

# Cấu trúc cổng nối tiếp

Cổng nối tiếp được sử dụng để truyền dữ liệu hai chiều giữa máy tính và ngoại vi, có các ưu điểm sau:

- Khoảng cách truyền xa hơn truyền song song.
- Số dây kết nối ít.
- Có thể truyền không dây dùng hồng ngoại.
- Có thể ghép nối với vi điều khiển hay PLC (Programmable Logic Device).
- Cho phép nối mạng.
- Có thể tháo lắp thiết bị trong lúc máy tính đang làm việc.
- Có thể cung cấp nguồn cho các mạch điện đơn giản

Các thiết bị ghép nối chia thành 2 loại: DTE (Data Terminal Equipment) và DCE (Data Communication Equipment). DCE là các thiết bị trung gian như MODEM còn DTE là các thiết bị tiếp nhận hay truyền dữ liệu như máy tính, PLC, vi điều khiển, ... Việc trao đổi tín hiệu thông thường qua 2 chân RxD (nhận) và TxD (truyền). Các tín hiệu còn lại có chức năng hỗ trợ để thiết lập và điều khiển quá trình truyền, được gọi là các tín hiệu bắt tay (handshake). Ưu điểm của quá trình truyền dùng tín hiệu bắt tay là có thể kiểm soát đường truyền.

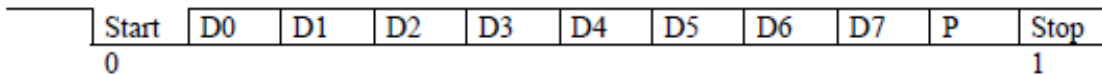
Tín hiệu truyền theo chuẩn RS-232 của EIA (Electronics Industry Associations). Chuẩn RS-232 quy định mức logic 1 ứng với điện áp từ -3V đến -25V (mark), mức logic 0 ứng với điện áp từ 3V đến 25V (space) và có khả năng cung cấp dòng từ 10 mA đến 20 mA. Ngoài ra, tất cả các ngõ ra đều có đặc tính chống chập mạch.

Chuẩn RS-232 cho phép truyền tín hiệu với tốc độ đến 20.000 bps nhưng nếu cáp truyền đủ ngắn có thể lên đến 115.200 bps.

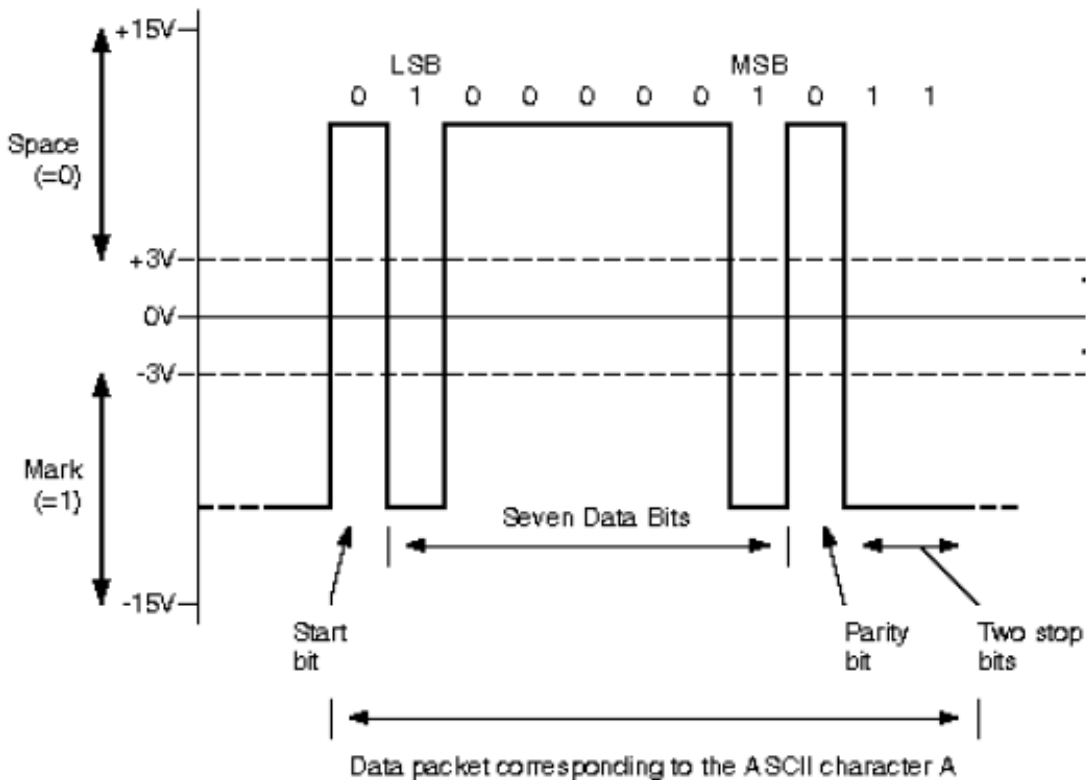
Các phương thức nối giữa DTE và DCE:

- Đơn công (simplex connection): dữ liệu chỉ được truyền theo 1 hướng.
- Bán song công (half-duplex): dữ liệu truyền theo 2 hướng, nhưng mỗi thời điểm chỉ được truyền theo 1 hướng.

- Song công (full-duplex): số liệu được truyền đồng thời theo 2 hướng. Định dạng của khung truyền dữ liệu theo chuẩn RS-232 như sau:



Khi không truyền dữ liệu, đường truyền sẽ ở trạng thái mark (điện áp -10V). Khi bắt đầu truyền, DTE sẽ đưa ra xung Start (space: 10V) và sau đó lần lượt truyền từ D0 đến D7 và Parity, cuối cùng là xung Stop (mark: -10V) để khôi phục trạng thái đường truyền. Dạng tín hiệu truyền mô tả như sau (truyền ký tự A):

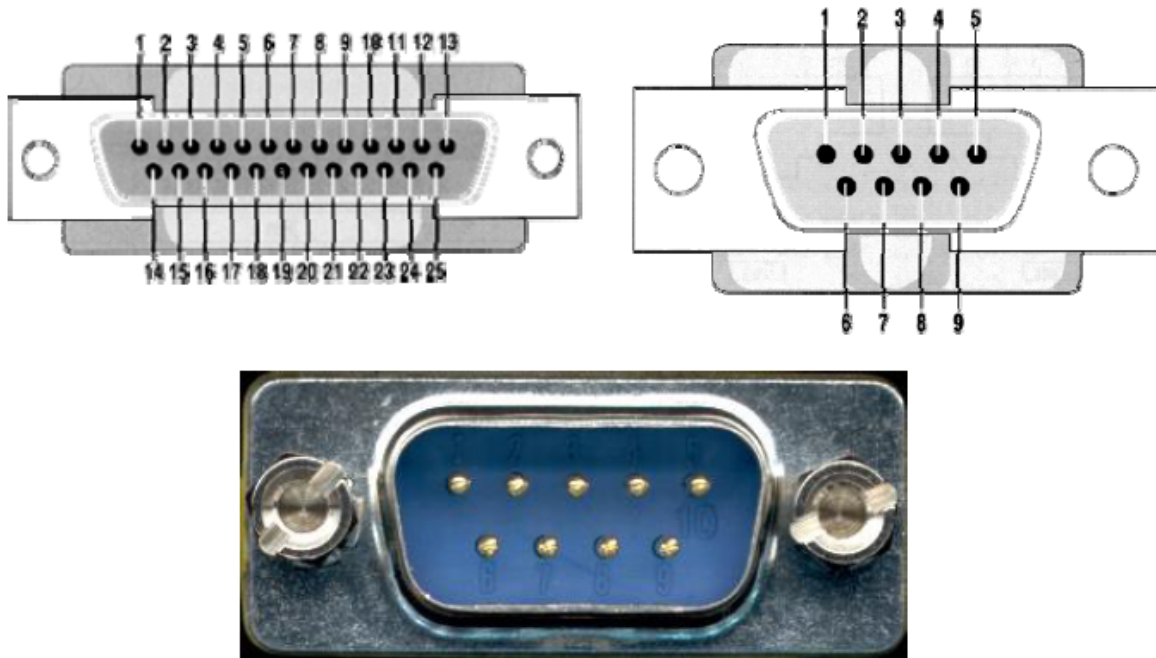


*Tín hiệu truyền của ký tự 'A'*

Chiều dài cable cực đại	15m
Tốc độ dữ liệu cực đại	20 Kbps
Điện áp ngõ ra cực đại	± 25V
Điện áp ngõ ra có tải	± 5V đến ± 15V
Trở kháng tải	3K đến 7K
Điện áp ngõ vào	± 15V
Độ nhạy ngõ vào	± 3V
Trở kháng ngõ vào	3K đến 7K

Các tốc độ truyền dữ liệu thông dụng trong cổng nối tiếp là: 1200 bps, 4800 bps, 9600 bps và 19200 bps.

- **Sơ đồ chân:**



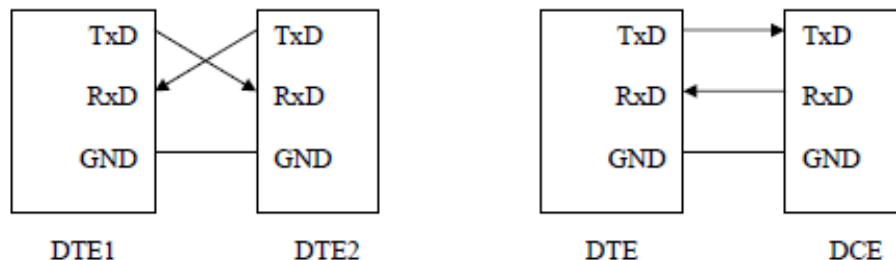
*Sơ đồ chân cổng nối tiếp*

Cổng COM có hai dạng: đầu nối DB25 (25 chân) và đầu nối DB9 (9 chân) mô tả như hình 2. Ý nghĩa của các chân mô tả như sau:

D25	D9	Tín hiệu	Hướng truyền	Mô tả
1	-	-	-	Protected ground: nối đất bảo vệ
2	3	TxD	DTE→DCE	Transmitted data: dữ liệu truyền
3	2	RxD	DCE→DTE	Received data: dữ liệu nhận
4	7	RTS	DTE→DCE	Request to send: DTE yêu cầu truyền dữ liệu
5	8	CTS	DCE→DTE	Clear to send: DCE sẵn sàng nhận dữ liệu
6	6	DSR	DCE→DTE	Data set ready: DCE sẵn sàng làm việc
7	5	GND	-	Ground: nối đất (0V)
8	1	DCD	DCE→DTE	Data carrier detect: DCE phát hiện sóng mang
20	4	DTR	DTE→DCE	Data terminal ready: DTE sẵn sàng làm việc
22	9	RI	DCE→DTE	Ring indicator: báo chuông
23	-	DSRD	DCE→DTE	Data signal rate detector: dò tốc độ truyền
24	-	TSET	DTE→DCE	Transmit Signal Element Timing: tín hiệu định thời truyền đi từ DTE
15	-	TSET	DCE→DTE	Transmitter Signal Element Timing: tín hiệu định thời truyền từ DCE để truyền dữ liệu
17	-	RSET	DCE→DTE	Receiver Signal Element Timing: tín hiệu định thời truyền từ DCE để truyền dữ liệu
18	-	LL		Local Loopback: kiểm tra công
21	-	RL	DCE→DTE	Remote Loopback: Tạo ra bởi DCE khi tín hiệu nhận từ DCE lỗi
14	-	STxD	DTE→DCE	Secondary Transmitted Data
16	-	SRxD	DCE→DTE	Secondary Received Data
19	-	SRTS	DTE→DCE	Secondary Request To Send
13	-	SCTS	DCE→DTE	Secondary Clear To Send
12	-	SDSRD	DCE→DTE	Secondary Received Line Signal Detector
25	-	TM		Test Mode
9	-			Dành riêng cho chế độ test
10	-			Dành riêng cho chế độ test
11				Không dùng

# Truyền thông giữa hai nút

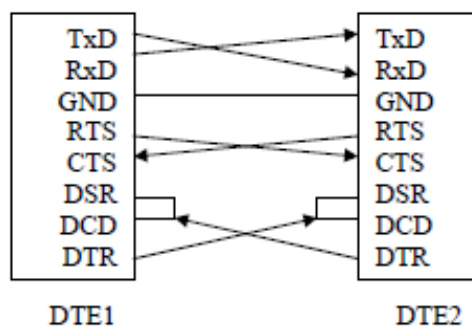
Các sơ đồ khi kết nối dùng công nối tiếp:



*Kết nối đơn giản trong truyền thông nối tiếp*

Khi thực hiện kết nối như trên, quá trình truyền phải bảo đảm tốc độ ở đầu phát và thu giống nhau. Khi có dữ liệu đến DTE, dữ liệu này sẽ được đưa vào bộ đệm và tạo ngắt.

Ngoài ra, khi thực hiện kết nối giữa hai DTE, ta còn dùng sơ đồ sau:



*Kết nối trong truyền thông nối tiếp dùng tín hiệu bắt tay*

Khi DTE1 cần truyền dữ liệu thì cho DTR tích cực → tác động lên DSR của DTE2 cho biết sẵn sàng nhận dữ liệu và cho biết đã nhận được sóng mang của MODEM (ảo). Sau đó, DTE1 tích cực chân RTS để tác động đến chân CTS của DTE2 cho biết DTE1 có thể nhận dữ liệu. Khi thực hiện kết nối giữa DTE và DCE, do tốc độ truyền khác nhau nên phải thực hiện điều khiển lưu lượng. Quá trình điều khiển này có thể thực hiện bằng phần mềm hay phần cứng. Quá trình điều khiển bằng phần mềm thực hiện bằng hai ký tự Xon và Xoff. Ký tự Xon được DCE gửi đi khi rảnh (có thể nhận dữ liệu). Nếu DCE bận thì sẽ gửi ký tự Xoff. Quá trình điều khiển bằng phần cứng dùng hai chân RTS và CTS. Nếu DTE muốn truyền dữ liệu thì sẽ gửi RTS để yêu cầu truyền, DCE nếu có khả năng nhận dữ liệu (đang rảnh) thì gửi lại CTS.

# Truy xuất trực tiếp thông qua cổng

Các cổng nối tiếp trong máy tính được đánh số là COM1, COM2, COM3, COM4 với các địa chỉ như sau:

Tên	Địa chỉ	Ngắt	Vị trí chứa địa chỉ
COM1	3F8h	4	0000h:0400h
COM2	2F8h	3	0000h:0402h
COM3	3E8h	4	0000h:0404h
COM4	2E8h	3	0000h:0406h

Giao tiếp nối tiếp trong máy tính sử dụng vi mạch UART với các thanh ghi cho trong bảng sau:

Offset	DLAB	R/W	Tên	Chức năng
0	0	W	THR	Transmitter Holding Register (đệm truyền)
	0	R	RBR	Receiver Buffer Register (đệm thu)
	1	R/W	BRDL	Baud Rate Divisor Latch (số chia byte thấp)
1	0	R/W	IER	Interrupt Enable Register (cho phép ngắt)
	1	R/W	BRDH	Số chia byte cao
2		R	IIR	Interrupt Identification Register (nhận dạng ngắt)
		W	FCR	FIFO Control Register
3		R/W	LCR	Line Control Register (điều khiển đường dây)
4		R/W	MCR	Modem Control Register (điều khiển MODEM)
5		R	LSR	Line Status Register (trạng thái đường dây)
6		R	MSR	Modem Status Register (trạng thái MODEM)
7		R/W		Scratch Register (thanh ghi tạm)

Các thanh ghi này có thể truy xuất trực tiếp kết hợp với địa chỉ cổng (ví dụ như thanh ghi cho phép ngắt của COM1 có địa chỉ là  $BACOM1 + 1 = 3F9h$ ).

- **IIR (Interrupt Identification):**

IIR xác định mức ưu tiên và nguồn gốc của yêu cầu ngắt mà UART đang chờ phục vụ. Khi cần xử lý ngắt, CPU thực hiện đọc các bit tương ứng để xác định nguồn gốc của ngắt. Định dạng của IIR như sau:

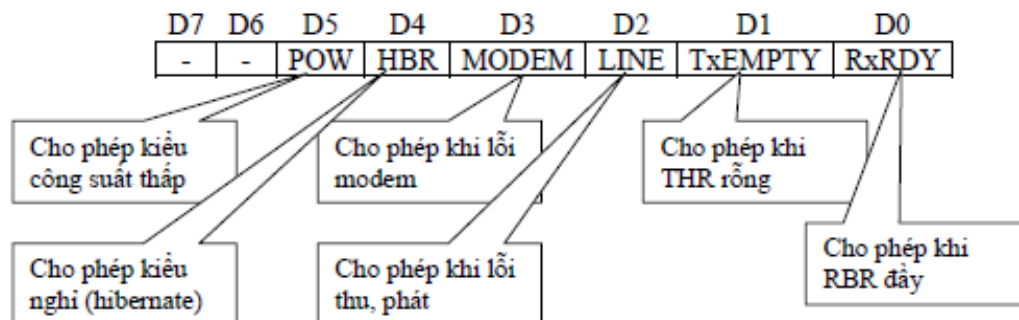
D7	D6	D5	D4	D3	D2	D1	D0
00: không có FIFO 11: cho phép FIFO	Cho phép FIFO 64 byte (trong 16750)		-	1: ngắt time-out (trong 16550)	Xác định nguồn gốc ngắt		0: có ngắt 1: không ngắt

D2	D1	Ưu tiên	Tên	Nguồn	D2 - D0 bị xoá khi
0	0	4	Đường truyền	Lỗi khung, thu đề, lỗi parity, gián đoạn khi thu	Đọc LSR
0	1	3	Đệm thu	Đệm thu đầy	Đọc RBR
1	0	2	Đệm phát	Đệm phát rỗng	Đọc IIR, ghi THR
1	1	1	Modem	CTS, DSR, RI, RLSD	Đọc MSR

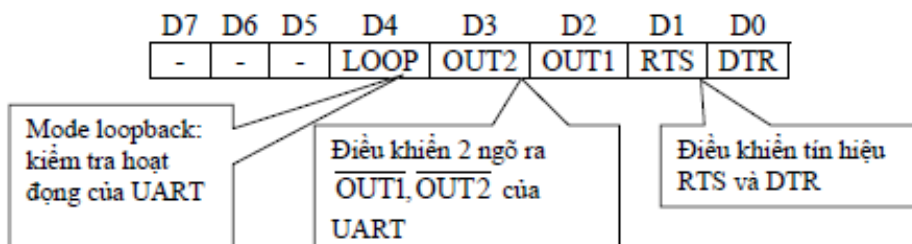
(mức 1 ưu tiên cao nhất)

- **IER (Interrupt Enable Register):**

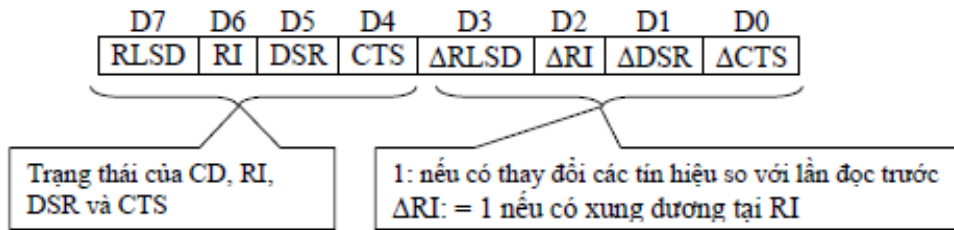
IER cho phép hay cấm các nguyên nhân ngắt khác nhau (1: cho phép, 0: cấm ngắt)



- **MCR (Modem Control Register):**



- **MSR (Modem Status Register):**



• **LSR (Line Status Register):**

D7	D6	D5	D4	D3	D2	D1	D0
FIE	TSRE	THRE	BI	FE	PE	OE	RxD

FIE: FIFO Error - sai trong FIFO

TSRE: Transmitter Shift Register Empty - thanh ghi dịch rỗng (=1 khi đã phát 1 ký tự và bị xoá khi có 1 ký tự chuyển đến từ THR).

THRE: Transmitter Holding Register Empty (=1 khi có 1 ký tự đã chuyển từ THR -TSR và bị xoá khi CPU đưa ký tự tới THR).

BI: Break Interrupt (=1 khi có sự gián đoạn khi truyền, nghĩa là tồn tại mức logic 0 trong khoảng thời gian dài hơn khoảng thời gian truyền 1 byte và bị xoá khi CPU đọc LSR)

FE: Frame Error (=1 khi có lỗi khung truyền và bị xoá khi CPU đọc LSR) PE: Parity Error (=1 khi có lỗi parity và bị xoá khi CPU đọc LSR)

OE: Overrun Error (=1 khi có lỗi thu đề, nghĩa là CPU không đọc kịp dữ liệu làm cho quá trình ghi chồng lên RBR xảy ra và bị xoá khi CPU đọc LSR)

RxD: Receiver Data Ready (=1 khi đã nhận 1 ký tự và đưa vào RBR và bị xoá khi CPU đọc RBR).

• **LCR (Line Control Register):**

D7	D6	D5	D4	D3	D2	D1	D0
DLAB	SBCB	PS2	PS1	PS0	STB	WLS1	WLS0

DLAB (Divisor Latch Access Bit) = 0: truy xuất RBR, THR, IER, = 1 cho phép đặt bộ chia tần trong UART để cho phép đạt tốc độ truyền mong muốn.

UART dùng dao động thạch anh với tần số 1.8432 MHz đưa qua bộ chia 16 thành tần số 115,200 Hz. Khi đó, tùy theo giá trị trong BRDL và BRDH, ta sẽ có tốc độ mong muốn.



Ví dụ như đường truyền có tốc độ truyền 2,400 bps có giá trị chia  $115,200 / 2,400 = 48d = 0030h \rightarrow BRDL = 30h, BRDH = 00h$ .

Một số giá trị thông dụng xác định tốc độ truyền cho như sau:

Tốc độ (bps)	BRDH	BRDL
1,200	00h	60h
2,400	00h	30h
4,800	00h	18h
9,600	00h	0Ch
19,200	00h	06h
38,400	00h	03h
57,600	00h	02h
115,200	00h	01h

SBCB (Set Break Control Bit) =1: cho phép truyền tín hiệu Break (=0) trong khoảng thời gian lớn hơn một khung

PS (Parity Select):

PS2	PS1	PS0	Mô tả
X	X	0	Không kiểm tra
0	0	1	Kiểm tra lẻ
0	1	1	Kiểm tra chẵn
1	0	1	Parity là mark
1	1	1	Parity là space

STB (Stop Bit) = 0: 1 bit stop, =1: 1.5 bit stop (khi dùng 5 bit dữ liệu) hay 2 bit stop (khi dùng 6, 7, 8 bit dữ liệu).

WLS (Word Length Select):

WLS1	WLS0	Độ dài dữ liệu
0	0	5 bit
0	1	6 bit
1	0	7 bit
1	1	8 bit

Một ví dụ khi lập trình trực tiếp trên cổng như sau:

```
.MODEL SMALL

.STACK 100h .DATA
```

```

Com1 EQU 3F8h

Com_int EQU 08h

Buffer DB 251 DUP(?)

Bufferin DB 0

Bufferout DB 0

Char DB ?

Seg_com DW ?

Off_com DW ?

Mask_int DB ?

Msg DB 'Press any key to exit$'

.CODE

Main PROC

MOV AX,@DATA MOV DS,AX

MOV AH,35h

MOV AL,Com_int

INT 21h

MOV Seg_com,ES

MOV Off_com,BX

PUSH DS

MOV BX,CS

MOV DS,BX

LEA DX,Com_ISR MOV AH,35h

```

```
MOV AL,Com_int
INT 21h
POP DS
MOV DX,Com1+3 MOV AL,80h
OUT DX,AL
MOV DX,Com1
MOV AL,0Ch
OUT DX,AL
MOV DX,Com1+1
MOV AL,00h
OUT DX,AL
MOV DX,Com1+3
MOV AL,03h
OUT DX,AL
MOV DX,Com1+4
MOV AL,03h
OUT DX,AL
MOV DX,21h
IN AL,DX
MOV Mask_int,AL
AND AL,0EFh
OUT DX,AL
```

```
MOV AL,01h
MOV DX,Com1+1
OUT DX,AL
MOV AH,09h
LEA Dx,Msg
INT 21h
Lap:
MOV AH,0Bh
INT 21h
CMP AL,0FFh
JE Exit
MOV AL,bufferin
CMP AL,bufferout
JE Lap
MOV AL,buffer[bufferout]
MOV char,AL
INC bufferout
MOV AL,bufferout CMP AL,251
JNE Next
MOV bufferout,0 Next:
MOV DL,char
MOV AH,02h
```

```
INT 21h

MOV AL, char

MOV DX, Com1

OUT DX, AL

JMP Lap

Exit:

MOV AL, Mask_int

OUT 21h, AL

MOV DX, Off_com

MOV BX, Seg_com

MOV DS, BX

MOV AH, 35h

MOV AL, Com_int

INT 21h

MOV AH, 4Ch

INT 21h

Main ENDP

Com_ISR PROC

MOV DX, Com1+5

IN AL, DX

AND AL, 1

JZ exit_ISR
```

```
MOV DX, Com1

IN AL, DX

MOV buffer[bufferin], AL

INC bufferin

MOV AL, bufferin

CMP AL, 251

JNE Exit_ISR

MOV bufferin, 0

Exit_ISR:
MOV AL, 20h ;Báo cho PIC kết thúc ngắt

OUT 20h, AL

IRET

Com_ISR ENDP

END Main
```

# Truyền thông nối tiếp dùng ActiveX

## Mô tả

Việc truyền thông nối tiếp trên Windows được thực hiện thông qua một ActiveX có sẵn là Microsoft Comm Control.. ActiveX này được lưu trữ trong file MSCOMM32.OCX. Quá trình này có hai khả năng thực hiện điều khiển trao đổi thông tin:

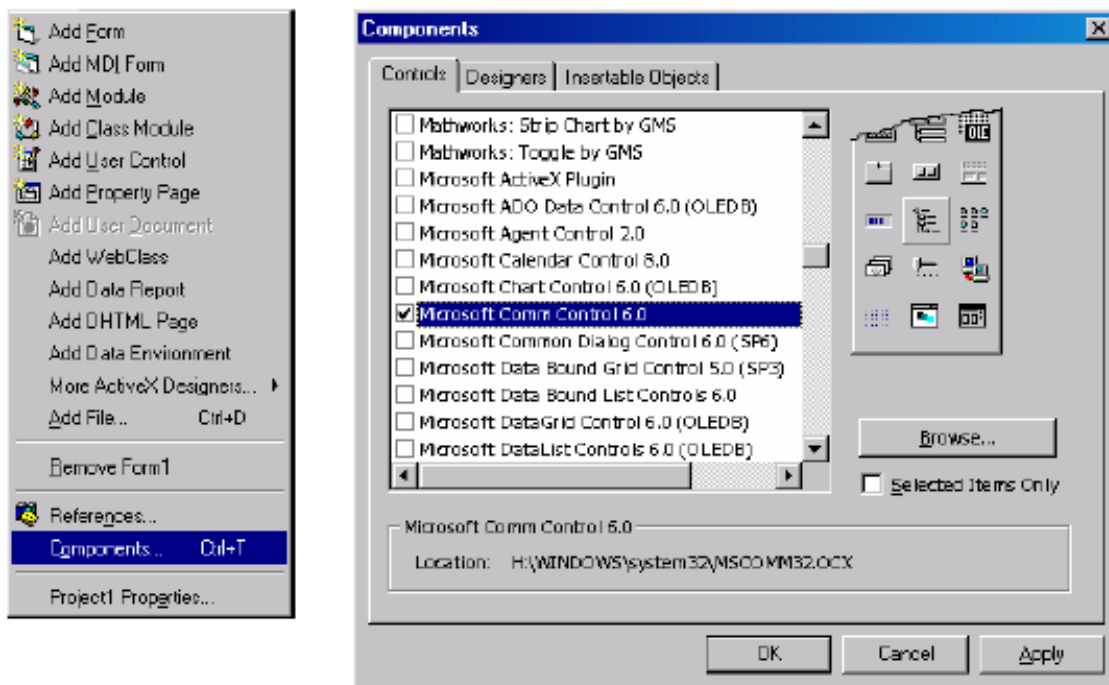
### - Điều khiển sự kiện:

Truyền thông điều khiển sự kiện là phương pháp tốt nhất trong quá trình điều khiển việc trao đổi thông tin. Quá trình điều khiển thực hiện thông qua sự kiện OnComm.


### - Hỏi vòng:

Quá trình điều khiển bằng phương pháp hỏi vòng thực hiện thông qua kiểm tra các giá trị của thuộc tính CommEvent sau một chu kỳ nào đó để xác định xem có sự kiện nào xảy ra hay không. Thông thường phương pháp này sử dụng cho các chương trình nhỏ.

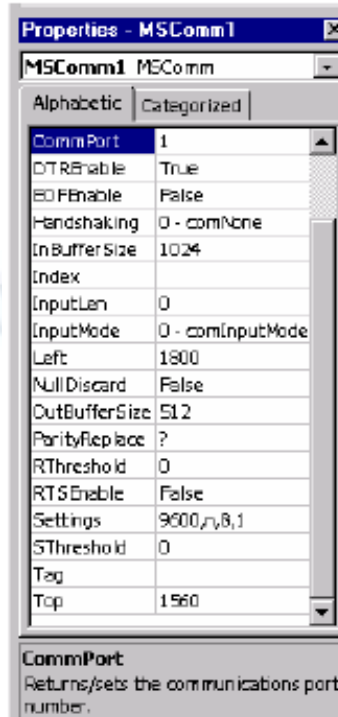
ActiveX MsComm được bổ sung vào một Visual Basic Project thông qua menu **Project > Components:**



Bổ sung đối tượng MsComm vào VBP

Biểu tượng của MsComm:  và các thuộc tính cơ bản mô tả như sau:

Thuộc tính	Mô tả
CommPort	Số thứ tự cổng truyền thông
Input	Nhận ký tự từ bộ đệm
Output	Xuất ký tự ra cổng nối tiếp
PortOpen	Mở / đóng cổng
Settings	Xác định các tham số truyền



*Các thuộc tính của đối tượng MSComm*

## Các thuộc tính

- **Settings:**

Xác định các tham số cho cổng nối tiếp. Cú pháp:

```
MSComm1.Settings = ParamString MSComm1: tên đối tượng
```

ParamString: là một chuỗi có dạng như sau: "BBBB,P,D,S"

BBBB: tốc độ truyền dữ liệu (bps) trong đó các giá trị hợp lệ là:



110	2400	38400
300	9600 (mặc định)	56000
600	14400	188000
1200	19200	256000

P: kiểm tra chẵn lẻ, với các giá trị:

Giá trị	Mô tả
O	Odd (kiểm tra lẻ)
E	Even (kiểm tra chẵn)
M	Mark (luôn bằng 1)
S	Space (luôn bằng 0)
N	Không kiểm tra

D: số bit dữ liệu (4, 5, 6, 7 hay 8), mặc định là 8 bit S: số bit stop (1, 1.5, 2)

MSComm1.**Settings** = "9600,O,8,1" sẽ xác định tốc độ truyền 9600bps, kiểm tra parity chẵn với 1 bit stop và 8 bit dữ liệu.

- **CommPort:**

Xác định số thứ tự của cổng truyền thông, cú pháp:

MSComm1.**CommPort** = PortNumber

PortNumber là giá trị nằm trong khoảng từ 1 →99, mặc định là 1.

MSComm1.**CommPort** = 1 xác định sử dụng COM1

- **PortOpen:**

Đặt trạng thái hay kiểm tra trạng thái đóng / mở của cổng nối tiếp. Nếu dùng thuộc tính này để mở cổng nối tiếp thì phải sử dụng trước 2 thuộc tính Settings và CommPort. Cú pháp:

MSComm1.**PortOpen** = True | False

Giá trị xác định là True sẽ thực hiện mở cổng và False để đóng cổng đồng thời xoá nội dung của các bộ đệm truyền, nhận.

VD: Mở cổng COM1 với tốc độ truyền 9600 bps

MSComm1.**Settings** = "9600,N,8,1"

```
MSCComm1. CommPort = 1
```

```
MSCComm1. PortOpen = True
```

- **Các thuộc tính nhận dữ liệu:**

**Input:** nhận một chuỗi ký tự và xoá khỏi bộ đệm. Cú pháp:

```
InputString = MSCComm1.Input
```

Thuộc tính này kết hợp với **InputLen** để xác định số ký tự đọc vào. Nếu **InputLen** = 0 thì sẽ đọc toàn bộ dữ liệu có trong bộ đệm.

**InBufferCount:** số ký tự có trong bộ đệm nhận. Cú pháp:

```
Count = MSCComm1 . InBufferCount
```

Thuộc tính này cũng được dùng để xoá bộ đệm nhận bằng cách gán giá trị 0.  

```
MSCComm1.InBufferCount = 0
```

**InBufferSize:** đặt và xác định kích thước bộ đệm nhận (tính bằng byte). Cú pháp:  

```
MSCComm1.InBufferCount = NumByte
```

Giá trị mặc định là 1024 byte. Kích thước bộ đệm này phải đủ lớn để tránh tình trạng mất dữ liệu.

VD: Đọc toàn bộ nội dung trong bộ đệm nhận nếu có dữ liệu

```
MSCComm1. InputLen = 0
```

```
If MSCComm1. InBufferCount <> 0 Then InputString = MSCComm1.  
Input End If
```

- **Các thuộc tính xuất dữ liệu:**

Bao gồm các thuộc tính **Output**, **OutBufferCount** và **OutBufferSize**, chức năng của các thuộc tính này giống như các thuộc tính nhập.

- **CDTimeout:**

Đặt và xác định khoảng thời gian lớn nhất (tính bằng ms) từ lúc phát hiện sóng mang cho đến lúc có dữ liệu. Nếu quá khoảng thời gian này mà vẫn chưa có dữ liệu thì sẽ gán thuộc tính **CommEvent** là **CDTO** (Carrier Detect Timeout Error) và tạo sự kiện **OnComm**. Cú pháp:

`MSComm1.CDTimeout = NumTime`

- **DSRTimeout:**

Xác định thời gian chờ tín hiệu DSR trước khi xảy ra sự kiện OnComm.

- **CTSTimeout:**

Đặt và xác định khoảng thời gian lớn nhất (tính bằng ms) đợi tín hiệu CTS trước khi đặt thuộc tính CommEvent là CTSTO và tạo sự kiện OnComm. Cú pháp:

`MSComm1.CTSTimeout = NumTime`

- **CTSHolding:**

Xác định đã có tín hiệu CTS hay chưa, tín hiệu này dùng cho quá trình bắt tay bằng phần cứng (cho biết DCE sẵn sàng nhận dữ liệu), trả về giá trị True hay False.

- **DSR Holding:**

Xác định trạng thái DSR (báo hiệu sự tồn tại của DCE), trả về giá trị True hay False.

- **CDHolding:**

Xác định trạng thái CD, trả về giá trị True hay False.

- **DTREnable:**

Đặt hay xoá tín hiệu DTR để báo sự tồn tại của DTE. Cú pháp:

`MSComm1.DTREnable = True | False`

- **RTSEnable:**

Đặt hay xoá tín hiệu RTS để yêu cầu truyền dữ liệu đến DTE. Cú pháp:

`MSComm1.RTSEnable = True | False`

- **NullDiscard:**

Cho phép nhận các ký tự NULL (rỗng) hay không (= True: cấm). Cú pháp:

`MSComm1.NullDiscard = True | False`

- **SThreshold:**

Số byte trong bộ đệm truyền làm phát sinh sự kiện OnComm. Nếu giá trị này bằng 0 thì sẽ không tạo sự kiện OnComm. Cú pháp:

```
MSCComm1. SThreshold = NumChar
```

- **HandShaking:**

Chọn giao thức bắt tay khi thực hiện truyền dữ liệu. Cú pháp:

```
MSCComm1. HandShaking = Protocol
```

Các giao thức truyền bao gồm:

Protocol	Giá trị	Mô tả
ComNone	0	Không bắt tay (mặc định)
ComXon/Xoff	1	Bắt tay phần mềm (Xon/Xoff)
ComRTS	2	Bắt tay phần cứng (RTS/CTS)
ComRTSXon/Xoff	3	Bắt tay phần cứng và phần mềm

- **CommEvent:**

Trả lại các lỗi truyền thông hay sự kiện xảy ra tại cổng nối tiếp

**Các sự kiện:**

Sự kiện	Giá trị	Mô tả
ComEvSend	1	Đã truyền ký tự
ComEvReceive	2	Khi có ký tự trong bộ đệm nhận
ComEvCTS	3	Có thay đổi trên CTS (Clear To Send)
ComEvDSR	4	Có thay đổi trên DSR (Data Set Ready)
ComEvCD	5	Có thay đổi trên CD (Carrier Detect)
ComEvRing	6	Phát hiện chuông
ComEvEOF	7	Nhận ký tự kết thúc file

**Các lỗi truyền thông:**