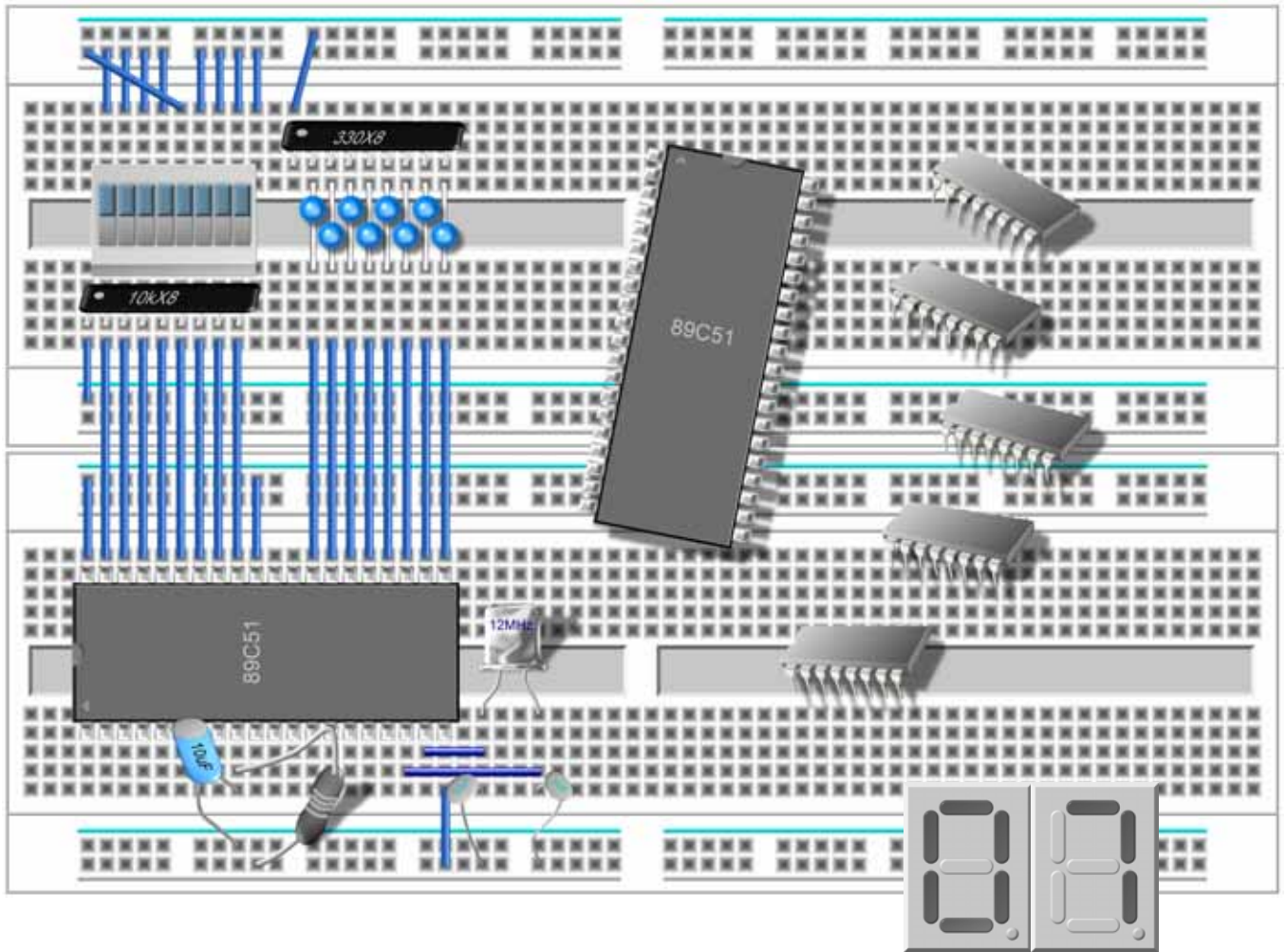




串列埠之應用

本章內容豐富，主要包括三部分：

● 硬體部分：

介紹 8051 串列埠、串列資料轉並列資料 IC、並列資料轉串列資料 IC，以及 MAX232 等。

● 程式與實作部分：

Mode 0 串列埠的接收與傳送、其它工作模式的應用、兩台 8051 小系統的互傳應用等。

7-1

資料傳輸概念

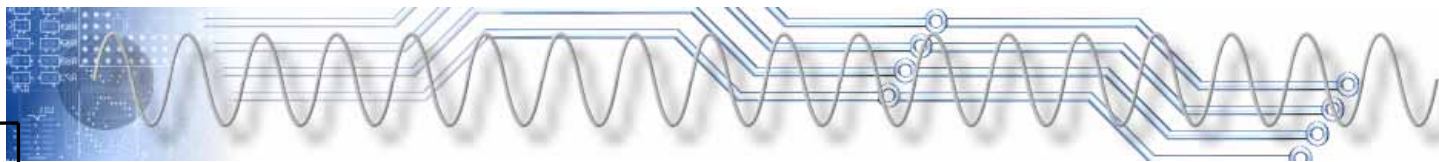
在微電腦與數位系統裡，資料傳輸可分為並列式傳輸與串列式傳輸，並列式傳輸一次傳輸多個位元(通常是 8 位元)。因此，連接兩個系統之間的傳輸線，就必須很多條。當然，一次傳輸多個位元，傳輸的速度就比較快；但其傳輸線數比較多，線路費用相對提高，線路阻抗匹配、雜訊等問題也較多，所以不適合長距離的通訊。串列式傳輸每次傳輸一個位元，資料傳輸的速度似乎不怎麼快，但連兩個系統之間，只要兩條傳輸線即可，適合於長距離的通訊。

到底要使用並列式傳輸還是串列式傳輸，全視資料量與環境而定。在前面的單元裡，不管使用哪個 PORT，大都是一次輸出入 8 個位元，屬於近距離的並列式資料傳輸。若要將 8051 系統的資料傳至另一 8051 系統，則可使用串列式資料傳輸。最典型的串列式資料傳輸，算是 RS232C，例如個人電腦裡的 com1、com2 就屬 RS232C。雖然個人電腦裡的輸出入介面，逐漸被新一代的 USB、網卡所取代，但 USB、網卡也屬於串列式資料傳輸。串列式資料傳輸的媒介，除銅導線外，還可透過光纖，以及電波(無線傳輸)，可說是多采多姿，而其速度更逐日飆高。反觀並列式資料傳輸的速度與傳輸距離，一直遲滯不前，導致並列式資料傳輸日漸式微。

在串列式資料傳輸裡，有單工及雙工之分，單工就是一條線只能有一種用途，例如輸出線就只能將資料傳出、輸入線就只能將資料傳入。而雙工就是在同一條線上，可傳入資料，也可傳出資料。若在系統上，只有一條傳輸線，而在該傳輸線同一個時間裡，不是進行傳入資料，就是傳出資料，稱之為「**半雙工**」。若在系統上有兩條傳輸線，而這兩條傳輸線上，可同時進行資料傳入與傳出，稱之為「**全雙工**」。

通常以每秒傳輸多少位元(**bit per sector**，簡稱 **bps**)表示串列式資料傳輸的速率，又稱為**鮑率(baud)**。8051 提供一個全雙工的萬用非同步串列埠(**Universal Asynchronous Receiver Transmitter**，簡稱 **UART**)，這個串列埠有四種工作模式(mode)，使用不同工作模式，其鮑率各有不同，如下說明：

- **mode 0**：由 8051 主導，其鮑率為 8051 系統時鐘脈波的 12 分之 1，即 $OSC/12$ ，以 12MHz 的系統為例，則其鮑率為 1M bps。



- **mode 1 或 mode 3**：此模式為可變鮑率的串列埠，主要是為了配合所連接系統的時序，以達到不同系統的資料傳輸。
- **mode 2**：此模式提供兩種不同鮑率的選擇，即 OSC/32 或 OSC/64，其中的 OSC 為 8051 系統時鐘脈波。

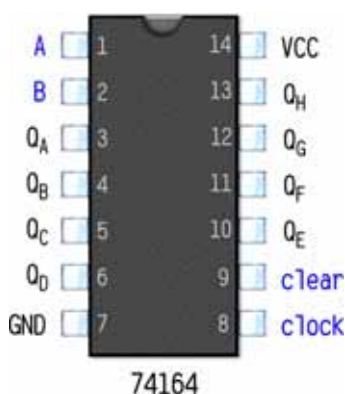
通常，微處理機裡的資料處理，屬於並列式處理，以 8051 而言，一次處理一個位元組，也就是 8 個位元不管怎樣，串列式資料與並列式資料之間的轉換是無法避免的。在 8051 裡，若要把 8 位元的並列資料傳出去，只要把資料放入串列緩衝器(SBUF)即可，8051 就會幫我們把這筆資料，一個位元一個位元丟出去。接收串列資料也是一樣，8051 會把外面傳入的資料，一個位元一個位元塞進 SBUF，當 SBUF 塞滿後，即為並列資料。

7-2

認識 74164/74165

若從 8051 的串列埠，將資料傳出後，則可以透過串列轉並列的 IC(如 74164)，將串列資料轉換成並列資料；若要把並列資料，經由 8051 的串列埠傳入 8051，則需經並列轉串列的 IC(如 74165)，將並列資料轉換成串列資料，即可傳入 8051。以下將介紹 74164 及 74165：

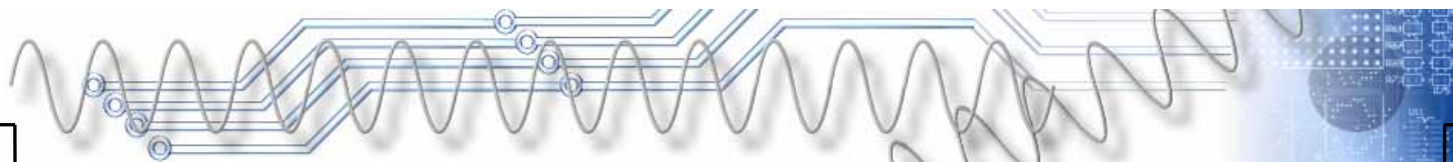
● 74164



(圖1) 74164 接腳圖

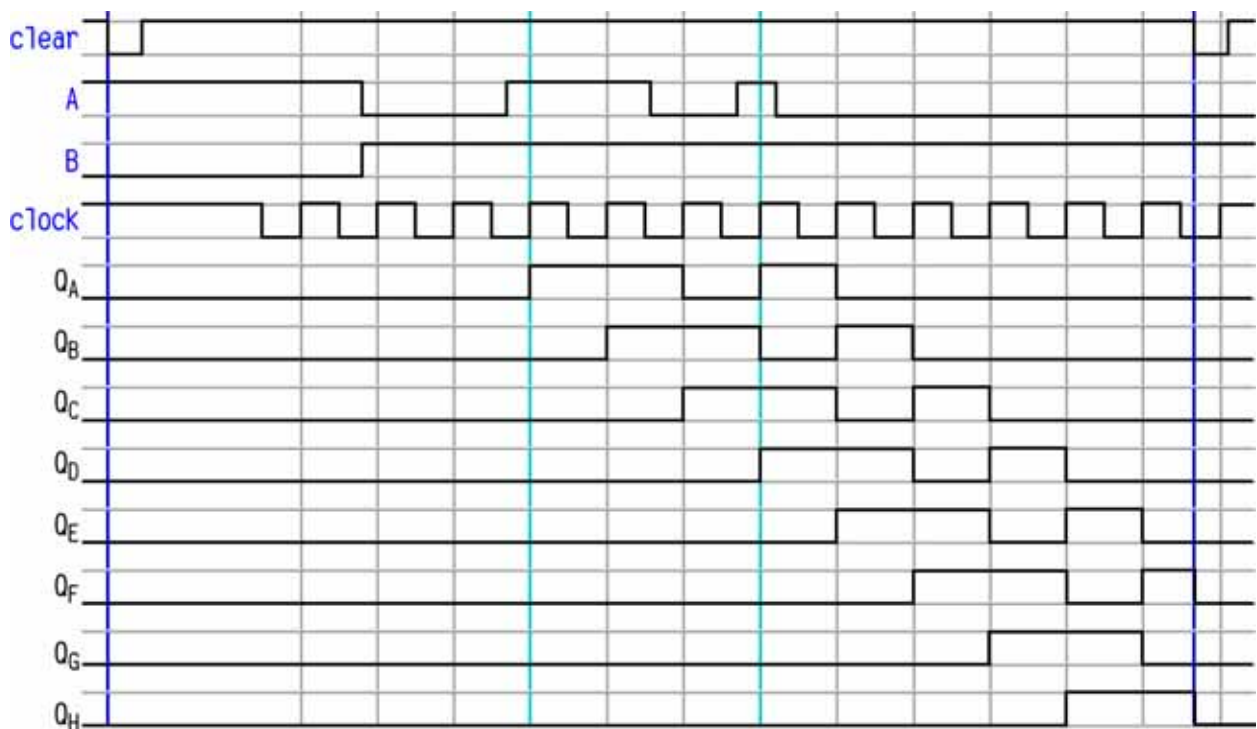
如上圖所示，74164 是一顆串列轉並列的 IC，其中各接腳如下說明：

- **clear**：本接腳為清除接腳，當此接腳為低態時，則並列輸出接腳



將全部變為低態。

- **A、B**：串列資料輸入接腳，這兩隻接腳完全一樣，我們可將這兩隻接腳連接在一起，再接到串列資料來源。也可將其中一腳連接到 VCC，另一隻接腳連接串列資料來源。
- **clock**：本接腳為時鐘脈波接腳，而此 IC 為正緣觸發型，也就是輸出接腳的狀態變化是發生在時鐘脈波由低態變為高態的時候。

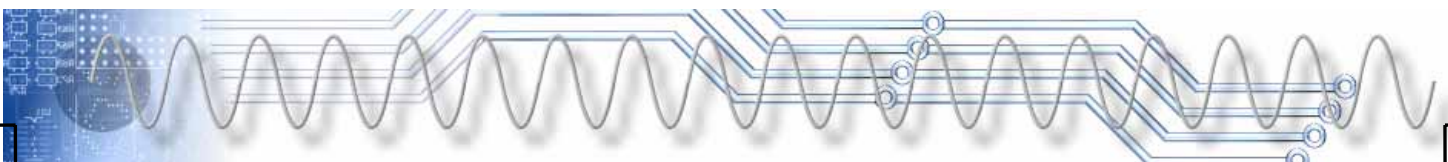


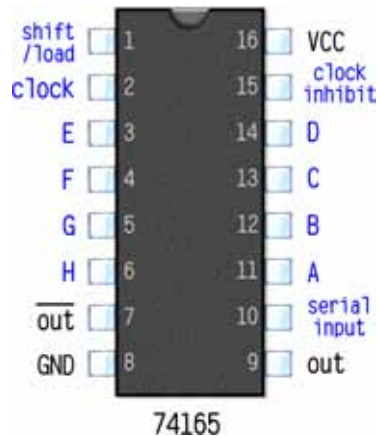
(圖2) 74164 時序圖

- **QA~QH**：資料輸出接腳，這八隻接腳為 b_0 到 b_7 的 8 位元並列資料輸出接腳。

74165

如下圖所示，74165 是一顆並列轉串列的 IC，其中各接腳如下說明：

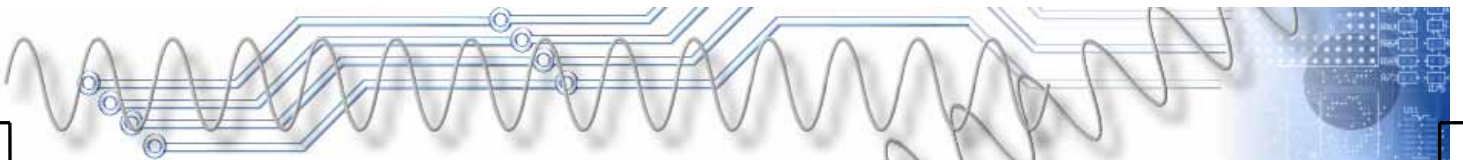




(圖3) 74165 接腳圖

- **shift/load**：本接腳為資料載入與移位控制接腳，當此接腳為低態時，則並列輸入接腳的狀態將全部被載入。當此接腳為高態時，即可隨時鐘脈波進行移位式串列輸出。
- **clock inhibit**：本接腳為時鐘脈波禁致接腳，當本接腳為高態時，輸出接腳將不隨時鐘脈波而變化。當本接腳為低態時，輸出接腳將隨時鐘脈波，而進行變化移位式串列輸出。
- **clock**：本接腳為時鐘脈波接腳，而此 IC 為正緣觸發型，也就是輸出接腳的狀態變化是發生在時鐘脈波由低態變為高態的時候。
- **serial input**：本接腳為串列輸入接腳，若要進行並列資料輸入串列資料輸出的轉換，則此接腳保持為低態即可。若要進串並列資料輸入串列資料輸出的轉換，則此接腳連接串列資料來源。
- **A~H**：並列資料輸入接腳，這八隻接腳為 b_0 到 b_7 的 8 位元並列資料輸入接腳，接到並列資料來源。
- **out**：本接腳為串列資料輸出接腳。
- $\overline{\text{out}}$ ：本接腳為串列資料反相輸出接腳。

74165 的動作如下真值表所示：



輸 入					輸 出	
shift /load	clock inhibit	clock	serial input	並列輸入 A~H	out	$\overline{\text{out}}$
0	x	x	x	a~h	h	$\overline{h}$
1	0	0	x	x	Q_{ho}	$\overline{Q_{ho}}$
1	0	↑	1	x	Q_n	$\overline{Q_n}$
1	0	↑	0	x	Q_n	$\overline{Q_n}$
1	1	↑	x	x	Q_{ho}	$\overline{Q_{ho}}$

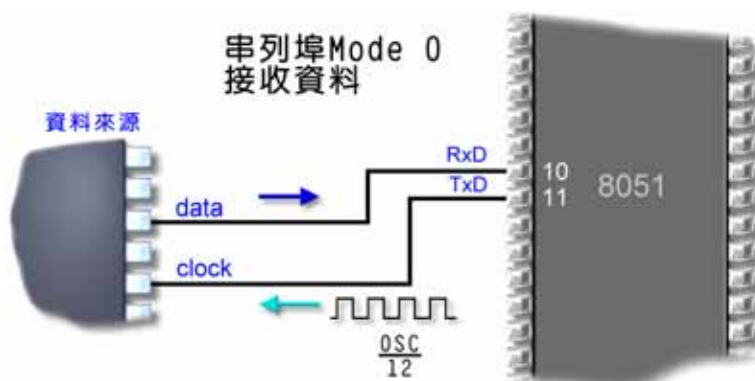
7-3

認識 MCS-51 之串列埠

MCS-51 的串列埠為全雙工的串列埠，可接收及傳送串列資料，不管是接收或傳送，都使用串列緩衝器(SBUF)，接收串列資料時，所輸入的串列資料將塞進 SBUF，塞滿後提出中斷，再將 SBUF 裡的 8 位元資料移做他用。傳送資料時，先將所要送出的資料放入 SBUF，即可一個位元一個位元地傳到目的地。很明顯的，SBUF 扮演關鍵性的角色，在 8051 內容，雖然都叫做 SBUF，但接收用的 SBUF 與傳送用的 SBUF，分別是兩個不同位址的 8 位元暫存器。不過，誰理它？反正對於使用者沒有影響就好了。

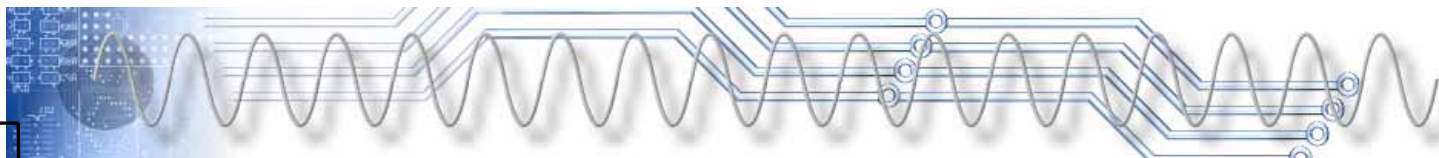
MCS-51 的串列埠有四種工作模式，如下說明：

mode 0

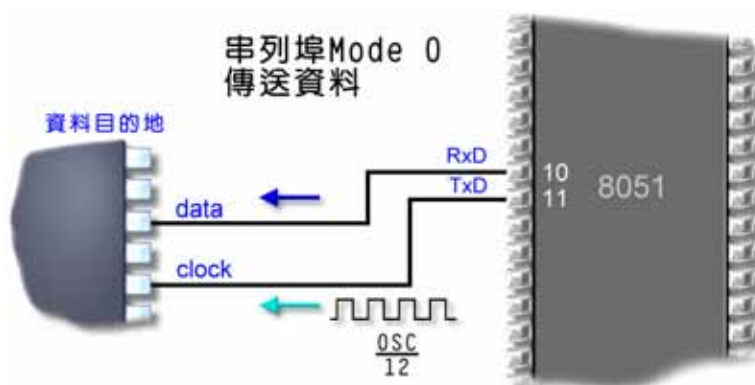


(圖4) 接收資料

mode 0 工作模式是以固定鮑率之移位式資料傳輸，其鮑率為



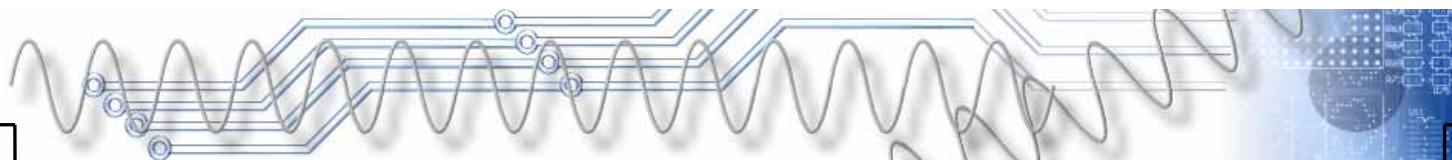
OSC/12，若在 12MHz 下，則其速率為 1Mbps。在此模式下，不管是接收資料還是資料傳出，CPU 的 RxD 接腳(P3.0)連接串列資料線，TxD 接腳(P3.1)連接移位脈波線。執行資料接收時，由 TxD 接腳送出移位脈波，而由 RxD 接腳收下串列資料，如圖 4 所示；執行資料傳送時，也是依據 TxD 接腳所送出的移位脈波，而由 RxD 接腳送出資料串列資料，如圖 5 所示。



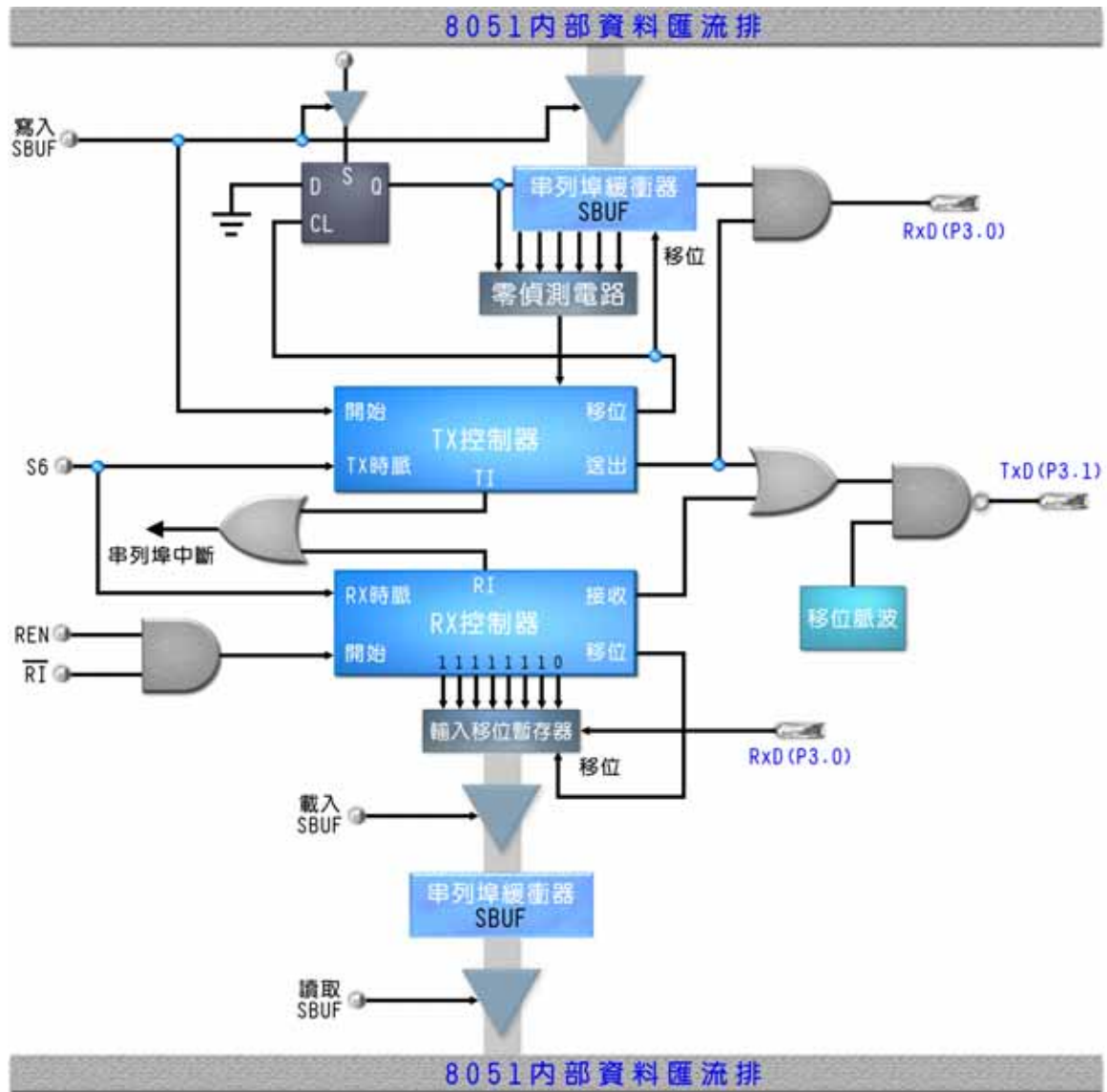
(圖5) 傳送資料

如圖 6 所示為 8051 內部串列埠的架構，只要將所要送出得資料放入 SBUF，例如「MOV SBUF, A」，則寫入 SBUF 的動作會在 S6 的 2 週期，將「1」放入 SBUF 左邊的正反器(視為第 9 位元)，以做為結束傳送的識別位元，同時啟動 TX 控制器的開始信號，而 TX 控制器的送出信號輸出 1，則 TxD 接腳即可輸出移位脈波，開始傳出資料。緊接著，SBUF 內的資料，依序移位經 RxD 接腳輸出，同時在 SBUF 左邊補入 0。當送出八個位元時，SBUF 裡除最右邊為 1 外，其餘皆為 0，零偵測電路(只偵測 SBUF 左邊七個位元)將偵測出 0 的信號，以驅動 TX 控制器，TX 控制器隨即提出一個 TI 中斷信號，並將送出信號輸出為 0，停止傳送。

若要進行接收資料，則須先將 SCON 暫存器的 REN 位元設定為 1、RI 位元設定為 0，方能啟動 RX 控制器的開始信號，RX 控制器將「11111110」載入輸入移位暫存器。緊接著，RX 控制器的接收信號輸出 1，則 TxD 接腳即可輸出移位脈波，開始接收資料。每當一個位元串列資料經 RxD 接腳輸入到輸入移位暫存器時，RX 控制器將送出一個移位信號，讓輸入移位暫存器左移一位。當輸入移位暫存



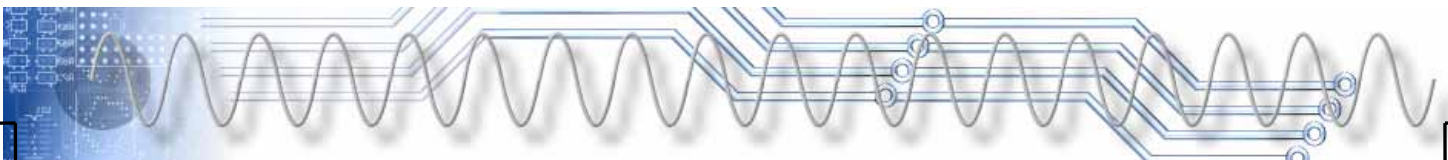
器最左位元為 0 時，RX 控制器將再進行一個位元的輸入，然後提出一個 RI 中斷信號，並將接收信號輸出為 0，停止接收。

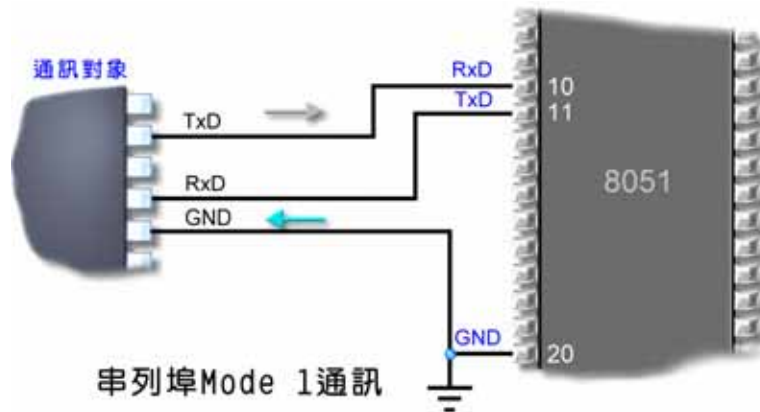


(圖6) 串列埠架構

mode 1

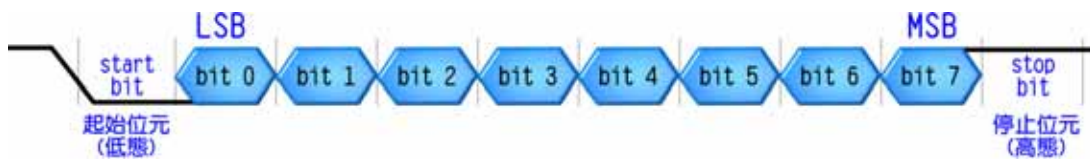
mode 1 工作模式是以可變的鮑率進行串列資料的傳輸，其鮑率可由 Timer 1 來控制(若是 8052 則還可使用 Timer 2 控制鮑率)。在此模式下，8051 的 RxD 接腳連接來源的 TxD 接腳、8051 的 TxD 接腳連接目的地的 RxD 接腳。





(圖7) mode 1 串列資料傳輸

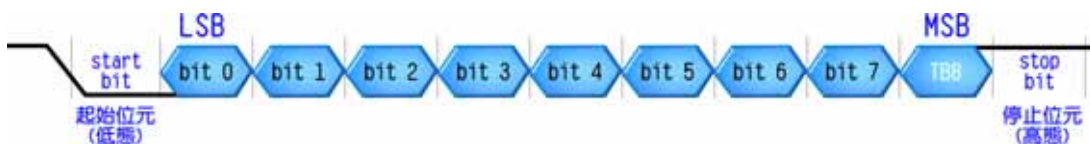
在 mode 1 下，每筆資料是由 10 位元所組成，包括起始位元(start bit)、8 個位元的資料，以及停止位元(stop bit)，其中第一個位元就是低態的起始位元，緊接著是 8 位元資料的 bit 0(即 LSB)，接續於 bit 7(MSB)之後的是高態的停止位元，如圖下所示：



(圖8) mode 1 的資料串格式

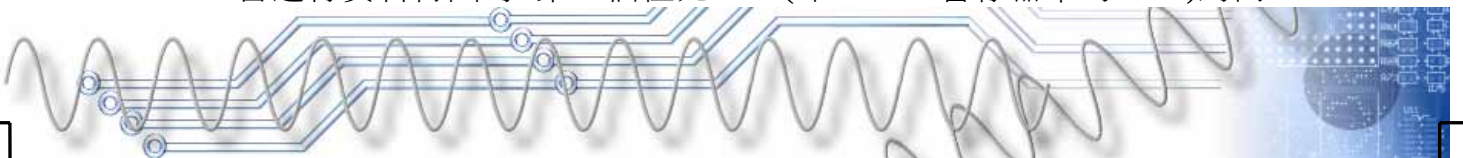
mode 2

mode 2 工作模式是以 OSC/32 或 OSC/64 的鮑率進行串列資料的傳輸，而其線路的連接，也是 8051 的 RxD 接腳連接來源的 TxD 接腳、8051 的 TxD 接腳連接目的地的 RxD 接腳。而在 mode 2 下，每筆資料是由 11 位元所組成，包括起始位元(start bit)、8 個位元的資料、同位位元(parity bit)，以及停止位元(stop bit)，其中第一個位元就是低態的起始位元，緊接著是 8 位元資料的 bit 0(即 LSB)，接續於 bit 7 之後的是同位位元，最後則是高態的停止位元，如圖下所示：



(圖9) mode 2 的資料串格式

當進行資料傳出時，第 9 個位元 TB8(即 SCON 暫存器中的 TB8)為同



位位元，可取自程式狀態字組暫存器 PSW 中的 P 位元，以達到同位檢查的目的。當接收資料時，第 9 個位元將直接移入 SCON 暫存器中的 RB8，而不必管停止位元。

mode 3

mode 3 工作模式是以可變的鮑率進行串列資料的傳輸，其鮑率可由 Timer 1 來控制(若是 8052 則還可使用 Timer 2 控制鮑率)。除此之外，mode 3 與 mode 2 幾乎完全一樣。

7-4

認識 SCON 串列埠控制暫存器



(圖10) SCON 暫存器

串列埠控制暫存器(serial port control register，簡稱 SCON)是一個 8 位元、可位元定址的暫存器，其功能是設定與控制串列埠，其中個位元如下說明：

SM0 與 SM1

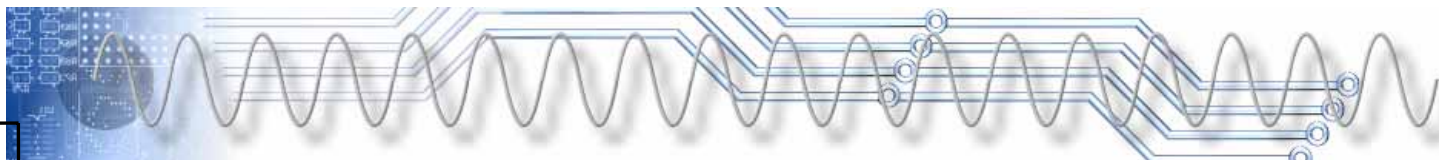
這兩個位元的功能是設定串列埠的模式，如下表所示：

SM1	SM0	mode	功 能 簡 介	鮑 率
0	0	0	移位暫存器	OSC/12
0	1	1	8 位元 UART	可 變
1	0	2	9 位元 UART	OSC/32 或 OSC/64
1	1	3	9 位元 UART	可 變

SM2

本位元為多重處理器通訊致能位元，如下說明：

- mode 0 時，SM2=0。
- mode 1 時，若 SM2=1，且收到有效的停止位元，則 RI=1(提出 RI 中斷)，否則 RI=0。



- mode 2 或 mode 3 時，若 SM2=1，且收到的第 9 位元為 1，則 RI=1(提出 RI 中斷)；第 9 位元為 0，則 RI=0。

REN

本位元為串列接收致能位元，如下說明：

- REN=1，開始接收。
- REN=0，停止接收。

TB8

mode 2 或 mode 3 傳送資料時，本位元為第 9 傳送位元，可用軟體來設定或清除。

RB8

- mode 2 或 mode 3 接收資料時，本位元為第 9 個接收位元。
- mode 1 時，若 SM2=0，則本位元為停止位元。
- mode 0 時，本位元無作用。

TI

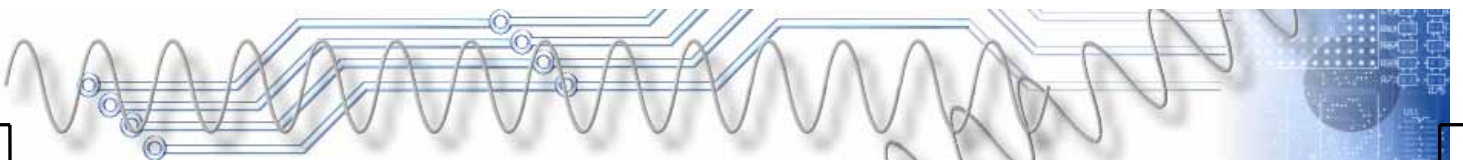
本位元為傳送中斷旗標，當中斷結束時，本位元並不會恢復為 0，必須由軟體清除。

- mode 1、mode 2 或 mode 3 時，若完成傳送停止位元，則本位元自動設定為 1，並提出 TI 中斷。
- mode 0 時，若完成傳送第 8 位元，則本位元自動設定為 1，並提出 TI 中斷。

RI

本位元為接收中斷旗標，當中斷結束時，本位元並不會恢復為 0，必須由軟體清除。

- mode 1、mode 2 或 mode 3 時，若完成接收到停止位元，則本位元自動設定為 1，並提出 RI 中斷。
- mode 0 時，若完成接收第 8 位元，則本位元自動設定為 1，並提出 RI 中斷。



7-5 鮑率設定

8051 串列埠的鮑率設定方式，如下說明：

mode 0

在 mode 0 下，鮑率固定為 $OSC/12$ ，不必要設定！完全依系統的時鐘脈波而定，非軟體設計所能改變的。

mode 2

在 mode 2 下，其鮑率可為 $OSC/32$ 或 $OSC/64$ ，即：

$$\text{鮑率} = \frac{2^{\text{SMOD}}}{64} \times OSC$$

其中 SMOD 為 PCON 暫存器中的 bit 7，若將 SMOD 設定為 0，則設定採用的鮑率 $OSC/64$ ；若將 SMOD 設定為 1，則設定採用的鮑率 $OSC/32$ 。以 12MHz 的系統為例，

$$\text{若 SMOD}=0, \text{ 則 鮑率} = \frac{2^0}{64} \times 12M = 187.5K \text{ (bps)}$$

$$\text{若 SMOD}=1, \text{ 則 鮑率} = \frac{2^1}{64} \times 12M = 375K \text{ (bps)}$$

mode 1 或 mode 3

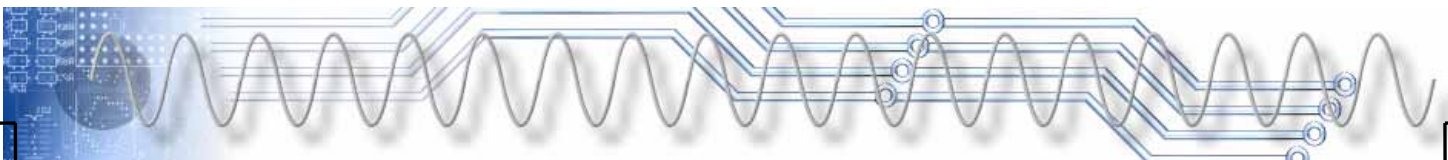
在 mode 1 或 mode 3 下，鮑率可由 Timer 1(8052 則還可選擇 Timer 2) 的溢位脈波所控制，以 Timer 1 採用具自動載入功能的 mode 2 為例，其產生的鮑率為：

$$\text{鮑率} = \frac{2^{\text{SMOD}}}{32} \times \frac{OSC}{12 \times (256 - TH1)}$$

若在 11.0592MHz 的系統下，想要產生 19.2K bps 的鮑率，且 SMOD=1，則

$$19200 = \frac{2^1}{32} \times \frac{11059200}{12 \times (256 - TH1)} = \frac{11059200}{16 \times 12 \times (256 - TH1)}$$

$$256 - TH1 = 3, TH1 = 253 = FDH$$

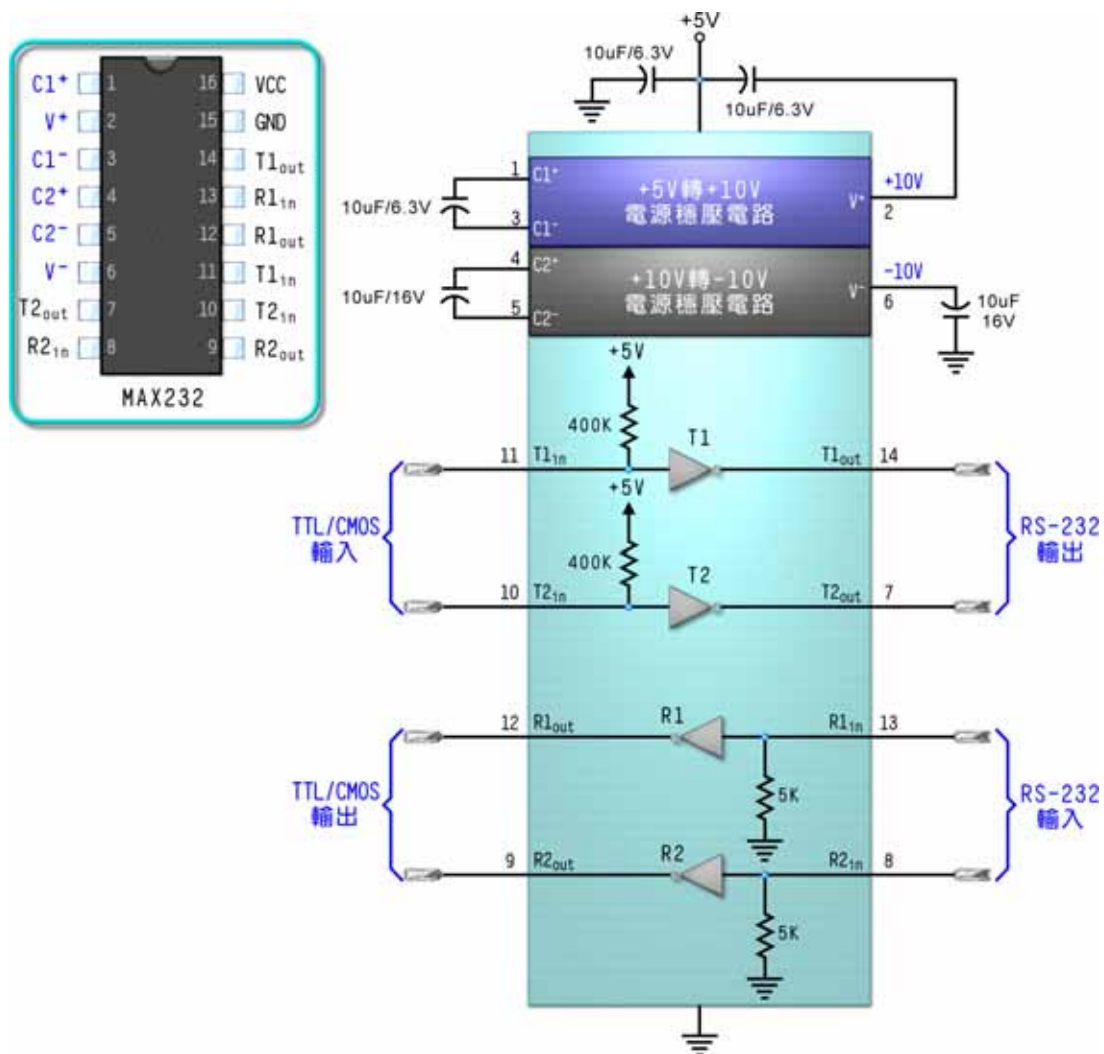


如下表所示是使 Timer 1，以設定常用的鮑率：

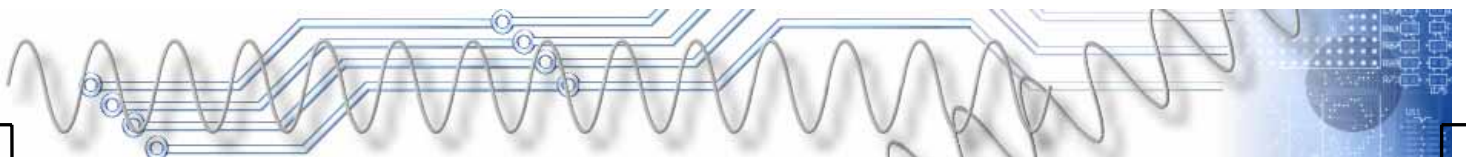
鮑 率	OSC	SMOD	TH1
62.5K	12MHz	1	FFH
19.2K	11.0592MHz	1	FDH
9600	11.0592MHz	0	FDH
4800	11.0592MHz	0	FAH
2400	11.0592MHz	0	F4H
1200	11.0592MHz	0	E8H
137.5	11.0592MHz	0	1DH

7-6

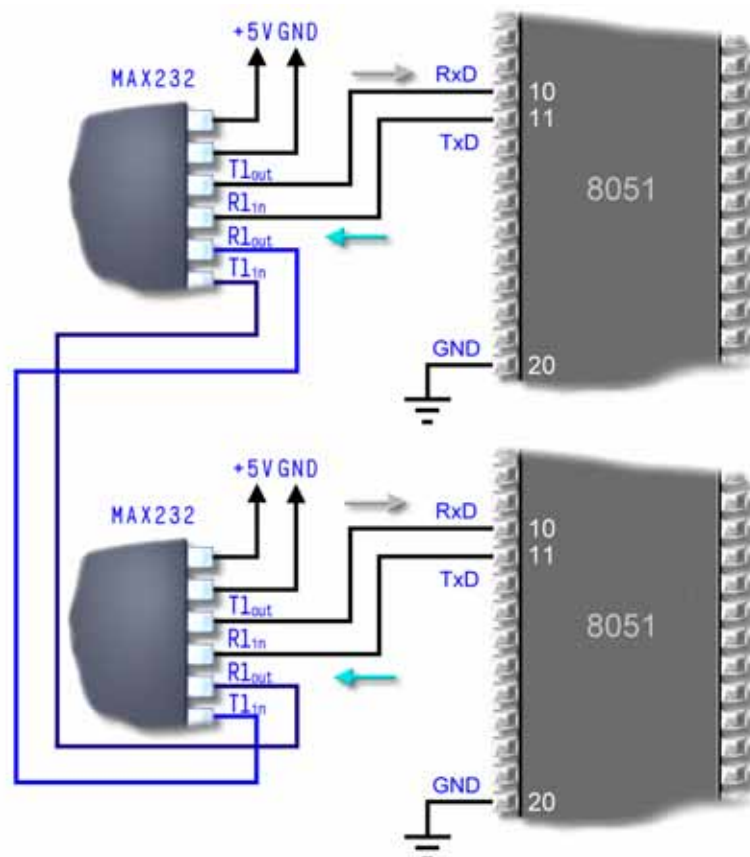
認識 MAX232



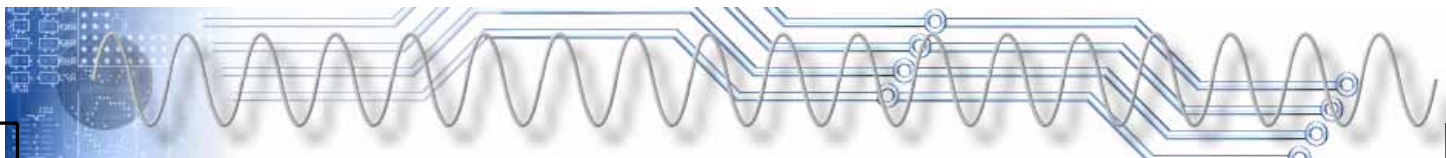
(圖11) MAX232



如果是短距離的串列資料傳輸，則標準的 TTL 或 CMOS 足以應付；若要進行距離的串列資料傳輸，使標準的 TTL 或 CMOS，恐怕驅動能力不足，且雜訊邊限太小，通訊品質很差！RS-232 是一種可長距離傳輸的通訊方式，因此，必須突破雜訊邊限太小與驅動能力不足的限制，於是相關的驅動 IC 因應而生，Maxim 公司的 MAX232 系列就屬這類 IC，以其 MAX232 為例，如圖 11 所示，這顆 IC 提供 RS232 傳送與接收的驅動，在傳送方面，MAX232 內部將+5V 電源提升為+10V 及-10V，然後接受 TTL/CMOS 的+5V 準位，而轉換成±10V 的信號，再送到線路上。在接收方面，MAX232 從線路上接受±10V 的信號，經內部緩衝器，轉換成 TTL/CMOS 的+5V 準位。說穿了，MAX232 只不過是個準位轉換裝置而已，但只要+5V 電源與 4 至 5 個小電容(10 μ F 即可)，就能同時提供雙向的準位調整。對於使用者而言，就把它當成一般的緩衝器來使用即可。



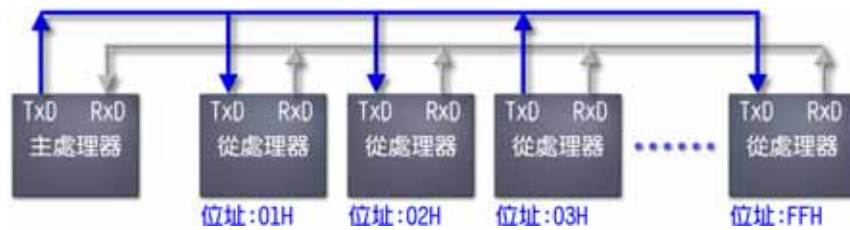
(圖12) RS232 線路



7-7

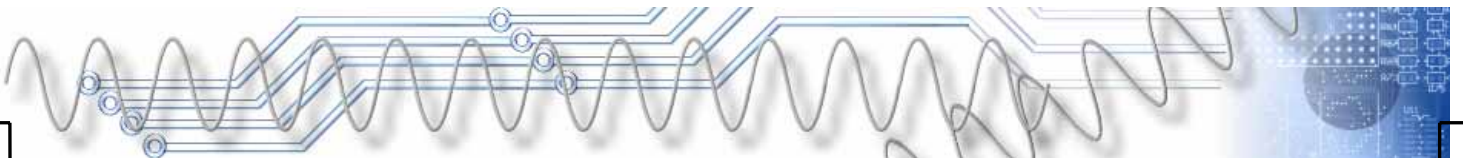
多處理器間之資料傳輸

前面所介紹的串列式資料傳輸，可說是一對一的通訊，也就是說 A 系統透過兩條傳輸線與 B 系統通訊。而 8051 也支援多處理器之間的資料傳輸，如下圖所示，主處理器(master processpr)與數個從處理器(slave processpr)構成一個主從式架構，每個處理器的 TxD 線都接在一起，而 Rx/D 線也接在一起，對於每個從處理器都予編碼定址。



(圖13) 主從式處理器架構

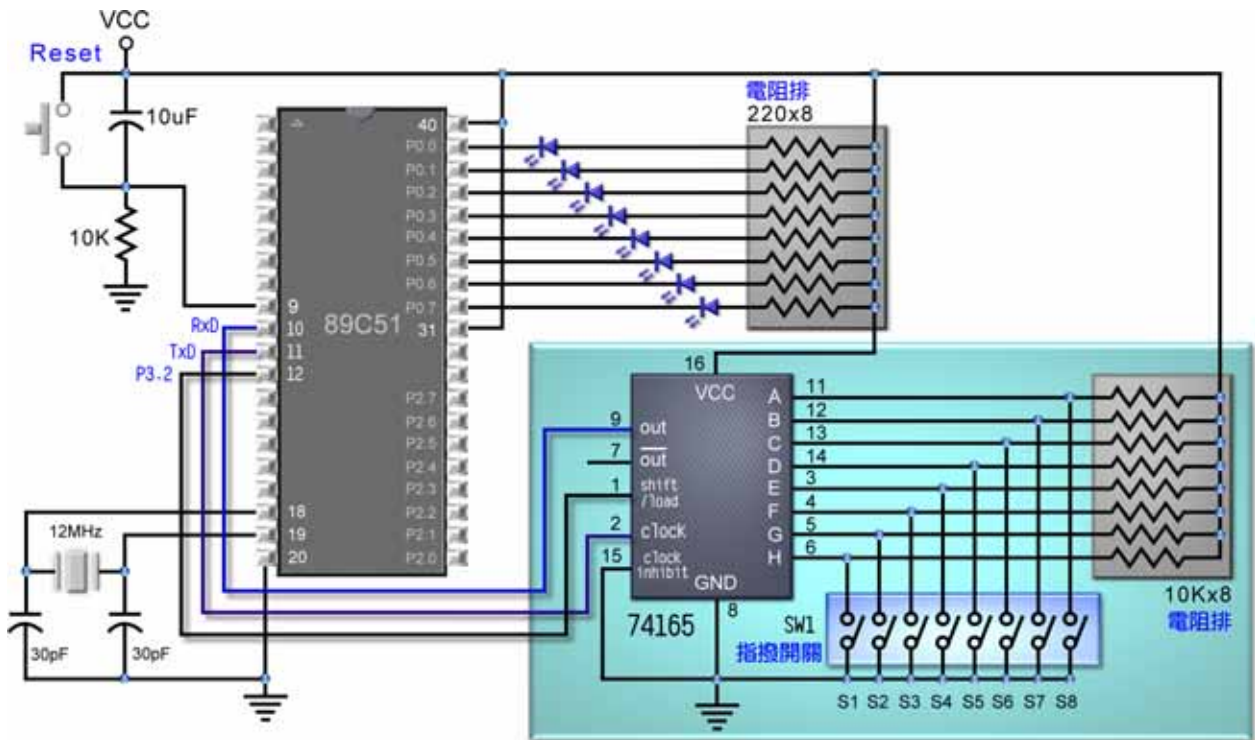
當 8051 的串列埠工作於 mode 2 或 mode 3 時，若主處理器要傳一筆資料給某個從處理器時，先送出目的從處理器的位址位元組(address byte)，而資料位元組與位址位元組之差別在於其第 9 位元，若第 9 位元為 1，代表該筆資料為位址；若第 9 位元為 0，代表該筆資料為資料。若從處理器的 SM2 位元為 1 時，如接收到的是位址位元組，將會產生中斷；如接收到的是資料位元組，則不會產生中斷。因此，當主處理器送出位址位元組時，所有從處理器都將產生中斷，然後各自判斷所接收到的地址，是否就是自己的地址？若是(表示接下來的資料就是要給它的)，則將 SM2 位元設為 0，然後準備接收下一筆資料。而其它非此地址的從處理器，將不接收下一筆資料。



7-8 實例演練

在本單元裡提供六個範例，如下所示：

7-8-1 移位式資料串入實例演練



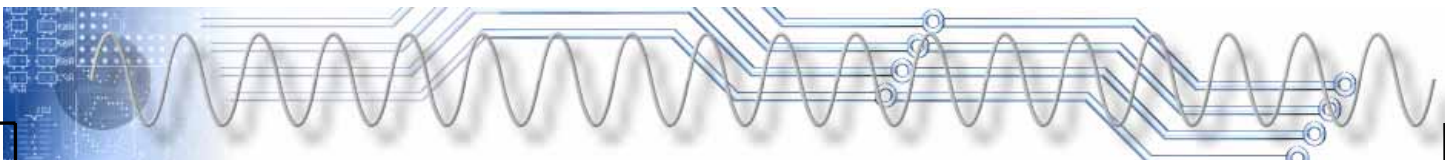
(圖 14) 電路圖

● 功能說明

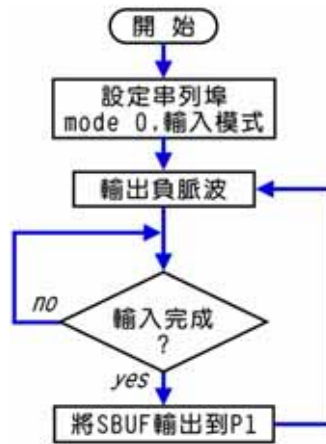
如圖 14 所示，串列資料的來源是利用 74165，將指撥開關 DIP SW 的狀態轉換成串列資料而由 P2 輸入，而其狀態將反應到 P0 所連接的 LED 上。

● 參考程式

根據 74165 的真值表(7-6 頁)，若要將並列資料載入 74165，則需給 74165 的第 1 腳(shift/load)一個負脈波，在此利用 P3.2 接腳連接到 74165 的第 1 腳，所以利用「CLR P3.2」將 P3.2 接腳變為低態，再以兩個「NOP」無作為指令多延遲 2 微秒，最後再以「SETB P3.2」將 P3.2 接腳恢復為高態。74165 載入 DIP SW 上的並列資料後，隨



即依 8051 透過 TxD 接腳傳來的移位脈波，將資料一位元一位元地由 RxD 接腳傳入 8051。當 SBUF 滿了，即產生 RI 中斷，我們只要將 SBUF 緩衝器裡的資料，複製到 P0 即可。



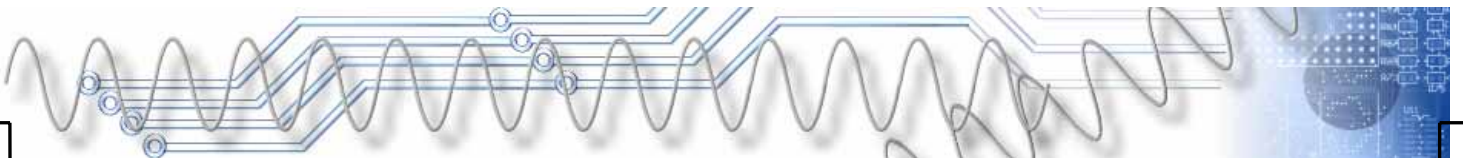
```

START:      ORG    0           ;程式從 0 位址開始
            MOV    SCON, #11H   ;設定為 mode 0、REN=1、RI=1
            ;====b7===b6===b5===b4===b3===b2===b1===b0===
            ;===SM0==SM1==SM2==REN==TB8==RB8===TI==RI==
            ;=====0=====0=====1=====0=====0=====0=====1=====
LOOP:        CLR    P3.2        ;輸出負脈波
            NOP                ; 讓 74165 載入
            NOP                ; DIP SW 並列資料
            SETB    P3.2        ;
            CLR    RI           ;清除 RI
            JNB     RI, $        ;等待 RI 串列輸入中斷
            MOV     P1, SBUF     ;直到將串列輸入資料輸出到 P0
            JMP     LOOP        ;跳至 LOOP 形成一個迴圈
            END
  
```

Mode 0 串列輸入實驗(ch7-1.asm)

● 操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 14 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是



否正常？

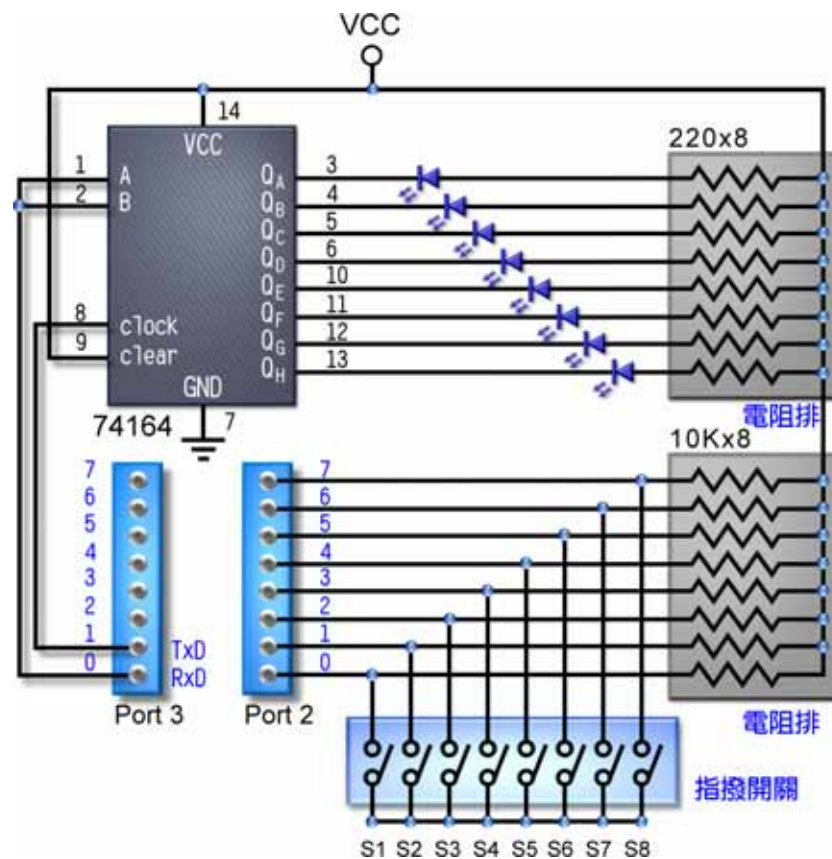
4. 撰寫實驗報告。

● 思考一下

1. 在本實驗裡，有沒有「彈跳」的困擾？
2. 若希望指撥開關中的 S1、S3、S5 三個開關都 on，則前四個 LED 亮；S2 或 S4 或 S6 開關 on，則後四個 LED 亮；S7 及 S8 開關 on，則所有 LED 全亮，程式應如何撰寫？

7-8-2

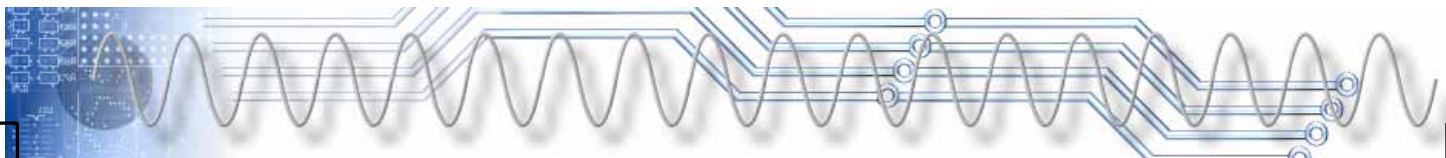
移位式資料串出實例演練



(圖 15) 電路圖

● 功能說明

如圖 15 所示，串列資料由 8051 的 RxD 接腳連接到 74164 的 A、B 輸入接腳，而 8051 的 TxD 接腳連接到 74164 的 clock 接腳，形成一個串列輸出的電路。當然，74164 的 clear 接腳必須連接到 VCC，74164

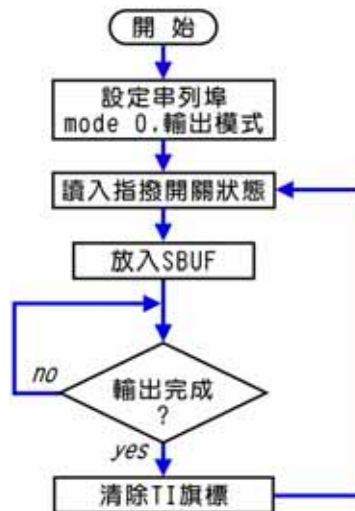


才會正常工作。在此我們希望將 8P 指撥開關 DIP SW 的狀態，經 Port 2 輸入到 8051，再由 8051 透過上述串列路徑，輸出到 74164，而 74164 的並列輸出接腳，連接 8 個 LED，讓這 8 個 LED 反應指撥開關的狀態。



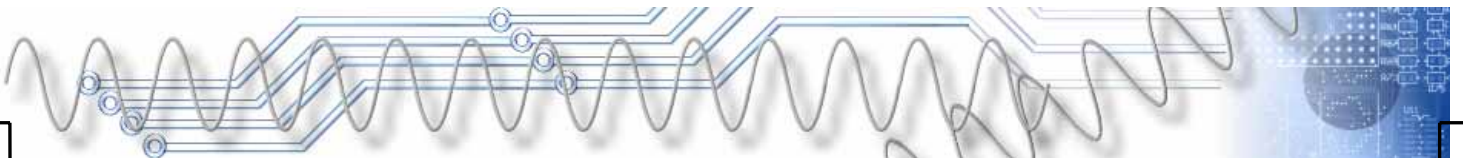
參考程式

在此要以 mode 0 進行資料的串列輸出，所以將 SCON 設定為 0，即為 mode 0。緊接著，從 P2 讀取指撥開關的狀態，然後把它丟入 SBUF，CPU 即進行串列資料的輸出。程式只要等待 TI 中斷，表示 SBUF 裡的資料以全部傳出，再重新讀取 Port 2 的開關狀態，進行同樣的串列輸出即可。



START:	ORG 0	; 程式從 0 位址開始
	MOV SCON, #0H	; 設定為 mode 0
	;====b7===b6===b5===b4===b3===b2===b1===b0===	
	;===SM0===SM1===SM2===REN===TB8===RB8===TI===RI===	
	;=====0=====0=====0=====0=====0=====0=====0=====0=====	
LOOP:	MOV P2, #FFH	; 將 P2 設定為輸入功能
	MOV A, P2	; 讀取指撥開關狀態
	MOV SBUF, A	; 將 ACC 的內容複製到 SBUF
	JNB TI, \$	; 等待 TI 串列輸出中斷
	CLR TI	; 清除 TI 旗標
	JMP LOOP	; 跳至 LOOP 形成一個迴圈
	END	

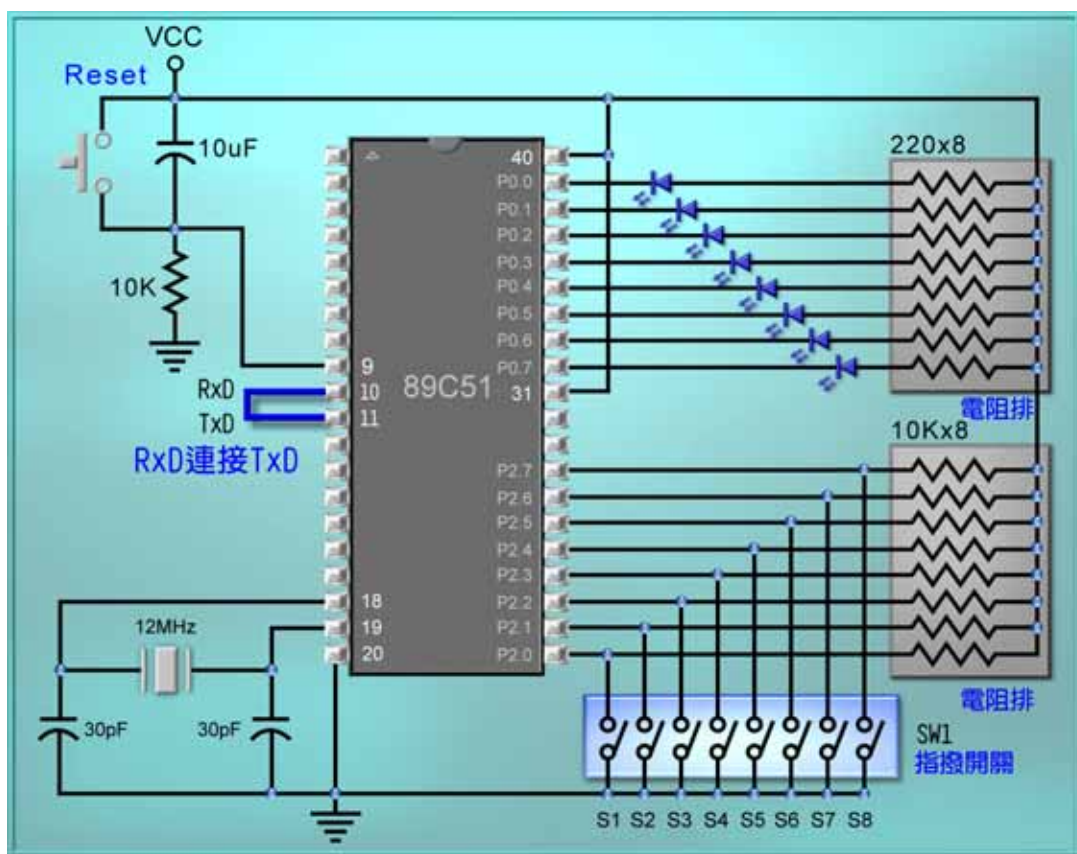
Mode 0 串列輸出實驗 (ch7-2.asm)



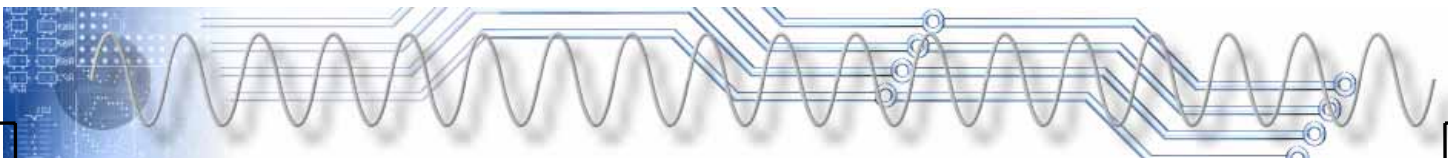
 操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 14 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。

7-8-3 mode 1 實例演練



(圖 16) 電路圖

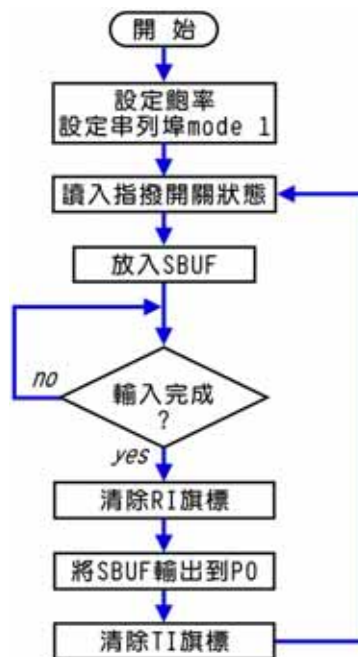


功能說明

如圖 16 所示，我們將 89C51 的第 10 與 11 腳短路，讓串列資料輸出連接到串列輸入，以達到自己傳給自己的目的。另外，第 11 腳所要傳出的資料是來自 Port 2 所連接的指撥開關 DIP SW 狀態；而第 11 腳所接收到的串列資料，將反應到 P0 所連接的 LED 上。

參考程式

在此所要採用的鮑率為 9600bps，首先利用 TMOD 暫存器，將 Timer 1 設定為 Mode 2，再將 PCON 暫存器的 SMOD 位元設定為 0，然後 TH1 暫存器載入 FDH(即 253)，最後啟動 Timer 1，即可產生 9600bps 的鮑率。另外，在 SCON 暫存器裡，將串列埠設定為 mode 1。即可由 Port 2 所讀取的資料，放入 SBUF 暫存器(傳送之 SBUF 暫存器)，CPU 即自動傳送。另一方面，CPU 也自動接收，當接收之 SBUF 暫存器滿了，即產生 RI 中斷，我們只要將 SBUF 緩衝器裡的資料，複製到 P0 即可。



START:	ORG	0	; 程式從 0 位址開始
	MOV	TMOD, #20H	; 將 Timer 1 設定 mode 2
			; 做為鮑率產生
	ANL	PCON, #3FH	; 將 SMOD 設定為 0
	MOV	TH1, #FDH	; 鮑率設定為 9600bps
	SETB	TR1	; 啟動 Timer 1

```

MOV     SCON, #01010000B    ;設定為 mode 1、REN=1、RI=0
;====b7===b6===b5===b4===b3===b2===b1===b0===
;===SM0==SM1==SM2==REN==TB8==RB8===TI===RI===
;=====0=====1=====0=====1=====0=====0=====0=====0=====
LOOP:   MOV     P2, #FFH      ;將 P2 設定為輸入功能
        MOV     A, P2        ;讀取指撥開關狀態
        MOV     SBUF, A      ;將 ACC 的內容複製到 SBUF
        JNB     RI, $        ;檢查是否完成接收?
                                ;完成接收，則 RI 串列輸入中斷
        CLR     RI          ;清除 RI 旗標
        MOV     P0, SBUF     ;將所接收的資料輸出到 Port 0
        CLR     TI          ;清除 TI 旗標
        JMP     LOOP        ;跳至 LOOP 形成一個迴圈
        END

```

Mode 1 實驗 (ch7-3.asm)

● 操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 16 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。

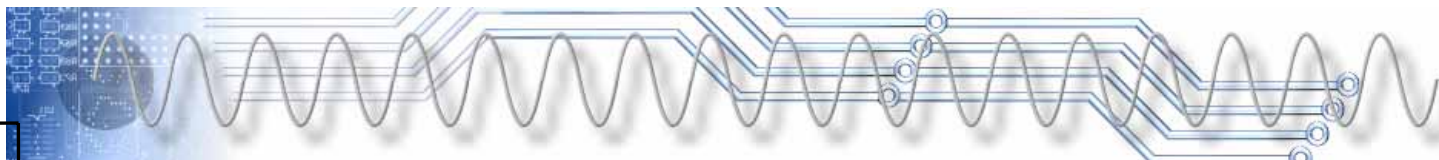
7-8-4 mode 2 實例演練

● 功能說明

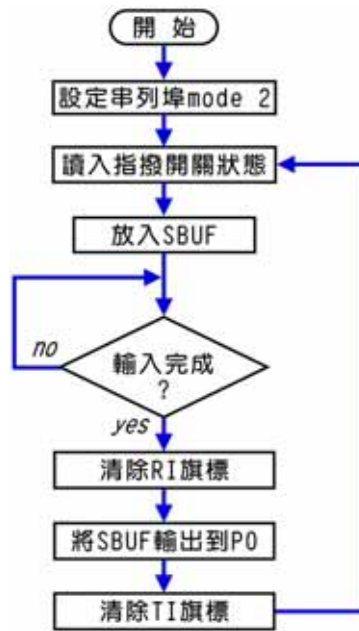
如圖 16 所示，其功能與 7-8-3 節相同，唯，在此採用 mode 2 工作模式。

● 參考程式

在鮑率的設定方面，串列埠的 mode 2 比 mode 1 單純！只有 OSC/64 及 OSC/32 兩種，若 SMOD 位元設定為 0，則為 OSC/64、SMOD 位元設定為 1，則為 OSC/32。在此採用較快的鮑率(OSC/32)，因此，



將 PCON 暫存器的 SMOD 位元設定為 1，然後在 SCON 暫存器裡，將串列埠設定為 mode 2、REN 為 1。即可由 Port 2 所讀取的資料，放入 SBUF 暫存器(傳送之 SBUF 暫存器)，CPU 即自動傳送。另一方面，CPU 也自動接收，當接收之 SBUF 暫存器滿了，即產生 RI 中斷，我們只要將 SBUF 緩衝器裡的資料，複製到 P0 即可。

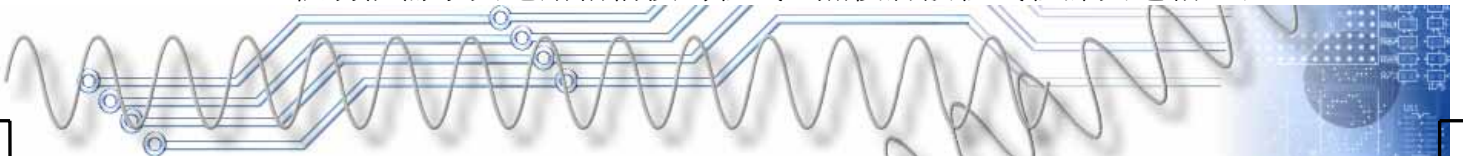


START:	ORG 0	;程式從 0 位址開始
	ORL PCON, #80H	;將 SMOD 設定為 1
	MOV SCON, #10010000B	;設定為 mode 2、REN=1、RI=0
	;====b7===b6===b5===b4===b3===b2===b1===b0===	
	;===SM0==SM1==SM2==REN==TB8==RB8===TI==RI==	
	;=====1====0====0====1====0====0====0====0====	
LOOP:	MOV P2, #FFH	;將 P2 設定為輸入功能
	MOV A, P2	;讀取指撥開關狀態
	MOV SBUF, A	;將 ACC 的內容複製到 SBUF
	JNB RI, \$	;檢查是否完成接收?
	;完成接收，則 RI 串列輸入中斷	
	CLR RI	;清除 RI 旗標
	MOV P0, SBUF	;將所接收的資料輸出到 Port 0
	CLR TI	;清除 TI 旗標
	JMP LOOP	;跳至 LOOP 形成一個迴圈
	END	

Mode 2 實驗(ch7-4.asm)

● 操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以



產生*.HEX 檔。

2. 請按圖 16 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。

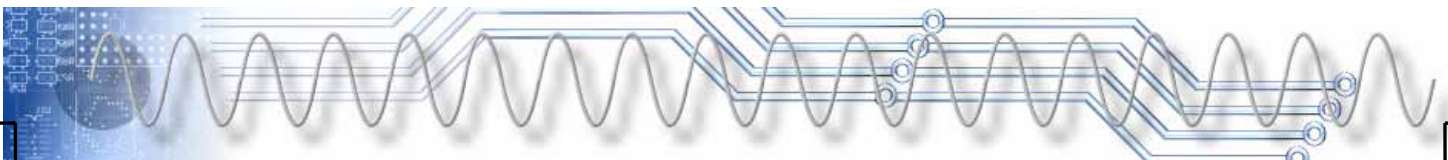
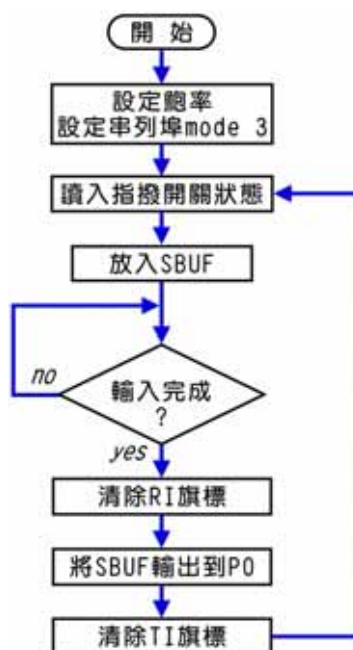
7-8-5 mode 3 實例演練

功能說明

如圖 16 所示，其功能與 7-8-3 節相同，唯，在此採用 mode 3 工作模式。

參考程式

在此所要採用的鮑率也是 9600bps，因此，除了將 mode 2 的設定，改成 mode 3 外，本程式與 Ch7-2.asm 大致相同。



```

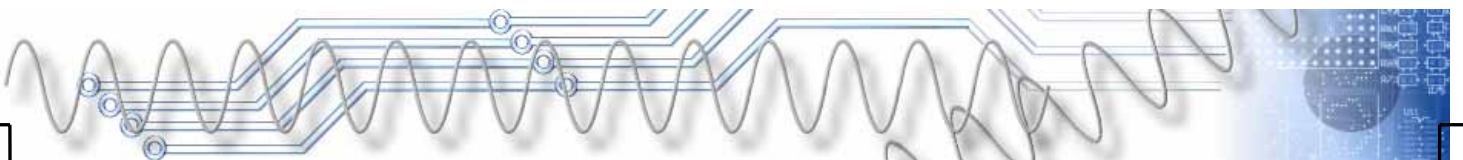
START:      ORG      0           ;程式從 0 位址開始
            MOV      TMOD, #20H   ;將 Timer 1 設定 mode 2
                                   ;做為鮑率產生
            ANL      PCON, #3FH   ;將 SMOD 設定為 0
            MOV      TH1, #FDH    ;鮑率設定為 9600bps
            SETB     TR1          ;啟動 Timer 1
            MOV      SCON, #11010000B ;設定為 mode 3、REN=1、RI=0
                                   ;====b7===b6===b5===b4===b3===b2===b1===b0===
                                   ;===SM0==SM1==SM2==REN==TB8==RB8===TI===RI===
                                   ;=====1=====1=====0=====1=====0=====0=====0=====0=====
LOOP:        MOV      P2, #FFH     ;將 P2 設定為輸入功能
            MOV      A, P2        ;讀取指撥開關狀態
            MOV      SBUF, A      ;將 ACC 的內容複製到 SBUF
            JNB      RI, $         ;檢查是否完成接收?
                                   ;完成接收，則 RI 串列輸入中斷
            CLR      RI           ;清除 RI 旗標
            MOV      P0, SBUF     ;將所接收的資料輸出到 Port 0
            CLR      TI           ;清除 TI 旗標
            JMP      LOOP        ;跳至 LOOP 形成一個迴圈
            END

```

Mode 3 實驗(ch7-5.asm)

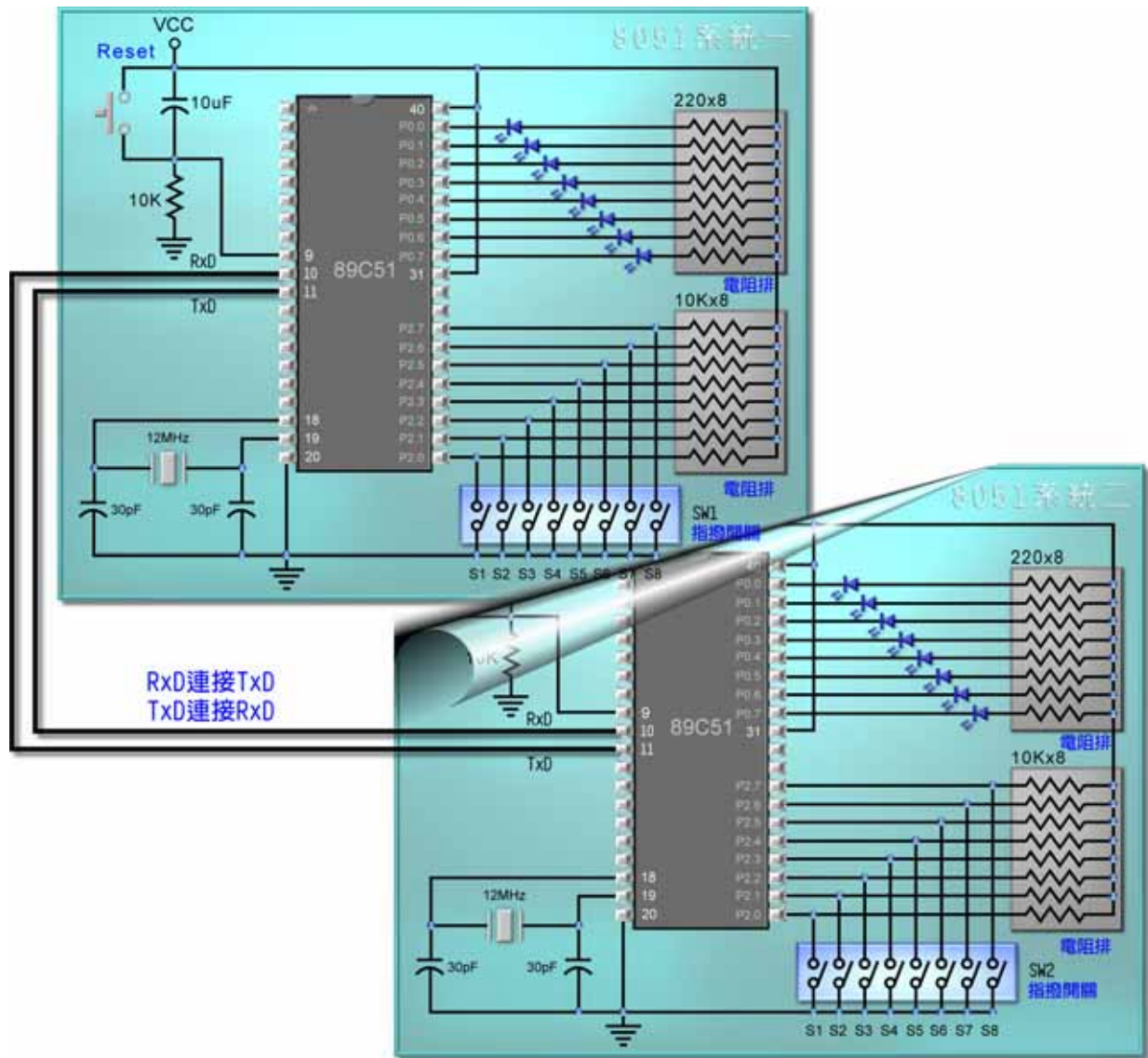
操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 16 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。



7-8-6

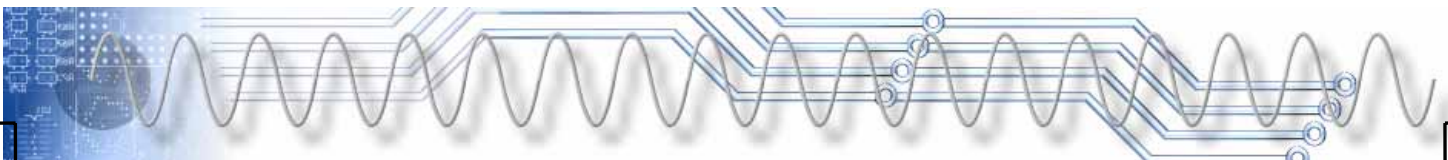
8051 vs 8051 實例演練



(圖 17) 電路圖

● 功能說明

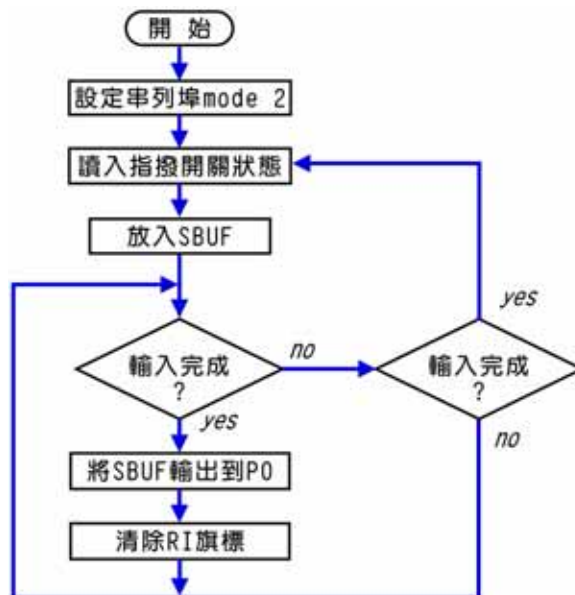
如圖 17 所示，「8051 系統一」與「8051 系統二」為兩個獨立的小系統，透過 Tx D 與 Rx D 傳輸線進行全雙工的資料傳輸，將 8051 系統一的指撥開關狀態，經串列埠傳輸到 8051 系統二 P0 上的 LED。同時，將 8051 系統二的指撥開關狀態，經串列埠傳輸到 8051 系統一 P0 上的 LED。



參考程式

根據 74165 的真值表(7-6 頁)，若要將並列資料載入 74165，則需給 74165 的第 1 腳(shift/load)一個負脈波，在此利用 P3.2 接腳連接到 74165 的第 1 腳，所以利用「CLR P3.2」將 P3.2 接腳變為低態，再以兩個「NOP」無作為指令多延遲 2 微秒，最後再以「SETB P3.2」將 P3.2 接腳恢復為高態。74165 載入 DIP SW 上的並列資料後，隨即依 8051 透過 TxD 接腳傳來的移位脈波，將資料一位元一位元地由 RxD 接腳傳入 8051。當 SBUF 滿了，即產生 RI 中斷，我們只要將 SBUF 緩衝器裡的資料，複製到 P0 即可。

在兩個 89C51 裡都必須存入下列程式：



```

START:      ORG      0           ;程式從 0 位址開始
            ORL      PCON, #80H   ;將 SMOD 設定為 1
            MOV      SCON, #10010000B ;設定為 mode 2、REN=1、RI=0
                                           ;採 OSC/32 鮑率
            ;====b7===b6===b5===b4===b3===b2===b1===b0===
            ;===SM0==SM1==SM2==REN==TB8==RB8==TI==RI==
            ;=====1=====0=====0=====1=====0=====0=====0=====0=====

LOOP:       MOV      P2, #FFH     ;將 P2 設定為輸入功能
            MOV      A, P2        ;讀取指撥開關狀態
            MOV      SBUF, A      ;將 ACC 的內容複製到 SBUF
            ;開始傳送資料

WAIT:       JBC      RI, RECEIVED ;檢查是否完成接收?
            ;若完成接收，則 RI 串列輸入中斷
            JBC      TI, LOOP     ;檢查是否完成傳送?
            ;若完成傳送，則重新開始
  
```

RECEIVED:	JMP	WAIT	;跳至 WAIT 繼續接收資料
	MOV	P0, SBUF	;將所接收的資料輸出到 Port 0
	CLR	RI	;清除 RI 旗標
	JMP	WAIT	;跳至 WAIT 繼續接收資料
	END		

8051 互傳實驗(ch7-6.asm)

● 操作

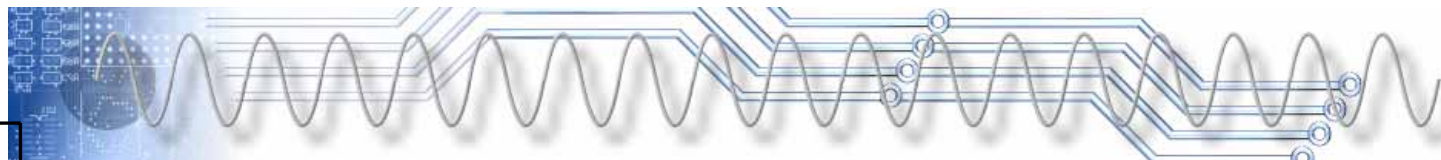
1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 17 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。

7-9

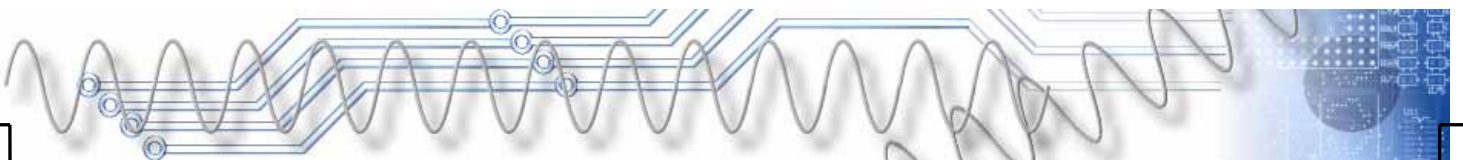
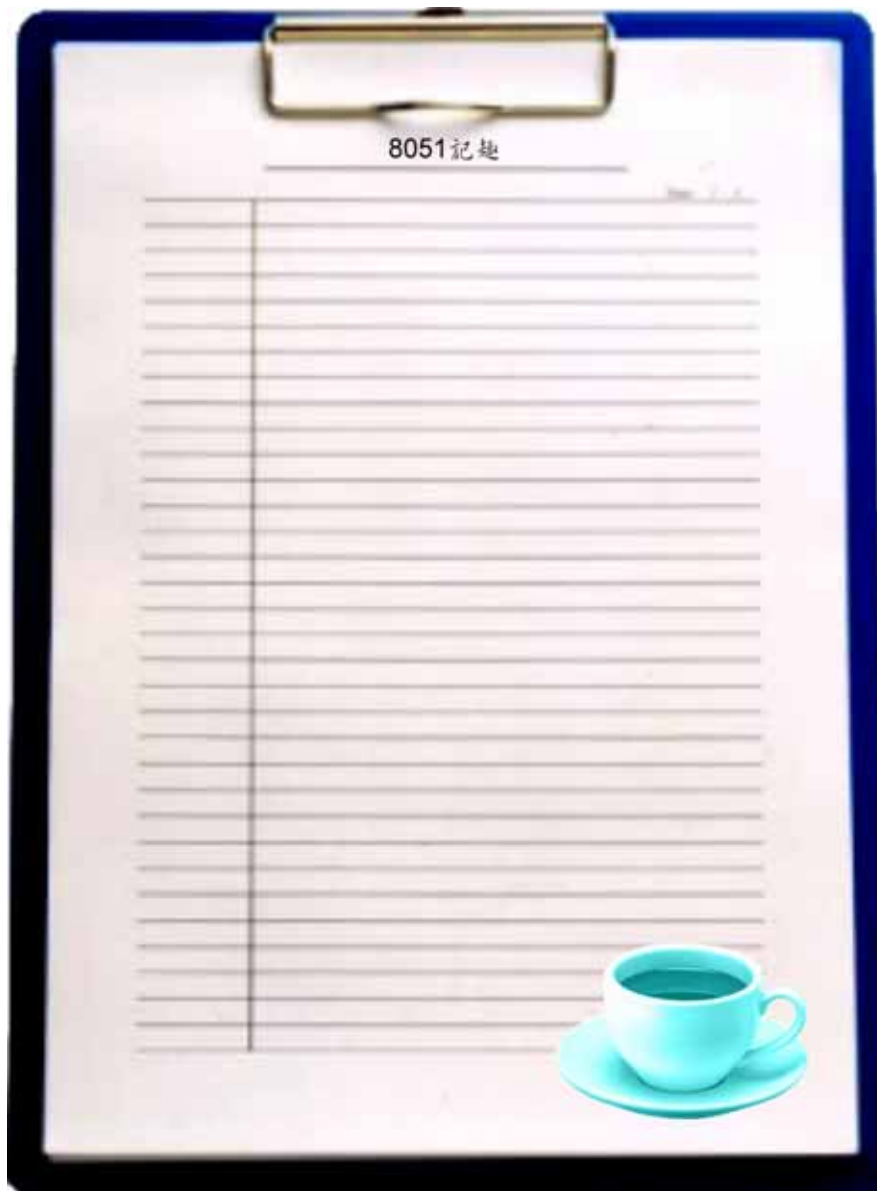
即時練習

在本章裡探討串列式資料傳輸的概念、8051 的串列埠、串列與並列資料的轉換 IC，以及 RS-232 的驅動 IC 等。在此請試著回答下列問題，以確認對於此部分的認識程度。

1. 試寫出 8051 串列埠四個工作模式的鮑率，及其設定方法？
2. 試說明 8051 串列埠 mode 2 與 mode 3 的差異？
3. 當使用 8051 串列埠，以 mode 0 工作模式將資料傳出時，其 TxD 接腳與 RxD 接腳各擔任何種任務？應如何接線？
4. 當使用 8051 串列埠，以 mode 0 工作模式接收資料時，其 TxD 接腳與 RxD 接腳各擔任何種任務？應如何接線？
5. 試說明 8051 串列埠 mode 1 的資料格式？
6. 試說明 8051 之 SCON 暫存器中，各位元的功能？
7. 試說明何謂「單工」、「半雙工」與「全雙工」？
8. 何謂「UART」？



9. 試簡述如何應用 74164 及 74165？
10. 試說明 MAX232 的功能？若要將 8051 系統的串列埠透過 MAX232 長距離傳輸，要如何連接？



心得筆記

Handwriting practice lines consisting of 18 horizontal lines, each starting with a blue dot on the left.

