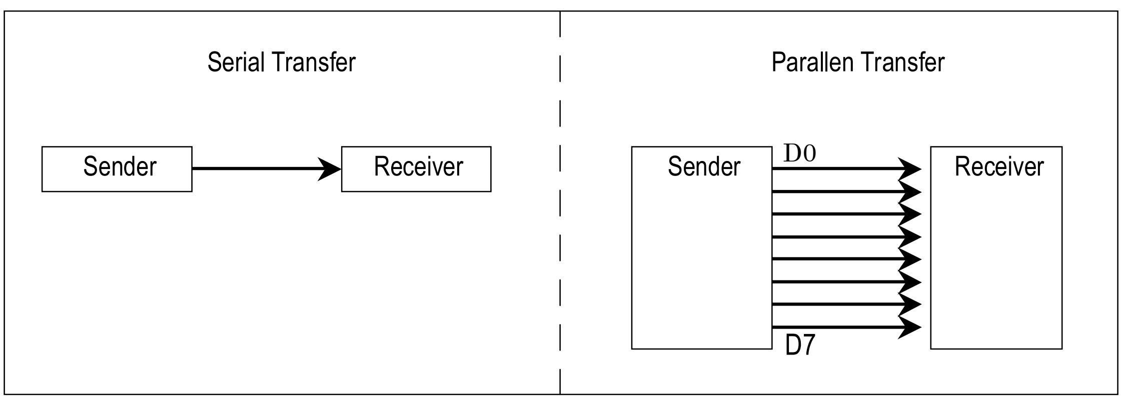
CHƯƠNG 10
Truyền thông nối tiếp của 8051

Các máy tính truyền dữ liệu theo hai cách: Song song và nối tiếp. Trong truyền dữ liệu song song thường cần 8 hoặc nhiều đường dây dẫn để truyền dữ liệu đến một thiết bị chỉ cách xa vài bước. Ví dụ của truyền dữ liệu song song là các máy in và các ổ cứng, mỗi thiết bị sử dụng một đường cáp với nhiều dây dẫn. Mặc dù trong các trường hợp như vậy thì nhiều dữ liệu được truyền đi trong một khoảng thời gian ngắn bằng cách dùng nhiều dây dẫn song song nhưng khoảng cách thì không thể lớn được. Để truyền dưdx liệu đi xa thì phải sử dụng phương pháp truyền nối tiếp. Trong truyền thông nối tiếp dữ liệu được gửi đi từng bít một so với truyền song song thì một hoặc nhiều byte được truyền đi cùng một lúc. Truyền thông nối tiếp của 8051 là chủ đề của chương này. 8051 đã được cài sẵn khả năng truyền thông nối tiếp, do vậy có thể truyền nhánh dữ liệu với chỉ một số ít dây dẫn.

10.1 Các cơ sở của truyền thông nối tiếp.

Khi một bộ vi xử lý truyền thông với thế giới bên ngoài thì nó cấp dữ liệu dưới dạng từng khúc 8 bít (byte) một. Trong một số trường hợp chẳng hạn như các máy in thì thông tin đơn giản được lấy từ đường bus dữ liệu 8 bít và được gửi đi tới bus dữ liệu 8 bít của máy in. Điều này có thể làm việc chỉ khi đường cáp bus không quá dài vì các đường cáp dài làm suy giảm thậm chí làm méo tín hiệu. Ngoài ra, đường dữ liệu 8 bít giá thường đắt. Vì những lý do này, việc truyền thông nối tiếp được dùng để truyền dữ liệu giữa hai hệ thống ở cách xa nhau hàng trăm đến hàng triệu dặm. Hình 10.1 là sơ đồ truyền nối tiếp so với sơ đồ truyền song song.



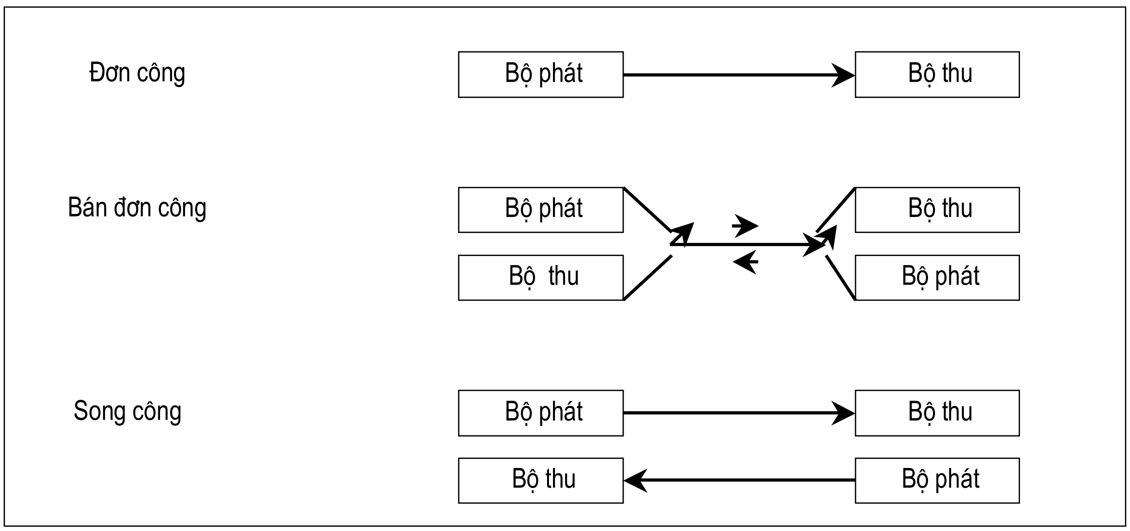
Hình 10.1: Sơ đồ truyền dữ liệu nối tiếp so với sơ đồ truyền song song.

Thực tế là trong truyền thông nối tiếp là một đường dữ liệu duy nhất được dùng thay cho một đường dữ liệu 8 bít của truyền thông song song làm cho nó không chỉ rẻ hơn rất nhiều mà nó còn mở ra khả năng để hai máy tính ở cách xa nhau có truyền thông qua đường thoại.

Đối với truyền thông nối tiếp thì để làm được các byte dữ liệu phải được chuyển đổi thành các bít nối tiếp sử dụng thanh ghi giao dịch vào - song song - ra - nối tiếp. Sau đó nó có thể được truyền quan một đường dữ liệu đơn. Điều này cũng có nghĩa là ở đầu thu cũng phải có một thanh ghi vào - nối tiếp - ra - song song để nhận dữ liệu nối tiếp và sau đó gói chúng thành từng byte một. Tất nhiên, nếu dữ liệu được truyền qua đường thoại thì nó phải được chuyển đổi từ các số 0 và 1 sang âm thanh ở dạng sóng hình sin. Việc chuyển đổi này thực thi bởi một thiết bị có tên gọi là Modem là chữ viết tắt của "Modulator/ demodulator" (điều chế/ giải điều chế).

Khi cự ly truyền ngắn thì tín hiệu số có thể được truyền như nói ở trên, một dây dẫn đơn giản và không cần điều chế. Đây là cách các bàn PC và IBM truyền dữ liệu đến bo mạch mẹ. Tuy nhiên, để truyền dữ liệu đi xa dùng các đường truyền chẳng hạn như đường thoại thì việc truyền thông dữ liệu nối tiếp yêu cầu một modem để điều chế (chuyển các số 0 và 1 về tín hiệu âm thanh) và sau đó giải điều chế (chuyển tín hiệu âm thanh về các số 0 và 1).

Truyền thông dữ liệu nối tiếp sử dụng hai phương pháp đồng bộ và dị bộ. Phương pháp đồng bộ truyền một khối dữ liệu (các ký tự) tại cùng thời điểm trong khi đó truyền dị bộ chỉ truyền từng byte một. Có thể viết phần mềm để sử dụng một trong hai phương pháp này, những chương trình có thể rất dài và buồn tẻ. Vì lý do này mà nhiều nhà sản xuất đã cho ra thị trường nhiều loại IC chuyên dụng phục vụ cho truyền thông dữ liệu nối tiếp. Những IC này phục vụ như các bộ thu - phát dị bộ tổng hợp VART (Universal Asynchronous Receiver Transmitter) và các bộ thu - phát đồng - dị bộ tổng hợp UBART (Universal Asynchronous Receiver Transmitter). Bộ vi điều khiển 8051 có một cài sẵn một UART mà nó sẽ được bàn kỳ ở mục 10.3.



Hình 10.2: Truyền dữ liệu đơn công, bán công và song công.

10.1.1 Truyền dữ liệu bán công và song công.

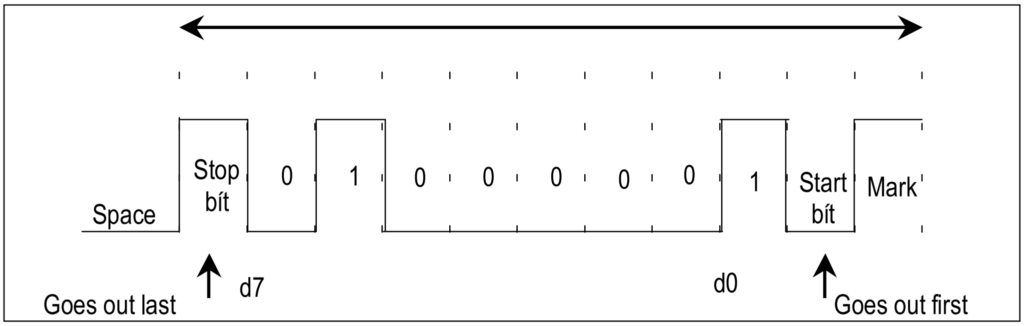
Trong truyền dữ liệu nếu dữ liệu có thể được vừa phát và vừa được thu thì gọi là truyền song công. Điều này tương phản với truyền đơn công chẳng hạn như các máy in chỉ nhận dữ liệu từ máy tính. Truyền song công có thể có hai loại là bán song công và song công hoàn toàn phụ thuộc vào truyền dữ liệu có thể xảy ra đồng thời không? Nếu dữ liệu được truyền theo một đường tại một thời điểm thì được gọi là truyền bán song công. Nếu dữ liệu có thể đi theo cả hai đường cùng một lúc thì gọi là song công toàn phần. Tất nhiên, truyền song công đòi hỏi hai đường dữ liệu (ngoài đường âm của tín hiệu), một để phát và một để thu dữ liệu cùng một lúc.

10.1.2 Truyền thông nối tiếp dị bộ và đóng khung dữ liệu.

Dữ liệu đi vào ở đầu thu của đường dữ liệu trong truyền dữ liệu nói tiếp toàn là các số 0 và 1, nó thật là khó làm cho dữ liệu ấy có nghĩa là nếu bên phát và bên thu không cùng thống nhất về một tệp các luật, một thủ tục, về cách dữ liệu được đóng gói, bao nhiêu bít tạo nên một ký tự và khi nào dữ liệu bắt đầu và kết thúc.

10.1.3 Các bít bắt đầu và dừng.

Truyền thông dữ liệu nối tiếp dị bộ được sử dụng rộng rãi cho các phép truyền hướng kỹ tự, còn các bộ truyền dữ liệu theo khối thì sử dụng phương phát đồng bộ. Trong phương pháp dị bộ, mỗi ký tự được bố trí giữa các bít bắt đầu (start) và bít dừng (stop). Công việc này gọi là đóng gói dữ liệu. Trong đóng gói dữ liệu đối với truyền thông dị bộ thì dữ liệu chẳng hạn là các ký tự mã ASCII được đóng gói giữa một bít bắt đầu và một bít dừng. Bít bắt đầu luôn luôn chỉ là một bít, còn bít dừng có thể là một hoặc hai bít. Bít bắt đầu luôn là bít thấp (0) và các bít dừng luôn là các bít cao (bít 1). Ví dụ, hãy xét ví dụ trên hình 10.3 trong đó ký tự "A" của mã ASCII (8 bít nhị phân là 0100 0001) đóng gói khung giữa một bít bắt đầu và một bít dừng. Lưu ý rằng bít thấp nhất LSB được gửi ra đầu tiên.



Hình 10.3: Đóng khung một ký tự "A" của mã ASCII (41H) có tín hiệu là 1 (cao) được coi như là một dấu (mark), còn không có tín hiệu tức là 0 (thấp) thì được coi là khoảng trống (space). Lưu ý rằng phép truyền bắt đầu với start sau đó bít D0, bít thấp nhất LSB, sau các bít còn lại cho đến bít D7, bít cao nhất MSB và cuối cùng là bít dừng stop để báo kết thúc ký tự "A".

Trong truyền thông nối tiếp dị bộ thì các chíp IC ngoại vi và các modem có thể được lập trình cho dữ liệu với kích thước theo 7 bít hoặc 8 bít. Đây là chưa kể các bít dừng stop có thể là 1 hoặc 2 bít. Trong khi các hệ ASCII cũ hơn (trước đây) thì các ký tự là 7 bít thì ngay nay do việc mở rộng các ký tự ASCII nên dữ liệu nhìn chung là 8 bít. Trong các hệ cũ hơn do tốc độ chậm của các thiết bị thu thì phải sử dụng hai bít dừng để đảm bảo thời gian tổ chức truyền byte kế tiếp. Tuy nhiên, trong các máy tính PC hiện tại chỉ sử dụng 1 bít stop như là chuẩn.

Giả sử rằng chúng ta đang truyền một tệp văn bản các ký tự ASCII sử dụng 1 bít stop thì ta có tổng cộng là 10 bít cho mỗi ký tự gồm: 8 bít cho ký tự ASCII chuẩn và 1 bít start cùng 1 bít stop. Do vậy, đối với mỗi ký tự 8 bít thì cần thêm 2 bí vị chi là mất 25% tổng phí.

Trong một số hệ thống để nhằm duy trì tính toàn vẹn của dữ liệu thì người ta còn thêm một bít lẻ (parity bít). Điều này có nghĩa là đối với mỗi ký tự (7 hoặc 8 bít tuỳ từng hệ) ta có thêm một bít ngoài các bít start và stop. Bít chẵn lẻ là bít chẵn hoặc bít lẻ. Nếu là bít lẻ là số bít của dữ liệu bao gồm cả bít chẵn lẻ sẽ là một số lẻ các sô 1. Tương tự như vậy đối với trường hợp bít chẵn thì số bít của dữ liệu bao gồm cả bít chẵn - lẻ sẽ là một số chẵn của các số 1. Ví dụ, ký tự "A" của mã ASCII ở dạng nhị phân là 0100 0001, có bít 0 là bít chẵn. Các chíp UART đều cho phép việc lập trình bít chẵn - lẻ về chẵn, lẻ hoặc không phân biệt chẵn lẻ.

10.1.4 Tốc độ truyền dữ liệu.

Tốc độ truyền dữ liệu trong truyền thông dữ liệu nối tiếp được gọi là bít trong giây bps (bít per second). Ngoài ra, còn được sử dụng một thuật ngữ rộng rãi nữa là tốc độ baud. Tuy nhiên, các tốc baud và bps là hoàn toàn không bằng nhau. Điều này là do tốc baud là thuật ngữ của modem và được định nghĩa như là sô lần thay đổi của tín hiệu trong một giây. Trong các modem có những trường hợp khi một sự thay đổi của tín hiệu thì nó truyền vài bít dữ liệu. Nhưng đối với một dây dẫn thì tốc độ baud và bps là như nhau nên trong cuốn sách này chúng ta có thể dùng thay đổi các thuật ngữ này cho nhau.

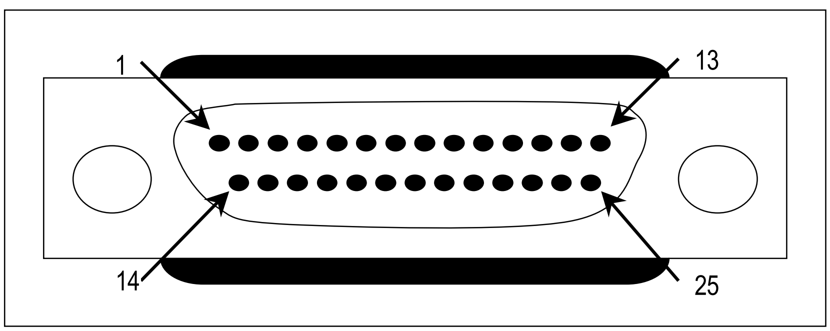
Tốc độ truyền dữ liệu của một hệ máy tính đã cho phụ thuộc vào các cổng truyền thông kết nối vào trong hệ thống đo. Ví dụ, các máy tính PC/XT trước đây của IBM có thể truyền dữ liệu với tốc độ 100 đến 9600 bps. Tuy nhiên, trong những năm gần đây thì các máy tính PC dựa trên Pentium truyền dữ liệu với tốc độ lên tới 56kbps. Cần phải nói thêm rằng trong truyền thông dữ liệu nối tiếp dị bộ thì tốc độ baud nhìn chung là bị giới hạn ở 100.000 bps.

10.1.5 Các chuẩn RS232.

Để cho phép tương thích giữa các thiết bị truyền thông dữ liệu được sản xuất bởi các hãng khác nhau thì một chuẩn giao diện được gọi là RS232 đã được thiết lập bởi hiệp hội công nghiệp điện tử EIA vào năm 19960. Năm 1963 nó được sửa chỉnh và được gọi là RS232A và vào các năm 1965 và 1969 thì được đổi thành RS232B và RS232C. Ở đây chúng ta đơn giản chỉ nói đến RS232. Ngày nay RS232 là chuẩn giao diện I/O vào - ra nối tiếp được sử dụng rộng rãi nhất. Chuẩn này được sử dụng trong máy tính PC và hàng loạt các thiết bị khác nhau. Tuy nhiên, vì nó được thiết lập trước họ lô-gíc TTL rất lâu do vậy điện áp đầu vào và đầu ra của nó không tương thích với mức TTL. Trong RS232 thì mức 1 được biểu diễn bởi - 3v đến 25v trong khi đó mức 0 thì ứng với điện áp + 3v đến +25v làm cho điện áp - 3v đến + 3v là không xác định. Vì lý do này để kết nối một RS232 bất kỳ đến một hệ vi điều khiển thì ta phải sử dụng các bộ biến đổi điện áp như MAX232 để chuyển đổi các mức lô-gíc TTL về mức điện áp RS232 và ngược lại. Các chíp IC MAX232 nhìn chung được coi như cá bộ điều khiển đường truyền. Kết nối RS232 đến MAX232 được thoả thuận ở phần 10.2.

10.1.6 Các chân của RS232.

Bảng 10.1 cung cấp sơ đồ chân của cáp RSE232 và các tên gọi của chúng thường được gọi là đầu nối DB - 25. Trong lý hiệu thì đầu nối cắm vào (đầu đực) gọi là DB - 25p và đầu nối cái được gọi là DB - 25s.



Hình 10.4: Đầu nối DB - 25 của RS232.

Vì không phải tất cả mọi chân đều được sử dụng trong cáp cảu máy tính PC, nên IBM đưa ra phiên bản của chuẩn vào/ra nối tiếp chỉ sử dụng có 9 chân gọi là DB - 9 như trình bày ở bảng 10:2 và hình 10.5.

Bảng 10.1: Các chân của RS232, 25 chân (DB - 25).

Số chân	Mô tả
1	Đất cách ly (Protective Cround)
2	Dữ liệu được truyền TxD (TráNsmitted data)
3	Dữ liệu được phân RxD (Received data)
4	Yêu cầu gửi RTS (Request To Send)
5	Xoá để gửi CIS (Clear To Send)
6	Dữ liệu sẵn sàng DSR (Data Set Ready)
7	Đất của tín hiệu GND (Signal Cround)
8	Tách tín hiệu mạng dữ liệu DCD (Data Carrier Detect)
9/10	Nhận để kiểm tra dữ liệu (Received for data testing)

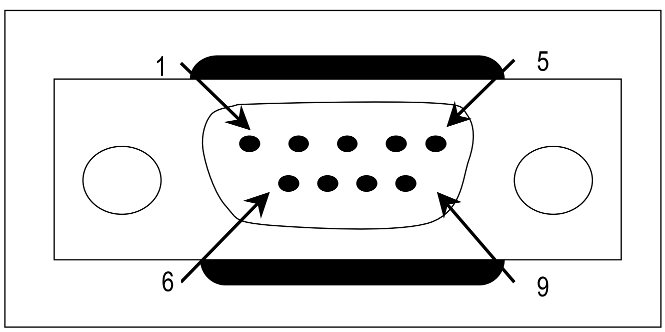
11	Chưa dùng
12	Tách tín hiệu mạng dữ liệu thứ cấp (Secondary data carrier detect)
13	Xoá để nhận dữ liệu thứ cấp (Secondary Clear to Send)
14	Dữ liệu được truyền thứ cấp (Secondary Transmit Signal Element Timing)
15	Truyền phân chia thời gian phần tử tín hiệu (Transmit Signal Element
16	Timing)
17	Dữ liệu được nhận thứ cấp (Secondary Received data)
18	Nhận phân chia thời gian phần tử tín hiệu (Receiveo Signal Element
19	Timing)
20	Chưa dùng
21	Yêu cầu để nhận thứ cấp (Secondary Request to Send)
22	Đầu dữ liệu sẵn sàng (Data Terminal Ready)
23	Phát hiện chất lượng tín hiệu (Signal Qualyty Detector)
24	Báo chuông (Ring Indicator)
25	Chọn tốc độ tín hiệu dữ liệu (Data Signal Rate Select)
	Truyền phân chia thời gian tín hiệu (Transmit Signal Element Timing)
	Chưa dùng

10.1.7 Phân loại truyền thông dữ liệu.

Thuật ngữ hiện nay phân chia thiết bị truyền thông dữ liệu thành một thiết bị đầu cuối dữ liệu DTE (Data Terminal Equipment) hoặc thiết bị truyền thông dữ liệu DCE (Data Communication Equipment). DTE chủ yếu là các máy tính và các thiết bị đầu cuối gửi và nhận dữ liệu, còn DCE là thiết bị truyền thông chẳng hạn như các modem chịu trách nhiệm về truyền dữ liệu. Lưu ý rằng tất cả mọi định nghĩa về chức năng các chân RS232 trong các bảng 10.1 và 10.2 đều xuất phát từ gốc độ của DTE.

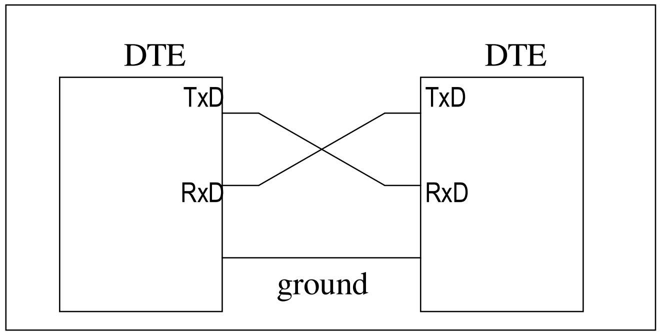
Kết nối đơn giản nhất giữa một PC và bộ vi điều khiển yêu cầu tối thiểu là những chân sau: TxD, RxD và đất như chỉ ra ở hình 10.6. Để ý rằng trên hình này thì các chân TxD và RxD được đổi cho nhau.

Hình 10.5: Sơ đồ đầu nối DB - 9 của RS232.



Bảng 10.2: Các tín hiệu của các chân đầu nối DB - 9 trên máy tính IBM PC.

Mô tả	Số chân	
1	Da ta carrier detect (DCD)	Tránh tín hiệu mạng dữ liệu
2	Received data (RxD)	Dữ liệu được nhận
3	Transmitted data (TxD)	Dữ liệu được gửi
4	Data terminal ready (DTR)	Đầu dữ liệu sẵn sàng
5	Signal ground (GND)	Đất của tín hiệu
6	Data set ready (DSR)	Dữ liệu sẵn sàng
7	Request to send (RTS)	Yêu cầu gửi
8	Clear to send (CTS)	Xoá để gửi
9	Ring indicator (RL)	Báo chuông



Hình 10.6: Nối kết không modem.

10.1.8 Kiểm tra các tín hiệu bắt tay của RS232.

Để bảo đảm truyền dữ liệu nhanh và tin cậy giữa hai thiết bị thì việc truyền dữ liệu phải được phối hợp tốt. Chẳng hạn như trong trường hợp của máy in, do một thực tế là trong truyền thông dữ liệu nối tiếp thiết bị thu có thể không có chỗ để chứa dữ liệu, do đó phải có cách để báo cho bên phát dừng gửi dữ liệu. Rất nhiều chân của RS232 được dùng cho các tín hiệu bắt tay. Dưới đây là mô tả về chúng như là một tham khảo và chúng có thể được bỏ qua vì chúng không được hỗ trợ bởi chíp UART của 8051.

1. Đầu dữ liệu sẵn sàng DTR: Khi thiết bị đầu cuối (hoặc một cổng COM của PC) được bật thì sau khi tự kiểm tra nó gửi một tín hiệu DTR báo rằng nó sẵn sàng cho truyền thông. Nếu có một cái gì đó trục trặc với cổng COM thì tín hiệu này không được kích hoạt. Đây là tín hiệu tích cực mức thấp và có thể được dùng để báo cho modem biết rằng máy tính đang hoạt động và đang sẵn sàng. Đây là chân đầu ra từ DTC (cổng COM của PC) và chân đầu ra của modem.
2. Dữ liệu sẵn sàng QSR: Khi DCE (chẳng hạn modem) được bật lên và đã chạy xong chương trình tự kiểm tra thì nó đòi hỏi DSR để báo rằng có đã sẵn sàng cho truyền thông. Do vậy, nó là đầu ra của modem (DCE) và đầu vào của PC (DTE). Đây là tín hiệu tích cực mức thấp. Nếu vì lý do nào đó mà modem không kích hoạt báo cho PC biết (hoặc thiết bị đầu cuối) rằng nó không thể nhận hoặc gửi dữ liệu.
3. Yêu cầu gửi RTS: Khi thiết bị DTE (chẳng hạn một PC) có một byte dữ liệu cần gửi thì nó yêu cầu RTS để báo cho modem biết rằng nó có một byte cần phải gửi đi. RTS là một đầu ra tích cực mức thấp từ DTE và một đầu vào tới modem.
4. Tín hiệu xáo để gửi CTS: Để đáp lại RTS thì khi modem có để chứa dữ liệu mà nó cần nhận thì nó gửi một tín hiệu CTS tới DTE (PC) để báo rằng bây giờ nó có thể nhận dữ liệu. Tín hiệu đầu vào này tới DTE dùng để khởi động việc truyền dữ liệu.
5. Tách tín hiệu mang dữ liệu DCD: Modem yêu cầu tín hiệu DCD báo cho DTE biết rằng đã tách được một tín hiệu mang dữ liệu hợp lệ và rằng kết nối giữa nó và modem khác đã được thiết lập. Do vậy, DCD là một đầu ra của modem và đầu vào của PC (DTE).
6. Báo chuông RI: Một đầu ra từ modem (DCE) và một đầu vào tới máy tính PC (DTE) báo rằng điện thoại đang báo chuông. Nó tắt và bật đồng bộ với âm thanh đang đổ chuông. Trong 6 tín hiệu bắt tay thì tín hiệu này là ít được dùng nhất do một thực tế là các modem đã chịu trách nhiệm về trả lời điện thoại. Tuy nhiên, nếu trong một hệ thống đã cho mà PC phải chịu trách nhiệm trả lời điện thoại thì tín hiệu này có thể được dùng.

Từ mô tả trên thì việc truyền thông PC và modem có thể được tóm tắt như sau: Trong khi các tín hiệu DTR và DSR được dùng bởi PC và modem để báo rằng chúng đang hoạt động tốt thì các tín hiệu RTS và CTS thực tế đang kiểm tra luồng dữ liệu. Khi PC muốn gửi dữ liệu thì nó yêu cầu RTS và đáp lại, nếu modem sẵn sàng (có chỗ chứa dữ liệu) để nhận dữ liệu thì nó gửi lại tín hiệu CTS. Còn nếu không có chỗ cho dữ liệu thì modem không kích hoạt CTS và PC thôi không yêu cầu DTR và thử lại. Các tín hiệu RTS và CTS cũng được cọi như tín hiệu luồng điều khiển phần cứng.

Đến đây kết thúc sự mô tất 9 chân quan trọng nhất của các tín hiệu bắt tay RS232 và các tín hiệu TxD, RxD và đất (Ground). Tín hiệu Ground này cũng được coi như là tín hiệu SG - đất của tín hiệu.

10.1.9 Các cổng COM của IBM PC và tương thích.

Các máy tính IBM PC và tương thích dựa trên các bộ vi xử lý × 86 (8086, 286, 384, 486 và Pentium) thường có hai cổng COM. Cả hai cổng COM đều có các đầu nối kiểu RS232. Nhiều máy tính PC sử dụng mỗi đầu nối một kiểu ổ cắm DB - 25 và DB - 9. Trong những năm gần đây, cổng COM1 được dùng cho chuột và COM2 được dùng cho các thiết bị chẳng hạn như Modem. Chúng ta có thể nối cổng nối tiếp của 8051 đến cổng COM2 của một máy tính PC cho các thí nghiệm về truyền thông nối tiếp.

Với nền kiến thức về truyền thông nối tiếp này chúng ta đã sẵn sàng làm việc với 8051.

10.2 Nối ghép 8051 tới RS232.

Như đã nói ở phần 10.1, chuẩn RS232 không tương thích với mức lô-gíc TTL, do vậy nó yêu cầu một bộ điều khiển đường truyền chẳng hạn như chíp MAX232 để chuyển đổi các mức điện áp RS232 về các mức TTL và ngược lại. Nội dung chính của phần này là bàn về nối ghép 8051 với các đầu nối RS232 thông qua chíp MAX232.

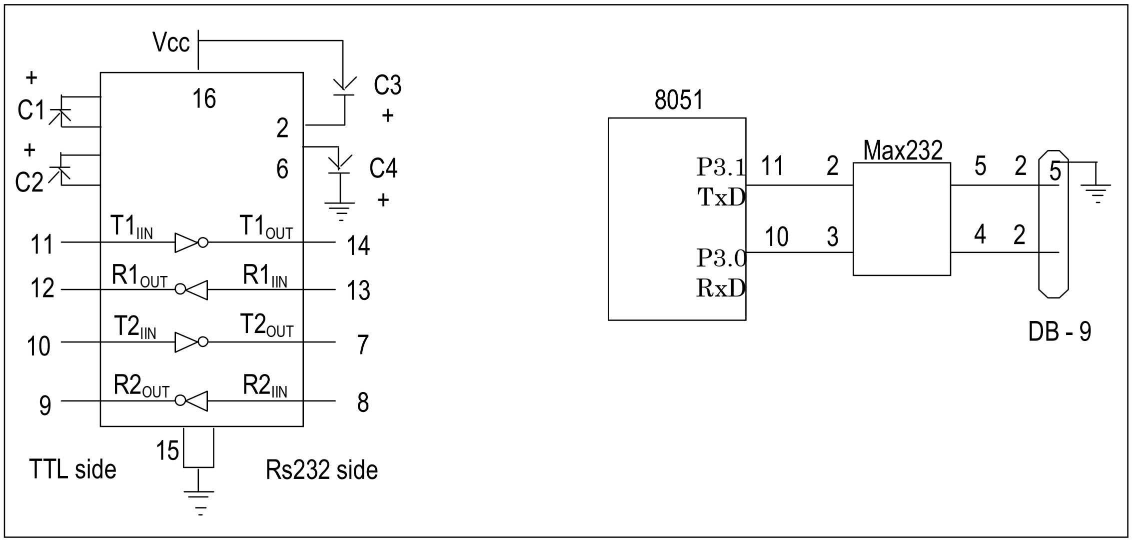
10.2.1 Các chân RxD và TxD trong 8051.

8051 có hai chân được dùng chuyên cho truyền và nhận dữ liệu nối tiếp. Hai chân này được gọi là TxD và RxD và là một phần của cổng P3 (đó là P3.0 và P3.1). chân 11 của 8051 là P3.1 được gán cho TxD và chân 10 (P3.0) được dùng cho RxD. Các chân này tương thích với mức lô-gích TTL. Do vậy chúng đòi hỏi một bộ điều khiển đường truyền để chúng tương thích với RS232. Một bộ điều khiển như vậy là chíp MAX232.

10.2.2 Bộ điều khiển đường truyền MAX232.

Vì RS232 không tương thích với các bộ vi xử lý và vi điều khiển hiện nay nên ta cần một bộ điều khiển đường truyền (bộ chuyển đổi điện áp) để chuyển đổi các tín hiệu RS232 về các mức điện áp TTL sẽ được chấp nhận bởi các chân TxD và RxD của 8051. Một ví dụ của một bộ chuyển đổi như vậy là chíp MAX232 từ hàng Maxim địa chỉ Website của hãng www.maxim-ic.com. Bộ MAX232 chuyển đổi từ các mức điện áp RS232 sẽ về mức điện áp TTL và ngược lại. Một điểm mạnh của chíp MAX232 là nó dùng điện áp nguồng +5v cùng với điện áp nguồn của 8051. Hay nóic cách khác với nguồn điện áp nuối +5 chúng ta mà có thể nuôi 8051 và MAX232 mà không phải dùng hai nguồn nuôi khác nhau như phổ biến trong các hệ thống trước đây.

Bộ điều khiển MAX232 có hai bộ điều khiển thường để nhận và truyền dữ liệu như trình bày trên hình 10.7. Các bộ điều khiển đường được dùng cho TxD được gọi là T1 và T2. Trong nhiều ứng dụng thì chỉ có một cặp được dùng. Ví dụ T1 và R1 được dùng với nhau đối với TxD và RxD của 8051, còn cặp R2 và T2 thì chưa dùng đến. Để ý rằng trong MAX232 bộ điều khiển T1 có gán $T1_{in}$ và $T1_{out}$ trên các chân số 11 và 1 tương ứng. Chân $T1_{in}$ là ở phía TTL và được nối tới chân RxD của bộ vi điều khiển, còn $T1_{out}$ là ở phía RS232 được nối tới chân RxD của đầu nối DB của RS232. Bộ điều khiển đường R1 cũng có gán $R1_{in}$ và $R1_{out}$ trên các chân số 13 và 12 tương ứng. Chân $R1_{in}$ (chân số 13) là ở phía RS232 được nối tới chân TxD của đầu nối DB của RS232 và chân $R1_{out}$ (chân số 12) là ở phía TTL mà nó được nối tới chân RxD của bộ vi điều khiển, xem hình 10.7. Để ý rằng nối ghép modem không là nối ghép mà chân TxD bên phát được nối với RxD của bên thu và ngược lại.



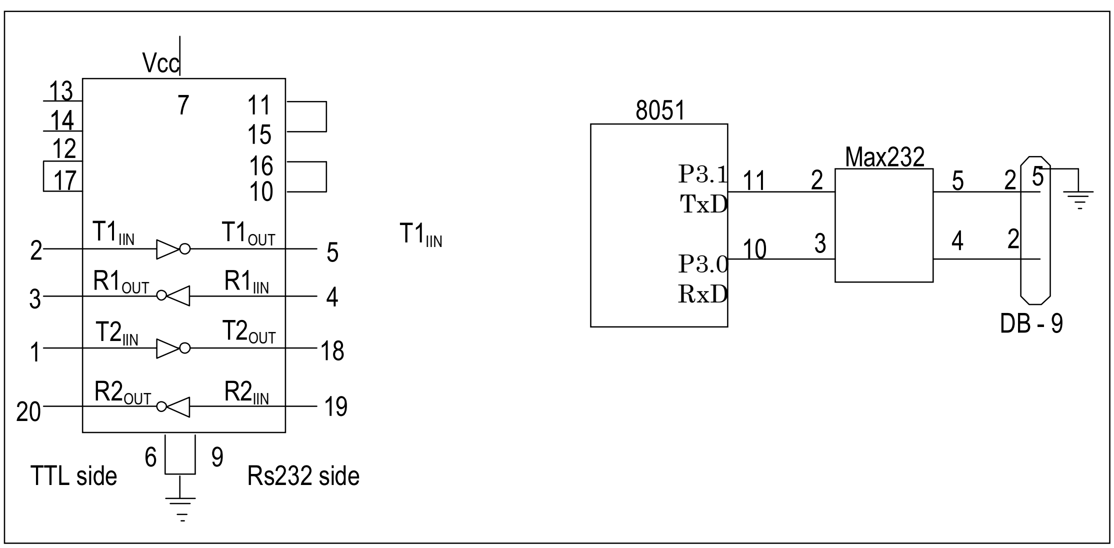
Hình 10.7: a) Sơ đồ bên trong của MAX232

b) Sơ đồ nối ghép của MAX232 với 8051 theo moden không.

Bộ MAX232 đòi hỏi 4 tụ điện giá trị từ 1 đến 22μF. Giá trị phổ biến nhất cho các tụ này là 22μF.

10.2.3 Bộ điều khiển MAX232.

Để tiết kiệm không gian trên bảng mạch, nhiều nhà thiết kế sử dụng chíp MAX232 từ hãng Maxim. Bộ điều khiển MAX232 thực hiện cùng những công việc như MAX232 lại không cần đến các tụ điện. Tuy nhiên, chíp MAX232 lại đắt hơn rất nhiều so với MAX233 không có sơ đồ chân giống nhau (không tương thích). Chúng ta không thể lấy một chíp MAX232 ra khỏi một bảng mạch và thay vào đó RS233. Hãy xem hình 10.8 để thấy MAX233 không cần đến tụ.



Hình 10.8: a) Sơ đồ bên trong của MAX233.

b) Sờ đồ nối ghép của MAX233 với 8051 theo modem không.

10.3 Lập trình truyền thông nối tiếp cho 8051.

Trong phần này chúng ta sẽ nghiên cứu về các thanh ghi truyền thông nối tiếp của 8051 và cách lập trình chúng để truyền và nhận dữ liệu nối tiếp. Vì các máy tính IBM PC và tương thích được sử dụng rất rộng rãi để truyền thông với các hệ dựa trên 8051, do vậy ta chủ yếu tập trung vào truyền thông nối tiếp của 8051 với cổng COM của PC. Để cho phép truyền dữ liệu giữa máy tín PC và hệ thống 8051 mà không có bất kỳ lỗi nào thì chúng ta phải biết chắc rằng tốc độ baud của hệ 8051 phải phù hợp với tốc độ baud của công COM máy tính PC được cho trong bảng 10.3. Chúng ta có thể kiểm tra các tốc độ baud này bằng cách vào chương trình Windows Terminal và bấm chuột lên tuỳ chọn Communication Settings. Chương trình Terminal.exe của Window3.1 cũng làm việc tốt trên Windows95 và Window98. Trong Window95 và cao hơn ta có thể sử dụng chức năng Hyperterrminal. Hàm Hyperterminal hỗ trợ các tốc độ Baud cao hơn nhiều so với các tốc độ cho trong bảng 10.3.

Bảng 10.3: Các tốc độ Baud của máy tính PC486 và Pentium cho trong BIOS.

100	150	300	600	1200	2400	4800	9600	19200

Ví dụ 10.1:

Với tần số XTAL là 11.0592MHz. Hãy tìm giá trị TH1 cần thiết để có tốc độ baud sau:

a) 9600 b) 2400 c) 1200

Lời giải:

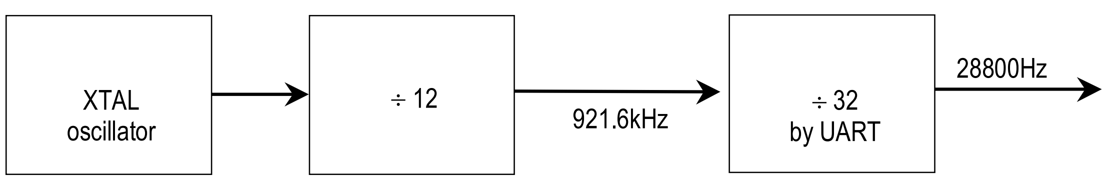
Với tần số XTAL là 11.0592MHz thì ta có tần số chu trình máy của 8051 là 11.0592MHz : 12 = 921.6kHz và sau đó lấy 921.6kHz/32 = 28.800Hz là tần số được cấp bởi UART tới bộ định thời Timer1 để thiết lập tốc độ.

a) 28.800/3 = 9600 trong đó - 3 = FD được nạp vào TH1

b) 28.800/12 = 2400 trong đó - 12 = F4 được nạp vào TH1

c) 28.800/24 = 1200 trong đó - 24 = F8 được nạp vào TH1

Lưu ý rằng việc chia 1/12 của tần số thạch anh cho 32 là giá trị mặc định khi kích hoạt chân RESET của 8051. Chúng ta có thể thay đổi giá trị cài đặt mặc định này. Điều này sẽ được giải thích ở cuối chương.



10.3.1 Tốc độ baud trong 8051.

8051 truyền và nhận dữ liệu nối tiếp theo nhiều tốc độ khác nhau. Tốc độ truyền của nó có thể lập trình được. Điều ày thực hiện nhờ sự trợ giúp của bộ định thời Timer1. Trước khi ta đi vào bàn cách làm điều đó như thế nào thì ta sẽ xét quan hệ giữa tần số thạch anh và tốc độ baud trong 8051.

Như ta đã nói ở chương trước đây thì 8051 chia sóo thạch anh cho 12 để lấy tần số chu trình máy. Trong trường hợp XTAL = 11.0592MHz thì tần số chu trình là 921.6kHz (11.0592MHz : 12 = 921.6kHz). Mạch điện UART truyền thông nối tiếp của 8051 lại chia tần số chu trình máy cho 32 một lần nữa trước khi nó được dùng bởi bộ định thời gian Timer1 để tạo ra tốc độ baud. Do vậy, 921.6kHz : 32 = 28.800Hz. Đây là số ta sẽ dùng trong cả phần này để tìm giá trị của Timer1 để đặt tốc độ baud. Muốn Timer1 đặt tốc độ baud thì nó phải được lập trình về chế độ làm việc mode2, đó là chế độ thanh ghi 8 bít tự động nạp lại. Để có tốc độ baud tương thích với PC ta phải nạp TH1 theo các giá trị cho trong bảng 10.3. Ví dụ 10.1 trình bày cách kiểm tra giá trị dữ liệu cho trong bảng 10.3.

Bảng 10.3: Các giá trị của thanh ghi TH1 trong Timer1 cho các tốc độ baud khác nhau.

Tốc độ baud	TH1 (thập phân)	TH1 (số Hex)
9600	- 3	FD
4800	- 6	FA
2400	- 12	F4
1200	- 24	F8

10.3.2 Thanh ghi SBUF.

SBUF là thanh ghi 8 bít được dùng riêng cho truyền thông nối tiếp trong 8051. Đối với một byte dữ liệu cần phải được truyền qua đường TxD thì nó phải được đặt trong thanh ghi SBUF. Tương tự như vậy SBUF giữ một byte dữ liệu khi nó được nhận bở đường RxD của 8051. SBUF có thể được truy cập bởi mọi thanh ghi bất kỳ trong 8051. Xét một ví dụ dưới đây để thấy SBUF được truy cập như thế nào?

```
MOV    SBUF, # “D”    ; Nạp vào SBUF giá trị 44H mã ACSII của ký tự D.
MOV    SBUF, A        ; Sao thanh ghi A vào SBUF.
MOV    A, SBUF        ; Sao SBUF vào thanh ghi A.
```

Khi một byte được ghi vào thanh ghi SBUF nó được đóng khung với các bít Start và Stop và đường truyền nối tiếp quan chân TxD. Tương tự như vậy, khi các bít được nhận nối tiếp từ RxD thì 8051 mở khung nó để loại trừ các bít Start và Stop để lấy ra một byte từ dữ liệu nhận được và đặt nó vào thanh ghi SBUF.

10.3.3 Thanh ghi điều khiển nối tiếp SCON.

Thanh ghi SCON là thanh ghi 8 bít được dùng để lập trình việc đóng khung bít bắt đầu Start, bít dừng Stop và các bít dữ liệu cùng với việc khác.

Dưới đây là mô tả các bít khác nhau của SCON:

SM0	SM1	SM2	REN	TB8	RB8	T1	R1

SM0	SCON.7	Số xác định chế độ làm việc cổng nối tiếp
SM1	SCON.6	Số xác định chế độ làm việc cổng nối tiếp
SM2	SCON.5	Dùng cho truyền thông giữa các bộ vi xử lý (SM2 = 0)
REN	SCON.4	Bật/xoá bằng phần mềm để cho phép/ không cho thu
TB8	SCON.3	Không sử dụng rộng rãi
RB8	SCON.2	Không sử dụng rộng rãi
T1	SCON.1	Cờ ngắt truyền } đặt bằng phần cứng khi bắt đầu bít Stop ở chế độ 1.
R1	SCON.0	Cờ ngắt thu } Xoá bằng phần mềm.

Hình 10.2: Thanh ghi điều khiển cổng nối tiếp SCON.

10.3.3.1 Các bít SM0, SM1.

Đây là các bít D7 và D6 của thanh ghi SCON. Chúng được dùng để xác định chế độ đóng khung dữ liệu bằng cách xác định số bít của một ký tự và các bít Start và Stop. Các tổ hợp của chúng là:

SM0	*SM1*	
0	0	Chế độ nối tiếp 0
0	1	Chế độ nối tiếp 1, 8 bít dữ liệu, Start, Stop
1	0	Chế độ nối tiếp 2
1	1	Chế độ nối tiếp 3

Trong bốn chế độ ta chỉ quan tâm đến chế độ 1, các chế độ khác được giải thích ở Appendisk A3. Trong thanh ghi SCON khi chế độ 1 được chọn thì dữ liệu được đóng khung gồm 8 bít dữ liệu, 1 bít Start, 1 bít Stop để tương thích với cổng COM của IBM PC và các PC tương thích khác. Quan trọng hơn là chế độ nối tiếp 1 cho phép tốc độ baud thay đổi và được thiết lập bởi Timer1 của 8051. Trong chế độ nối tiếp 1 thì mỗi ký tự gồm có 10 bít được truyền trong đó có bít đầu là bít Start, sau đó là 8 bít dữ liệu và cuối cùng là bít Stop.

10.3.3.2 Bít SM2.

Bít SM2 là bít D5 của thanh ghi SCON. Bít này cho phép khả năng đa xử lý của 8051 và nó nằm ngoài phạm vi trình bày của chương này. Đối với các ứng dụng của chúng ta đặt SM2 = 0 vì ta không sử dụng 8051 trong môi trường đa xử lý.

10.3.3.3 Bít REN.

Đây là bít cho phép thu (Receive Enable), bít D4 của thanh ghi SCON. Bít REN cũng được tham chiếu như là SCON.4 vì SCON là thanh ghi có thể đánh địa chỉ theo bít. Khi bít REN cao thì nó cho phép 8051 thu dữ liệu trên chân RxD của nó. Và kết quả là nếu ta muốn 8051 vừa truyền và nhận dữ liệu thì bít REN phải được đặt lên 1. Khi đặt REN thì bộ thu bị cấm. Việc đặt REN = 1 hay REN = 0 có thể đạt được bằng lệnh “SETB SCON.4” và “CLR SCON.4” tương ứng. Lưu ý rằng các lệnh này sử dụng đặc điểm đánh địa chỉ theo bít của thanh ghi SCON. Bít này có thể được dùng để khống chế mọi việc nhận dữ liệu nối tiếp và nó là bít cực kỳ quan trọng trong thanh ghi SCON.

10.3.3.4 Bít TB8 và RB8.

Bít TB8 là bít SCON.3 hay là bít D3 của thanh ghi SCON. Nó được dùng để cho chế độ nối tiếp 2 và 3. Ta đặt TB8 vì nó không được sử dụng trong các ứng dụng của mình.

Bít RB8 (bít thu 8) là bít D2 của thanh ghi SCON. Trong chế độ nối tiếp 1 thì bít này nhận một bản sao của bít Stop khi một dữ liệu 8 bít được nhận. Bít này cũng như bít TB8 rất hiếm khi được sử dụng. Trong các ứng dụng của mình ta đặt RB8 = 0 vì nó được sử dụng cho chế độ nối tiếp 2 và 3.

10.3.3.5 Các bít TI và RI.

Các bít ngắt truyền TI và ngắt thu RI là các bít D1 và D0 của thanh ghi SCON. Các bít này là cực kỳ quan trọng của thanh ghi SCON. Khi 8051 kết thúc truyền một ký tự 8 bít thì nó bật TI để báo rằng nó sẵn sàng truyền một byte khác. Bít TI được bật lên trước bít Stop. Còn khi 8051 nhận được dữ liệu nối tiếp qua chân RxD và nó tách các bít Start và Stop để lấy ra 8 bít dữ liệu để đặt vào SBUF, sau khi hoàn tất nó bật cờ RI để báo rằng nó đã nhận xong một byte và cần phải lấy đi kẻo nó bị mất cờ RI được bật khi đang tách bít Stop. Trong các ví dụ dưới đây sẽ nói về vai trò của các bít TI và RI.

10.3.4 Lập trình 8051 để truyền dữ liệu nối tiếp.

Khi lập trình 8051 để truyền các byte ký tự nối tiếp thì cần phải thực hiện các bước sau đây:

1. Nạp thanh ghi TMOD giá trị 204 báo rằng sử dụng Timer1 ở chế độ 2 để thiết lập chế độ baud.

2. Nạp thanh ghi TH1 các giá trị cho trong bảng 10.4 để thiết lập chế độ baud truyền dữ liệu nối tiếp (với giả thiết tần số XTAL = 11.0592MHz).
3. Nạp thanh ghi SCON giá trị 50H báo chế độ nối tiếp 1 để đóng khung 8 bít dữ liệu, 1 bít Start và 1 bít Stop.
4. Bật TR1 = 1để khởi động Timer1.
5. Xoá bít TI bằng lệnh "CLR TI"
6. Byte ký tự cần phải truyền được ghi vào SBUF.
7. Bít cờ TI được hiển thị bằng lệnh "JNB TI, xx" để báo ký tự đã được truyền hoàn tất chưa.
8. Để truyền ký tự tiếp theo quay trở về bước 5.

Ví dụ 10.2 trình bày chương trình để truyền nối tiếp với tốc độ 4800 baud. Ví dụ 10.3 trình bày cách truyền liên tục chữ "YES".

Ví dụ 10.2:

Hãy viết chương trình cho 8051 để truyền nối tiếp một ký tự "A" với tốc độ 4800 baud liên tục.

Lời giải:

```
        MOV     TMOD, #20H      ; Chọn Timer1, chế độ 2 (tự động nạp lại)
        MOV     TH1, # - 6      ; Chọn tốc độ 4800 baud
        MOV     SCON, #A"       ; Truyền 8 bít dữ liệu, 1 bít Stop cho phép thu
        SETB    TR1             ; Khởi động Timer1
AGAIN:  MOV     SBUF, #"A"      ; Cần truyền ký tự "A"
HERE:   JNB     TI, HERE        ; Chờ đến bít cuối cùng
        CLR     TI              ; Xoá bít TI cho ký tự kế tiếp
        SJMP    AGAIN           ; Tiếp tục gửi lại chữ A
```

Ví dụ 10.3:

Hãy viết chương trình để truyền chữ "YES" nối tiếp liên tục với tốc độ 9600 baud (8 bít dữ liệu, 1 bít Stop).

Lời giải:

```
       MOV    TMOD, #20H     ; Chọn bộ Timer1, chế độ 2
       MOV    TH1, # - 3     ; Chọn tốc độ 9600 baud
       MOV    SCON, #50H     ; Truyền 8 bít dữ liệu, 1 bít Stop cho phép thu
       SETB   TR1            ; Khởi động Timer1
AGAIN: MOV    A, # "Y"       ; Truyền ký tự "Y"
       ACALL  TRANS
       MOV    A, # "E"       ; Truyền ký tự "E"
       ACALL  TRANS
       MOV    A, # "S"       ; Truyền ký tự "S"
       ACALL  TRANS
       SJMP   AGAIN          ; Tiếp tục
; Chương trình con truyền dữ liệu nối tiếp.
TRANS:  MOV   SBUF, A        ; Nạp SBUF
HERE:   JNB   TI, HERE       ; Chờ cho đến khi truyền bít cuối cùng
        CLR   TI             ; Chờ sẵn cho một byte kế tiếp
        RET
```

10.3.4.1 Tầm quan trọng của cờ TI.

Để hiểu tầm quan trọng của cờ ngắt TI ta hãy xét trình tự các bước dưới đây mà 8051 phải thực hiện khi truyền một ký tự quan đường TxD:

1. Byte ký tự cần phải truyền được ghi vào SBUF.
2. Truyền bít Start
3. Truyền ký tự 8 bít lần lượt từng bít một.
4. Bít Stop được truyền xong, trong quá trình truyền bít Stop thì cờ TI được bật (TI = 1) bởi 8051 để báo sẵn sàng để truyền ký tự kế tiếp.

5. Bằng việ hiển thị cờ TI ta biết chắc rằng ta không nạp quá vào thanh ghi SBUF. Nếu ta nạp một byte vào SBUF trước ghi TI được bật thì phần dữ liệu của byte trước chưa truyền hết sẽ bị mất. Hay nói cách khác là 8051 bật cờ TI khi đã truyền xong một byte và nó sẵn sàng để truyền byte kế tiếp.
6. Sau khi SBOF được nạp một byte mới tì cờ nhầm để có thể truyền byte mới này.

Từ phần trình bày trên đây ta kết luận rằng bằng việc kiểm tra bít cờ ngắt TI ta biết được 8051 có sẵn sàng để truyền một byte khác không. Quan trọng hơn cần phải nói ở đây là bít cờ TI được bật bởi từ 8051 khi nó hoàn tất việc truyền một byte dữ liệu, còn việc xoá nó thì phải được lập trình viên thực hiện bằng lệnh "CLR TI". Cũng cần lưu ý rằng, nếu ta ghi một byte vào thanh ghi SBUF trước khi cờ TI được bật thì sẽ có nguy cơ mất phần dữ liệu đang truyền. Bít cờ TI có thể kiểm tra bằng lệnh "JNB TI …" hoặc có thể sử dụng ngắt như ta sẽ thấy trong chương 11.

10.3.5 Lập trình 8051 để nhận dữ liệu.

Trong lập trình của 8051 để nhận các byte ký tự nối tiếp thì phải thực hiện các bước sau đây.

1. Nạp giá trị 20H vào thanh ghi TMOD để báo sử dụng bộ Timer1, chế độ 2 (8 bítm, tự động nạp lại) để thiết lập tốc độ baud.
2. Nạp TH1 các giá trị cho trong bảng 10.4 để tạo ra tốc độ baud với giả thiết XTAL = 10.0592MHz.
3. Nạp giá trị 50H vào thanh ghi SCON để báo sử dụng chế độ truyền nối tiếp 1 là dữ liệu được đóng gói bởi 8 bít dữ liệu, 1 bít Start và 1 bít Stop.
4. Bật TR1 = 1 để khởi động Timer1.
5. Xoá cờ ngắt RI bằng lệnh "CLR RI"
6. Bít cờ RI được hiển thị bằng lệnh "JNB RI, xx" để xem toàn bộ ký tự đã được nhận chưa.
7. Khi RI được thiết lập thì trong SBUF đã có 1 byte. Các nội dung của nó được cất lưu vào một nơi an toàn.
8. Để nhận một ký tự tiếp theo quay trở về bước 5.

Ví dụ 10.4:

Hãy lập trình cho 8051 để nhận các byte dữ liệu nối tiếp và đặt chúng vào cổng P1. Đặt tốc độ baud là 4800, 8 bít dữ liệu và 1 bít Stopl.

Lời giải:

```
        MOV    TMOD, #20H          ; Chọn bộ Timer1, chế độ 2 (tự động nạp lại)
        MOV    TH1, # - 6          ; Chọn tốc độ 4800 baud
        MOV    SCON, #50H          ; Chọn khung dữ liệu 8 bít Stop, bít.
        SETB   TR1                 ; Khởi động bộ Timer1
HERE:   JNB    R1, HERE            ; Đợi nhận toàn bộ lý tự vào hết
        MOV    A, SBUF             ; Lưu cất ký tự vào thanh A
        MOV    P1, A               ; Gửi ra cổng P.1
        CLR    RI                  ; Sẵn sàng nhận byte kế tiếp
        SJMP   HERE                ; Tiếp tục nhận dữ liệu
```

Ví dụ 10.5:

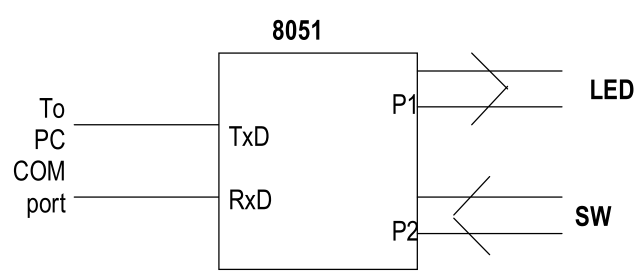
Giả sử cổng nối tiếp của 8051 được nối vào cổng COM của máy tính IBM CP và mà đang sử chương trình Termina. Exe để gửi và nhận dữ liệu nối tiếp. Cổng P1 và P2 của 8051 được nố tới các đầ LED và các công tắc chuyển mạch tương ứng. Hãy viết một chương trrình cho 8051.

a) Gửi thông báo "We Are Ready" (chúng tôi đã sẵn sàng) tới máy tính PC.

b) Nhận bất kỳ dữ liệu gì được PC gửi đến và chuyển đến các đèn LED đang nối đến các chân của cổng P1.
c) Nhận dữ liệu trên các chuyển mạch được nối tới P2 và gửi nó tới máy tính PC nối tiếp. Chương trình phải thực hiện một lần a), nhưng b) và c) chạy liên tục với tốc độ 4800 baud.

Lời giải:

```
                ORG     0
                MOV     P2, #0FFH              ; Lấy cổng P2 làm cổng vào
                MOV     TMOD, #20H             ; Chọn bộ Timer1, chế độ 2 (tự động nạp lại)
                MOV     TH1, # 0FAH            ; Chọn tốc độ 4800 baud
                MOV     SCON, #50H             ; Tạo khung dữ liệu 8 bít, 1bít Stop cho phép
                                                 REN.
                SETB    TR1                    ; Khởi động bộ Timer1
                MOV     DPTR, #MYDATA          ; Nạp con trỏ đến thông báo
 H - 1:         CLR     A
                MOVC  A, 'A + DPTR             ; Lấy ký tự
                JZ      DPTR                   ; Nếu ký tự cuối cùng muốn gửi ra
                ACALL   SEND                   ; Nếu chưa thì gọi chương trình con SEND
                INC     DPTR                   ; Chạy tiếp
                SJMP    H - 1                  ; Quay lại vòng lặp
B - 1:          MOV     A, P2                  ; Đọc dữ liệu trên cổng P2
                ACALL   RECV                   ; Truyền nó nối tiếp
                ACALL   RECV                   ; Nhận dữ liệu nối tiếp
                MOV     F1, A                  ; Hiển thị nó ra các đền LED
                SJMP    B - 1                  ; ở lại vòng lặp vô hạn
; — — — — — Truyền dữ liệu nối tiếp ACC có dữ liệu
SEND:           MOV     SBUF, A                ; Nạp dữ liệu
H- 2:           JNB     TI, H - 2              ; ở lại vòng lặp vô hạn
                CLR     TI                     ; Truyền dữ liệu nối tiếp
                RET                            ; Nhận dữ liệu
; — — — — — —Truyền dữ liệu nối tiếp ACC có dữ liệu
RECV:           JNB     RI, RECV               ; Nạp dữ liệu
                MOV     A, SBUF                ; ở lại đây cho đến khi gửi bít cuối cùng
                CLR     RI                     ; Sẵn sàng cho ký tự mới
                RET                            ; Trở về mời gọi
; — — — — — —Nhận dữ liệu nối tiếp trong ACC
RECV:           JNB     RI, RECV               ; Đợi ở đây nhận ký tự
                MOV     A, SBUF                ; Lưu nó vào trong ACC
                CLR     RI                     ; Sẵn sàng nhận ký tự mã tiếp theo
                RET                            ; Trở về nời gọi
; — — — — — — — Ngăn xếp chưa thông báo
MYDATA:         DB      "Chúng tôi đã sẵn sàng" 0
                END
```



10.3.5.1 Tầm quan trọng của cờ RT.

Khi nhận các bít quan chân RxD của nó thì 8051 phải đi quan các bước sau:

1. Nó nhận bít Start báo rằng bít sau nó là bít dữ liệu đầu tiên cần phải nhận.

2. Ký tự 8 bít được nhận lần lượt từng bít một. Khi bít cuối cùng được nhận thì một byte được hình thành và đặt vào trong SBUF.
3. Khi bít Stop được nhận thì 8051 bật RT = 1 để báo rằng toàn bộ ký tự được nhận và phải lấy đi trước khi nó bị byte mới nhận về ghi đè lên.
4. Bằng việc kiểm tra bít cờ RI khi nó được bật lên chúng ta biết rằng một ký tự đã được nhận và đang nằm trong SBUF. Tại sao nội dung SBUF vào nơi an toàn trong một thanh ghi hay bộ nhớ khác trước khi nó bị mất.
5. Sau khi SBUF được ghi vào nơi an toàn thì cờ RI được xoá về 0 bằng lệnh "CLR RI" nhằm cho các ký tự kế tiếp nhận được đưa vào SBUF. Nếu không làm được điều này thì gây ra mất ký tự vừa nhận được.

Từ mô tả trên đây ta rút ra kết luận rằng bằng việc kiểm tra cờ RI ta biết 8051 đã nhận được một byte ký tự chưa hay rồi. Nếu ta không sao được nội dung của thanh ghi SBUF vào nơi an toàn thì có nguy cơ ta bị mất ký tự vừa nhận được. Quan trọng hơn là phải nhớ rằng cờ RI được 8051 bật lên như lập trình viên phải xoá nó bằng lệnh "CLR RI". Cũng nên nhờ rằng, nếu ta sao nội dung SBUF vào nơi an toàn trước khi RI được bật ta mạo hiểm đã sao dữ liệu chưa đầy đủ. Bít cờ RI có thể được kiểm tra bởi lệnh "JNB RI, xx" hoặc bằng ngắt sẽ được bàn ở chương 11.

10.3.6 Nhân đôi tốc độ baud trong 8051.

Có hai cách để tăng tốc độ baud truyền dữ liệu trong 8051.

1. Sử dụng tần số thạch anh cao hơn.
2. Thay đổi một bít trong thanh ghi điều khiển công suất PCON (Power Control) như chỉ ra dưới đây.

D7							D0
SMOD	–	–	–	GF0	GF0	PD	IDL

Phương án một là không thực thi trong nhiều trường hợp vì tần số thạch anh của hệ thống là cố định. Quan trọng hơn là nó không khả thi vì tần số thạch anh mới không tương thích với tốc độ baud của các cổng COM nối tiếp của IBM PC. Do vậy, ta sẽ tập trung thăm dò phương án hai, có một cách nhân đôi tần số baud bằng phần mềm trong 8051 với tần số thạch anh không đổi. Điều này được thực hiện nhờ thanh ghi PCON, đây là thanh ghi 8 bít. Trong 8 bít này thì có một số bít không được dùng để điều khiển công suất của 8051. Bít dành cho truyền thông là D7, bít SMOD (chế độ nối tiếp - serial mode). Khi 8051 được bật nguồn thì bít SMOD của thanh ghi PCON ở mức thấp 0. Chúng ta có thể đặt nó lên 1 bằng phần mềm và do vậy nhân đôi được tốc độ baud. Thứ tự các lệnh được sử dụng để thiết lập bít D7 của PCON lên cao như sau (thanh ghi PCON là thể đánh địa chỉ theo bít).

```
MOV   A, PCON         ; Đặt bản sao của PCON vào ACC
SETB  ACC.7           ; Đặt D7 của ACC lên 1.
MOV   PCON, A         ; Bây giờ SMOD = 1 mà không thay đổi bất kỳ bít nào khác.
```

Để biết tốc độ baud được tăng lên gấp đôi như thế nào bằng phương pháp này ta xét vai trò của bít SMOD trong PCON khi nó là 0 và 1.

a) Khi SMOD = 0.

Khi SMOD = 0 thì 8051 chia 1/12 tần số thạch anh cho 32 và sử dụng nó cho bộ Timer1 để thiết lập tốc độ baud. Trong trường hợp XTAL = 11.0592MHz thì ta có:

Tần số chu trình máy $= \frac{11.0592\text{MHz}}{12} = 921.6\text{kHz}$ và $\frac{921.6\text{kHz}}{32} = 28.800\text{Hz}$ vì SMOD = 0.

Đây là tần số được Timer1 sử dụng để đặt tốc độ baud. Đây là cơ sở cho tất cả ví dụ từ trước đến giờ vì nó là giá trị mặc định của 8051 khi bật nguồn. Các tốc độ baud đối với SMOD = 0 được cho trong bảng 10.4.

b) Khi SMOD = 1.

Với tần số cố định thạch anh ta có thể nhân đôi tốc độ baud bằng cách đặt bít SMOD = 1. Khi bít D7 của PCON (bít SMOD) được đưa lên 1 thì 1/12 tần số XTAL được chia cho 16 (thay vì chia cho 32 như khi SMOD = 0) và đây là tần số được Timer dùng để thiết lập tốc độ baud. Trong trường hợp XTAL = 11.0592MHz ta có:

Tần số chu trình máy $= \frac{11.0592\text{MHz}}{12} = 921.6\text{kHz}$ và $\frac{921.6\text{kHz}}{16} = 57.600\text{kHz}$ vì SMOD = 1.

Đây là tần số mà Timer1 dùng để đặt tốc độ baud. Bảng 10.5 là các giá trị cần được nạp vào TH1 cùng với các tốc độ baud của 8051 khi SMOD = 0 và 1.

Bảng 10.5: So sánh tốc độ baud khi SMOD thay đổi.

TH1 (thập phân)	TH1 (Hex)	Tốc độ baud	
		SMOD = 0	SMOD = 1
-3	FD	9600	19200
-6	DA	4800	9600
-12	F4	2400	4800
-24	E8	1200	2400

Ví dụ 10.6:

Giả sử tần số XTAL = 11.0592MHz cho chương trình dưới đây, hãy phát biểu a) chương trình này làm gì? b) hãy tính toán tần số được Timer1 sử dụng để đặt tốc độ baud? và c) hãy tìm tốc độ baud truyền dữ liệu.

```
        MOV     A, PCON         ; Sao nội dung thanh ghi PCON vào thanh ghi ACC
        SETB    ACC.7           ; Đặt D7 = 0
        MOV     PCON, A         ; Đặt SMOD = 1 để tăng gấp đôi tần số baud với tần số XTAL cố định
                                ;
        MOV     TMOD, #20H      ; Chọn bộ Timer1, chế độ 2, tự động nạp lại
        MOV     TH1, - 3        ; Chọn tốc độ baud 19200 (57600/3=19200) vì SMOD = 1
                                ;
        MOV     SCON, #50H      ; Đóng khung dữ liệu gồm 8 bít dữ liệu, 1 Stop và cho phép RI.
        SETB    TR1             ; Khởi động Timer1
        MOV     A, #"B"         ; Truyền ký tự B
A-1:    CLR     TI              ; Khẳng định TI = 0
        MOV     SBUF, A         ; Truyền nó
H-1:    JNB     TI, H-1         ; Chờ ở đây cho đến khi bít cuối được gửi đi
        SJMP    A-1             ; Tiếp tục gửi "B"
```

Lời giải:

a) Chương trình này truyền liên tục mã ASCII của chữ B (ở dạng nhị phân là 0100 0010)

b) Với tần số XTAL = 11.0592MHz và SMOD = 1 trong chương trình trên ta có:

11.0592MHz/12 = 921.6kHz là tần số chu trình máy

921.6kHz/16 = 57.6kHz là tần số được Timer1 sử dụng để đặt tốc độ baud

c) 57.6kHz/3 = 19.200 là tốc độ cần tìm

Ví dụ 10.7:

Tìm giá trị TH1 (ở dạng thập phân và hex) để đặt tốc độ baud cho các trường hợp sau.

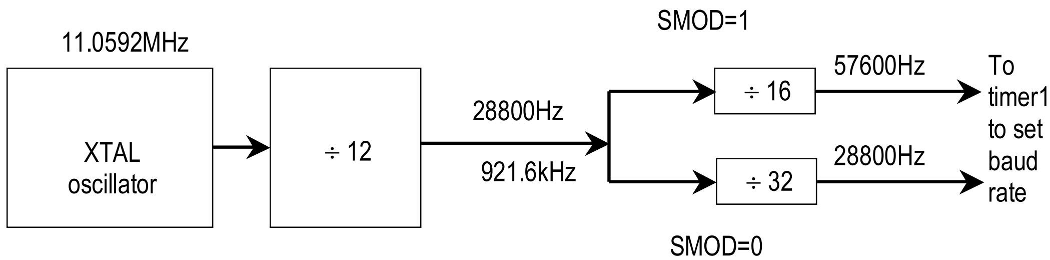
a) 9600 b) 4800 nếu SMOD = 1 và tần số XTAL = 11.0592MHz

Lời giải:

Với tần số XTAL = 11.0592MHz và SMOD = 1 ta có tần số cấp cho Timer1 là 57.6kHz.

a) 57.600/9600 = 6 do vậy TH1 = - 6 hay TH1 = FAH

b) 57.600/4800 = 12 do vậy TH1 = - 12 hay TH1 = F4H



Ví dụ 10.8:

Hãy tìm tốc độ baud nếu TH1 = -2, SMOD = 1 và tần số XTAL = 11.0592MHz. Tốc độ này có được hỗ trợ bởi các máy tính IBM PC và tương thích không?

Lời giải:

Với tần số XTAL = 11.0592MHz và SMOD = 1 ta có tần số cấp cho Timer1 là 57.6kHz. Tốc độ baud là 57.600kHz/2 = 28.800. Tốc độ này không được hỗ trợ bởi các máy tính IBM PC và tương thích. Tuy nhiên, PC có thể được lập trình để truyền dữ liệu với tốc độ như vậy. Phần mềm của nhiều modem có thể làm cho điều này và Hyperterminal của Windows 95 cũng có thể hỗ trợ tốc độ này và các tốc độ khác nữa.

10.3.7 Truyền dữ liệudựa trên các ngắt.

Ta phải thấy rằng thật lãng phí để các bộ vi điều khiển phải bật lên xuống các cờ TI và RI. Do vậy, để tăng hiệu suất của 8051 ta có thể lập trình các cổng truyền thông nối tiếp của nó bằng các ngắt. Đây chính là nội dung chính sẽ bàn luận ở chương 11 dưới đây.