

第 9 章

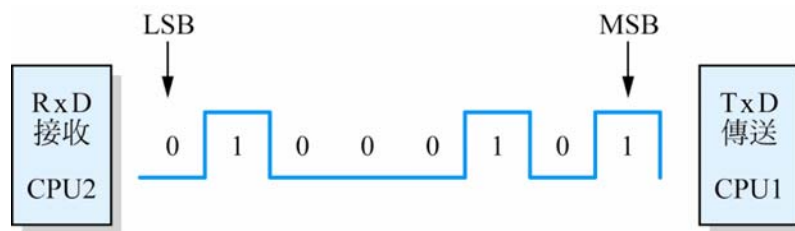
串列傳輸

7-1	UART 資料傳輸方式	7-2
7-2	UART 資料傳輸的同步問題	7-2
7-3	傳輸速率	7-3
7-4	UART 工作模式	7-3
7-5	UART 工作模式分析	7-5
7-6	UART 模式 0 分析	7-6
7-7	UART 模式 1 分析	7-11
7-8	UART 模式 2 分析	7-13
7-9	UART 模式 3 分析	7-15
7-10	各種模式的鮑率分析	7-15
7-11	讓 UART 正確工作的程式設定	7-17

MCS-51 內部提供的串列傳輸介面，是一種"非同步式串列資料傳輸" (Universal Asynchronous Receiver Transmitter，簡稱 UART)。因此我們都將 MCS-51 串列傳輸介面稱作 UART。

7-1 UART 資料傳輸方式

串列式傳輸是如何透過一條傳輸線將一筆資料傳送給對方的呢？方法是使用分時傳送方式，由傳送端每隔一段時間就將一位元的資料狀態傳送出去，直到這筆資料（8 位元）輸送完畢為止，如此即完成了一筆資料的傳送工作。而接收端也必須以相同的速度，以分時的方式一個位元接一個位元的讀入。如圖 7-1 所示，是一筆 8 bits 的串列資料傳送時，在傳輸線上所看到的脈波圖。

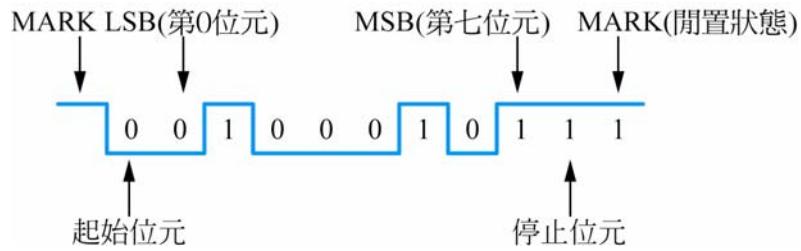


● 圖 7-1 串列傳送 10100010 資料時的脈波

7-2 UART 資料傳輸的同步問題

一筆資料假若如同上述所示，一個 bit 接一個 bit 傳送出去，而不加入作為同步的信號，則接收端要如何知道傳送端何時開始送資料，又怎麼知道資料已傳送完畢了？這就是接下來要講的 UART 的同步方式。

UART 的同步方式，是在 8 個資料位元的前面加上一個同步用的起始位元 (Start bit)，以及在 8 個資料位元的後面再加上另一個同步用的停止位元 (Stop bit)。並且規定起始位元為 "0"，停止位元為 "1"。若是我們將圖 7-1 中的 10100010 資料，以 UART 方式來傳送這一個 Byte 的標準樣式，將如圖 7-2 所示。



● 圖 7-2 以 UART 傳送資料 10100010 的串列脈波

在圖 7-2 中，我們可以看到 UART 每傳一個 Byte 需花 10 個 bit 的時間(8 個資料位元及 2 個同步位元)，其工作效率雖是不高的，但是卻大大提高了串列傳輸的可靠性。

當傳送端每傳完一個 Byte，就會因最後停止位元的關係而使得傳輸線維持"1"的狀態 (MARK)。因此要再傳另一個 Byte 時，傳送端會先送出一個起始位元"0"，接收端只要檢知傳輸線上的信號由"1"變為"0"，即表示傳送端又將傳送下一個 Byte 的第一個 bit 了。如此傳送端與接收端就靠這個起始位元與停止位元取得同步。每傳一個 Byte，共花 10 個 bit 的時間，就同步一次。

7-3 傳輸速率

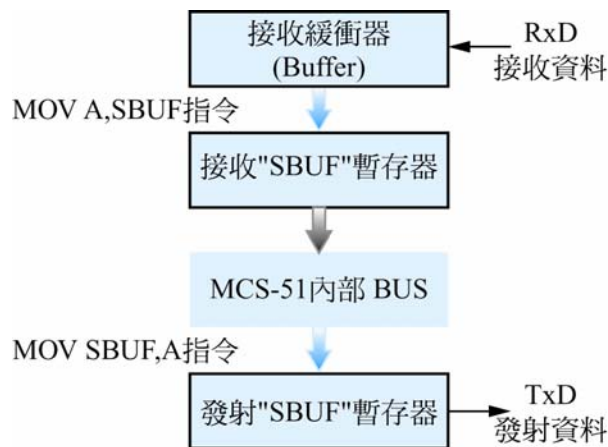
串列式傳輸是以分時的方式，將一個 bit 資料狀態("0"或"1")呈現在傳輸線上，如果這個 bit 在傳輸上所呈現的時間愈短，則資料傳輸的速度愈快，通常我們是以每秒傳幾個 bit 的方式來衡量傳輸速率，其單位為 bit/sec，我們稱之為位元率 (bit rate) 或鮑率 (Baudrate)。目前較常用的鮑率有 19200、9600、4800、2400、1200。鮑率在通訊協定上是一個很重要的參數。如果傳送端的傳送資料速度，與接收端的接收資料速度不一樣，那一定無法收到正確的資料。

7-4 UART 工作模式

UART 是一個全雙工串列埠，意思是說它可以在同一時間內進行發送與接收的工作。UART 的接收端具有緩衝器 (Buffer) 的功能，當 UART 接收到一個 Byte 的資料後，會將這個 Byte 放在緩衝器裡，然後繼續接收下一個 Byte 的資料。當第 2 個 Byte 被接收完畢時，若第一個 Byte 尚未被 CPU 提取，則第一個 Byte 資

料將會被覆蓋掉。所以，在第二個 Byte 被接收完畢前，就要把第一個 Byte 從緩衝器中提出來。

UART 與 CPU 之間的溝通都是靠 MCS-51 內部的 SCON 這個 8 位元暫存器。當 MCS-51 要透過 UART 以串列方式傳送一個 Byte 資料出去時，只要將這個 Byte 寫入 SBUF 暫存器中，UART 就會將這個 Byte 裡的 8 Bits 轉換成串列資料從 TxD 腳送出去。而 UART 則是透過 RxD 腳接收由外部送過來的串列資料，UART 的接收 Buffer 會將這些串列位元蒐集成一個 Byte，然後放到 SBUF 暫存器中等待 CPU 來讀取，如圖 7-3 所示。



● 圖 7-3 MCS-51 串列埠簡單結構圖

MCS-51 UART 的傳送與接收雖然都是使用 SBUF 暫存器，實際上傳送時所使用的 SBUF 與接收時所使用的 SBUF 是完全獨立的兩個暫存器。MCS-51 是將這兩個不同的暫存器放在相同的記憶位址上，並且使用相同的名稱。CPU 是利用讀取 (Read) 與寫入 (Write) 的動作來區分這兩個暫存器，指令 "MOV SBUF, A"，CPU 會發出寫入的信號，而將要傳送出去的資料放到傳送用的 SBUF。而指令 "MOV A, SBUF"，CPU 則會發出讀取的信號，去讀取接收用的 SBUF 資料，所以並不會互相干擾。

7-5 UART 工作模式分析

MCS-51 UART 提供四種工作模式，由設計者來自由使用。

1. 模式 0 (Mode 0)：I/O 擴充用通信模式。
2. 模式 1 (Mode 1)：10 位元可變鮑率 (Baudrate) 通信模式。
3. 模式 2 (Mode 2)：11 位元固定鮑率 (Baudrate) 通信模式。
4. 模式 3 (Mode 3)：11 位元可變鮑率 (Baudrate) 通信模式。

以上這四種串列埠是由 SCON(Serial Port Control Register) 串列埠控制暫存器來選擇模態及操作控制。

串列埠控制暫存器 (SERIAL PORT CONTROL REGISTER 可位元定址)

SCON：

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

SM0：串列埠模式選擇 0

SM1：串列埠模式選擇 1

SM0	SM1	模式	功 能	鮑 率
0	0	0	移位暫存器	$F_{osc}/12$
0	1	1	8 位元之 UART	可軟體規劃
1	0	2	9 位元之 UART	$F_{osc}/64$ 或 $F_{osc}/32$
1	1	3	9 位元之 UART	可軟體規劃

※ F_{osc} ：外接石英晶體振盪頻率

SM2：1. 在模式 0 時，SM2 必須清除為 0。

2. 在模式 1 時，若 SM2=1，則必須接收到有效的停止位元時，硬體才會設定接收中斷旗標 RI=1。

3. 在模式 2 或模式 3 時，若 SM2=1 則必須所接收到的第 9 個位元 RB8=1，硬體才會設定 RI=1。此功能大多使用在多工處理器通信功能上。

REN：串列埠接收致能位元

1. REN=1 則致能接收動作，令 REN=0 則抑制接收動作。

2. 此位元用指令設定或清除。

TB8：在模式 2 或 3 時，此位元被當作第 9 個資料位元傳送出去。

RB8：1. 在模式 0 時，此位元未被使用。

2. 在模式 1 時，若 SM2=1，接收到的"停止位元"會自動存入 RB8。

3. 在模式 2 或模式 3 中，接收到的"第 9 位元"會自動存入 RB8。

TI：發射中斷旗標

1. 在模式 0 時，當傳送出第 8 個資料位元結束時，硬體會設定 TI=1。

2. 在 1、2、3 模式時，當停止位元傳送出去後，硬體會設定 TI=1。

RI：接收中斷旗標

1. 在模式 0 中，接收到最後一個位元(bit 7)後，硬體會設 RI=1。

2. 在 1、2、3 模式，當接收到停止位元時，硬體會設 RI=1。

7-6 UART 模式 0 分析

1. 模式 0 的鮑率

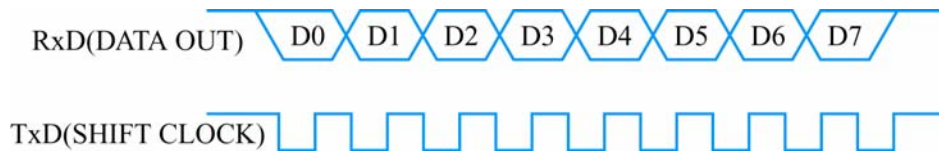
在 Mode 0 之工作模式下，串列埠資料的傳送與接收都是經由 "RxD" 端，TxD 腳則負責輸出移位用的脈波。每次發送與接收都是以 8 個位元為一單位（沒有起始位元與結束位元），最先發送出去的是 D0(LSB) 然後依序至 D7(MSB)。而發送與接收之鮑率與移位脈波的速率固定為振盪器頻率÷12。如果振盪器頻率是 12MHz，則發送接收之鮑率 (Baud rate) 與 TxD 腳發出的移位脈波速度為 1MHz。

2. 模式 0 的資料傳送

如圖 7-4 所示，是 UART 工作於 Mode 0 時，傳送 8 個 bit 資料到外部的時序圖。當您設定 UART 工作模式為 Mode 0 後，只要將要傳到外部的 8 Bit 資料寫入 SBUF，UART 就會將這個 Byte 轉換成如圖 7-4 的串列脈波送出去。歸納而言，UART 模態 0 發送資料的處理過程如下：

- (1) 設定 UART 模式 0 及令暫存器 SCON 內之 TI 位元為 0。
- (2) 將欲送出去的資料寫入至發射 SBUF 內去。

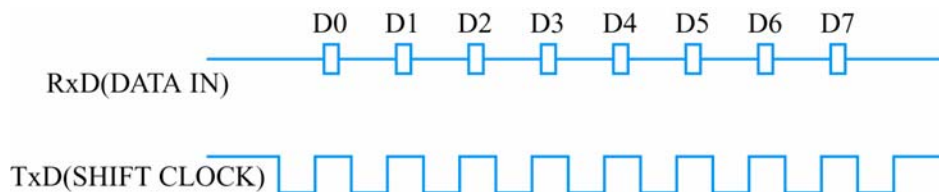
- (3) 當發送完畢後，旗號 TI 會被設定為 1。如果在允許中斷情況之下，此 TI =1 將會向 CPU 要求中斷，而去執行中斷服務程式。CPU 也可經由檢查旗號 TI 是否已經為 1，若為 1 則表示先前載入到 SBUF 內的資料已傳送完畢，CPU 將可以繼續把下一個 Byte 的資料送到 SBUF 內去傳送。



● 圖 7-4 Mode 0 的傳送資料時序圖

3. 模式 0 的資料接收

UART 工作在 Mode 0 時，要經由 RxD 腳接收 8 個位元的資料到 SBUF 暫存器裡去，只要設定 REN=1、RI=0，UART 就會產生如圖 7-5 的脈波時序圖，並且到外部讀入 8 位元資料。當 UART 從 RxD 腳讀完 8 個位元之後，會將 RI 設為 1。因此 CPU 只要去檢查 RI 位元，若 RI=1 則表示已經讀完 8 位元的資料放在 SBUF 暫存器了。



● 圖 7-5 Mode 0 的資料輸入時序圖

4. 輸出入埠的擴展

UART 的 Mode 0 主要用途是在於擴充輸出入埠。

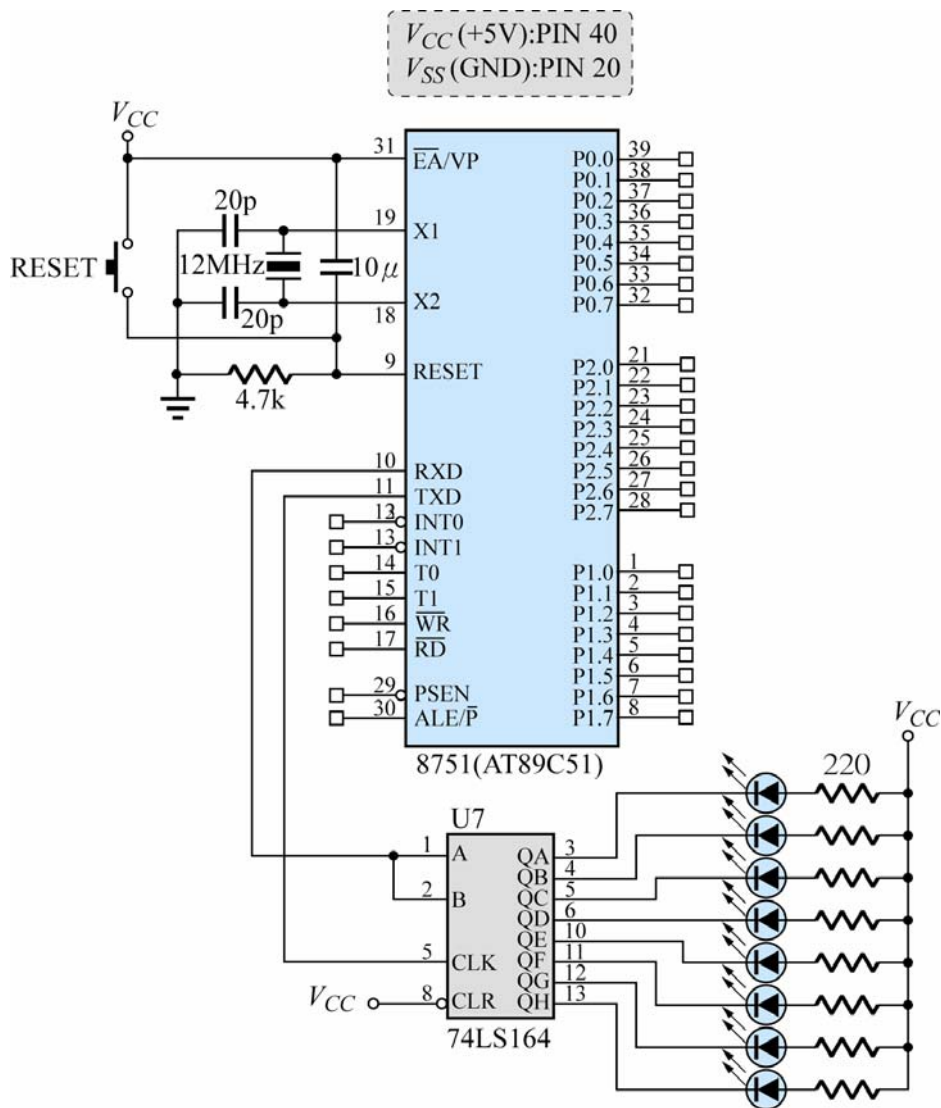
輸出擴展

如圖 7-6 我們只要將 TxD 接到外部 8 位元 SIPO (串列輸入並列輸出) IC 的 Clock 腳，及 RxD 接到 SIPO IC 的串列資料輸入端，就可以利用 UART 移位脈衝的配合，將 RxD 腳準備輸出的資料傳給 SIPO。SIPO 在 TTL IC 族裡有 74LS164。

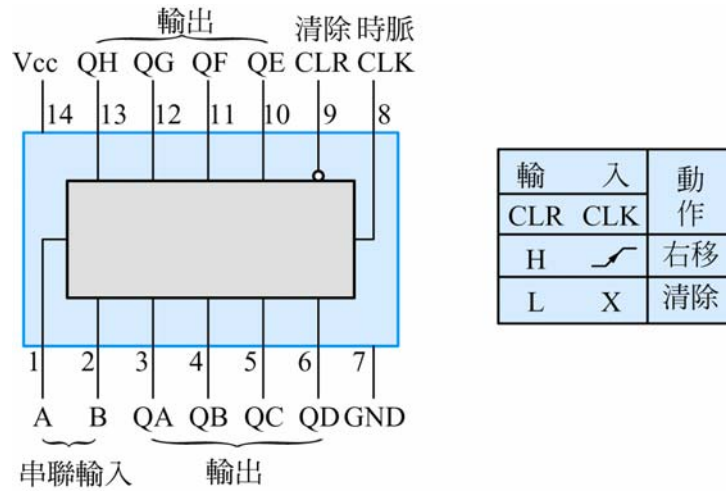
UART 模式 0 接收資料的處理過程如下：

- (1) 設定 UART 模式 0 及令暫存器 SCON 內的 RI 位元為 0。
- (2) 設定 SCON 暫存器內的 REN 位元為 0，啟動接收狀態。
- (3) 當接收完 1Byte 的資料，旗號 RI 會被設定為 1。

如果在允許中斷情況下，此 RI=1 將會向 CPU 要求中斷，而去執行中斷服務程式。CPU 也可經由檢查訊號 RI 是否已經為 1，若為 1 則表示 SBUF 內已接收完成 1Byte 資料。



● 圖 7-6 Mode 0 的輸出埠擴展



● 圖 7-7 TTL 74LS164 資料

輸入擴展

我們可使用一個 8 位元並列載入串列輸出 IC(PISO)，將 TxD (移位脈波輸出腳) 接至 PISO IC 的 Clock，將 RxD (資料接收腳) 接至 PISO IC 的資料輸出腳 QH。如此就可以使用 UART 的串列輸入移位脈衝，到外部去讀入一個 8 bit 的資料至 SBUF。PISO 在 TTL IC 族裡有 74LS165。74LS165 的第 1 支腳 SH/LD，是作為資料載入或移位使用。

7-7 UART 模式 1 分析

在 MCS-51 系統中，真正做串列通訊用的是在 Mode 1。在模式 1 中，傳送端是 TxD，接收端則是 RxD。每次發送或接收以 10 個 bit 為一單位，其中包含 1 個起始位元 ("0")，8 個資料位元 (D7~D0) D0 最先發送出去，及一個停止位元 ("1")。如圖 7-10 所示為 Mode 1 傳送/接收的串列資料格式。



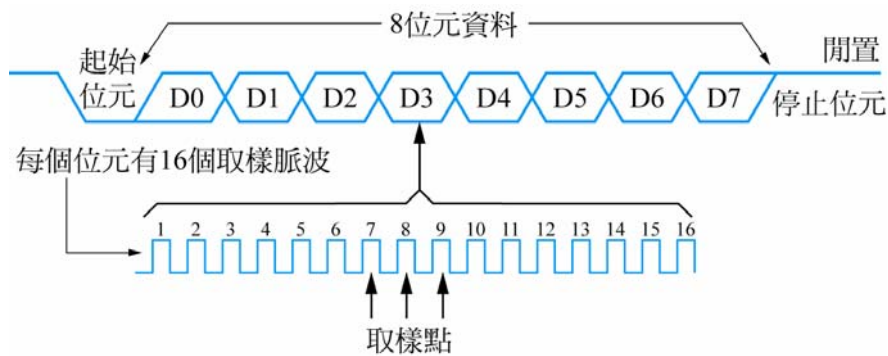
● 圖 7-10 Mode 1 傳送/接收的串列資料格式

1. 模式 1 的資料傳送

當 UART 的工作模式設定在 Mode 1 時，當您要將一個 Byte 的資料透過 UART 傳送出去時，您只須將這個 Byte 寫入 SBUF 暫存器，UART 就會自行將這個 Byte 資料轉換成圖 7-10 的串列脈波從 TxD 腳輸出。

2. 模式 1 的資料接收

當我們透過 UART 欲從 RxD 端接收外面資料時，當接收端檢測到 RxD 接腳上有 1→0 的變化 (Start bit)，就知道即將有 8 bit 的串列資料要輸入。無論鮑率值是多少，從 RxD 端輸入的每一個位元都將被取樣 (Sample) 16 次，但真正取樣有效的是第 7、8、9 次的值。因為第 7、8、9 次正好是該位元的中央處，誤差影響會較小。而真正確認的則是根據此三次中的多數 (例如若取樣是 110 或 101 則確認為 1，若是 010 或 000 則確認為 0)。如此做的目的，是為了避免外界雜訊的影響，並可提高雜訊免疫能力，如圖 7-11 所示。



● 圖 7-11 Mode 1 的串列資料接收取樣時序圖

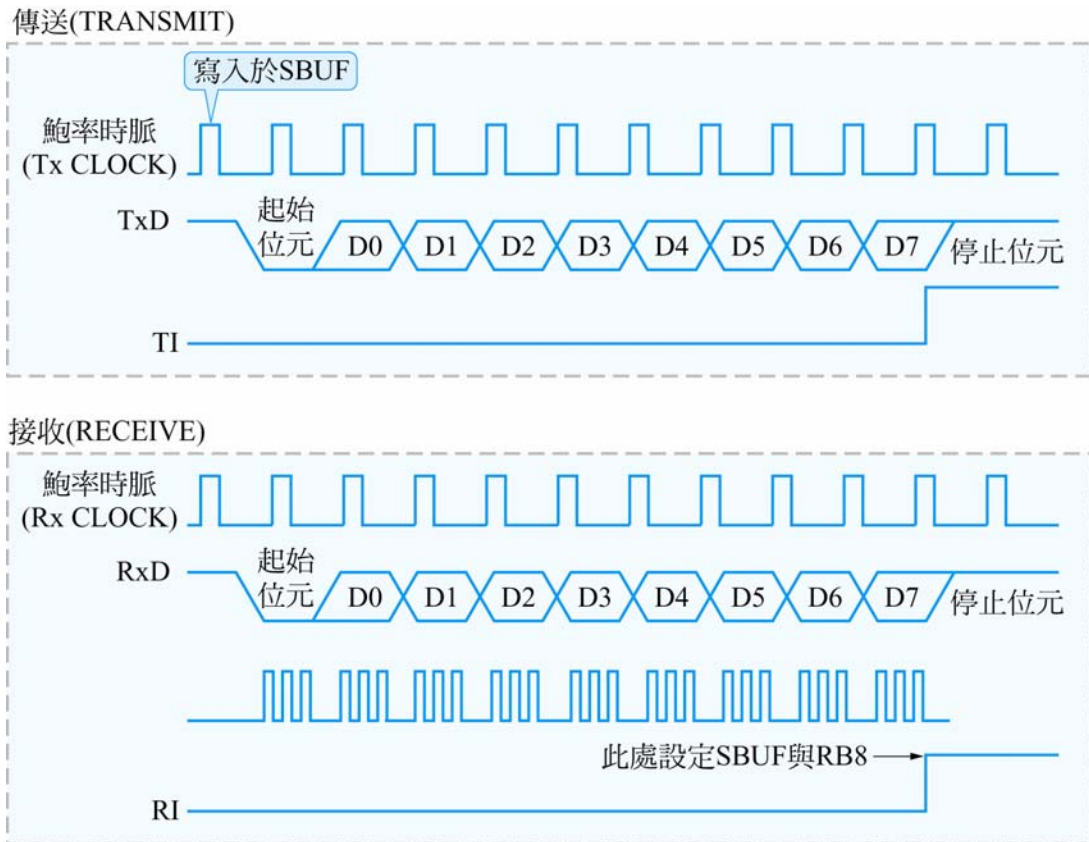
UART 在 Mode 1 的串列埠接收資料時，會將停止位元載入儲存位於 SCON 暫存器中的 RB8 位元。8 個資料位元則載入到 SBUF，形成一個 Byte 的資料。且將 SCON 暫存器中的 RI 旗號設定為 1，等待 CPU 來讀取，因此 CPU 只要檢查 RI 位元就可以決定 SBUF 暫存器中的內容是否有效。

所接收到的資料是否有效，要視下列兩個條件而定，若有一項不成立，則串列埠所接收到的資料將被放棄。

- (1) RI=0（接收前必須先以程式清除為 0）。
- (2) 若 SM2=1，則接收停止位元=1。
若 SM2=0，則無接收停止位元考量。

3. 模式 1 的飽率

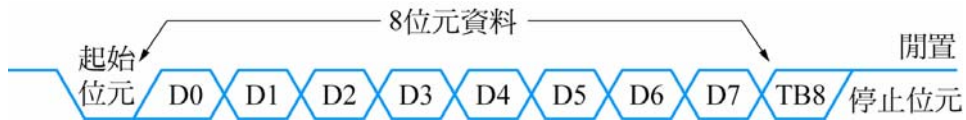
模式 1 的飽率是由計時／計數器 1 (Timer 1) 的溢位率所控制，因此可由程式設計者自行來規劃設定。



● 圖 7-12 Mode 1 的串列傳送／接收時序圖

7-8 UART 模式 2 分析

模式 2 的發送資料端是 TxD，接收資料端是 RxD，每次以 11 個位元為一單位，其中包括一個起始位元 "0"，8 個資料位元 (D0 最先發送出去或最先接收進來)，1 個可由程式設計者自行設定其值為 1 或 0 的第 9 資料位元 TB8，以及 1 個停止位元 "1"。其傳送／接收的串列資料格式如圖 7-13 所示。



● 圖 7-13 Mode 2 的串列資料格式圖

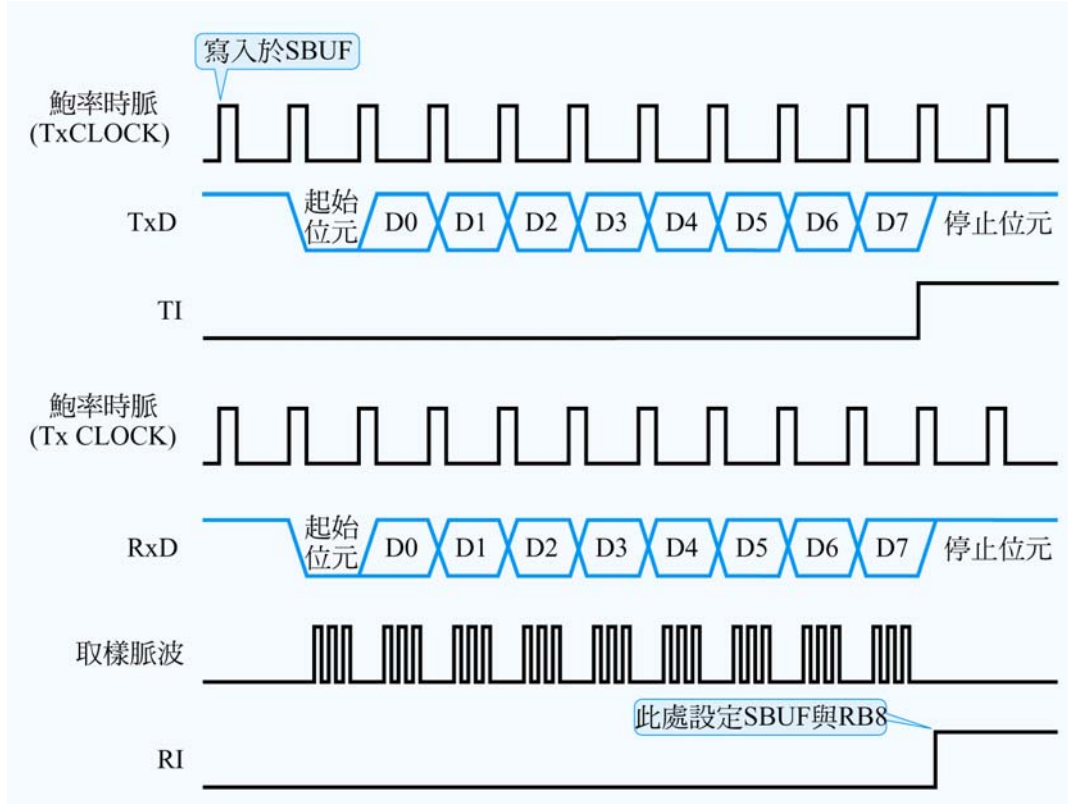
1. 模式 2 的資料傳送與接收

模式 2 的資料傳送與接收方式其實與模式 1 大致相同，只是模式 2 的傳送/接收資料位元中多了一個 TB8 或 RB8 而已。

當傳送時，這第 9 個資料位元（存於 SCON 的 TB8）可以視需要將其設定為 1 或者為 0。TB8 位元是 MCS-51 為了利用 UART 作多個 CPU 之間的通信所設計的一個特殊位元，可經由此位元的狀態來判斷所接收的 D0~D7 是屬於位址或者是資料（使用在多工處理器系統中）。若是您不做多工處理器通信時，TB8 可以用來當作同位位元 (Parity) 使用。例如在程式狀態字元 PSW 中的極性位元 P，我們就可以移入到 TB8 中，以作為接收端用來驗證傳送途中極性是否正確之用。當接收時，對方傳送過來的 TB8 位元，將會被存到 SCON 暫存器的 RB8 位元中。

2. 模式 2 的鮑率

在此模式 2 時，有關鮑率的速率，可以規劃為振盪頻率值的 1/32 或 1/64。



● 圖 7-14 Mode 2 的串列傳送/接收時序圖

7-9 UART 模式 3 分析

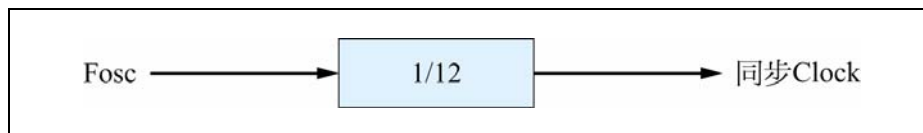
模式 3 的發送端是 "TxD"，接受端是 "RxD"，每一筆傳送碼是 11 個位元，其中包括 1 個起始位元，8 個資料位元，1 個可程式規劃設定之 "TB8" 位元，以及 1 個停止位元。模式 3 與模式 2 大致相同，唯一不同點是模式 2 的鮑率是固定的，而模式 3 的鮑率，與模式 1 一樣，是由 Timer 1 的溢位率決定，可由程式設計者自行規劃。

7-10 各種模式的鮑率分析

1. Mode 0 的鮑率

Mode 0 的鮑率是固定的，為振盪器頻率÷12。例如：MCS-51 的振盪器工作頻率使用 12MHz，則 Mode 0 的鮑率為 $12\text{MHz} \div 12 = 1\text{M bit/sec}$ ，即每秒鐘傳送 10^6 個 bit。

模式 0 的鮑率 = $f_{osc} / 12$ f_{osc} ：外部振盪頻率

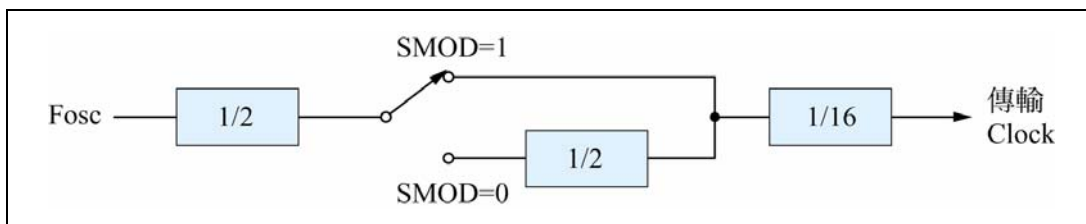


2. Mode 2 的鮑率

Mode 2 的鮑率值與 PCON 中 "SMOD" 位元有關，



Mode 2 的鮑率 = $2^{SMOD} / 64 \times (f_{osc})$ f_{osc} ：外部振盪頻率

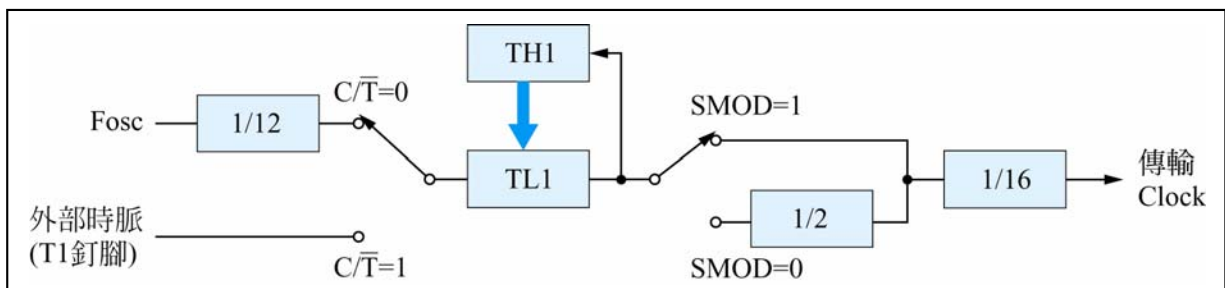


例如：石英晶體振盪器頻率為 12MHz，SMOD=1，
則 Mode 2 的鮑率為 $2^{12}/64 \times (12\text{MHz}) = 375\text{K}(\text{bit}/\text{sec})$

3. Mode 1 及 Mode 3 的鮑率

在串列埠規劃成模式 1 或模式 3 時，其鮑率將由計時器 1 來產生，因而鮑率值的大小，將視計時器 1 的溢位率以及位元 "SMOD" 的值來決定。其計算公式如下：

Mode 1 或 3 的鮑率 = $2^{\text{SMOD}}/32 \times (\text{計時器 1 的溢位率})$



由上面公式可知，Mode 1 或 Mode 3 的鮑率值，實際上是在控制 Timer 1 的計時時間而已，此時可採用 Timer 1 中的 Mode 0、1、2 中任何一種模式工作之。但在實際應用上則採用具有自動載入功能的 Mode 2 模式，使用上較為簡單。由此得知 UART 的 Mode 1 或 3 的鮑率計算公式又可寫成如下：

$$\text{Mode 1 或 3 的鮑率} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{fosc}}{12 \times [256 - (\text{TH1})]} \quad \text{fosc : 外部振盪頻率}$$

通常我們往往是先曉得鮑率值是多少，然後才去找出 TH1 的值，再由已找出的 TH1 值，以程式寫入 Timer 1 的 TH1 暫存器中。將上式的公式整理一下，就可得如下求 TH1 的公式：

$$\text{TH1} = 256 - \frac{2^{\text{SMOD}} \times \text{fosc}}{384 \times \text{鮑率}}$$

例如：我們使用的振盪頻率是 11.0592MHz，要得到 2400 的鮑率時

$$\begin{aligned} \text{TH1} &= 256 - \frac{2^0 \times 11.0592 \times 10}{384 \times 2400} ; \text{SMOD}=0 \\ &= 256 - 12 = 244 = \text{F4H} \end{aligned}$$

利用上式的公式，我們可以找出一些常用的鮑率與系統振盪頻率值 f_{osc} ；
Timer 1 中 TH1 及位元 "SMOD" 相互之間的關係表如下：

表 7-1 計時器 1(Timer 1) 常用鮑率所需的設定表

鮑率 (Baud Rate)	Fosc (晶體振盪頻率)	SMOD	Timer 1		
			C/T	模式	自動載入值 (TH1)
Mode 0 最大：1MHz	12 MHz	X	X	X	X
Mode 2 最大：375K	12 MHz	1	X	X	X
Mode 1 或 3：62.5K	12 MHz	1	0	2	FFH(255)
19.2K	11.059 MHz	1	0	2	FDH(253)
9.6K	11.059 MHz	0	0	2	FDH(253)
4.8K	11.059 MHz	0	0	2	FAH(250)
2.4K	11.059 MHz	0	0	2	F4H(244)
1.2K	11.059 MHz	0	0	2	E8H(232)
137.5	11.059 MHz	0	0	2	1DH(29)
110	6 MHz	0	0	2	72H(114)
110	12 MHz	0	0	1	FFEBH

※灰底部分較常用

7-11 UART 正確工作的程式設定

讓 MCS-51 的 UART 正確工作的程式設定有下列二項：

1. 設定 UART 的工作模式（規劃 SCON）。
2. 設定 UART 的傳送／接收速率（鮑率）。

1. 設定 UART 的工作模式

MCS-51 的 UART 工作模式，是透過規劃 SCON 暫存器的 SM0 和 SM1 位元所加以設定的。

SM1	SM0	模 式
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Mode 3

如果我們是要利用 UART 的 Mode 1 將外部資料接收進來，則我們必須設定 REN=1，以致能 UART 的 RxD 腳，如此即可接收外部輸入的串列脈波，其程式如下：

```
MOV  SCON, #01010000B
```

2. 設定 UART 的傳送/接收速率

規劃好 UART 的工作模式之後，接下來就是要設定 UART 的傳送/接收速率，也就是鮑率。在前面的鮑率分析中提過，UART 若工作在 Mode 0 或 Mode 2 時，其鮑率是固定的，而工作在 Mode 1 或 Mode 3 時，其鮑率是由 Timer 1 的溢位率來控制，也就是要計算出正確的 TH1 來（一般 Timer 1 是工作在模式 2 的自動重新載入的模式下）。例如我們延續上面的 UART Mode 1 的規劃，設定其鮑率為 9600bit/sec，且採用 12MHz 的石英晶體振盪器，則由表 7-1 中對照查出，應將 Timer 1 的 TH1 暫存器設定為 253(FDH)。

程式設計如下：

Initial_UART:

```
MOV  SCON, #01000000B    ; 設定 UART 工作在 Mode 1
MOV  TMOD, #00100000B    ; 設定 Timer 1 工作在模式 2
MOV  TL0, #253           ;
MOV  TH1, #253           ; 設定鮑率為 9600 bit/sec
SETB TR1                 ; 啓動 Timer 1
RET
```

以上所列 Initial_UART 副程式，是一個 UART 的啟動程式，只要執行完以上的程式，MCS-51 就會處在一個等待傳送和接收的狀態。如果您要透過 UART 傳資料出去時，只要將資料透過累加器 ACC 寫入到 SBUF 暫存器中，UART 就會將這個資料暫存成 Mode 1 的 10bit 串列資料格式，由 1 個 Start bit、8 個 data bit 和 1 個 Stop bit 組成，透過 TxD 傳送出去。當 UART 傳送完這 10 bit 的串列資料之後，會主動地將 SCON 暫存器的 TI 位元設定為 1，以告訴 MCS-51 的 CPU，剛剛傳送的那筆資料已傳送完畢，可以再送另一筆資料來了。

```
MOV  SBUF,A              ; 將資料透過 SBUF 暫存器傳送出去
```

當 MCS-51 處在一個等待接收的狀態時，UART 會自動地去檢測 RxD 串列資料輸入腳，當它檢知 RxD 接腳上有輸入 1→0 信號 (Start bit) 時，就知道外界已開始要傳送資料進來，RxD 開始讀入 8 bit 資料，存入 SBUF 暫存器中。且將 SCON 暫存器中的 RI 位元設為 1，以告訴 CPU，UART 已接收一 Byte 資料存入 SBUF 中，可以將它取走以便繼續接收下一筆資料。若 CPU 在第二筆資料接收完畢存入 SBUF 之前，未將第一筆 SBUF 中的資料取走的話，第二筆資料將會覆蓋掉第一筆資料，存入 SBUF 暫存器中。CPU 是靠著判斷 RI 位元而得知 UART 是否接收到資料的。

MOV A,SBUF ; 將 RxD 腳接收到的資料經由 ACC 取出

練習 1 ... UART 的 Mode 0 實驗—輸出埠擴充

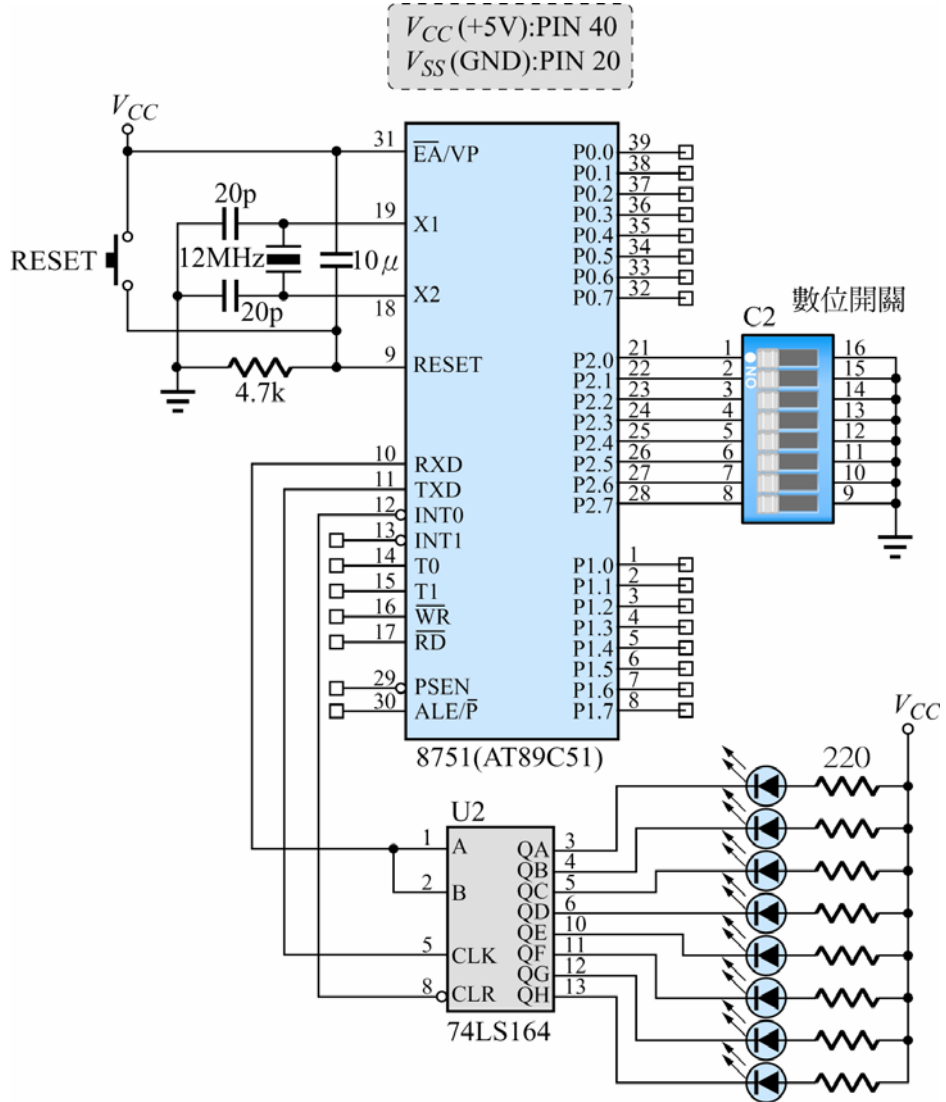
1. 學習目的

學習 UART 的 Mode 0 工作模式，在外部擴充一個 8 位元輸出埠。

2. 動作說明

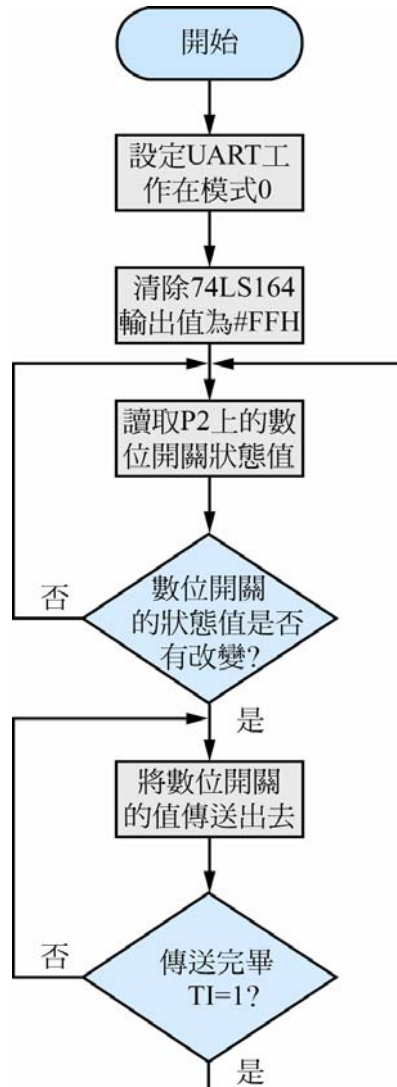
利用 TTL 的 74LS164 與 MCS-51 UART 的 Mode 0 來擴充一個 8 位元的外部輸出埠。接在 P2 上的數位開關當作資料輸入，UART 的 Mode 0 透過 RxD 這支腳傳出數位開關的狀態至 74LS164 的資料輸入端 A、B，並且利用 TxD 這支接腳輸出移位用的脈波至 74LS164 的 CLOCK 輸入端，並將資料輸入端的資料依序地由 QA~QH 輸出端輸出，顯示在 LED 上。

3. 電路圖



● 圖 7-15 UART 的 Mode 0 實驗—輸出埠擴充電路圖

4. 流程圖



● 圖 7-16 程式 P07-1 流程圖

5. 程式及程式說明

；程式檔名：P07-1.asm

```

org      0
jmp      Start
Start:
mov      sp,#6fh
mov      scon,#00000000B    ；設定 UART 的工作模式為 Mode 0
clr      p3.2                ；輸出一個負脈衝信號以便清除 IC 74LS164
    
```

```

        setb    p3.2          ; 的輸出動作
        mov     r3,#0ffh     ; 設定輸入值的初始值為 FFH

Loop:
        mov     a,p2         ; 輸入數位開關的值，並存入 A 中
        xrl    a,r3         ; 檢查輸入值是否和上一次輸入值相同
        jz     Loop         ; 若改變(r3 ≠ A)則往下執行，否(r3=A)
                                ; 則跳至 Loop
        mov     r3,p2       ; 將最新的輸入值存入 r3
        mov     a,p2       ; 取數位開關內容值，並將它反相，以配
        cpl    a           ; 合 P0 上 LED 的顯示 (LED 亮代表 0，LED
                                ; 滅代表 1)
        mov     sbuf,a      ; 將 A 的內容放到 SBUF 經由 TxD 腳傳
                                ; 送出去

UART_Wait:
        jbc    ti,UART_Ok   ; 去檢查 UART 是否已將 SBUF 暫存器中
                                ; 的 8 Bit 資料傳送完畢，若還沒有傳送完
                                ; 畢則繼續 (此時 TI=0)，等到傳送完畢
                                ; (TI=1)，則將 TI 清除為 0 且跳至位址
                                ; UART_Ok
        jmp    UART_Wait

UART_Ok:
        jmp    Loop        ; 回到 Loop 繼續下一次的傳送
    .end

```

6. 討 論

當 UART 將 8 位元的資料傳送完畢後會將 TI 位元設為 1，當 CPU 檢知到 TI=1 時，就知道上一位元組已傳送完畢，可以繼續傳送下一個位元組資料了。但要注意的是，當 UART 傳送完 8 位元資料而設定 TI=1 之後，TI 並不會被自動清除。因此要繼續傳送下一位元組資料前，應先將 TI 清除為 0，以避免在傳送下一位元組資料時被誤認為資料已傳送完畢。

練習 2... UART 的 Mode 0 實驗－輸入埠擴充

1. 學習目的

學習 UART 的 Mode 0 工作模式，及在外部擴充一個 8 位元輸入埠。

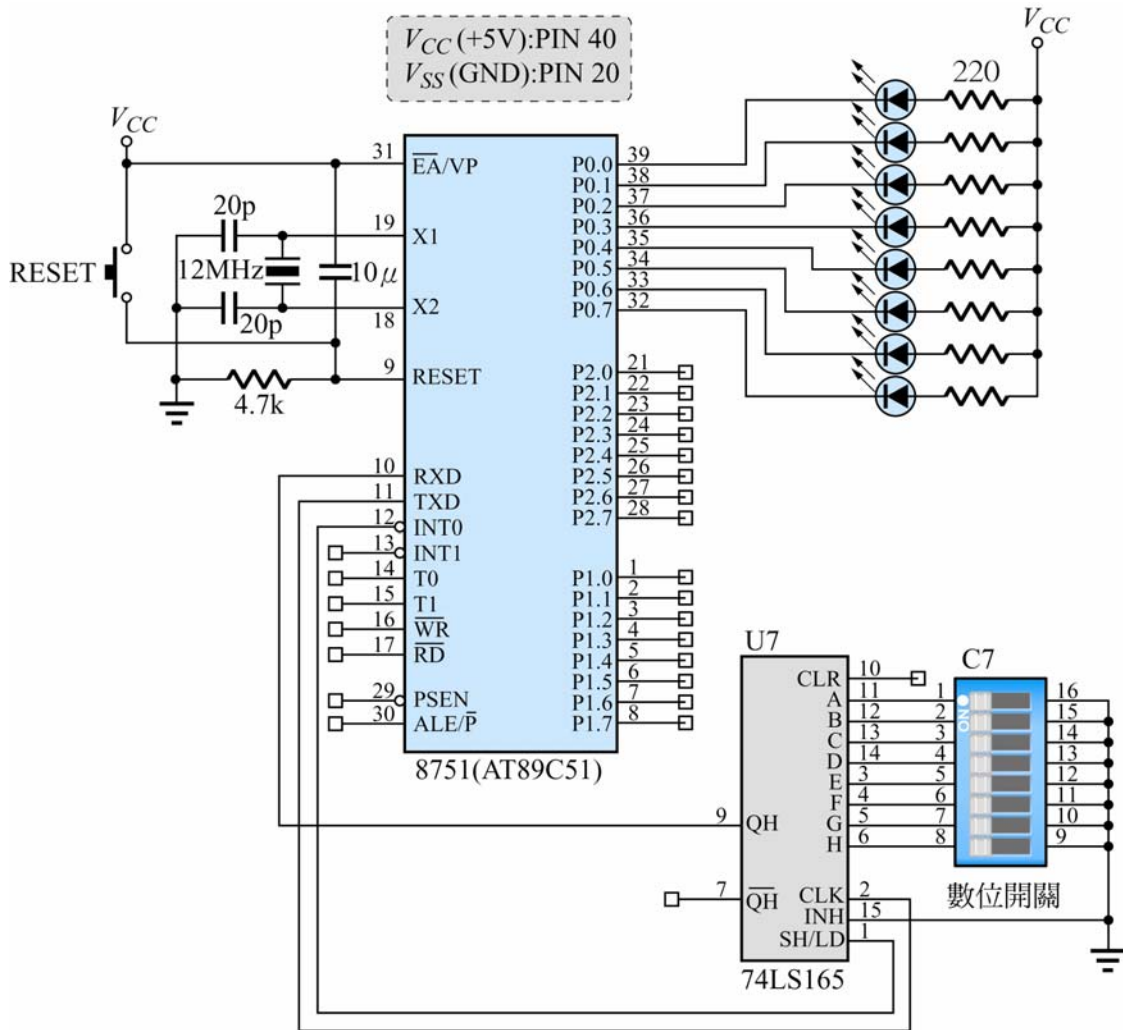
2. 功能說明

利用 TTL 的 74LS165 與 MCS-51 UART 的 Mode 0 來擴充一個 8 位元的外部輸入埠。將接在輸入擴充埠 74LS165 上的 8 位元數位開關狀態資料讀入，再將此狀態顯示到 P0 的 LED 上。

如電路圖所示，要將並入串出 (PISO) 移位暫存器 74LS165 輸入端的 8 個數位開關狀態資料讀入須有以下的步驟：

- (1) 輸出一個 "↓" Low 信號到 74LS165 的 $\overline{SH/LD}$ 腳 (由 P3.2 輸出)，將數位開關的狀態資料載入至 74LS165 的內部暫存器。
- (2) 啟動 UART，利用 TxD 腳輸出移位脈波接至 74LS165 的 CLOCK 輸入腳 CLK；將存於 74LS165 內部暫存器的資料依序由 QH 腳輸出至 UART 之 RxD 腳。

3. 電路圖



● 圖 7-17 UART 的 Mode 0 實驗－輸入埠擴充

4. 程式及程式說明

；程式檔名：P07-2.asm

```
org 0
```

```
jmp Start
```

Start:

```
mov sp,#6fh
```

```
mov scon,#00010000B ;設定 UART 的工作模式為 Mode 0
```

```
;(REN=1, 致能接收器)
```



```

    clr    ri                ; 啓動 UART 的讀入接收動作，開始讀
                          ; 入 74LS165 暫存器中的資料

Loop:
    clr    p3.2            ; 輸出一個負脈衝信號到 74LS165 的
                          ; SH/LD 腳，以便將數位開關的值載
    setb   p3.2            ; 入至 74LS165 內部暫存器中載入完
                          ; 成後開始做移位(Shift)動作

Receive_Wait:
    Jbc    ri,Receive_Ok  ; 檢查 UART 是否已將 8 Bit 資料讀入
                          ; (接收) 完畢，若還沒有讀入完畢則繼續
                          ; (此時 RI=0)，等到讀入完畢(RI=1)，則跳
                          ; 至位址 Receive_Ok，並且將 RI 清除為 0

    jmp    Receive_Wait

Receive_Ok:
    mov    a,sbuf          ; 將接收進來的資料反相，以配合 P0 上
    cpl    a               ; 的 LED 顯示 (LED 亮代表 0，LED 滅代
                          ; 表 1)

    mov    p0,a
    mov    r7,#5           ;
    call   Delay           ; 延時
    jmp    Loop            ; 回到 Loop 繼續下一次的傳送

Delay:
    mov    r6,#250
    djnz   r6,$
    djnz   r6,Delay
    ret
.end

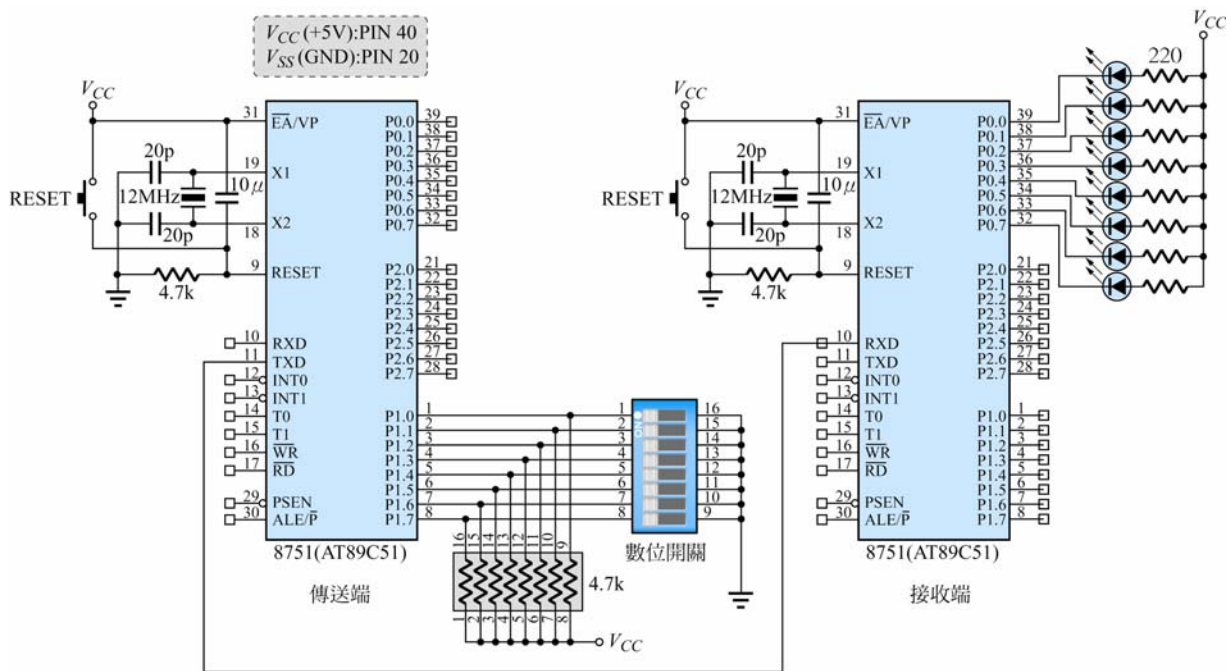
```

練習 3... UART 的 Mode 1 實驗－兩個 MCS-51 間資料的單向傳輸實習

1. 學習目的：了解 UART 的 Mode 1 工作模式及使用情形。
2. 功能說明

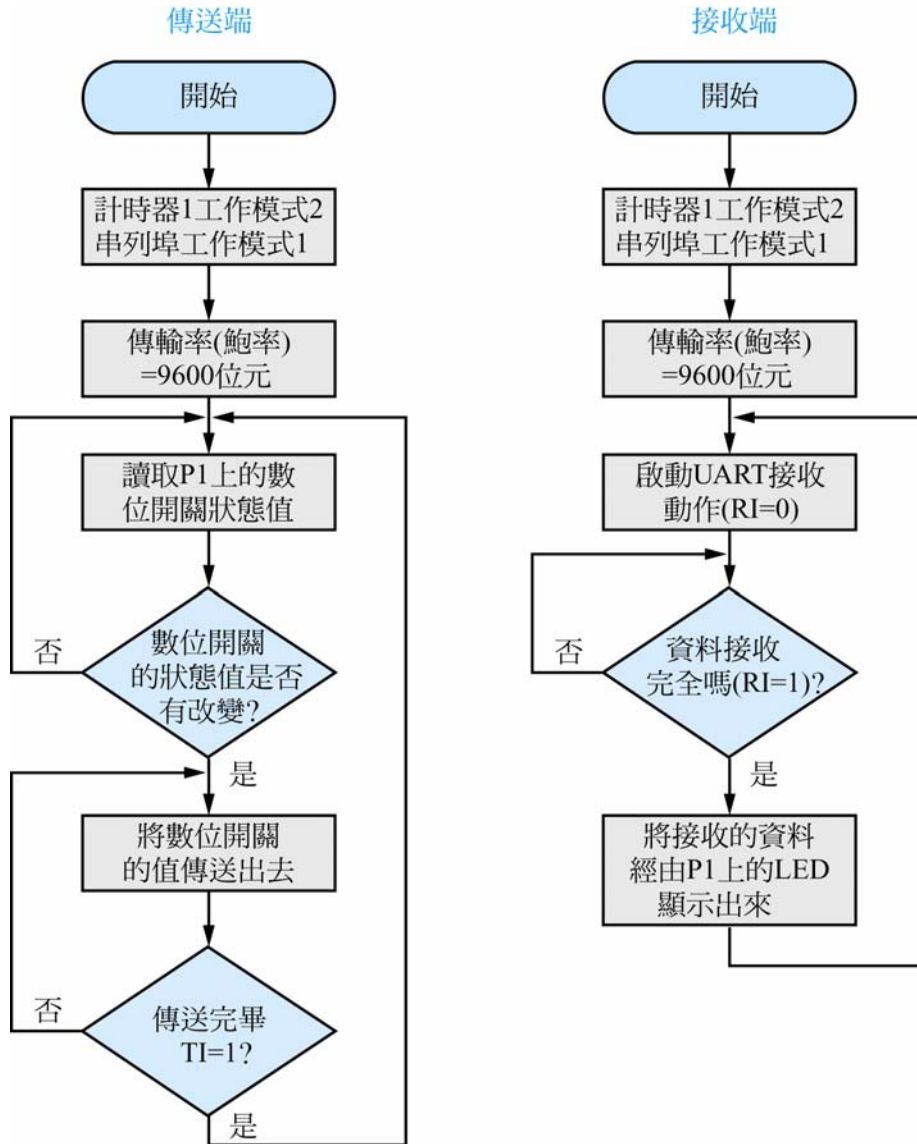
本實驗需要兩個 AT89C51，一個稱為傳送端，另一個稱接收端。傳送端的工作是檢查在 P1 上的數位開關，當檢查數位開關的開關狀態有變化時，就將變化後的數位開關狀態讀入，且透過 TxD 腳將這開關之資料狀態傳給接收端，當接收端接收到這個位元組的資料後，就將它輸出到 P0 上的 LED 顯示出來。

3. 電路圖



● 圖 7-18 兩個 MCS-51 間資料的單向傳輸實驗

4. 流程圖



● 圖 7-19 程式 P07-3T/3R 流程圖

5. 程式及程式說明

；接收端程式

；程式檔名：P07-3R.asm

```

org    0
jmp    Start
    
```

Start:

```

mov    sp,#6fh
mov    tmod,#00100000b    ; 設定 Timer 1 工作在 Mode 2 自動載入
                                ; 模式，當作 UART 的鮑率控制
anl    pcon,#01111111b    ; 設定 SMOD=0，詳細說明請參考鮑率
                                ; 計算說明部分
mov    th1,#253           ; 傳送速率（鮑率）=9600 bit/sec(@fosc
                                ; =11.059MHz)
setb   tr1                ; 啟動 Timer 1
mov    scon,#01010000b    ; 設定 UART 工作在 Mode 1 的接收模
                                ; 式（REN=1，啓用接收）
clr    ri                  ; 清除接收完畢旗標

```

Receive_Wait:

```

jbc    ri,Receive_Ok      ; 檢查 UART 是否將 8 bit 的資料讀入
                                ; (接收) 完畢，若還沒讀入完成則繼續
                                ; (RI=0)，讀入的資料被存入在 SBUF
                                ; 中等到讀入完成(RI=1)，則將跳至
                                ; Receive_Ok 且清除 RI=0
jmp    Receive_Wait       ; 跳至 Receive_Wait 繼續檢查 RI 位元

```

Receive_Ok:

```

mov    a,sbuf             ; 將接收進來的資料(SBUF 暫存器的內
                                ; 容)經由 A 暫存器反相後，由 P0 上的
mov    p0,a               ; LED 顯示出來
jmp    Receive_Wait       ; 接收完畢跳至 Receive_Wait 繼續下一筆
                                ; 資料接收

```

```

.end

```

: 傳送端程式

: 程式檔名 : P07-3T.asm

```

Buffer    equ    30h                ; Buffer 為外部數位開關狀態暫存器
          org    0
          jmp    Start

Start:
          mov    sp,#6fh
          mov    tmod,#00100000b    ; 設定 Timer 1 工作在 Mode 2 自動載入
                                     ; 模式，當作 UART 的鮑率控制
          anl    pcon,#01111111b    ; 設定 SMOD=0，詳細說明請參考鮑率
                                     ; 計算說明部分
          mov    th1,#253           ; 傳送速率 (鮑率) =9600 bit/sec(@fosc
                                     ; =11.059MHz)
          setb   tr1                ; 啓動 Timer 1
          mov    scon,#01000000b    ; 設定 UART 工作在 Mode 1
          mov    Buffer,#0ffh        ; 將外部數位開關的暫存器 Buffer 內容
                                     ; 設定為 ffh

TransmitLoop:
          mov    a,p1               ; 輸入數位開關的狀態值
          cjne   a,Buffer,Transmit  ; 檢查數位開關的狀態是否有改變，若
                                     ; 有則跳至 Transmit
          jmp    TransmitLoop       ; 數位開關的狀態沒有改變，跳至
                                     ; TransmitLoop 繼續檢查

Transmit:
          mov    Buffer,a            ; 將最新的數位開關狀態，存放在
                                     ; Buffer 中
          mov    sbuf,a             ; 將數位開關的狀態值，透過 UART 的
                                     ; TxD 傳送至接收端

```

Transmit_Wait:

```
    jbc    ti,Transmit_Ok      ; 檢查 UART 是否將 Buffer 內容(8 bit)
                                ; 傳送完畢，若還沒則繼續，等到傳送
                                ; 完畢(TI=1)，則將 TI 清除為 0 且跳至
                                ; 位址 Transmit_Ok
    jmp    Transmit_Wait      ; 繼續檢查 TI 位元
```

Transmit_Ok:

```
    jmp    TransmitLoop      ; 回到 TransmitLoop 繼續下一次的傳送
    .end
```

6. 討 論

以上是兩個 CPU 傳送／接收資料的實驗，在兩個 CPU 使用 UART 互傳資料時，傳送端與接收端必須以相同速率才可能互傳資料，也就是說傳送端與接收端必須使用相同鮑率。

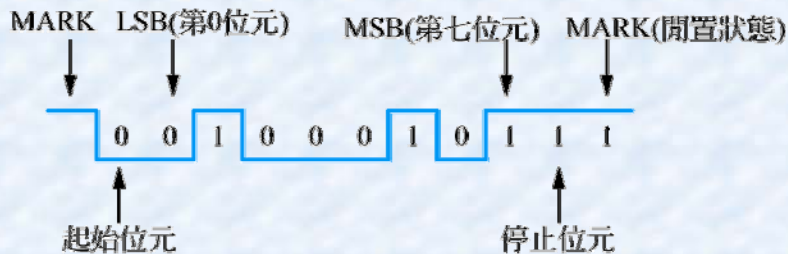
MCS-51 的 UART 工作在 Mode 1 時，鮑率是由 Timer 1 產生，資料由 TxD 傳送，由 RxD 接收，而且每次皆以 10 位元為一單位，只要將欲傳送的 8 位元資料放進 SBUF 暫存器，MCS-51 會自動加上起始位元及停止位元。UART 的 Mode 1 較適合做一對一的通訊傳輸。



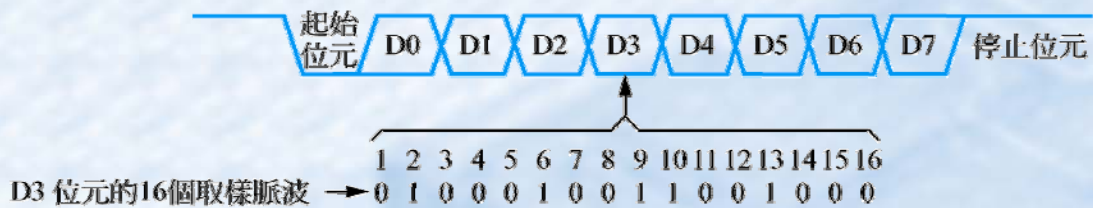
學後評量 ●●●

一、選擇題

- () 1. MCS-51 內部提供的串列傳輸介面是哪一種？
 (A)同步式 (B)非同步式 (C)以上兩者皆有 (D)以上兩者皆非。
- () 2. 如下圖的串列傳輸資料中，傳送資料為 (A)11010001b
 (B)00100010b (C)01000100b (D)10100010b。



- () 3. 下列何者不是 MCS-51 的 UART 工作模式？ (A)I/O 擴充模式
 (B)10 位元可變鮑率通信模式 (C)11 位元固定鮑率通信模式
 (D)12 位元可變鮑率通信模式。
- () 4. 下圖是 MSC-51 透過 UART 從 RxD 端接收外面資料位元 3 的取樣脈波圖，根據取樣結果位元 3 準位為 (A)0 (B)1 (C)don't care
 (D)高阻抗。

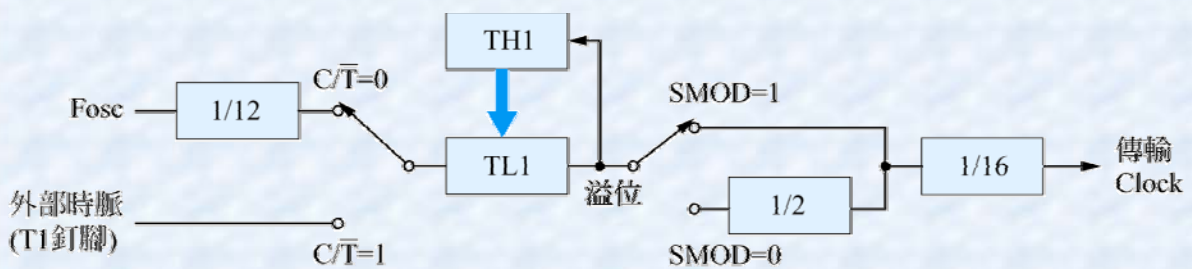


- () 5. 試計算 MCS-51 UART 的 Mode1 鮑率，假設 $f_{osc}=6\text{MHz}$ ， $\text{SMOD}=1$ ，則 $\text{rate}=\text{ (A)93.7K (B)187.5K (C)375K (D)750K bit/sec}$ 。

- () 6. 根據下圖及 UART mode1 鮑率公式，採用 fosc=11.0592MHz，SMOD=0，要得到 rate=4800 Bit/sec 時 TH1 必須載入多少值 (A)253 (B)250 (C)244 (D)232。

(1) Mode 1 或 3 的鮑率 = $\frac{2^{SMOD}}{32} \times \frac{fosc}{12 \times [256 - (TH1)]}$ fosc : 外部振盪頻率

(2) TH1 = $256 - \frac{2^{SMOD} \times fosc}{384 \times \text{鮑率}}$



- () 7. 參考上圖下列程式何者可啟用 UART MODE1 rate=9600 Bit/sec ?

- (A) `mov scon,#01000000b`
`mov tmod,#00100000b`
`anl pcon,#01111111b`
`mov tl0,#253`
`mov th1,#253`
`setb tr1`
- (B) `mov scon,#01000000b`
`mov tmod,#00100000b`
`anl pcon,#01111111b`
`mov tl0,#244`
`mov th1,#244`
`setb tr1`
- (C) `mov scon,#01000000b`
`mov tmod,#00100000b`
`anl pcon,#01111111b`


```

mov    tl0,#253
mov    th1,#253
setb   tr0

(D) mov    scon,#10000000b
mov    tmod,#00100000b
anl    pcon,#01111111b
mov    tl0,#253
mov    th1,#253
setb   tr1

```

相關暫存器名稱

暫存器	位元 7	位元 6	位元 5	位元 4	位元 3	位元 2	位元 1	位元 0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
PCON	SMOD	—	—	—	GF1	GF0	PD	IDC
IE	EA	—	ET2	ES	ET1	EX1	ET0	EX0
IP	—	—	PT2	PS	PT1	PX1	PT0	PX0

二、實作題

1. 更改圖 7-18 電路圖的程式，使傳送端數位開關設定為 1 時，接收端的 LED 單燈右移。數位開關設定為 2 時，接收端的 LED 單燈左移，數位開關設定為 3 時，LED 閃爍。



心得筆記



A series of horizontal lines for writing, with a dashed midline on each line, set against a background of a blue-tinted circuit board.