



BÀI GIẢNG

ĐIỆN TỬ MÁY TÍNH

Lời nói đầu

Ngày nay, với sự phát triển vượt bậc của các thiết bị điện tử, kỹ thuật ghép nối với máy tính đang được ứng dụng ngày càng nhiều đem lại hữu ích cho cuộc sống. Máy tính được sử dụng trong các nhà máy xí nghiệp trong các dây chuyền sản xuất, trong các ứng dụng thiết kế ngôi nhà thông minh. Các thiết bị, hệ thống đo lường ghép nối với máy tính có độ chính xác cao.

Giáo trình này giới thiệu cấu trúc các cổng, khe cắm và cách thiết kế các mạch đo lường, điều khiển ghép nối với máy tính. Khi ghép nối với máy tính, ngoài phần cứng ghép nối ta còn phải viết chương trình trên máy tính để giao tiếp với các cổng, khe cắm. Giáo trình cũng đưa ra rất nhiều các ví dụ bổ ích với đầy đủ sơ đồ nguyên lý mạch điện và chương trình viết bằng ngôn ngữ lập trình Visual Basic, ta cũng có thể viết bằng các ngôn ngữ lập trình C, Pascal nhưng Visual Basic cho ta giao diện đẹp và tiện dụng hơn. Nội dung cuốn sách bao gồm 4 chương:

Chương 1: Giới thiệu qua các khái niệm về cấu trúc cơ bản của một máy tính, các dạng thông tin trước khi đưa vào máy tính, cấu trúc cơ bản của một khối ghép nối.

Chương 2: Giới thiệu phương thức truyền tin song song sử dụng cổng song song và khe cắm mở rộng. Nội dung chương này cũng giới thiệu cấu trúc của cổng song song và các khe cắm như ISA, PCI,... cách lập trình giao tiếp qua các cổng này.

Chương 3: Giới thiệu phương thức truyền tin nối tiếp, từ đó trình bày cấu trúc cổng nối tiếp RS-232 và cổng USB, cách lập trình giao tiếp qua các cổng này.

Chương 4: Giới thiệu các bước cơ bản trong quá trình thiết kế ứng dụng các Modul ghép nối với máy tính và các ứng dụng đo lường và điều khiển thông qua các cổng của máy tính từ đó các bạn đọc có thể thiết kế, chế tạo được các ứng dụng thực tế, hữu ích.

Mặc dù tài liệu này đã được sử dụng để giảng dạy tại trường Đại Học Công nghiệp Hà Nội nhiều năm nhưng cũng không tránh khỏi những sai sót. Chúng tôi rất mong nhận được những ý kiến đóng góp của bạn đọc để lần tái bản tới được hoàn thiện hơn:

Mọi đóng góp xin gửi về: Bộ môn Điện tử máy tính, khoa Điện tử, trường đại học Công nghiệp Hà Nội, Minh Khai – Từ Liêm – Hà Nội, điện thoại 043 7655121 (266)

CÁC TÁC GIẢ

Mục lục

Lời nói đầu.....	3
Chương 1 Máy tính và khối ghép nối.....	6
1.1 Máy tính và khối ghép nối.....	6
1.1.1 Các dạng tin trao đổi của máy tính.....	7
1.1.2 Các loại thông tin trao đổi của máy tính.....	7
1.1.3 Các phương thức trao đổi tin của máy tính.....	8
1.2 Vai trò, nhiệm vụ và cấu trúc của khối ghép nối.....	10
1.2.1 Vai trò.....	10
1.2.2 Nhiệm vụ.....	10
1.3 Cấu trúc chung của khối ghép nối.....	11
1.3.1 Khối phối hợp đường dây.....	11
1.3.2 Khối giải mã địa chỉ - lệnh.....	11
1.3.3 Khối xử lý ngắt.....	12
1.4 Bài tập cuối chương.....	13
Chương 2 Ghép nối trao đổi tin song song theo chương trình.....	14
2.1 Các vi mạch đệm, chốt song song thông dụng.....	14
2.1.1 Vi mạch 74HC240.....	14
2.1.2 Vi mạch 74HC244.....	14
2.1.3 Vi mạch 74HC245.....	15
2.1.4 Vi mạch 74HC373.....	15
2.1.5 Vi mạch 74HC573.....	16
2.2 Cổng song song.....	16
2.2.1 Giới thiệu.....	16
2.2.2 Giao diện một hướng ở cổng song song.....	21
2.2.3 Giao diện 2 hướng dùng cổng song song - module vào ra 8 bit.....	29
2.3 Rãnh cắm mở rộng.....	32
2.3.1 Giới thiệu.....	32
2.3.2 Giới thiệu một số loại BUS.....	33
2.4 Bài tập cuối chương.....	37
Chương 3 Ghép nối trao đổi tin nối tiếp.....	39
3.1 Khái niệm về truyền tin nối tiếp.....	39
3.1.1 Khái niệm.....	39
3.1.2 Các phương thức truyền tin nối tiếp.....	39
3.2 Cổng nối tiếp.....	40
3.2.1 Giới thiệu.....	40
3.2.2 Lập trình cho cổng nối tiếp RS232.....	42
3.2.3 Modul vào ra 8 bit dùng cổng RS232.....	50
3.3 Cổng USB (Universal Serial Bus).....	58
3.3.1 Giới thiệu.....	58
3.3.2 Những đặc trưng của USB.....	59
3.3.3 Cấu trúc cổng USB.....	59
3.3.4 Truyền dữ liệu qua cổng USB.....	61
3.3.5 Hub USB.....	61
3.4 Bài tập cuối chương.....	62
Chương 4 Thiết kế ứng dụng đo lường điều khiển bằng máy tính.....	64

4.1 Quy trình thiết kế các ứng dụng đo lường điều khiển bằng máy tính.....	64
4.2 Các vi mạch số thông dụng.....	66
4.2.1 Vi mạch ghép nối vào ra song song theo chương trình 8255A	66
4.2.2 Vi mạch đếm định thời lập trình được 8253 (PROGRAMABLE COUNTER AND TIMER).....	71
4.2.3 Các bộ biến đổi AD (ANALOG DIGITAL CONVERTER)	79
4.2.4 Các bộ biến đổi DA (DIGITAL ANALOG CONVERTER).....	82
4.3 Các thiết kế ứng dụng ghép nối với máy tính.....	83
4.3.1 Điều khiển vi mạch 8255 qua cổng song song	83
4.3.2 Điều khiển vi mạch 8253 qua cổng song song	85
4.3.3 Mạch đếm sản phẩm qua cổng RS-232	87
4.3.4 Voltmet điện tử ghép nối qua cổng RS-232	91
4.4.5 Điều khiển Led 7 đoạn qua cổng USB	93
4.4 Bài tập cuối chương	97
<i>Phụ lục</i>	98
<i>Bảng mã ASCII</i>	98
<i>Tài liệu tham khảo</i>	101

Chương 1 Máy tính và khối ghép nối

Khi sản xuất máy tính, các nhà sản xuất đã dự trữ sẵn các con đường cho phép người sử dụng có thể ghép nối với máy tính. Chúng ta có thể sử dụng máy tính để điều khiển các thiết bị bên ngoài bằng cách sử dụng cổng song song, cổng nối tiếp hay cổng USB. Trước khi nghiên cứu cấu trúc của các cổng, các khe cắm, chương này sẽ giới thiệu về tổng quan về các dạng thông tin trao đổi với máy tính, các phương thức trao đổi thông tin với máy tính và cấu trúc của một khối ghép nối.

1.1 Máy tính và khối ghép nối

Cấu trúc của một máy tính có thể được phân chia thành ba khối chính:

- Khối xử lý trung tâm (CPU): Làm nhiệm vụ thu thập và xử lý mọi dữ liệu.
- Khối nhớ (Memory): Lưu trữ các loại dữ liệu khác nhau đưa vào, lấy ra từ CPU.
- Khối phối hợp vào ra (I/O): Làm nhiệm vụ tương thích giữa các thiết bị ngoài và đường dây (bus) trong của máy tính.

Trong máy tính thường có một số thiết bị ngoài thông dụng như: Màn hình, bàn phím, chuột, máy in, loa, các ổ đĩa ngoài,... Với các thiết bị ngoài đó, máy tính đều có khối ghép nối tương ứng, ví dụ, khối ghép nối giữa màn hình và bus máy tính là card màn hình (VGA); khối ghép nối giữa loa và bus máy tính là card sound,... Thông thường, các máy tính thế hệ hiện nay thì các khối ghép nối cho các thiết bị ngoài vì thông dụng này được tích hợp cả trên một bảng mạch chính gọi là Main hay Main Board.

Máy tính không phải là một hệ thống khép kín mà máy tính còn sử dụng rộng rãi trong nhiều lĩnh vực, đặc biệt, sử dụng máy tính trong đo lường và điều khiển cho ta tốc độ nhanh và kết quả rất chính xác. Mỗi một ứng dụng sẽ cần thiết kế một khối ghép nối khác nhau. Tất cả các khả năng đó đều được các nhà sản xuất máy tính lưu tâm tới và họ đã dự trữ rất nhiều các cổng và các khe cắm mở rộng để có thể ghép với bus của máy tính. Đây chính là con đường cho những ai muốn nghiên cứu mở rộng thêm phạm vi ứng dụng của máy tính.

Nội dung môn học này đi vào nghiên cứu các cổng (cổng song song, cổng nối tiếp), các khe cắm mở rộng của máy tính để từ đó thiết kế các khối ghép nối phục vụ mục đích đo lường và điều khiển trong công nghiệp.

1.1.1 Các dạng tin trao đổi của máy tính

- Dạng số (Digital)

Đây là một chuỗi các bit 0,1 được biểu diễn theo các hệ đếm như: Hệ nhị phân, hệ thập lục phân... Các tín hiệu số này có thể ở dạng nối tiếp hoặc song song và mức có thể là RS hoặc TTL.

- Dạng chữ (Text)

Đây chính là dạng biểu diễn của các kí tự dưới dạng số, trên thế giới hiện nay thông dụng nhất là cách biểu diễn theo mã ASCII. Theo cách này, các kí tự được biểu diễn bằng một số các bit 0,1 trên hệ thập lục phân, ví dụ: mã của ký tự A là 41h. Dạng tín hiệu này cũng có thể coi là tín hiệu số.

- Dạng tương tự (Analog)

Đây là các dòng điện hay điện áp biến đổi liên tục theo thời gian. Điển hình là đại lượng vật lý thu thập được từ các bộ cảm biến (sensor). Muốn xử lí được dạng tin này, máy tính (khối ghép nối) phải chuyển sang dạng số bằng các bộ ADC.

- Dạng âm tần

Đây là dạng tổ hợp của nhiều tín hiệu tương tự với các tần số và biên độ khác nhau. Cũng có thể coi đây là một dạng của tín hiệu tương tự.

1.1.2 Các loại thông tin trao đổi của máy tính

Trong quá trình gửi tin từ thiết bị ngoài vào máy tính có hai loại thông tin sau:

- *Tin về trạng thái của thiết bị ngoài*
- *Tin mang dữ liệu cần trao đổi*

Trong quá trình ngược lại:

- *Tin về địa chỉ:* Đây chính là địa chỉ của các thanh ghi đệm nằm trong khối ghép nối, ví dụ: 3F8h là địa chỉ thanh ghi đệm đọc/viết ở cổng nối tiếp (RS232).
- *Tin về dữ liệu trao đổi.*
- *Tin mang lệnh điều khiển.*

1.1.3 Các phương thức trao đổi tin của máy tính

Máy tính có thể trao đổi với thiết bị ngoài theo hai phương thức:

- Trao đổi theo chương trình.
- Trao đổi trực tiếp với khối nhớ (Direct Memory Access - DMA).

a. Chế độ trao đổi tin theo chương trình

Đây là chế độ trao đổi tin trong đó máy tính trao đổi với thiết bị ngoài bằng các lệnh vào ra, các lệnh dịch chuyển dữ liệu giữa các thanh ghi, cụ thể như sau:

- Trong ngôn ngữ Assembly các lệnh sau được dành cho trao đổi: IN, OUT, MOV.
- Trong ngôn ngữ Pascal:

+ Đọc một byte dữ liệu: $x := \text{port} [\text{địa chỉ}]$;

+ Đưa ra 1 byte dữ liệu: $\text{port} [\text{địa chỉ}] := y$;

(y là byte dữ liệu đưa ra, x chứa byte dữ liệu được đọc vào).

- Trong ngôn ngữ C:

+ Đọc 1 byte dữ liệu: $x = \text{inport}[\text{địa chỉ}]$;

+ Đưa 1 byte dữ liệu: $\text{outport}(\text{địa chỉ}, y)$;

(x là byte dữ liệu được đọc vào, y là byte dữ liệu xuất ra).

- Trong ngôn ngữ VB: Nếu sử dụng thư viện liên kết động Inpout32.dll

+ Đọc 1 byte dữ liệu : $x = \text{Inp}(\text{địa chỉ})$

+ Đưa 1 byte dữ liệu: $\text{Out}(\text{địa chỉ}) = y$

(x là byte dữ liệu được đọc vào, y là byte dữ liệu xuất ra).

Trong chế độ trao đổi theo chương trình có 3 phương pháp:

- Phương pháp trao đổi đồng bộ

Ở phương pháp này, máy tính sẽ tiến hành trao đổi tin ngay với thiết bị ngoài khi khởi động xong mà không cần biết trạng thái của đường dây cũng như thiết bị ngoài.

Để có thể thực hiện được phương pháp này thì yêu cầu:

- + Tốc độ trao đổi tin của thiết bị ngoài lớn hơn hoặc bằng tốc độ trao đổi tin của máy tính.
- + Thiết bị ngoài phải ở trạng thái sẵn sàng ngay khi máy tính khởi động xong.
- + Phương pháp này có ưu điểm là tốc độ trao đổi tin nhanh nhưng nhược điểm là dễ bị mất tin khi thiết bị ngoài chưa ở trạng thái sẵn sàng.

- *Phương pháp không đồng bộ*

Trong phương pháp này, trước khi trao đổi tin, máy tính tiến hành đọc, kiểm tra trạng thái của thiết bị ngoài, nếu thiết bị ngoài đã sẵn sàng thì sẽ tiến hành trao đổi tin còn ngược lại sẽ chờ.

Ngoài ra trong quá trình trao đổi, nếu tin bị lỗi cũng yêu cầu phía phát phải truyền lại.

Phương pháp này có độ tin cậy cao nhưng tốc độ chậm hơn phương pháp đồng bộ.

- *Phương pháp trao đổi theo ngắt chương trình*

Phương pháp này lợi dụng được ưu điểm, khắc phục được nhược điểm của hai phương pháp trên. Trình tự tiến hành như sau:

- Khi thiết bị ngoài có yêu cầu trao đổi sẽ gửi tín hiệu yêu cầu (ngắt) đến máy tính.
- Máy tính dừng chương trình đang phục vụ (nếu thiết bị ngoài đang yêu cầu có mức ưu tiên cao hơn) và nhớ lại điểm dừng đồng thời gửi tín hiệu xác nhận, yêu cầu thiết bị ngoài trao đổi tin.
- Máy tính và thiết bị ngoài trao đổi tin theo chương trình (gọi là chương trình con phục vụ ngắt).
- Kết thúc trao đổi, máy tính trở lại chương trình chính từ điểm dừng.

Phương pháp trao đổi theo ngắt chương trình khắc phục được nhược điểm của hai phương pháp đồng bộ và không đồng bộ, nó cho phép tận dụng được tối đa thời gian làm việc của máy tính.

b. Trao đổi DMA

Đây là phương thức trao đổi trực tiếp với khối nhớ của máy tính mà không thông qua CPU. Khi đó, CPU sẽ ở trạng thái treo, nhường quyền điều khiển BUS cho khối ghép nối. Thiết bị ngoài và khối nhớ của máy tính sẽ tiến hành trao đổi (đọc/ghi dữ liệu), sau khi quá trình kết thúc sẽ nhường lại quyền điều khiển BUS cho CPU.

1.2 Vai trò, nhiệm vụ và cấu trúc của khối ghép nối

1.2.1 Vai trò

Trong quá trình trao đổi giữa máy tính và thiết bị ngoài, khối ghép nối giữ vai trò trung chuyển tin. Trung chuyển ở đây có nghĩa tích cực vì trong quá trình nhận tin từ thiết bị ngoài vào máy tính, khối ghép nối nhận tin từ thiết bị ngoài, xử lý và gửi cho máy tính theo khuôn dạng tin, tốc độ thích hợp thích hợp. Ngược lại, trong quá trình gửi tin từ máy ra thiết bị ngoài, khối ghép nối nhận tin từ máy tính, xử lý và gửi cho thiết bị ngoài theo dạng phù hợp với thiết bị ngoài tương ứng.

1.2.2 Nhiệm vụ

Để đáp ứng được các vai trò trên, đòi hỏi khối ghép nối phải thực hiện các nhiệm vụ sau:

a. Phối hợp về mức và công suất của tín hiệu

Mức tín hiệu của các đường dây máy tính là mức TTL (nằm trong khoảng 0V-5V), công suất thường rất nhỏ trong khi mức tín hiệu của thiết bị ngoài rất đa dạng và công suất thường lớn hơn do vậy yêu cầu khối ghép nối phải có khả năng phối hợp mức và công suất của tín hiệu. Để thực hiện chức năng này, khối ghép nối thường chứa các bộ chuyển đổi mức, các bộ khuếch đại, phối hợp công suất.

b. Phối hợp về dạng tín

Tín hiệu ở đường dây máy tính là tín hiệu số ở dạng song song trong khi tín hiệu của thiết bị ngoài có thể là tín hiệu số, tương tự có thể ở dạng nối tiếp, song song...có thể ở dạng mã khác. Vì vậy, khối ghép nối phải có nhiệm vụ biến đổi tương thích khuôn dạng tín hiệu giữa thiết bị ngoài và máy tính. Các bộ biến đổi số/tương tự,

tương tự/số; các bộ chuyên đổi nối tiếp/song song song song/nối tiếp trong khối ghép nối sẽ thực hiện nhiệm vụ này.

c. Phối hợp về tốc độ trao đổi tin

Tốc độ trao đổi tin của máy tính lớn hơn nhiều lần so với tốc độ trao đổi tin của thiết bị ngoài vì vậy khối ghép nối thường phải nhận tin theo xung nhịp thiết bị ngoài và phát tin theo xung nhịp của máy tính. Để thực hiện được nhiệm vụ này, khối ghép nối thường có các bộ nhớ đệm.

d. Phối hợp về phương thức trao đổi tin

Một khối ghép nối đôi khi là cả một hệ thống nhỏ, ở đó cũng có cả phần mềm thậm chí cả hệ điều hành. Một khối ghép nối như vậy đương nhiên có thể phối hợp với máy tính trong phương pháp trao đổi tin theo chương trình cũng như độc lập hoạt động trong phương pháp trao đổi DMA.

Ngoài những nhiệm vụ trên, khối ghép nối còn có khả năng phối hợp về trở kháng, cảm kháng, dung kháng... giữa các mạch điện tử của máy tính và thiết bị ngoài.

1.3 Cấu trúc chung của khối ghép nối

Hình 1.1 mô tả cấu trúc chung của một khối ghép nối.

1.3.1 Khối phối hợp đường dây

Khối này có nhiệm vụ:

- Phối hợp về mức, công suất, khuôn dạng tín hiệu của đường dây máy tính với đường dây thiết bị ngoài. Khối này thường chứa các bộ chuyển mức, chuyển mạch, khuếch đại công suất, ADC, DAC...
- Cô lập đường dây máy tính khi không có trao đổi tin (trạng thái điện trở cao)
- Điều khiển đưa tin ra, vào máy tính.

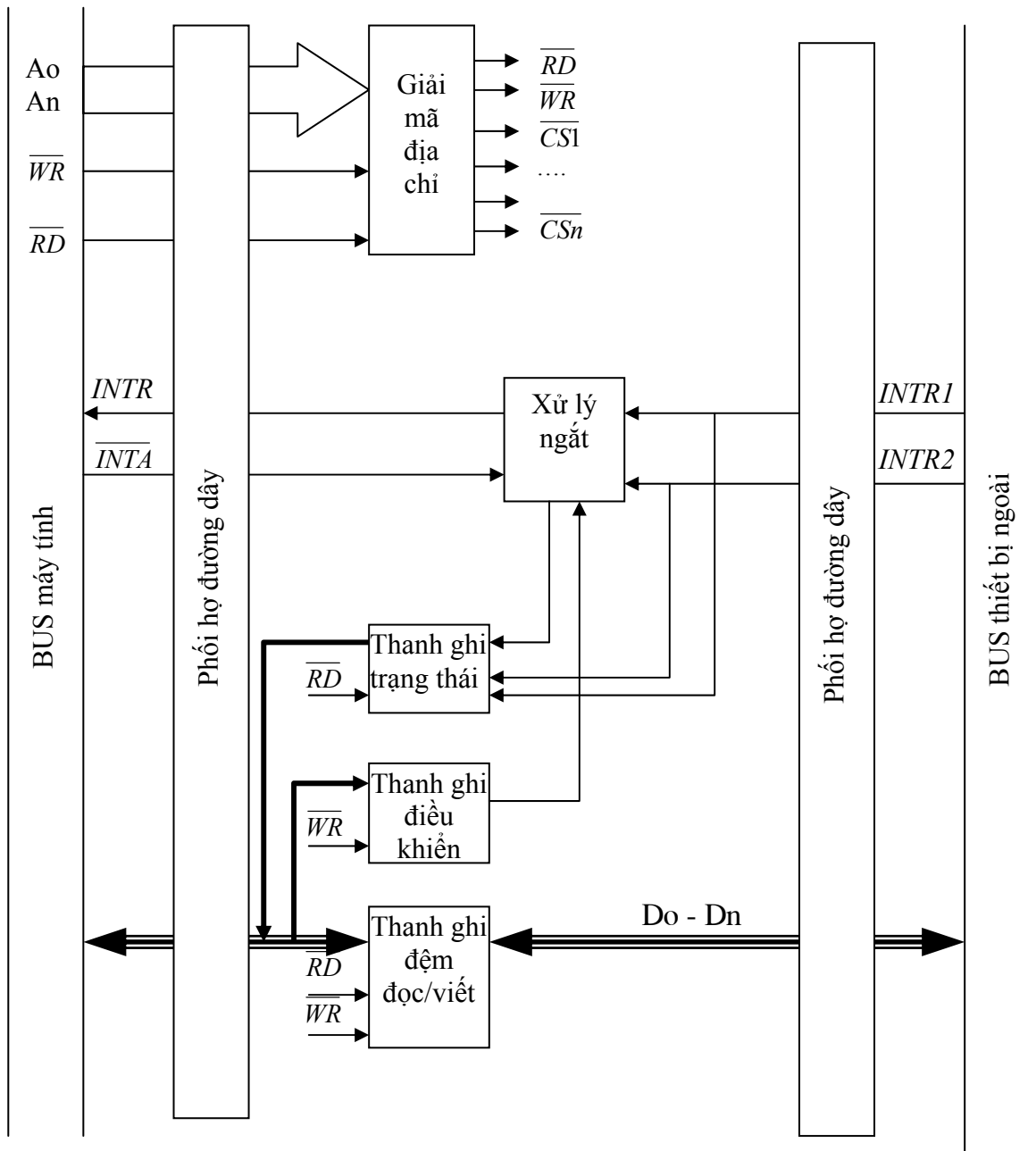
1.3.2 Khối giải mã địa chỉ - lệnh

Khối này làm nhiệm vụ giải mã địa chỉ cho các thanh ghi bên trong khối ghép nối, các địa chỉ này sẽ tùy thuộc vào các lệnh mà khối ghép nối nhận để thực hiện. Kết quả mà khối này thực hiện sẽ là các xung cho phép đọc/ghi đối với từng thanh ghi của khối ghép nối.

1.3.3 Khối xử lý ngắt

Trong chế độ trao đổi tin theo ngắt chương trình, khối này giữ vai trò tiếp nhận các yêu cầu ngắt từ thiết bị ngoài, sắp xếp chúng theo thứ tự ưu tiên nhất định và thông báo về CPU lần lượt từng yêu cầu ngắt được ưu tiên phục vụ. Khi được CPU thông báo chấp nhận ngắt, khối này cũng nhận các thông báo đó, gửi ngược trở lại cho thiết bị ngoài.

Ngoài các khối chính trên, mỗi khối ghép nối còn có khối điều khiển điều khiển toàn bộ hệ thống, khối phát xung nhịp đồng bộ hệ thống.



Hình 1.1 Cấu trúc chung của một khối ghép nối

1.4 Bài tập cuối chương

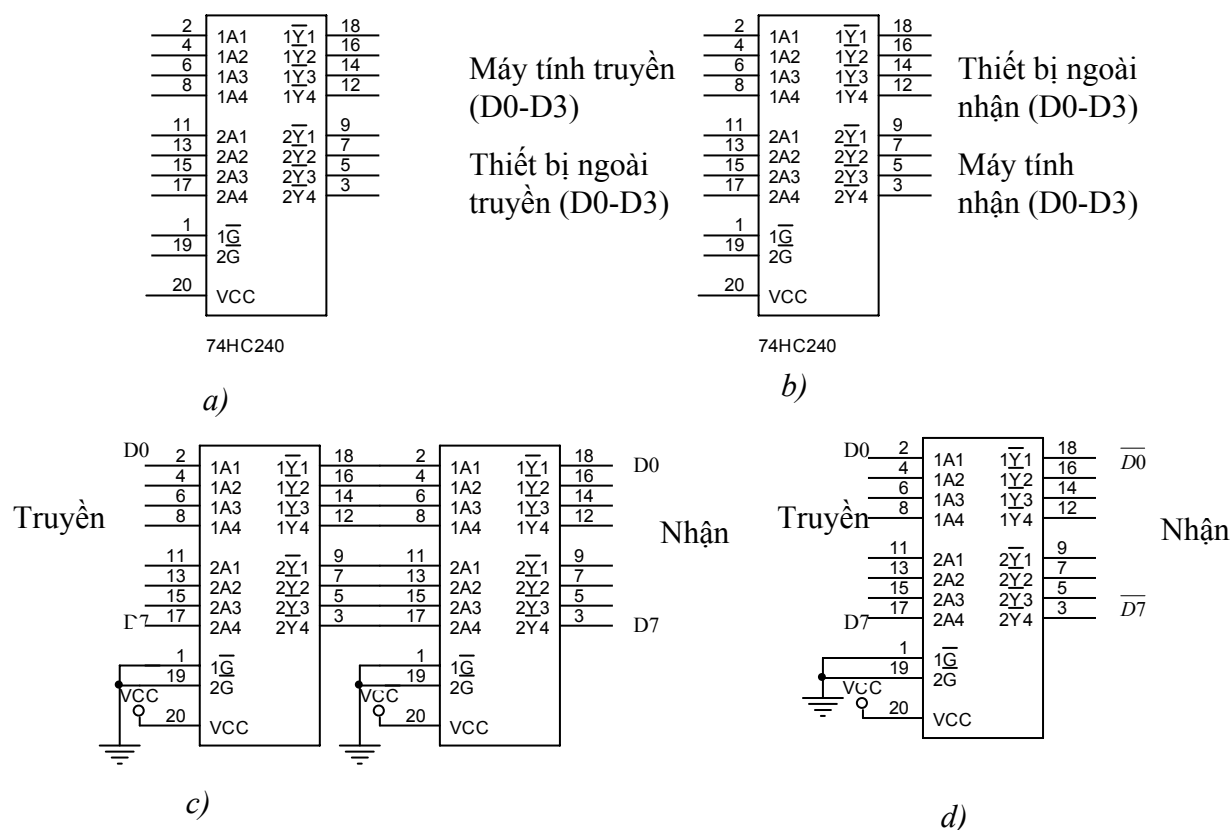
1. Hãy nêu cấu trúc chung của một máy tính và nhiệm vụ chức năng của từng khối
2. Hãy nêu các dạng tin trao đổi của máy tính.
3. Hãy nêu cấu trúc một khối ghép nối và chức năng của từng khối.
4. Hãy đổi các số sau ở hệ mười sang hệ mười sáu: 1029; 65530; 540.
5. Hãy đổi các số sau sang hệ hai và hệ mười: 005EH; 12A0H; 0FFFEH.

Chương 2 Ghép nối trao đổi tín hiệu song song theo chương trình

Chương này sẽ giới thiệu về cấu trúc của cổng song song và đưa ra một số các mạch thiết kế đơn giản ghép nối qua cổng song song ứng dụng đo lường và điều khiển. Trong các mạch thiết kế, cần phải sử dụng các bộ đệm để không làm suy hao tín hiệu và các bộ chốt để tách các tín hiệu địa chỉ.

2.1 Các vi mạch đệm, chốt song song thông dụng

2.1.1 Vi mạch 74HC240

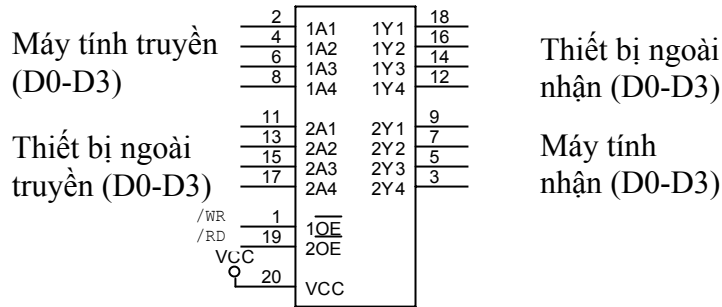
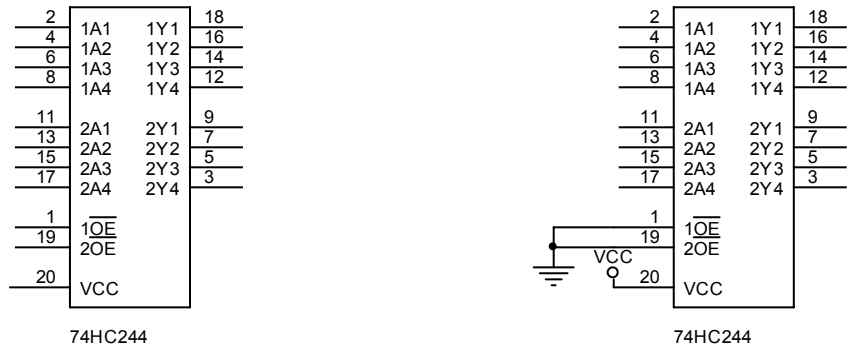


Hình 2.1 Sơ đồ nguyên lý và các ứng dụng của '240

Vi mạch '240 chứa 8 bộ chốt 3 trạng thái với các đầu vào A và đầu ra \bar{Y} . Các bộ này được điều khiển bởi $\overline{1G}$ và $\overline{2G}$. Vi mạch này có thể dùng làm bộ truyền nhận 4 bit có đảo (Hình 2.1.b); Bộ đệm 8 bit có đảo và không đảo (Hình 2.1.d,c).

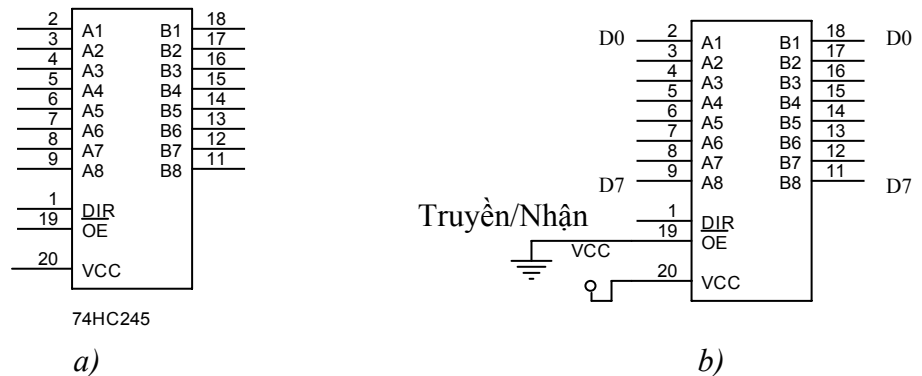
2.1.2 Vi mạch 74HC244

Khác với '240, '244 chứa 8 chốt 3 trạng thái "không đảo", cách điều khiển hoàn toàn như nhau. Các ứng dụng có thể nêu ra là: Bộ đệm 8 bit (hình 2.2.b), bộ truyền nhận 4 bit (hình 2.2.c)...



Hình 2.2 Sơ đồ nguyên lý và các ứng dụng của '244

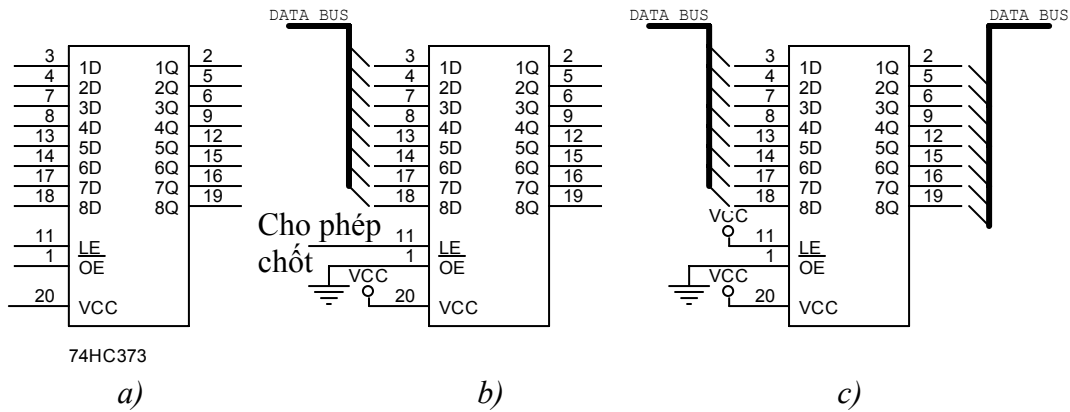
2.1.3 Vi mạch 74HC245



Hình 2.3 Sơ đồ nguyên lý và ứng dụng của '245

Vi mạch '245 chứa 16 chốt 3 trạng thái tạo thành 8 cặp (hình 2.3.a). Các chốt này được cho phép bởi \overline{OE} và điều khiển bởi DIR. Khi DIR=1, tín hiệu được phép truyền từ các đầu vào A sang B và ngược lại. Ứng dụng điển hình của vi mạch này là bộ truyền nhận 8 bit song song (hình 2.3.b).

2.1.4 Vi mạch 74HC373

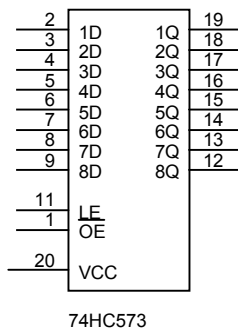


Hình 2.4 Sơ đồ nguyên lý và các ứng dụng của '373

Vi mạch '373 chứa 8 D-FF và 8 cổng 3 trạng thái được điều khiển bởi các chân LE(11) và \overline{OE} (1). Các D-FF này có thể hoạt động ở chế độ chốt (Latch), chốt các bit dữ liệu ở BUS dữ liệu (khi LE chuyển trạng thái từ 1 sang 0), tạo thành thanh ghi 8 bit (Hình b). Ngoài ra nếu để các D-FF này hoạt động như một FF thường thì '373 lại trở thành bộ đệm BUS 8 bit (Hình c).

2.1.5 Vi mạch 74HC573

'573 có cấu tạo hoàn toàn giống '373 chỉ khác ở cách bố trí chân. Cách bố trí chân của '573 thuận tiện hơn nhiều so với '373 trong việc thiết kế mạch in.



Hình 2.5 Sơ đồ nguyên lý của '573

2.2 Cổng song song

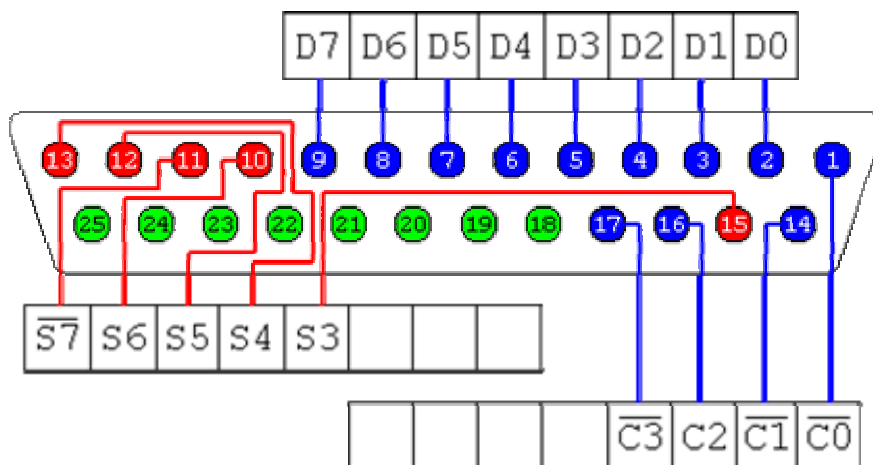
2.2.1 Giới thiệu

Cổng song song được thiết kế đầu tiên bởi công ty Centronics nhằm mục đích ghép nối máy tính với máy in. Sau này cổng này được tiêu chuẩn hoá và có mặt ở hầu hết các máy tính. Tên gọi của cổng song song bắt nguồn từ kiểu truyền dữ liệu qua

cổng này, các bit truyền song song còn các byte truyền nối tiếp. Ngoài tên gọi này ra cổng song song còn có tên gọi là cổng LPT hoặc cổng Centronics.

Cấu trúc của cổng song song gồm 8 đường dữ liệu, bốn đường dẫn điều khiển và năm đường dẫn trạng thái. Các đường dẫn này đều tương thích mức TTL(0;5V) do vậy khá thuận tiện, đơn giản cho việc ghép nối vì nhiều linh kiện, mạch điện tương thích với mức logic trên. Sử dụng cổng song song trong đo lường và điều khiển tương đối đơn giản. Khoảng cách truyền của cổng song song bị hạn chế do điện dung ký sinh, hiện tượng cảm ứng và bị suy giảm công suất. Khoảng cách này bị giới hạn trong khoảng 2m. Nếu cần ghép nối ở khoảng cách xa hơn cần có các bộ đệm, các phương pháp làm giảm điện dung ký sinh, hiện tượng cảm ứng (chẳng hạn kẹp mass giữa các đường tín hiệu). Nếu muốn có khoảng cách xa hơn nữa, nên chọn giải pháp ghép nối trao đổi tin nối tiếp vì truyền dữ liệu qua cổng song song với khoảng cách xa sẽ tăng khả năng gây lỗi dữ liệu được truyền mà còn tăng chi phí. Tốc độ truyền qua cổng song song có thể đạt đến giá trị 1 Mbit/s

Trong các máy tính thế hệ cũ, cổng song song có tới 36 chân nhưng ngày nay để giảm chi phí, người ta đã chuẩn hoá thành 25 chân, trong số 25 chân này chỉ có 18 chân có ý nghĩa, các chân còn lại đều là các chân nối mass. Dưới đây là sơ đồ giao diện cổng song song trên máy tính:



Hình 2.6 Giao diện cổng song song trên máy tính PC

Ký hiệu và ý nghĩa của các chân cắm trên cổng song song khi kết nối với máy in như bảng 2.1:

Bảng 2.1 Sắp xếp các chân tín hiệu trên ổ cắm

25 Chân	36 Chân	Ký hiệu	Vào/ra(I/O)	Mô tả
1	1	\overline{STROBE}	Ra	Tín hiệu thông báo có 1byte sẵn sàng được in
2-9	2-9	D0-D7	Ra	Các đường dữ liệu từ D0-D7
10	10	\overline{ACK} (Acknowledge)	Vào	Tín hiệu xác nhận đã nhận được 1byte của máy in đối với máy tính
11	11	BUSY	Vào	Tín hiệu báo bận của máy in
12	12	PE (Paper Empty)	Vào	Tín hiệu báo hết giấy của máy in
13	13	SLCT (Select)	Vào	Tín hiệu báo trạng thái sẵn sàng của máy in
14	14	\overline{AF} (Auto Linefeed)	Ra	Tín hiệu yêu cầu nạp một dòng mới của máy tính đối với máy in
15	32	\overline{ERROR}	Vào	Tín hiệu thông báo lỗi của máy in đối với máy tính
16	31	INIT(RESET)	Ra	Tín hiệu khởi động lại của máy tính đối với máy in.
17	36	SLCTIN (Select Input)	Ra	Tín hiệu lựa chọn máy in của máy tính
18-25	19-30, 33	GND		Tín hiệu nối mass
	16			Tín hiệu nối mass
	17			Tín hiệu nối mass
	18			+5V
	34,35			Không sử dụng

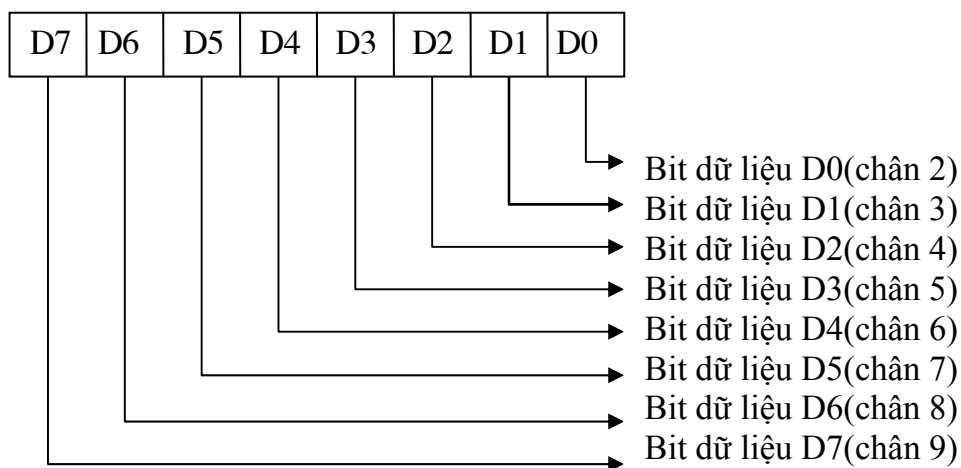
Các đường dẫn tín hiệu trên cổng song song được chia thành 3 nhóm:

- Các đường dẫn tín hiệu xuất ra từ máy tính và điều khiển máy in được gọi là các đường dẫn điều khiển.

- Các đường dẫn tín hiệu đưa các thông báo ngược lại từ máy in về máy tính được gọi là các *đường dẫn trạng thái*.
- Các đường dẫn xuất ra từ máy tính để truyền các byte ký tự cần in là *các đường dẫn dữ liệu*.

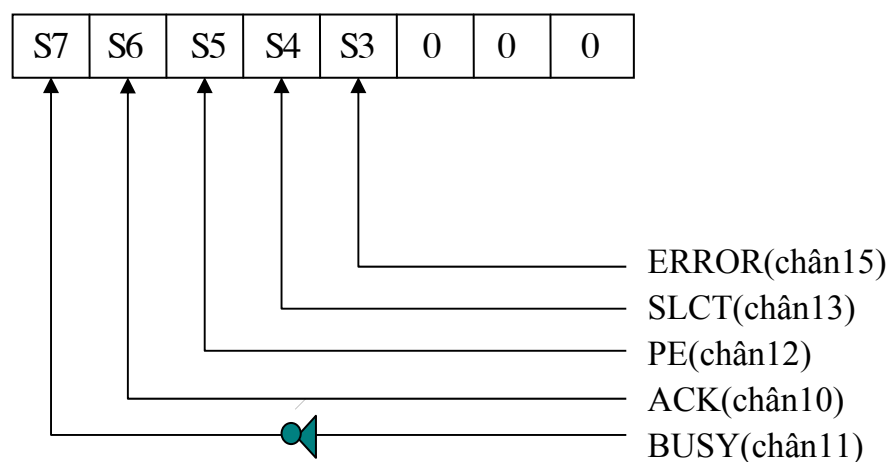
Các nhóm đường dẫn tín hiệu trên liên quan tới các thanh ghi bên trong cổng song song. Để có thể ghép nối các thiết bị ngoại vi hay các mạch điện ứng dụng trong đo lường và điều khiển với cổng song song ta phải tìm hiểu cách trao đổi với các thanh ghi thông qua cách sắp xếp đường dẫn và địa chỉ của các thanh ghi và các phần mềm liên quan. Có ba thanh ghi bên trong cổng song song:

- *Thanh ghi dữ liệu (có địa chỉ cơ sở)*



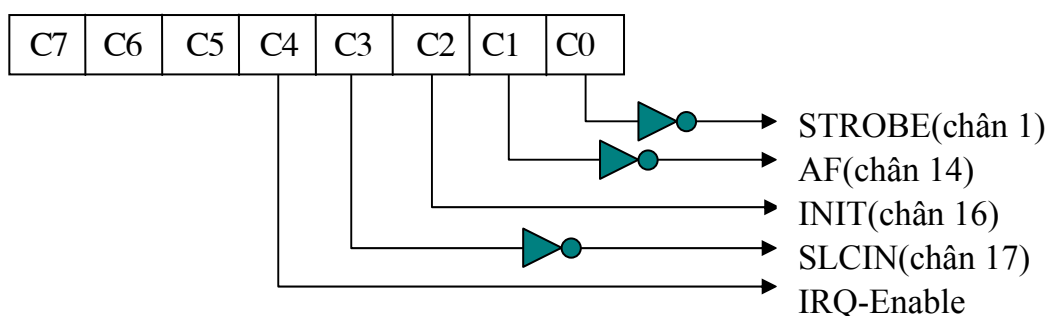
Hình 2.7 Kết nối các chân trên cổng song song và thanh ghi dữ liệu

- *Thanh ghi trạng thái (có địa chỉ cơ sở + 1)*



Hình 2.8 Kết nối các chân trên cổng song song và thanh ghi trạng thái

- Thanh ghi điều khiển (có địa chỉ có số + 2)



Hình 2.9 Kết nối các chân trên cổng song song và thanh ghi điều khiển

Qua cấu trúc các thanh ghi ta thấy có 8 đường dẫn dữ liệu dẫn tới 8 bit nhớ trên thanh ghi dữ liệu và 4 đường dẫn điều khiển Strobe, Auto Linefeed, Reset, Select Input dẫn tới bốn ô nhớ trên thanh ghi điều khiển, cuối cùng là 5 đường dẫn trạng thái Acknowledge, Busy, Paper Empty, Select, Error nối tới 5 ô nhớ trên thanh ghi trạng thái. Riêng ở thanh ghi điều khiển cần chú ý tới bit IRQ-Enable được sử dụng cho mục đích ghép nối nhưng lại không được nối với ổ cắm 25 chân, bit này có thể được sử dụng để xoá một ngắt có liên quan với đường dẫn Acknowledge.

Hệ điều hành DOS dự tính đến bốn cổng song song đặt tên là: LPT1, LPT2, LPT3 và LPT4. Tuy nhiên, hầu hết các máy vi tính chỉ có một cổng song song thậm chí không có cổng song song vì lý do kinh tế. Khi bật máy, BIOS sẽ kiểm tra xem trên máy tính có trang bị mấy cổng song song. Các cổng song song được BIOS tìm thấy sẽ được sắp theo các tên lần lượt là LPT1, LPT2... Phần lớn các phiên bản của BIOS chạy trong giai đoạn khởi động (Boot phase) của máy tính, khi đó, các thông số về phần cứng cũng như các cổng song song tìm thấy sẽ hiển thị trong một khung hình chữ nhật. Ta có thể dừng lại quá trình khởi động của máy tính bằng phím <Pause> để quan sát kỹ các thông số được liệt kê trong bảng 2.2.

Bảng 2.2 Các địa chỉ của cổng song song trên máy tính PC

Cổng song song (LPT)	Địa chỉ thanh ghi dữ liệu	Địa chỉ thanh ghi trạng thái	Địa chỉ thanh ghi điều khiển
LT1	3BCh	3BDh	3BEh
LPT2	<u>378h</u>	<u>379h</u>	37Ah
LPT3	278h	279h	27Ah
LPT4	2BCh	2BDH	2BEh

Thông qua địa chỉ, ta có thể trao đổi bằng phần mềm với các thanh ghi của cổng song song bằng cách thiết lập cho các chân của cổng lên mức cao (High) hoặc xuống mức thấp (Low).

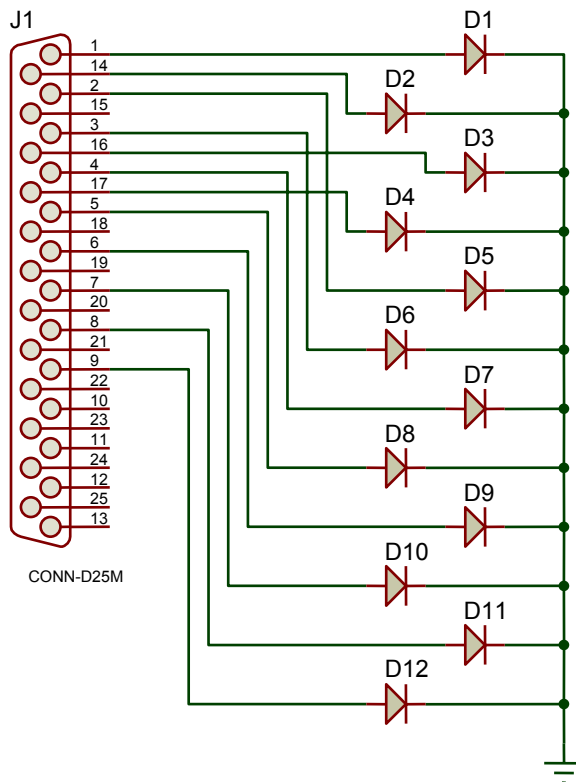
Đối với cổng song song, chiều của các thanh ghi chỉ là một hướng. Thanh ghi điều khiển có chiều ra, thanh ghi trạng thái có chiều vào, riêng thanh ghi dữ liệu với một số máy có thể có hai hướng nhưng tính hai hướng bị hạn chế.

2.2.2 Giao diện một hướng ở cổng song song

Giao diện một hướng (hướng từ máy tính đưa ra) sử dụng các chân tín hiệu của thanh ghi dữ liệu và thanh ghi điều khiển, các thanh ghi này đều có chiều ra. Nếu sử dụng giao diện một hướng tối đa sẽ có 12 chân tín hiệu ra độc lập.

Để sử dụng giao diện này trong các ứng dụng điều khiển, trước tiên phải chắc chắn xem các chân tín hiệu trên hai đường dẫn điều khiển và đường dẫn dữ liệu còn tốt hay không, sử dụng một phần cứng và một phần mềm đơn giản như sau để kiểm tra:

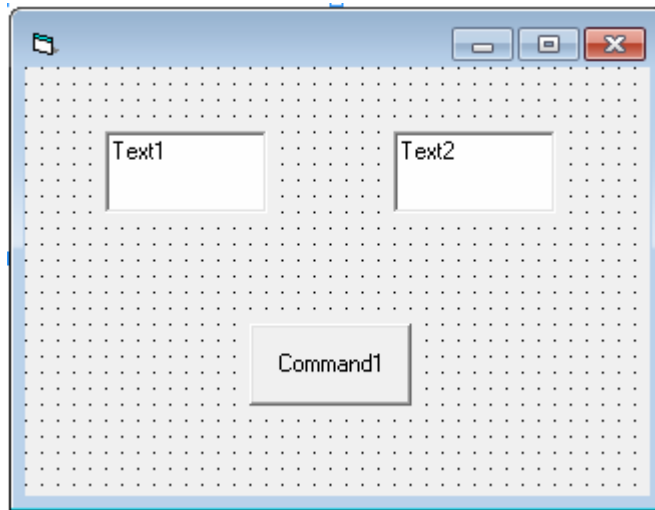
Ghép nối các Led đơn với các chân tín hiệu trên đường dẫn dữ liệu và điều khiển



Hình 2.10 Sơ đồ nguyên lý mạch kiểm tra giao diện một hướng của cổng song song

Chương trình điều khiển các Led sáng tắt viết bằng ngôn ngữ Visual Basic (VB) làm theo các bước sau:

Bước 1: Xây dựng giao diện.



Hình 2.11 Giao diện chương trình vào ra trên cổng song song

Bước 2: Viết chương trình.

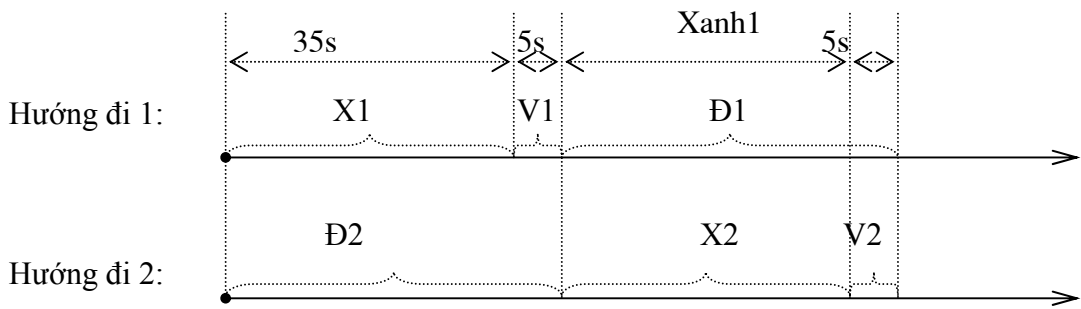
```
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _  
    (ByVal PortAddress As Integer, ByVal Value As Integer) _  
    Dim t As Integer  
-----  
Private Sub Command1_Click()  
    Out &H378, Val(Text1.Text)  
    Out &H37A, Val(Text2.Text)  
End Sub
```

Khi chạy đoạn chương trình trên cho tùy theo giá trị nhập vào text1 và text2 các Led sẽ sáng hoặc tắt khi ta nhấn command1. Ví dụ, muốn tắt cả các Led sáng hết khi nhấn command1 thì phải nhập giá trị 255 (11111111B) vào ô text1 và giá trị 4 (00000100) vào ô text2, muốn các Led tắt hết nhập giá trị 0(00000000B) vào ô text1 và 11 (00001011B) vào ô text2. Các bit C0, C1, C3 đảo trạng thái trước khi đưa ra các chân 1, 14, 17.

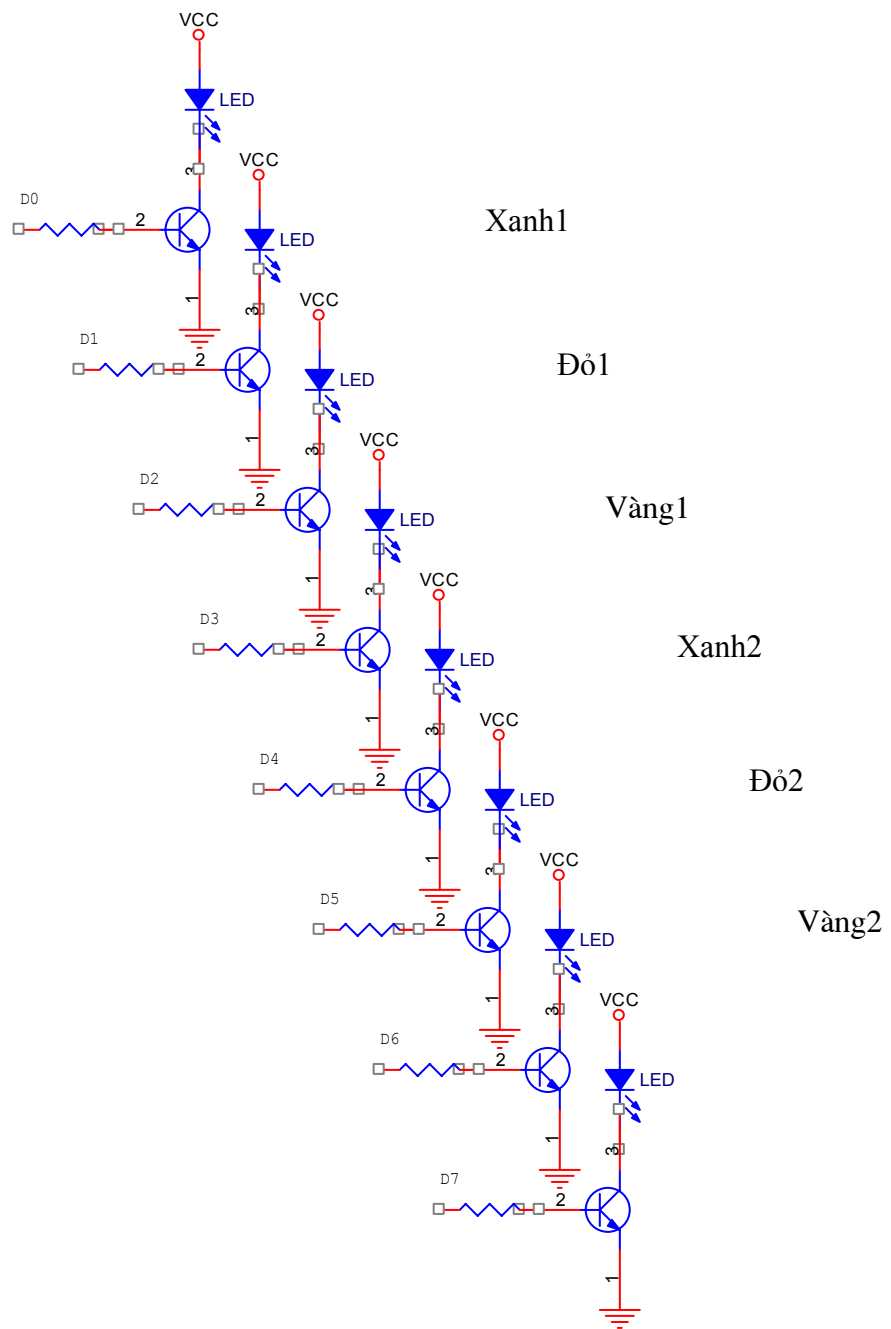
Chú ý: Để chạy được chương trình trên file inpout32.dll phải được copy vào ổ đĩa C theo đường dẫn: C:\Windows\System32.

Đoạn chương trình viết bằng ngôn ngữ VB ở trên có thể khẳng định được các chân tín hiệu thuộc thanh ghi dữ liệu và thanh ghi điều khiển của cổng song song hoạt động tốt. Dưới đây là các ứng dụng điều khiển dùng cổng song song.

Ví dụ 1: Thiết kế mô hình điều khiển đèn giao thông theo thời gian thực bằng cổng song song của máy tính PC.



Hình 2.12 Giản đồ thời gian thực của hệ thống điều khiển đèn giao thông ở chế độ giờ cao điểm



Hình 2.13 Sơ đồ nguyên lý mạch điều khiển đèn giao thông

Hệ thống đèn giao thông trong ví dụ này được thiết kế cho ngã tư có các đèn xanh1, vàng1, đỏ1, xanh2, vàng2, đỏ2 trên hai hướng. Chế độ giờ cao điểm đèn xanh sáng trong 35 giây, đèn vàng sáng 5 giây và đèn đỏ sáng 40 giây, chế độ nửa đêm đèn vàng nhấp nháy 2 giây.

Hình 2.11 là giản đồ thời gian thực mô tả trạng thái các đèn sáng ở chế độ giờ cao điểm, thiết kế mạch ghép nối hình 2.12 và dưới đây là chương trình điều khiển:

Thiết kế giao diện



Hình 2.14 Giao diện chương trình điều khiển đèn giao thông

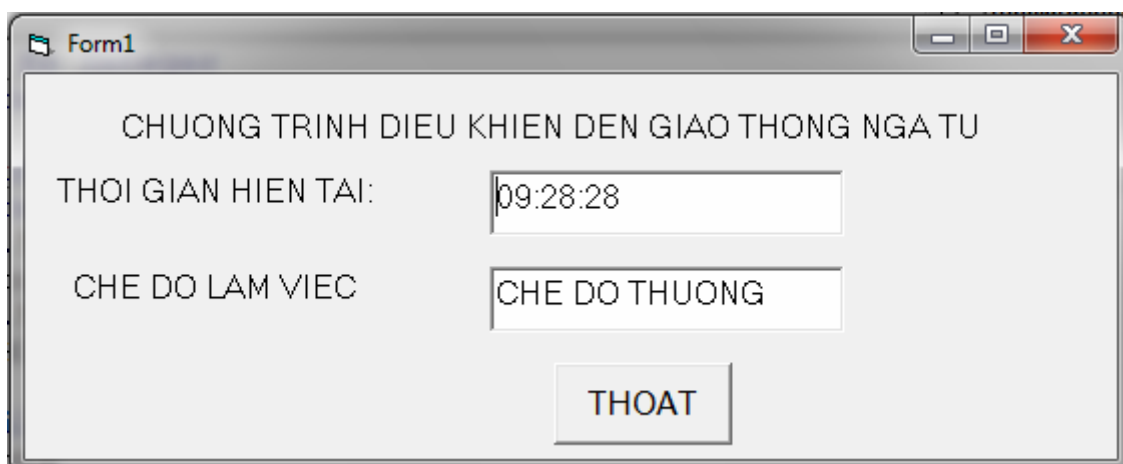
Viết chương trình

```
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
    (ByVal PortAddress As Integer, ByVal Value As Integer)
Dim t As Integer
Private Sub Command1_Click()
End
End Sub
Private Sub Timer1_Timer()
Text1.Text = Format(Now, "hh:nn:ss")
If (Hour(Text1.Text) >= 7 And Hour(Text1.Text) < 21) Then
Text2.Text = "CHE DO THUONG"
Out &H378, &HEE '11101110B
t = 0
Do While t < 35
```

```

DoEvents
Loop
Out &H378, &HEB '11101011B
t = 0
Do While t < 5
DoEvents
Loop
Out &H378, &HF5 '11110101B
t = 0
Do While t < 35
DoEvents
Loop
Out &H378, &HDD '11011101B
t = 0
Do While t < 5
DoEvents
Loop
Else
Text2.Text = "CHE DO BAN DEM"
Out &H378, &HDB '11111011B
t = 0
Do While t < 2
DoEvents
Loop
Out &H378, &HDD '11011101B
t = 0
Do While t < 2
DoEvents
Loop
End If
End Sub
Private Sub Timer2_Timer()
t = t + 1
End Sub

```



Hình 2.15 Giao diện khi chạy chương trình

Ví dụ 2: Thiết kế mạch điều khiển mô tơ bước qua công song song của máy tính.

Có 3 phương pháp điều khiển motor bước:

+ Điều khiển 1 pha:

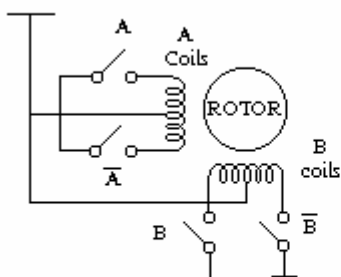
Ở phương pháp này các bước của motor sẽ từ : $0^{\circ}, 1.8^{\circ}, 3.6^{\circ}, \dots 258.2^{\circ}$. Motor sẽ quay 200 bước/vòng. Các bước của motor được thiết lập theo bảng I.

+ Điều khiển 2 pha:

Ở phương pháp này các bước motor sẽ là: $0.9^{\circ}, 2.7^{\circ}, 4.5^{\circ}, \dots 359.1^{\circ}$. Motor sẽ quay 200 bước/vòng. Các bước của motor được thiết lập theo bảng II.

+ Điều khiển hỗn hợp một pha và hai pha:

Với phương pháp này các bước của motor sẽ là: $0^{\circ}, 0.9^{\circ}, 1.8^{\circ}, 2.7^{\circ}, 3.6^{\circ}, 4.5^{\circ}, \dots 359.1^{\circ}$. Motor sẽ quay 400 bước/vòng. Các bước của motor được thiết lập theo bảng III.



	A	B	\bar{A}	\bar{B}
BUỐC 1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1

Bảng I

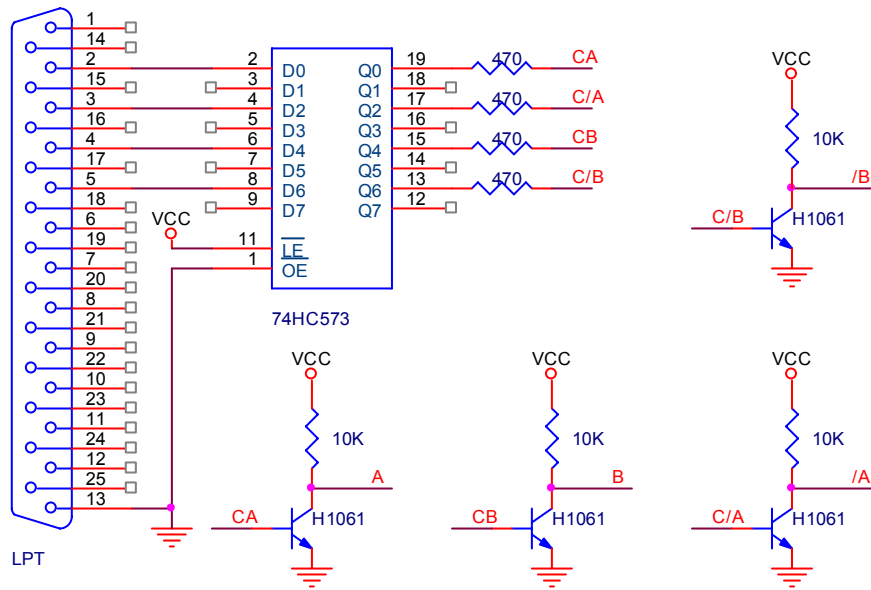
	A	B	\bar{A}	\bar{B}
BUỐC 1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1
5	1	1	0	0
6	0	1	1	0
7	0	0	1	1
8	1	0	0	1

Bảng II

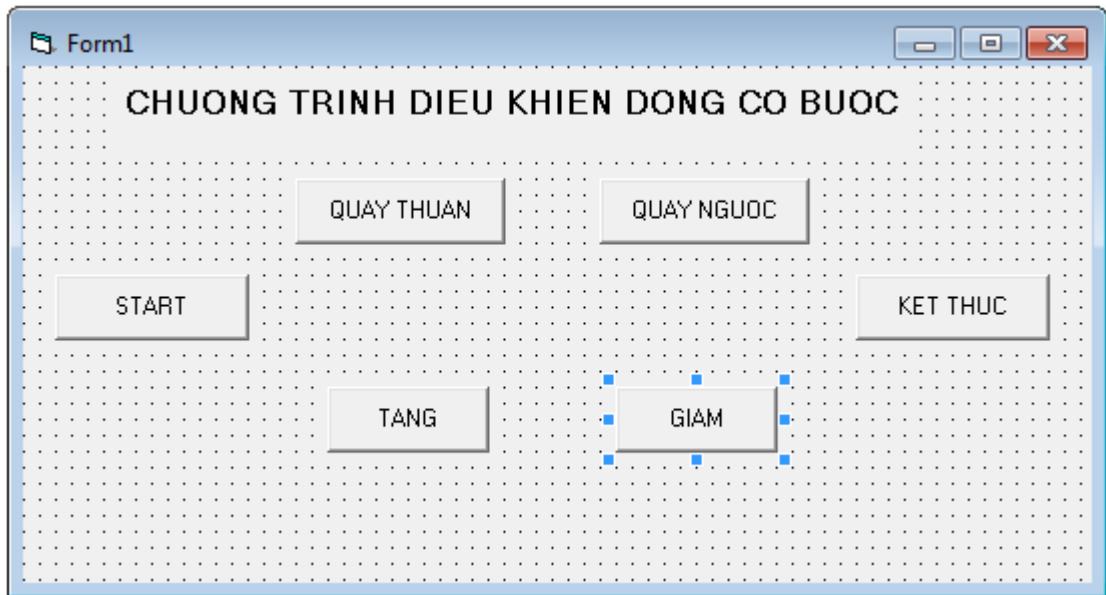
	A	B	\bar{A}	\bar{B}
BUỐC 1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Bảng III

Hình 2.16 Mô tả hoạt động của Motor bước



Hình 2.17 Sơ đồ nguyên lý mạch điều khiển Motor bước bằng cổng song song
 Thiết kế giao diện điều khiển Motor bước như hình 2.18 dưới đây:



Hình 2.18 Giao diện chương trình điều khiển Motor bước

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal Value As Integer)
Dim i As Integer
Dim t As Double
Dim k As Integer
```

```

Sub thuan(t As Double)
i = 1
k = 0
Do While i > 0
Out &H378, 14
TimeDelay (t)
Out &H378, 11
TimeDelay (t)
Out &H378, 13
TimeDelay (t)
Out &H378, 7
TimeDelay (t)
Loop
End Sub

```

```

Sub nguoc(t As Double)
i = 1
k = 1
Do While i > 0
Out &H378, 14
TimeDelay (t)
Out &H378, 11
TimeDelay (t)
Out &H378, 13
TimeDelay (t)
Out &H378, 7
TimeDelay (t)
Loop
End Sub

```

```

Private Sub Command1_Click()
Call thuan(t)
End Sub

```

```

Function TimeDelay(Delay As Double)
Dim PauseTime, START
PauseTime = Delay ' Set duration.
START = Timer ' Set start time.
Do While Timer < START + PauseTime
DoEvents ' Yield to other processes.
Loop
End Function

```

```

Private Sub QUAYNGUOC_Click()
Call nguoc(t)
End Sub

```

```

Private Sub QUAYTHUAN_Click()
Call thuan(t)
End Sub

```

```

Private Sub TANG_Click()
t = t / 1.4
If k = 0 Then
Call thuan(t)
Else

```

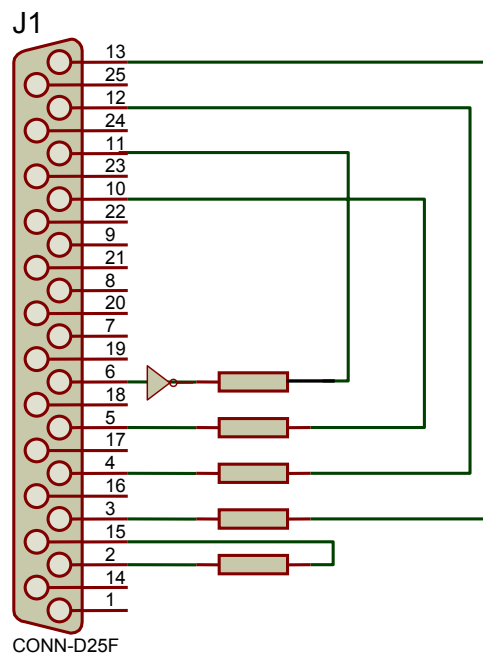
```

Call nguoc(t)
End If
End Sub
-----
Private Sub GIAM_Click()
t = t / 1.4
If k = 0 Then
Call thuan(t)
Else
Call nguoc(t)
End If
End Sub
-----
Private Sub Command5_Click()
End
End Sub
-----
Private Sub START_Click()
t = 0.2
End Sub

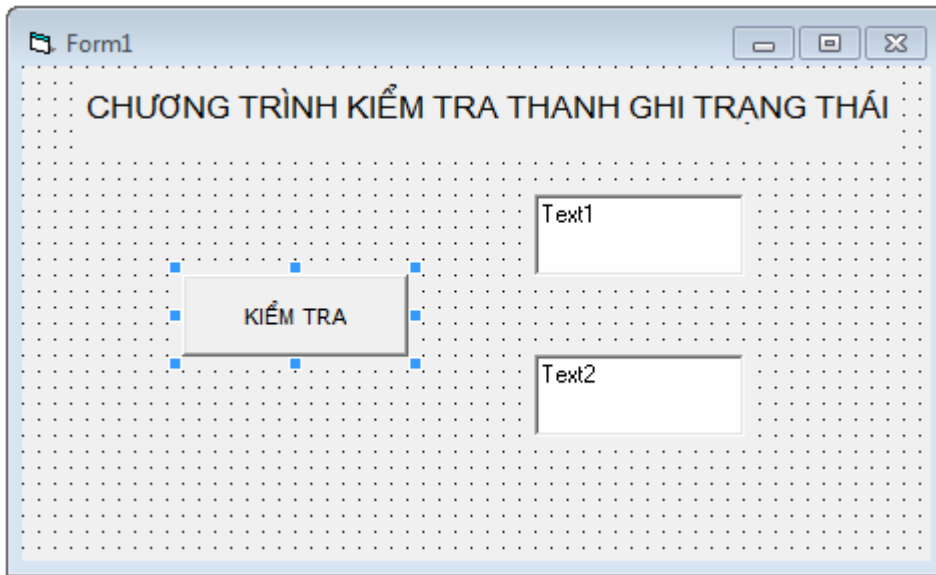
```

2.2.3 Giao diện 2 hướng dùng cổng song song - module vào ra 8 bit

Trước hết để chắc chắn rằng các đường dẫn trạng thái đang hoạt động tốt, chúng ta tiến hành kiểm tra bằng phần cứng gồm 05 điện trở (<math><100 \Omega</math>) đấu nối các đường dẫn dữ liệu và trạng thái với nhau và phần mềm sau:



Hình 2.19 Giao diện kiểm tra thanh ghi trạng thái cổng song song

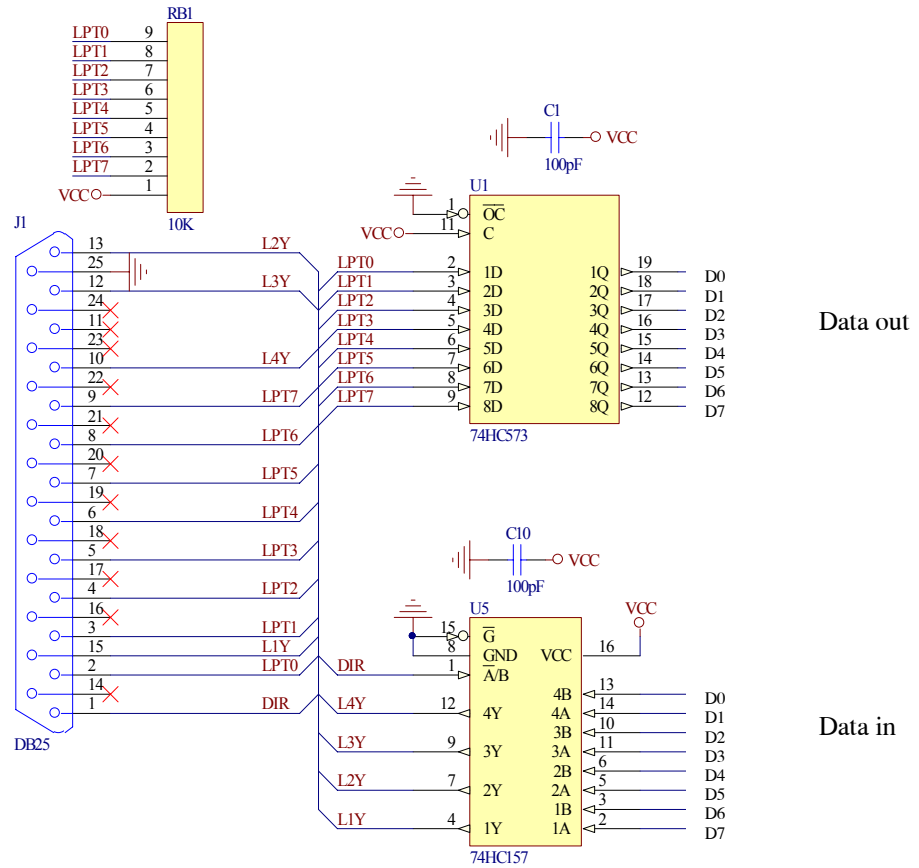


Hình 2.19 Giao diện chương trình kiểm tra thanh ghi trạng thái cổng song song

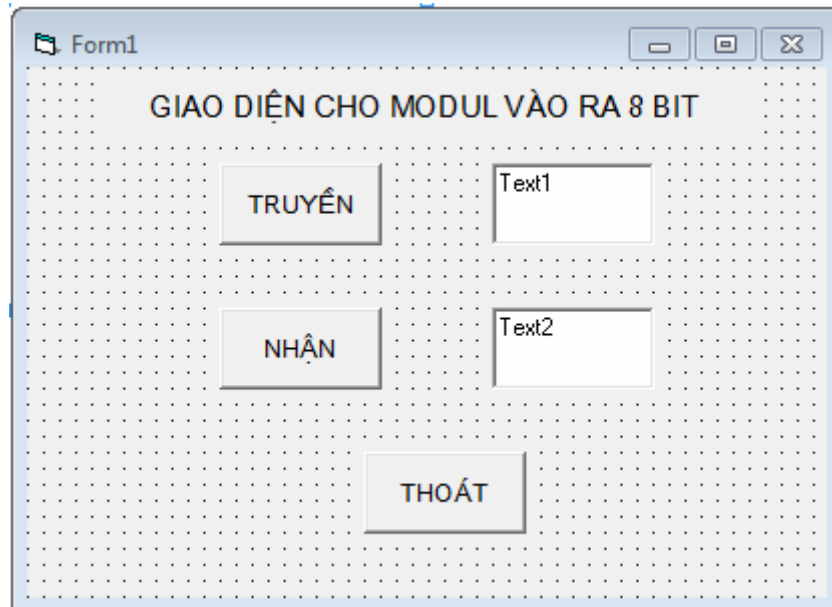
```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal Value As Integer)
Dim x As Integer
Private Sub KIEMTRA_Click()
Out &H378, Val(Text1.Text)
x = Inp(&H379)
x = x / 8
x = x And &H1F
Text2.Text = x
End Sub
```

Sau khi chạy chương trình trên, nhập giá trị trong khoảng từ 0 đến 31 vào ô text1 và bấm vào nút kiểm tra sẽ xuất hiện giá trị từ 0 đến 31 tương ứng trong ô text2, nếu giá trị đọc vào giống với giá trị xuất ra thì cổng song song có thể sử dụng tốt.

Trên cổng song song chỉ có 5 đường dẫn trạng thái có lỗi vào mà trên thực tế thường tối thiểu phải có 8 đường (8bit) để trở thành một giao diện hai hướng, ta có thể khắc phục tình trạng trên bằng một phần cứng và một phần mềm như sau:



Hình 2.20 Giao diện vào ra 8bit ở cổng song song



Hình 2.21 Giao diện chương trình vào ra 8 bit ở cổng song song

```

Private Sub Command1_Click()
Out &H378, Val(Text1.Text)
End Sub

```

```

Private Sub Command2_Click()
Dim x, y, d0, d1, d2, d3, d4, d5, d6, d7 As Byte
Out &H37A, 1
x = Inp(&H379)
d6 = x And &H40
d4 = x And &H20
d2 = x And &H10
d0 = x And &H8
Out &H37A, 0
x = Inp(&H379)
d7 = x And &H40
d5 = x And &H20
d3 = x And &H10
d1 = x And &H8
y = d7 Or d6 Or d5 Or d4 Or d3 Or d2 Or d1 Or d0
Text2.Text = (y)
End Sub

```

```

Private Sub Command3_Click()
End
End Sub

```

2.3 Rãnh cắm mở rộng

2.3.1 Giới thiệu

Các rãnh cắm mở rộng là một dạng thể hiện bằng phần cứng của các bus trên bảng mạch chính, trên đó cho phép cắm các card mở rộng để nâng cấp hoặc thay đổi cấu hình của máy tính. Cho đến nay tồn tại một số bus (rãnh cắm mở rộng) sau:

- Bus PC (còn gọi là ISA 8 bit)
- Bus ISA (16 bit)
- Bus PC/ MCIA (16 bit)
- Bus VESA Local (32 bit)
- Bus SCCI (16/32bit)
- Bus EISA (32bit)
- Bus MCA (32bit)
- Bus PCI (32/64 bit)

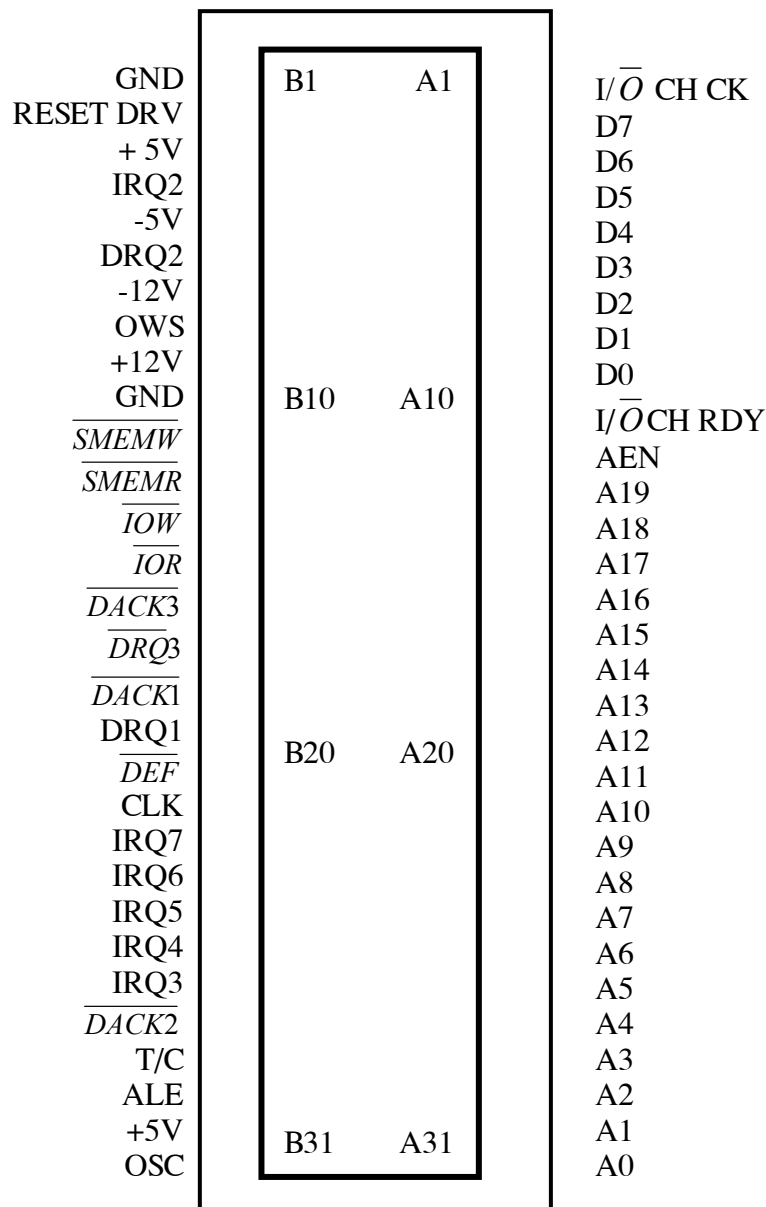
- Bus AGP (32/64 bit).

Ngoại trừ bus ISA, các bus còn lại đều có tốc độ quá nhanh, khoảng cách giữa các chân cắm quá nhỏ rất khó cho việc thiết kế mạch in và quan trọng nhất là khó tìm được linh kiện tương thích về tốc độ, do vậy các mạch điện đo lường và điều khiển sẽ chỉ tập trung vào 2 loại bus đó là:

- ISA 8 bit (PC 8 bit)
- ISA 16 bit.

2.3.2 Giới thiệu một số loại BUS

2.3.2.1. Bus PC



Hình 2.22 Sơ đồ chân của Bus PC (ISA 8bit)

Kích thước hình học:

+ Chiều cao: 106,7 mm (hay 4,2 inch)

+ Chiều dài: 333,5 mm (hay 13,3 inch)

+ Chiều dày - kể cả linh kiện: 12,7 mm (hay 0,5 inch)

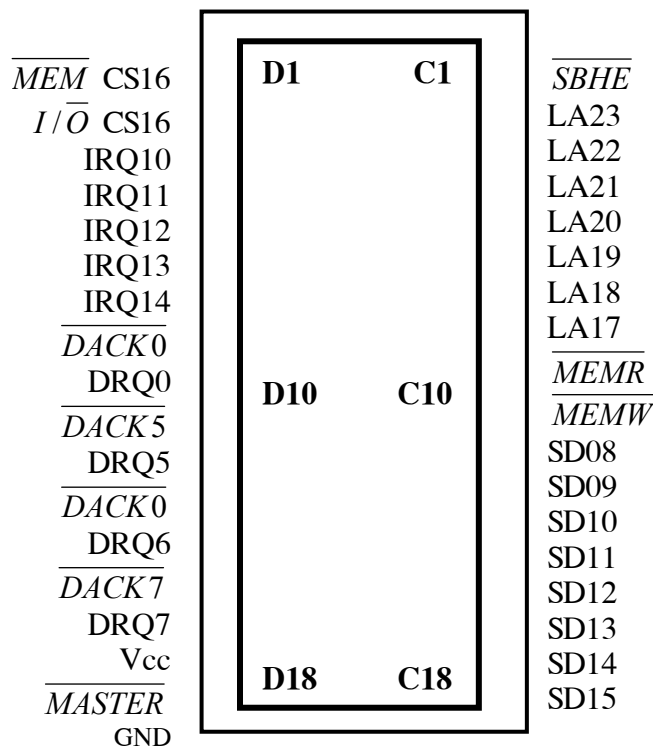
Bus PC là loại bus xuất hiện trên máy tính PC/XT, loại bus này tận dụng cấu trúc của bộ vi xử lý Intel 8088 nên có 1 bus dữ liệu 8 bit và bus địa chỉ 20 bit. Rãnh cắm nối với bus PC có 62 chân cho phép cắm vào 1 card mở rộng làm từ một tấm mạch in 2 mặt có 62 tiếp điểm. Vì bus này có 8 bit dữ liệu nên còn được gọi là bus 8 bit hay bus ISA 8 bit. Bus PC hoạt động ở tần số ở 4,772727 MHz.

Bảng 2.2 Chức năng các chân của bus PC

Tín hiệu	Mô tả
A0-A19	Bus địa chỉ 20 bit
AEN	Cho phép địa chỉ (Address enable - lối ra)
D0- D7	Bus dữ liệu (lối vào/ lối ra)
CLK	Tín hiệu giữ nhịp (4,772727 MHz)
ALE	Cho phép chốt địa chỉ (Address latch enable - lối ra)
IOR	Tín hiệu cho phép đọc
IOW	Tín hiệu cho phép ghi
SMEMR	Tín hiệu đọc bộ nhớ hệ thống, báo hiệu một chu trình bus đọc bộ nhớ hệ thống trong phạm vi bus địa chỉ 20 bit (System memory read)
SMEMW	Tín hiệu ghi bộ nhớ hệ thống, báo hiệu một chu trình bus ghi bộ nhớ hệ thống trong phạm vi bus địa chỉ 20 bit (System memory write)
DRQ1-DRQ3	Tín hiệu yêu cầu truy nhập bộ nhớ trực tiếp từ ngoài
DACK1-DACK3	Tín hiệu chấp nhận yêu cầu truy nhập bộ nhớ trực tiếp
IRQ2-IRQ7	Tín hiệu yêu cầu ngắt từ thiết bị ngoài
OSC	Tín hiệu đồng hồ có tần số 14.31838 MHz
RESET	Đặt lại trạng thái ban đầu cho bản mạch mở rộng cắm vào bus ISA.
±5V ±12V và GND	Nguồn dùng cho card mở rộng.

3.2.2. Bus ISA (16 bit)

Đây là loại bus được cấu trúc theo tiêu chuẩn công nghiệp (ISA - Industry Standard Architecture). Ưu điểm của bus này là có thể cho phép cùng một lúc xử lý với 16 bit dữ liệu. Bus ISA thực chất là bus PC được bổ sung thêm 1 rãnh cắm thứ hai thẳng hàng trên đó chứa 8 bit dữ liệu và 4 đường dẫn địa chỉ.



Hình 2.23 Sơ đồ rãnh cắm thứ hai của Bus ISA

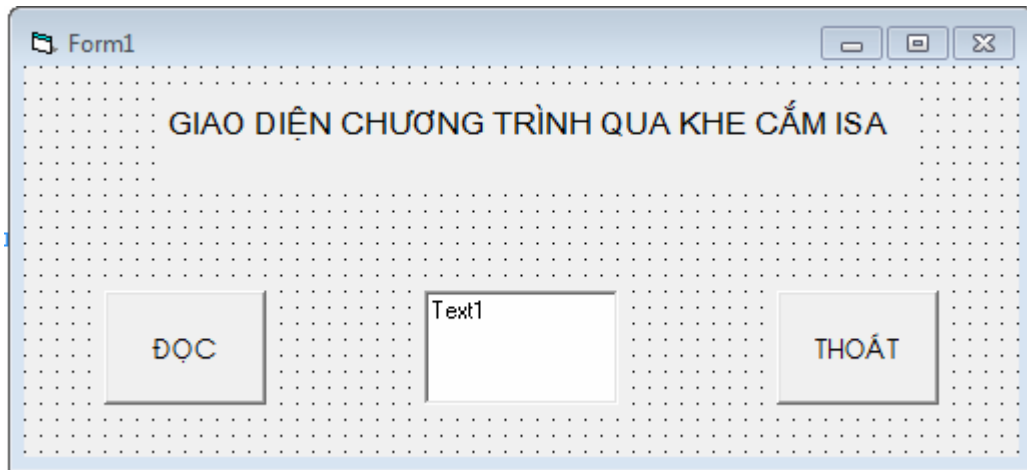
Bảng 2.3 Chức năng các chân của bus ISA

Tín hiệu	Mô tả
LA17-LA23	Phần cao của bus địa chỉ.
SBHE	Cho phép byte cao của bus dữ liệu (System byte high enable)
SD08-SD15	8 bit cao của bus dữ liệu
MEMR	Tín hiệu đọc bộ nhớ
MEMW	Tín hiệu ghi bộ nhớ
DRQ0, DRQ5-DRQ7	Các tín hiệu yêu cầu truy nhập trực tiếp bộ nhớ.
DACK0- DACK7	Các tín hiệu xác nhận yêu cầu truy nhập trực tiếp bộ nhớ.
MASTER	Tín hiệu này cho phép bộ vi xử lý khác điều khiển: đường dẫn điều khiển, đường dẫn dữ liệu và đường dẫn địa chỉ của hệ thống.
IRQ10-IRQ12; IRQ14-IRQ15	Tín hiệu yêu cầu ngắt

Vcc	Điện áp nguồn +5V
-----	-------------------

3.2.3. Ghép nối với bus ISA

Hình 2.25 là một ví dụ điển hình cho kỹ thuật ghép nối ở BUS ISA. Vi mạch 74HC688 có chức năng so sánh địa chỉ được đưa vào từ các đường địa chỉ của BUS ISA và địa chỉ được đặt trước ở các chân Q0 đến Q7 của vi mạch này, nếu các tín hiệu từ A2 đến A9 bằng Q0 đến Q7 sẽ có một mức logic “0” ở đầu ra $\overline{P=Q}$. Mức logic này sẽ dùng để điều khiển bộ điều khiển BUS 74HC245 và bộ giải mã ghi đọc 74HC138. Vi mạch ADC0804 được ghép vào BUS ISA thông qua 74HC245 và được lựa chọn bằng đầu ra Y0 của 74HC138. Như vậy, địa chỉ của ADC trong mạch này là 300h. Dưới đây là giao diện chương trình hiển thị giá trị điện áp đọc được từ ADC qua mạch ghép nối ISA.

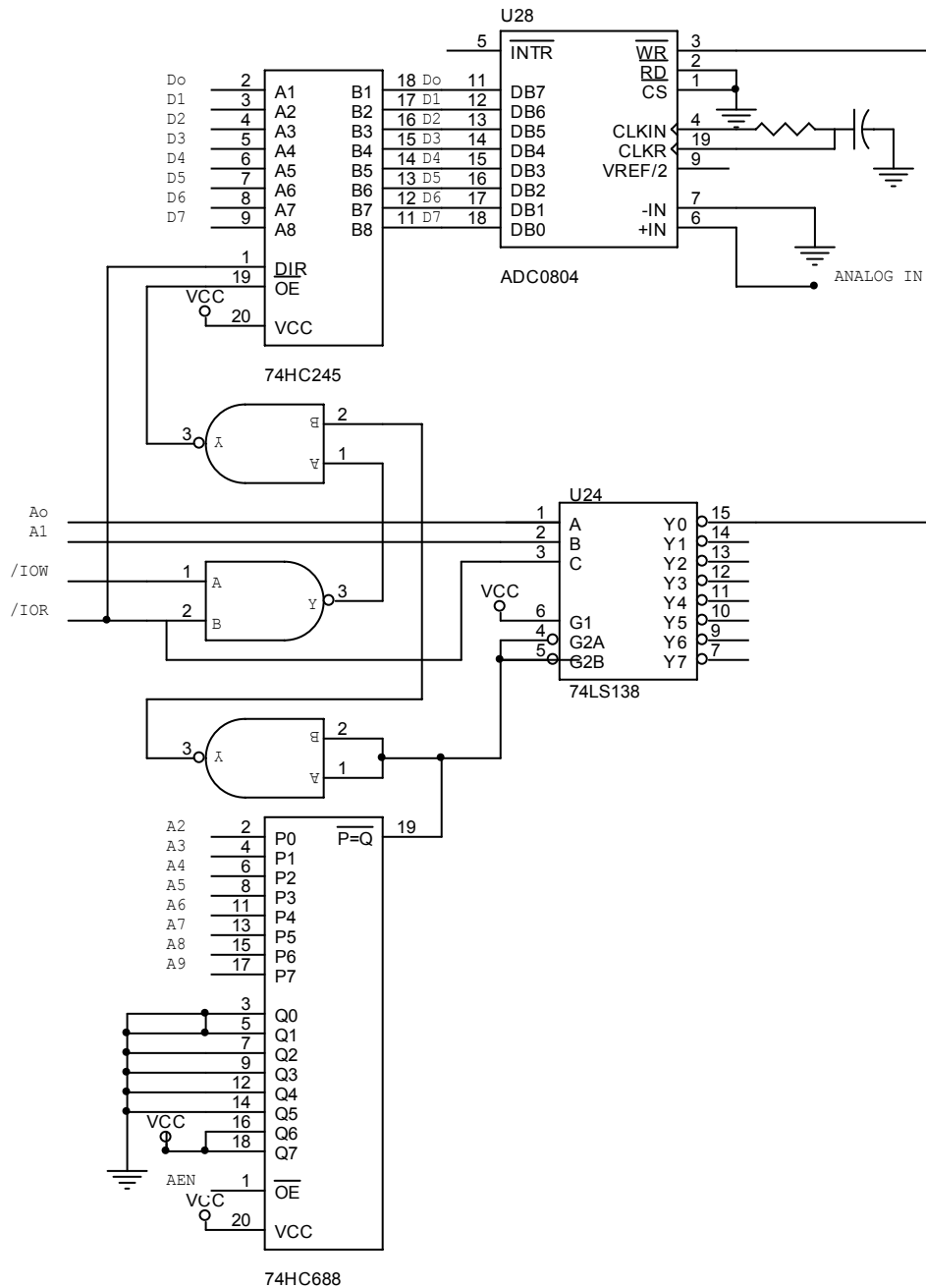


Hình 2.24 Giao diện chương trình hiển thị giá trị điện áp đọc được từ ADC

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal Value As Integer)

Private Sub Command1_Click()
Dim x As Byte
Dim volt As Single
x = Inp(&H300)
volt = x / 3
Text1.Text = volt
End Sub

Private Sub Command2_Click()
End
End Sub
```



Hình 2.25 Bộ ADC0804 được ghép vào BUS ISA

Cấu trúc bên trong cổng song song có ba thanh ghi trước khi đưa ra các chân ngoài cổng. Thực chất việc trao đổi dữ liệu với cổng song song là truy nhập các thanh ghi này. Muốn đưa dữ liệu trong máy tính ra dùng thanh ghi dữ liệu và thanh ghi điều khiển, thanh ghi trạng thái để đưa thông tin từ ngoài vào máy tính.

2.4 Bài tập cuối chương

1. Hãy nêu cấu chức năng các thanh ghi bên trong của cổng song song.

2. Hãy nêu chức năng của các vi mạch đệm chốt, khi nào các vi mạch đó được sử dụng làm bộ đệm, khi nào thì được sử dụng làm bộ chốt.
3. Thiết kế Modul vào ra 8 bit qua cổng máy in, thiết kế giao diện và viết chương trình xuất dữ liệu ra cổng máy in qua thanh ghi dữ liệu sau đó đọc vào qua thanh ghi trạng thái với giao diện hình 2.26.
4. Thiết kế mạch điều khiển động cơ điện một chiều qua cổng song song, điều khiển dừng, chạy, đảo chiều, tăng tốc độ bằng các phím từ bàn phím hoặc các nút điều khiển trên giao diện.

The image shows a software interface for an 8-bit parallel port module. The title is "TRUYỀN NHẬN 8 BIT QUA CỔNG SONG SONG". On the left, there is a label "Giá trị truyền" above a single text input box. On the right, there is a label "Giá trị nhận được" above a row of eight checkboxes labeled D7, D6, D5, D4, D3, D2, D1, and D0. Below the input field is a "TRUYỀN" button. Below the checkboxes is a "NHẬN" button. To the right of the "NHẬN" button is a "THOÁT" button.

Hình 2.26 Giao diện chương trình bài tập 2

Chương 3 Ghép nối trao đổi tin nối tiếp

Phương pháp ghép nối trao đổi tin nối tiếp là một trong những kỹ thuật được sử dụng phổ biến nhất để ghép nối các thiết bị ngoại vi với máy tính. Qua cổng nối tiếp ta có thể ghép nối chuột, bàn phím, máy in, modem và các thiết bị đo lường và điều khiển. Đặc biệt là ghép nối qua cổng USB thì số lượng và chủng loại các thiết bị đứng hàng đầu trong số các khả năng ghép nối với máy tính. Với các mạch thiết kế chuyên dụng như những modul ghép nối với máy tính để điều khiển các thiết bị bên ngoài hay thu thập các tín hiệu vật lý thì có thể chọn giải pháp là ghép nối qua cổng RS-232 hay cổng USB. Tùy theo các ứng dụng thiết kế đơn giản hay phức tạp mà chọn giải pháp ghép nối qua cổng RS-232 hay qua cổng USB. Chương này sẽ giới thiệu về cổng nối tiếp và cổng USB và phương pháp ghép nối để truyền nhận dữ liệu.

3.1 Khái niệm về truyền tin nối tiếp

3.1.1 Khái niệm

Truyền tin nối tiếp là phương thức truyền tin trong đó các bit mang thông tin được truyền kế tiếp nhau trên một đường dẫn vật lý. Tại một thời điểm phía bên truyền cũng như bên nhận chỉ có thể truyền/nhận được 1 bit.

So với truyền tin song song thì truyền tin nối tiếp có một số ưu, nhược điểm sau:

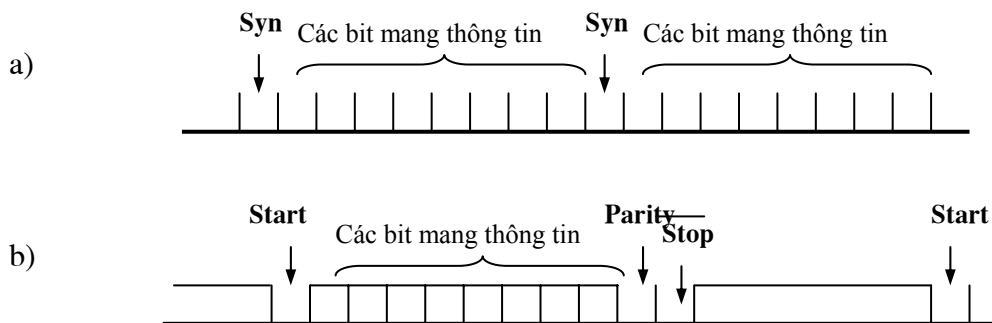
- Ưu điểm: Truyền được ở khoảng cách xa, tiết kiệm đường truyền dẫn, truyền tin nối tiếp có ưu thế hơn so với truyền tin song song.
- Nhược điểm: Tốc độ truyền chậm hơn, phức tạp hơn do các thiết bị thường phải có các khối chuyển đổi nối tiếp sang song song, song song sang nối tiếp khi sử dụng phương thức này để trao đổi tin.

3.1.2 Các phương thức truyền tin nối tiếp

Có 3 phương thức truyền tin nối tiếp:

- Phương thức đồng bộ: Các byte chứa các bit thông tin được truyền liên tiếp trên đường truyền và chỉ được ngăn cách nhau bằng bit đồng bộ khung (Syn). Hình 3.1.a.

- Phương thức không đồng bộ: Các byte chứa các bit thông tin được chứa trong một khung. Một khung được bắt đầu bằng 1 bit Start, tiếp theo là các bit mang thông tin, kế tiếp là bit kiểm tra chẵn lẻ và kết thúc là bit Stop. Khoảng các giữa các khung là các bit dừng bất kỳ, Khi đó đường truyền được đẩy lên mức cao (Hình 3.1.b).
- Phương thức lai: Đây là phương thức kết hợp của 2 phương thức trên, trong đó các bit trong 1 khung được truyền theo phương thức không đồng bộ còn các byte được truyền theo phương thức đồng bộ.



Hình 3.1 Các phương thức truyền tin nối tiếp
 a, Phương thức truyền tin đồng bộ
 b, Phương thức truyền tin không đồng bộ

3.2 Cổng nối tiếp

3.2.1 Giới thiệu

Cổng nối tiếp là một giao diện rất phổ biến trên máy tính, cổng này có một số tên gọi thông dụng như cổng COM (COM 1, COM 2...) hoặc cổng RS-232 (Recommended Standard).

Chuẩn RS-232 chỉ cho phép sử dụng đường truyền ngắn với tốc độ bit thấp. Các tiêu chuẩn truyền thông ra đời sau như RS-422, RS-449 hay RS-485 cho phép truyền với khoảng cách dài và tốc độ bit rất cao.

Giống như cổng máy in, cổng nối tiếp được sử dụng khá rộng rãi và thuận tiện cho việc ghép nối máy tính với các thiết bị ngoài. Khoảng cách truyền ở cổng nối tiếp được cải thiện hơn so với cổng song song vì điện áp chênh lệch giữa 2 mức logic 0 và 1 là khá lớn.



Hình 3.2. Giao diện của cổng nối tiếp trên máy tính

Bảng 3.1 Chức năng các chân tín hiệu ở cổng nối tiếp

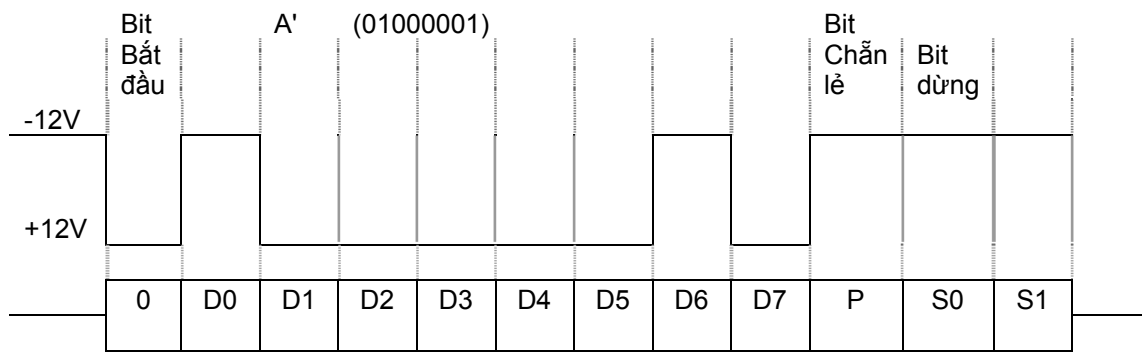
Chân	Ký hiệu	Ý nghĩa
1	DCD	Data carrier detect - Phát hiện tín hiệu mang dữ liệu
2	RxD	Receive data - Nhận dữ liệu
3	TxR	Transmit data - Truyền dữ liệu
4	DTR	Data terminal ready - Dữ liệu đầu cuối sẵn sàng
5	GND	Signal ground - Nối đất
6	DSR	Data set ready- Dữ liệu sẵn sàng được nhận
7	RTS	Request to send - Tín hiệu yêu cầu gửi
8	CTS	Clear to send - Tín hiệu yêu cầu xoá để gửi tiếp
9	RI	Ring indicator – Báo chuông

Một số thông số đặc trưng của cổng nối tiếp:

- Điện áp quy định đối với mức logic 1 nằm trong khoảng từ $-3V$ đến $-12V$; mức logic 0 là $+3V$ đến $+12V$.
- Tốc độ truyền dữ liệu được quy định chuẩn gồm các tốc độ sau: 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, và 56600 baud
- Độ dài cực đại của cáp kết nối là 15m (Trong trường hợp không có các bộ đệm, bộ lặp).

Việc truyền dữ liệu qua cổng nối tiếp RS-232 được thực hiện theo kiểu không đồng bộ, các byte chứa các bit thông tin được chứa trong một khung. Khuôn mẫu khung truyền được bắt đầu bằng bit Start, bit này luôn ở mức 0. Tiếp theo là các bit

mang thông tin, các bit mang tin trong khung truyền thường là 5, 6, 7 hoặc 8 bit được gửi dưới dạng mã ASCII, rồi đến một bit chẵn lẻ và cuối cùng là bit dừng (Stop), bit dừng luôn ở mức 1. Khoảng thời gian phân cách giữa các bit quy định tốc độ truyền. Cả bộ truyền và bộ nhận đều phải được đặt tốc độ bằng nhau vì tín hiệu dữ liệu chỉ xuất hiện trong khoảng thời gian tương đối ngắn. Hình 3.3 mô tả giản đồ mức logic thể hiện một khung truyền dữ liệu nối tiếp qua RS232.



Hình 3.3 Khuôn mẫu khung truyền ký tự 'A' qua cổng nối tiếp

Trên máy tính có thể có 4 cổng nối tiếp, tuy nhiên ngày nay vì lý do kinh tế nên hầu hết các ở máy tính để bàn người ta chỉ bố trí 2 cổng gọi là COM1 và COM2. Địa chỉ của các cổng nối tiếp như sau:

Cổng Com 1	Địa chỉ cơ bản: 3F8H
Cổng Com 2	Địa chỉ cơ bản: 2F8H
Cổng Com 3	Địa chỉ cơ bản: 3E8H
Cổng Com 4	Địa chỉ cơ bản: 2E8H

Sự trao đổi của các đường dẫn tín hiệu:

3.2.2 Lập trình cho cổng nối tiếp RS232

Trên máy tính có 1 vi mạch có chức năng truyền/nhận dữ liệu thông qua cổng nối tiếp, vi mạch đó gọi là UART (Universal Asynchronous Receiver/Transmitter - bộ truyền nhận nối tiếp không đồng bộ). UART phổ biến nhất là vi mạch 8250 của hãng Intel và các phiên bản cao hơn của vi mạch này như 16450, 16550, 16750... Vi mạch này thường được hàn cố định trên Main gần giao diện cổng nối tiếp. Việc trao đổi dữ liệu qua cổng nối tiếp thực chất là việc truy xuất các thanh ghi của vi mạch. Trên hệ điều hành Windows cũng có sẵn một chương trình truyền nhận có tên là Hyper Terminal, chương trình này cho phép truyền/nhận các ký tự nên có thể dùng để kiểm

tra các mạch ghép nối ở cổng nối. Tuy nhiên, để có các ứng dụng thì cần phải lập trình chứ không thể dựa vào một chương trình có sẵn đó được. Điều đầu tiên khi lập trình với cổng nối tiếp là phải tìm hiểu các thanh ghi của bộ UART tương ứng trên máy tính đó.

3.2.2.1 Các thanh ghi của cổng nối tiếp

Thanh ghi điều khiển modem (Địa chỉ cơ sở +4 (3FCH))

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	Loop	Out2	Out2	RTS	DTR

Trạng thái của 2 bit C1 và C0 trên thanh ghi này có liên quan trực tiếp tới các đường dẫn RTS (7) và DTR (4). D1=D0=1 thì trên 2 chân RTS và DTR sẽ có mức điện áp tương ứng mức logic 1 (-12V) và ngược lại.

LOOP: Bit này đặt bằng 1 thì 8250 hoạt động ở chế độ vòng (TXD nối với RXD) chế độ này cho phép kiểm tra đường truyền nhận của cổng COM trên cùng một máy tính.

Câu lệnh sau:

```
Out &H3FC, 0
```

thiết lập cho chân 7, 4 của cổng RS232 mức 0 tương ứng + 12V

```
Out &H3FC, 3
```

thiết lập cho chân 7, 4 của cổng RS232 mức 1 tương ứng -12V

Thanh ghi trạng thái modem (Địa chỉ cơ sở +6).(3FEH)

S7	S6	S5	S4	S3	S2	S1	S0
RLSD	RI	DSR	CTS	DRLSD	TERI	DDSR	DCTS

Thanh ghi này có chức năng thông báo về trạng thái các đường dẫn bắt tay. Điều chú ý ở thanh ghi này là có 3 bit D4, D5, D6, đây chính là lỗi vào của các đường dẫn CTS (8), DSR(6) và RI (9).

Thanh ghi điều khiển đường truyền (Địa chỉ: 3FBH)

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

Thanh ghi điều khiển đường truyền có chức năng đặt các thông số cho đường truyền, ý nghĩa từng bit như sau:

C1,C0 : Đặt số bit trong một byte:	00 - 5 bit, 01 – 6 bit 10 – 7 bit, 11 – 8 bit
C2: Đặt số bit dừng	0 – 1 bit dừng 1 – 1,5 bit dừng
C3 : Bit kiểm tra chẵn lẻ (parity)	0 - Không kiểm tra chẵn lẻ 1 - Có
C4 : Loại parity	0 - Parity lẻ 1 - Parity chẵn
C5 : Bit đánh dấu khung cần kiểm tra chẵn lẻ (Stick bit)	0 – Không có stick bit 1 – Stick bit
C6 : Bit điều khiển đường truyền (break bit)	0 – Truyền nhận bình thường 1 – Dừng (không truyền nhận)

C7: DLAB (Divisor Latch Access Bit): Bit phân chia truy nhập cho các thanh ghi có cùng địa chỉ (do 8250 có 3 bit địa chỉ nhưng có tới 9 thanh ghi).

Nếu DLAB = 1 thì:

Địa chỉ 3F8H là địa chỉ của thanh ghi chứa byte thấp của số chia tốc độ baud

Địa chỉ 3F9H là địa chỉ của thanh ghi chứa byte cao của số chia tốc độ baud

Nếu DLAB = 0 thì:

Địa chỉ 3F8H là địa chỉ của thanh ghi đệm ghi/đọc

Địa chỉ 3F9H là địa chỉ của thanh ghi cho phép ngắt

Thanh ghi trạng thái đường truyền (Địa chỉ cơ sở +5-3FDH)

Thanh ghi này phản ánh trạng thái của đường truyền nối tiếp.

0	S6	S5	S4	S3	S2	S1	S0
---	----	----	----	----	----	----	----

S0 = 1 khi có 1 byte mới nhận được

S1 = 1 khi ký tự trước chưa được đọc, ký tự mới đến sẽ xoá ký tự cũ trong bộ đệm

S2 = 1 khi có lỗi chẵn lẻ

S3 = 1 khi có lỗi khung truyền

S4 = 1 khi có gián đoạn đường truyền

S5 = 1 khi bộ truyền rỗng, công nối tiếp có thể truyền/nhận.

S6 = 1 khi bộ đệm truyền rỗng.

S7 = 0 (Không sử dụng)

Bit S0 tương đương RI; bit S6 tương đương TI

Thanh ghi đệm ghi/đọc (Địa chỉ cơ sở là 3F8H).

Thanh ghi có địa chỉ cơ sở là thanh ghi đệm ghi/đọc khi DLAB = 1 cũng Tương đương SBUF

Thanh ghi này lưu trữ tạm thời dữ liệu được truyền/nhận từ 2 chân RXD và TXD.

Thanh ghi chứa số chia tốc độ Baud (byte thấp: 3F8H-DLAB=1).

Thanh ghi này gồm 8 bit, chứa phần thập của số chia tốc độ baud. Số chia tốc độ baud được tính theo công thức sau:

$$\text{Số chia tốc độ baud} = \frac{1843200}{16 * t} \quad (t \text{ là tốc độ bound cần thiết lập})$$

Thanh ghi chứa số chia tốc độ Baud (byte cao: 3F9H-DLAB=1).

Nếu cần có tốc độ baud là 9600 thì số chia tương ứng sẽ là 12, khi đó byte thấp của số chia tốc độ baud sẽ được nạp giá trị bằng 12, byte cao được nạp giá trị bằng 0. Ta có thể thực hiện bằng các câu lệnh như sau:

```
Out &H3FB, &H80
Out &H3F8, 12
Out &H3F9, 0
```

Thanh ghi cho phép ngắt (Địa chỉ cơ sở +1; DLAB=0).

x	x	x	x	E3	E2	E1	E0
---	---	---	---	----	----	----	----

Khi các bit cho phép ngắt được đặt bằng 1 thì cho phép ngắt

Khi các bit cho phép ngắt được đặt bằng 0 thì cấm ngắt tương ứng

E0: Đặt ngắt khi nhận được 1 ký tự

E1: Đặt ngắt khi bộ đệm truyền rỗng

E2: Đặt ngắt khi thay đổi trạng thái đường truyền

E3: Đặt ngắt khi thay đổi trạng thái Modem

Thanh ghi nhận dạng ngắt. (Địa chỉ cơ sở +2)

0	0	0	0	0	S2	S1	S0
---	---	---	---	---	----	----	----

Bảng 3.2 Các mức ưu tiên của từng ngắt

S2	S1	S0	Mức ưu tiên	Nguồn gây ngắt	Đặt lại ngắt
0	0	1	Không kiểm tra ngắt		
1	1	0	Cao nhất	Lỗi đường nhận dữ liệu	Đọc thanh ghi trạng thái đường truyền
1	0	0	Thứ 2	Có dữ liệu nhận	Đọc thanh ghi đệm nhận
0	1	0	Thứ 3	Thanh ghi đệm truyền rỗng	Đọc thanh ghi IR hoặc ghi ghi vào bộ đệm truyền
0	0	0	Thứ 4	Các trạng thái của Modem	Đọc thanh ghi trạng thái Modem

3.2.2.2 Lập trình cho cổng nối tiếp

Một chương trình trao đổi dữ liệu qua cổng nối tiếp được bắt đầu bằng việc đặt các thông số cho đường truyền nối tiếp, tiếp đó là các thao tác truyền/nhận qua thanh ghi đệm. Đoạn chương trình ví dụ sau đây sẽ khởi tạo cho cổng nối tiếp ở tốc độ 9600 baud, 1 bit dừng, không kiểm tra chẵn lẻ, không stick, không break và không dùng ngắt:

```
Private Sub Form_Load()  
Out &H3FB, 128  
Out &H3F8, 12  
Out &H3F9, 0  
Out &H3FB, 3  
End Sub
```

Việc đọc 1 byte dữ liệu được bắt đầu bằng việc kiểm tra bộ đệm nhận. Nếu có 1 byte dữ liệu được nhận, bit Do của thanh ghi trạng thái đường truyền sẽ được đặt bằng 1. Dấu hiệu này cũng có thể được kiểm tra ở thanh ghi nhận dạng ngắt (nếu có sử dụng ngắt). Chương trình dưới đây sẽ minh họa cụ thể hơn:

```
Dim x As Byte  
Dim y As Byte  
Private Sub Form_Load()  
Out &H3FB, 128      \ đặt DLAB = 1  
Out &H3F8, 12  
Out &H3F9, 0        \ 9600 baud  
Out &H3FB, 3  
Out &H3F9, 0        \ Không dùng ngắt  
Out &H3FC, 0        \ Không kết nối modem  
Do  
x = Inp(&H3F8 + 5) And 1  
Loop Until x = 1  
y = Inp(&H3F8)  
Out &H3F8 + 5, &H7E  
End Sub
```

Việc gửi 1 byte dữ liệu cũng tương tự như nhận. Trước tiên ta cần kiểm tra bộ đệm truyền, nếu bộ đệm truyền rỗng (bit D6 của thanh ghi trạng thái đường truyền bằng 1) sẽ tiến hành truyền bằng cách nạp byte dữ liệu cần truyền vào bộ đệm truyền. Chương trình dưới đây sẽ minh họa cụ thể:

```

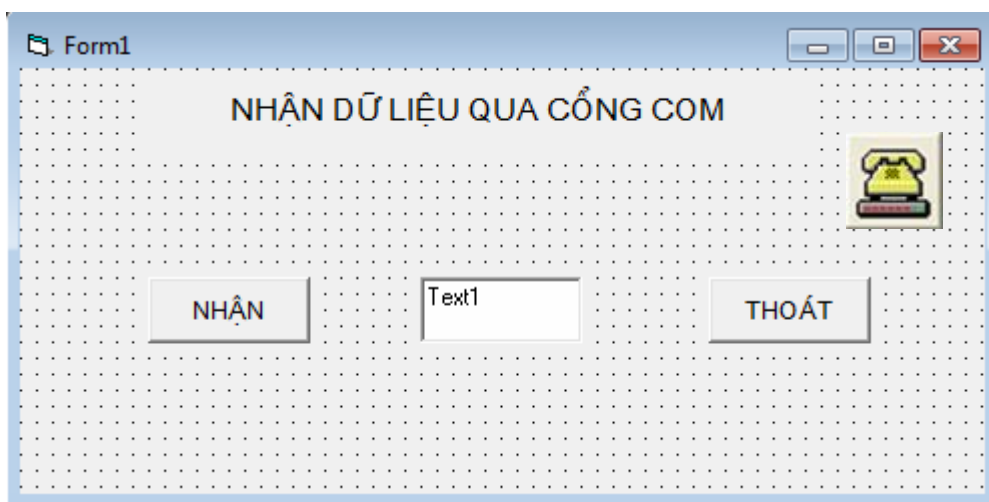
Dim x, y As Byte
Private Sub Form_Load()
Out &H3FB, 128      \ đặt DLAB = 1
Out &H3F8, 12
Out &H3F9, 0        \ 9600 baud
Out &H3FB, 3
Out &H3F9, 0        \ Không dùng ngắt
Out &H3FC, 0        \ Không kết nối modem
Do
x = Inp(&H3F8 + 5) And &H40
Loop Until x = &H40
Out &H3F8, y
Out &H3F8 + 5, &H3F
End Sub

```

Khi truyền/nhận nhiều byte trong 1 chương trình thì cần phải xóa cờ truyền/nhận (bit S6 và S0 của thanh ghi trạng thái đường truyền) sau mỗi lần truyền/nhận.

Trong ngôn ngữ lập trình Visual Basic ta có sẵn các hàm thiết lập các thông số cho cổng nối tiếp. Để sử dụng ta thêm đối tượng MSComm bằng cách chọn **Microsoft comm. Control 6.0** trong **components**.

Dưới đây là đoạn chương trình khởi tạo cho cổng nối tiếp ở tốc độ 9600 baud, 1 bit dừng, không kiểm tra chẵn lẻ, cho phép nhận ký tự và hiển thị trong ô text1.



Hình 3.4 Giao diện nhận một byte qua cổng nối tiếp

```

Private Sub Form_Load()
    With MSComm1 ' thiết đặt các thông số cho cổng COM1
        .Settings = "9600,N,8,1" ' thiết đặt cấu hình cổng
        .CommPort = 1
        .RThreshold = 1
        .SThreshold = 0
        .InputMode = 0 'comInputModeText 0
        .InputLen = 0
        .Handshaking = comNone
        .InBufferSize = 1024
        .OutBufferSize = 1024
        If .PortOpen = False Then
            .PortOpen = True
        End If
    End With
End Sub

```

```

Private Sub MSComm1_OnComm()
    If MSComm1.CommEvent = 2 Then
        kt = MSComm1.Input
        temp = Asc(kt) ' tra ve ma Asc của ký tự nhận
        Text1.Text = Str(temp)
    End If
End Sub

```

Dưới đây là đoạn chương trình khởi tạo cho cổng nối tiếp ở tốc độ 9600 baud, 1 bit dừng, không kiểm tra chẵn lẻ, cho phép nhận chuỗi ký tự với độ dài của chuỗi là 3 byte.

```

Private Sub Form_Load()
    Dim kt As String
    Text3.Text = " "
    MSComm1.CommPort = 1
    MSComm1.Settings = "9600,N,8,1"
    MSComm1.RThreshold = 1
    MSComm1.SThreshold = 1
    MSComm1.InputLen = 3
    MSComm1.InputMode = 0 'mode text
    MSComm1.PortOpen = True
End Sub

```



```

Private Sub MSComm1_OnComm()
    Dim x, y, z As String
    If MSComm1.CommEvent = comEvReceive Then
        x = MSComm1.Input
    End If
End Sub

```

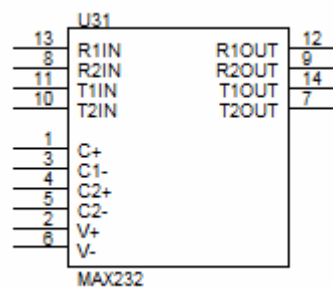
3.2.3 Modul vào ra 8 bit dùng cổng RS232

Đa phần các thiết bị ghép nối thường là 8 bit dữ liệu hoặc 16 bit, để ghép nối trao đổi tin nối tiếp ta phải thiết kế Modul có chức năng chuyển dữ liệu từ song song thành nối tiếp và ngược lại. Hạt nhân của Modul này là một bộ vi điều khiển, vi mạch này có thể đảm nhiệm chức năng như một bộ thu phát nối tiếp không đồng bộ (UART) lập trình được. Có thể dùng sử dụng bất kỳ vi điều khiển nào miễn là vi điều khiển đó có cổng nối tiếp tương thích với RS-232, giáo trình này sẽ giới thiệu về vi điều khiển 89S52. Ngoài ra, phần cứng của Modul còn sử dụng cả vi mạch MAX232 làm nhiệm vụ chuyển đổi mức từ TTL sang RS232 và ngược lại.

3.2.3.1. Vi mạch MAX232.

MAX232 là vi mạch của hãng MAXIM chứa hai bộ chuyển đổi mức từ TTL sang RS232 và hai bộ làm chức năng ngược lại.

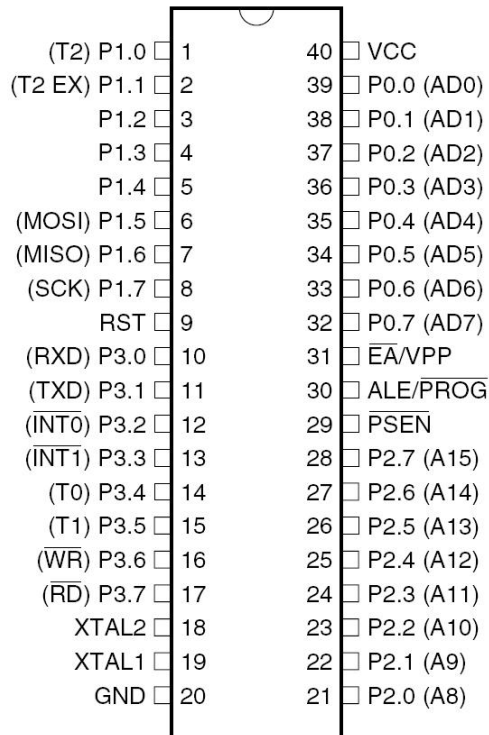
Trong hình 3.5, hai bộ chuyển mức R1IN→R1OUT, R2IN→R2OUT làm chức năng chuyển mức từ RS232 sang TTL còn hai bộ chuyển mức T1IN → T1OUT, T2IN → T2OUT làm chức năng chuyển mức từ TTL sang RS232.



Hình 3.5 Sơ đồ cấu trúc vi mạch MAX232

3.2.3.2. Vi điều khiển 89S52

89S52 là một bộ vi điều khiển bên trong nó có một bộ UART có khả năng đặt cấu hình bằng phần mềm rất tiện dụng khi thiết kế một Modul vào ra 8 bit ở cổng nối tiếp.



Hình 3.6 Sơ đồ 89C52/89S52 kiểu DIP 40 chân

Bộ vi điều khiển AT89S52 gồm các khối chức năng chính sau đây:

CPU (Central processing unit) bao gồm :

- Thanh ghi tích lũy A
- Thanh ghi tích lũy phụ B, dùng cho phép nhân và phép chia
- Đơn vị logic học (ALU : Arithmetic Logical Unit)
- Từ trạng thái chương trình (PSW : Program Status Word)
- Bốn bảng thanh ghi
- Con trỏ ngăn xếp

Bộ nhớ chương trình (Bộ nhớ ROM) gồm 8Kb Flash.

Bộ nhớ dữ liệu (Bộ nhớ RAM) gồm 256 byte.

Bộ UART (Universal Asynchronous Receiver and Transmitter) có chức năng truyền nhận nối tiếp, AT89S52 có thể giao tiếp với cổng nối tiếp của máy tính thông qua bộ UART.

3 bộ Timer/Counter 16 bit thực hiện các chức năng định thời và đếm sự kiện.

WDM (Watch Dog Timer): WDM được dùng để phục hồi lại hoạt động của CPU khi nó bị treo bởi một nguyên nhân nào đó. WDM ở AT89S52 gồm một

bộ Timer 14 bit, 1 bộ Timer 7 bit, thanh ghi WDTPRG (WDT programable) điều khiển Timer 7 bit và một thanh ghi chức năng WDTRST (WDM register). Bình thường WDT không hoạt động (bị cấm), để cho phép WDT, các giá trị 1EH và E1H cần phải được ghi liên tiếp vào thanh ghi WDTRST. Timer 14 bit của WDT sẽ đếm tăng dần sau mỗi chu kỳ đồng hồ cho đến giá trị 16383 thì xảy ra tràn. Khi xảy ra tràn, chân Reset sẽ được đặt ở mức cao trong khoảng thời gian 96 Tosc ($Tosc=1/Fosc$) và AT89S52 sẽ được reset. Khi WDT hoạt động, ngoại trừ Reset phần cứng và Reset do WDT tràn thì không có cách nào có thể cấm được WDT, vì vậy khi sử dụng WDT thì các đoạn mã của chương trình phải được đặt trong các khe thời gian giữa các lần WDT được khởi tạo lại.

Thanh ghi WDTPRG:

7	6	5	4	3	2	1	0
-	-	-	-	-	S2	S1	S0

Khởi điều khiển ngắt với 2 nguồn ngắt ngoài và 4 nguồn ngắt trong.

Bộ lập trình (ghi chương trình lên Flash ROM) cho phép người sử dụng có thể nạp các chương trình cho chip mà không cần các bộ nạp chuyên dụng.

Bộ chia tần số với hệ số chia là 12.

4 Ports xuất nhập với 32 chân.

a, Chức năng của các ports

Port 0 gồm 8 chân, ngoài chức năng xuất nhập, Port 0 còn là bus đa hợp dữ liệu và địa chỉ (AD0-AD7), chức năng này sẽ được sử dụng khi AT89S52 giao tiếp với các thiết bị ngoài có kiến trúc Bus như các vi mạch nhớ, mạch PIO....

Chức năng duy nhất của Port 1 là chức năng xuất nhập, cũng như các Port khác, Port 1 có thể xuất nhập theo bit và theo byte. Riêng dòng 89Sxx, ba chân P1.5, P1.6, P1.7 được dùng để nạp ROM theo chuẩn ISP ; hai chân P1.0 và P1.1 được dùng cho bộ Timer 2.

Port 2 ngoài chức năng là cổng vào/ra như port 0 và port còn là byte cao của bus địa chỉ khi sử dụng bộ nhớ ngoài.

Mỗi chân trên Port 3 ngoài chức năng xuất nhập còn có một chức năng riêng, cụ thể như sau:

Bit	Tên	Chức năng
P3.0	RXD	Dữ liệu nhận cho Port nối tiếp
P3.1	TXD	Dữ liệu phát cho Port nối tiếp
P3.2	INT0	Ngắt 0 bên ngoài
P3.3	INT1	Ngắt 1 bên ngoài
P3.4	T0	Ngõ vào của Timer/counter 0

P3.5	T1	Ngõ vào của Timer/counter 1
P3.6	/WR	Xung ghi bộ nhớ dữ liệu ngoài
P3.7	/RD	Xung đọc bộ nhớ dữ liệu ngoài

b, Chức năng của các bộ định thời

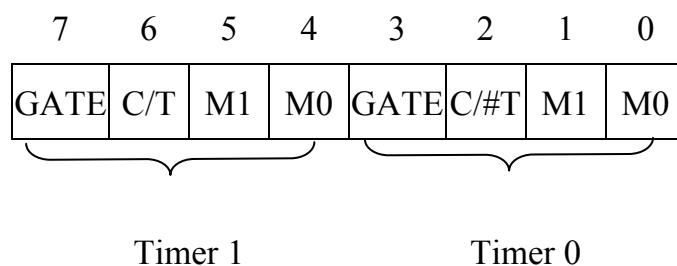
Bộ định thời (Timer) được sử dụng trong hầu hết ứng dụng điều khiển. AT89S52 có ba bộ Timer 16 bit. Hai bộ Timer 0 và Timer 1 có bốn chế độ hoạt động, Timer 2 có ba chế độ hoạt động. Các bộ Timer được dùng để hẹn giờ, đếm sự kiện xảy ra bên ngoài bộ vi điều khiển hoặc tạo tốc độ baud cho cổng nối tiếp bên trong.

Trong các ứng dụng định khoảng thời gian, Timer được lập trình sao cho sẽ tràn sau một khoảng thời gian và thiết lập cờ tràn bằng 1. Cờ tràn được dùng để đồng bộ chương trình nhằm thực hiện một tác động như kiểm tra trạng thái của các ngõ vào hoặc gửi các sự kiện ra các ngõ ra. Các ứng dụng khác có thể sử dụng việc tạo xung nhịp đều đặn của Timer để đo thời gian trôi qua giữa hai sự kiện (ví dụ : đo độ rộng xung).

Đếm sự kiện dùng để xác định số lần xảy ra của một sự kiện. Trong ứng dụng này người ta tìm cách quy các sự kiện thành sự chuyển mức từ 1 xuống 0 trên các chân T0,T1,T2 để dùng các Timer đếm các sự kiện đó.

Các Timer còn được sử dụng để tạo tốc độ baud cho cổng nối tiếp.

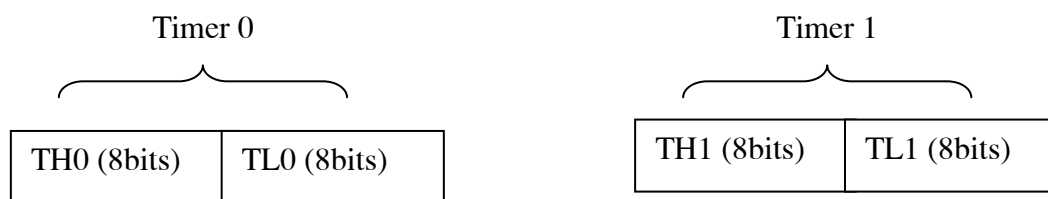
Thanh ghi TMOD chứa hai nhóm 4 bit dùng để đặt chế độ làm việc cho Timer 0 và Timer 1.



Thanh ghi TCON chứa các bit trạng thái và các bit điều khiển cho Timer 0 và Timer 1.

TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
TF1	TR1	TF0	TR0	IT1	IE1	IT0	IE0

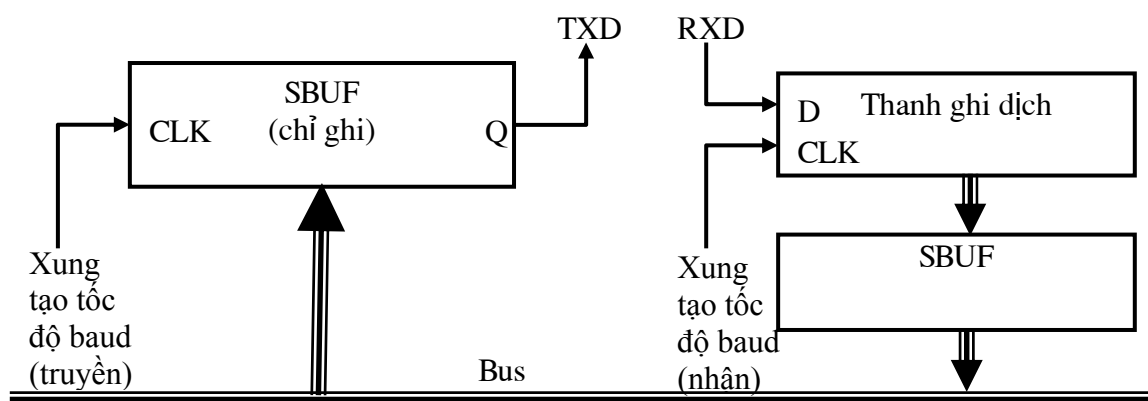
Các Timer0 và Timer1 đều là các Timer 16 bit, mỗi Timer có 2 thanh ghi 8 bit dùng để chứa giá trị khởi tạo hoặc giá trị hiện thời của các Timer. Cụ thể Timer 0 có TH0 và TL0; Timer 1 có TH1 và TL1. Lưu ý các thanh ghi này không định địa chỉ bit.



Timer 2 có ba chế độ hoạt động chế độ thu nhận (Captuer), chế độ tự nạp lại (Auto - reload) và chế độ cung cấp tốc độ baud cho cổng nối tiếp (Baud Rate Generator).

b, Cổng nối tiếp

AT89S52 có một cổng (Port) nối tiếp trong chip có thể hoạt động ở nhiều chế độ khác nhau với các tốc độ khác nhau. Chức năng chủ yếu của Port nối tiếp là thực hiện chuyển đổi song song sang nối tiếp với dữ liệu xuất và chuyển đổi nối tiếp sang song song với dữ liệu nhập để có thể giao tiếp với máy tính (hình 2.30) qua cổng nối tiếp hoặc các thiết bị tương tự.

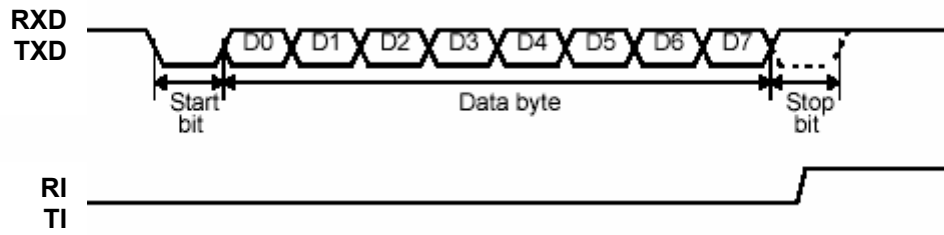


Hình 3.7 Sơ đồ khối Port nối tiếp của 89S52

Port nối tiếp có bốn chế độ hoạt động, trong đó ở chế độ 1, Port nối tiếp hoạt động như một bộ UART 8 bit có tốc độ thay đổi. Dữ liệu được truyền nối tiếp trên chân TXD và nhận nối tiếp trên chân RXD, chế độ này cung cấp cho AT89S52 một công cụ giao tiếp với máy tính qua cổng Com.

Với chế độ 1, 1 khung truyền sẽ gồm 10 bit, ngoài 8 bit dữ liệu ra còn có 1 bit start (ở mức thấp) và 1 bit stop (ở mức cao), LSB cũng được truyền trước, MSB được truyền sau.

Tốc độ baud của Port nối tiếp trong chế độ 1 có thể được cung cấp bởi Timer 1 hoặc Timer 2 hoặc đồng thời cả 2 bộ Timer nếu muốn tốc độ truyền và tốc độ nhận khác nhau (hình 3.10).



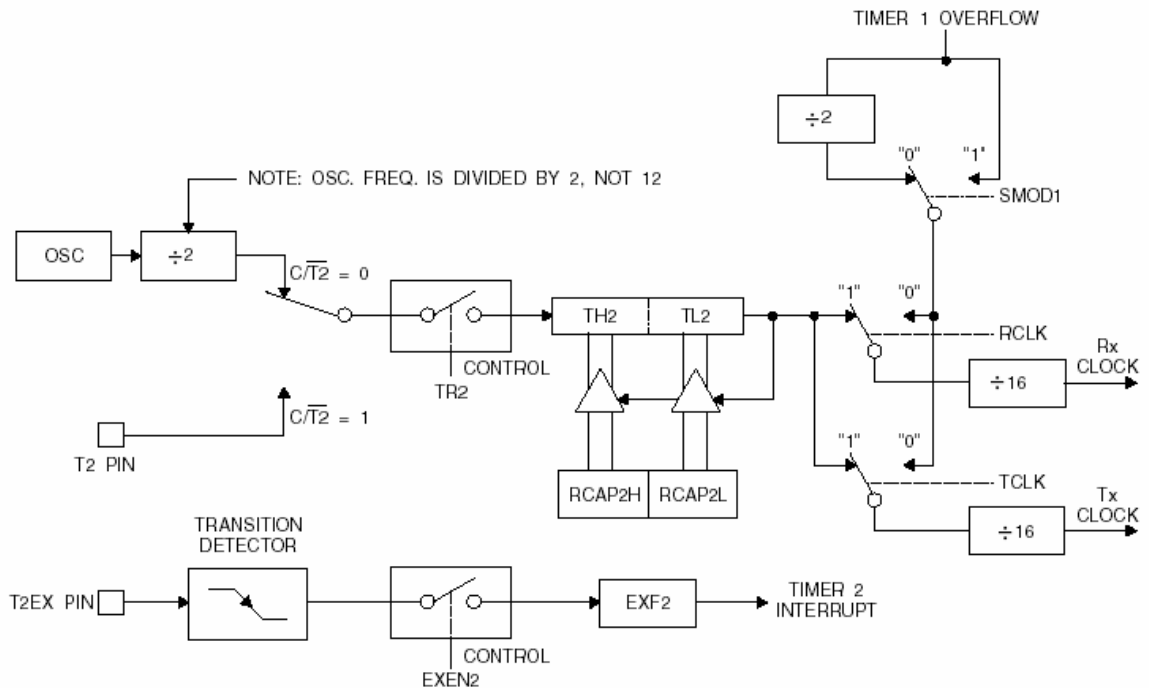
Hình 3.9 Giải đồ truyền nhận dữ liệu ở chế độ 1.

Khi sử dụng các bộ Timer cung cấp tốc độ baud cho Port nối tiếp thì thanh anh có tần số 11.0592 Mhz được khuyến cáo nên dùng vì với tần số này sẽ tạo được các tốc độ baud chuẩn với sai số bằng 0.

Ví dụ muốn có tốc độ baud là 9600 thì cần có tốc độ tràn của Timer 1 là $f_1 = 9600 \times 32 = 307200$ (Hz). Nếu sử dụng thanh anh 11.0592 Mhz thì tần số của xung clock cấp cho Timer 1 sẽ là $f_2 = 11059200 / 12$ (Hz). Như vậy cần khởi tạo cho Timer 1 giá trị nhỏ hơn giá trị xảy ra tràn là $f_2/f_1=3$ nghĩa là Timer 1 sẽ được đặt ở chế độ 2 và giá trị nạp cho TH1 là -3.

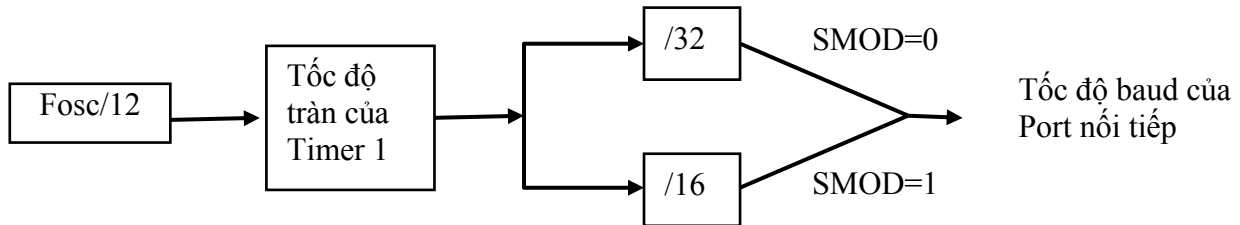
Trên hình 3.10, khi $TCLK = RCLK = 1$ thì tốc độ baud của Port nối tiếp được cung cấp bởi Timer 2.

Khác với Timer 1, Timer 2 được cấp xung clock có tần số bằng 1/2 tần số của bộ dao động thạch anh, theo hình 2.37, giả sử cần tốc độ baud là 9600 thì giá trị nạp cho Timer 2 sẽ là $-(11059200/2)/(9600 \times 16) = -36$.

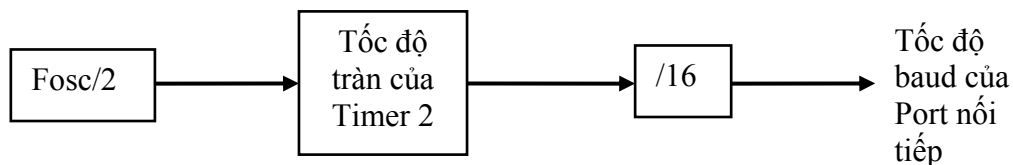


Hình 3.10 Dùng Timer 1, 2 cung cấp tốc độ baud cho Port nối tiếp.

Khi cần có tốc độ baud khác nhau cho đường truyền và đường nhận thì có thể sử dụng cả 2 bộ Timer (hình 2.35). Nếu đặt TCLK=1 và RCLK=0 thì tốc độ baud của đường truyền sẽ được cung cấp bởi Timer 2, tốc độ baud của đường nhận sẽ được cung cấp bởi Timer 1. Nếu đặt TCLK=0 và RCLK=1 thì tốc độ baud của đường truyền sẽ được cung cấp bởi Timer 1, tốc độ baud của đường nhận sẽ được cung cấp bởi Timer 2.



Hình 3.11 Dùng Timer 1 cung cấp tốc độ baud cho Port nối tiếp
(TCLK = RCLK = 0).

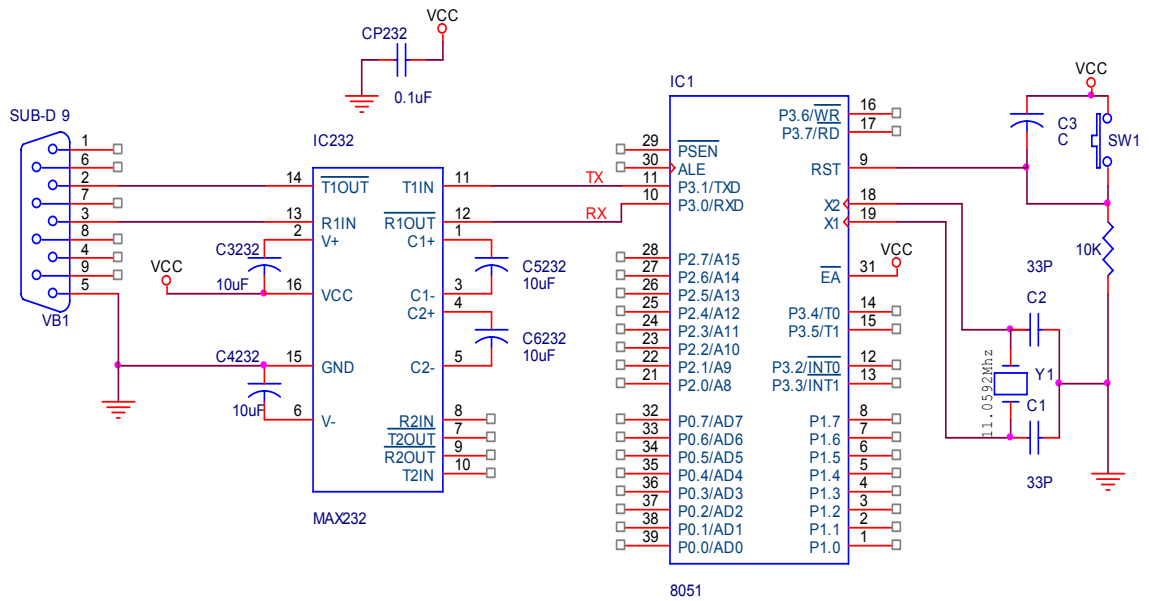


Hình 3.12 Dùng Timer 2 cung cấp tốc độ baud cho Port nối tiếp

Thao tác trao đổi dữ liệu qua Port nối tiếp không đơn thuần chỉ là việc ghi/đọc dữ liệu như trao đổi dữ liệu trực tiếp qua các Port mà còn bao gồm 3 thao tác chính như sau:

- Khởi tạo cổng nối tiếp:
 - + Truy xuất SCON để đặt các thông số như chế độ hoạt động, cho phép thu...
 - + Thiết lập hoặc xoá bit SMOD của thanh ghi PCON để đặt hệ số chia của tốc độ baud.
 - + Truy xuất các thanh ghi của các bộ Timer 1 và Timer 2 để đặt tốc độ baud cho Port nối tiếp (chỉ với chế độ 1 và 3).
- Kiểm tra cờ TI (khi truyền) và kiểm tra cờ RI (khi nhận).
- Ghi/đọc byte dữ liệu ở SBUF.

3.2.3.3 Modul vào ra 8 bit qua cổng RS-232



Hình 3.13 Module vào ra 8bit ở cổng nối tiếp

Dưới đây là chương trình truyền nhận ký tự ở cổng nối tiếp của vi điều khiển viết bằng ngôn ngữ lập trình C.

Chương trình truyền 1 byte từ cổng nối tiếp của vi điều khiển

```
#include<stdio.h>
#include<reg52.h>
char x;
void main(void)
{
    SCON = 0x52;        //Port nối tiếp chế độ 1, REN=TI=1.
    TMOD = 0x20;       //Timer 1 mode 2
    TH1 = TL1 = -3;    //Tốc độ baud là 9600
    TR1 = 1;
    While ( !TI);     //chờ TI=1
    TI=0;              //xoá TI
    SBUF=x;           //truyền byte dữ liệu trong biến x
}

```

Chương trình nhận 1 byte từ cổng nối tiếp của vi điều khiển

```
#include<stdio.h>
#include<reg52.h>
char x;
void main(void)
{
    SCON = 0x52;        //Port nối tiếp chế độ 1, REN=TI=1.

```



```

TMOD = 0x20; //Timer 1 mode 2
TH1 = TL1 = -3; //Tốc độ baud là 9600
TR1 = 1;
While ( !RI) ; //chờ RI=1
RI=0 ; //xoá RI
x=SBUF ; //nhận byte dữ liệu chứa trong biến x
}

```

3.3 Cổng USB (Universal Serial Bus)

3.3.1 Giới thiệu

USB (Universal Serial Bus) là một chuẩn kết nối tuần tự trong máy tính. USB sử dụng để kết nối các thiết bị ngoại vi với máy tính, chúng thường được thiết kế dưới dạng các đầu cắm cho các thiết bị tuân theo chuẩn cắm-là-chạy (plug-and-play) với tính năng cắm và ngắt các thiết bị không cần phải khởi động lại hệ thống.

Các giao diện song song, nối tiếp, máy quét hình, máy ảnh số, modem, thẻ nhớ v.v... đều có thể đấu nối vào bus USB. Bus USB được đưa ra sử dụng lần đầu tiên vào đầu năm 1996 nhưng phải đến giữa năm 1998 mới thực sự được hỗ trợ đầy đủ. Các thông số kỹ thuật của bus USB đã được các công ty lớn cùng tham gia xây dựng như Compaq, Digital Equipment, IBM, Intel, Microsoft, NEC và Northern Telecom. Có thể nói bus USB đã nhanh chóng trở thành một chuẩn không chính thức. Hình 3.14 là biểu tượng gắn trên các thiết bị ghép nối với cổng USB.



Hình 3.14 Biểu tượng của cổng USB

Sau khi được giới thiệu đã có rất nhiều thiết bị sử dụng trong lĩnh vực điện tử gia dụng, truyền thông được thiết kế để có thể đấu nối vào bus USB. Sự hỗ trợ bus USB thể hiện qua Win32 Driver Model (WDM) và nhờ vậy cho phép lập trình các phần mềm điều khiển thống nhất dùng cho Windows 9x và NT.

Trong các hệ điều hành ra đời từ năm 1998 đều đã có sự hỗ trợ đầy đủ cho bus USB, chẳng hạn trong Windows 98 hoặc Windows NT 5.0. Trên thực tế, trong các phiên bản nâng cấp của Windows 95 (từ phiên bản OEM-2.1) đã bắt đầu có tính năng hỗ trợ. Từ phiên bản OSR 2.0 của Windows 95 sự hỗ trợ cho USB đã có thể nhận được từ chương trình cài đặt.

Các máy tính đời mới hiện nay đều có trang bị rất nhiều cổng USB và thường bỏ bớt cổng LPT và cổng RS-232 để gọn nhẹ và giảm chi phí. Như vậy, sự ra đời của USB nhằm thống nhất các kiểu ghép nối máy tính về cùng một dạng.

3.3.2 Những đặc trưng của USB

- USB hỗ trợ các cuộc truyền đẳng thời, các tín hiệu video và âm thanh, các tín hiệu truyền được đảm bảo
- USB hỗ trợ cả các thiết bị không đẳng thời, các thiết bị đẳng thời hay không đẳng thời đều có thể tồn tại ở cùng một thời điểm.
- Có khả năng mở rộng tới 127 thiết bị có thể kết nối cùng vào một máy tính trên một cổng USB duy nhất (bao gồm các hub USB) và có các thao tác xảy ra đồng thời.
- Những sợi cáp USB riêng lẻ có thể dài tới 5 mét; với những hub, có thể kéo dài tới 30 mét.
- Với USB 2.0 (tốc độ cao), đường truyền đạt tốc độ tối đa đến 480 Mbps.
- Trên sợi nguồn, máy tính có thể cấp nguồn lên tới 500mA ở điện áp 5V một chiều (DC).
- Những thiết bị tiêu thụ công suất thấp (ví dụ: chuột, bàn phím, loa máy tính công suất thấp...) được cung cấp điện năng cho hoạt động trực tiếp từ các cổng USB mà không cần có sự cung cấp nguồn riêng thậm trí cả các thiết bị giải trí số như SmartPhone, PocketPC ngày nay sử dụng các cổng USB để sạc pin.
- Những thiết bị USB có đặc tính cắm nóng, điều này có nghĩa các thiết bị có thể được kết nối (cắm vào) hoặc ngắt kết nối (rút ra) trong mọi thời điểm mà người sử dụng cần mà không cần phải khởi động lại hệ thống.
- Nhiều thiết bị USB có thể được chuyển về trạng thái tạm ngừng hoạt động khi máy tính chuyển sang chế độ tiết kiệm điện.
- Hỗ trợ khả năng nhận dạng các thiết bị mắc lỗi

3.3.3 Cấu trúc cổng USB

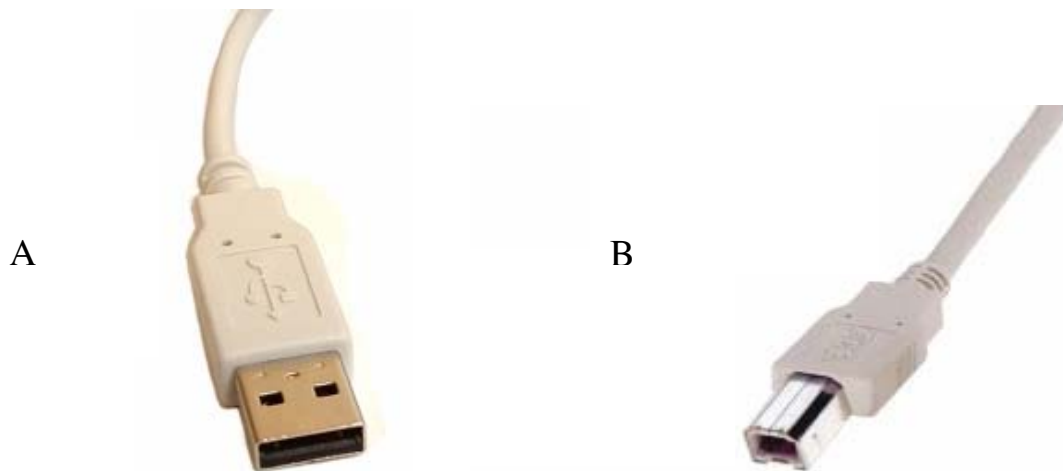
Bus USB có hai kiểu đầu nối khác nhau là kiểu A và kiểu B (hình 3.15). Hệ thống được thiết kế sao cho không xảy ra hiện tượng đấu nối nhầm. Khác với chuẩn RS-232, việc đấu nối ở bus USB không chia ra những kiểu nối thẳng hay bắt chéo. Bus USB sử dụng một cặp bốn sợi dây để nối với các thiết bị, trong đó một cặp đường truyền hai sợi xoắn được dùng làm các đường dẫn dữ liệu vi phân (D+ và D-), còn hai cặp kia được dùng làm đường dẫn 5V và đường nối đất chung GND. Cáp nối luôn thực hiện liên kết 1:1. Sự sắp xếp các chân ở đầu nối cáp luôn tuân theo quy định như bảng 3.4 .

Tùy vào máy tính PC có ổ cắm kiểu A hay kiểu B mà việc đấu nối với máy tính được thực hiện bằng một cáp kiểu A-B hay cáp kiểu A-A. Các cáp dùng để kéo dài khoảng cách từ máy tính PC đến thiết bị thường là kiểu A-A. Cho đến nay, các cáp nối

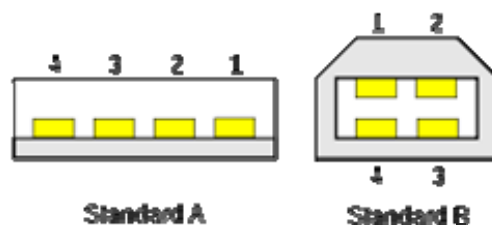
USB đều đã được các nhà sản xuất cung cấp dưới dạng hoàn chỉnh, trên đó, đầu cắm, độ dài, chất lượng bọc kim chống nhiễu đều không thể thay đổi được. Vì vậy, tùy theo mục đích sử dụng ta phải lựa chọn các thông số cấp cho chính xác, từ chiều dài đến loại đầu nối (A hay B). Chuẩn A dùng để cắm vào máy tính, chuẩn B cắm vào thiết bị ngoại vi.

Bảng 3.4 Ý nghĩa chân ở cổng USB

Chân	Tên gọi	Màu dây	Mô tả
1	Vcc	Đỏ	+5 VDC
2	D-	Trắng	Dữ liệu -
3	D+	Xanh lục	Dữ liệu +
4	GND	Đen	Nối đất



Hình 3.15 Các đầu cắm kiểu A và kiểu B



Hình 3.16 Sơ đồ chân cổng USB kiểu A và kiểu B

Hình 3.16 là sơ đồ 4 đường bên trong USB ở một dây dẫn kết nối USB, trong đó, chân 1, 4 là đường nguồn 5VDC chân 2, 3 là đường tín hiệu.

Ổ cắm USB lấy ra từ máy tính có thể lấy ra điện áp 5V với dòng điện tiêu thụ khoảng 100mA. Tín hiệu trên đường dẫn D+ và D- là các tín hiệu vi phân với mức điện áp bằng 0/3.3V

3.3.4 Truyền dữ liệu qua cổng USB

Bus USB là một bus nối tiếp. Dữ liệu truyền trên bus tương tự trong trường hợp cổng nối tiếp, cụ thể là theo từng bit kế tiếp nhau. Một điểm đáng lưu ý là dữ liệu được truyền trên cùng các đường dẫn theo hai hướng, trong khi tiêu chuẩn RS-232 dữ liệu truyền trên các đường dẫn khác nhau, mỗi đường chỉ theo một hướng. Sự khác nhau cơ bản thể hiện ở chỗ các giao diện nối tiếp từ trước đến nay chỉ có thể sử dụng cho một thiết bị nhưng USB lại cho phép đấu nối đến 127 thiết bị, cũng chính vì vậy được gọi là một bus.

Mỗi thiết bị đấu nối vào bus nhận được một địa chỉ và cũng chính qua địa chỉ này thiết bị có thể trao đổi dữ liệu với máy tính PC cũng như các thiết bị khác. Về mặt tốc độ việc trao đổi dữ liệu qua bus USB nhanh hơn qua cổng RS-232.

Khi một máy tính được cấp nguồn, nó truy vấn tất cả thiết bị được kết nối vào đường truyền và gán mỗi thiết bị một địa chỉ. Quy trình này được gọi là liệt kê – những thiết bị được liệt kê khi kết nối vào đường truyền. Máy tính cũng tìm ra từ mỗi thiết bị cách truyền dữ liệu nào mà nó cần để hoạt động:

- Ngắt: Một thiết bị như chuột hoặc bàn phím, gửi một lượng nhỏ dữ liệu, sẽ chọn chế độ ngắt.
- Hàng loạt: Một thiết bị như một chiếc máy in, nhận dữ liệu trong một gói lớn, sử dụng chế độ truyền hàng loạt. Một khối dữ liệu được gửi đến máy in (một khối 64 byte) và được kiểm tra để chắc chắn nó chính xác.
- Đẳng thời: Một thiết bị truyền dữ liệu theo chuỗi (lấy ví dụ như loa) sử dụng chế độ đẳng thời. Những dòng dữ liệu giữa thiết bị và máy trong thời gian thực, và không có sự sửa lỗi ở đây.

Máy tính có thể gửi lệnh hay truy vấn tham số để điều khiển những gói tin. Khi những thiết bị được liệt kê, máy tính sẽ giữ sự kiểm tra đối với tổng băng thông mà tất cả những thiết bị đẳng thời và ngắt yêu cầu. Chúng có thể tiêu hao tới 90% của 480Mbps băng thông cho phép. Sau khi 90% được sử dụng, máy tính sẽ từ chối mọi truy cập của những thiết bị đẳng thời và ngắt khác. Điều khiển gói tin và gói tin cho truyền tải hàng loạt sử dụng mọi băng thông còn lại (ít nhất 10%). USB chia băng thông cho phép thành những khung và máy tính điều khiển những khung đó. Khung chứa 1500 byte và một khung mới bắt đầu mỗi ms. Thông qua một khung, những thiết bị đẳng thời và ngắt lấy được một vị trí do đó chúng được đảm bảo băng thông mà chúng cần. Truyền tải hàng loạt và điều khiển truyền tải sử dụng phần còn lại.

3.3.5 Hub USB

Khi cần đấu nối nhiều thiết bị ta cần một hộp phân phối được gọi là Hub, Phần lớn những máy tính ta mua ngày nay có nhiều đầu cắm USB được thiết kế sẵn trên các cổng vào/ra hoặc các đầu cắm trên bo mạch chủ. Tuy nhiên người sử dụng có thể sử dụng các thiết bị ngoại vi hơn số cổng sẵn có qua khả năng mở rộng thiết bị trên các cổng USB thông qua các USB hub. Hub cũng tránh xảy ra tình trạng tín hiệu tốc độ

cao được chuyển giao xuống tín hiệu có tốc độ thấp. Trên hình 3.17 một USB hub cho ra 4 cổng USB 2.0



Hình 3.17 Hub USB

Các hub này có thể mở rộng ra rất nhiều cổng và nếu chúng được cung cấp nguồn điện từ bên ngoài (sử dụng các bộ adapter cấp nguồn riêng) sẽ cho phép các thiết bị USB sử dụng năng lượng từ hub mà không bị hạn chế bởi công suất giới hạn trên cổng USB trên máy tính.

Các USB hub hiện nay rất đa dạng về chủng loại, chuẩn hỗ trợ, số cổng mở rộng, hình dạng và thiết kế tích hợp. Nhiều thiết bị ngoại vi đã tích hợp các hub giúp cho người sử dụng dễ dàng cắm các thiết bị kết nối qua cổng USB, màn hình máy tính, bàn phím máy tính...cũng có thể được tích hợp USB hub.

Một số thiết bị ngoại vi sử dụng các cổng USB để cấp nguồn cho chúng (như các ổ đĩa cứng gắn ngoài không có nguồn độc lập) với yêu cầu cắm vào đồng thời hai cổng USB thì điều này có nghĩa rằng chúng cần một công suất lớn hơn so với khả năng cung cấp của một cổng USB trên máy tính. Nếu sử dụng USB hub loại không có nguồn điện ngoài thì cũng trở thành vô nghĩa bởi đầu cắm còn lại của thiết bị ngoại vi này chỉ dùng để lấy điện. Sự vô ý này của rất nhiều người sử dụng đã làm hư hỏng bo mạch chủ bởi sự cung cấp điện năng quá tải giới hạn cho mỗi đầu ra USB.

3.4 Bài tập cuối chương

1. Hãy nêu ưu nhược điểm khi truyền tin bằng cổng nối tiếp.
2. Tại sao khi truyền tin ở khoảng cách xa người ta hay dùng cổng nối tiếp.
3. Viết chương trình gửi liên tiếp các ký tự từ 'A' đến 'Z' ra cổng nối tiếp của máy tính.
4. Viết chương trình nhận liên tiếp 10 ký tự bất kỳ từ cổng nối tiếp và lưu các ký tự đó vào một mảng.

Chương 4 Thiết kế ứng dụng đo lường điều khiển bằng máy tính

Khi ghép nối các thiết bị với máy tính có thể tạo nên một hệ thống đo lường và điều khiển tự động có chất lượng cao. Các tín hiệu được thu thập và xử lý ngay trên máy tính. Trong chương này sẽ trình bày một số ứng dụng ghép nối qua cổng song song, cổng nối tiếp và cổng USB. Trong các ứng dụng thiết kế có những ứng dụng dùng các vi mạch số lập trình được như vi mạch vào ra song song lập trình được 8255A, ứng dụng dùng vi mạch tạo xung 8253 và chắc chắn không thể thiếu các bộ biến đổi tương tự số ADC hay các bộ biến đổi DAC. Các vi mạch này cũng được giới thiệu trong chương này.

4.1 Quy trình thiết kế các ứng dụng đo lường điều khiển bằng máy tính

Việc thiết kế ứng dụng ghép nối với máy tính có những bước cơ bản sau:

Bước 1: Phân tích bài toán

Mục đích của việc phân tích bài toán là để xác định các yêu cầu kỹ thuật như yêu cầu về khả năng tính toán (tốc độ và độ chính xác); số đầu vào/ra số; số đầu vào/ra tương tự; số lượng bộ đếm/định thời; ước lượng dung lượng bộ nhớ chương trình, bộ nhớ dữ liệu cần thiết và các yêu cầu khác như các kênh PWM, truyền thông qua RS232, USB, CAN, LIN...

Bước 2: Lựa chọn các linh kiện ngoài.

Các tiêu chí chính lựa chọn linh kiện khi thiết kế là: Số lượng đầu vào/ra, số lượng bộ đếm/định thời, số lượng, số bit và tốc độ biến đổi của các bộ biến đổi A/D, D/A. Nếu mạch thiết kế cần dùng đến bộ vi điều khiển thì ta cần quan tâm đến kiểu đóng vỏ (liên quan đến việc thiết kế mạch in), hiệu năng của bộ vi điều khiển: 8 bit/16 bit hay 32 bit, tần số xung nhịp bao nhiêu MHz, dung lượng bộ nhớ chương trình, bộ nhớ số liệu. Công suất tiêu thụ, điều này đặc biệt quan trọng với các ứng dụng sử dụng pin, ắc quy.

Bước 3: Xây dựng sơ đồ nguyên lý.

Sau bước 2 ta đã chọn lựa được vi điều khiển và các linh kiện ngoài phụ trợ ta tiến hành thiết kế mạch nguyên lý cho ứng dụng, việc thiết kế có thể được thực hiện bằng các phần mềm hỗ trợ như ORCAD, PROTEL, ALTIUM...

Bước 4: Lập trình cho vi điều khiển và biên dịch chương trình.

Căn cứ vào yêu cầu của bài toán tiến hành lập lưu đồ thuật toán và viết chương trình điều khiển.

Sử dụng trình dịch để biên dịch tập tin nguồn (tập tin .asm hoặc .c hoặc .cpp) thành tập tin .hex theo chuẩn của Intel. Đối với vi điều khiển họ 8051 ta có thể dùng trình dịch Keil C51 của Keil để biên dịch các tập tin nguồn viết bằng ngôn ngữ C. Đối với vi điều khiển họ AVR ta có thể dùng AVR Studio hoặc CodeVisionAVR để soạn và biên dịch chương trình.

Bước 5: Chạy thử nghiệm trên máy tính.

Sử dụng các phần mềm hỗ trợ thiết kế ứng dụng có linh kiện lập trình như Altium, Proteus... để chạy mô phỏng ứng dụng sau khi đã nhập sơ đồ nguyên lý và tập tin .hex cho vi điều khiển.

Nếu kết quả chạy thử nghiệm trên máy tính không đáp ứng được các yêu cầu bài toán thì phải quay lại bước 4 để xem lại chương trình, nếu không phải do chương trình phải quay lại kiểm tra từ bước 3.

Bước 6: Thiết kế, làm mạch in và lắp ráp linh kiện trên mạch in

Bước 7: Nạp chương trình vào vi điều khiển

Việc nạp chương trình cho vi điều khiển thực chất là ghi các thông tin trong tập tin .hex vào bộ nhớ chương trình của vi điều khiển. Việc nạp chương trình vào vi điều khiển thường được thực hiện bằng bộ lập trình chuyên dụng (Programmer). Nếu vi điều khiển hỗ trợ khả năng lập trình trực tiếp trên hệ thống (ISP, SPI) thì ta có thể nạp chương trình cho vi điều khiển ngay trên bản mạch ứng dụng đã gắn vi điều khiển. Một cách đơn giản, để thực hiện việc nạp chương trình xuống vi điều khiển có hỗ trợ ISP như 89S51, 89S52 bạn đọc có thể sử dụng chương trình và mạch nạp qua cổng song song của máy tính trên trang web www.kmitl.ac.th/~kswichit/ISP-Pgm3v0/ISP-Pgm3v0.html

Bước 8: Thiết kế giao diện và viết chương trình giao tiếp với máy tính

Có thể viết chương trình bằng ngôn ngữ Visual Basic hoặc có thể viết bằng ngôn ngữ lập trình C, Pascal hay Assembly.

Bước 9: Chạy thử nghiệm hệ thống.

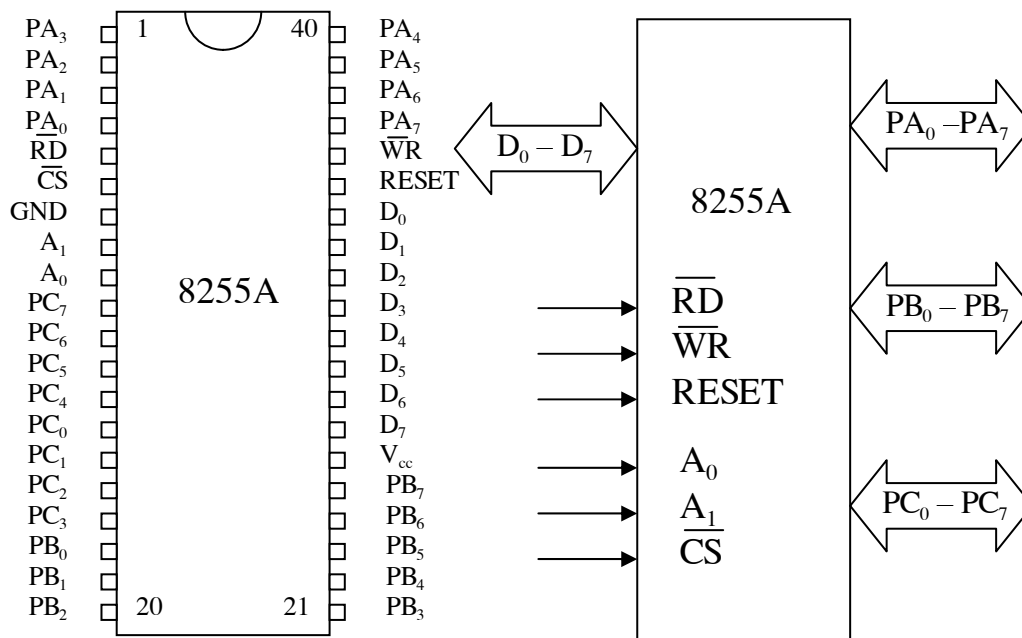
Việc chạy thử nghiệm hệ thống để kiểm tra xem hệ thống được thiết kế, lắp đặt có hoạt động và đạt các yêu cầu đề ra hay không.

Nếu trong ứng dụng không cần dùng vi điều khiển thì bỏ qua các bước 4, bước 5 và bước 7.

4.2 Các vi mạch số thông dụng

4.2.1 Vi mạch ghép nối vào ra song song theo chương trình 8255A

4.2.1.1 Sơ đồ cấu trúc, sơ đồ chân vi mạch 8255A



Hình 4.1 Sơ đồ chân, sơ đồ logic của 8255A

$D_0 - D_7$: BUS dữ liệu (hai chiều).

Reset: Cho phép khởi tạo trạng thái ban đầu cho vi mạch.

\overline{CS} (Chip Select): Tín hiệu chọn vi mạch

\overline{RD} (Read): Tín hiệu cho phép đọc

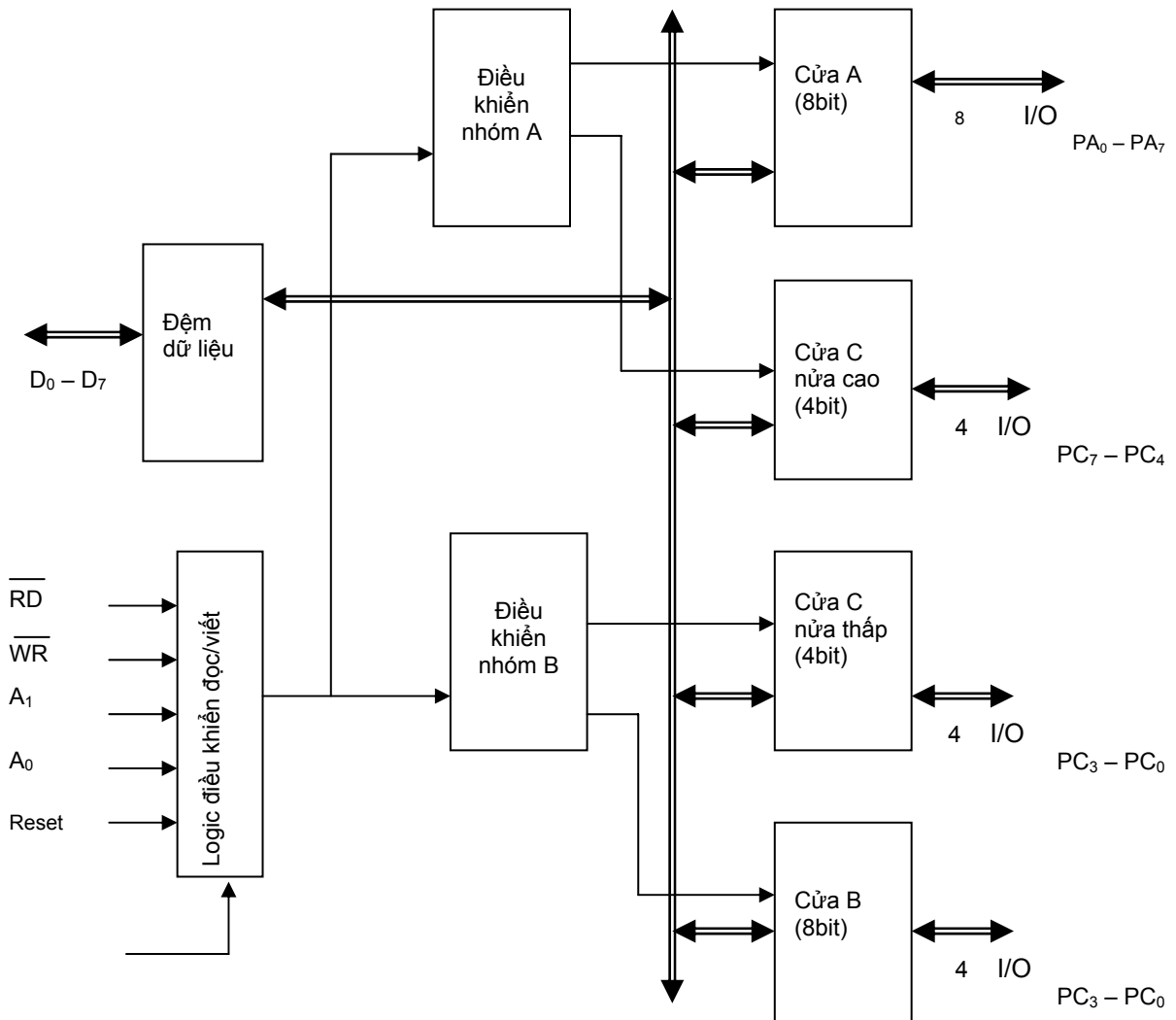
\overline{WR} (Write): Tín hiệu cho phép ghi

$A_0 - A_1$: Tín hiệu địa chỉ

$PA_7 - PA_0$: Cổng A

PB₇ – PB₀: Cổng B

PC₇ – PC₀: Cổng C



Hình 4.2 Sơ đồ khối vi mạch 8255A

Phần ghép nối với vi xử lý hoặc máy tính bao gồm:

- Bộ đệm dữ liệu để trao đổi tin về dữ liệu hai chiều (vào/ra) giữa đường dây của vi xử lý và đường dây BUS nội của 8255A.

- Bộ logic điều khiển đọc/viết: Là bộ giải mã địa chỉ lệnh do các thanh ghi đếm và thanh ghi điều khiển.

Phần ghép nối với thiết bị ngoài bao gồm:

- Cổng A: Thanh ghi đếm dữ liệu (8 bit) vào/ra tùy theo chương trình khởi phát.
- Cổng B: Thanh ghi đếm số liệu (8 bit) vào/ra tùy theo chương trình khởi phát.

- Cổng C: Nửa cao (4 bit).

- Cổng C: Nửa thấp (4 bit).

Tùy theo chế độ sử dụng ghi bởi từ điều khiển cổng C có thể được dùng:

- Trao đổi dữ liệu vào hoặc ra.

- Điều khiển hoặc đối thoại với thiết bị ngoài và vi xử lý khi cổng A và B ở chế độ 0 bằng cách xác lập và xoá từng bit PC_i .

- Điều khiển hoặc đối thoại với thiết bị ngoài và vi xử lý khi cổng A và B ở chế độ 1 và 2.

Phần các mạch điều khiển nội bộ

Có các khối điều khiển nhóm A, nhóm B; khối điều khiển các cổng A, B và C.

4.2.1.2. Các lệnh ghi/đọc các cổng và thanh ghi điều khiển

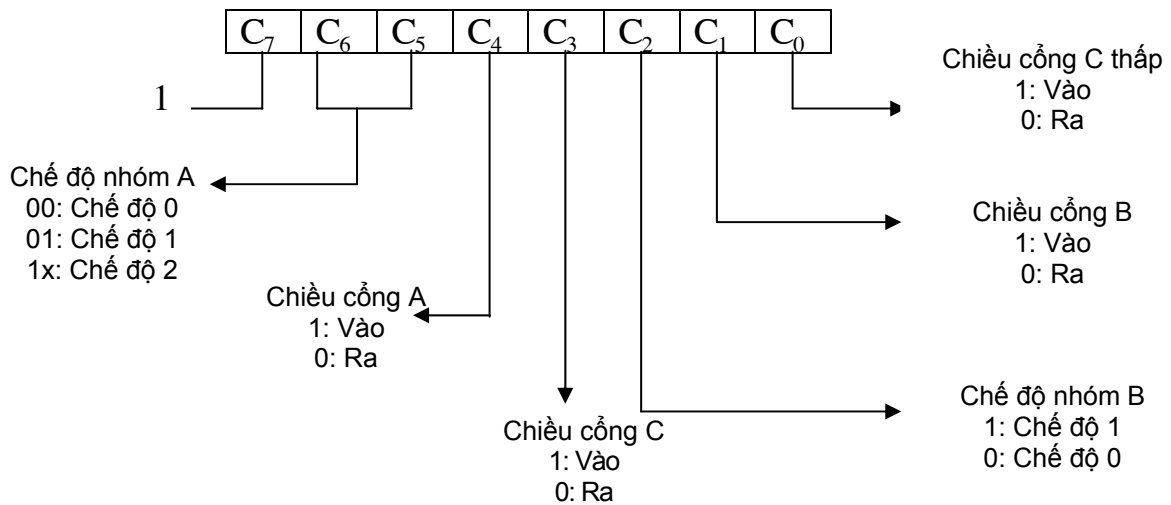
Với tổ hợp của các tín hiệu địa chỉ (A_0A_1), tín hiệu chọn vi mạch \overline{CS} , các tín hiệu đọc \overline{RD} , ghi \overline{WR} của vi xử lý, ta có thể có các lệnh ghi và đọc khác nhau cho các cổng (A, B, C) và thanh ghi điều khiển tạo ra sự di chuyển dữ liệu giữa đường dây dữ liệu, các cổng và thanh ghi điều khiển. Như vậy, vi mạch 8255A có đặc điểm là không có lệnh đọc thanh ghi trạng thái mà dùng lệnh đọc cổng C khi vi mạch ở chế độ 1 và 2, còn ở chế độ 0, không đọc được trạng thái.

Bảng 4.1 Logic ghi/đọc của 8255A

A_1	A_0	\overline{CS}	\overline{RD}	\overline{WR}	Lệnh (của VXL)	Chiều di chuyển dữ liệu
0	0	0	0	1	Đọc cổng A	Cổng A $\rightarrow D_0 - D_7$
0	1	0	0	1	Đọc cổng B	Cổng B $\rightarrow D_0 - D_7$
1	0	0	0	1	Đọc cổng C	Cổng C $\rightarrow D_0 - D_7$
1	1	0	0	1		Không có giá trị
0	0	0	1	0	Ghi cổng A	$D_0 - D_7 \rightarrow$ Cổng A
0	1	0	1	0	Ghi cổng B	$D_0 - D_7 \rightarrow$ Cổng B
1	0	0	1	0	Ghi cổng C	$D_0 - D_7 \rightarrow$ Cổng C
1	1	0	1	0	Ghi thanh ghi điều khiển	$D_0 - D_7$ thanh ghi điều khiển
x	x	1	x	x	Vi mạch ở trạng thái điện trở cao	Không có trao đổi dữ liệu

4.2.1.3 Các chế độ

Tùy giá trị ghi vào thanh ghi điều khiển khi khởi tạo, vi mạch có thể hoạt động ở các chế độ 0, 1, 2 khác nhau, chiều của các cổng A,B,C có thể là ra hoặc vào.

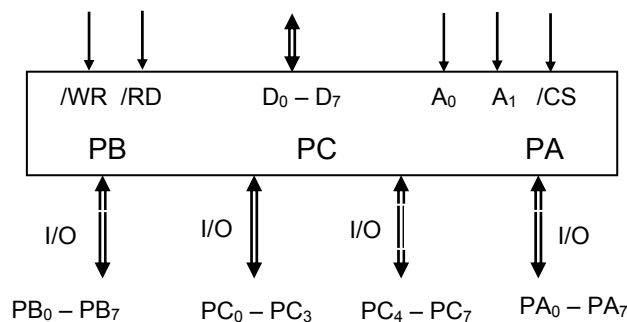


Hình 4.3 Cấu trúc từ điều khiển của 8255A

Chế độ 0.

Chế độ này còn được gọi là chế độ vào/ra cơ sở vì:

- Các cổng A, B và C được sử dụng độc lập với nhau.
- Các cổng A, B và C có thể vào hoặc ra tùy thuộc giá trị của từ điều khiển chế độ ghi vào thanh ghi điều khiển.
- Dữ liệu ra được chốt.
- Dữ liệu vào không được chốt.
- Không có sự đối thoại với thiết bị ngoài.



Hình 4.4 Mô tả hoạt động của 8255A ở chế độ 0

Chế độ 1

Chế độ này còn gọi là chế độ vào/ra có đột cửa hay đối thoại với các bit của cổng C. Các cổng A, B, C được chi thành 2 nhóm:

- Nhóm A gồm cổng A để trao đổi dữ liệu và cổng C cao ($PC_7 - PC_4$) để đối thoại với vi xử lý và thiết bị ngoài.

- Nhóm B gồm cổng B để trao đổi dữ liệu và cổng C thấp ($PC_0 - PC_3$) để đối thoại với vi xử lý và thiết bị ngoài.

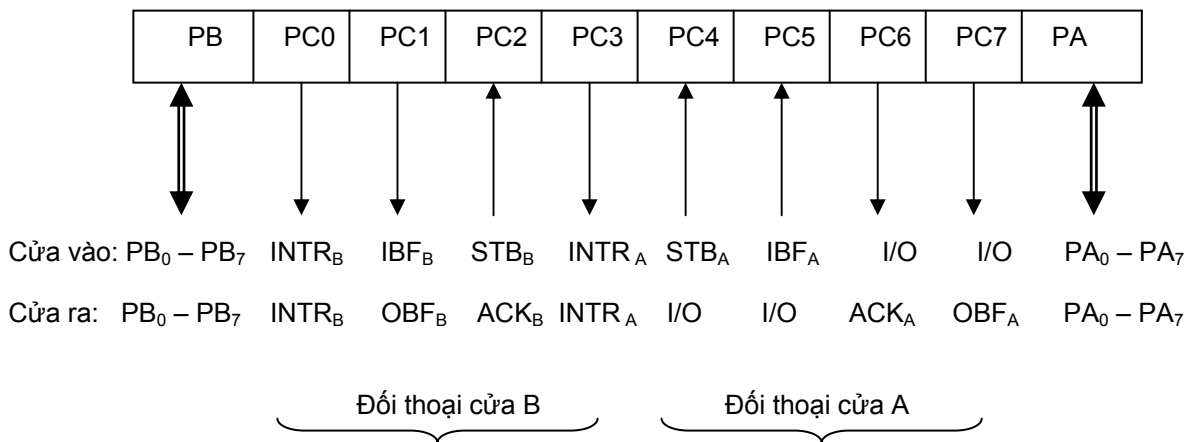
Chiều và chế độ của cổng A, B do từ điều khiển quyết định, còn các tín hiệu đối thoại PC_i còn phụ thuộc chiều cổng vào hay ra.

Ở chế độ 1 ta thấy:

- PC_0 luôn là tín hiệu $INTR_B$ tín hiệu yêu cầu ngắt chương trình cho cổng B.

- PC_3 luôn là tín hiệu $INTR_A$ tín hiệu yêu cầu ngắt chương trình cho cổng A.

- PC_2 luôn là tín hiệu vào, nhận các tín hiệu yêu cầu STB_B và xác nhận ACK_B của thiết bị ngoài cho cổng B tương ứng với chiều vào hay chiều ra. Còn cổng A nếu là cổng vào, PC_4 nhận STB_A của thiết bị ngoài và PC_6 nhận ACK_A của thiết bị ngoài nếu cổng A là cổng ra.



Hình 4.5 Mô tả hoạt động của 8255A ở chế độ 1

Chế độ 2

Chế độ này chỉ dùng cho cổng A với vào, ra thuận nghịch và các bit PC_3 , $PC_4 - PC_7$ dùng làm các tín hiệu đối thoại trong đó:

- PC_3 cho tín hiệu yêu cầu ngắt $INTR_A$ chung cho cả hai chiều và giống chế độ 1.

- PC_4 cho tín hiệu vào STB_A khi cổng A có chiều vào.

- PC_6 cho tín hiệu vào ACK_A khi cổng A có chiều ra.

Chung cả hai chế độ 1 và 2 các bit còn lại dùng làm đối thoại của cổng C đều là các tín hiệu ra:

- IBF_A, IBF_B chỉ dữ liệu vào đã ghi đầy vào các cổng A hoặc B để yêu cầu thiết bị ngoài không đưa dữ liệu vào nữa.

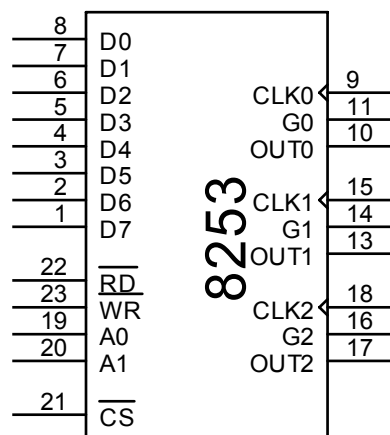
- OBF_A, OBF_B chỉ dữ liệu ra đã ghi đầy vào các cổng A hoặc B để yêu cầu thiết bị ngoài đọc dữ liệu.

4.2.2 Vi mạch đếm định thời lập trình được 8253 (PROGRAMABLE COUNTER AND TIMER).

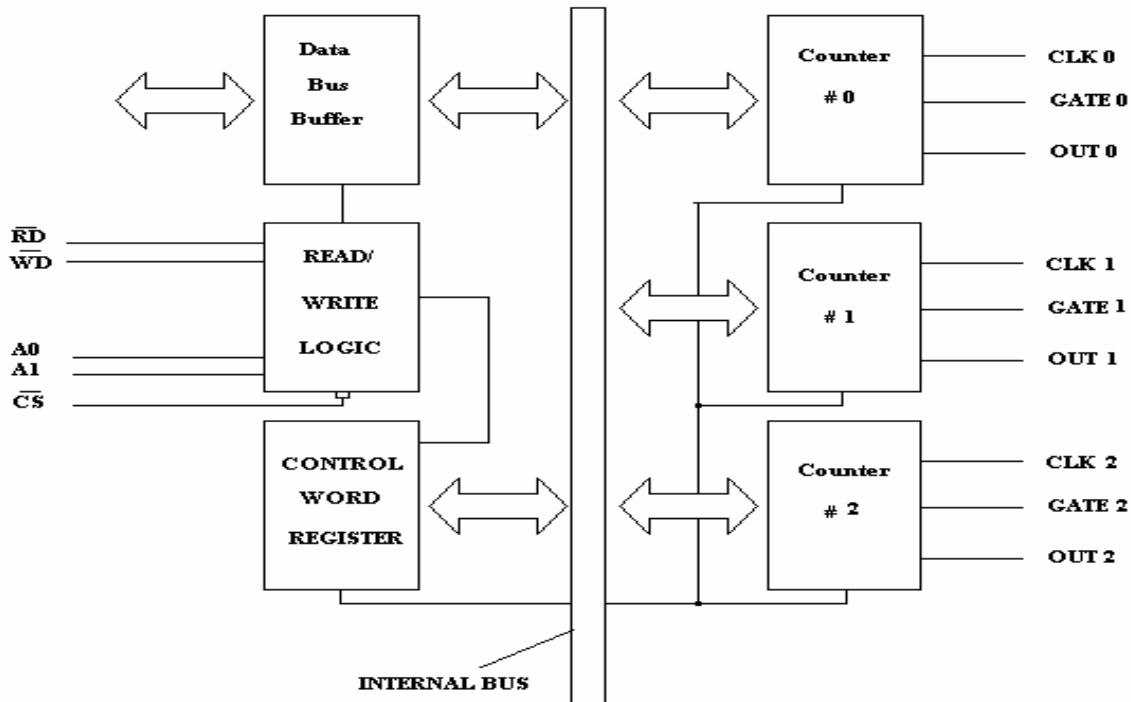
4.2.2.1 Nguyên tắc cấu tạo và hoạt động của 8253.

Vi mạch 8253 là một bộ đếm và định thời được ứng dụng khá rộng rãi. Có thể kể ra một số ứng dụng điển hình: bộ đếm tần số, bộ phát xung lập trình được, bộ đếm xung, thiết bị thử phản ứng, máy đo điện dung, bộ điều khiển độ sáng, bộ biến đổi A-D...

8253 chứa 3 bộ đếm lùi 16 bit độc lập với nhau. Mỗi một bộ đếm có 3 chân tương ứng: lối vào giữ nhịp CLK, lối vào điều khiển GATE và lối tín hiệu ra OUT. Khi hoạt động mỗi bộ đếm cần phải được khởi tạo một giá trị xuất phát, sau mỗi sườn âm ở lối vào giữ nhịp CLK, giá trị của bộ đếm sẽ giảm đi 1. Chân Gate có chức năng điều khiển hoạt động của mỗi bộ đếm, tùy theo chế độ hoạt động của bộ đếm mà các lối ra OUT có dạng tín hiệu khác nhau.



Hình 4.6 Vi mạch 8253



Hình 4.8 Sơ đồ khối của 8253

4.2.2.2 Các lệnh ghi/đọc các bộ đếm và thanh ghi điều khiển

Bảng 4.2 Logic ghi/đọc của 8253

A1	A0	\overline{CS}	\overline{RD}	\overline{WR}	Chức năng
0	0	0	0	1	Đọc ra từ bộ đếm 0
0	1	0	0	1	Đọc ra từ bộ đếm 1
1	0	0	0	1	Đọc ra từ bộ đếm 2
0	0	0	1	0	Nạp giá trị xuất phát cho bộ đếm 0
0	1	0	1	0	Nạp giá trị xuất phát cho bộ đếm 1
1	0	0	1	0	Nạp giá trị xuất phát cho bộ đếm 2
1	1	0	1	0	Nạp từ điều khiển cho thanh ghi đi khiển
X	X	1	X	X	Vi mạch không hoạt động, bus ở trạng thái trở kháng cao
X	X	0	1	1	Vi mạch không hoạt động, bus ở trạng thái trở kháng cao

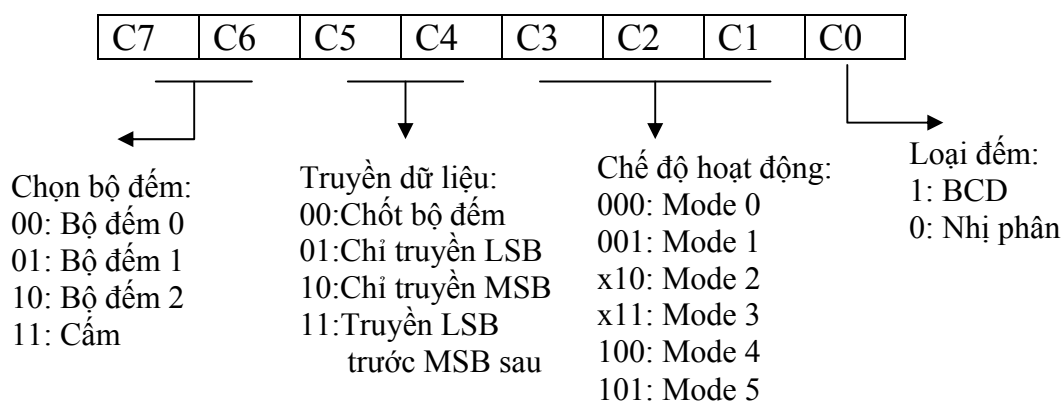
Bộ đếm của 8253 là 16 bit, trong khi bus dữ liệu lại là 8 bit, vì vậy khi khởi tạo bộ đếm bằng một số lớn hơn 255 ta phải truyền byte thấp trước, byte cao sau: Chẳng

hạn như để nạp giá trị xuất phát 50.000 vào một bộ đếm thì trước hết giá trị byte thấp là 80 sau đó byte cao là 195 cần phải được ghi. Phép tính tương ứng như sau: $50.000/256 = 195 \text{ dư } 80$; ở đây 195 là MSB còn 80 là LSB.

Từ điều khiển cũng có thể được tính theo công thức tổng quát sau:

$$\text{Controlword} = 2 \cdot \text{Mode} + 64 \cdot \text{Counternumber} + 48$$

Trong đó: Mode là chế độ hoạt động, counternumber là số thứ tự của bộ counter/timer. Ví dụ: bộ đếm 0 hoạt động ở chế độ 2 thì giá trị của từ điều khiển sẽ là: 52.

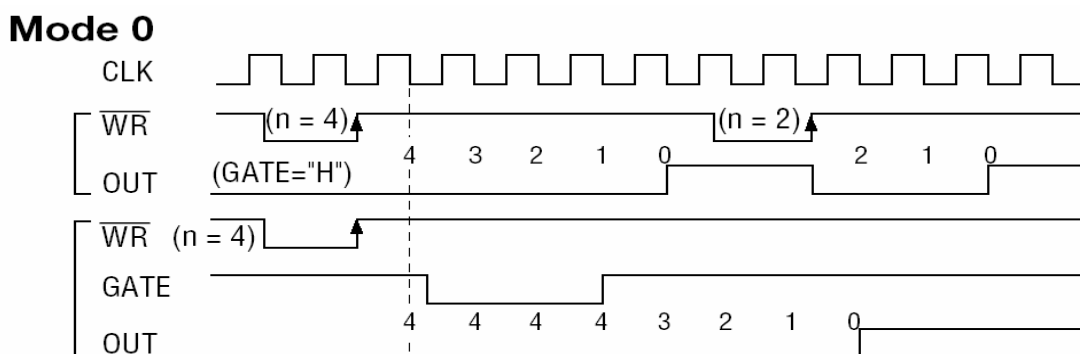


Hình 4.9 Cấu trúc từ điều khiển 8253.

4.2.2.3 Các chế độ

Vi mạch 8253 có 6 chế độ (mode) hoạt động, tùy theo sử dụng 8253 để tạo xung, đếm xung hay định khoảng thời gian mà ta chọn bộ đếm của 8253 làm việc ở chế độ nào cho phù hợp.

Chế độ 0 (mode 0), sườn dương bị làm trễ



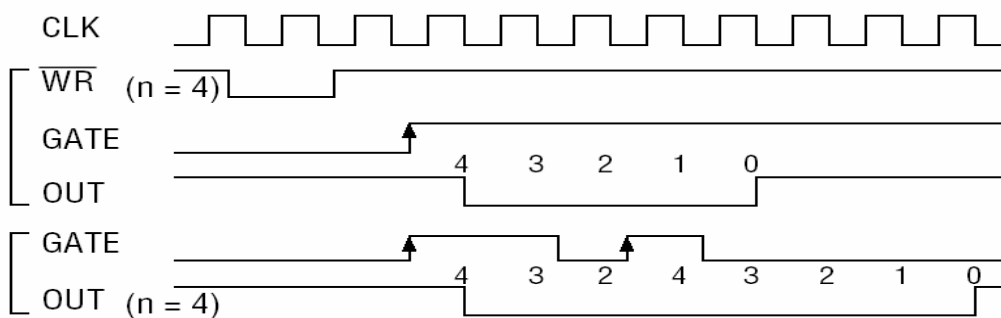
Hình 4.10 Giải đồ xung của chế độ hoạt động 0

Trong kiểu hoạt động này, lối ra OUT bắt đầu ở mức logic thấp ngay sau khi chọn chế độ và giữ nguyên trạng thái trong n chu kỳ giữ nhịp của xung CLK (n là giá trị khởi tạo của bộ đếm), sau đó chuyển sang mức cao khi bộ đếm đạt giá trị 0. Nếu chân điều khiển GATE ở mức thấp, đầu ra OUT ở trạng thái điện trở cao. Chế độ này không tự nạp lại được.

Chế độ 1: Mạch Monoflop lập trình được

Gần giống với kiểu hoạt động 0, nhưng đầu ra bắt đầu ở mức logic cao, sau đó xuống thấp và giữ nguyên trạng thái cho đến khi bộ đếm đạt giá trị 0, sau đó lại trở lại mức cao. Quá trình đếm được bắt đầu bằng sườn lên của xung ở chân điều khiển GATE, sau đó, GATE ở mức cao hay thấp đều không ảnh hưởng tới bộ đếm. Chế độ này không tự nạp lại được.

Mode 1

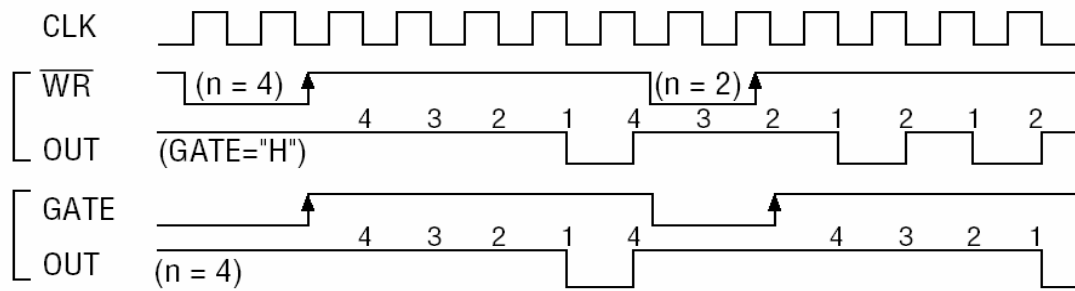


Hình 4.11 Giản đồ xung của chế độ hoạt động 1

Chế độ 2: Bộ chia 1/n

Trong kiểu hoạt động này lối ra OUT chuyển sang mức LOW ở hiện mức 1 của bộ đếm trong khoảng thời gian 1 chu kỳ giữ nhịp. Sau khi đạt tới hiện mức 0 bộ đếm lối ra OUT chuyển lại sang mức HIGH và bộ đếm bắt đầu đếm lùi từ giá trị xuất phát. Đây là điểm khác nhau cơ bản so với kiểu hoạt động 0 và 1, ở kiểu hoạt động này, sau khi đạt hiện mức đếm 0, bộ đếm bắt đầu đếm lùi từ giá trị xuất phát. Ở đây mức độ kéo dài của mức HIGH ở tín hiệu lối ra OUT là $(n-1)$ chu kỳ giữ nhịp, còn mức LOW kéo dài 1 chu kỳ giữ nhịp nên chu kỳ của tín hiệu lối ra OUT bằng n lần tín hiệu CLOCK. Điều đáng chú ý là bộ đếm được khởi động mới bằng mỗi sườn dương ở lối vào GATE. Nhờ vậy mà lối vào GATE có thể được tận dụng để đồng bộ bộ đếm. Bộ đếm có thể tự nạp lại. Trong thực tế phải đặc biệt chú ý là sườn âm đầu tiên ở lối vào CLOCK còn chưa được đếm, mà chỉ có giá trị xuất phát được chấp nhận như là hiện mức hiện thời ở bộ đếm.

Mode 2



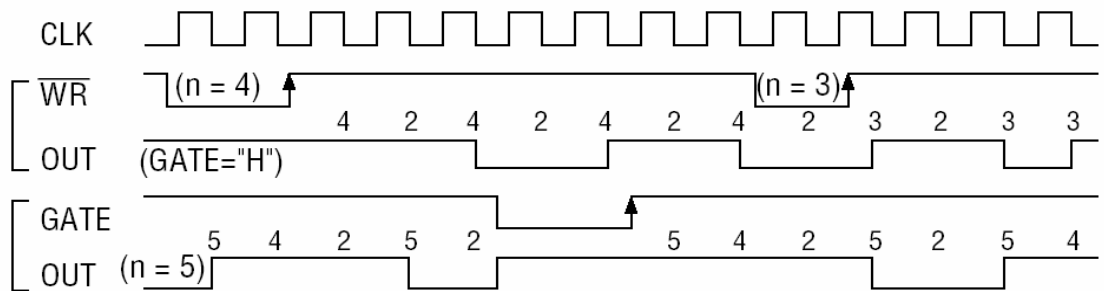
Hình 4.12 Giản đồ xung của chế độ hoạt động 2

Chế độ 3: Bộ phát xung vuông lập trình được:

Trong kiểu hoạt động 3, bộ đếm của 8253 có thể được sử dụng như là một bộ phát tín hiệu vuông lập trình được. Trong kiểu hoạt động 3, tín hiệu lối ra OUT có mức HIGH trong $n/2$ chu kỳ đầu của tín hiệu giữ nhịp CLK và ở mức thấp trong $n/2$ chu kỳ còn lại nếu n là một số chẵn. Trường hợp n là một số lẻ thì tín hiệu lối ra OUT có mức HIGH trong $(n+1)/2$ chu kỳ đầu của tín hiệu giữ nhịp CLK và ở mức thấp trong $(n-1)/2$ chu kỳ còn lại. Bộ đếm có thể tự nạp lại.

Xung nhịp kế sát theo quá trình sẽ giảm bớt hiện mức ở bộ đếm đi 3, còn các xung nhịp tiếp theo lại chỉ giảm bớt đi 2. Quá trình này được lặp lại và nhờ thế đầu ra và trở lên LOW trong $(n-1)/2$ chu kỳ giữ nhịp.

Mode 3

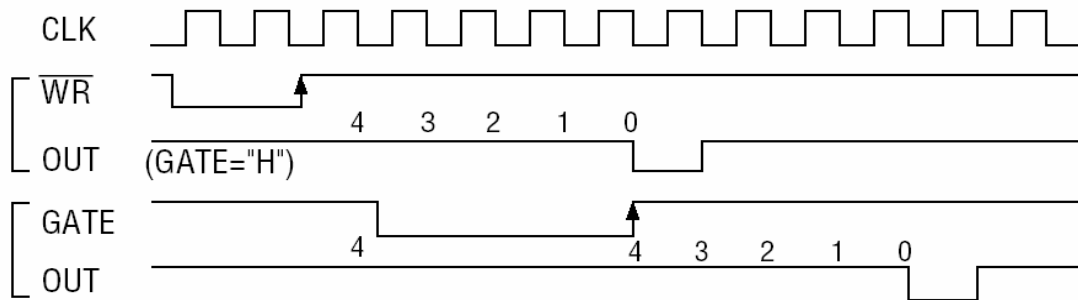


Hình 4.13 Giản đồ xung của chế độ hoạt động 3

Chế độ 4

Ở kiểu hoạt động này lối ra được giữ ở mức cao trong suốt n chu kỳ giữ nhịp của CLK. Khi bộ đếm đạt hiện mức 0, lối ra OUT sẽ ở mức thấp trong 1 chu kỳ kế tiếp của CLK. Bộ đếm không có khả năng tự nạp lại.

Mode 4

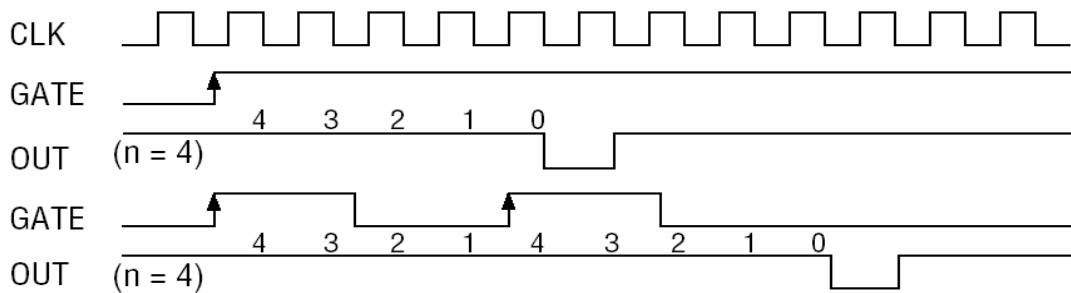


Hình 4.14 Giải đồ xung của chế độ hoạt động 4

Chế độ 5

Trong kiểu hoạt động này bộ đếm bắt đầu quá trình đếm bằng sườn dương của xung ở lối vào GATE và nếu trong quá trình đếm có xung dương ở Gate thì bộ đếm sẽ đếm lại từ đầu, dạng xung ở lối ra OUT giống chế độ 4.

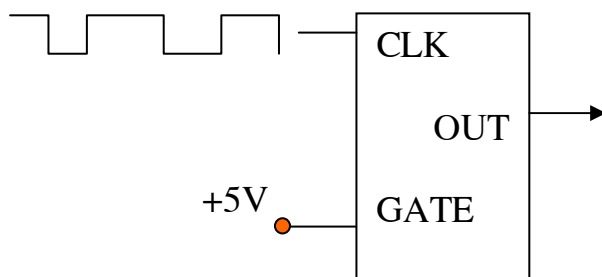
Mode 5



Hình 4.15 Giải đồ xung của chế độ hoạt động 5

4.2.2.4 Một số ứng dụng của 8253.

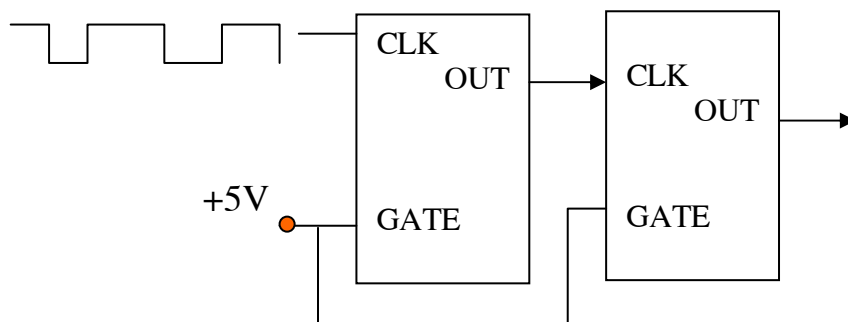
Bộ đếm xung 16 bit



Hình 4.16 Ứng dụng thiết kế bộ đếm xung 16 bit

Giả sử một bộ đếm được đặt ở chế độ 2, ban đầu được nạp một giá trị xuất phát là X_0 . Xung cần đếm được đưa vào lối vào giữ nhịp CLK của bộ đếm. Sau một khoảng thời gian giá trị hiện thời của bộ đếm còn lại là Y_0 . Như vậy số xung đếm được sẽ là $X_0 - Y_0$.

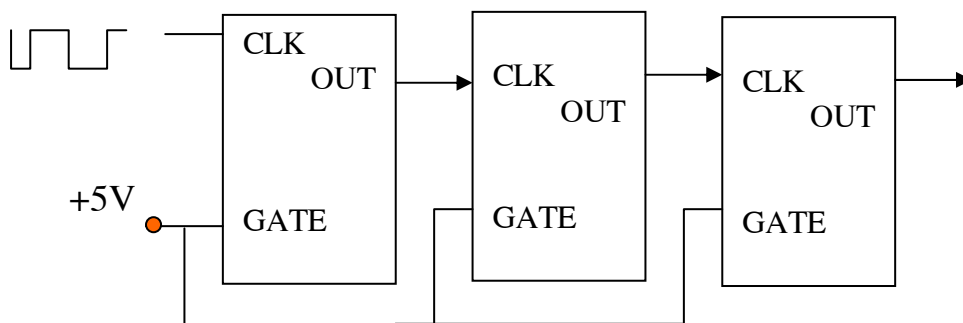
Bộ đếm xung 32 bit



Hình 4.17 Ứng dụng thiết kế bộ đếm xung 32 bit

Giả sử một bộ đếm được đặt ở chế độ 2, ban đầu được nạp một giá trị xuất phát là X_0 . Xung cần đếm được đưa vào lối vào giữ nhịp CLK của bộ đếm. Sau một khoảng thời gian giá trị hiện thời của bộ đếm còn lại là Y_0 . Như vậy số xung đếm được sẽ là $X_0 - Y_0$.

Bộ phát xung lập trình được:



Hình 4.18 Bộ phát xung lập trình được

Trong ứng dụng trên, hai bộ đếm thứ nhất và thứ hai sẽ hoạt động ở chế độ 2 hoặc chế độ 3 đóng vai trò như một bộ chia tần số lập trình được. Tỷ số chia phụ thuộc vào giá trị khởi tạo cho các bộ đếm. Bộ đếm thứ ba đóng vai trò như bộ phát xung, dạng xung ra sẽ tùy thuộc và người lập trình đặt chế độ hoạt động của bộ đếm.

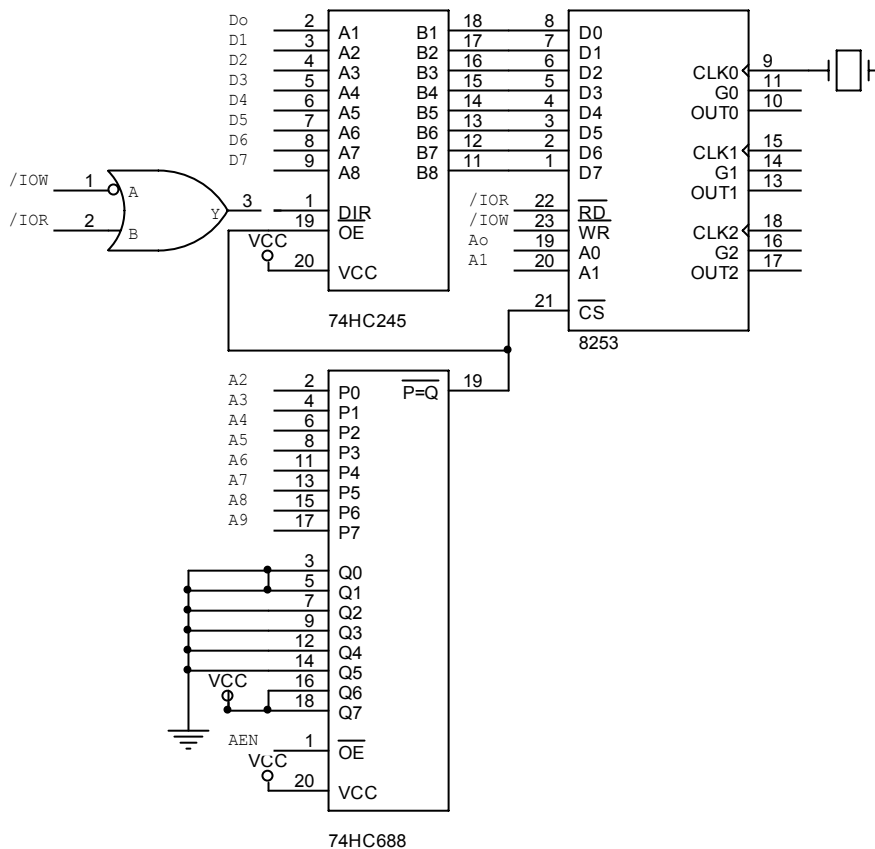
Nếu chỉ dùng 8253 cho mục đích điều khiển thì có thể dùng cổng song song để phối ghép hoặc nếu chỉ dùng 8253 cho mục đích đo lường thì có thể dùng cổng nối

tiếp để phối ghép còn nếu muốn dùng trong cả 2 trường hợp thì khe ISA là tối ưu hơn cả. Hình 4.19 dưới đây là một mạch điện phối ghép 8253 dùng SLOT ISA.

Mạch phối ghép trên sử dụng địa chỉ cơ sở \$300. Các bộ đếm và thanh ghi điều khiển có thể đạt tới bằng cách cộng vào địa chỉ cơ sở các giá trị thích hợp, cụ thể như sau:

- Bộ đếm 0 Địa chỉ cơ sở
- Bộ đếm 1 Địa chỉ cơ sở +1
- Bộ đếm 2 Địa chỉ cơ sở + 2
- Thanh ghi điều khiển Địa chỉ cơ sở + 3

Cũng giống như các vi mạch khả trình khác, muốn 8253 hoạt động trước tiên phải ghi cho nó từ điều khiển, tiếp đó sẽ là giá trị xuất phát của bộ đếm. Đoạn chương trình dưới đây sẽ khởi tạo cho bộ đếm 0 ở chế độ 3, kiểu đếm là đếm mười, truyền byte thấp trước, byte cao sau, giá trị xuất phát là 200.

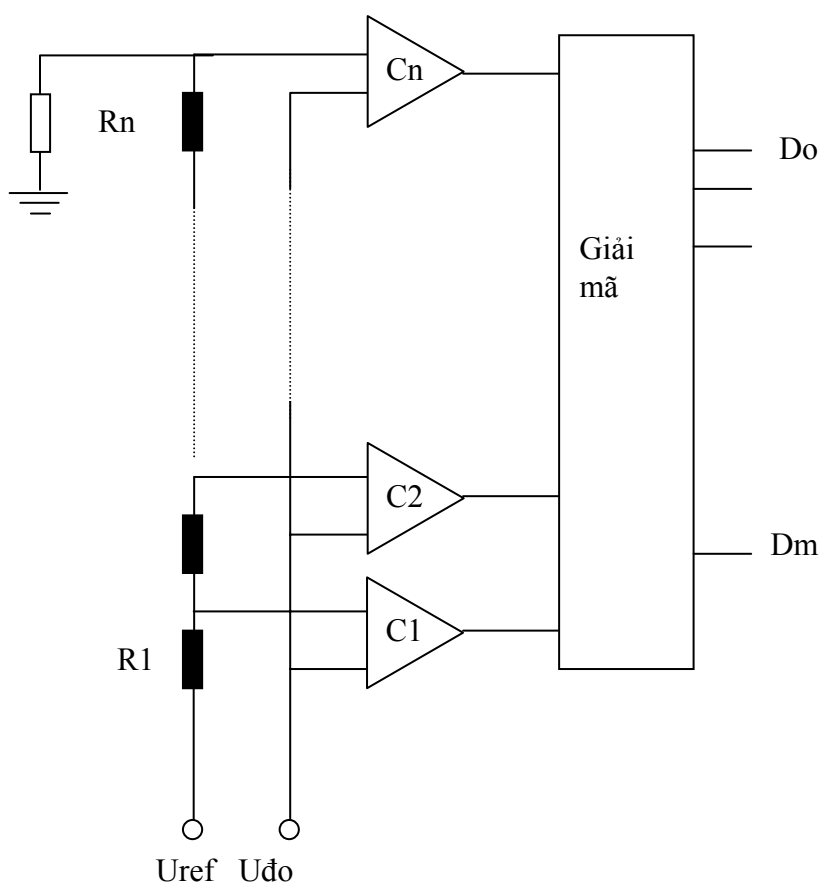


Hình 4.19 Sơ đồ nguyên lý mạch phối ghép 8253 ở SLOT ISA

4.2.3 Các bộ biến đổi AD (ANALOG DIGITAL CONVERTER)

4.2.3.1 Nguyên tắc chung

Hầu hết các bộ AD đều được cấu tạo và hoạt động theo nguyên tắc so sánh gần đúng liên tiếp. Mạng điện trở gồm nhiều điện trở được mắc như hình vẽ sẽ là nhiệm vụ tạo các điện áp so sánh khác nhau đặt lên các bộ so sánh. Trong đó, $U_1 > U_2 > U_3 \dots > U_n$. Đầu còn lại của bộ so sánh sẽ là điện áp cần biến đổi. Các bộ so sánh $C_1, C_2 \dots C_n$ hoạt động theo nguyên tắc nếu “=” thì đưa ra một mức logic “1” ở đầu ra, không “=” thì đưa ra mức logic “0” ở đầu ra. Bộ giải mã nhận các mức logic 1, 0 ở đầu ra của bộ các so sánh chuyển thành các giá trị số (nhị phân) tương ứng.

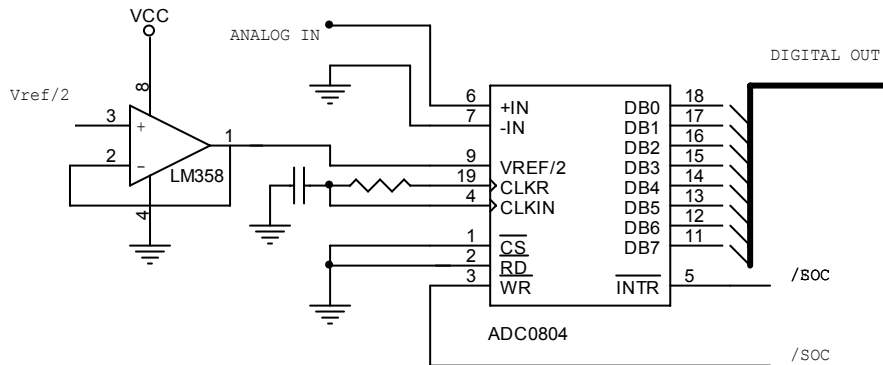


Hình 4.20 Nguyên tắc chung của các bộ AD

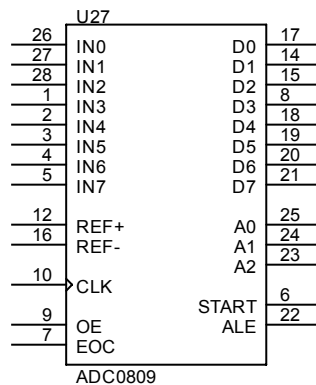
4.2.3.2 Các bộ ADC080X

Các bộ ADC080X có cấu tạo và nguyên tắc hoạt động gần giống nhau. Chúng đều là các bộ AD 8bit hoặc lớn hơn với lối vào Analog vi phân, các tín hiệu ra tương thích TTL. Mỗi bộ đều có bộ phát xung nhịp trên chip và có thể đặt điện áp so sánh ở

chân $V_{ref}/2$ như hình vẽ 4.20. Mỗi chu kỳ biến đổi được bắt đầu bằng một xung Low ở chân $/WR$ và kết thúc một chu kỳ biến đổi có một xung Low sinh ra ở chân $/INTR$. Trong họ ADC080X có bộ ADC0809 có 8 lối vào tương tự có khả năng quét động bằng logic địa chỉ. Bộ này dùng trong trường hợp cần nhiều lối vào tương tự.



Hình 4.21 Bộ ADC0804



Hình 4.22 Sơ đồ chân ADC0809

Để phối ghép các bộ ADC080X vào máy tính có thể ghép nối với cổng nối tiếp, cổng song song hay SLOT ISA.

Hình (4.23) sẽ mô tả một mạch điện phối ghép ADC0804 ở SLOT ISA.

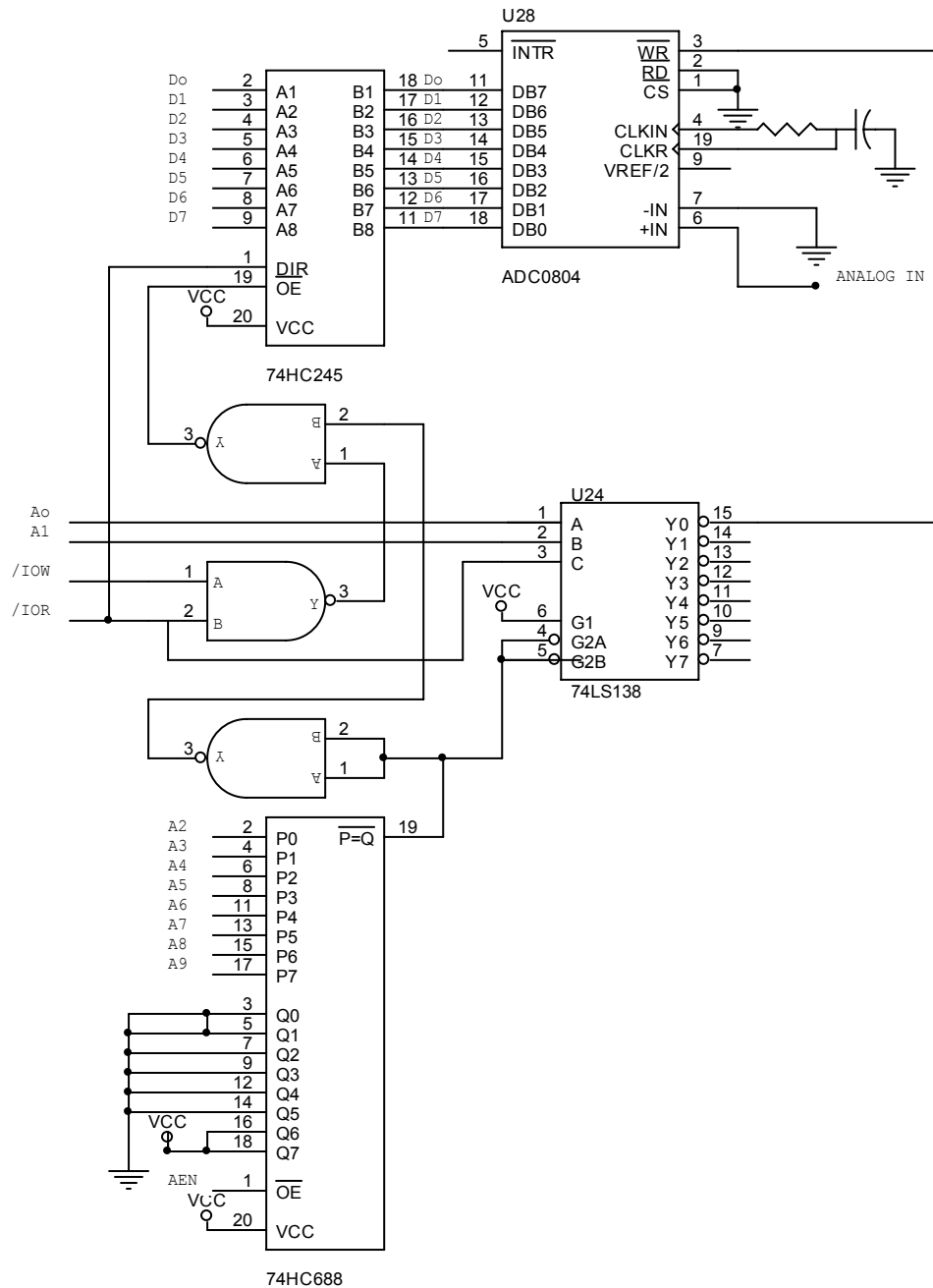
Nếu như ta đặt địa chỉ của 0804 là 300h thì mỗi lần truy nhập sẽ có một xung Low ở $/WR$. Xung này sẽ bắt đầu cho một chu kỳ biến đổi. Sau một khoảng thời gian cỡ 100 μ s tín hiệu số được biến đổi sẽ có ở lối ra $D_0 - D_7$. Để đảm bảo tính chính xác, việc đọc sẽ lặp lại nhiều lần và chia trung bình.

4.2.3.3 Bộ biến đổi AD TCL549.

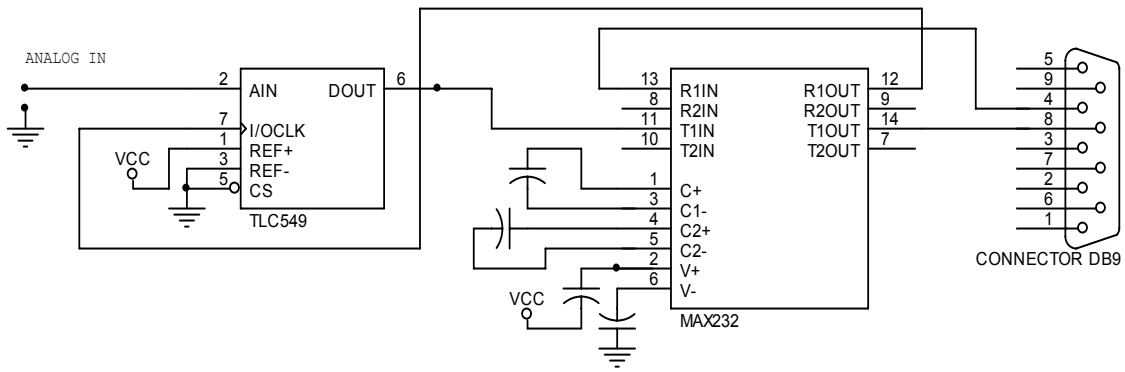
TCL549 là bộ biến đổi AD 8bit với lối ra nối tiếp vì vậy có thể ghép trực tiếp vào cổng nối tiếp (hình 4.24).

Một chu kỳ biến đổi bắt đầu khi có một xung Low ở chân /CS, ngay sau đó sẽ xuất hiện bit D7 ở lối ra Data out, các bit D6-D0 xuất hiện lần lượt sau mỗi sườn âm của xung đến từ I/O clock.

Xung I/O clock có thể tạo bằng cách lập/xóa bit Do (chân DTR) của thanh ghi điều khiển MODEM. Sau mỗi lần tạo một xung âm sẽ phải đọc ngay bit dữ liệu ở chân CTS và sắp xếp thành một byte hoàn chỉnh.



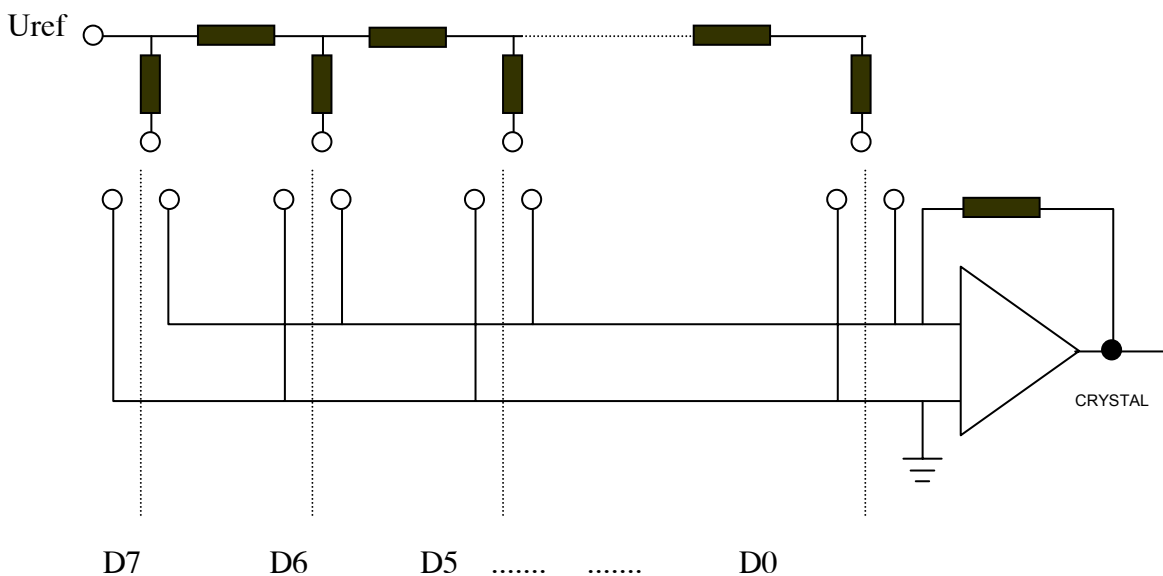
Hình 4.23 Phối ghép ADC 0804 ở SLOT ISA



Hình 4.24 Phối ghép TCL549 ở cổng nối tiếp

4.2.4 Các bộ biến đổi DA (DIGITAL ANALOG CONVERTER)

4.2.4.1 Nguyên tắc chung



Hình 4.25 Nguyên tắc chung của các bộ DAC

Hầu hết các bộ DAC đều có nguyên tắc cấu tạo chung như hình vẽ. Mỗi bộ gồm một số khoá điện tử (số lượng tùy thuộc vào số bit) và một mạng điện trở phân dòng. Các khoá điện tử sẽ lật vị trí tùy thuộc vào bit giá trị số tương ứng là 0 hay 1. Dòng điện tổng sẽ là tổng của các dòng điện ứng với các bit 1.

4.2.4.2 Bộ biến đổi DAC0800

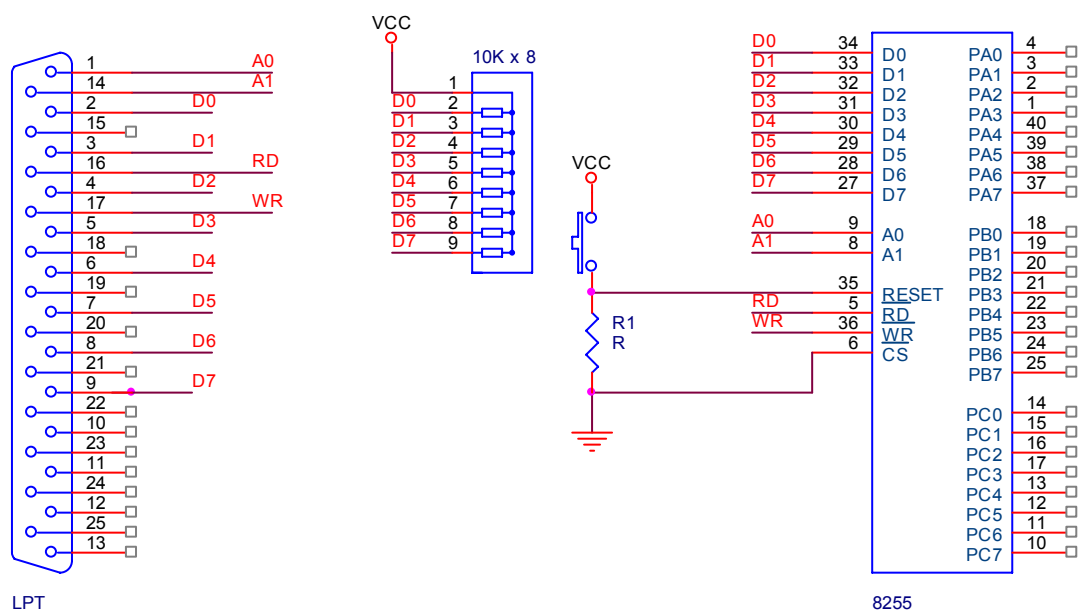
DAC0800 là bộ biến đổi số-tương tự 8 bit có tốc độ biến đổi khá cao (cỡ 100ns) hoạt động theo nguyên tắc trên. Vi mạch này cho phép đặt điện áp tiêu chuẩn ở chân 14 và 15, dòng điện tỷ lệ với điện áp biến đổi được đưa ra từ chân 4.

AD7545 là bộ biến đổi số tương tự với 12 bit, cách hoạt động cũng như DAC0800. Với độ phân giải là 12 bit bộ biến đổi số tương tự này có thể đặt 8191 mức điện áp với độ chính xác tới 0.61 mV nếu như điện áp so sánh là 5 V.

4.3 Các thiết kế ứng dụng ghép nối với máy tính

4.3.1 Điều khiển vi mạch 8255 qua cổng song song

Mạch ghép nối vi mạch 8255 và cổng song song được thiết kế như hình 4.26. Khi nào các đường dẫn ở cổng song song không đủ để sử dụng thì phải sử dụng vi mạch 8255 để tăng đường dữ liệu.



Hình 4.26 Ghép nối 8255 với cổng song song

Việc điều khiển 8255A bằng cổng song song phải được thực hiện theo kiểu giả điều khiển. Muốn ghi ra thanh ghi điều khiển của 8255 phải thực hiện 4 bước sau:

- Tích cực mức logic thích hợp cho các chân tín hiệu điều khiển: $\overline{CS} = 0$, $\overline{RD} = 1$, $\overline{WR} = 1$, A1A0 = 11
- Đưa dữ liệu lên BUS (D0 – D7).
- Tích cực mức thấp cho \overline{WR} .
- Trả lại mức cao cho \overline{WR} .

Muốn ghi ra các cổng PA, PB, PC thì cũng phải làm 4 bước như trên nhưng trước tiên phải ghi ra thanh ghi điều khiển để đặt chế độ hoạt động và chiều cho các cổng của 8255

Muốn đọc các cổng PA, PB, PC vào máy tính phải thực hiện các bước sau:

- Tích cực mức logic thích hợp cho các chân tín hiệu điều khiển: $\overline{CS} = 0$, $\overline{RD}=1$, $\overline{WR} = 1$, A1, A0 = 00
- Tích cực mức thấp cho \overline{RD} .
- Đọc dữ liệu từ BUS (D0 – D7).
- Trả lại mức cao cho \overline{RD} .

Ghi hay đọc phụ thuộc vào chân \overline{RD} và \overline{WR} , ghi đọc cổng nào phụ thuộc giá trị của A1, A0

Chương trình sau đây điều khiển ghi ra cổng PA, PB, PC giá trị nhập trong các ô text1, text2, text3.

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal Value As Integer)
Private Declare Sub Sleep Lib "kernel32" _
(ByVal dwMilliseconds As Long)

Private Sub Command1_Click()
'ghi ra cong PA gia tri trong text1
Out &H37A, &H7
Out &H378, Val(Text1.Text)
Out &H37A, &HF
Sleep (100)
Out &H37A, &H7
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Command3click()
'ghi ra cong PB gia tri trong text2
Out &H37A, &H6
Out &H378, Val(Text2.Text)
Out &H37A, &HE
```

```

Sleep (100)
Out &H37A, &H6
End Sub

Private Sub Command2_Click()
'ghi ra cong PC gia tri trong text3
Out &H37A, &H5
Out &H378, Val(Text3.Text)
Out &H37A, &HD
Sleep (100)
Out &H37A, &H5
End Sub

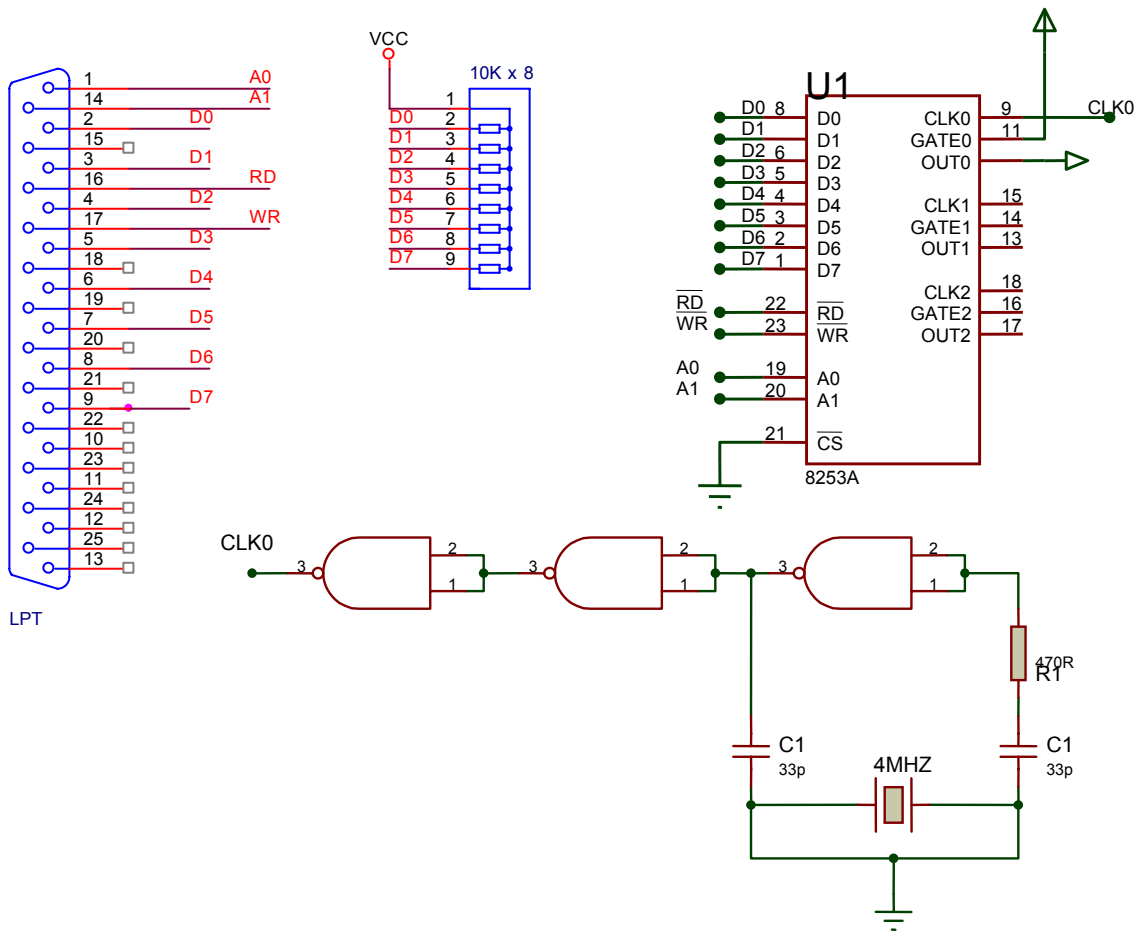
Private Sub Form_Load()
'ghi ra thanh ghi dieu khien 80hexa
Out &H37A, &H4
Out &H378, Val(Text3.Text)
Out &H37A, &HC
Sleep (100)
Out &H37A, &H4
End Sub

```

Hình 4.27 Giao diện chương trình xuất dữ liệu ra cổng PA, PB, PC

4.3.2 Điều khiển vi mạch 8253 qua cổng song song

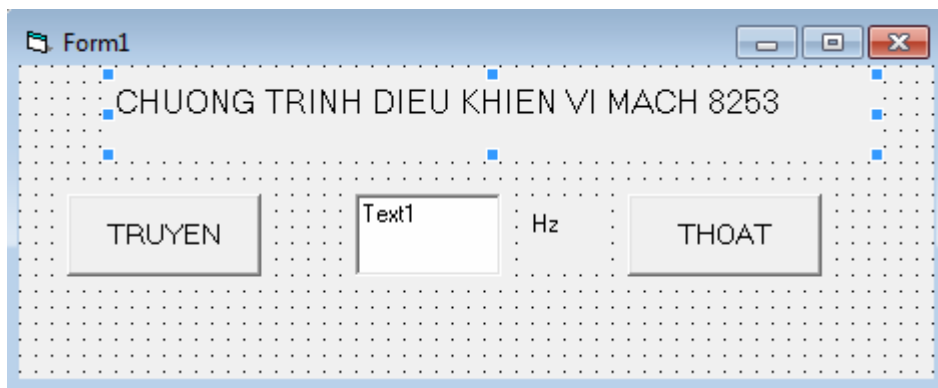
Mạch ghép nối 8253 với cổng song song như hình 4.28



Hình 4.28 Mạch ghép nối vi mạch 8253 với cổng song song

Việc điều khiển 8255A bằng cổng song song phải được thực hiện theo kiểu giả điều khiển giống như vi mạch 8255.

Giao diện hình 4.29 lập trình cho Timer 0 hoạt động ở chế độ 3, tạo xung vuông ở đầu ra OUT0 với giá trị tần số được chia từ bộ tạo dao động thạch anh 4Mhz đưa vào chân CLK1, giá trị tần số cần tạo ra nhập ở ô text1. Timer của vi mạch 8253 có 16 bit, nếu chỉ dùng Timer0 để chia tần thì ta chỉ có thể tạo ra tần số với dải tần khoảng từ 62Hz đến 4Mhz. Nếu ghép nối tiếp 3 timer với nhau để chia tần thì có thể tạo ra khoảng thời gian chính xác đến vài giờ thập chí vài ngày.



Hình 4.29 Giao diện chương trình tạo xung nhập từ bàn phím

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal Value As Integer)
Private Declare Sub Sleep Lib "kernel32" _
(ByVal dwMilliseconds As Long)
```

```
Private Sub Command1_Click()
'ghi ra thanh ghi dieu khien &H37
Out &H37A, &H4
Out &H378, &H37
Out &H37A, &HC
Sleep (100)
Out &H37A, &H4
'ghi ra timer0 byte thap, byte cao
Out &H37A, &H7
Out &H378, 4 * 10 ^ 6 Mod Val(Text1.Text)
Out &H37A, &HF
Sleep (100)
Out &H37A, &H7
Out &H378, 4 * 10 ^ 6 \ Val(Text1.Text)
Out &H37A, &HF
Sleep (100)
Out &H37A, &H7
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

4.3.3 Mạch đếm sản phẩm qua cổng RS-232

Hầu hết các thiết kế các ứng dụng qua cổng RS-232 đều phải sử dụng vi điều khiển làm khối trung gian chuyển tiếp tín hiệu nối tiếp sang song song và ngược lại. Trong ứng dụng này thiết kế hai cặp cảm biến hồng ngoại đưa vào các chân INT0 và INT1, vi điều khiển có chức năng đếm số xung đưa vào qua chân INT0 và INT1 sau đó gửi giá trị này lên máy tính.

Dưới đây là chương trình nạp cho vi điều khiển viết bằng ngôn ngữ lập trình C, sơ đồ nguyên lý và chương trình giao tiếp với máy tính.

```
#include <stdio.h>
#include <reg52.h>
unsigned int i;
unsigned char M[10];
unsigned int sp0; //san pham dem bang ngat 0
unsigned int sp1; //san pham dem bang ngat 1
```

```

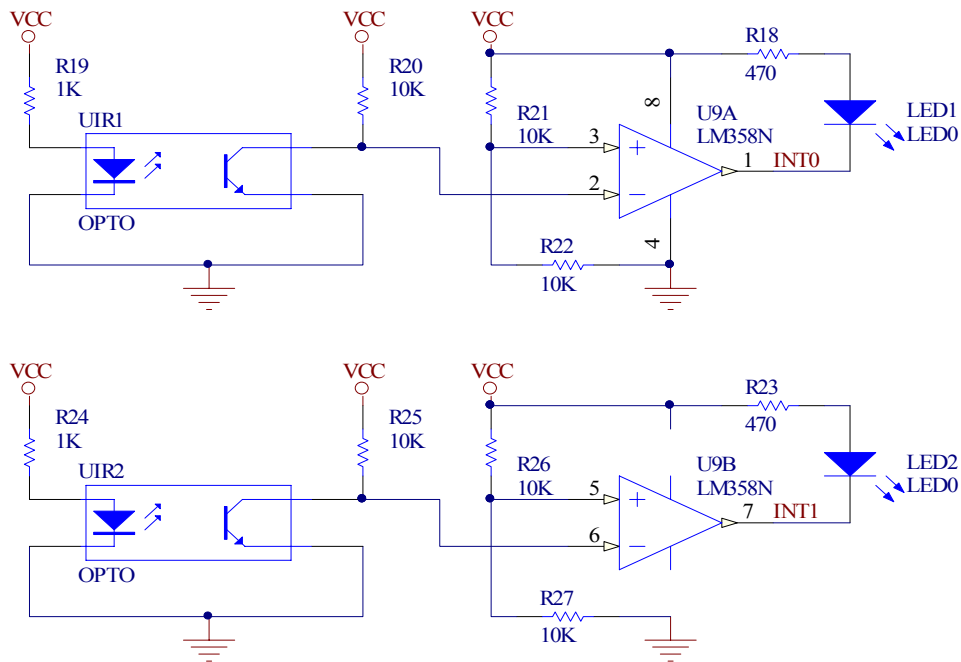
void NGATINT0 (void) interrupt 0
{
unsigned int j;
for(j=0;j<=500;++j);
++sp0;
}
void NGATINT1 (void) interrupt 2
{
unsigned int k;
for(k=0;k<=500;++k);
++sp1;
}
void main (void)
{
unsigned char x;
// khoi tao uart 8 bit toc do 9600baud,1 bit dung
// stick=break=patity=0
SCON=0x52;
TMOD=0x20;
TH1=TL1=-3;
TR1=1;
do
x=_getkey();
while(x!='@');
for(i=1;i<=3;++i) //nhan 3 byte tu may tinh
M[i]=_getkey();
if((M[1]=='I') && (M[2]=='R') && (M[3]=='U')) //3 byte la IRU
{
IE=0x81;
IT0=1;
while(1)
{ putchar('a');
putchar(sp0%256);
putchar(sp0/256);
}
}
if((M[1]=='I') && (M[2]=='R') && (M[3]=='1')) //3 byte la IR1
{
IE=0x84;
IT1=1;
while(1)
{
putchar('a');
putchar(sp1%256);
putchar(sp1/256);
}
}
if((M[1]=='2') && (M[2]=='I') && (M[3]=='R')) //3 byte la 2IR
{
IE=0x85;
IT0=IT1=1;
while(1)

```

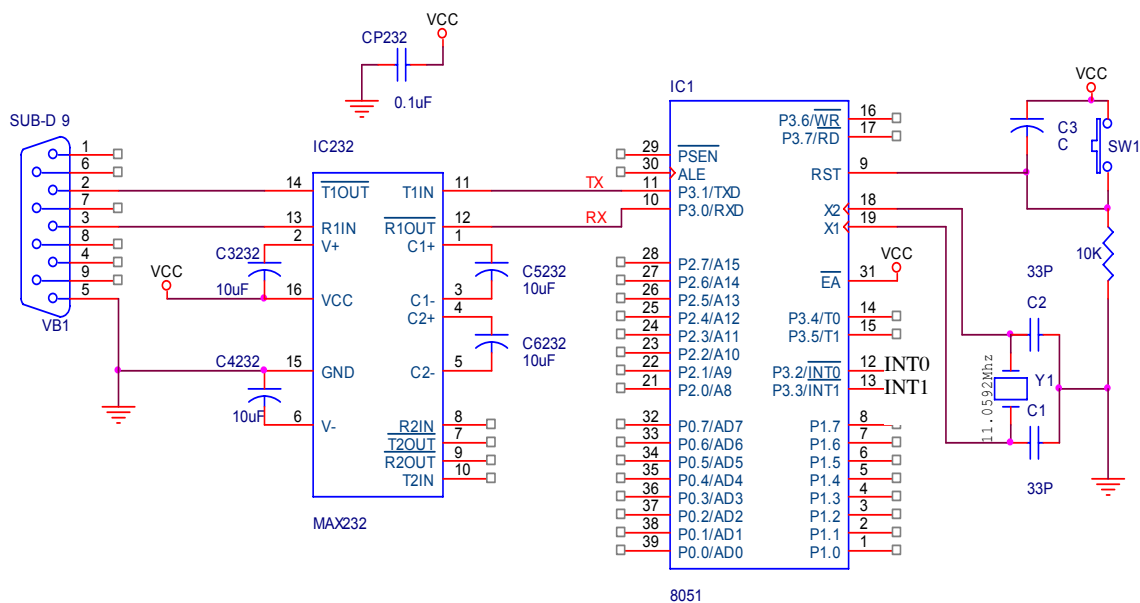
```

{
  putchar('x');      //gui so sp dem duoc o cam bien 0
  putchar(sp0%256);
  putchar(sp0/256);
  putchar(sp1%256); //gui so sp dem duoc o cam bien 1
  putchar(sp1/256);
}
}
}

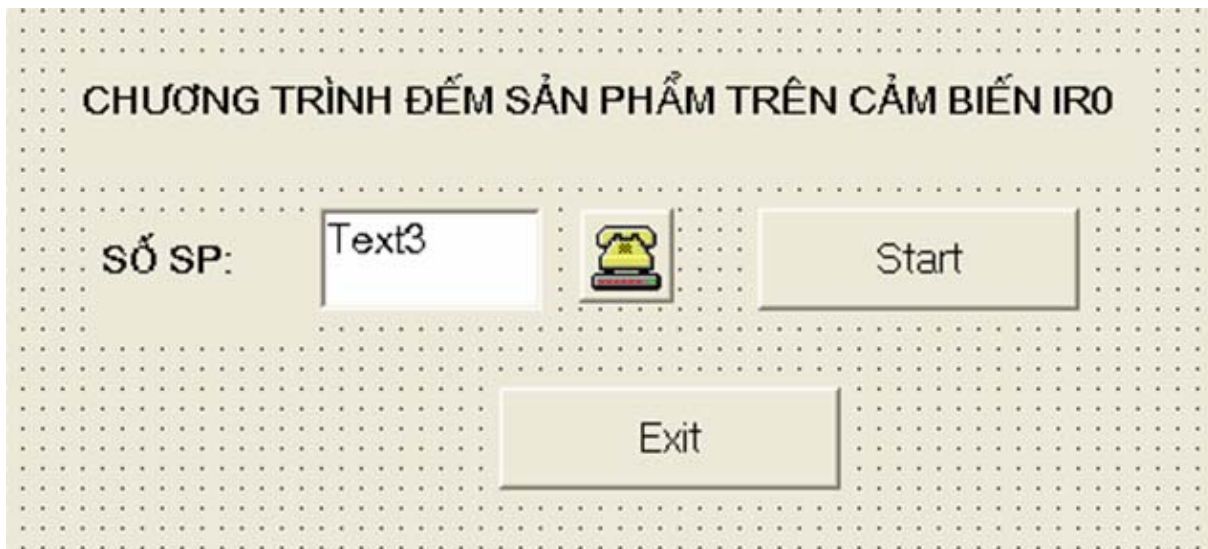
```



Hình 4.30 Sơ đồ mạch cảm biến hồng ngoại



Hình 4.31 Sơ đồ mạch giao tiếp máy tính qua cổng com



Hình 4.32 Giao diện chương trình đọc sản phẩm trên cảm biến IR0

```

Private Sub Form_Load()
    Dim kt As String
    Text3.Text = " "
    MSComm1.CommPort = 1
    MSComm1.Settings = "9600,N,8,1"
    MSComm1.RThreshold = 1
    MSComm1.SThreshold = 1
    MSComm1.InputLen = 3 'mỗi lần đọc 3 byte
    MSComm1.InputMode = 0 'mode text
    MSComm1.PortOpen = True
End Sub

Private Sub MSComm1_OnComm()
    Dim x, y, z As String
    Dim a, b, c As Integer
    x = y = z = " "
    If MSComm1.CommEvent = comEvReceive Then 'comEvReceive
        x = MSComm1.Input
        If Left(x, 1) = "a" Then
            If Len(x) = 3 Then
                y = Left(Right(x, 2), 1) 'byte thấp
                z = Right(Right(x, 2), 1) 'byte cao
                c = Asc(y) + Asc(z) * 256
                Text3.Text = c
            End If
        End If
        x = y = z = " " 'xoa x,y,z
    End If
End Sub

Sub send(buffer As String)
    If MSComm1.PortOpen = True Then
        MSComm1.Output = buffer
    End If

```

```
End Sub
```

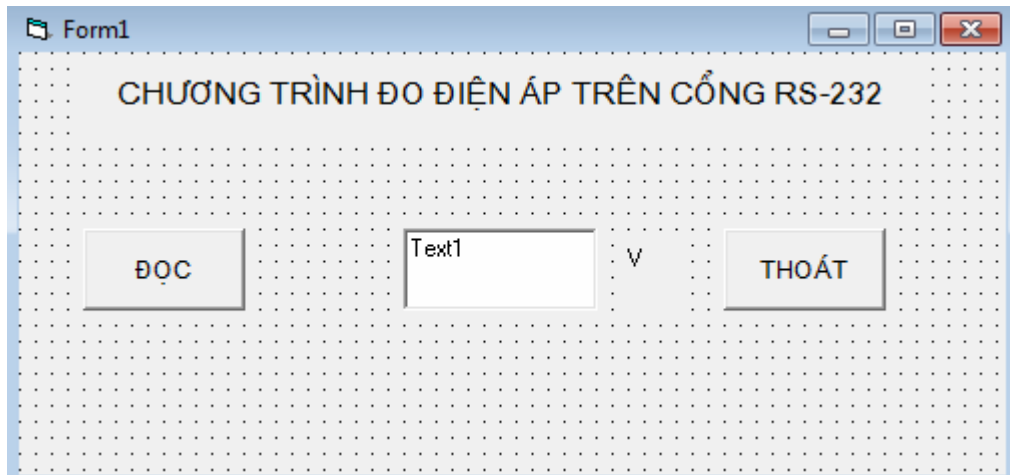
```
Private Sub Command1_Click()      'start
    kt = "@IR0"
    send (kt)
End Sub
```

```
Private Sub Command2_Click()      'exit
If MSCComm1.PortOpen = True Then
MSCComm1.PortOpen = False
End If
End
End Sub
```

4.3.4 Voltmet điện tử ghép nối qua cổng RS-232

Chương trình nạp cho vi điều khiển:

```
#include <stdio.h>
#include <reg52.h>
sbit WR_ADC=P3^6;
unsigned int i;
unsigned char M[10];
unsigned int gtso; //gia tri doc duoc tu ADC
void main (void)
{
    unsigned char x;
    // khoi tao uart 8 bit toc do 9600baud,1 bit dung
    // stick=break=patity=0
    SCON=0x52;
    TMOD=0x20;
    TH1=TL1=-3;
    TR1=1;
    do
    x=_getkey();
    while(x!='@');
    for(i=1;i<=3;++i) //nhan 3 byte tu may tinh
    M[i]=_getkey();
    if((M[1]=='A') && (M[2]=='D') && (M[3]=='C'))
    {
        while(1)
        {
            WR_ADC=0;
            for(i=1;i<100;++i);
            WR_ADC=1;
            for(i=1;i<2000;++i);
            gtso=P2;
            putchar('a');
            putchar(gtso);
        }
    }
}
```



Hình 4.34 Giao diện chương trình đọc điện áp và hiển thị trên máy tính

```

Private Sub Form_Load()
    Dim kt As String
    Text3.Text = " "
    MSComm1.CommPort = 1
    MSComm1.Settings = "9600,N,8,1"
    MSComm1.RThreshold = 1
    MSComm1.SThreshold = 1
    MSComm1.InputLen = 2 'mỗi lần đọc 2 byte
    MSComm1.InputMode = 0 'mode text
    MSComm1.PortOpen = True
End Sub

```

```

Private Sub MSComm1_OnComm()
    Dim x, y As String
    Dim c As Integer
    x = y = " "
    If MSComm1.CommEvent = comEvReceive Then 'comEvReceive
        x = MSComm1.Input
        If Left(x, 1) = "a" Then
            If Len(x) = 2 Then
                y = Right(x, 1) 'dien ap
                c = Asc(y) * 5 / 255
                Text3.Text = c
            End If
        End If
        x = y = " " 'xoa x,y
    End If
End Sub

```

```

Sub send(buffer As String)
    If MSComm1.PortOpen = True Then
        MSComm1.Output = buffer
    End If
End Sub

```

```

Private Sub Command1_Click() 'start

```

```

kt = "@ADC"
send (kt)
End Sub

```

```

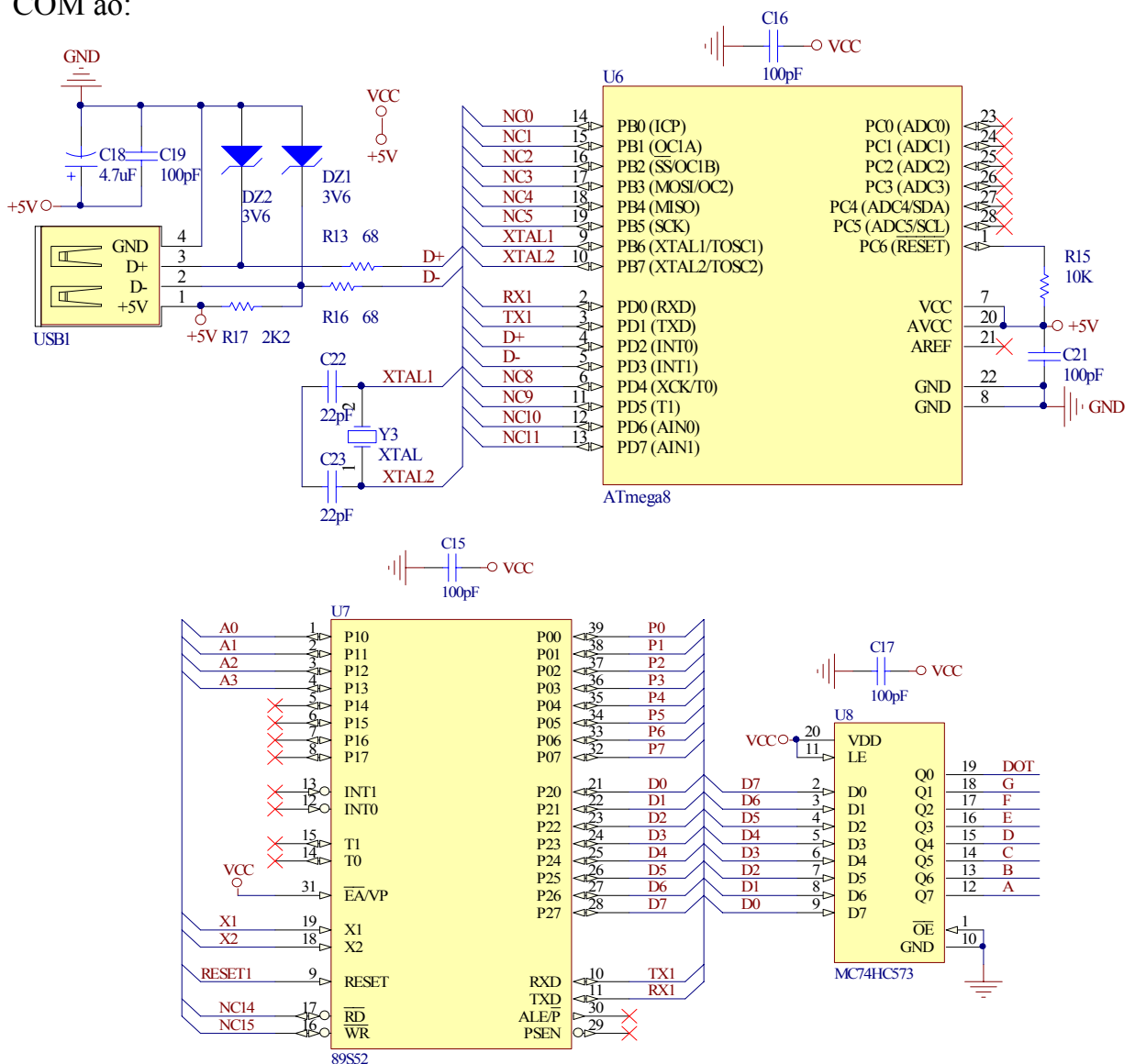
Private Sub Command2_Click() 'exit
If MSComm1.PortOpen = True Then
MSComm1.PortOpen = False
End If
End
End Sub

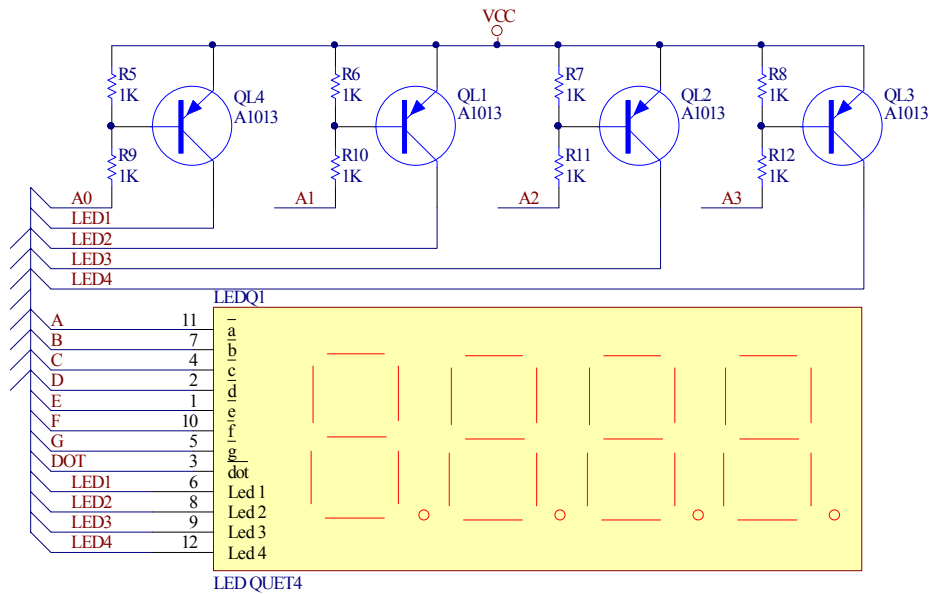
```

4.4.5 Điều khiển Led 7 đoạn qua cổng USB

Phần này giới thiệu cách thiết kế mạch ghép nối qua cổng USB, vì điều khiển trong Modul này có chức năng chuyển cổng USB thành cổng COM (cổng COM ảo), sau đó ta thiết kế mạch và lập trình như đối với cổng COM.

Dưới đây là mạch ghép nối qua cổng USB với Led 7 đoạn thông qua cổng COM ảo:





Hình 4.35 Ghép nối Led 7 đoạn qua cổng USB

Chương trình viết cho vi điều khiển 89S52

```
#include<reg52.h> // khai bao thu vien
#include<stdio.h>
#include<math.h>
sbit LED1=P1^0; // quet led hang don vi
sbit LED2=P1^1; // quet led hang chuc
sbit LED3=P1^2; // quet led hang tram
sbit LED4=P1^3; // quet led hang nghin
unsigned int digit4,digit3,digit2,digit1,i,j,product;
unsigned char temp, index=0;
unsigned char digit[10] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99,
0x92, 0x82, 0xF8, 0x80, 0x90};
void delay(int t) // chuong trinh tao tre
{
for (i=1;i<=t;++i );
}
void timer0 (void) interrupt 1
{
TH0=0xF8;
TL0=0x40;
switch(index)
{
case 0:
LED2=LED3=LED4=1; n
P2=digit[digit1]; // digit1 chinh la don vi
LED1=0; // quet transistor led don vi
break;
case 1:
LED1=LED3=LED4=1;
P2=digit[digit2]; // digit2 chinh la don vi
```

```

        LED2=0;          // LED hang chuc
        break;
    case 2:
        LED1=LED2=LED4=1;
        P2=digit[digit3];
        LED3=0; // led hang tram
        break;
    case 3:
        LED1=LED2=LED3=1;
        P2=digit[digit4];;
        LED4=0; // led hang nghin
        break;

    }
    if(++index==4) index=0;
}
void main() //chuong trinh chinh
{
    SCON=0x52;
    TMOD=0x20;
    TH1=TL1=-3;
    TR1=1;
    TH0=0xF8; // su dung ngat trong Timer0
    TL0=0x30;
    TR0=1; // co cho phep bat dau dem o timer0
    ET0=EA=EX0=1;
    PT0=1; //cho phep uu tien ngat timer0
    temp=0;
    product=0;
    while(1) //cho ngat
    {
        do
        temp=_getkey();
        while(temp!='@');
        temp=_getkey(); //byte thap
        product=temp;
        temp=_getkey(); //byte cao
        product=temp*256 + product;
        digit4=product/1000;
            digit3=(product%1000)/100;
        digit2=((product%1000)%100)/10;
            digit1=((product%1000)%100)%10;
    };
}

```

Chương trình dưới đây truyền các số từ 0 đến 9999 xuống LED.



Hình 4.35 Giao diện chương trình truyền dữ liệu qua com ảo

```

Private Declare Sub Sleep Lib "kernel32" (ByVal
dwMilliseconds As Long)
Dim i As Integer
Private Sub Command1_Click()
For i = 0 To 9999
send ("@" )
send (Chr(i Mod 256)) 'byte thap
send (Chr(i \ 256)) 'byte cao
Sleep (20)
Next
End Sub
Private Sub Command2_Click()
If MSComm3.PortOpen = True Then
MSComm3.PortOpen = False
End If
End
End Sub

Private Sub Form_Load()
MSComm3.CommPort = 3
MSComm3.Settings = "9600,N,8,1"
'MSComm3.RThreshold = 1
MSComm3.InputMode = 1
MSComm3.SThreshold = 1
If MSComm3.PortOpen = False Then
MSComm3.PortOpen = True
End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
If (MSComm3.PortOpen) Then
MSComm3.PortOpen = False
End If
End Sub
End Sub

```

```
Sub send(buffer As String)
    If MSComm3.PortOpen = True Then
        MSComm3.Output = buffer
    End If
End Sub
```

4.4 Bài tập cuối chương

1. Thiết kế mạch đo nhiệt độ qua cổng RS-232, viết chương trình vẽ đồ thị mà nhiệt độ được vẽ bằng các vòng tròn tô màu, các vòng tròn liên tiếp nhau được nối với nhau bằng các line, các vòng tròn cách nhau 144 Pixel, mỗi lần lấy mẫu cách nhau 500ms, khi vẽ hết màn hình xóa và vẽ lại từ đầu. thang đo 0 đến 100 °C
2. Thiết kế mạch Ampemet qua cổng RS-232, viết chương trình vẽ đồ thị mà giá trị dòng điện được vẽ bằng các vòng tròn tô màu, các vòng tròn liên tiếp nhau được nối với nhau bằng các line, các vòng tròn cách nhau 144 Pixel, mỗi lần lấy mẫu cách nhau 500ms, khi vẽ hết màn hình xóa và vẽ lại từ đầu. thang đo 0 đến 100 °C
3. Cho mạch thiết kế trong mục 4.4.5, viết chương trình nhập một số bất kỳ (từ 0 đến 9999) vào khung text1 và truyền số đó xuống hiển thị trên LED 7 đoạn.

Phụ lục

Bảng mã ASCII

Kí tự điều khiển ASCII

Nhị phân	Thập phân	Thập lục	Biểu diễn	Truy nhập (bàn phím)	Tên/Ý nghĩa (tiếng Anh)	Tên/Ý nghĩa (tiếng Việt)
000 0000	0	0	NUL	^@	Null character	Kí tự rỗng
000 0001	1	1	SOH	^A	Start of Header	Bắt đầu Header
000 0010	2	2	STX	^B	Start of Text	Bắt đầu văn bản
000 0011	3	3	ETX	^C	End of Text	Kết thúc văn bản
000 0100	4	4	EOT	^D	End of Transmission	Kết thúc truyền
000 0101	5	5	ENQ	^E	Enquiry	Truy vấn
000 0110	6	6	ACK	^F	Acknowledgement	
000 0111	7	7	BEL	^G	Bell	Chuông
000 1000	8	8	BS	^H	Backspace	Xoá ngược
000 1001	9	9	HT	^I	Horizontal Tab	Tab ngang
000 1010	10	0A	LF	^J	Line feed	
000 1011	11	0B	VT	^K	Vertical Tab	Tab dọc
000 1100	12	0C	FF	^L	Form feed	
000 1101	13	0D	CR	^M	Carriage return	
000 1110	14	0E	SO	^N	Shift Out	
000 1111	15	0F	SI	^O	Shift In	
001 0000	16	10	DLE	^P	Data Link Escape	
001 0001	17	11	DC1	^Q	Device Control 1 - oft. XON	
001 0010	18	12	DC2	^R	Device Control 2	
001 0011	19	13	DC3	^S	Device Control 3 - oft. XOFF	
001 0100	20	14	DC4	^T	Device Control 4	
001 0101	21	15	NAK	^U	Negative Acknowledgement	
001 0110	22	16	SYN	^V	Synchronous Idle	
001 0111	23	17	ETB	^W	End of Trans. Block	
001 1000	24	18	CAN	^X	Cancel	
001 1001	25	19	EM	^Y	End of Medium	
001 1010	26	1A	SUB	^Z	Substitute	
001 1011	27	1B	ESC	^[hay ESC	Escape	
001 1100	28	1C	FS	^\	File Separator	
001 1101	29	1D	GS	^]	Group Separator	
001 1110	30	1E	RS	^^	Record Separator	
001 1111	31	1F	US	^_	Unit Separator	
111 1111	127	7F	DEL	DEL hay Backspace	Delete	

Kí tự ASCII in được

Nhị phân	Thập phân	Thập lục	Đồ hoạ	Nhị phân	Thập phân	Thập lục	Đồ hoạ
010 0000	32	20	Khoảng trống (<code>␣</code>)	110 0010	98	62	b

010 0001	33	21	!	110 0011	99	63	c
010 0010	34	22	"	110 0100	100	64	d
010 0011	35	23	#	110 0101	101	65	e
010 0100	36	24	\$	110 0110	102	66	f
010 0101	37	25	%	110 0111	103	67	g
010 0110	38	26	&	110 1000	104	68	h
010 0111	39	27	'	110 1001	105	69	i
010 1000	40	28	(110 1010	106	6A	j
010 1001	41	29)	110 1011	107	6B	k
010 1010	42	2A	*	110 1100	108	6C	l
010 1011	43	2B	+	110 1101	109	6D	m
010 1100	44	2C	,	110 1110	110	6E	n
010 1101	45	2D	-	110 1111	111	6F	o
010 1110	46	2E	.	111 0000	112	70	p
010 1111	47	2F	/	111 0001	113	71	q
011 0000	48	30	0	111 0010	114	72	r
011 0001	49	31	1	111 0011	115	73	s
011 0010	50	32	2	111 0100	116	74	t
011 0011	51	33	3	111 0101	117	75	u
011 0100	52	34	4	111 0110	118	76	v
011 0101	53	35	5	111 0111	119	77	w
011 0110	54	36	6	111 1000	120	78	x
011 0111	55	37	7	111 1001	121	79	y
011 1000	56	38	8	111 1010	122	7A	z
011 1001	57	39	9	111 1011	123	7B	{
011 1010	58	3A	:	111 1100	124	7C	↓
011 1011	59	3B	;	111 1101	125	7D	}
011 1100	60	3C	<	111 1110	126	7E	≈
011 1101	61	3D	=	101 1010	90	5A	Z
011 1110	62	3E	>	101 1011	91	5B	[
011 1111	63	3F	?	101 1100	92	5C	↓
100 0000	64	40	@	101 1101	93	5D]
100 0001	65	41	A	101 1110	94	5E	^
100 0010	66	42	B	101 1111	95	5F	_
100 0011	67	43	C	110 0000	96	60	·
100 0100	68	44	D	110 0001	97	61	a
100 0101	69	45	E	110 0010	98	62	b
100 0110	70	46	F	110 0011	99	63	c
100 0111	71	47	G	110 0100	100	64	d
100 1000	72	48	H	110 0101	101	65	e
100 1001	73	49	I	110 0110	102	66	f
100 1010	74	4A	J	110 0111	103	67	g
100 1011	75	4B	K	110 1000	104	68	h
100 1100	76	4C	L	110 1001	105	69	i
100 1101	77	4D	M	110 1010	106	6A	j
100 1110	78	4E	N	110 1011	107	6B	k
100 1111	79	4F	O	110 1100	108	6C	l
101 0000	80	50	P	110 1101	109	6D	m

101 0001	81	51	Q	110 1110	110	6E	n
101 0010	82	52	R	110 1111	111	6F	o
101 0011	83	53	S	111 0000	112	70	p
101 0100	84	54	T	111 0001	113	71	q
101 0101	85	55	U	111 0010	114	72	r
101 0110	86	56	V	111 0011	115	73	s
101 0111	87	57	W	111 0100	116	74	t
101 1000	88	58	X	111 0101	117	75	u
101 1001	89	59	Y	111 0110	118	76	v
101 1010	90	5A	Z	111 0111	119	77	w
101 1011	91	5B	[111 1000	120	78	x
101 1100	92	5C	\	111 1001	121	79	y
101 1101	93	5D]	111 1010	122	7A	z
101 1110	94	5E	^	111 1011	123	7B	{
101 1111	95	5F	_	111 1100	124	7C	↓
110 0000	96	60	`	111 1101	125	7D	}
110 0001	97	61	a	111 1110	126	7E	≈

Tài liệu tham khảo

[1] *Kiều Xuân Thực, Vũ Thị Thu Hương, Vũ Trung Kiên*

Vi điều khiển cấu trúc lập trình và ứng dụng – Nhà xuất bản giáo dục, 2008

[2] *Ngô Diên Tập*

Đo lường và điều khiển bằng máy tính – Nhà xuất bản khoa học và kỹ thuật, 2002

[3] *Ngô Diên Tập*

Kỹ thuật ghép nối với máy tính – Nhà xuất bản khoa học và kỹ thuật, 2001

[4] *www.datasheet.com*

