

實驗八 UART 傳輸

1. 實驗目的

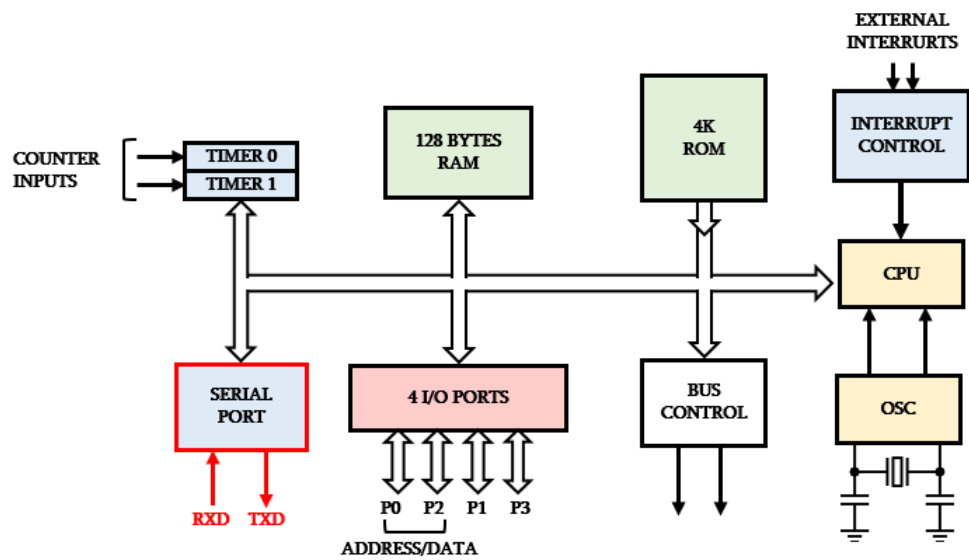


圖 8-1、Block Diagram of the 8051 Core

8051 透過 UART 並使用輪詢和中斷兩種方式接收來自 PC 的資料，8051 再自動回傳字元資料和數值資料至 PC 終端機顯示結果並比較差別。

2. 材料清單

表 8-1、材料清單

器材名稱		數量
AT89S51		1
12MHz 石英震盪器		1
LED 二極體		8
CP2102 USB to TTL		1
電阻	1kΩ	8
	10 kΩ	1
電容	20pF	2
	10μF	1

3. 元件原理

通訊介面

串列埠中斷

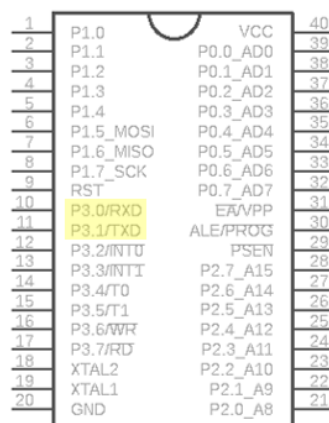


圖 8-2、TXD、RXD 腳位

CPU 透過 RXD 接腳 (Pin10，與 P3.0 GPIO 共用接腳) 來接收串列資料，及 TXD 接腳 (Pin11，與 P3.1 GPIO 共用接腳) 來傳送串列資料。8051 使用這兩個接腳來接收 (RI) 中斷需求或者傳送 (TI) 中斷需求。

RS232

8051 有提供一組全雙工的串列傳輸介面 UART，由 TXD (腳位 11) 來傳送串列資料而由 RXD (腳位 10) 來接收資料，其邏輯位準為 TTL 準位 (+2V ~ +5V, 0V ~ 0.8V)，當傳輸距離較長時，為了避免訊號的衰減造成資料判讀出錯，則可藉由經過 RS232 信號，邏輯準位為 (-15V ~ -3V, +3V ~ +12V) 來傳輸。而要轉換 TTL 到 RS232 的準位，目前通常使用準位轉換 IC HIN23 或 MAX232，只要使用轉換 IC，即可完成介面準位的轉換，在將轉換後的串列訊號連上 DB-9PIN 的 RS232 連接埠即可和 PC 連線建立 RS232 的通訊介面，故可以說 RS232 定義的是傳輸的實體層。

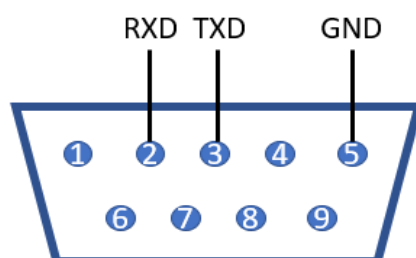


圖 8-3、RS232

因為現在的電腦許多都沒有 RS232 的连接埠，所以目前實驗用 UART 轉 USB 的 CP2102 來取代 RS232 電路。

CP2102 擁有 USB2.0 功能控制器、晶體振盪器、USB 接收發送器、EEPROM 及非同步串列資料匯流排 UART，支援全雙工訊號傳輸。CP2102 透過驅動程式模擬虛擬 COM 連接埠來達成轉換的目的。CP2102 的系統方塊圖可參考：[CP2102 datasheet p.1 Figure 1. Example System Diagram](#)。



圖 8-4、CP2102 USB to UART

UART

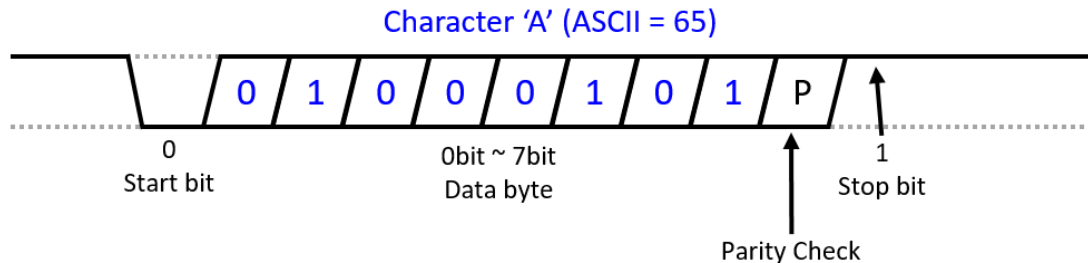


圖 8-5、UART Packet

UART (Universal Asynchronous Receiver Transmitter)，萬用非同步串列埠在傳送資料時，不需要額外使用傳輸線來同步訊號，但必須在資料前後加上同步訊號，並將同步訊息和資料一起由同一傳輸線送出。序列通訊在使用前須做些許設定，最常見的標準設定為：

- 速率 (Baud)：鮑 (Baud) 是指單位時間內傳輸多少位元 (bit per second)，也稱位元傳輸率。
- Start bit：1 個位元，通常為低電位。
- Data byte：資料位元，為 8 個 bit 所組成，是傳送的資料存放的位置。
- Stop Bit：1、1.5 或 2 位元，由使用傳送與接收兩方決定，但兩者的選擇必須一致，通常為高電位。

- 同位檢查 (Parity Check)：可驗證資料的正確性，可選擇奇同位或偶同位，通常沒在使用。

奇同位：傳送的所有 bits (包含字元和檢查位元) 中「1」的個數為奇數。

偶同位：傳送的所有 bits (包含字元和檢查位元) 中「1」的個數為偶數。

同位檢查可以看出傳送中是否發生錯誤，若某一個位元組中同位檢查位元發生錯誤，則位元組在傳輸中有發生錯誤。若同位檢查正確，則可能無發生錯誤或是發生偶數個錯誤。

電源控制暫存器 (PCON 暫存器)

PCON 暫存器中只有 SMOD 位元與串列埠傳輸速度有關，其他位元則是用於省電模式的設定。在 SMOD 的設定上要注意，PCON 暫存器不可位元定址，故可使用 `ORL PCON #80H` 來設定 SMOD 為 1，用 `ANL PCON #7FH` 清除 SMOD 為 0。

表 8-2、PCON 暫存器

	7	6	5	4	3	2	1	0
PCON	SMOD	-	-	-	GF1	GF0	PD	IDL

串列埠控制暫存器 (SCON 暫存器)

SCON 暫存器為一個 8 位元的可位元定址暫存器，其中各位元如下表格所示：

表 8-3、SCON 暫存器

	7	6	5	4	3	2	1	0
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

- SM0 和 SM1

這兩個位元用來設定串列埠的模式，如下表所示：

表 8-4、設定 SM0、SM1 之串列埠模式

SM0	SM1	MODE	功能	鮑 (Baud)
0	0	0	位移暫存器	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

在模式 0 時，SM2 必須清除為零。

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

在模式 2 或模式 3 時，若 SM2=1，則必須接收到的第 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訊功能上。

- REN

若為 0，則 RXD 接腳不接收資料。

若為 1，則 RXD 接腳可接收資料。

- TB8

在模式 2 和模式 3 時，此位元為第九個傳送位元，即為同位位元。

- RB8

在模式 0 時，此位元無作用。

在模式 1 時，若 SM2=0，此位元為停止位元。

在模式 2 和模式 3 時，此位元為第九個接收位元。

- TI

傳送中斷旗標，當中斷結束時 TI 位元不會自動清為 0，需自行清除。

在模式 1、模式 2 和模式 3 時，若停止位元完成傳送，此位元會設為 1 並提出 TI 中斷。

模式 0 時，當第 8 個位元傳送完成，TI 被設為 1 並提出 RI 中斷。

注意：當提出中斷時並不代表進入中斷，得確認 IE 暫存器是否有開啟 UART 中斷。

- RI

接收中斷旗標，當中斷結束時 RI 位元不會自動清為 0，需自行清除。

在模式 1、模式 2 和模式 3 時，若接收到正確停止位元後，RI 被設為 1 並提出 RI 中斷。模式 0 時，當第 8 個位元接收完成，RI 被設為 1 並提出 RI 中斷。

串列埠的 4 種模式與鮑率設定

Mode 0

Mode 0 下的鮑率，固定為 OSC/12，不用設定。故此模式用固定的鮑率

(OSC/12) 來傳輸資料，不管是接收資料還是傳出資料，處理器的 RxD 接腳都連接串列資料線，TxD 接腳連接位移脈波線。在資料傳送時依據 TxD 傳出的位移脈波，並使用 RxD 負責送出串列資料。而接收資料時，也由 TxD 傳出位移脈波，RxD 接腳負責接收串列資料，因用到同步脈波並只有用一條線來傳輸或接收資料，故為一種半雙工模式。

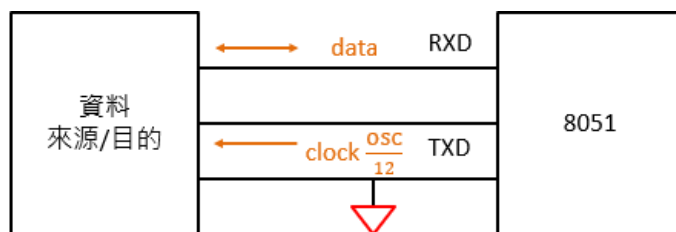


圖 8-6、Serial Port Mode 0

Mode 1

在 Mode 1 或 Mode 3 下，鮑率可由 Timer1 的溢位脈波所影響，Timer 1 通常使用有自動載入功能的 Mode2 模式。

鮑率的計算為：

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

例如：若 Baud rate 想設定為 1200 而 OSC 為 12MH，

$$1200 = \frac{2^0}{32} \times \frac{12000000}{12 \times (256 - \text{TH1})}$$

256-TH1=26.04，故取整數 TH1 大概約為 230₍₁₀₎ = 0xe6。

這邊發現最後算出的 TH1 並非整除，這個狀況其實會造成鮑率誤差，如果太嚴重的話還會造成 UART 讀取傳輸資料的錯誤。由於 11.0592M 為 baud rate 9600、19200 的倍數，故使用 11.0592MH 的石英震盪晶體，以獲得整除無鮑率誤差的 TH1，且 11.0592MHz 最接近 8051 最高支持的 12MH 石英震盪器，故使用 11.0592MH 的石英震盪晶體可以在 8051 損失最少效能的情況下保持最高的傳輸速和準確性。

Mode1 用可變的 Baud 來傳輸資料，而接腳的部分，8051 的 RxD 接通訊對象的 TxD 接腳，而 8051 的 TxD 接腳則接通訊對象的 RxD 接腳。在 Mode1 下，一筆資料會由 10 個 bit 組成，一開始為 start bit (低準位) 接著為 8 個位元的資料，最後為 stop bit (高準位) 所組成。

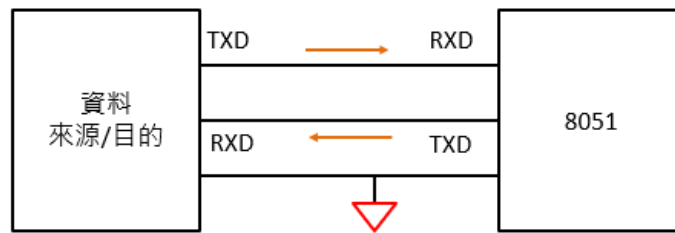


圖 8-7、Serial Port Mode 1

Mode 2

Mode 2 下，鮑率可以是 $OSC/32$ 或 $OSC/64$ ，若將 SMOD (PCON 的第 7 個 bit) 設為 0，採用 $OSC/64$ ；將 SMOD 設為 1，則採用 $OSC/32$ 。

此模式下接線方式和 Mode1 一樣，但傳輸的資料變為 11bit 除了一樣的 start bit 和 stop bit 外，在 8 位元資料後面多了一個同位位元。而在傳送資料時，同位位元的來源為 TB8，TB8 的值可由 PSW 暫存器中的 P 位元取得，達成同位位元的目的。接收時，直接將第 9 位元移入 RB8。

Mode 3

鮑率的設定和 Mode1 的設定相同，除了用可變的 Baud 值來進行串列埠的傳輸外，傳輸資料的部分和 Mode 2 幾乎相同，傳輸的資料變為 11bit。

串列資料緩衝器 SBUF (Serial Data Buffer)

SBUF 是由兩個不同的 8bit 暫存器所組成：傳送緩衝和接收緩衝暫存器。當把資料傳送到 SBUF 時，資料會被放到傳送緩衝暫存器等待傳送，事實上對使用者來說，當把資料丟到 SBUF 上時，傳送就開始了。當從 SBUF 讀取資料時，會讀取來自接收暫存器的資料。由於傳送與接收實際上是操作兩個不同的暫存器，所以 8051 可以允許傳送與接收同時進行，雖然這是兩個不同的暫存器，但在 8051 中他們使用的是一樣的位址來表示 (SBUF 在 SFR 的位址=99H)。

4. 實驗內容

利用終端機軟體輸入資料給 8051 並透過 UART 傳送給 8051，8051 接過判斷是否為正確字串後，傳送判斷結果給 PUTTY 終端機呈現結果，資料需嘗試以字元和數值方式傳送。

PuTTY 終端機使用說明

請至 PuTTY 官方網頁下載：<https://www.putty.org/>，並依照以下步驟設定。

- 下載後開啟 putty.exe

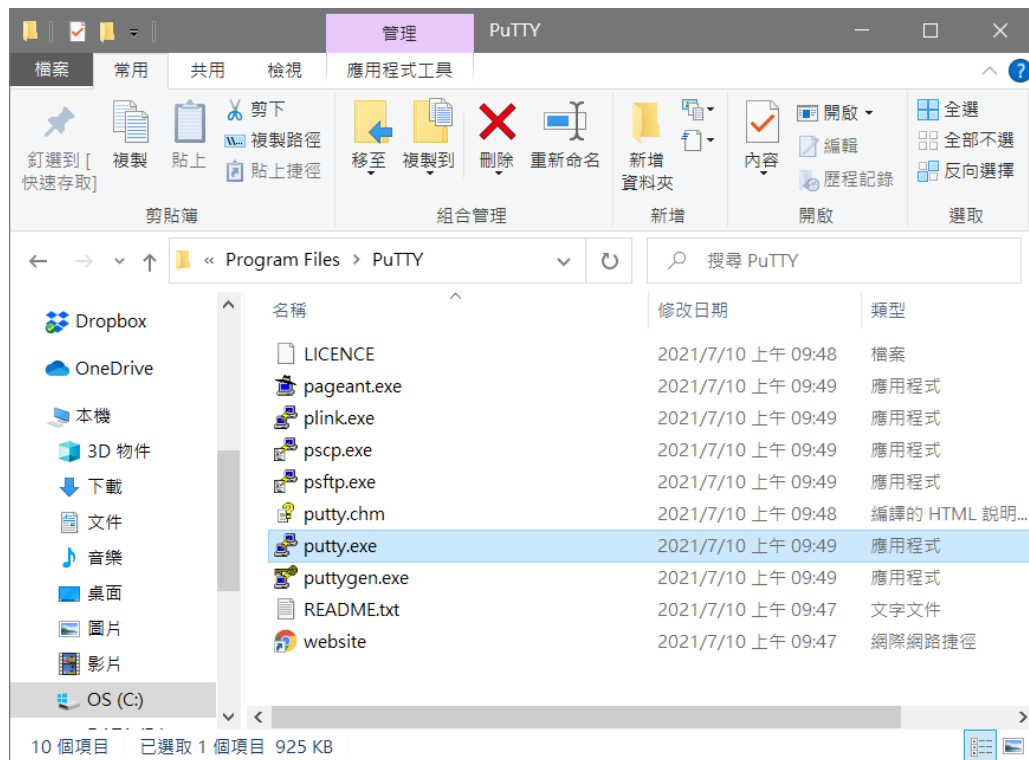


圖 8-8、putty.exe

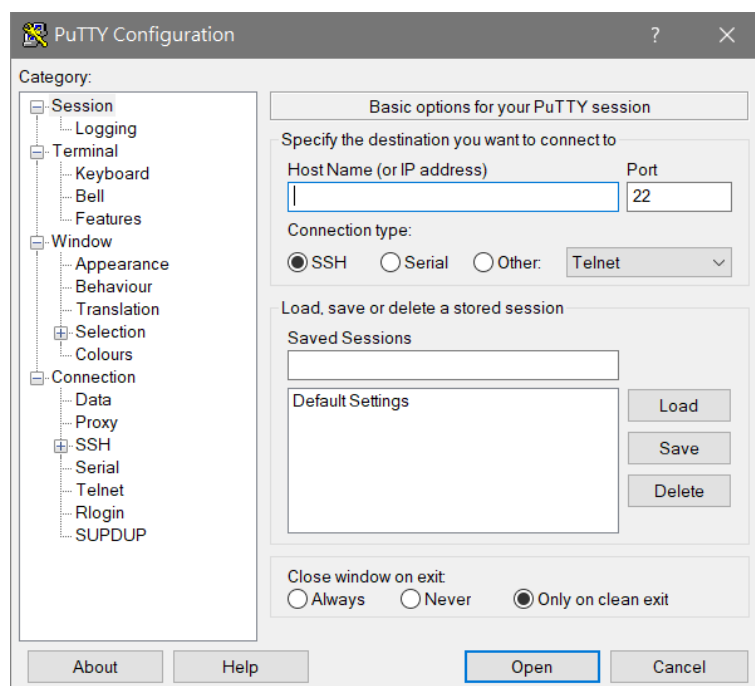


圖 8-9、PuTTY Configuration

- [Session] → 選擇[Serial]

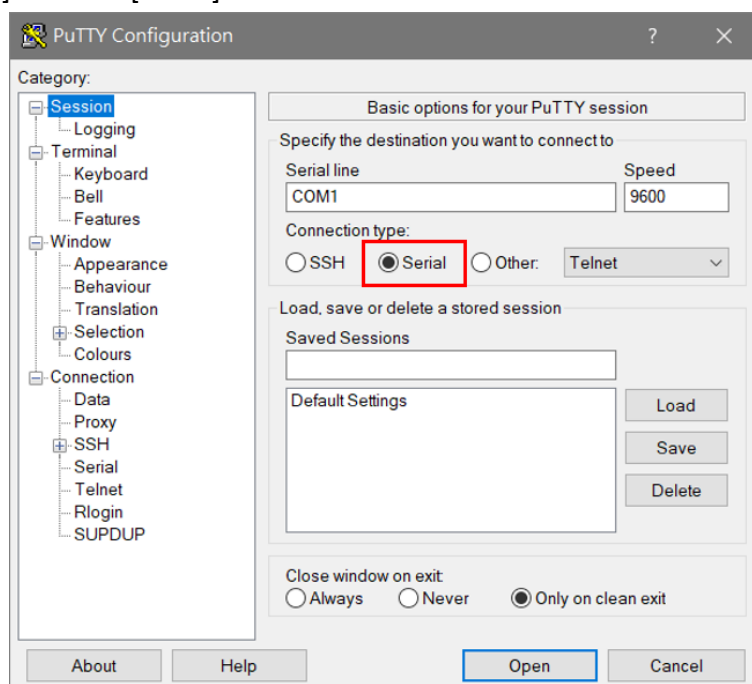


圖 8-10、PuTTY 設定

- 修改 COM port 以及鮑率 (如何確認 COM port：先將圖 8-4 的 CP2102 USB to UART 模組插入電腦的 USB 孔，再以右鍵點擊[開始] → [裝置管理員] → [連接埠])。

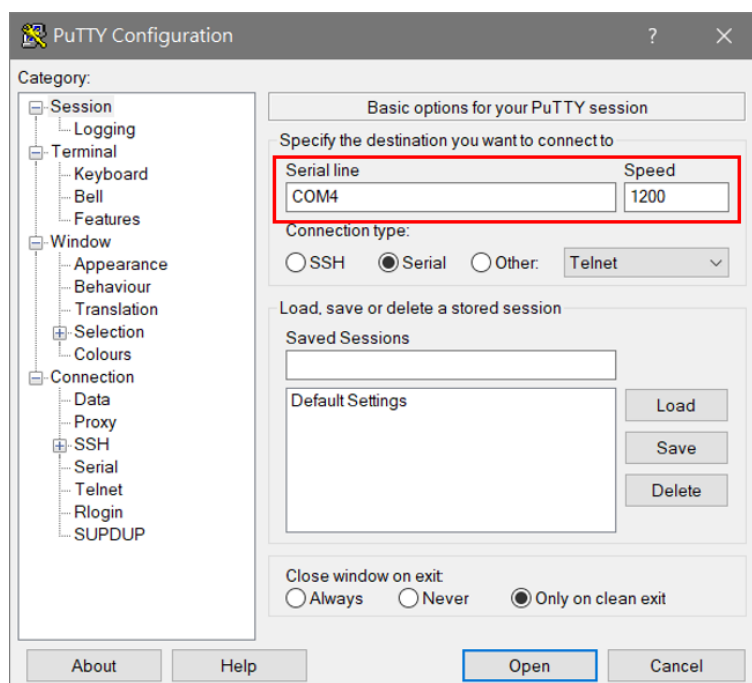


圖 8-11、PuTTY 設定

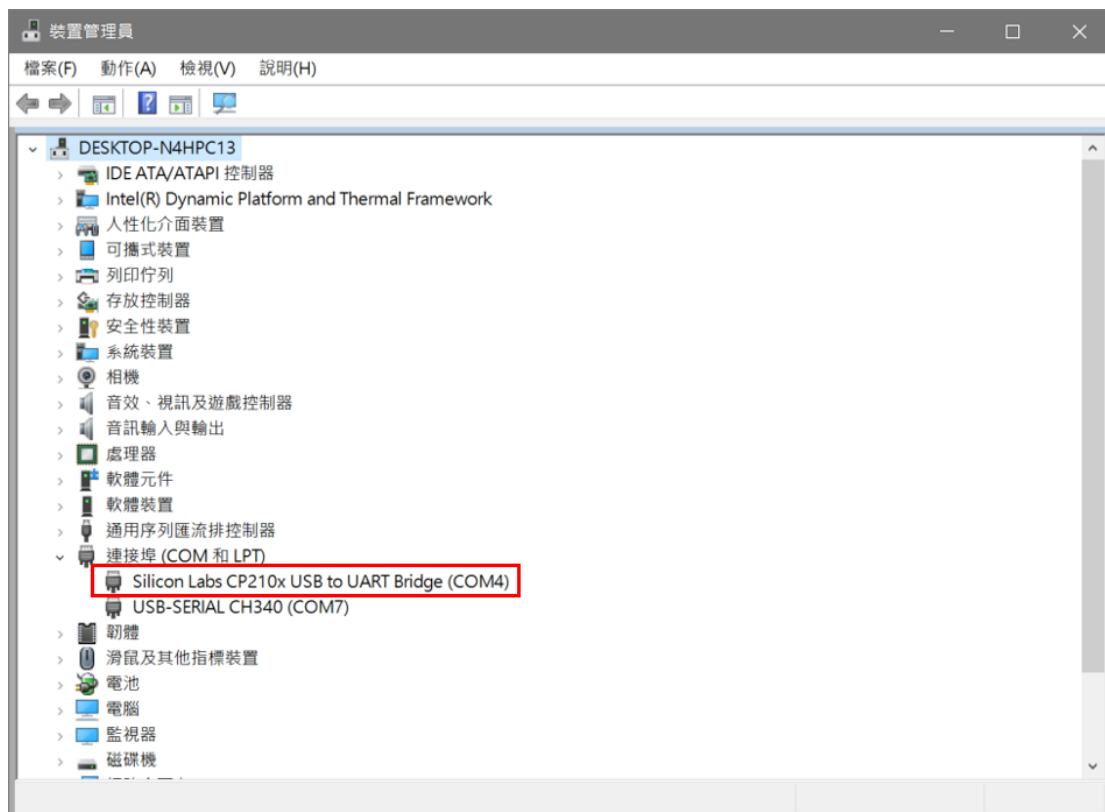


圖 8-12、裝置管理員 – COM Port

- [Terminal] → [Local echo] → 選擇[Force on]，並點選[Open]開啟 PuTTY 終端機。

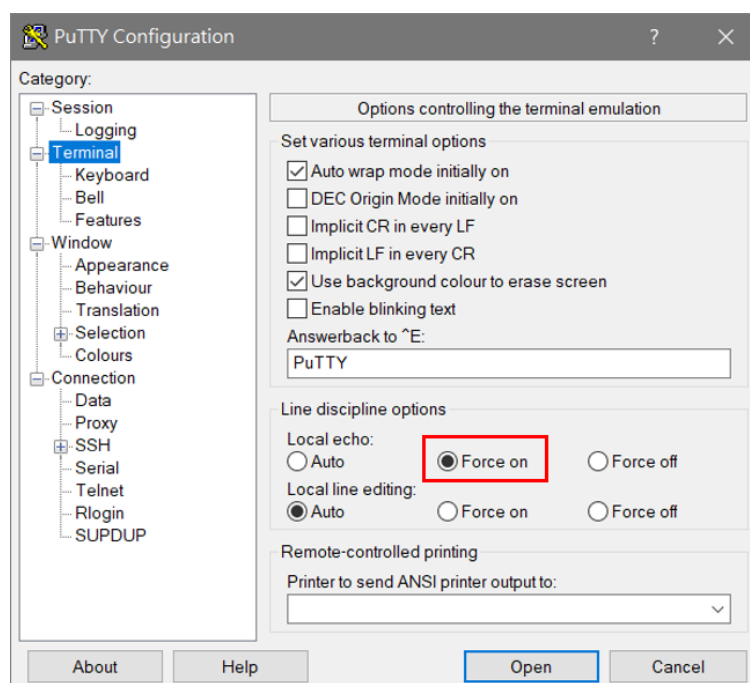


圖 8-13、PuTTY 設定

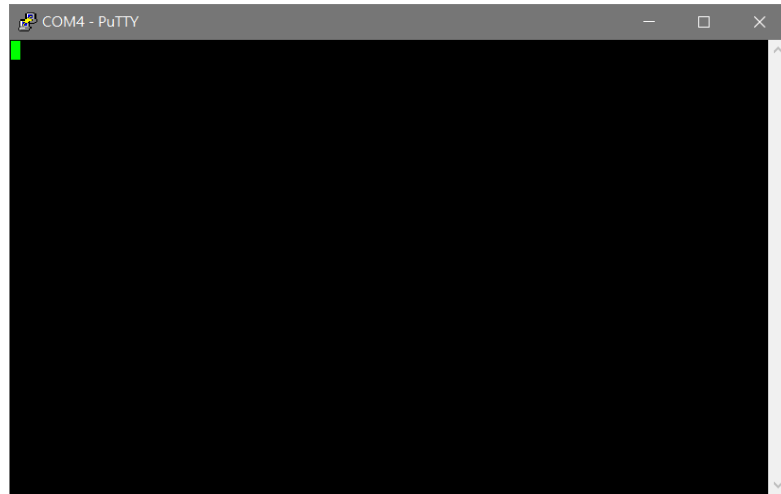


圖 8-14、PuTTY 終端機

- 在終端機上方按右鍵，點選[Reset Terminal]則可以清除內容。

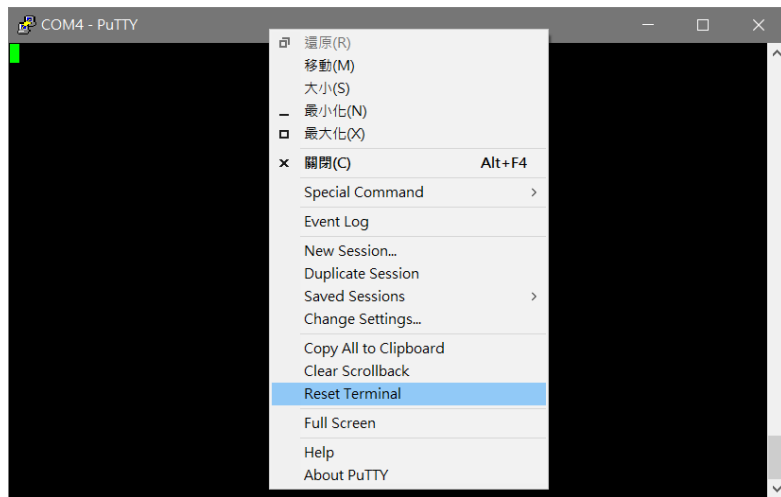


圖 8-15、PuTTY 終端機 – Reset Terminal

5. 實驗電路圖

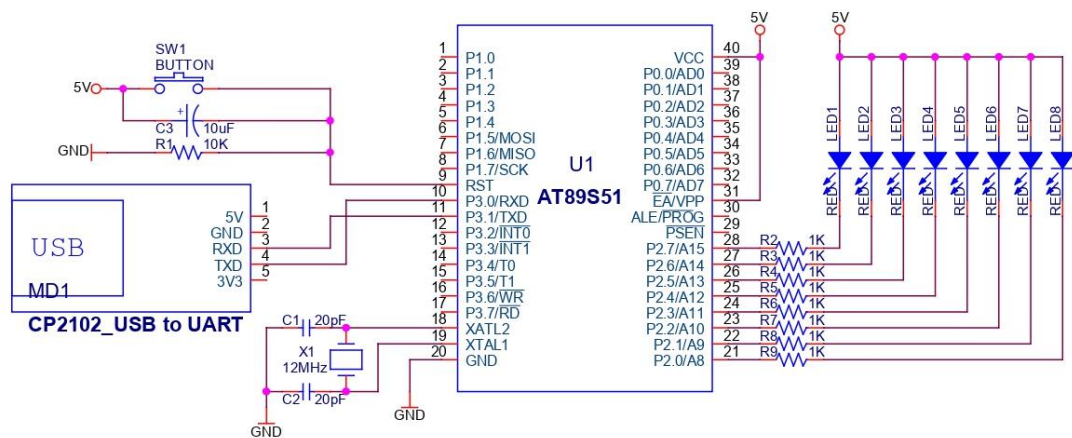


圖 8-16、實驗八基礎題參考電路圖

6. 軟體流程圖

中斷法

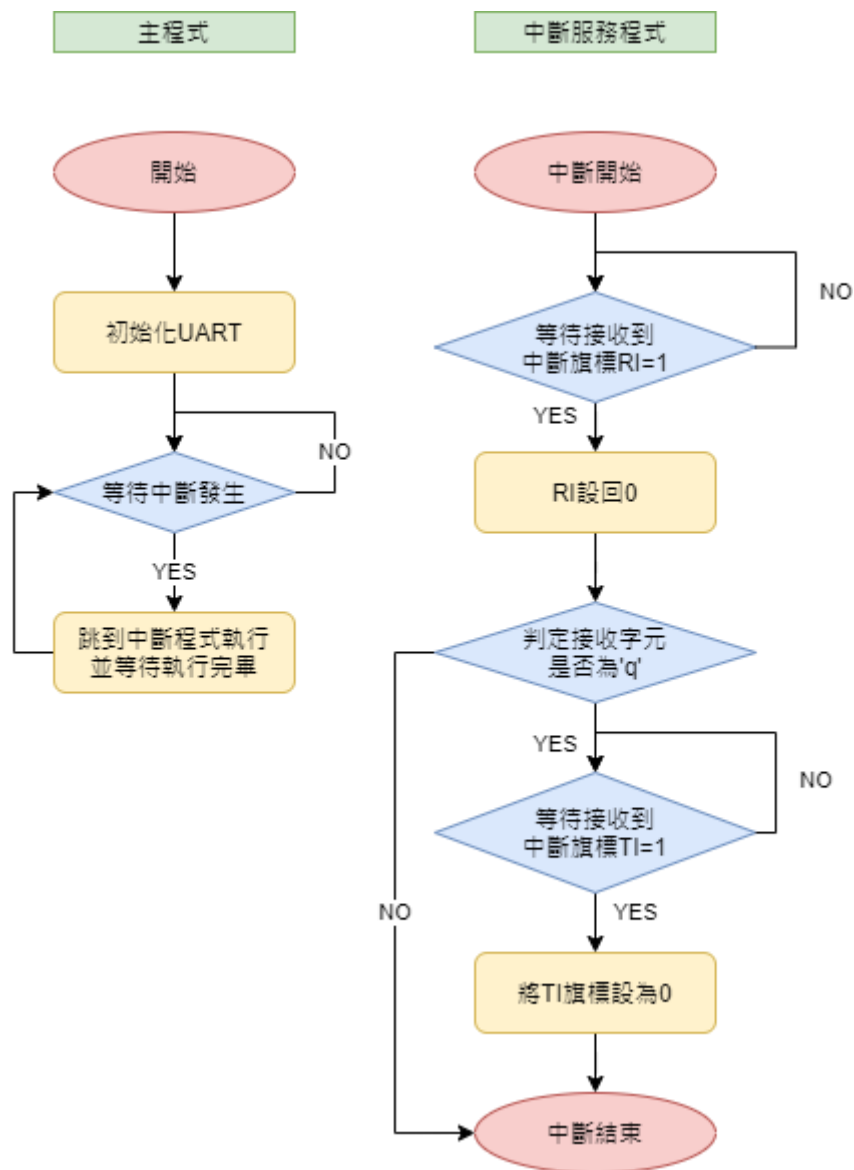


圖 8-17、UART 傳輸中斷法

輪詢法

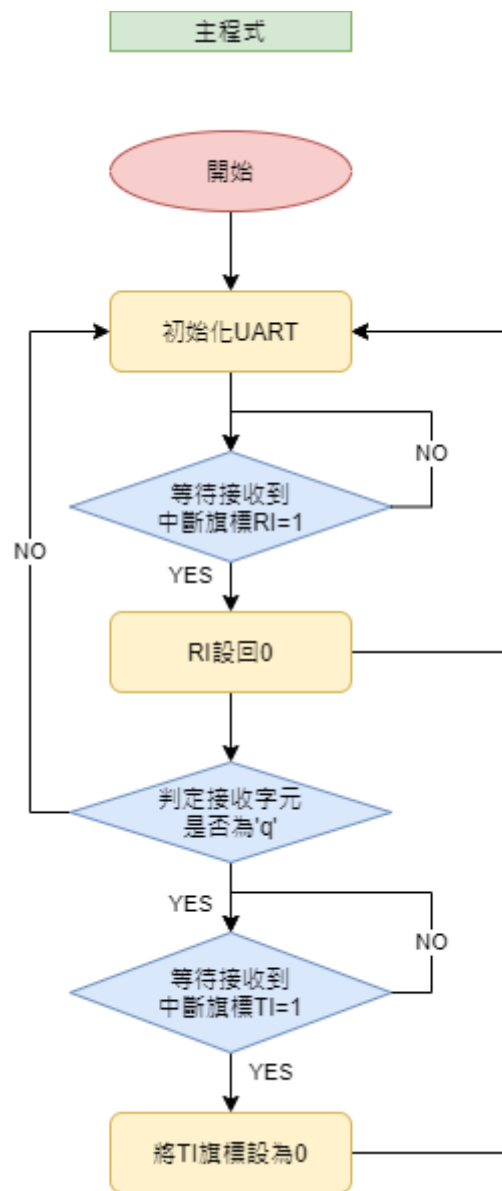


圖 8-18、UART 傳輸輪詢法

7. 範例程式碼一

中斷法

```
1  #include <regx51.h>
2
3  void init_uart( );           //declare UART function
4  char recivevalue;           //receive data
5
6  void main( )
7  {
8      init_uart( );
9      while (1);
10 }
11
12 void init_uart( )
13 {
14     SCON = 0x50;              //Mode1
15     TMOD = 0x20;              //Timer1 Mode2
16     TH1 = 0xe6;               //set Baud=1200
17     TR1 = 1;                  //enable TCON Timer1
18     IE = 0x90;                //enable UART
19 }
20
21 void UART_Isr(void) interrupt 4 //UART
22 {
23     while (RI == 0);          //wait recieve flag == 1
24     RI = 0;                    //clear flag
25     recivevalue = SBUF;
26     P2 = recivevalue;          //LED display data
27
28     if (recivevalue == 'q')
29     {
30         SBUF = recivevalue;    //load data
31         while (TI == 0);       //wait transmit flag == 1
32         TI=0;                  //clear flag
33     }
34 }
```

輪詢法

```
1  #include <regx51.h>
2
3  void init_uart( );           //declare UART function
4  char recivevalue;           //receive data
5
6  void main( )
7  {
8      init_uart( );
9      while (1)
10     {
11         while (RI == 0);      //wait recieve flag == 1
12         RI = 0;               //clear flag
13         recivevalue = SBUF;
14         P2 = recivevalue;      //LED display data
15
16         if (recivevalue == 'q')
17         {
18             SBUF = recivevalue; //load data
19             while (TI == 0);    //wait transmit flag
20             TI=0;               //clear flag
21         }
22     }
23 }
24
25 void init_uart( )
26 {
27     SCON=0x50;                //Mode1
28     TMOD=0x20;                //Timer1 Mode2
29     TH1=0xe6;                 //set BAUD=1200
30     TR1=1;                    //enable TCON Timer1
31 }
```

8. 整理的題目，選擇/是非題