

Chương 4

GIAO TIẾP CÔNG NỐI TIẾP

1. Cấu trúc công nối tiếp

Công 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:

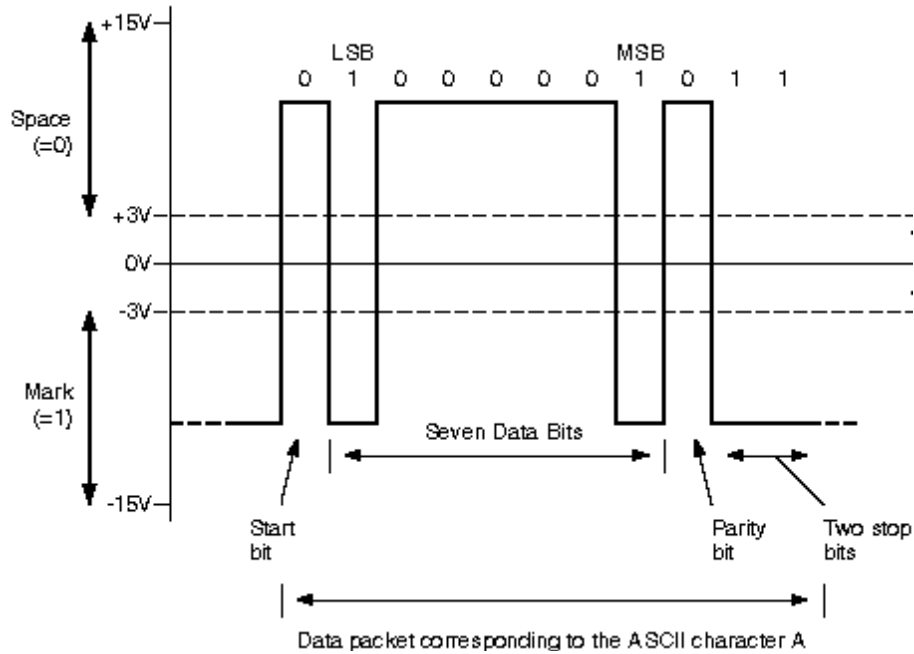
- Đơn công (simplex connection): dữ liệu chỉ được truyền theo 1 hướng.
- Bán song công (half-duplex): dữ liệu truyền theo 2 hướng, nhưng mỗi thời điểm chỉ được truyền theo 1 hướng.
- Song công (full-duplex): số liệu được truyền đồng thời theo 2 hướng.

Định dạng của khung truyền dữ liệu theo chuẩn RS-232 như sau:

Start	D0	D1	D2	D3	D4	D5	D6	D7	P	Stop
0										1

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):



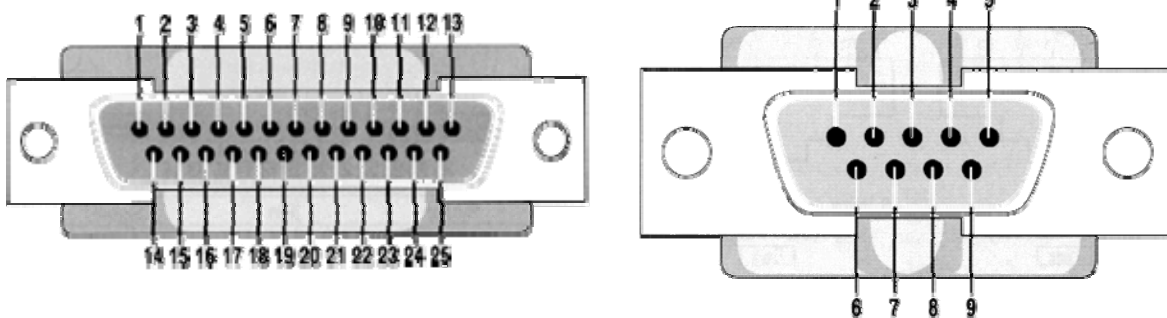
Hình 4.1 – Tín hiệu truyền của ký tự ‘A’

Các đặc tính kỹ thuật của chuẩn RS-232 như sau:

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:





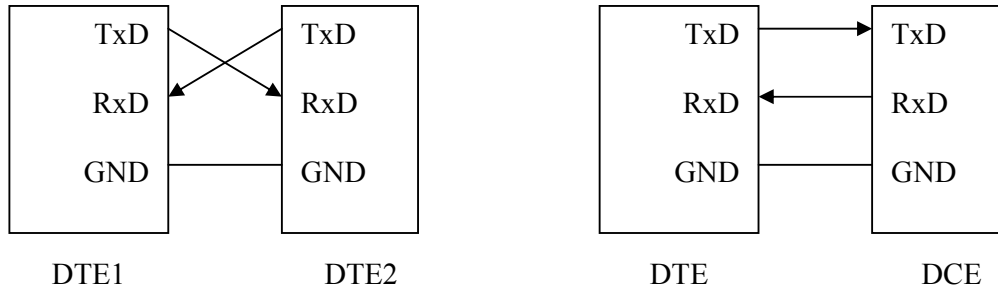
Hình 4.2 – Sơ đồ chân cổng nối tiếp

Cổng COM có hai dạng: đầu nối DB25 (25 chân) và đầu nối DB9 (9 chân) mô tả như hình 4.2. 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 cổng
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

2. Truyền thông giữa hai nút

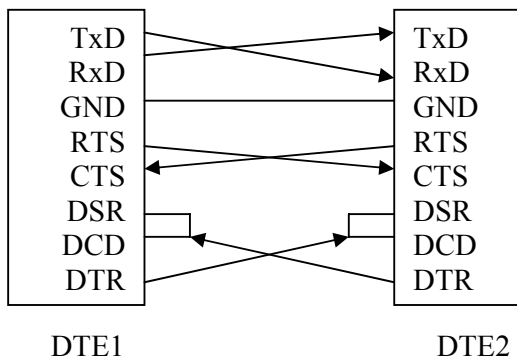
Các sơ đồ khi kết nối dùng cổng nối tiếp:



Hình 4.3 – 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:



Hình 4.4 – 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.

3. 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 $BA_{COM1} + 1 = 3F9h$).

❖ **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:

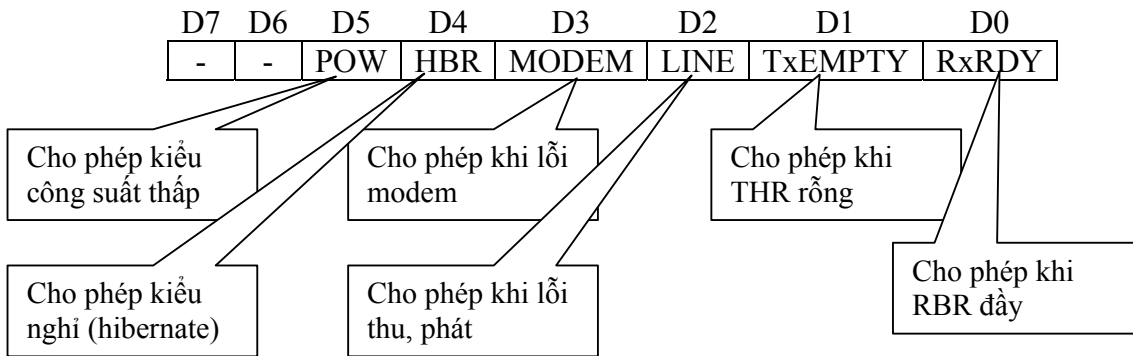
D7	D6	D5	D4	D3	D2	D1	D0
00: không có FIFO 11: cho phép FIFO	Cho phép FIFO 64 byte (trong 16750)	-	1: ngắt time-out (trong 16550)	Xác định nguồn gốc ngắt	0: có ngắt 1: không ngắt		

D2	D1	Ưu tiên	Tên	Nguồn	D2 – D0 bị xoá khi
0	0	4	Đường truyền	Lỗi khung, thu dề, 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

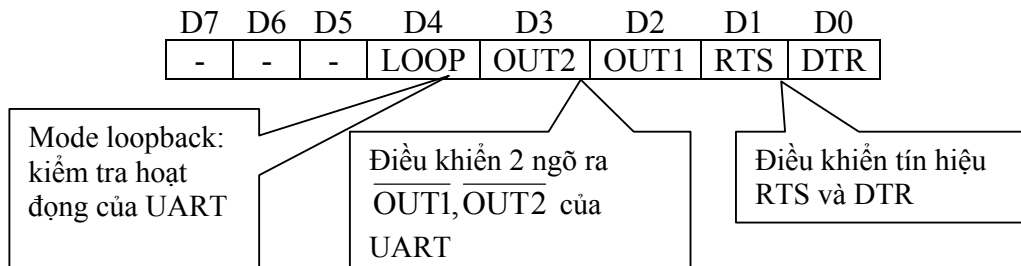
(mức 1 ưu tiên cao nhất)

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

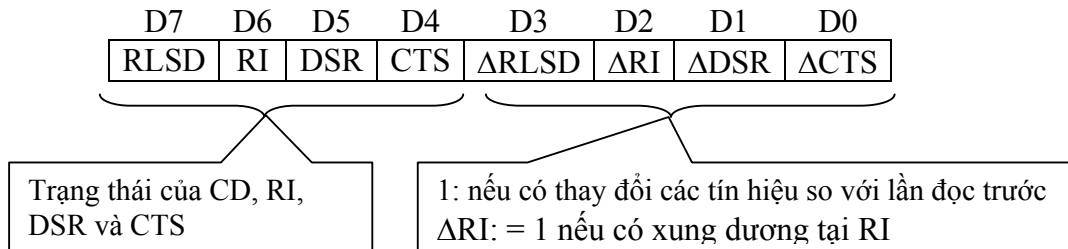
IER cho phép hay cấm các nguyên nhân ngắt khác nhau (1: cho phép, 0: cấm ngắt)



❖ **MCR (Modem Control Register):**



❖ **MSR (Modem Status Register):**



❖ **LSR (Line Status Register):**



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)

RxDR: 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 ? ; Vector ng•t c•
    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 ; L•u vector ng•t c•
    MOV Off_com,BX

    PUSH DS
    MOV BX,CS
    MOV DS,BX
    LEA DX,Com_ISR
    MOV AH,35h ;G•n vector ng•t m•i
    MOV AL,Com_int
    INT 21h
    POP DS

    MOV DX,Com1+3 ; ••a ch• LCR
    MOV AL,80h ; Set DLAB = 1 cho phép ••nh t•c
    OUT DX,AL ; •• truy•n d• li•u
```

```

MOV DX, Com1      ; G•i byte th•p
MOV AL, 0Ch
OUT DX, AL

MOV DX, Com1+1
MOV AL, 00h      ; G•i byte cao → 000Ch: xác ••nh
OUT DX, AL      ; t•c •• truy•n 9600bps

MOV DX, Com1+3   ; LCR = 0000 0011B
MOV AL, 03h     ; DLAB = 0, SBCB = 0 → c•m Break
OUT DX, AL     ; PS = 000 → no parity
                ; STB = 0 → 1 stop bit
                ; WLS = 11 → 8 bit d• li•u

MOV DX, Com1+4   ; Tác ••ng ••n DTR và RTS
MOV AL, 03h     ; MCR = 0000 0011b → DTR=RTS = 1
OUT DX, AL     ; → ng• DTR và RTS c•a c•ng n•i
                ; ti•p = 0

MOV DX, 21h     ; Ki•m tra tr•ng th•i ng•t
IN AL, DX      ; D7 - D0 xác ••nh các IRQi
MOV Mask_int, AL ; =0: cho phép, =1: c•m

AND AL, 0EFh   ; = 1110 1111b → cho phép IRQ4
OUT DX, AL     ; → cho phép COM1

MOV AL, 01h    ; IER = 0000 0001b → cho phép
MOV DX, Com1+1 ; ng•t khi RBR ••y
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          ; Xu•t giá tr• ra màn hình
MOV AH, 02h
INT 21h

MOV AL, char          ; Xu•t ra c•ng noi ti•p
MOV DX, Com1
OUT DX, AL
JMP Lap

Exit:
MOV AL, Mask_int
OUT 21h, AL           ; Khôi ph•c tr•ng thái ng•t

MOV DX, Off_com
MOV BX, Seg_com
MOV DS, BX
MOV AH, 35h          ; Khôi ph•c vector ng•t
MOV AL, Com_int
INT 21h

MOV AH, 4Ch
INT 21h
Main ENDP

Com_ISR PROC
MOV DX, Com1+5       ; ••c noi dung LSR
IN AL, DX
AND AL, 1            ; N•u D0 = 1 thì có d• li•u
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 k•t thúc ng•t
OUT 20h, AL
IRET
Com_ISR ENDP
END Main

```

4. Truyền thông nối tiếp dùng ActiveX

4.1. 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:

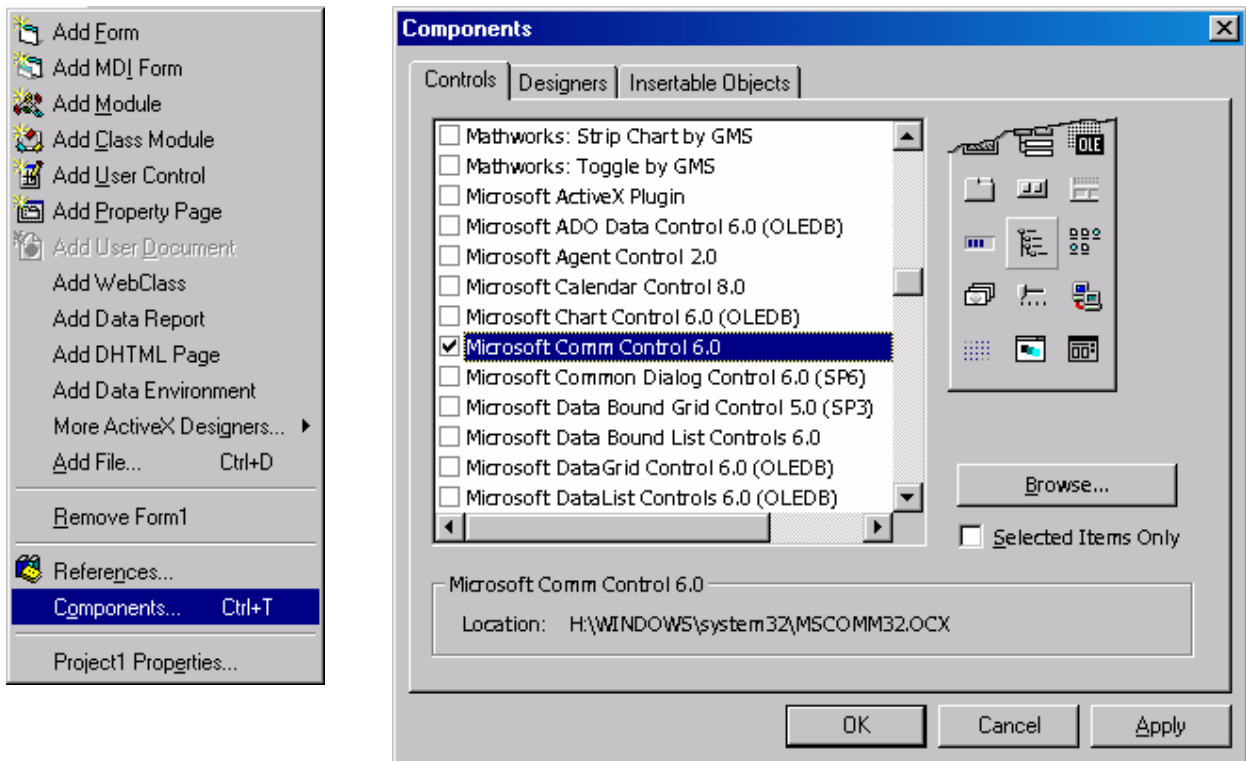
- **Điều khiển sự kiện:**

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.

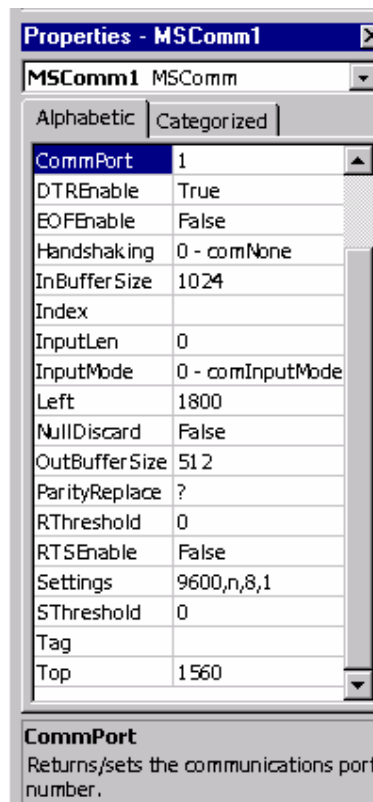
ActiveX MsComm được bổ sung vào một Visual Basic Project thông qua menu **Project > Components:**



Hình 4.5 – 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



Hình 4.6 – Các thuộc tính của đối tượng MSComm

4.2. 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)

VD:

`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.

VD:

`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"`

`MSComm1.CommPort = 1`

`MSComm1.PortOpen = True`

❖ **Các thuộc tính nhận dữ liệu:**

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

`InputString = MSComm1.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 = MSComm1.InBufferCount`

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

`MSComm1.InBufferCount = 0`

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

`MSComm1.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

```

MSComm1.InputLen = 0
If MSComm1.InBufferCount <> 0 Then
    InputString = MSComm1.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.

❖ DSRHOLDING:

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:

MSComm1.**SThreshold** = NumChar

❖ HandShaking:

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

MSComm1.**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:

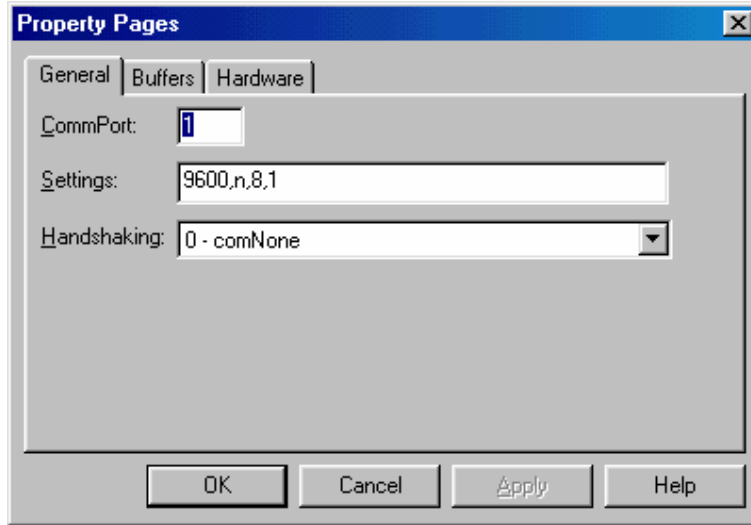
Lỗi	Giá trị	Mô tả
ComBreak	1001	Nhận tín hiệu Break
ComCTSTO	1002	Carrier Detect Timeout
ComFrame	1004	Lỗi khung
ComOver	1006	Phần cứng không đọc ký tự trước khi gọi ký tự kế
ComCDTO	1007	Carrier Detect Timeout
ComRxOver	1008	Tràn bộ đệm nhận
ComRxParity	1009	Lỗi parity
ComTxFull	1010	Tràn bộ đệm truyền

4.3. Sự kiện OnComm

Sự kiện OnComm xảy ra bất cứ khi nào giá trị của thuộc tính CommEvent thay đổi. Các thuộc tính RThreshold và SThreshold = 0 sẽ cấm sự kiện OnComm khi thực hiện nhận hay gửi dữ liệu. Thông thường, SThreshold = 0 và RThreshold = 1.

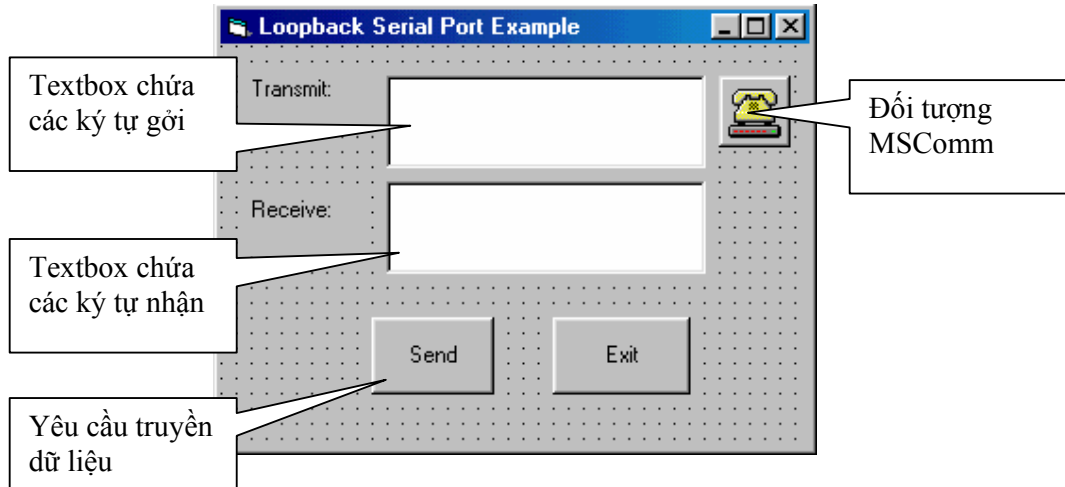
Một chương trình truyền nhận đơn giản thực hiện bằng cách nối chân TxD với RxD của cổng COM1 (loopback). Phương pháp này dùng để kiểm tra cổng nối tiếp.

Thuộc tính cơ bản của cổng nối tiếp:



Hình 4.7 – Các thuộc tính cơ bản của MSComm

Cửa sổ chương trình thực thi:



Hình 4.8 – Cửa sổ chương trình loopback

Chương trình nguồn:

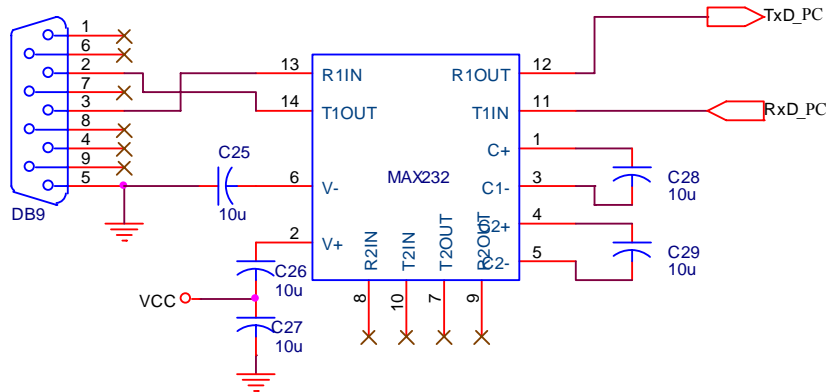
```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
```

```
Begin VB.Form Form1
  Caption           =      "Loopback Serial Port
Example"
  ClientHeight     =      3195
  ClientLeft       =      60
  ClientTop        =      345
  ClientWidth      =      4680
  LinkTopic        =      "Form1"
  ScaleHeight      =      3195
  ScaleWidth       =      4680
  StartUpPosition =      3   'Windows Default
Begin VB.CommandButton cmdExit
  Caption          =      "Exit"
  Height           =      615
  Left             =      2640
  TabIndex        =      5
  Top              =      2160
  Width           =      1095
End
Begin VB.CommandButton cmdSend
  Caption          =      "Send"
  Height           =      615
  Left             =      1200
  TabIndex        =      4
  Top              =      2160
  Width           =      975
End
Begin VB.TextBox txtReceive
  Height           =      735
  Left             =      1320
  Locked           =      -1   'True
  TabIndex        =      3
  Top              =      1080
  Width           =      2535
End
Begin VB.TextBox txtTransmit
  Height           =      735
  Left             =      1320
  TabIndex        =      0
  Top              =      240
  Width           =      2535
End
Begin MSCommLib.MSComm MSComm1
  Left             =      3960
  Top              =      240
  _ExtentX        =      1005
  _ExtentY        =      1005
```

```
        _Version           = 393216
        DTREnable          = -1   `True
        RThreshold         = 1
    End
    Begin VB.Label Label2
        Caption             = "Receive:"
        Height              = 375
        Left                 = 240
        TabIndex            = 2
        Top                  = 1200
        Width                = 855
    End
    Begin VB.Label Label1
        Caption             = "Transmit:"
        Height              = 375
        Left                 = 240
        TabIndex            = 1
        Top                  = 240
        Width                = 975
    End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub cmdExit_Click()
MSComm1.PortOpen = False   `Đóng cổng
End
End Sub
Private Sub cmdSend_Click()
MSComm1.Output = Trim(txtTransmit.Text) `Gửi dữ liệu
End Sub
Private Sub Form_Load()
MSComm1.CommPort = 1           `COM1
MSComm1.Settings = "9600,n,8,1" `Tốc độ 9600bps
MSComm1.PortOpen = True       ` Mở cổng
End Sub
Private Sub MSComm1_OnComm()
If (MSComm1.CommEvent = comEvReceive) Then
txtReceive.Text = txtReceive.Text + MSComm1.Input
End If
End Sub
```

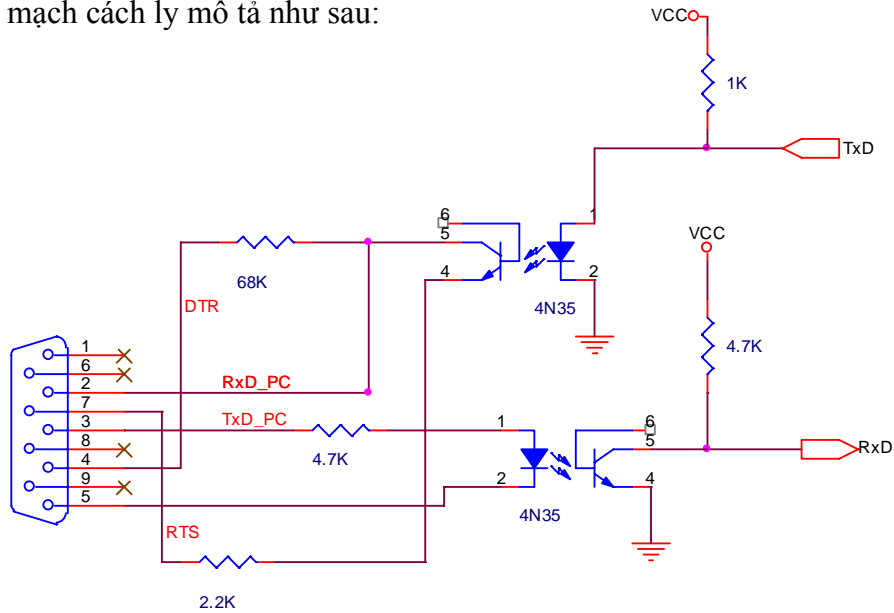
5. Giao tiếp với vi điều khiển

Khi thực hiện giao tiếp với vi điều khiển, ta phải dùng thêm mạch chuyển mức logic từ TTL → 232 và ngược lại. Các vi mạch thường sử dụng là MAX232 của Maxim hay DS275 của Dallas. Mạch chuyển mức logic mô tả như sau:



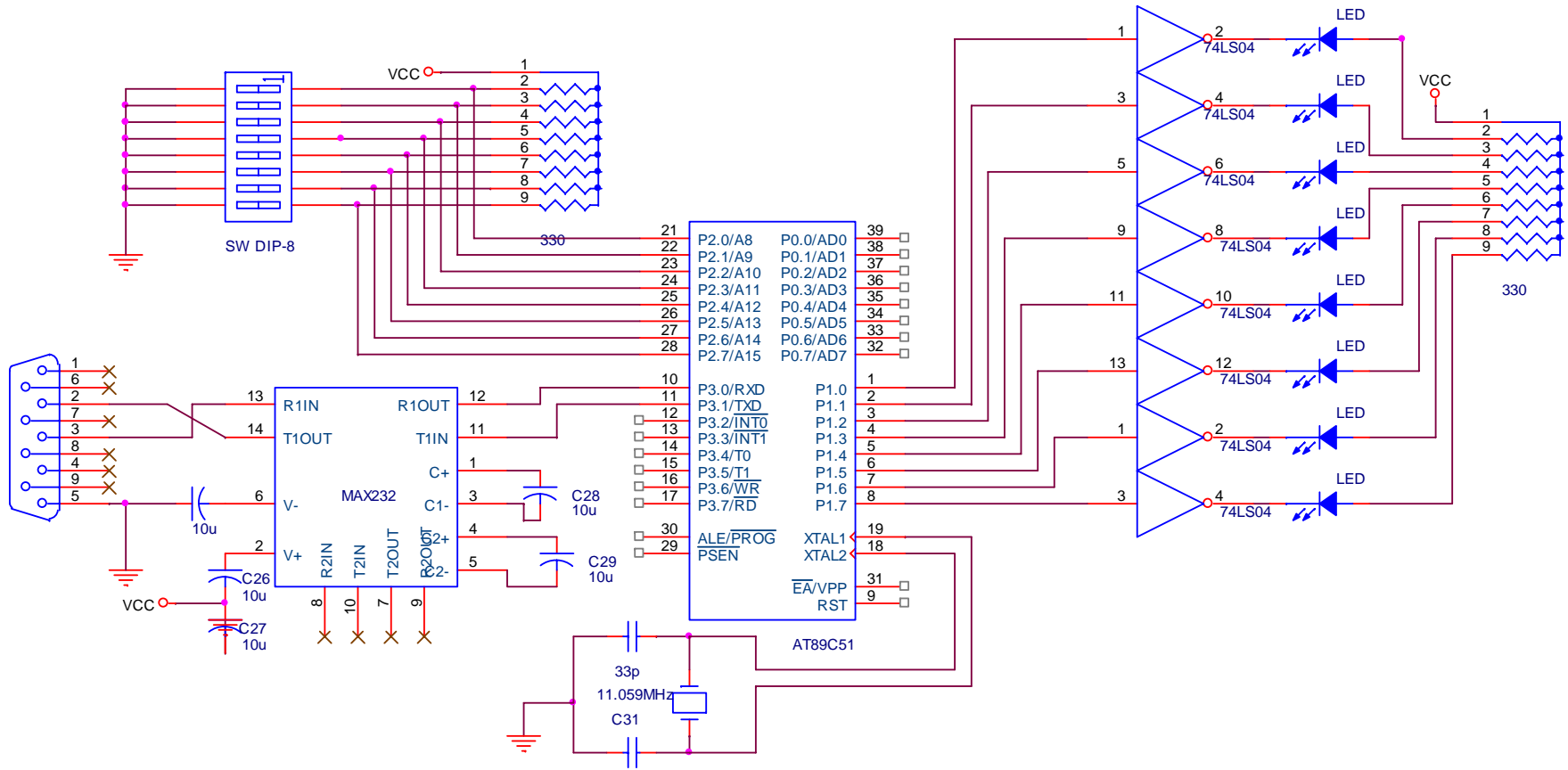
Hình 4.9 – Mạch chuyển mức logic TTL ↔ RS232

Tuy nhiên, khi sử dụng mạch chuyển mức logic dùng các vi mạch thì đòi hỏi phải dùng chung GND giữa máy tính và vi mạch → có khả năng làm hỏng cổng nối tiếp khi xảy ra hiện tượng chập mạch ở mạch ngoài. Do đó, ta có thể dùng thêm opto 4N35 để cách ly về điện. Sơ đồ mạch cách ly mô tả như sau:



Hình 4.10 – Mạch chuyển mức logic TTL ↔ RS232 cách ly

Khi giao tiếp, vi điều khiển chính là một DTE nên sẽ nối RxD của máy tính với TxD của vi điều khiển và ngược lại. Mạch kết nối đơn giản giữa vi điều khiển và máy tính như sau:



Hình 4.11 – Kết nối với vi điều khiển

Chương trình nguồn cho vi điều khiển AT89C51:

```
MOV  TMOD, #20h
MOV  SCON, #52h; Truyền 8 bit dữ liệu, no parity
MOV  TH1, #(-3); Tốc độ truyền 9600 bps
MOV  TL1, #(-3)
SETB TR1
```

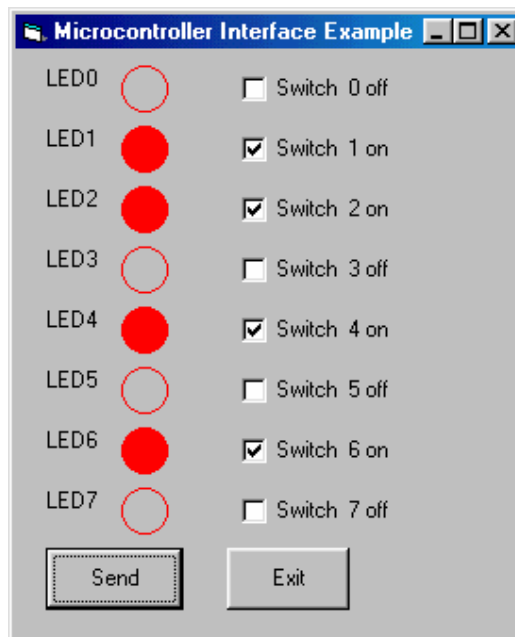
Receive:

```
JNB  RI, Transmit ; Có dữ liệu hay không
CLR  RI
MOV  A, SBUF      ; Nếu có thì xuất ra LED
MOV  P1, A
```

Transmit:

```
JNB  TI, Receive ; Đã truyền xong chưa
CLR  TI
MOV  A, P2       ; Nếu xong thì truyền trạng thái
MOV  SBUF, A    ; của công tắc SW DIP-8
JMP  Receive
```

Giao diện của chương trình trên máy tính:



Hình 4.12 – Chương trình giao tiếp với vi điều khiển

Chương trình nguồn:

```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
```

```
Begin VB.Form Form1
  Caption           =      "Microcontroller Interface
Example"
  ClientHeight      =      4665
  ClientLeft        =      60
  ClientTop         =      345
  ClientWidth       =      4020
  LinkTopic         =      "Form1"
  ScaleHeight       =      4665
  ScaleWidth        =      4020
  StartupPosition  =      3  'Windows Default
Begin VB.CheckBox chkSW
  Height            =      375
  Index             =      7
  Left              =      1800
  TabIndex          =      17
  Top               =      3480
  Width             =      1575
End
Begin VB.CheckBox chkSW
  Height            =      375
  Index             =      6
  Left              =      1800
  TabIndex          =      16
  Top               =      3000
  Width             =      1575
End
Begin VB.CheckBox chkSW
  Height            =      375
  Index             =      5
  Left              =      1800
  TabIndex          =      15
  Top               =      2520
  Width             =      1575
End
Begin VB.CheckBox chkSW
  Height            =      375
  Index             =      4
  Left              =      1800
  TabIndex          =      14
  Top               =      2040
  Width             =      1575
End
Begin VB.CheckBox chkSW
  Height            =      375
  Index             =      3
  Left              =      1800
```

```
        TabIndex      = 13
        Top           = 1560
        Width         = 1575
    End
    Begin VB.CheckBox chkSW
        Height         = 375
        Index          = 2
        Left           = 1800
        TabIndex      = 12
        Top           = 1080
        Width         = 1575
    End
    Begin VB.CheckBox chkSW
        Height         = 375
        Index          = 1
        Left           = 1800
        TabIndex      = 11
        Top           = 600
        Width         = 1575
    End
    Begin VB.CheckBox chkSW
        Height         = 375
        Index          = 0
        Left           = 1800
        TabIndex      = 10
        Top           = 120
        Width         = 1575
    End
    Begin VB.CommandButton cmdExit
        Caption        = "Exit"
        Height         = 495
        Left           = 1680
        TabIndex      = 9
        Top           = 3960
        Width         = 975
    End
    Begin MSCommLib.MSComm MSComm1
        Left           = 3360
        Top           = 3960
        _ExtentX       = 1005
        _ExtentY       = 1005
        _Version       = 393216
        DTREnable      = -1 'True
        RThreshold     = 1
    End
    Begin VB.CommandButton cmdSend
        Caption        = "Send"
```

```
        Height           = 495
        Left             = 240
        TabIndex        = 8
        Top              = 3960
        Width           = 1095
    End
    Begin VB.Label lblLED
        BackStyle        = 0   'Transparent
        Caption          = "LED7"
        Height           = 375
        Index            = 7
        Left             = 240
        TabIndex        = 7
        Top              = 3480
        Width           = 1095
    End
    Begin VB.Label lblLED
        BackStyle        = 0   'Transparent
        Caption          = "LED6"
        Height           = 375
        Index            = 6
        Left             = 240
        TabIndex        = 6
        Top              = 3000
        Width           = 975
    End
    Begin VB.Label lblLED
        BackStyle        = 0   'Transparent
        Caption          = "LED5"
        Height           = 375
        Index            = 5
        Left             = 240
        TabIndex        = 5
        Top              = 2520
        Width           = 975
    End
    Begin VB.Label lblLED
        BackStyle        = 0   'Transparent
        Caption          = "LED4"
        Height           = 375
        Index            = 4
        Left             = 240
        TabIndex        = 4
        Top              = 2040
        Width           = 975
    End
    Begin VB.Label lblLED
```

```
        BackStyle      = 0  'Transparent
        Caption       = "LED3"
        Height        = 375
        Index         = 3
        Left          = 240
        TabIndex      = 3
        Top           = 1560
        Width         = 975
    End
    Begin VB.Label lblLED
        BackStyle      = 0  'Transparent
        Caption       = "LED2"
        Height        = 375
        Index         = 2
        Left          = 240
        TabIndex      = 2
        Top           = 1080
        Width         = 975
    End
    Begin VB.Label lblLED
        BackStyle      = 0  'Transparent
        Caption       = "LED1"
        Height        = 375
        Index         = 1
        Left          = 240
        TabIndex      = 1
        Top           = 600
        Width         = 975
    End
    End
    Begin VB.Label lblLED
        BackStyle      = 0  'Transparent
        Caption       = "LED0"
        Height        = 375
        Index         = 0
        Left          = 240
        TabIndex      = 0
        Top           = 120
        Width         = 975
    End
    End
    Begin VB.Shape shpLED
        BorderColor    = &H000000FF&
        FillColor      = &H000000FF&
        FillStyle      = 0  'Solid
        Height         = 375
        Index          = 7
        Left           = 840
        Shape          = 3  'Circle
    End
```

```
        Top           = 3480
        Width        = 375
    End
    Begin VB.Shape shpLED
        BorderColor   = &H000000FF&
        FillColor     = &H000000FF&
        FillStyle     = 0 'Solid
        Height        = 375
        Index         = 6
        Left          = 840
        Shape         = 3 'Circle
        Top           = 3000
        Width        = 375
    End
    Begin VB.Shape shpLED
        BorderColor   = &H000000FF&
        FillColor     = &H000000FF&
        FillStyle     = 0 'Solid
        Height        = 375
        Index         = 5
        Left          = 840
        Shape         = 3 'Circle
        Top           = 2520
        Width        = 375
    End
    Begin VB.Shape shpLED
        BorderColor   = &H000000FF&
        FillColor     = &H000000FF&
        FillStyle     = 0 'Solid
        Height        = 375
        Index         = 4
        Left          = 840
        Shape         = 3 'Circle
        Top           = 2040
        Width        = 375
    End
    Begin VB.Shape shpLED
        BorderColor   = &H000000FF&
        FillColor     = &H000000FF&
        FillStyle     = 0 'Solid
        Height        = 375
        Index         = 3
        Left          = 840
        Shape         = 3 'Circle
        Top           = 1560
        Width        = 375
    End
```

```
Begin VB.Shape shpLED
    BorderColor      = &H000000FF&
    FillColor        = &H000000FF&
    FillStyle        = 0 'Solid
    Height           = 375
    Index            = 2
    Left             = 840
    Shape            = 3 'Circle
    Top              = 1080
    Width            = 375
End
Begin VB.Shape shpLED
    BorderColor      = &H000000FF&
    FillColor        = &H000000FF&
    FillStyle        = 0 'Solid
    Height           = 375
    Index            = 1
    Left             = 840
    Shape            = 3 'Circle
    Top              = 600
    Width            = 375
End
Begin VB.Shape shpLED
    BorderColor      = &H000000FF&
    FillColor        = &H000000FF&
    FillStyle        = 0 'Solid
    Height           = 375
    Index            = 0
    Left             = 840
    Shape            = 3 'Circle
    Top              = 120
    Width            = 375
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub cmdExit_Click()
    If MSComm1.PortOpen Then
        MSComm1.PortOpen = False
    End If
End
End
End Sub

Private Sub cmdSend_Click()
```

```

Dim t As Integer
Dim i As Integer
t = 0
For i = 0 To 7
    t = t + (2 ^ i) * (1 - shpLED(i).FillStyle)
Next i
MSComm1.Output = Chr(t)
End Sub

Private Sub Form_Load()
MSComm1.Settings = "9600,N,8,1"
MSComm1.CommPort = 1
MSComm1.PortOpen = True
End Sub

Private Sub lblLED_Click(Index As Integer)
shpLED(Index).FillStyle = 1 - shpLED(Index).FillStyle
End Sub

Private Sub MSComm1_OnComm()
Dim t As String
Dim n As Integer
Dim i As Integer
If MSComm1.CommEvent = comEvReceive Then
    n = Asc(MSComm1.Input)
    For i = 0 To 7
        chkSW(i).Value = n Mod 2
        If chkSW(i).Value = 0 Then
            chkSW(i).Caption = "Switch " & Str(i) &
" off"
        Else
            chkSW(i).Caption = "Switch " & Str(i) &
" on"
        End If
        n = Fix(n / 2)
    Next i
End If
End Sub

```

6. Giao tiếp với MODEM

6.1. Giao tiếp

Quá trình trao đổi dữ liệu giữa máy tính và Modem được thực hiện theo cơ chế bắt tay phân cứng hay phân mềm.

- **Bắt tay phần cứng:** máy tính muốn truyền dữ liệu thì cho RTS = 1 và chờ Modem trả lời bằng tín hiệu CTS. Ngược lại, Modem muốn truyền dữ liệu thì cho DSR = 1 và chờ tín hiệu DTR từ máy tính.
- **Bắt tay phần mềm:** dùng ký tự Xon (Ctrl-S) và Xoff (Ctrl-Q) để bắt đầu truyền hay kết thúc truyền.

Các giao thức truyền dữ liệu trên Modem:

- **XModem:** chia thành khối 128 byte, mỗi khối chèn thêm CRC 4 byte.
- **YModem:** khối 1024 byte.
- **ZModem:** khối có kích thước thay đổi tùy theo đường truyền.

Quy tắc truyền lệnh trên Modem:

- Mỗi dòng lệnh của modem bắt đầu bằng ký tự AT, ngoại trừ lệnh A/ và +++.
- Dòng lệnh có thể chứa nhiều lệnh.
- Kết thúc lệnh bằng ký tự Enter (mã ASCII là 13) ngoại trừ lệnh A/ và +++.
- Dòng lệnh cuối cùng được lưu trong modem. Có thể dùng lệnh A/ để thực hiện lại lệnh này.
- Thông báo kết quả thực hiện lệnh của modem có thể ở dạng từ chữ hay số (giá trị mặc định là chữ). Có thể sử dụng lệnh V để lựa chọn dạng thông báo là chữ hay số.
- Để hoạt động đúng, modem cần có các thông số xác định. Nếu không có sự thay đổi cần thiết, modem hoạt động theo giá trị mặc định(default). Nếu thông số trong lệnh bị bỏ qua, giá trị thông số mặc định là 0.

6.2. Các lệnh cơ bản của Modem

Lệnh	Mô tả
+++	Chuyển Modem sang chế độ lệnh
A/	Lặp lại lệnh trước
A	Cho phép kết nối và phát tín hiệu sóng mang. Modem sẽ báo tín hiệu CONNECT nếu thu được tín hiệu sóng mang từ modem đầu cuối. Nếu không thu được sóng mang, modem sẽ gác máy và thông báo NO CARRIER
DPn	Quay số điện thoại n dạng xung
DTn	Quay số điện thoại n dạng tone
H0	Gác máy
H1	Nhấc máy
O0	Chuyển về chế độ dữ liệu
O1	Chế độ điều chỉnh Modem
Q0	Cho phép Modem gửi thông báo đến DTE (mặc định)
Q1	Cấm Modem gửi thông báo
Q2	Gửi thông báo khi Modem chủ động kết nối, không gửi khi Modem nhận cuộc gọi
V0	Nhận thông báo dạng số

V1	Nhận thông báo dạng ký tự (mặc định)
Sn = V	Nạp giá trị V vào thanh ghi Sn S0 = V: chờ V hồi chuông trước khi trả lời, V = 0 – 255 (mặc định V = 0: không trả lời) S6 = V: chờ V giây trước khi quay số (mặc định V = 2) S7 = V: chờ V giây kể từ lúc gọi đến lúc nhận được tín hiệu, nếu không sẽ thông báo lỗi
Sn?	Đọc nội dung thanh ghi Sn
Z0	Reset Modem về cấu hình 0
Z1	Reset Modem về cấu hình 1
L0, L1, L2, L3	Âm lượng loa Modem
M0	Tắt loa
M1	Mở loa cho đến khi nhận được sóng mang (mặc định)
M2	Mở loa
M3	Tắt loa khi quay số và nhận sóng mang

6.3. Các thanh ghi thông dụng trên modem

Thanh ghi S0: xác định số hồi chuông nhận được mà sau đó modem sẽ trả lời một cách tự động. Giá trị trong thanh ghi này có thể thay đổi trong khoảng từ 0-255. mặc định giá trị là 0 (không trả lời).

Thanh ghi S1: Thanh ghi S1 chỉ có tác dụng khi thanh ghi S0 khác 0, dùng để đếm số hồi chuông thu được.

Thanh ghi S2: xác định giá trị thập phân của các ký tự (mã ASSCII) được dùng làm ký tự thoát, Giá trị mặc định là 43(+)

Thanh ghi S3: xác định ký tự được dùng để kết thúc một dòng lệnh, mặc nhiên là 13 (tương ứng là Enter)

Thanh ghi S4: xác định ký tự xuống dòng sau ký tự kết thúc, giá trị mặc nhiên là 10 (line feed)

Thanh ghi S5: xác định phím xoá lui, giá trị mặc nhiên là 8 (backspace)

Thanh ghi S6: xác định thời gian đợi sau khi truy cập đường điện thoại và trước khi tiến hành quay digit đầu tiên trong một lệnh quay số. Đây là thời gian trì hoãn cho phép để dial tone cung cấp từ đường truyền. Giá trị mặc nhiên và tối thiểu là 2s.

Thanh ghi S7: xác định thời gian mà modem đợi tín hiệu sóng mang trước khi gác máy. Giá trị mặc định là 30s.

Thanh ghi S8: xác định thời gian tạm dừng cho mỗi dấu phẩy ',' trong chuỗi lệnh quay số. Giá trị mặc định là 2s

Thanh ghi S9: xác định thời gian mà tín hiệu sóng mang phải hiện diện để modem có thể nhận biết được, giá trị mặc định là 600ms. Giá trị này nếu quá lớn sẽ gây lỗi trong dữ liệu truyền.

Thanh ghi S10: xác định thời gian cho phép tín hiệu sóng mang có thể biến mất trong chốc lát nào đó mà không cắt cuộc nối. Ôn định trong khoảng 100-25500ms, giá trị mặc nhiên tùy vào khả năng chống nhiễu của từng modem, thường là 700ms.

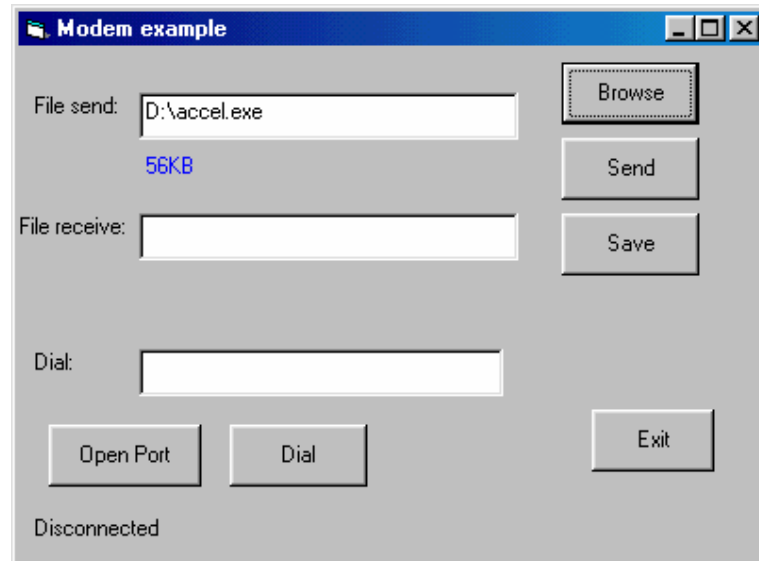
Thanh ghi S11: xác định tốc độ quay số khi sử dụng phương pháp quay số tone, giá trị mặc nhiên tùy vào modem, thường vào khoảng 70ms.

Thanh ghi S12: xác định thời gian an toàn khi truy nhập vào ký tự thoát (+++). Nếu giá trị nhỏ quá có thể nhập không kịp, giá trị lớn quá so với tốc độ nhập cũng không thể thoát được.

6.4. Các thông báo của Modem

Dạng ký tự	Dạng số	Ý nghĩa
OK	0	Lệnh thành công
CONNECT	1	Kết nối 300 bps
RING	2	Có tín hiệu chuông
NO CARRIER	3	Không có sóng mang
ERROR	4	Lỗi: nhận lệnh không giá trị, sai kiểm tra, hàng lệnh quá dài
CONNECT 1200	5	Kết nối 1200bps
NO DIAL TONE	6	Không có âm hiệu mời quay số
BUSY	7	Máy bận
NO ANSWER	8	Không có tín hiệu trả lời
CONNECT 2400	10	Kết nối 2400bps
CONNECT 4800	11	Kết nối 4800bps
CONNECT 9600	12	Kết nối 9600bps
CONNECT 14400	13	Kết nối 14400bps
CONNECT 19200	14	Kết nối 19200bps
CONNECT 16800	15	Kết nối 16800bps
CONNECT 57600	18	Kết nối 57600bps
CONNECT 7200	24	Kết nối 7200bps
CONNECT 12000	25	Kết nối 12000bps
CONNECT 28800	32	Kết nối 28800bps
CONNECT 115200	33	Kết nối 115200bps
CARRIER 300	40	Phát hiện sóng mang
CARRIER 9600	50	Phát hiện sóng mang
CARRIER 28800	58	Phát hiện sóng mang

Ví dụ lập trình điều khiển Modem như sau:



Hình 4.13 – Giao tiếp và điều khiển Modem

Chương trình nguồn:

```

VERSION 5.00
Object           =      "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Object           =      "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "COMDLG32.OCX"
Begin VB.Form frmModem
    Caption       =      "Modem example"
    ClientHeight  =      4065
    ClientLeft    =      60
    ClientTop     =      345
    ClientWidth   =      5925
    LinkTopic     =      "Form1"
    ScaleHeight   =      4065
    ScaleWidth    =      5925
    StartUpPosition = 3 'Windows Default
Begin VB.CommandButton cmdSave
    Caption       =      "Save"
    Height        =      495
    Left          =      4320
    TabIndex     =      14
    Top          =      1320
    Width        =      1095
End
Begin VB.TextBox txtReceive
    Height        =      375
    Left         =      960

```

```
        TabIndex      = 12
        Top           = 1320
        Width        = 3015
    End
    Begin VB.Timer Timer1
        Enabled        = 0      'False
        Interval       = 1000
        Left           = 4920
        Top            = 2400
    End
    Begin VB.CommandButton cmdExit
        Caption        = "Exit"
        Height         = 495
        Left           = 4560
        TabIndex       = 10
        Top            = 2880
        Width          = 975
    End
    Begin VB.TextBox txtDial
        Height         = 375
        Left           = 960
        TabIndex       = 7
        Top            = 2400
        Width          = 2895
    End
    Begin VB.CommandButton cmdDial
        Caption        = "Dial"
        Height         = 495
        Left           = 1680
        TabIndex       = 5
        Top            = 3000
        Width          = 1095
    End
    Begin VB.CommandButton cmdSend
        Caption        = "Send"
        Height         = 495
        Left           = 4320
        TabIndex       = 4
        Top            = 720
        Width          = 1095
    End
    Begin VB.CommandButton cmdOpen
        Caption        = "Open Port"
        Height         = 495
        Left           = 240
        TabIndex       = 3
        Top            = 3000
    End
```

```
        Width           = 1215
    End
    Begin VB.CommandButton cmdBrowse
        Caption          = "Browse"
        Height           = 495
        Left              = 4320
        TabIndex         = 1
        Top               = 120
        Width             = 1095
    End
    Begin MSComDlg.CommonDialog diagSend
        Left              = 4200
        Top               = 3120
        _ExtentX          = 847
        _ExtentY          = 847
        _Version          = 393216
    End
    Begin VB.TextBox txtSend
        Height            = 375
        Left              = 960
        TabIndex         = 0
        Top               = 360
        Width             = 3015
    End
    Begin MSCommLib.MSComm MSComm1
        Left              = 5160
        Top               = 3000
        _ExtentX          = 1005
        _ExtentY          = 1005
        _Version          = 393216
        DTREnable        = -1 'True
        Handshaking       = 2
        NullDiscard       = -1 'True
        RThreshold        = 1
        RTSEnable         = -1 'True
    End
    Begin VB.Label Label3
        Caption           = "File receive:"
        Height            = 375
        Left              = 0
        TabIndex         = 13
        Top               = 1320
        Width             = 855
    End
    Begin VB.Label lblReceive
        Caption           = "Receive file !!! Select
file name."
```

```
        ForeColor      =    &H000000FF&
        Height         =    375
        Left           =    840
        TabIndex       =    11
        Top            =    1920
        Visible        =    0    'False
        Width          =    2895
    End
    Begin VB.Label lblStatus
        Caption         =    "Disconnected"
        Height         =    375
        Left           =    120
        TabIndex       =    9
        Top            =    3720
        Width          =    5775
    End
    Begin VB.Label Label2
        Caption         =    "Dial:"
        Height         =    375
        Left           =    120
        TabIndex       =    8
        Top            =    2400
        Width          =    735
    End
    Begin VB.Label Label1
        Caption         =    "File send:"
        Height         =    375
        Left           =    120
        TabIndex       =    6
        Top            =    360
        Width          =    735
    End
    Begin VB.Label lblSize
        ForeColor      =    &H00FF0000&
        Height         =    375
        Left           =    960
        TabIndex       =    2
        Top            =    840
        Width          =    1815
    End
End
Attribute VB_Name = "frmModem"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Connected As Boolean
```

```
Private SendFlag As Boolean
Private ReceiveFlag As Boolean
Private FileReceive As Integer
Private CRFlag As Boolean
Private Sub cmdBrowse_Click()
On Error GoTo Loi
diagSend.FileName = ""
diagSend.Filter = "All files (*.*)|*.*"
diagSend.InitDir = App.Path
diagSend.ShowOpen
txtSend.Text = diagSend.FileName
lblSize.Caption = Str(Round(FileLen(txtSend.Text) /
1024, 2)) + "KB"
Exit Sub
Loi:
lblSize.Caption = "0 KB"
txtSend.Text = ""
End Sub

Private Sub cmdDial_Click()
If Not MSComm1.PortOpen Then
MsgBox "Comm Port Closed. Open first!!!",
vbOKOnly + vbCritical, "Error"
ElseIf Trim(txtDial.Text) = "" Then
MsgBox "Enter phone's number!!!", vbOKOnly +
vbCritical, "Error"
Else
If cmdDial.Caption = "Dial" Then
MSComm1.Output = "ATDT" & Trim(txtDial.Text)
+ vbCr
cmdDial.Caption = "Hang up"
lblStatus.Caption = "Dialing ..."
Else
MSComm1.Output = "ATH1" + vbCr
cmdDial.Caption = "Dial"
lblStatus.Caption = "Hang up"
End If
End If
End Sub

Private Sub cmdExit_Click()
If MSComm1.PortOpen Then
MSComm1.PortOpen = False
End If
End
End Sub
```

```
Private Sub cmdOpen_Click()
MSComm1.PortOpen = Not MSComm1.PortOpen
If MSComm1.PortOpen Then
    cmdOpen.Caption = "Close Port"
    MSComm1.Output = "ATS0=5" + vbCrLf
    Call Form_Load
Else
    cmdOpen.Caption = "Open Port"
    lblStatus.Caption = "Disconnected"
End If
End Sub

Private Sub cmdSave_Click()
FileReceive = FreeFile
ReceiveFlag = True
Timer1.Enabled = False
Do
    diagSend.FileName = ""
    diagSend.ShowSave
    If Trim(diagSend.FileName) = "" Then
        MsgBox "File name error!!",
vbCritical + vbOKOnly, "Error"
    End If
    Loop While Trim(diagSend.FileName) = ""
    txtReceive.Text = diagSend.FileName
    MSComm1.Output = "RECEIVE" + vbCrLf
    Open Trim(txtReceive.Text) For Output As
#FileReceive
End Sub

Private Sub cmdSend_Click()
Dim FileNum As Integer
Dim Buffer As String
If Not MSComm1.PortOpen Then
    MsgBox "Comm Port Closed. Open first!!!",
vbOKOnly + vbCritical, "Error"
ElseIf Not Connected Then
    MsgBox "Not connected!!!", vbOKOnly +
vbCritical, "Error"
ElseIf Trim(txtSend.Text) = "" Then
    MsgBox "Select a file to send!!!", vbOKOnly +
vbCritical, "Error"
Else
    MSComm1.Output = "SEND" + vbCrLf
    Do
        DoEvents
    
```

```
    Loop While Not SendFlag
    FileNum = FreeFile
    Open Trim(txtSend.Text) For Input As #FileNum
    Do
        Input #FileNum, Buffer
        If Right(Buffer, 1) <> vbCr Then Buffer =
Buffer + vbCrLf
        MSComm1.Output = Buffer
    Loop While Not EOF(FileNum)
    MSComm1.Output = "END FILE"
    Close #FileNum
    SendFlag = False
End If
End Sub

Private Sub Form_Load()
    Connected = False
    SendFlag = False
    ReceiveFlag = False
    CRFlag = False
End Sub

Private Sub MSComm1_OnComm()
    Dim Buffer As String
    Dim Buffer1 As String
    Dim Buff As String
    Dim i As Integer
    Select Case MSComm1.CommEvent
        Case comEvRing
            lblStatus.Caption = "Ringing..."
        Case comEvCD
            If MSComm1.CD Holding Then
                lblStatus.Caption = "Connected"
                Connected = True
            Else
                lblStatus.Caption = "Disconnected"
                Connected = False
            End If
        Case comEvReceive
            Buffer = MSComm1.Input
            If InStr(Buffer, "SEND") Then
                Timer1.Enabled = True
                Exit Sub
            End If
            If InStr(Buffer, "RECEIVE") Then
                SendFlag = True
                Timer1.Enabled = False
            End If
        End Select
End Sub
```

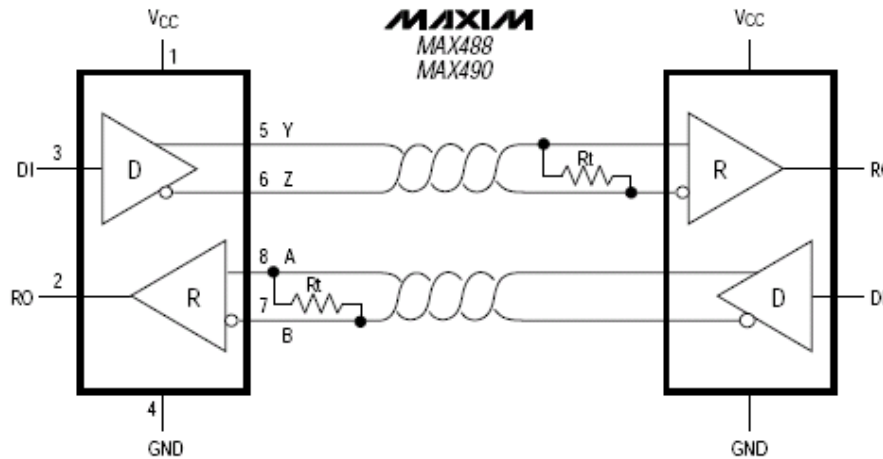
```
        Buffer = ""
        Exit Sub
    End If
    If InStr(Buffer, "CONNECT") Then
        Connected = True
        lblStatus.Caption = "Connected"
        Exit Sub
    End If
    If ReceiveFlag Then
        Buffer1 = ""
        For i = 1 To Len(Buffer)
            Buff = Mid$(Buffer, i, 1)
            If Buff = Chr$(13) Then
                CRFlag = True
                Buff = ""
            ElseIf Buff = Chr$(10) Then
                CRFlag = False
                Buff = ""
            If Not CRFlag Then
                Buffer1 = Buffer1 + Buff
            End If
        Next i
        Print #FileReceive, Buffer1
    End If
    If InStr(Buffer, "END FILE") Then
        Close #FileReceive
        Call Form_Load
    End If
    Case comEvEOF
        lblStatus = "Disconnected"
        Connected = False
End Select
End Sub

Private Sub Timer1_Timer()
    lblReceive.Visible = Not lblReceive.Visible
End Sub

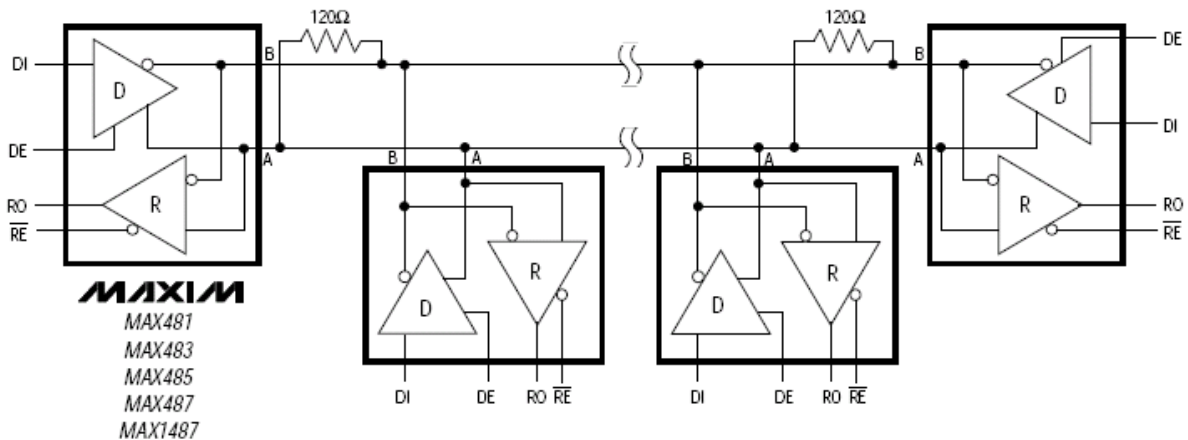
Private Sub txtSend_LostFocus()
    On Error GoTo Loi
    lblSize.Caption = Str(Round(FileLen(txtSend.Text) /
1024, 2)) + "KB"
    Exit Sub
Loi:
    lblSize.Caption = "0 KB"
    txtSend.Text = ""
End Sub
```

7. Mạng 485

Chuẩn RS232 dùng đường truyền không cân bằng vì các tín hiệu lấy chuẩn là GND chung nên dễ bị ảnh hưởng của nhiễu làm tốc độ và khoảng cách truyền bị giới hạn. Khi muốn tăng khoảng cách truyền, một phương pháp có thể sử dụng là dùng 2 dây truyền vì sai vì lúc này 2 dây có cùng đặc tính nên sẽ loại trừ được nhiễu chung. Hai chuẩn được sử dụng là RS422 và RS485 nhưng thông thường sử dụng RS485. Điện áp vi sai yêu cầu phải lớn hơn 200mV. Nếu $V_{AB} > 200 \text{ mV}$ thì tương ứng với logic 1 và $V_{AB} < -200 \text{ mV}$ tương ứng với logic 0. Chuẩn RS485 sử dụng hai điện trở kết thúc là 120Ω tại hai đầu xa nhất của đường truyền và sử dụng dây xoắn đôi.



Hình 4.13 – Chuẩn giao tiếp RS422



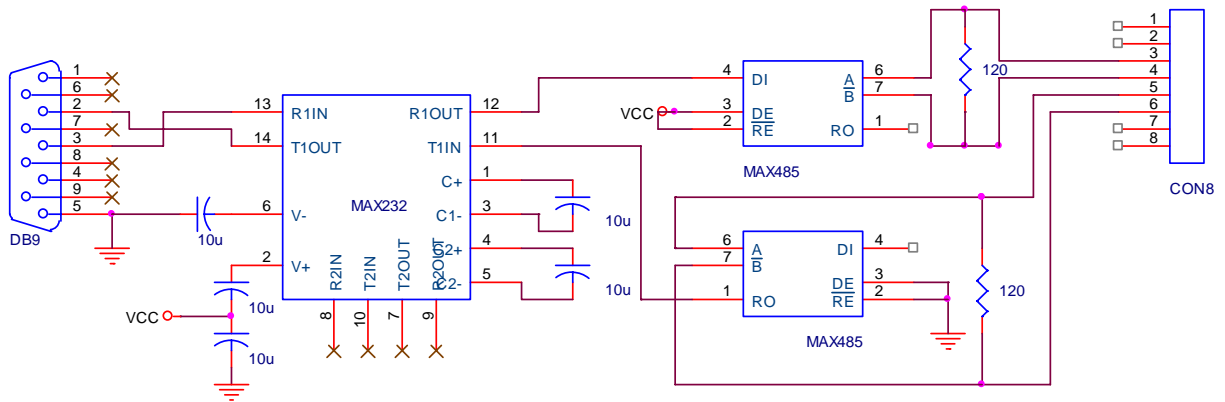
Hình 4.14 – Chuẩn giao tiếp RS485

Các đặc tính kỹ thuật:

Đặc tính	RS422	RS485
Số thiết bị truyền	1	32
Số thiết bị nhận	10	32

Chiều dài cable cực đại	1200m	1200m
Tốc độ truyền cực đại (từ 12 – 1200m)	10Mps – 100Kbps	10Mps – 100Kbps
Điện áp cực đại tại ngõ ra thiết bị truyền	-0.25V ÷ 6V	-7V ÷ 12V
Điện áp ngõ vào thiết bị nhận	-10V ÷ 10V	-7V ÷ 12V

Đối với chuẩn RS232, khoảng cách truyền không cho phép đi xa nên khi muốn thực hiện truyền ở khoảng cách xa thì phải chuyển từ RS232 sang chuẩn RS485 để truyền đi và sau đó chuyển từ RS485 sang RS232 để máy tính có thể nhận dạng được. Sơ đồ mạch chuyển đổi từ RS232 sang RS485 và ngược lại mô tả như sau:



Hình 4.15 – Chuyển đổi từ RS323 sang RS485 và ngược lại