

微介實驗八

UART傳輸

日期：11月26日

報告者：蕭力文

Outline

- 學習重點
- 實驗內容
- 材料清單
- 元件原理
- 實驗電路圖
- 軟體流程圖
- 範例程式碼
- 進階題

學習重點

- 了解UART傳輸的原理
- 了解鮑率的計算方式和石英震盪器頻率選擇的原因
- 比較Polling和Interrupt傳輸方式的差別

實驗內容

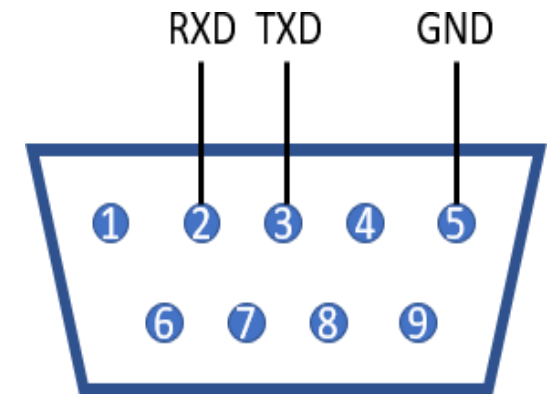
- 利用終端機軟體輸入資料給8051並透過UART傳送給8051，8051接過判斷是否為正確字串後，傳送判斷結果給PUTTY終端機 呈現結果
- 資料需以字元和數值方式傳送

材料清單

器材名稱		數量
AT89S51		1
12MHz 石英震盪器		1
按壓開關		1
LED		8
CP2102 USB TO TTL		1
電阻	1k Ω	10
電容	20pF	2
	10 μ F	1

RS232

- 8051提供一組全雙工的串列傳輸介面UART，因為其邏輯準位為TTL準位[+2V~+5V]U[0V~0.8V]，當傳輸距離較長時，訊號的衰減可能造成資料判讀出錯，故使用RS232訊號，邏輯準位為[-25V, -3V]U[+3V, +25V] 來傳輸
- (-3V, +3V): Undefined
- 使用準位轉換IC HIN23或MAX232，可轉換介面準位。



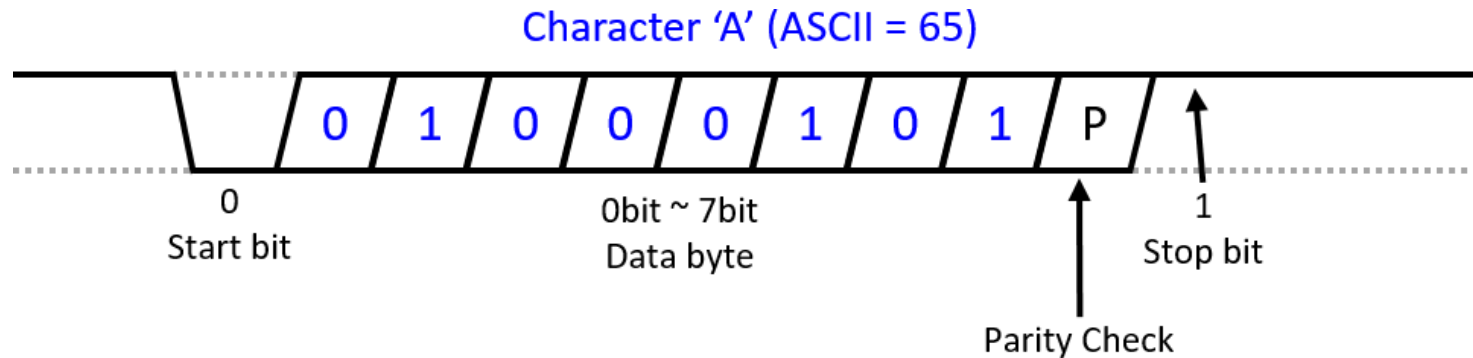
元件原理— 通訊介面

- 串列埠中斷: CPU透過RXD接腳(Pin 10, 與P3.0 GPIO共用接腳)、TXD接腳(Pin 11, 與P3.1 GPIO共用接腳), 來接收(RI)中斷需求或者傳送(TI)中斷需求。
- RXD: 接收串列資料
- TXD: 傳送串列資料
- 要傳(TXD)的byte要先方在SBUF register.
- 只要a byte寫進SBUF, 硬體自動加上framing (start bit, stop bit, parity bit等)傳出去
- 收到(RXD)的byte也是放在SBUF register.
- 收到frame, 自動去掉frame bits, 還原成1 byte.

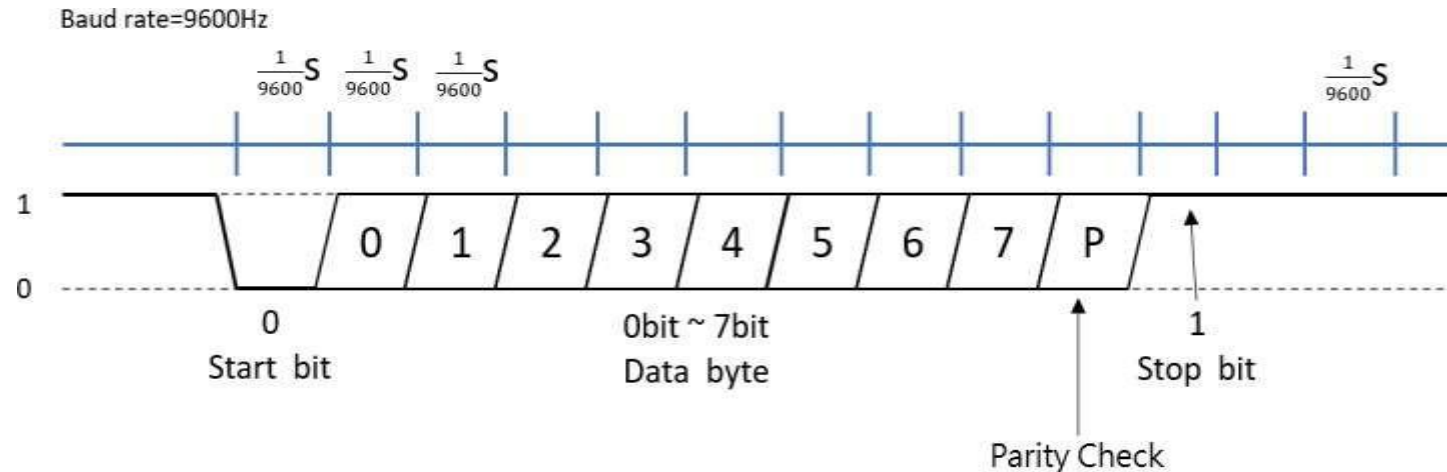
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD

UART (Universal Asynchronous Receiver Transmitter)

- 萬用非同步串列埠
- 不需要額外使用傳輸線來同步訊號
- 須在資料前後加上同步訊號，並一起送出



UART (Universal Asynchronous Receiver Transmitter)



- 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

元件原理—— SBUF (Serial Data Buffer)

- SBUF實際上是兩個不同8bit的暫存器。
 - 發送緩衝區暫存器
 - 接收緩衝區暫存器
- 發送資料而將資料移到SBUF時，此筆資料會存入發送暫存器，在移入後進行串列傳輸。
- 執行將資料從SBUF移出的指令時，這筆資料則是從接收暫存器取出。
- 雖然是兩個不同的暫存器，在8051中他們使用的是一樣的位址來表示(99H)，因此8051的串列埠可以同時進行資料的傳送與接收。

元件原理—串列埠控制暫存器

在SFR中與UART傳輸相關的暫存器：

- SCON暫存器(Serial Port Control register)
- PCON暫存器(Power Control register)

PCON暫存器

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

- 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。
- 用 **ORL** PCON #80H 來設定SMOD為1。
- 用 **ANL** PCON #7FH 來清除SMOD為0。

元件原理—串列埠控制暫存器(SCON暫存器)

SCON暫存器

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

- SCON暫存器為一個8位元的可位元定址暫存器
- 記憶體位址：98H
- 初始值：00000000

SM0和SM1

- 這兩個位元用來設定串列埠的模式，如下表所示：

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

串列埠控制暫存器(SCON暫存器)

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

TI

- 傳送中斷旗標，當中斷結束時TI位元不會自動清為0，需自行軟體清除。
- Mode 1 & Mode 2 & Mode 3：當停止位元完成傳送，TI被設為1並提出TI中斷。
- Mode 0：當第8個位元傳送完成，TI被設為1並提出TI中斷。
- 注意：當提出中斷時並不代表進入中斷，得確認是否有開啟UART中斷(IE 暫存器)。

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。
- Mode 1 & Mode 2 & Mode 3：若接收到正確停止位元後，RI被設為1並提出RI中斷。
- Mode 0：當第8個位元接收完成，RI被設為1並提出RI中斷。

串列埠控制暫存器(SCON暫存器)

- 傳stop bit時, $TI = 1$. 表示最後1-bit已送完,可以傳下一byte.
- 如果在 $TI=1$ 之前,就將新data寫入SBUF,則尚未傳完的data會lost.
- 當SBUF放入新data,一定要將 $TI=0$. (CLI TI指令)
- 收到stop bit時, $RI = 1$. 表示最後1-bit已收到, data 在 SBUF中.
- 要趕快將SBUF中data移出,以免被後面傳來data 蓋掉(overrun).
- 當SBUF中data移出後,一定要將 $RI=0$. (CLI RI指令)
- 如果 $RI=0$ 時,copy SBUF中的data,則取得的data可能不是我們要的

串列埠控制暫存器(SCON暫存器)

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

SM2

Mode 2 & Mode 3：為致能多重處理器通信功能。

Make it 0, since we are not using multiprocessors

Mode1：

- 若SM2等於1，當收到的停止位元不正確時，即為錯誤資料不接收，RI 不被設定為1。
- SM2等於0時，當收到資料時，即將RI設定為1。

Mode 0：

- 此位元必須設為0。enables multiprocessor

串列埠控制暫存器(SCON暫存器)

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。

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

TB8

- 在模式2和模式3時，此位元為第九個傳送位元，通常為同位位元。
- 多重處理器通信時，TB8=1時代表傳送的為位址訊息。

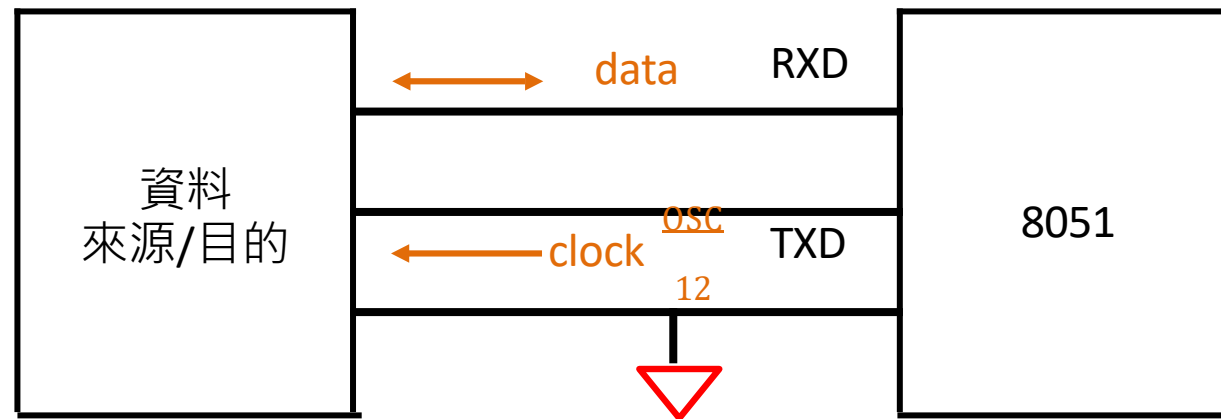
RB8

- Mode 2 & Mode 3：此位元為第九個接收位元。
- Mode 1：此位元儲存停止位元。
- Mode 0：此位元無作用。

元件原理—串列埠的4種模式

Mode 0 :

- Mode 0下的鮑率，固定為OSC/12，不用設定。
- 通常是用於I/O的擴充，而非用於串列通訊。
- 串列資料的傳送與接收都是利用RXD 接腳進行，而TXD接腳則做為輸出移位脈波。
- 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。



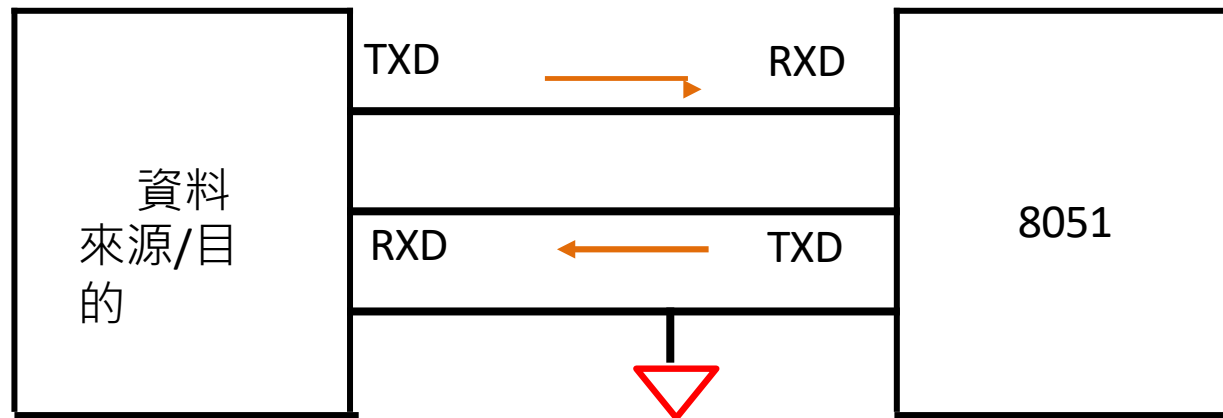
串列埠的4種模式

Mode 1：

- 在Mode 1或Mode 3下，可由Timer1的溢位脈波改變鮑率。
- 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，一筆資料基本會由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。

Mode 3：

- 串列埠設定為Mode 3時，其動作與Mode 2相似，其唯一的差別在於Mode 3的傳輸速度之鮑率設定與Mode 1相同，由Timer1設定可變的鮑率。



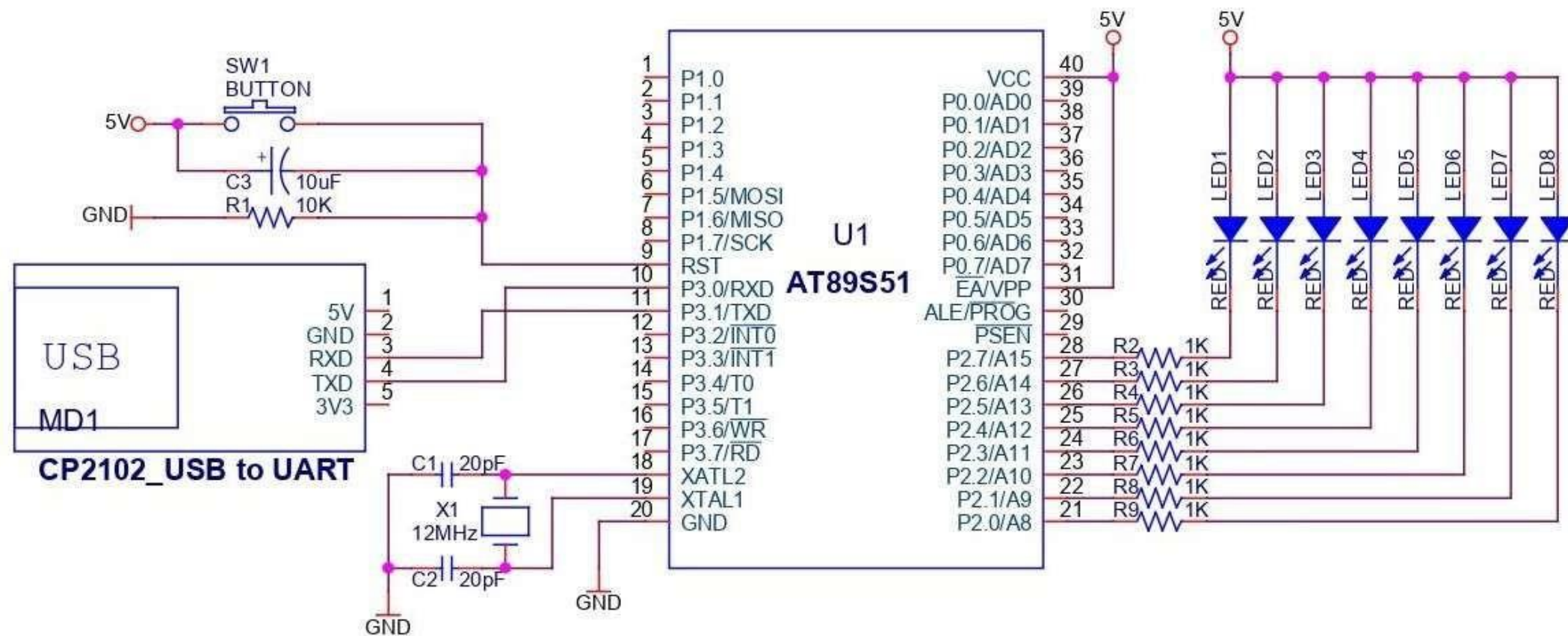
串列埠的4種模式

Mode 2 :

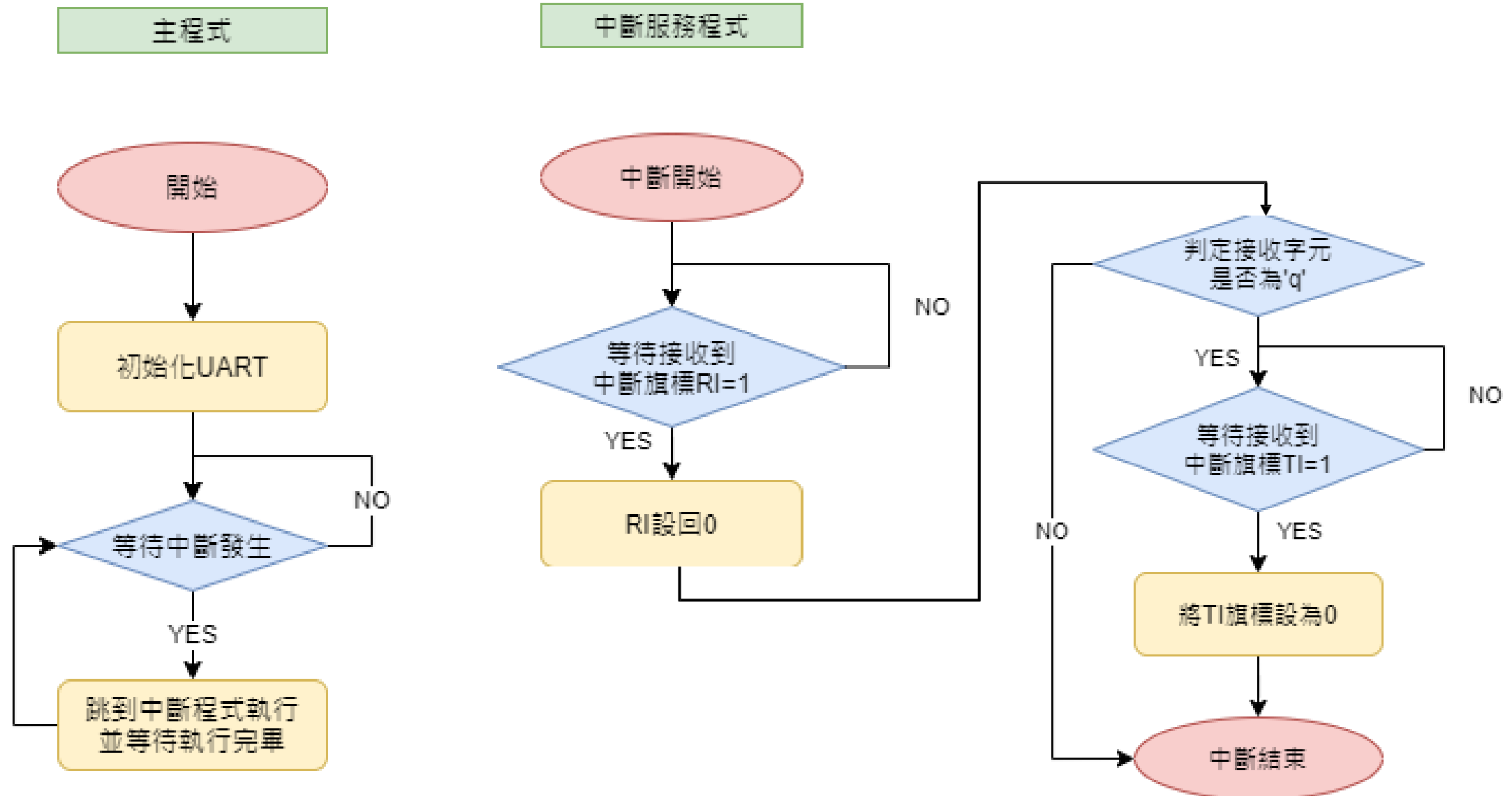
- 在Mode 2下，鮑率可以是OSC/32或OSC/64。
- 將PCON的第7個bit(SMOD)設為0，採用OSC/64。
- 將SMOD設為1，則採用OSC/32。
- 此模式下接線方式和Mode1一樣，但傳輸的資料變為11bit，除了一樣的start bit和stop bit外，在8位元資料後面多了一個同位位元。而在傳送資料時，同位位元的來源為TB8，TB8的值可由PSW暫存器中的P位元取得，達成同位位元的目的。接收時，直接將第9位元移入RB8。

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

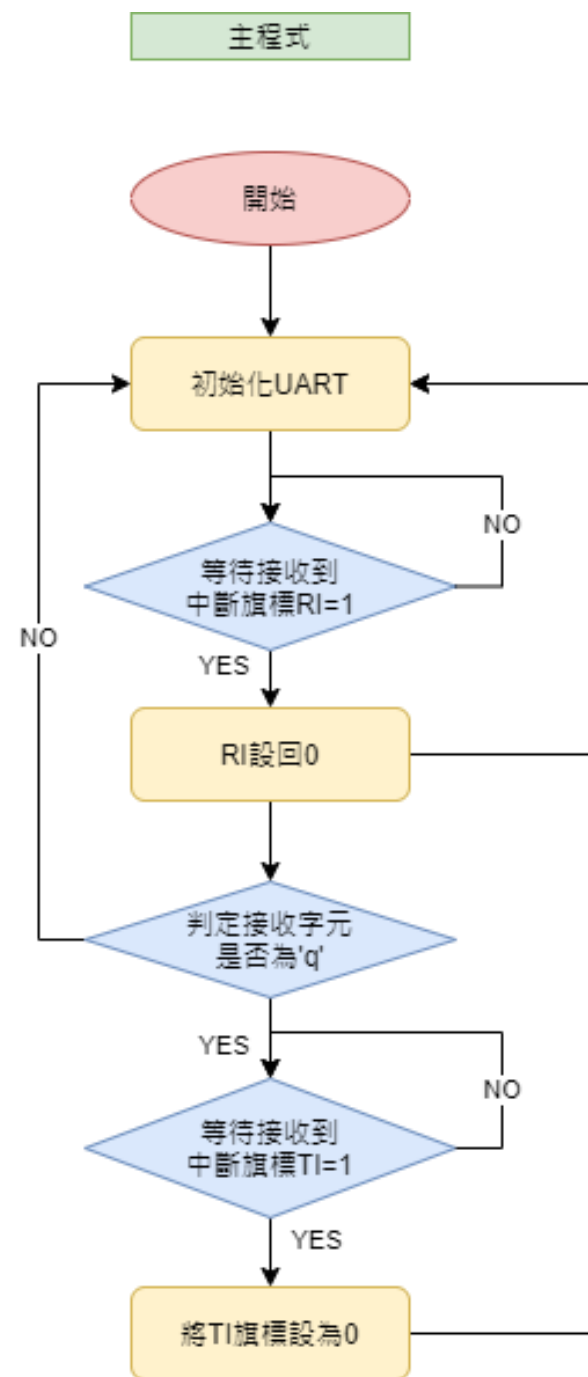
實驗電路圖



軟體流程圖 — 中斷



軟體流程圖 — 輪詢法



範例程式碼 — 中斷

```
1. #include <regx51.h>
2. void init_uart();           //宣告UART初始化函式
3. char recivevalue;           //接收輸入的資料
4.
5. void main()
6. {
7.     init_uart();             //開啟UART
8.     while(1);
9. }
```

範例程式碼 — 中斷

```
10. void init_uart(void)
11. {
12.     SCON=0x50;           //設定SCON暫存器傳輸模式為mode1
13.     TMOD=0x20;           //開啟TIMER1的Mode2
14.     TH1=0xe6;            //baud的設定 1200
15.     TR1=1;               //TCON中Timer1的啟動開關 TR1為1
16.     IE=0x90;             //開起UART中斷設定 EA跟ES位元為1
17. }
```


範例程式碼 — 中斷

```
18. void init_uart(void) interrupt 4 //UART中斷
19. {
20.     while(RI == 0);           //若為接收中斷
21.     RI=0;
22.     recivevalue=SBUF;         //當中斷將SBUF的值放入recivevalue中
23.     P2 = recivevalue;         //將輸入的值用LED顯示觀察
24.     if(recivevalue == 'q')    //當接收字元為q
25.     {
26.         SBUF= recivevalue;    //回傳接收到的值
27.         while(TI==0);        //若為傳送中斷
28.         TI=0;                //將TI旗標設回0
29.     }
```

範例程式碼 — 輪詢

```
1. #include <regx51.h>
2. void init_uart();           //宣告UART初始化函式
3. char recivevalue;           //接收輸入的資料
4.
5. void main()
6. {
7.     while(1);
8.     {
9.         while(RI == 0);      //若為接收中斷
10.        RI=0;
```

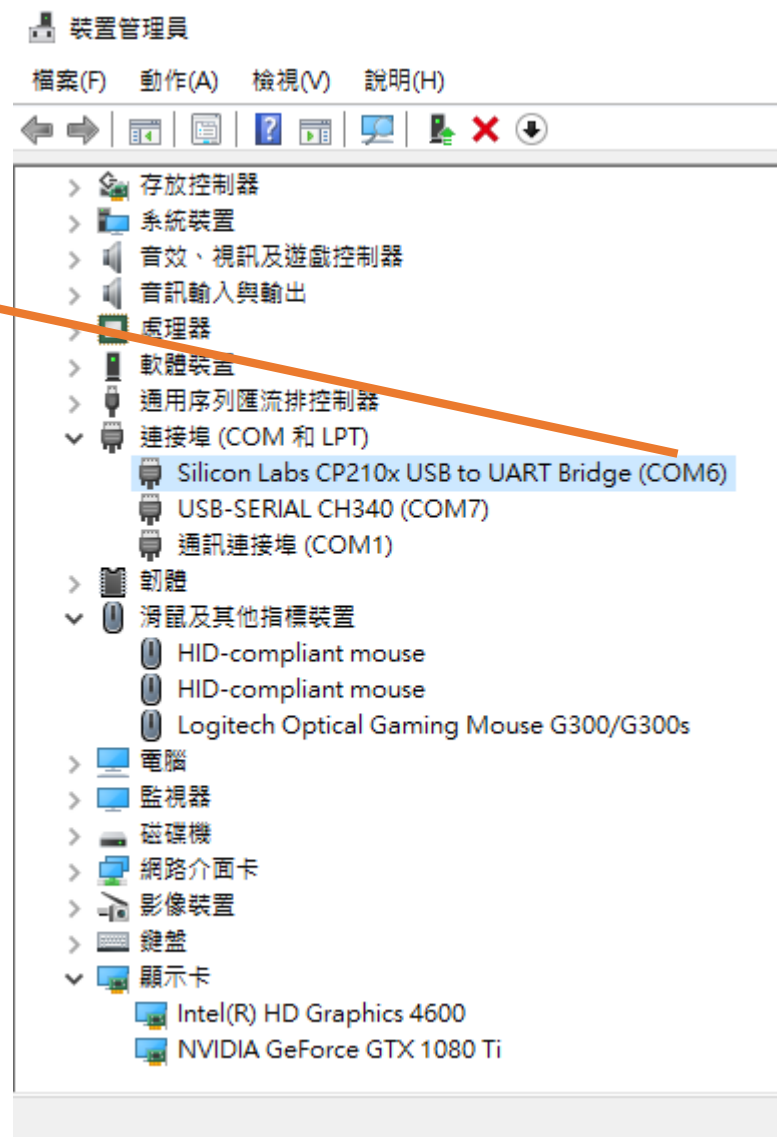
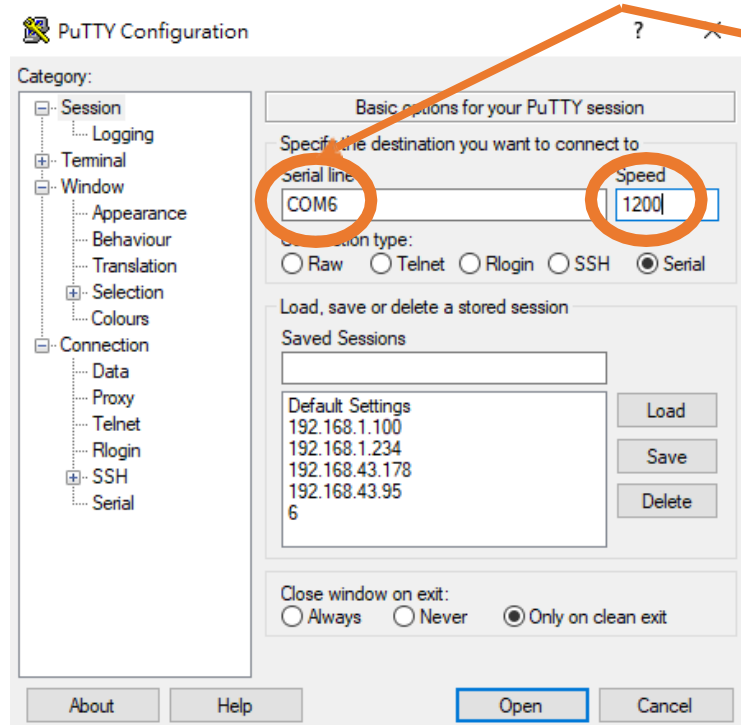
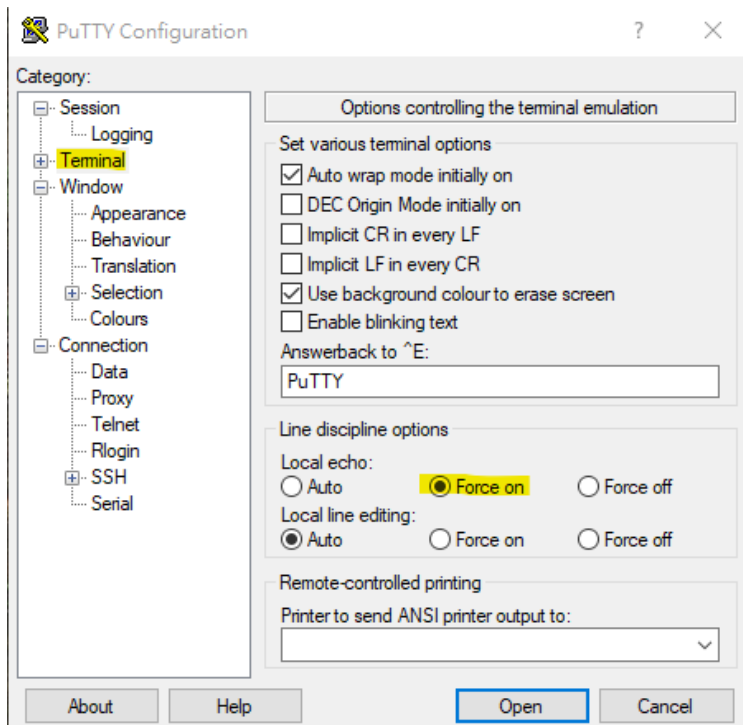
範例程式碼 — 輪詢

```
11.         recvvalue=SBUF;           //當中斷將SBUF的值放入recvvalue中
12.         P2 = recvvalue;           //將輸入的值用LED顯示觀察
13.         if(recvvalue == 'q')       //當接收字元為q
14.         {
15.             SBUF= recvvalue;       //回傳接收到的值
16.             while(TI==0);           //若為傳送中斷
17.             TI=0;                   //將TI旗標設回0
18.         }
19.     }
20. }
```

範例程式碼 — 輪詢

```
21. void init_uart(void)
22. {
23.     SCON=0x50;           //設定SCON暫存器傳輸模式為mode1
24.     TMOD=0x20;          //開啟TIMER1的Mode2
25.     TH1=0xe6;           //baud的設定 1200
26.     TR1=1;              //TCON中Timer1的啟動開關 TR1為1
27. }
```

PuTTY的設定



Q & A