以互斥或運作來達到反相效果
MOV    @R0, A         ; 反相後存回
INC    R0             ; 指向下一個位址

```

```

DJNZ  R1, MODE52
POP    A
POP    PSW
RETI

```

; =====工作模式 4 的處理，顯示身分證字號“T123456789”，每秒換一個字=====

```

DEC    COUNT0          ; 查看是否已經一秒了
MOV    A, COUNT0
JNZ    MODE42          ; 尚未到一秒，返回主程式
MOV    A, COUNT1       ; 取出計時初值
MOV    COUNT0, A       ; 存回計時器中，以重新計時一秒

```

; =====字型依序顯示“T123456789”，每秒換一個字，串一空白字元後又循環=====

```

MOV    R0, #40H        ; R0 指向字型區起點
MOV    R1, #08         ; 共要更新八位元組
MOV    A, #00          ; 沒有 MOV A, @DPTR 指令，所以 A 要為 00
MOVC   A, @A+DPTR      ; 取字型碼
MOV    @R0, A          ; 字型碼存入顯示緩衝區
INC    DPTR            ; 來源指標加一已指向下一個位元組
INC    R0              ; 目的地指標加一已指向下一個存放的位址
DJNZ   R1, STA41       ; 共要填入八位元組
DJNZ   R6, STA42       ; 是否已經一個循環結束了。
MOV    R6, #11         ; 字元計數還原
MOV    DPTR, #STRING   ; 字元指標指回“T”
POP    A
POP    PSW
RETI

```

; =====工作模式 3 的處理，讓英文字母“A”上移=====

```

DEC    COUNT0          ; 查看是否已經一秒了
MOV    A, COUNT0
JNZ    MODE32          ; 尚未到一秒，返回主程式
MOV    A, COUNT1       ; 取出計時初值
MOV    COUNT0, A       ; 存回計時器中，以重新計時一秒

```

; =====字型上移一個位元，字型“A”全部移出之後又從下邊移入的循環=====

```

MOV    R0, #40H        ; R0 指向字型區起點

```

```

MOV    R1, #41H           ; R1 指向字型區第二列
MOV    R2, #10H          ; 共要旋十六列的資料，以旋轉代替上移
MOV    A, @R0             ; 先取出最上一列字型碼，已備放到最後一列時用
MOV    R3, A              ; 將最上一列備作到 R3 中
MOV    A, @R1             ; 先取下一列的字型碼
MOV    @R0, A             ; 將資料存回上一列，達到上移一列的效果
INC    R0                 ; 處理下一列
INC    R1
DJNZ   R2, MODE31
MOV    A, R3              ; 取回原來最上一列
MOV    @R0, A             ; 存到最後一列
POP    A
POP    PSW
RETI

```

; ===== 工作模式 2 的處理，讓英文字母“A”上移 =====

```

DEC    COUNT0            ; 查看是否已經一秒了
MOV    A, COUNT0
JNZ    MODE22            ; 尚未到一秒，返回主程式
MOV    A, COUNT1         ; 取出計時初值
MOV    COUNT0, A         ; 存回計時器中，以重新計時一秒

```

; ===== 字型左移一個位元，字型“A”全部移出之後又從右邊移入的循環 =====

```

MOV    R0, #40H          ; R0 指向字型區起點
MOV    R1, #48H          ; R1 指向右邊所串接位元組的起點
MOV    R2, #08           ; 共要左旋八列資料，每列十六個位元，以旋代移
MOV    A, @R1            ; 先決定旋轉所串的 C 旗號值
RLC    A                 ; 取出位元 7 的狀態到 C 旗號中
MOV    A, @R0            ; 取資料
MOV    A                 ; 串上 C 旗號向左旋轉一個位元
MOV    @R0, A            ; 存回
MOV    A, @R1            ; 取資料
MOV    A                 ; 串上 C 旗號向左旋轉一個位元
MOV    @R1, A            ; 存回，共十六位元向左旋轉一個位元
INC    R0                 ; 指向顯示字型緩衝區下一列

```

```

INC    R1                ; 指向字型運作緩衝區下一列
DJNA   R2, MODE21
POP    A
POP    PSW
RETI

; -----工作模式 1 的處理不使用本計時-----
; ===== 字 型 區 =====
TABLE_A :      DB      1CH, 22H, 22H, 22H, 3EH, 22H, 22H, 00H ; “A”的字型碼

STRING :      DB      3EH, 08H, 08H, 08H, 08H, 08H, 08H, 00H ; “T”的字型碼
          DB      08H, 18H, 08H, 08H, 08H, 08H, 1CH, 00H ; “1”的字型碼
          DB      1CH, 22H, 02H, 04H, 08H, 10H, 3EH, 00H ; “2”的字型碼
          DB      3EH, 04H, 08H, 04H, 02H, 22H, 1CH, 00H ; “3”的字型碼
          DB      04H, 0CH, 14H, 24H, 3EH, 04H, 04H, 00H ; “4”的字型碼
          DB      3EH, 20H, 3CH, 02H, 02H, 22H, 1CH, 00H ; “5”的字型碼
          DB      1CH, 22H, 20H, 3CH, 22H, 22H, 1CH, 00H ; “6”的字型碼
          DB      3EH, 02H, 04H, 08H, 10H, 10H, 10H, 00H ; “7”的字型碼
          DB      1CH, 22H, 22H, 1CH, 22H, 22H, 1CH, 00H ; “8”的字型碼
          DB      1CH, 22H, 22H, 1EH, 02H, 22H, 1CH, 00H ; “9”的字型碼
          DB      00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, ; “ ”的字型碼

; =====程式到此結束=====

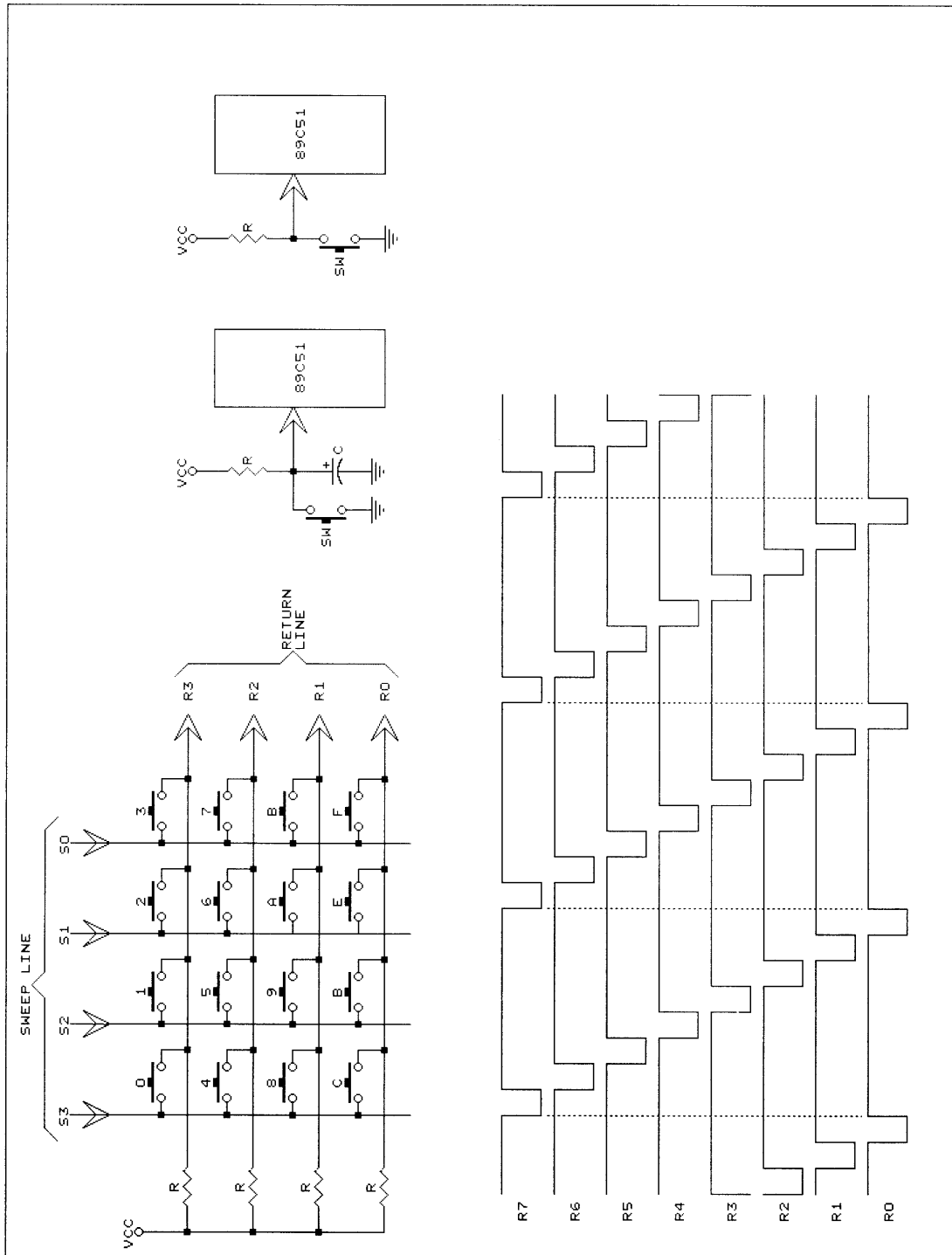
          END

```

伍、實體照片：

學習評量三

學員能在 30 分鐘內，在不參考任何資料下，繪出下圖之矩陣式鍵盤電路的掃描程式流程圖，已知每間隔 1ms 切換一條掃描線的掃描速率，以及 10ms 到 20ms 間的彈跳消除時間。



學習評量三答案

NONK：沒有按鍵的掃描次數

NEWK：上一次按鍵值

KEYV：此次掃描到的鍵值

KEYT：同一按鍵被掃描次數

KEY：已承認的按鍵值

初值：NONK=NEWK=00

KEYV=KEY =00

