

Khái niệm về vào ra nối tiếp

TS Nguyễn Hồng Quang

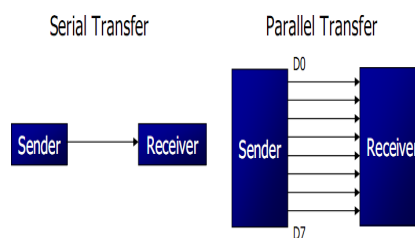


Electrical Engineering

1

Cơ bản về truyền thông

- Hai phương pháp truyền dữ liệu cơ bản
 - Truyền song song với 8 or 16 đường dây vật lý dùng truyền dữ liệu
 - Truyền nối tiếp với dữ liệu được chuyển thành từng bit và mỗi bit được truyền qua 2 dây
- Các phương án truyền nối tiếp cơ bản
 - RS232, RS422, RS485, CAN, Ethernet, SPI, I2C, USB



Electrical Engineering

2

Truyền dữ liệu song song tồn tại không ?

- Bus địa chỉ, dữ liệu
- VME Bus
- IEEE 488 (GPIB, HPIB)
- Parallel port
- HPIB phát triển bởi HP
- IEEE 488.1 định nghĩa về phần cứng và cách nối dây điện
- IEEE 488.2 định nghĩa về chuẩn truyền

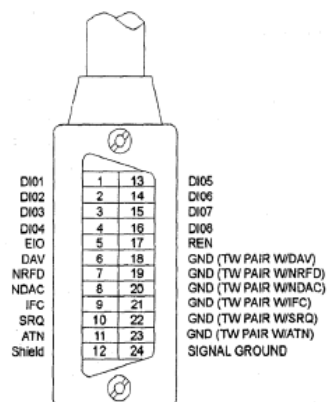


Electrical Engineering

3

Ví dụ

- 8 đường truyền dữ liệu
- 8 đường điều khiển
- BUS được điều khiển bởi chip GPIB chuyên dụng



Electrical Engineering

4

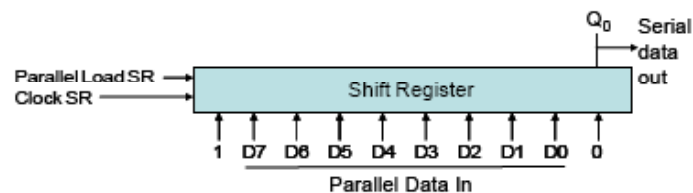
Các tham số chính trong truyền nối tiếp

- Phần cứng (vật lý)
 - Loại đầu nối (connector)
 - Mức điện áp
 - Đặc tính thời gian và phương pháp bắt tay
- Phần mềm (software protocol)
 - Phương án mã hóa và giải mã dữ liệu
 - Phương án kiểm tra lỗi đường truyền



Khái niệm cơ bản

- Tại đầu truyền, một byte sẽ chuyển thành chuỗi bit nối tiếp
- Tại đầu nhận, một chuỗi bit nối tiếp sẽ chuyển thành byte
- Tùy theo khoảng cách truyền mà cho phép có thêm các tín hiệu hỗ trợ ở giữa như repeater, modem.



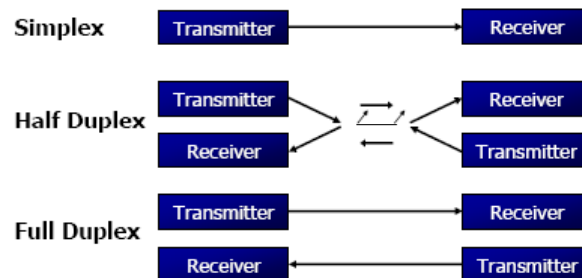
Cách thức truyền

- Truyền nối tiếp sử dụng 2 phương pháp
 - Truyền đồng bộ cho phép gửi một mảng dữ liệu vào cùng một thời điểm
 - Truyền không đồng bộ yêu cầu truyền từng byte trong một thời điểm
- Việc truyền dữ liệu này được thực hiện bởi IC chuyên dụng với 2 loại
 - UART (universal asynchronous Receiver transmitter)
 - USART (universal synchronous asynchronous Receiver-transmitter), thường dùng truyền ký tự



Tên phương thức truyền

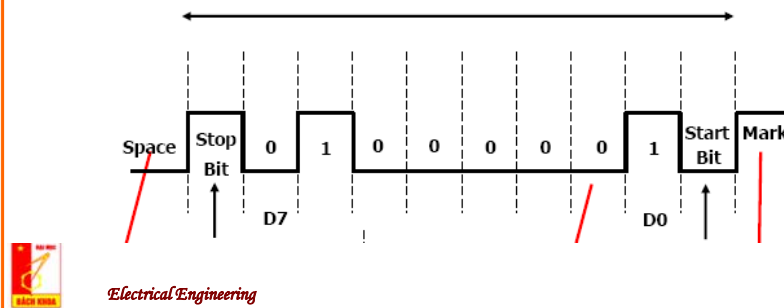
- Truyền song công
- Truyền bán song công
- Truyền đơn công



Ví dụ

- Start bit dùng bit 0
- Stop bit dùng bit 1
- Tín hiệu 1 (Mark) báo dừng truyền

ASCII character "A" (8-bit binary 0100 0001)



9

Khung truyền (frame)

- Start bit
- 8 bit dữ liệu (255 ký tự ASCII), hoặc 7bit
- Stop bit (1,2 stop bit)
- Bit chẵn lẻ cho kiểm tra



Electrical Engineering

10

Tốc độ truyền

- Được tính theo baud rate
 - 1200, 2400, 4800, 9600, 19200, 38400, 57600
- 115200 bit/sec thường dùng cho truyền theo chuẩn RS485
- Tốc độ càng cao thì dây truyền càng ngắn



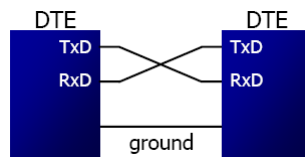
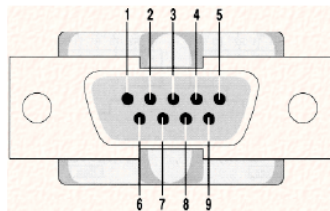
Chuẩn RS232

- Được đưa ra năm 1966 do EIA, và ứng dụng trong truyền thông tại công ty Bell Technology
- Cho phép truyền dữ liệu giữa máy tính và các Terminal DTE, hoặc các modem DCE
- Ứng dụng chủ yếu trong truyền dữ liệu trong khoảng cách gần ($<1.2\text{m}$)



Phần cứng RS232

RS232 Connector DB-9



RS232 DB-9 Pins

Pin	Description
1	Data carrier detect (-DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (-DSR)
7	Request to send (-RTS)
8	Clear to send (-CTS)
9	Ring indicator (RI)



Electrical Engineering

13

Giải thích các chân

- DTR (data terminal ready)
 - Đặt tính hiệu này lên 1 khi Terminal sẵn sàng
- DSR (data set ready)
 - Sẵn sàng truyền thông khi modem bắt đầu
- RTS (request to send)
 - Khi DTE có dữ liệu để truyền, RTS được đặt lên 1
- CTS (clear to send)
 - Có vùng trống để bắt đầu nhận dữ liệu

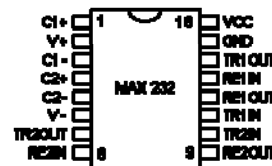
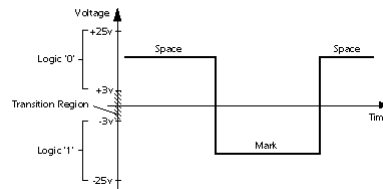


Electrical Engineering

14

Nối ghép giữa PC và 8051

- Không tương thích với chuẩn TTL
 - ✓ $3 \div 25$ V ứng với logic 0
 - ✓ $-3 \div -25$ V ứng với logic 1
- Sử dụng chuyển đổi mức chuyên dụng MAX 232, 233, MC 1488, 1489

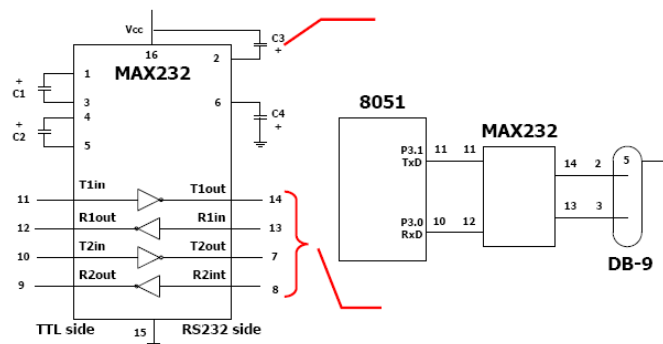


Electrical Engineering

15

Chuyển mức logic với 8051

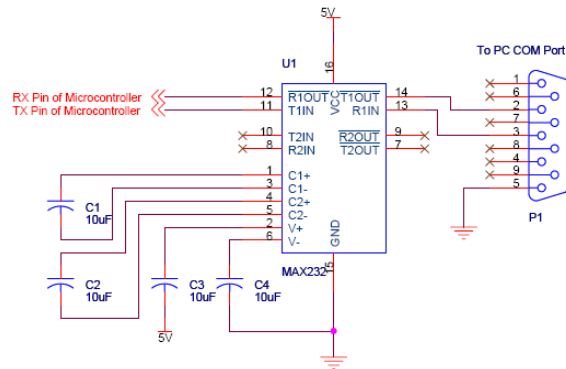
Tương ứng với chân P3.0, P3.1 là chân TxD and RxD trong truyền dữ liệu nối tiếp



Electrical Engineering

16

Ví dụ kết nối cổng RS232 (COM) của máy tính



Electrical Engineering

17

Thanh ghi SCON

Bit	Name	Bit Address	Giải thích	
7	SM0	9Fh	Set mode	
6	SM1	9Eh	Set mode	
5	SM2	9Dh	Cho phép truyền thông nhiều VXL	
4	REN	9Ch	Receiver Enable. Bit này lên 1 cho phép nhận ký tự	
3	TB8	9Bh	Transmit bit 8. Chỉ dùng mode 2 and 3.	
2	RB8	9Ah	Receive bit 8. Chỉ dùng mode 2 and 3.	
1	TI	99h	Transmit Flag. Lên 1 khi kết thúc truyền byte	
0	RI	98h	Receive Flag. Lên 1 khi kết thúc nhận byte	
SM0	SM1	Serial Mode	Ghi chú	Baud Rate
0	0	0	8-bit Shift Register	Oscillator / 12
0	1	1	8-bit UART	Set by Timer 1 (*)
1	0	2	9-bit UART	Oscillator / 64 (*)
1	1	3	9-bit UART	Set by Timer 1 (*)



Electrical Engineering

18

Giải thích thêm

- The **TB8** bit dùng trong mode 2 và 3 và truyền tổng cộng 9 bit. Bao gồm 8 bit dữ liệu và 1 bit thêm vào theo giá trị TB8.
- The **RB8** tương tự TB8, nhận 8 bit dữ liệu và bit thứ 9 trong RB8.



TI và RI

- TI (transmit interrupt)
 - Cờ TI đặt lên khi kết thúc quá trình truyền 1 byte và sẵn sàng cho truyền byte tiếp theo
- RI (receive interrupt)
 - Cờ RI đặt lên khi kết thúc quá trình nhận 1 byte và đặt dữ liệu trong thanh ghi SBUF (sau khi loại bit start và stop)



Tốc độ truyền

- Đơn vị là baud, 1 baud tương ứng với tốc độ 1 bit truyền trong 1 giây
- Tốc độ này do lịch sử để lại với giá trị
- Tốc độ của máy tính PC hiện nay có thể đạt tới 115.000 baud cho truyền RS232 và lên tới xấp xỉ 1M baud cho truyền 485

PC Baud Rates

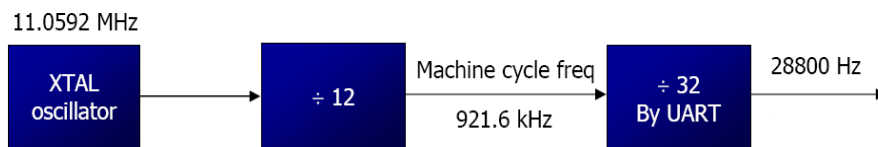
110
150
300
600
1200
2400
4800
9600
19200



Electrical Engineering

21

Đặt tốc độ truyền với 8051 với Timer 1



- $28,800 / 1 = 28800$ (hex) -1 = FE
- $28,800 / 3 = 9600$ (hex) -3 = FD
- $28,800 / 12 = 2400$ (hex) -12 = F4
- $28,800 / 24 = 1200$ -24 = E8h

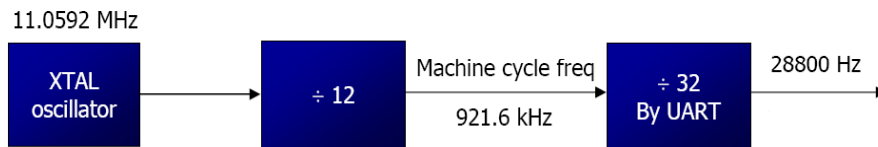
$$TH1 = 256 - \left\{ \frac{K * \text{Oscillator Frequency}}{32 * 12 * \text{Baud Rate}} \right\}$$



Electrical Engineering

22

Đặt tốc độ truyền với 8051 với Timer 2



$$\text{Baud Rate} = \left\{ \frac{\text{Oscillator Frequency}}{32 * [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \right\}$$

$$\text{RCAP2H}, \text{RCAP2L} = 65536 - \left\{ \frac{\text{Oscillator Frequency}}{32 * \text{Baud Rate}} \right\}$$



Thanh ghi SBUF

- SBUF là thanh ghi 8 bit chỉ dùng trong truyền thông nối tiếp.
 - Khi truyền dữ liệu qua dùng TxD, dữ liệu đặt trong SBUF
 - Khi dữ liệu nhận được ở RxD, phần cứng sẽ loại bỏ bit Start và Stop, rồi đặt trong SBUF
- MOV SBUF, #'D'
- MOV SBUF, A
- MOV A, SBUF



Ví dụ truyền dữ liệu

```
MOV TMOD,#20H ;timer 1,mode 2(auto reload)
MOV TH1,#-6      ;4800 baud rate
MOV SCON,#50H ;8-bit, 1 stop, REN enabled
SETB TR1 ;start timer 1
AGAIN:
MOV SBUF,#"A"
HERE: JNB TI,HERE
CLR TI
SJMP AGAIN
```



Ví dụ nhận dữ liệu

```
MOV TMOD,#20H ;timer 1,mode 2(auto reload)
MOV TH1,#-6 ;4800 baud rate
MOV SCON,#50H ;8-bit, 1 stop, REN enabled
SETB TR1 ;start timer 1
HERE: JNB RI,HERE
MOV A,SBUF
MOV P1,A
CLR RI
SJMP HERE
```



Ví dụ truyền và nhận sử dụng polling

```

org      00h
; Set up Timer 1 to act as clock for Serial Port

mov      tmod,#20h ; Set Timer1 to Mode 2
mov      th1,#0fdh ; Reload value
;=====
; SETUP SERIAL PORT

mov      scon,#50h ; Serial Mode
mov      tcon,#40h ; Start Timer
;=====
; Save char to memory - as a Log
mov      r0,#$60 ; Serial Buffer ptr
;=====
; Setup Done, Main Program starts here

lup:     acall    getch ; Get char from keybd
mov      @r0,a ; put into Log buffer
inc      r0 ; inc log pointer
clr      c
subb     a,#32 ; convert to upper case
acall    putch ; Display Character
sjmp     lup ; do again
;=====
; GETCH - get a character
getch:   jnb     ri,getch ; Character arrived?
mov      a,sbuf ; Get it
clr      ri ; Clear Ch Ready flag
ret      ; return char in A
;=====
; PUTCH - Output a character

putch:   clr      ti ; Reset TI bit
mov      sbuf,a ; Store ch in A to Output
gone:    jnb     ti,gone ; Wait until it's gone
ret      ; return

```



Electrical Engineering

27

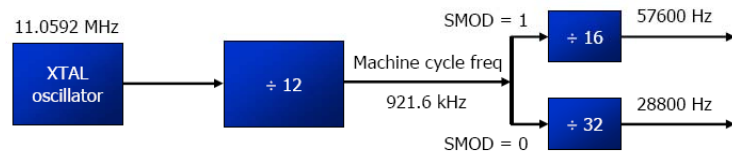
Tăng gấp đôi tốc độ truyền

SMOD	--	--	--	GF1	GF0	PD	IDL
------	----	----	----	-----	-----	----	-----

```

{ MOV  A,PCON ;place a copy of PCON in ACC
  SETB ACC.7 ;make D7=1
  MOV  PCON,A ;changing any other bits

```



Electrical Engineering

28

Ngắt với cổng nối tiếp

- Với 8051, ngắt sẽ được tạo ra với cả
 - Nhận và truyền dữ liệu
 - Bit 4, (IE.4) trong thanh ghi IE được đặt lên
 - PC nhảy về địa chỉ 0023H
- Phần mềm phải kiểm tra đây là ngắt do RI hay TI gây ra và viết chương trình tương ứng



Ví dụ

```
ORG 0000H
LJMP MAIN
ORG 23H
LJMP SERIAL ;jump to serial int ISR
ORG 30H
MAIN: MOV P1,#0FFH ;make P1 an input port
      MOV TMOD,#20H ;timer 1, auto reload
      MOV TH1,#0FDH ;9600 baud rate
      MOV SCON,#50H ;8-bit, 1 stop, ren enabled
      MOV IE,10010000B ;enable serial int.
      SETB TR1 ;start timer 1
BACK: MOV A,P1 ;read data from port 1
      MOV SBUF,A ;give a copy to SBUF
      MOV P2,A ;send it to P2
      SJMP BACK ;stay in loop indefinitely
ORG 100H
SERIAL: JB TI,TRANS;jump if TI is high
        MOV A,SBUF ;otherwise due to receive
        CLR RI ;clear RI since CPU doesn't
        RETI ;return from ISR
TRANS: CLR TI ;clear TI since CPU doesn't
        RETI ;return from ISR
```



Bảng tốc độ truyền

TH1		7.3728 MHz	8.00 MHz	11.0592 MHz	11.0592 MHz	12.00 MHz	12.00 MHz	14.7456 MHz	22.1184 MHz
				smod=0	smod=1	smod=0	smod=1		smod=1
E0		600	651	900		976		1,200	
E6	-26					1,202			
E8	-24			1,200	2,400				4,800
F0	-16	1,200	1,302	1,800		1,953		2,400	
F3	-13					2,404			
F4	-12			2,400	4,800				9,600
F8	-8	2,400	2,604	3,600		3,906		4,800	
F9	-7	2,743	2,976		8,299	4,464	8,923	5,486	
FA	-6	3,200	3,472	4,800	9,600	5,208		6,400	19,200
FD	-3			9,600	19,200				38,400
FF	-1	19,200	20,833	28,800	57.6K		62,500		115.2K

