

Bộ Công Thương  
Trường Cao Đẳng Kỹ Thuật Cao Thắng  
Khoa Điện Tử - Tin Học  
Bộ Môn Điện Tử Công Nghiệp  
∞📖∞



## Đồ Án Tốt Nghiệp

Đề Tài:

***Điều Khiển & Giám Sát Thiết Bị Điện Gia Đình***

GVHD : Nguyễn Phú Quới  
SVTH : Phạm Thanh Hiếu  
Phan Thanh Liêm  
Lớp : CĐĐT06  
Niên khóa : 2006- 2009

## **Đặt Vấn Đề**

Ngày nay khoa học công nghệ đã có những bước phát triển vượt bậc, với những ứng dụng vào nhiều lĩnh vực đời sống xã hội. Trước sự vận động và phát triển không ngừng của thế giới nói chung và của khoa học kỹ thuật nói riêng thì việc đáp ứng những nhu cầu của xã hội là điều hoàn toàn có thể. Một trong những công nghệ làm thay đổi cuộc sống là công nghệ điều khiển thiết bị bằng máy tính, máy tính cũng có thể giám sát tự động điều chỉnh các thông số cho phù hợp với những nhu cầu cụ thể.

Thế giới ngày càng phát triển thì lĩnh vực điều khiển càng được mở rộng hơn. Việc ứng dụng đó đã tạo rất nhiều thuận lợi cho con người, đặc biệt với nhu cầu sống ngày càng cao thì việc áp dụng vào điều khiển các thiết bị trong gia đình càng phổ biến. Với những yếu tố trên và cùng với mục đích nghiên cứu những cái mới, tiếp cận những nguồn kiến thức thời đại, tích lũy những kinh nghiệm cho bản thân nhóm đã quyết định chọn đề tài Điều khiển và giám sát thiết bị điện gia đình.

### **1.2 Tầm Quan Trọng Và Lý Do Chọn Đề Tài**

Giao tiếp máy tính với vi điều khiển để điều khiển và giám sát thiết bị là một đề tài hấp dẫn đối với nhiều sinh viên ham mê học hỏi. Điều mà mọi sinh viên đều nhắm đến là việc học hỏi được những kiến thức quý giá và những kinh nghiệm của việc thực hiện đề tài mang lại.

Tuy đã có nhiều tài liệu đã đề cập đến đề tài này nhưng việc tự tay đi vào thực hiện là một chuyện hoàn toàn khác, nhiều vấn đề nảy sinh và đi theo đó là nhiều kinh nghiệm quý giá học tập được trong quá trình thực hiện. Đó cũng là mục đích và là mục tiêu khi nhóm thực hiện đề tài này.

### **1.3 Giới Hạn Đề Tài**

Với điều kiện và thời gian không cho phép nhóm thực hiện chỉ chú trọng thực hiện những phần chính sau:

Hệ thống gồm 1 phòng với 4 thiết bị minh họa.

Hệ thống được điều khiển bằng phần mềm dung ngôn ngữ Visual Basic6.0

Điều khiển tắt,mở thiết bị

Điều chỉnh độ sáng đối với hệ thống đèn, tốc độ đối với hệ thống quạt...

Cài đặt thời gian tắt, mở thiết bị.

Theo dõi nhiệt độ môi trường, tự động điều chỉnh theo thiết lập của người điều khiển.

Lưu và truy xuất dữ liệu quá trình.

Việc điều khiển giữa hệ thống xử lý trung tâm và hệ thống đầu cuối thông qua đường truyền là dây dẫn.

### **1.4 Mục Đích Nghiên Cứu**

Đề tài này nhằm giúp nhóm tìm hiểu những ứng dụng của họ vi điều khiển PIC 16F877A, ngôn ngữ lập trình CCS, visual basic 6.0 và các phương thức truyền thông qua công giao tiếp nối tiếp..

Là cơ sở và là tài liệu tham khảo cho sinh viên các khóa sau.

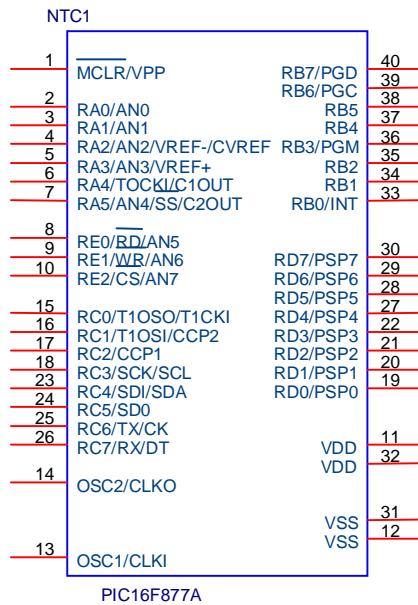
Nhằm tạo ra một sản phẩm tương đối hoàn thiện góp phần phục vụ cho các nhu cầu thực tế của con người.

# **CHƯƠNG 3**

## **LÝ THUYẾT CƠ SỞ**

### 3.1. Vi Xử Lý PIC 16F877A

#### 3.1.1. Cấu Trúc Của PIC 16F877A



Hình 3.1.1: Sơ đồ chân PIC 16F877A

Đây là vi điều khiển thuộc họ PIC16Fxxx với tập lệnh gồm 35 lệnh có độ dài 14 bit. Mỗi lệnh đều được thực thi trong một chu kỳ xung clock. Tốc độ hoạt động tối đa cho phép là 20 MHz với một chu kỳ lệnh là 200ns. Bộ nhớ chương trình 8Kx14 bit, bộ nhớ dữ liệu 368x8 byte RAM và bộ nhớ dữ liệu EEPROM với dung lượng 256x8 byte. Số PORT I/O là 5 với 33 pin I/O.

Các đặc tính ngoại vi bao gồm các khối chức năng sau:

- + Timer0: bộ đếm 8 bit với bộ chia tần số 8 bit.
- + Timer1: bộ đếm 16 bit với bộ chia tần số, có thể thực hiện chức năng đếm dựa vào xung clock ngoại vi ngay khi vi điều khiển hoạt động ở chế độ sleep.
- + Timer2: bộ đếm 8 bit với bộ chia tần số, bộ postcaler.
- + Hai bộ Capture/so sánh/điều chế độ rộng xung.
- + Các chuẩn giao tiếp nối tiếp SSP (Synchronous Serial Port), SPI và I2C.
- + Chuẩn giao tiếp nối tiếp USART với 9 bit địa chỉ.
- + Cổng giao tiếp song song PSP (Parallel Slave Port) với các chân điều khiển RD, WR, CS ở bên ngoài.

Các đặc tính Analog:

- + 8 kênh chuyển đổi ADC 10 bit.
- + Hai bộ so sánh.
- + Bên cạnh đó là một vài đặc tính khác của vi điều khiển như:
  - + Bộ nhớ flash với khả năng ghi xóa được 100.000 lần.
  - + Bộ nhớ EEPROM với khả năng ghi xóa được 1.000.000 lần.
  - + Dữ liệu bộ nhớ EEPROM có thể lưu trữ trên 40 năm.
  - + Khả năng tự nạp chương trình với sự điều khiển của phần mềm.
  - + Nạp được chương trình ngay trên mạch điện ICSP (In Circuit Serial Programming) thông qua 2 chân.
  - + Watchdog Timer với bộ dao động trong.
  - + Chức năng bảo mật mã chương trình.
  - + Chế độ Sleep.

+ Có thể hoạt động với nhiều dạng Oscillator khác nhau.

### 3.1.2 CÁC CÔNG XUẤT NHẬP CỦA PIC16F877A

Cổng xuất nhập (I/O port) chính là phương tiện mà vi điều khiển dùng để tương tác với thế giới bên ngoài. Sự tương tác này rất đa dạng và thông qua quá trình tương tác đó, chức năng của vi điều khiển được thể hiện một cách rõ ràng.

Một cổng xuất nhập của vi điều khiển bao gồm nhiều chân (I/O pin), tùy theo cách bố trí và chức năng của vi điều khiển mà số lượng cổng xuất nhập và số lượng chân trong mỗi cổng có thể khác nhau. Bên cạnh đó, do vi điều khiển được tích hợp sẵn bên trong các đặc tính giao tiếp ngoại vi nên bên cạnh chức năng là cổng xuất nhập thông thường, một số chân xuất nhập còn có thêm các chức năng khác để thể hiện sự tác động của các đặc tính ngoại vi nêu trên đối với thế giới bên ngoài. Chức năng của từng chân xuất nhập trong mỗi cổng hoàn toàn có thể được xác lập và điều khiển được thông qua các thanh ghi SFR liên quan đến chân xuất nhập đó.

Vi điều khiển PIC16F877A có 5 cổng xuất nhập, bao gồm PORTA, PORTB, PORTC, PORTD và PORTE.

#### 3.1.2.1 PORTA

PORTA (RPA) bao gồm 6 I/O pin. Đây là các chân “hai chiều” (bidirectional pin), nghĩa là có thể xuất và nhập được. Chức năng I/O này được điều khiển bởi thanh ghi TRISA (địa chỉ 85h). Muốn xác lập chức năng của một chân trong PORTA là input, ta “set” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA và ngược lại, muốn xác lập chức năng của một chân trong PORTA là output, ta “clear” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA. Thao tác này hoàn toàn tương tự đối với các PORT và các thanh ghi điều khiển tương ứng TRIS (đối với PORTA là TRISA, đối với PORTB là TRISB, đối với PORTC là TRISC, đối với PORTD là TRISD và đối với PORTE là TRISE). Bên cạnh đó PORTA còn là ngõ ra của bộ ADC, bộ so sánh, ngõ vào analog ngõ vào xung clock của Timer0 và ngõ vào của bộ giao tiếp MSSP (Master Synchronous Serial Port).

Các thanh ghi SFR liên quan đến PORTA bao gồm:

PORTA (địa chỉ 05h) : chứa giá trị các pin trong PORTA.

TRISA (địa chỉ 85h) : điều khiển xuất nhập.

CMCON (địa chỉ 9Ch) : thanh ghi điều khiển bộ so sánh.

CVRCON (địa chỉ 9Dh) : thanh ghi điều khiển bộ so sánh điện áp. ADCON1 (địa chỉ 9Fh): thanh ghi điều khiển bộ ADC.

#### 3.1.2.2 PORTB

PORTB (RPB) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISB. Bên cạnh đó một số chân của PORTB còn được sử dụng trong quá trình nạp chương trình cho vi điều khiển với các chế độ nạp khác nhau. PORTB còn liên quan đến ngắt ngoại vi và bộ Timer0. PORTB còn được tích hợp chức năng điện trở kéo lên được điều khiển bởi chương trình.

Các thanh ghi SFR liên quan đến PORTB bao gồm:

PORTB (địa chỉ 06h,106h) : chứa giá trị các pin trong PORTB

TRISB (địa chỉ 86h,186h) : điều khiển xuất nhập

OPTION\_REG (địa chỉ 81h,181h) : điều khiển ngắt ngoại vi và bộ Timer0.

#### 3.1.2.3 PORTC

PORTC (RPC) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISC. Bên cạnh đó PORTC còn chứa các chân chức năng của bộ so sánh, bộ Timer1, bộ PWM và các chuẩn giao tiếp nối tiếp I2C, SPI, SSP, USART.

Các thanh ghi điều khiển liên quan đến PORTC:

PORTC (địa chỉ 07h) : chứa giá trị các pin trong PORTC

TRISC (địa chỉ 87h) : điều khiển xuất nhập.

#### 3.1.2.4 PORTD

PORTD (RPD) gồm 8 chân I/O, thanh ghi điều khiển xuất nhập tương ứng là TRISD. PORTD còn là cổng xuất dữ liệu của chuẩn giao tiếp PSP (Parallel Slave Port).

Các thanh ghi liên quan đến PORTD bao gồm:

Thanh ghi PORTD : chứa giá trị các pin trong PORTD.

Thanh ghi TRISD : điều khiển xuất nhập.

Thanh ghi TRISE : điều khiển xuất nhập PORTE và chuẩn giao tiếp PSP.

### 3.1.2.5 PORTE

PORTE (RPE) gồm 3 chân I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISE. Các chân của PORTE có ngõ vào analog. Bên cạnh đó PORTE còn là các chân điều khiển của chuẩn giao tiếp PSP.

Các thanh ghi liên quan đến PORTE bao gồm:

PORTE : chứa giá trị các chân trong PORTE.

TRISE : điều khiển xuất nhập và xác lập các thông số cho chuẩn giao tiếp PSP.

ADCON1 : thanh ghi điều khiển khối ADC.

### 3.1.3 TIMER\_0

Đây là một trong ba bộ đếm hoặc bộ định thời của vi điều khiển PIC16F877A. Timer0 là bộ đếm 8 bit được kết nối với bộ chia tần số (prescaler) 8 bit. Cấu trúc của Timer0 cho phép ta lựa chọn xung clock tác động và cạnh tích cực của xung clock. Ngắt Timer0 sẽ xuất hiện khi Timer0 bị tràn. Bit TMR0IE (INTCON<5>) là bit điều khiển của Timer0. TMR0IE=1 cho phép ngắt Timer0 tác động, TMR0IF= 0 không cho phép ngắt Timer0 tác động.

Muốn Timer0 hoạt động ở chế độ Timer ta clear bit TOSC (OPTION\_REG<5>), khi đó giá trị thanh ghi TMR0 sẽ tăng theo từng chu kỳ xung đồng hồ (tần số vào Timer0 bằng ¼ tần số oscillator). Khi giá trị thanh ghi TMR0 từ FFh trở về 00h, ngắt Timer0 sẽ xuất hiện. Thanh ghi TMR0 cho phép ghi và xóa được giúp ta ấn định thời điểm ngắt Timer0 xuất hiện một cách linh động.

Muốn Timer0 hoạt động ở chế độ counter ta set bit TOSC (OPTION\_REG<5>). Khi đó xung tác động lên bộ đếm được lấy từ chân RA4/TOCK1. Bit TOSE (OPTION\_REG<4>) cho phép lựa chọn cạnh tác động vào bộ đếm. Cạnh tác động sẽ là cạnh lên nếu TOSE=0 và cạnh tác động sẽ là cạnh xuống nếu TOSE=1.

Khi thanh ghi TMR0 bị tràn, bit TMR0IF (INTCON<2>) sẽ được set. Đây chính là cờ ngắt của Timer0. Cờ ngắt này phải được xóa bằng chương trình trước khi bộ đếm bắt đầu thực hiện lại quá trình đếm. Ngắt Timer0 không thể “đánh thức” vi điều khiển từ chế độ sleep.

Bộ chia tần số (prescaler) được chia sẻ giữa Timer0 và WDT (Watchdog Timer). Điều đó có nghĩa là nếu prescaler được sử dụng cho Timer0 thì WDT sẽ không có được hỗ trợ của prescaler và ngược lại. Prescaler được điều khiển bởi thanh ghi OPTION\_REG. Bit PSA (OPTION\_REG<3>) xác định đối tượng tác động của prescaler. Các bit PS2:PS0 (OPTION\_REG<2:0>) xác định tỉ số chia tần số của prescaler. Xem lại thanh ghi OPTION\_REG để xác định lại một cách chi tiết về các bit điều khiển trên. Các lệnh tác động lên giá trị thanh ghi TMR0 sẽ xóa chế độ hoạt động của prescaler. Khi đối tượng tác động là Timer0, tác động lên giá trị thanh ghi TMR0 sẽ xóa prescaler nhưng không làm thay đổi đối tượng tác động của prescaler. Khi đối tượng tác động là WDT, lệnh CLRWDT sẽ xóa prescaler, đồng thời prescaler sẽ ngưng tác vụ hỗ trợ cho WDT.

Các thanh ghi điều khiển liên quan đến Timer0 bao gồm:

TMR0 (địa chỉ 01h, 101h) : chứa giá trị đếm của Timer0.

INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép ngắt hoạt động (GIE và PEIE).

OPTION\_REG (địa chỉ 81h, 181h): điều khiển prescaler.

### 3.1.4 TIMER\_1

Timer1 là bộ định thời 16 bit, giá trị của Timer1 sẽ được lưu trong hai thanh ghi (TMR1H:TMR1L). Cờ ngắt của Timer1 là bit TMR1IF (PIR1<0>). Bit điều khiển của Timer1 sẽ là TMR1IE (PIE<0>). Tương tự như Timer0, Timer1 cũng có hai chế độ hoạt động: chế độ định thời (timer) với xung kích là xung clock của oscillator (tần số của timer bằng ¼ tần số của oscillator) và chế độ đếm (counter) với xung kích là xung phản ánh các sự kiện cần đếm lấy từ bên ngoài thông qua chân RC0/T1OSO/T1CKI (cạnh tác động là cạnh lên). Việc lựa chọn xung tác động (tương ứng với việc lựa chọn chế độ hoạt động là timer hay counter) được điều khiển bởi bit TMR1CS (T1CON<1>).

Ngoài ra Timer1 còn có chức năng reset input bên trong được điều khiển bởi một trong hai khối CCP (Capture/Compare/PWM). Khi bit T1OSCEN (T1CON<3>) được set, Timer1 sẽ lấy xung clock từ hai chân RC1/T1OSI/CCP2 và RC0/T1OSO/T1CKI làm xung đếm. Timer1 sẽ bắt đầu đếm sau cạnh xuống đầu tiên của xung ngõ vào. Khi đó PORTC sẽ bỏ qua sự tác động của hai bit TRISC<1:0> và PORTC<2:1> được gán giá trị 0. Khi clear bit T1OSCEN Timer1 sẽ lấy xung đếm từ oscillator hoặc từ chân RC0/T1OSO/T1CKI. Timer1 có hai chế độ đếm là đồng bộ (Synchronous) và bất đồng bộ (Asynchronous). Chế độ đếm được quyết định bởi bit điều khiển (T1CON<2>). Khi =1 xung đếm lấy từ bên ngoài sẽ không được đồng bộ hóa với xung clock bên trong, Timer1 sẽ tiếp tục quá trình đếm khi vi điều khiển đang ở chế độ sleep và ngắt do Timer1 tạo ra khi bị tràn có khả năng “đánh thức” vi điều khiển. Ở chế độ đếm bất đồng bộ, Timer1 không thể được sử dụng để làm nguồn xung clock cho khối CCP (Capture/Compare/Pulse width modulation). Khi =0 xung đếm vào Timer1 sẽ được đồng bộ hóa với xung clock bên trong. Ở chế độ này Timer1 sẽ không hoạt động khi vi điều khiển đang ở chế độ sleep.

Các thanh ghi liên quan đến Timer1 bao gồm:

INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép ngắt hoạt động (GIE và PEIE).

PIR1 (địa chỉ 0Ch): chứa cờ ngắt Timer1 (TMR1IF).

PIE1 (địa chỉ 8Ch): cho phép ngắt Timer1 (TMR1IE).

TMR1L (địa chỉ 0Eh): chứa giá trị 8 bit thấp của bộ đếm Timer1.

TMR1H (địa chỉ 0Fh): chứa giá trị 8 bit cao của bộ đếm Timer1.

T1CON (địa chỉ 10h): xác lập các thông số cho Timer1.

### 3.1.5 TIMER\_2

Timer2 là bộ định thời 8 bit và được hỗ trợ bởi hai bộ chia tần số prescaler và postscaler. Thanh ghi chứa giá trị đếm của Timer2 là TMR2. Bit cho phép ngắt Timer2 tác động là TMR2ON (T2CON<2>). Cờ ngắt của Timer2 là bit TMR2IF (PIR1<1>). Xung ngõ vào (tần số bằng ¼ tần số oscillator) được đưa qua bộ chia tần số prescaler 4 bit (với các tỉ số chia tần số là 1:1, 1:4 hoặc 1:16 và được điều khiển bởi các bit T2CKPS1:T2CKPS0 (T2CON<1:0>)).

Timer2 còn được hỗ trợ bởi thanh ghi PR2. Giá trị đếm trong thanh ghi TMR2 sẽ tăng từ 00h đến giá trị chứa trong thanh ghi PR2, sau đó được reset về 00h. Khi I reset thanh ghi PR2 được nhận giá trị mặc định FFh. Ngõ ra của Timer2 được đưa qua bộ chia tần số postscaler với các mức chia từ 1:1 đến 1:16. Postscaler được điều khiển bởi 4 bit T2OUTPS3:T2OUTPS0. Ngõ ra của postscaler đóng vai trò quyết định trong việc điều khiển cờ ngắt.

Ngoài ra ngõ ra của Timer2 còn được kết nối với khối SSP, do đó Timer2 còn đóng vai trò tạo ra xung clock đồng bộ cho khối giao tiếp SSP.

Các thanh ghi liên quan đến Timer2 bao gồm:

INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép toàn bộ các ngắt (GIE và PEIE).

PIR1 (địa chỉ 0Ch): chứa cờ ngắt Timer2 (TMR2IF).

PIE1 (địa chỉ 8Ch): chứa bit điều khiển Timer2 (TMR2IE).

TMR2 (địa chỉ 11h): chứa giá trị đếm của Timer2.

T2CON (địa chỉ 12h): xác lập các thông số cho Timer2. PR2 (địa chỉ 92h): thanh ghi hỗ trợ cho Timer2.

Ta có một vài nhận xét về Timer0, Timer1 và Timer2 như sau:

Timer0 và Timer2 là bộ đếm 8 bit (giá trị đếm tối đa là FFh), trong khi Timer1 là bộ đếm 16 bit (giá trị đếm tối đa là FFFFh). Timer0, Timer1 và Timer2 đều có hai chế độ hoạt động là timer và counter. Xung clock có tần số bằng  $\frac{1}{4}$  tần số của oscillator. Xung tác động lên Timer0 được hỗ trợ bởi prescaler và có thể được thiết lập ở nhiều chế độ khác nhau (tần số tác động, cạnh tác động) trong khi các thông số của xung tác động lên Timer1 là cố định. Timer2 được hỗ trợ bởi hai bộ chia tần số prescaler và postcaler độc lập, tuy nhiên cạnh tác động vẫn được cố định là cạnh lên. Timer1 có quan hệ với khối CCP, trong khi Timer2 được kết nối với khối SSP. Một vài so sánh sẽ giúp ta dễ dàng lựa chọn được Timer thích hợp cho ứng dụng.

### 3.1.6 ADC

ADC (Analog to Digital Converter) là bộ chuyển đổi tín hiệu giữa hai dạng tương tự và số. PIC16F877A có 8 ngõ vào analog (RA4:RA0 và RE2:RE0). Hiệu điện thế chuẩn VREF có thể được lựa chọn là VDD, VSS hay hiệu điện thế chuẩn được xác lập trên hai chân RA2 và RA3. Kết quả chuyển đổi từ tín hiệu tương tự sang tín hiệu số là 10 bit số tương ứng và được lưu trong hai thanh ghi ADRESH:ADRESL. Khi không sử dụng bộ chuyển đổi ADC, các thanh ghi này có thể được sử dụng như các thanh ghi thông thường khác. Khi quá trình chuyển đổi hoàn tất, kết quả sẽ được lưu vào hai thanh ghi ADRESH:ADRESL, bit (ADCON0<2>) được xóa về 0 và cờ ngắt ADIF được set.

Qui trình chuyển đổi từ tương tự sang số bao gồm các bước sau:

#### 1. Thiết lập các thông số cho bộ chuyển đổi ADC:

Chọn ngõ vào analog, chọn điện áp mẫu (dựa trên các thông số của thanh ghi ADCON1)

Chọn kênh chuyển đổi AD (thanh ghi ADCON0).

Chọn xung clock cho kênh chuyển đổi AD (thanh ghi ADCON0).

Cho phép bộ chuyển đổi AD hoạt động (thanh ghi ADCON0).

#### 2. Thiết lập các cờ ngắt cho bộ AD

Clear bit ADIF.

Set bit ADIE.

Set bit PEIE.

Set bit GIE.

#### 3. Đợi cho tới khi quá trình lấy mẫu hoàn tất.

4. Bắt đầu quá trình chuyển đổi (set bit ).

5. Đợi cho tới khi quá trình chuyển đổi hoàn tất bằng cách:

Kiểm tra bit . Nếu =0, quá trình chuyển đổi đã hoàn tất.

Kiểm tra cờ ngắt.

6. Đọc kết quả chuyển đổi và xóa cờ ngắt, set bit (nếu cần tiếp tục chuyển đổi).

7. Tiếp tục thực hiện các bước 1 & 2 cho quá trình chuyển đổi tiếp theo

Cần chú ý là có hai cách lưu kết quả chuyển đổi AD, việc lựa chọn cách lưu được điều khiển bởi bit ADFM .

Các thanh ghi liên quan đến bộ chuyển đổi ADC bao gồm:

INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép các ngắt (các bit GIE, PEIE).

PIR1 (địa chỉ 0Ch): chứa cờ ngắt AD (bit ADIF).

PIE1 (địa chỉ 8Ch): chứa bit điều khiển AD (ADIE).

ADRESH (địa chỉ 1Eh) và ADRESL (địa chỉ 9Eh): các thanh ghi chứa kết quả chuyển đổi AD.

ADCON0 (địa chỉ 1Fh) và ADCON1 (địa chỉ 9Fh): xác lập các thông số cho bộ chuyển đổi AD.

PORTA (địa chỉ 05h) và TRISA (địa chỉ 85h): liên quan đến các ngõ vào analog ở PORTA.

PORTE (địa chỉ 09h) và TRISE (địa chỉ 89h): liên quan đến các ngõ vào analog ở PORTE.

### 3.1.7 COMPARATOR

Bộ so sánh bao gồm hai bộ so sánh tín hiệu analog và được đặt ở PORTA. ngõ vào bộ so sánh là các chân RA3:RA0, ngõ ra là hai chân RA4 và RA5. Thanh ghi điều khiển bộ so sánh là CMCON. Các bit CM2:CM0 trong thanh ghi CMCON đóng vai trò chọn lựa các chế độ hoạt động cho bộ Comparator (hình 2.10).



Cơ chế hoạt động của bộ Comparator như sau:

Tín hiệu analog ở chân VIN+ sẽ được so sánh với điện áp chuẩn ở chân VIN- và tín hiệu ở ngõ ra bộ so sánh sẽ thay đổi tương ứng như hình vẽ. Khi điện áp ở chân VIN+ lớn hơn điện áp ở chân VIN- ngõ ra sẽ ở mức 1 và ngược lại.

Dựa vào hình vẽ ta thấy đáp ứng tại ngõ ra không phải là tức thời so với thay đổi tại ngõ vào mà cần có một khoảng thời gian nhất định để ngõ ra thay đổi trạng thái (tối đa là 10 $\mu$ s). Cần chú ý đến khoảng thời gian đáp ứng này khi sử dụng bộ so sánh.

Cực tính của các bộ so sánh có thể thay đổi dựa vào các giá trị đặt vào các bit C2INV và C1INV (CMCON<4:5>).

#### **Các chế độ hoạt động của bộ comparator.**

Các bit C2OUT và C1OUT (CMCON<7:6>) đóng vai trò ghi nhận sự thay đổi tín hiệu analog so với điện áp đặt trước. Các bit này cần được xử lý thích hợp bằng chương trình để ghi nhận sự thay đổi của tín hiệu ngõ vào. Cờ ngắt của bộ so sánh là bit CMIF (thanh ghi PIR1). Cờ ngắt này phải được reset về 0. Bit điều khiển bộ so sánh là bit CMIE (Thanh ghi PIE).

Các thanh ghi liên quan đến bộ so sánh bao gồm:

CMCON (địa chỉ 9Ch) và CVRCON (địa chỉ 9Dh): xác lập các thông số cho bộ so sánh.

Thanh ghi INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): chứa các bit cho phép các ngắt (GIE và PEIE).

Thanh ghi PIR2 (địa chỉ 0Dh): chứa cờ ngắt của bộ so sánh (CMIF).

Thanh ghi PIE2 (địa chỉ 8Dh): chứa bit cho phép bộ so sánh (CNIE).

Thanh ghi PORTA (địa chỉ 05h) và TRISA (địa chỉ 85h): các thanh ghi điều khiển PORTA .

#### **3.1.8 BỘ TẠO ĐIỆN ÁP SO SÁNH**

Bộ so sánh này chỉ hoạt động khi bộ Comparator được định dạng hoạt động ở chế độ '110'. Khi đó các pin RA0/AN0 và RA1/AN1 (khi CIS = 0) hoặc pin RA3/AN3 và RA2/AN2 (khi CIS = 1) sẽ là ngõ vào analog của điện áp cần so sánh đưa vào ngõ VIN- của 2 bộ so sánh C1 và C2. Trong khi đó điện áp đưa vào ngõ VIN+ sẽ được lấy từ một bộ tạo điện áp so sánh.

Bộ tạo điện áp so sánh này bao gồm một thang điện trở 16 mức đóng vai trò là cầu phân áp chia nhỏ điện áp VDD thành nhiều mức khác nhau (16 mức). Mỗi mức có giá trị điện áp khác nhau tùy thuộc vào bit điều khiển CVRR (CVRCON<5>). Nếu CVRR ở mức logic 1, điện trở 8R sẽ không có tác dụng như một thành phần của cầu phân áp (BJT dẫn mạnh và dòng điện không đi qua điện trở 8R), khi đó 1 mức điện áp có giá trị VDD/24. Ngược lại khi CVRR ở mức logic 0, dòng điện sẽ qua điện trở 8R và 1 mức điện áp có giá trị VDD/32. Các mức điện áp này được đưa qua bộ MUX cho phép ta chọn được điện áp đưa ra pin RA2/AN2/VREF-/CVREF để đưa vào ngõ VIN+ của bộ so sánh bằng cách đưa các giá trị thích hợp vào các bit CVR3:CVR0.

Bộ tạo điện áp so sánh này có thể xem như một bộ chuyển đổi D/A đơn giản. Giá trị điện áp cần so sánh ở ngõ vào Analog sẽ được so sánh với các mức điện áp do bộ tạo điện áp tạo ra cho tới khi hai điện áp này đạt được giá trị xấp xỉ bằng nhau. Khi đó kết quả chuyển đổi xem như được chứa trong các bit CVR3:CVR0.

Các thanh ghi liên quan đến bộ tạo điện áp so sánh này bao gồm:

Thanh ghi CVRCON (địa chỉ 9Dh): thanh ghi trực tiếp điều khiển bộ so sánh điện áp.

Thanh ghi CMCON (địa chỉ 9Ch): thanh ghi điều khiển bộ Comparator.

#### **3.1.9 CCP**

CCP (Capture/Compare/PWM) bao gồm các thao tác trên các xung đếm cung cấp bởi các bộ đếm Timer1 và Timer2. PIC16F877A được tích hợp sẵn hai khối CCP : CCP1 và CCP2. Mỗi CCP có một thanh ghi 16 bit (CCPR1H:CCPR1L và CCPR2H:CCPR2L), pin điều khiển dùng cho khối CCPx là RC2/CCP1 và RC1/T1OSI/CCP2. Các chức năng của CCP bao gồm:

Capture.

So sánh (Compare).

Điều chế độ rộng xung PWM (Pulse Width Modulation).

Cả CCP1 và CCP2 về nguyên tắc hoạt động đều giống nhau và chức năng của từng khối là khá độc lập. Tuy nhiên trong một số trường hợp ngoại lệ CCP1 và CCP2 có khả năng phối hợp với

nhau để tạo ra các hiện tượng đặc biệt (Special event trigger) hoặc các tác động lên Timer1 và Timer2. Các trường hợp này được liệt kê trong bảng sau:

CCPx	CCPy	Tác động
Capture	Capture	Dùng chung nguồn xung clock từ TMR1
Capture	Compare	Tạo ra hiện tượng đặc biệt làm xóa TMR1
Compare	Compare	Tạo ra hiện tượng đặc biệt làm xóa TMR1
PWM	PWM	Dùng chung tần số xung clock và cùng chịu tác động của ngắt TMR2.
PWM	Capture	Hoạt động độc lập
PWM	Compare	Hoạt động độc lập

Khi hoạt động ở chế độ Capture thì khi có một “hiện tượng” xảy ra tại pin RC2/CCP1 (hoặc RC1/T1OSI/CCP2), giá trị của thanh ghi TMR1 sẽ được đưa vào thanh ghi CCPR1 (CCPR2). Các “hiện tượng” được định nghĩa bởi các bit CCPxM3:CCPxM0 (CCPxCON<3:0>) và có thể là một trong các hiện tượng sau:

- ◆ Mỗi khi có cạnh xuống tại các pin CCP.
- ◆ Mỗi khi có cạnh lên.
- ◆ Mỗi cạnh lên thứ 4.
- ◆ Mỗi cạnh lên thứ 16.

**Sơ đồ khối CCP (Capture mode).**

Sau khi giá trị của thanh ghi TMR1 được đưa vào thanh ghi CCPRx, cờ ngắt CCPIF được set và phải được xóa bằng chương trình. Nếu hiện tượng tiếp theo xảy ra mà giá trị trong thanh ghi CCPRx chưa được xử lý, giá trị tiếp theo nhận được sẽ tự động được ghi đè lên giá trị cũ. Một số điểm cần chú ý khi sử dụng CCP như sau:

Các pin dùng cho khối CCP phải được ấn định là input (set các bit tương ứng trong thanh ghi TRISC). Khi ấn định các pin dùng cho khối CCP là output, việc đưa giá trị vào PORTC cũng có thể gây ra các “hiện tượng” tác động lên khối CCP do trạng thái của pin thay đổi. Timer1 phải được hoạt động ở chế độ Timer hoặc chế độ đếm đồng bộ. Tránh sử dụng ngắt CCP bằng cách clear bit CCPxIE (thanh ghi PIE1), cờ ngắt CCPIF nên được xóa bằng phần mềm mỗi khi được set để tiếp tục nhận định được trạng thái hoạt động của CCP.

CCP còn được tích hợp bộ chia tần số prescaler được điều khiển bởi các bit CCPxM3:CCPxM0. Việc thay đổi đối tượng tác động của prescaler có thể tạo ra hoạt động ngắt. Prescaler được xóa khi CCP không hoạt động hoặc khi reset.

Xem các thanh ghi điều khiển khối CCP .

Khi hoạt động ở chế độ Compare, giá trị trong thanh ghi CCPRx sẽ thường xuyên được so sánh với giá trị trong thanh ghi TMR1. Khi hai thanh ghi chứa giá trị bằng nhau, các pin của CCP được thay đổi trạng thái (được đưa lên mức cao, đưa xuống mức thấp hoặc giữ nguyên trạng thái), đồng thời cờ ngắt CCPIF cũng sẽ được set. Sự thay đổi trạng thái của pin có thể được điều khiển bởi các bit CCPxM3:CCPxM0 (CCPxCON <3:0>).

Tương tự như ở chế độ Capture, Timer1 phải được ấn định chế độ hoạt động là timer hoặc đếm đồng bộ. Ngoài ra, khi ở chế độ Compare, CCP có khả năng tạo ra hiện tượng đặc biệt (Special Event trigger) làm reset giá trị thanh ghi TMR1 và khởi động bộ chuyển đổi ADC. Điều này cho phép ta điều khiển giá trị thanh ghi TMR1 một cách linh động hơn. Khi hoạt động ở chế độ PWM (Pulse Width Modulation \_ khối điều chế độ rộng xung), tín hiệu sau khi điều chế sẽ được đưa ra các pin của khối CCP (cần ấn định các pin này là output). Để sử dụng chức năng điều chế này trước tiên ta cần tiến hành các bước cài đặt sau:

1. Thiết lập thời gian của 1 chu kỳ của xung điều chế cho PWM (period) bằng cách đưa giá trị thích hợp vào thanh ghi PR2.
2. Thiết lập độ rộng xung cần điều chế (duty cycle) bằng cách đưa giá trị vào thanh ghi CCPRxL và các bit CCP1CON<5:4>.

3. Điều khiển các pin của CCP là output bằng cách clear các bit tương ứng trong thanh ghi TRISC.
4. Thiết lập giá trị bộ chia tần số prescaler của Timer2 và cho phép Timer2 hoạt động bằng cách đưa giá trị thích hợp vào thanh ghi T2CON.
5. Cho phép CCP hoạt động ở chế độ PWM.

Giá trị 1 chu kỳ (period) của xung điều chế được tính bằng công thức:

Bộ chia tần số prescaler của Timer2 chỉ có thể nhận các giá trị 1,4 hoặc 16 (xem lại Timer2 để biết thêm chi tiết). Khi giá trị thanh ghi PR2 bằng với giá trị thanh ghi TMR2 thì quá trình sau xảy ra:

Thanh ghi TMR2 tự động được xóa. Pin của khối CCP được set. Giá trị thanh ghi CCPRL (chứa giá trị ấn định độ rộng xung điều chế duty cycle) được đưa vào thanh ghi CCPRH. Độ rộng của xung điều chế (duty cycle) được tính theo công thức:

$$\text{PWM period} = [(PR2)+1]*4*TOSC*(\text{giá trị bộ chia tần số của TMR2})$$

PWM duty cycle = (CCPRxL:CCPxCON<5:4>)\*TOSC\*(giá trị bộ chia tần số TMR2) Như vậy 2 bit CCPxCON<5:4> sẽ chứa 2 bit LSB. Thanh ghi CCPRxL chứa byte cao của giá trị quyết định độ rộng xung. Thanh ghi CCPRxH đóng vai trò là buffer cho khối PWM. Khi giá trị trong thanh ghi CCPRxH bằng với giá trị trong thanh ghi TMR2 và hai bit CCPxCON<5:4> bằng với giá trị 2 bit của bộ chia tần số prescaler, pin của khối CCP lại được đưa về mức thấp. Một số điểm cần chú ý khi sử dụng khối PWM:

Timer2 có hai bộ chia tần số prescaler và postscaler. Tuy nhiên bộ postscaler không được sử dụng trong quá trình điều chế độ rộng xung của khối PWM.

Nếu thời gian duty cycle dài hơn thời gian chu kỳ xung period thì xung ngõ ra tiếp tục được giữ ở mức cao sau khi giá trị PR2 bằng với giá trị TMR2.

### 3.1.10 Giao Tiếp Nối Tiếp

#### 3.1.10.1 Usart

USART (Universal Synchronous Asynchronous Receiver Transmitter) là một trong hai chuẩn giao tiếp nối tiếp. USART còn được gọi là giao diện giao tiếp nối tiếp nối tiếp SCI (Serial Communication Interface). Có thể sử dụng giao diện này cho các giao tiếp với các thiết bị ngoại vi, với các vi điều khiển khác hay với máy tính. Các dạng của giao diện USART ngoại vi bao gồm:

Bất đồng bộ (Asynchronous).

Đồng bộ\_ Master mode.

Đồng bộ\_ Slave mode.

Hai pin dùng cho giao diện này là RC6/TX/CK và RC7/RX/DT, trong đó RC6/TX/CK dùng để truyền xung clock (baud rate) và RC7/RX/DT dùng để truyền data. Trong trường hợp này ta phải set bit TRISC<7:6> và SPEN (RCSTA<7>) để cho phép giao diện USART.

PIC16F877A được tích hợp sẵn bộ tạo tốc độ baud BRG (Baud Rate Genetator) 8 bit dùng cho giao diện USART. BRG thực chất là một bộ đếm có thể được sử dụng cho cả hai dạng đồng bộ và bất đồng bộ và được điều khiển bởi thanh ghi PSBRG. Ở dạng bất đồng bộ, BRG còn được điều khiển bởi bit BRGH ( TXSTA<2>). Ở dạng đồng bộ tác động của bit BRGH được bỏ qua. Tốc độ baud do BRG tạo ra được tính theo công thức sau:

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64 (X + 1))$	Baud Rate = $F_{osc}/(16 (X + 1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4 (X + 1))$	N/A

Trong đó X là giá trị của thanh ghi RSBRG ( X là số nguyên và  $0 < X < 255$ ).

Các thanh ghi liên quan đến BRG bao gồm:

TXSTA (địa chỉ 98h): chọn chế độ đồng bộ hay bất đồng bộ ( bit SYNC) và chọn mức tốc độ baud (bit BRGH).

RCSTA (địa chỉ 18h): cho phép hoạt động cổng nối tiếp (bit SPEN).

RSBRG (địa chỉ 99h): quyết định tốc độ baud.

### 3.1.10.2 Usart Bất Đồng Bộ

Ở chế độ truyền này USART hoạt động theo chuẩn NRZ (None-Return-to-Zero), nghĩa là các bit truyền đi sẽ bao gồm 1 bit Start, 8 hay 9 bit dữ liệu (thông thường là 8 bit) và 1 bit Stop. Bit LSB sẽ được truyền đi trước. Các khối truyền và nhận data độc lập với nhau sẽ dùng chung tần số tương ứng với tốc độ baud cho quá trình dịch dữ liệu (tốc độ baud gấp 16 hay 64 lần tốc độ dịch dữ liệu tùy theo giá trị của bit BRGH), và để đảm bảo tính hiệu quả của dữ liệu thì hai khối truyền và nhận phải dùng chung một định dạng dữ liệu.

### 3.1.10.3 Truyền Dữ Liệu Qua Chuẩn Giao Tiếp Usart Bất Đồng Bộ

Thành phần quan trọng nhất của khối truyền dữ liệu là thanh ghi dịch dữ liệu TSR (Transmit Shift Register). Thanh ghi TSR sẽ lấy dữ liệu từ thanh ghi đệm dùng cho quá trình truyền dữ liệu TXREG. Dữ liệu cần truyền phải được đưa trước vào thanh ghi TXREG. Ngay sau khi bit Stop của dữ liệu cần truyền trước đó được truyền xong, dữ liệu từ thanh ghi TXREG sẽ được đưa vào thanh ghi TSR, thanh ghi TXREG bị rỗng, ngắt xảy ra và cờ hiệu TXIF (PIR1<4>) được set. Ngắt này được điều khiển bởi bit TXIE (PIE1<4>). Cờ hiệu TXIF vẫn được set bất chấp trạng thái của bit TXIE hay tác động của chương trình (không thể xóa TXIF bằng chương trình) mà chỉ reset về 0 khi có dữ liệu mới được đưa vào thanh ghi TXREG.

Trong khi cờ hiệu TXIF đóng vai trò chỉ thị trạng thái thanh ghi TXREG thì cờ hiệu TRMT (TXSTA<1>) có nhiệm vụ thể hiện trạng thái thanh ghi TSR. Khi thanh ghi TSR rỗng, bit TRMT sẽ được set. Bit này chỉ đọc và không có ngắt nào được gắn với trạng thái của nó. Một điểm cần chú ý nữa là thanh ghi TSR không có trong bộ nhớ dữ liệu và chỉ được điều khiển bởi CPU.

Khối truyền dữ liệu được cho phép hoạt động khi bit TXEN (TXSTA<5>) được set. Quá trình truyền dữ liệu chỉ thực sự bắt đầu khi đã có dữ liệu trong thanh ghi TXREG và xung truyền baud được tạo ra. Khi khối truyền dữ liệu được khởi động lần đầu tiên, thanh ghi TSR rỗng. Tại thời điểm đó, dữ liệu đưa vào thanh ghi TXREG ngay lập tức được load vào thanh ghi TSR và thanh ghi TXREG bị rỗng. Lúc này ta có thể hình thành một chuỗi dữ liệu liên tục cho quá trình truyền dữ liệu. Trong quá trình truyền dữ liệu nếu bit TXEN bị reset về 0, quá trình truyền kết thúc, khối truyền dữ liệu được reset và pin RC6/TX/CK chuyển đến trạng thái high-impedance. Trong trường hợp dữ liệu cần truyền là 9 bit, bit TX9 (TXSTA<6>) được set và bit dữ liệu thứ 9 sẽ được lưu trong bit TX9D (TXSTA<0>). Nên ghi bit dữ liệu thứ 9 vào trước, vì khi ghi 8 bit dữ liệu vào thanh ghi TXREG trước có thể xảy ra trường hợp nội dung thanh ghi TXREG sẽ được load vào thanh ghi TSG trước, như vậy dữ liệu truyền đi sẽ bị sai khác so với yêu cầu.

Tóm lại, để truyền dữ liệu theo giao diện USART bất đồng bộ, ta cần thực hiện tuần tự các bước sau:

1. Tạo xung truyền baud bằng cách đưa các giá trị cần thiết vào thanh ghi RSBRG và bit điều khiển mức tốc độ baud BRGH.
2. Cho phép cổng giao diện nối tiếp nối tiếp bất đồng bộ bằng cách clear bit SYNC và set bit PSEN.
3. Set bit TXIE nếu cần sử dụng ngắt truyền.
4. Set bit TX9 nếu định dạng dữ liệu cần truyền là 9 bit.
5. Set bit TXEN để cho phép truyền dữ liệu (lúc này bit TXIF cũng sẽ được set).
6. Nếu định dạng dữ liệu là 9 bit, đưa bit dữ liệu thứ 9 vào bit TX9D.
7. Đưa 8 bit dữ liệu cần truyền vào thanh ghi TXREG.
8. Nếu sử dụng ngắt truyền, cần kiểm tra lại các bit GIE và PEIE (thanh ghi INTCON).

Các thanh ghi liên quan đến quá trình truyền dữ liệu bằng giao diện USART bất đồng bộ:

- Thanh ghi INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép tắt cả các ngắt.
- Thanh ghi PIR1 (địa chỉ 0Ch): chứa cờ hiệu TXIF. Thanh ghi PIE1 (địa chỉ 8Ch): chứa bit cho phép ngắt truyền TXIE.

- Thanh ghi RCSTA (địa chỉ 18h): chứa bit cho phép công truyền dữ liệu (hai pin RC6/TX/CK và RC7/RX/DT).
- Thanh ghi TXREG (địa chỉ 19h): thanh ghi chứa dữ liệu cần truyền.
- Thanh ghi TXSTA (địa chỉ 98h): xác lập các thông số cho giao diện.
- Thanh ghi SPBRG (địa chỉ 99h): quyết định tốc độ baud.

## 3.2. Truyền Thông Qua Cổng Giao Tiếp Nối Tiếp

### 3.2.1. Cấu trúc cổng nối tiếp

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

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

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

Các đường dẫn bắt tay lối vào: RI, DSR, CTS.

Các đường dẫn bắt tay lối ra: RTS, DTR.

Ngoài ra, tất cả các ngõ ra đều có đặc tính chống chập mạch.

RS232C là chuẩn đang được áp dụng hiện nay.

Điện áp sử dụng là  $\pm 12V$ . Trong đó:

-12V là mức logic 1 (HIGH)

+12V là mức logic 0 (LOW)

Cụ thể:

+3V -> +12V là mức 0.

+5V -> +12V là mức tín cậy (của mức 0).

-3V -> -12V là mức 0.

-5V -> -12V là mức tín cậy (của mức 1).

Bằng việc thu hẹp giới hạn điện áp đường truyền, tốc độ truyền dữ liệu được tăng lên đáng kể. Ngoài ra chuẩn RS232C cũng quy định trở kháng tải, giá trị này thuộc phạm vi  $3K\Omega$  đến  $7K\Omega$ ; đồng thời bộ đệm phải duy trì tăng điện áp tương đối lớn khoảng  $30V/\mu s$ . Các yêu cầu về mặt điện được quy định trong chuẩn RS232C như sau:

1. Mức logic 1 (mức dấu) nằm trong khoảng  $-3V$  ->  $-12V$ ; Mức logic 0 (Mức trống) nằm trong khoảng  $+3V$  ->  $+12V$ .
2. Trở kháng tải về phía bộ nhận của mạch phải nằm trong khoảng  $3K\Omega$  -  $7K\Omega$ .
3. Tốc độ truyền nhận cực đại 100 Kbit/s.
4. Các lối vào của bộ nhận phải có điện dung  $< 2500pF$ .
5. Độ dài của cáp nối giữa máy tính và thiết bị ghép nối qua cổng nối tiếp không thể vượt quá 15 mét nếu không sử dụng modem.
6. Các giá trị tốc độ truyền dữ liệu chuẩn là 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, ..., 56600 baud (bit/s).

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

Các phương thức nối giữa DTE và DCE:

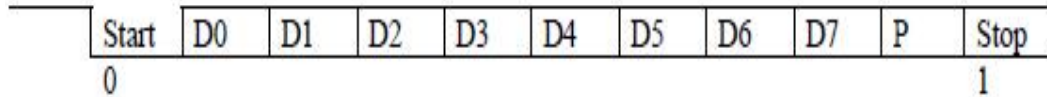
- Đơn công (simplex connection): dữ liệu chỉ được truyền theo 1 hướng.

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

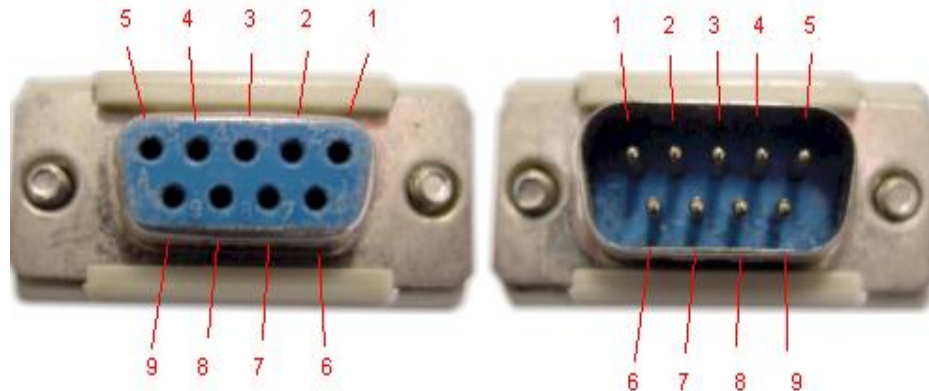
- Song công (full-duplex): số liệu được truyền đồng thời theo 2 hướng.

Đặc điểm của đường truyền dữ liệu qua cổng nối tiếp là tiến hành truyền và nhận trên các đường dẫn đơn lẻ, cho nên khi thiết bị truyền và thiết bị nhận được ghép nối với nhau thì đường truyền bên này sẽ được nối với đường nhận bên kia và ngược lại. Có như vậy mới hình thành được vòng kín của quá trình truyền dữ liệu. Để lưu ý mối quan hệ bất chéo tay như vậy người ta đã đưa vào dấu x ở giữa TD (Tx) và RD (Rx).

Việc truyền dữ liệu qua cổng nối tiếp RS232 được tiến hành theo kiểu không đồng bộ, trong đó khuôn mẫu dữ liệu có bit bắt đầu, bit dừng được chỉ ra như hình vẽ sau:



**Sơ đồ chân:**



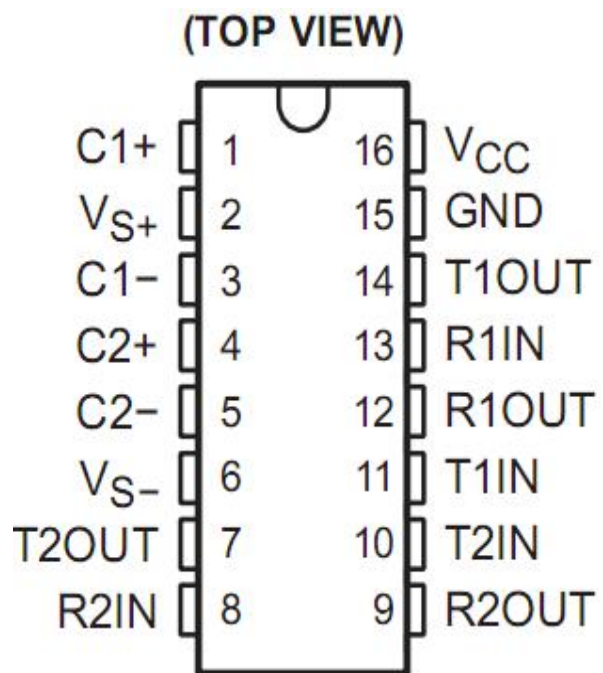
Chân (25 chân)	Chân (9 chân)	Lối vào/ra	Tên gọi	Chức năng
1	-	-	FG, Frame Ground	Đất vỏ máy
8	1	⇒	DCD, Data Carrier detect	Phát hiện tín hiệu mang dữ liệu
3	2	⇒	RxD, Receive Data	Nhận dữ liệu
2	3	⇐	TxD, Transmit Data	Truyền dữ liệu
20	4	⇐	DTR, Data terminal ready	Đầu cuối dữ liệu sẵn sàng; tính hoạt động giống với RTS nhưng được kích hoạt bởi bộ nhận khi muốn truyền dữ liệu
7	5		SG, Signal Ground	Đất của tín hiệu

6	6	⇒	DSR, Data Set Ready	Dữ liệu sẵn sàng; tính hoạt động giống với CTS nhưng được kích hoạt bởi bộ truyền khi nó sẵn sàng nhận dữ liệu
4	7	⇐	RTS, Request to Send	Yêu cầu gửi; bộ truyền đạt đường này lên mức hoạt động khi sẵn sàng truyền dữ liệu
5	8	⇒	CTS, Clear to Send	Xóa để gửi; bộ nhận đặt đường này lên mức hoạt động để thông báo cho bộ truyền là nó sẵn sàng nhận dữ liệu
22	9	⇒	RI, Ring indicate	Báo chuông cho biết là bộ phận đang nhận tín hiệu rung chuông

**3.2.2. Max232**

Vi mạch MAX 232 của hãng MAXIM là một vi mạch chuyên dùng trong giao diện nối tiếp với máy tính. Chúng có nhiệm vụ chuyển đổi mức TTL ở lối vào thành mức +10V hoặc -10V ở phía truyền và các mức +3...+15V hoặc -3...-15V thành mức TTL ở phía nhận.

Vi mạch MAX 232 có hai bộ đệm và hai bộ nhận. Đường dẫn điều khiển lối vào CTS, điều khiển việc xuất ra dữ liệu ở cổng nối tiếp khi cần thiết, được nối với chân 9 của vi mạch MAX 232. Còn chân RST (chân 10 của vi mạch MAX ) nối với đường dẫn bắt tay để điều khiển quá trình nhận. Thường thì các đường dẫn bắt tay được nối với cổng nối tiếp qua các cầu nối, để khi không dùng đến nữa có thể hở mạch các cầu này. Cách truyền dữ liệu đơn giản nhất là chỉ dùng ba đường dẫn TxD, RxD và GND (mass).



Hình 3.2.2: Sơ đồ chân Max232

**3.2.3. Visual Basic 6.0 Và Lập Trình Giao Tiếp Nối Tiếp**

**3.2.3.1. Mô tả**

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

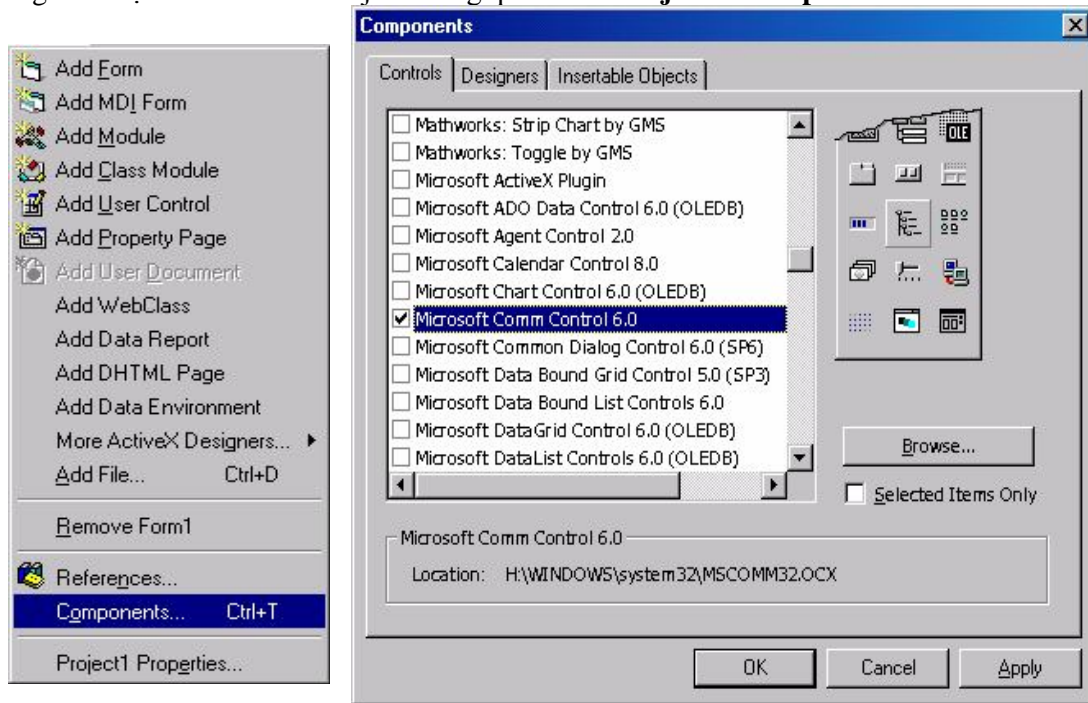
- **Điều khiển sự kiện:**



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

**- Hỏi vòng:**

Quá trình điều khiển bằng phương pháp hỏi vòng thực hiện thông qua kiểm tra các giá trị của thuộc tính CommEvent sau một chu kỳ nào đó để xác định xem có sự kiện nào xảy ra hay không. Thông thường phương pháp này sử dụng cho các chương trình nhỏ. ActiveX MsComm được bổ sung vào một Visual Basic Project thông qua menu **Project > Components**:



Hình 4.5 – Bổ sung đối tượng MsComm vào VBP

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

**Thuộc tính Mô tả**

Thuộc tính	Mô tả
commport	Số thứ tự cổng truyền thông
input	Nhận ký tự từ bộ đệm
output	Xuất ký tự ra cổng nối tiếp
portopen	Mở / đóng cổng
settings	Xác định các tham số truyền

**3.2.3.2. Các thuộc tính**

**- Settings:**

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

MSComm1.Settings = ParamString



**MSComm1**: tên đối tượng

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

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

**P**: kiểm tra chẵn lẻ, với các giá trị:

**D**: số bit dữ liệu (4, 5, 6, 7 hay 8), mặc định là 8 bit

**S**: số bit stop (1, 1.5, 2)

VD:

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

#### - **CommPort**:

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

**MSComm1.CommPort** = PortNumber

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

VD:

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

#### - **PortOpen**:

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

**MSComm1.PortOpen** = True | False

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

VD: Mở cổng COM1 với tốc độ truyền 9600 bps

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

**MSComm1.CommPort** = 1

**MSComm1.PortOpen** = True

#### - **Các thuộc tính nhận dữ liệu**:

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

**InputString** = **MSComm1.Input**

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

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

**Count** = **MSComm1.InBufferCount**

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

**MSComm1.InBufferCount** = 0

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

**MSComm1.InBufferCount** = NumByte

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

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

**MSComm1.InputLen** = 0

If **MSComm1.InBufferCount** <> 0 Then

**InputString** = **MSComm1.Input**

End If

#### - **Các thuộc tính xuất dữ liệu**:

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

#### - **CDTimeout**:

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

Cú pháp:

**MSComm1.CDTimeout** = NumTime

**- DSRTIMEOUT:**

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

**- CTSTIMEOUT:**

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

`MSComm1.CTSTIMEOUT = NumTime`

**- CTSHOLDING:**

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

**- DSRHOLDING:**

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

**- CDHOLDING:**

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

**- DTREnable:**

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

`MSComm1.DTREnable = True | False`

**- RTSEnable:**

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

`MSComm1.RTSEnable = True | False`

**- NullDiscard:**

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

`MSComm1.NullDiscard = True | False`

**- SThreshold:**

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

`MSComm1.STHRESHOLD = NumChar`

**- HandShaking:**

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

`MSComm1.HANDSHAKING = Protocol`

Các giao thức truyền bao gồm:

**- CommEvent:**

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

**3.2.3.3. Sự kiện OnComm**

Sự kiện OnComm xảy ra bất cứ khi nào giá trị của thuộc tính CommEvent thay đổi.

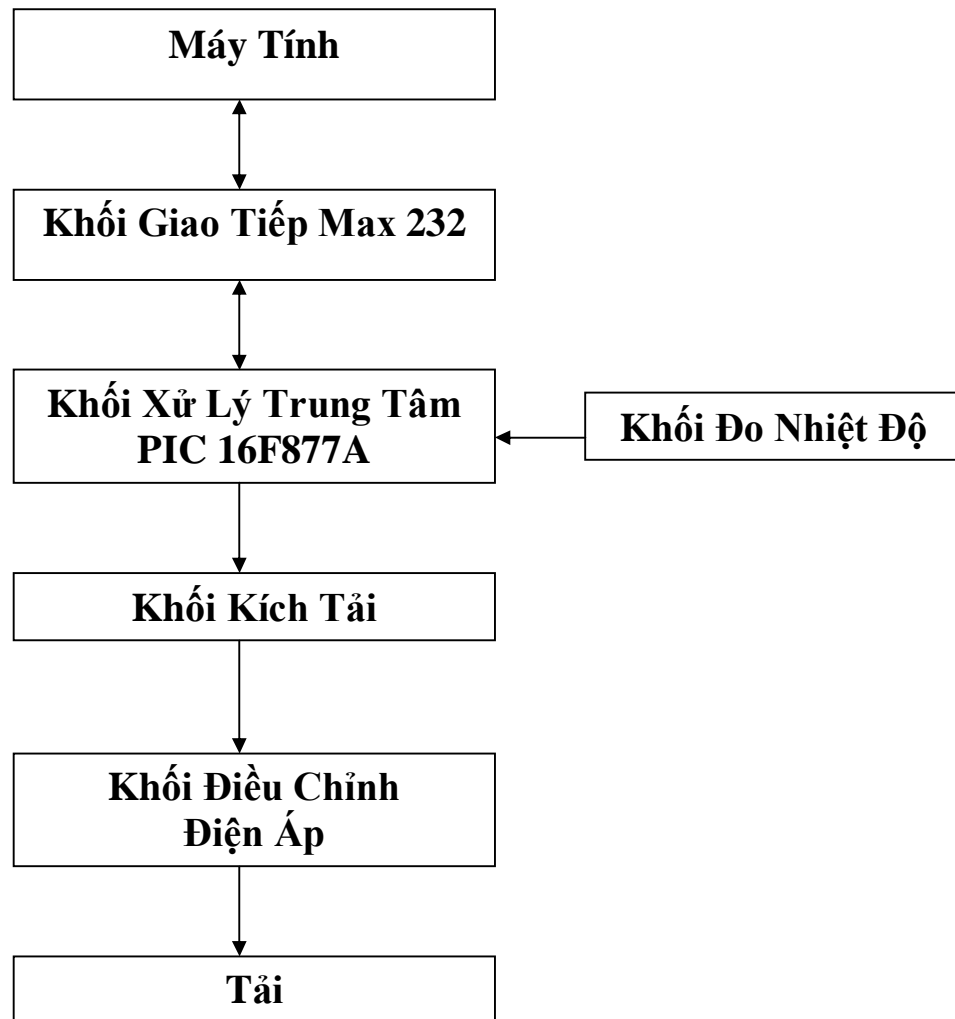
Các thuộc tính RThreshold và SThreshold = 0 sẽ cấm sự kiện OnComm khi thực hiện nhận hay gửi dữ liệu. Thông thường, SThreshold = 0 và RThreshold = 1.



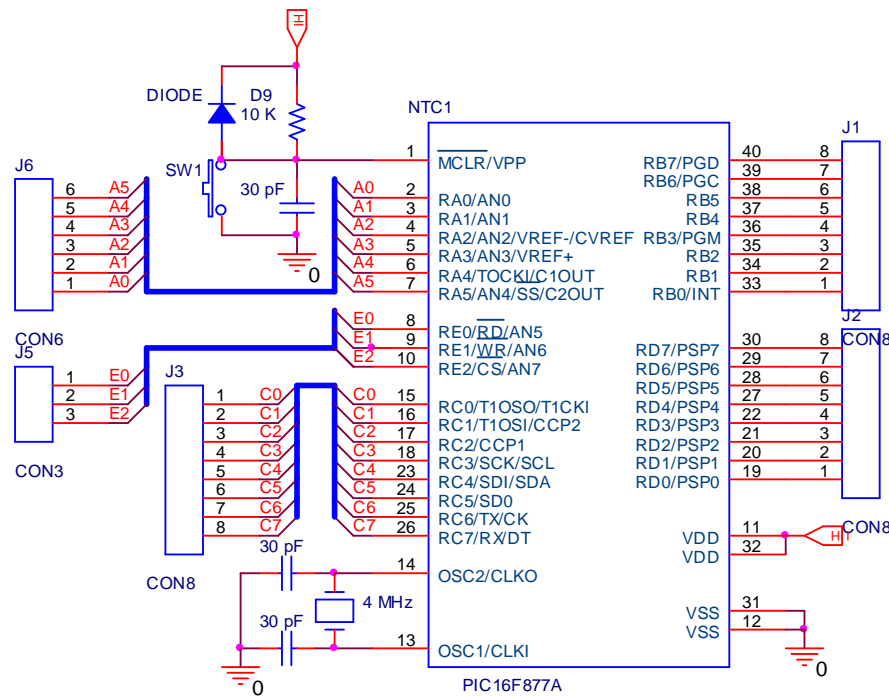
# **CHƯƠNG 4**

## **THIẾT KẾ & PHÂN TÍCH MẠCH**

## SƠ ĐỒ KHỐI



### 4.1. Khối Xử Lý Trung Tâm



Hình 4.1: Mạch Xử Lý Trung Tâm

Khối điều khiển là phần quan trọng nhất, có nhiệm vụ điều khiển mọi hoạt động, tín hiệu điều khiển được xuất ra các port I/O. Để thuận lợi trong việc thay thế hay mở rộng thêm port chúng tôi đã kéo ra cả 5 port và khi cần nâng cấp sẽ tiện lợi hơn.

Trong khối điều khiển có mạch reset, nhiệm vụ của khối reset cũng quan trọng không kém.

Khi pin MCLR ở mức logic thấp thì vi điều khiển sẽ được reset.

Với những yêu cầu:

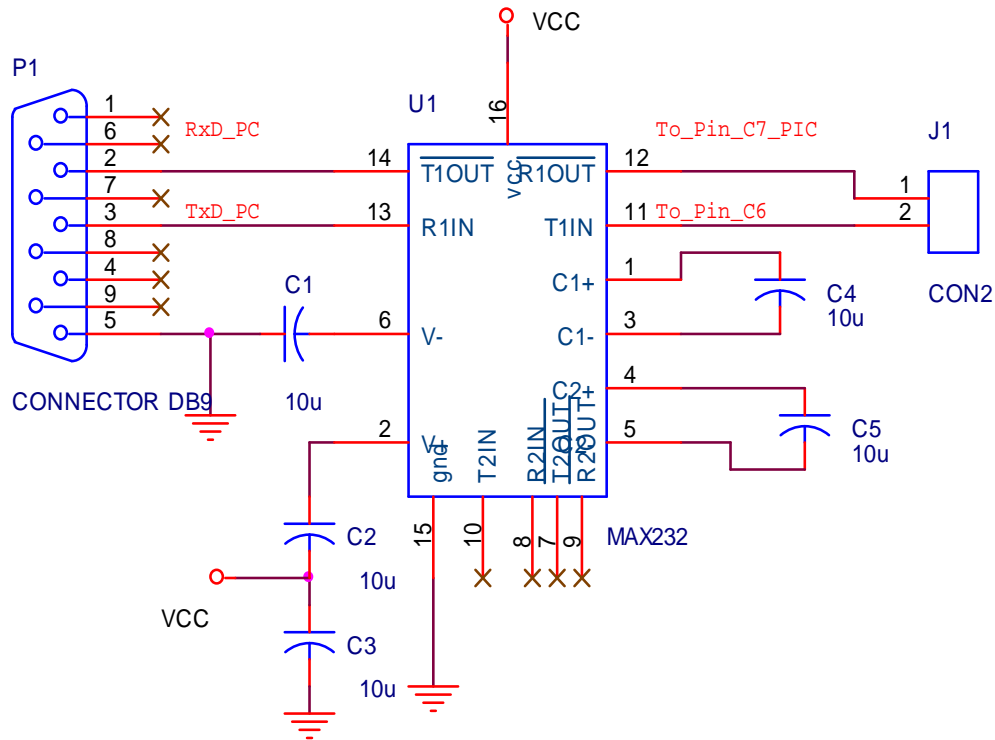
Không nối chân MCLR trực tiếp lên nguồn, cần thông qua điện trở và điện trở đó phải nhỏ hơn 40K, trong mạch này chúng ta chọn 10K để đảm bảo các đặc tính điện của vi điều khiển. Sử dụng thạch anh ngoài 4MHz, được kết nối vào 2 Pin OSC1/CLKI và OSC2/CLKO.

Trong đề tài này chúng tôi sử dụng khối xử lý trung tâm với nhiệm vụ sau:

- Tiếp nhận lệnh điều khiển từ máy tính, xử lý và thực hiện điều khiển thiết bị.
- Tiếp nhận tín hiệu từ khối đo nhiệt độ, thực hiện quá trình xử lý và truyền dữ liệu lên máy tính.

### 4.2. Khối Max232

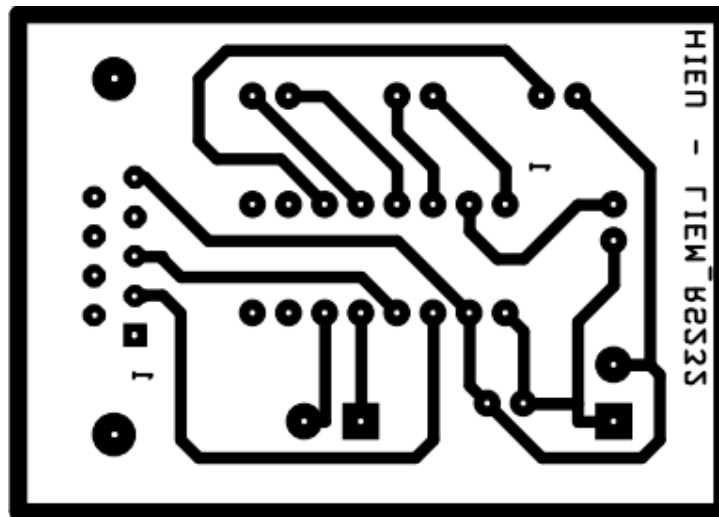
Khi thực hiện giao tiếp giữa máy tính với vi điều khiển, ta phải dùng thêm mạch chuyển mức logic từ TTL => 232 và ngược lại. Các vi mạch thường sử dụng là MAX232 của Maxim hay DS275 của Dallas. Mạch chuyển mức logic mô tả như sau:



Hình 4.2A: Mạch kết nối PIC với máy tính

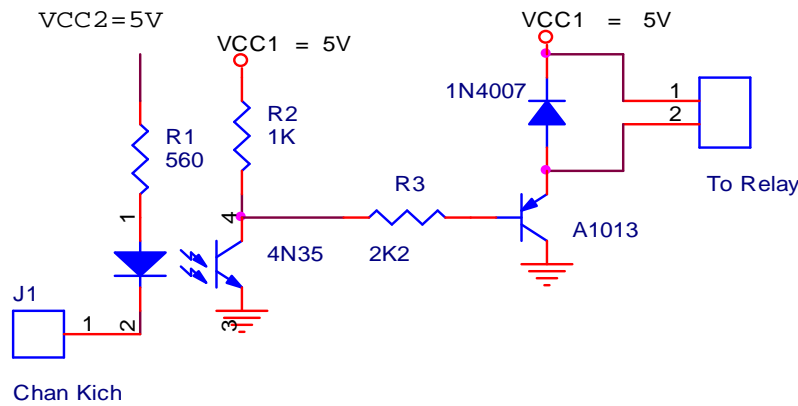
Khi giao tiếp, vi điều khiển chính là một DTE nên sẽ nối RxD của máy tính với TxD của vi điều khiển và ngược lại.

Mạch layout như sau:



Hình 4.2B: Mạch layout kết nối PIC với máy tính

### 4.3. Khối Relay Kích Tải



Hình 4.3A: Mạch kích tải

Tín hiệu điều khiển được xuất ra ngoài để kích On/Off relay thông qua các chân của port D. Với opto 4N35 có nhiệm vụ cách ly khối xử lý trung tâm với thiết bị - mục đích bảo vệ. Để opto dẫn thì ta chọn điện áp đặt trên diode của opto là 2V, dòng là 5mA. Như vậy:

$$R_1 = 5 - 2 / 5 = 600 \text{ ohm}$$

chọn  $R_1 = 560 \text{ ohm}$ .

Hiệu điện thế cách điện là 3350V. Transistor của opto có  $V_{CE} = 0.3$  khi bão hòa, dòng  $I_C = 5\text{mA}$ , vậy có thể tính giá trị  $R_2$  như sau:

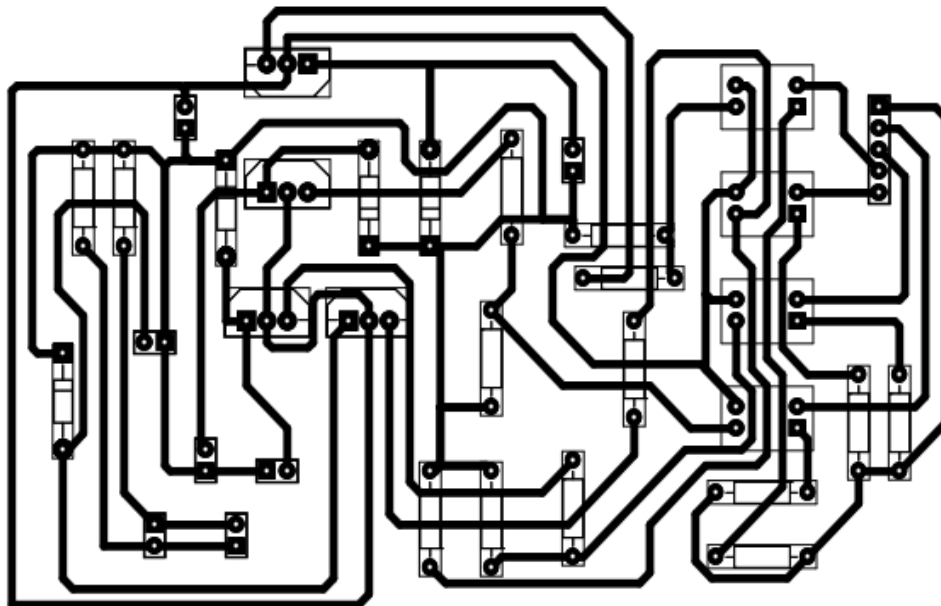
$$R_2 = (V_{CC} - V_{CE}) / I_C = (5 - 0.3) / 5 = 940 \text{ ohm.}$$

Chọn  $R_2 = 1\text{K}$ .

Khi tín hiệu từ chân kích ở mức 0 thì diode của opto dẫn làm cho transistor của nó bão hòa và lúc đó điện áp kích cho transistor A1013 sẽ ở mức 0. Để kích relay đóng ngắt thì trans phải ở thái bão hòa.

Chọn  $R_3 = 2.2\text{K}$  để hạn dòng cho transistor A1013.

Dưới đây mạch layout kích 4 relay:

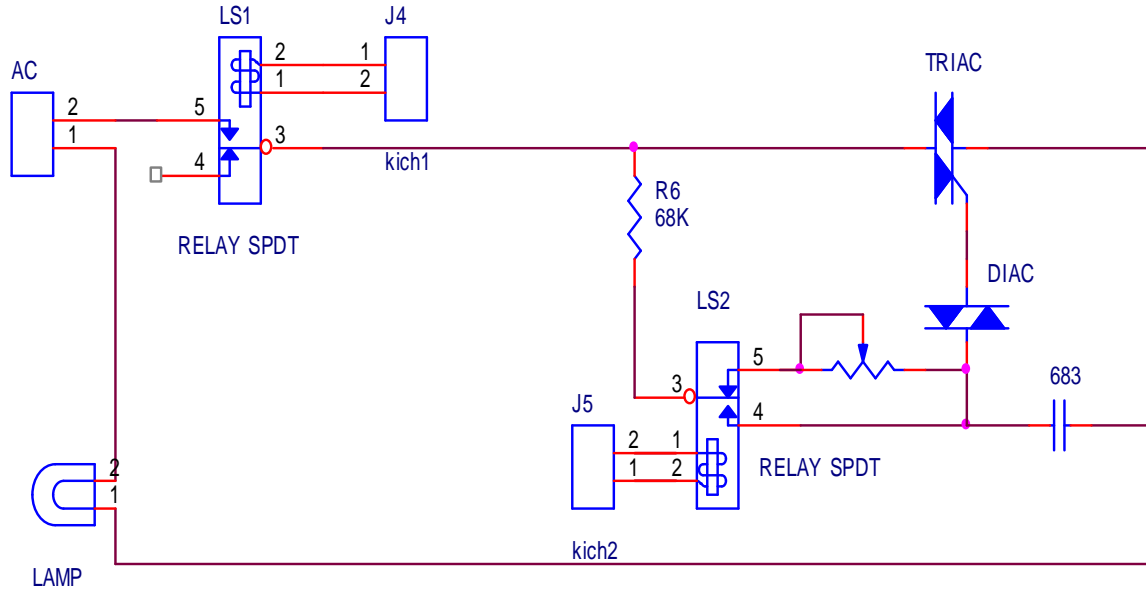


Hình 4.3B: Mạch Layout Kích Tải



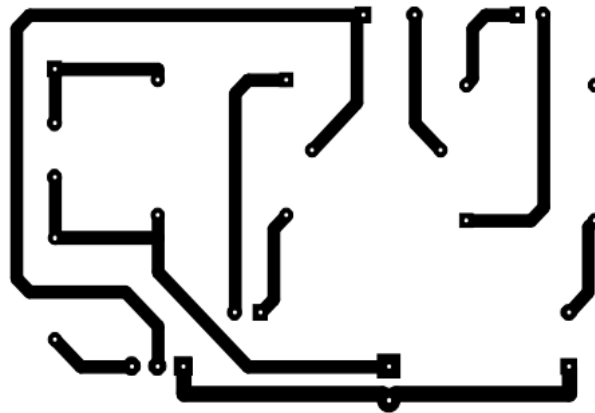


- Diac X805: định ngưỡng điện áp để triac dẫn.
- $R=68K$ : điện trở hạn chế.
- $C=683$ : tụ tạo điện áp ngưỡng để mở thông triac và mở thông diac.
- $V_R=500K$  : Biến trở điều chỉnh khoảng thời gian dẫn của triac.



Hình 4.5A: Mạch điều chỉnh điện áp

Nguyên lý hoạt động: khi relay LS1 được kích, đóng mạch, nguồn 220V hình sin được cấp, tại thời điểm điện áp đổi dấu triac chưa dẫn, tụ C được nạp. Điện áp trên tụ tăng dần cho đến khi điện áp trên tụ đạt ngưỡng mở thông diac, có dòng điều khiển chạy vào cực điều khiển của triac, triac được mở tại thời điểm đó cho tới khi dòng điện của nó bằng 0. Khi biến trở  $V_R$  thay đổi, hằng số nạp tụ thay đổi, thời điểm mở triac thay đổi, thời gian dẫn dòng điện của triac thay đổi, điện áp và dòng điện đưa vào thiết bị cũng thay đổi theo.



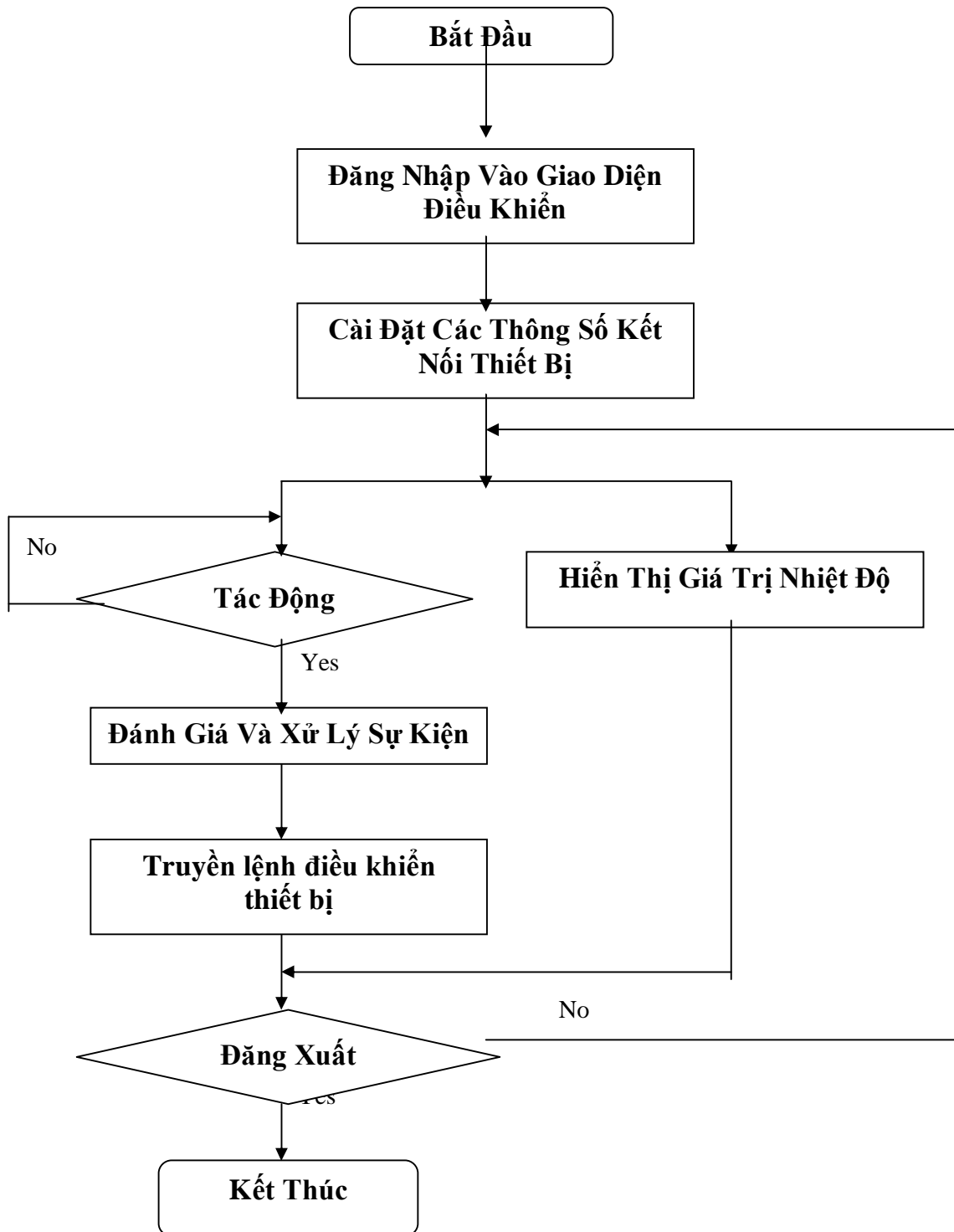
Hình 4.5B: Mạch layout điều chỉnh điện áp

# **CHƯƠNG 5**

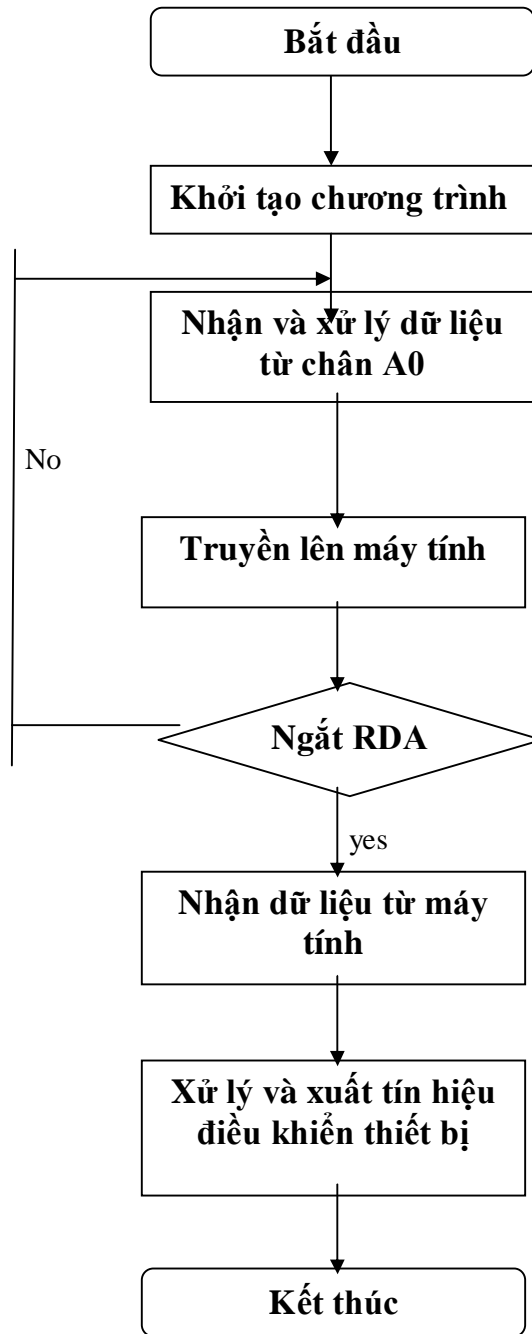
## **LƯU ĐỒ GIẢI THUẬT VÀ CHƯƠNG TRÌNH**

### 5.1. Xây Dựng Giao Diện Người Dùng Trên Máy Tính

Giao diện được thiết kế bằng ngôn ngữ lập trình Visual Basic 6.0. Thông qua các thiết bị như chuột, bàn phím người dùng có thể đưa ra lệnh điều khiển thiết bị. Với sơ đồ giải thuật như sau:

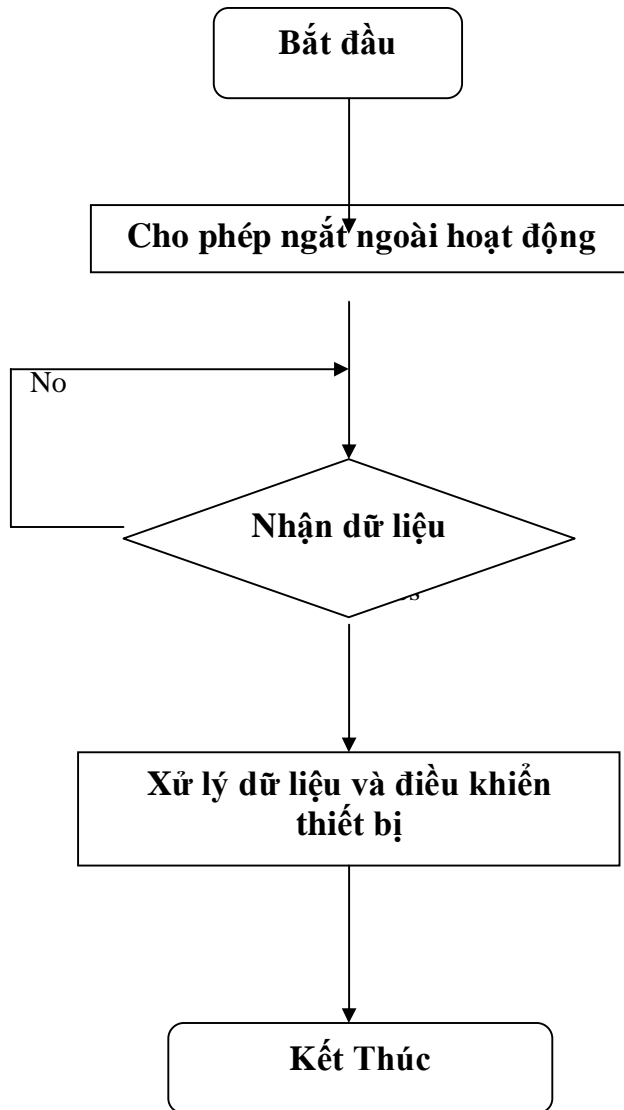


### 5.2. Xây Dựng Chương Trình Điều Khiển Trên Board Xử Lý Trung Tâm

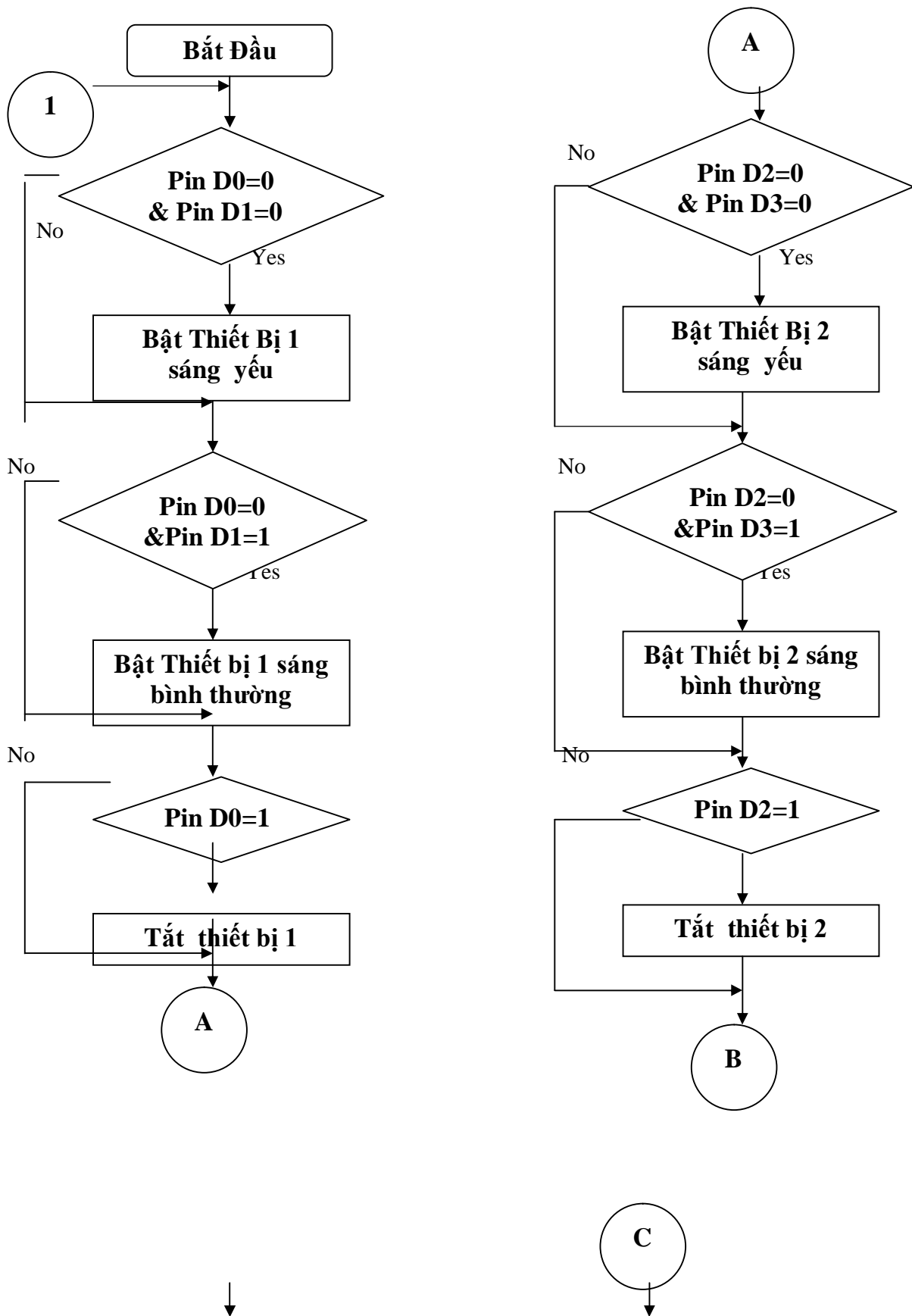


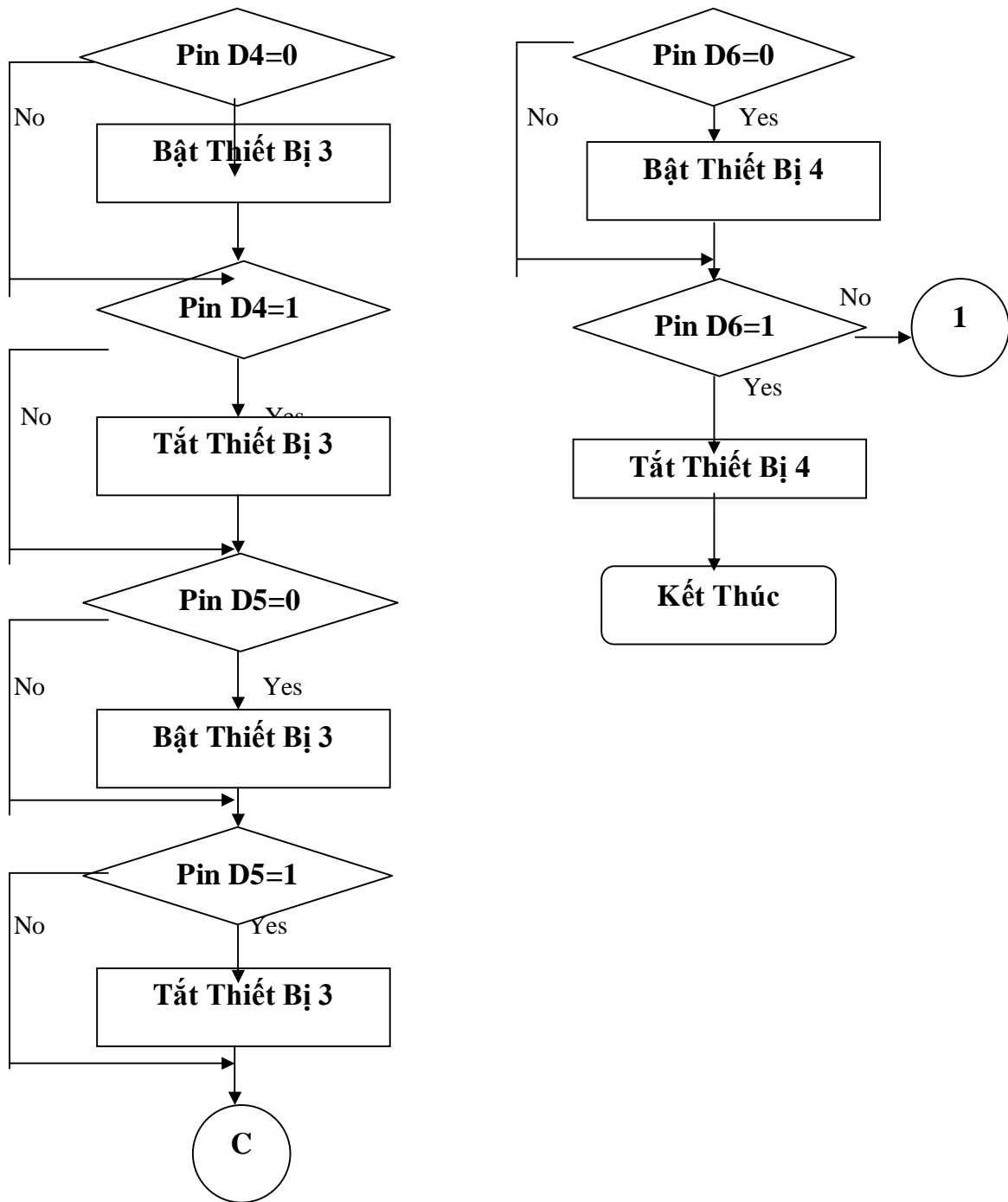
### 5.3. Xây Dựng Chương Trình Điều Khiển Trên Board Xử Lý Đầu Cuối

#### 5.3.1. Lưu đồ chương trình trên board xử lý đầu cuối



5.3.2. Lưu đồ chương trình xử lý dữ liệu và điều khiển thiết bị:







## **CHƯƠNG KẾT LUẬN**

## 1. Tự Đánh Giá Kết Quả

Sau hơn 4 tuần làm việc tương đối nghiêm túc với sự hướng dẫn tận tình của giáo viên hướng dẫn, sự góp ý của bạn bè và sự nỗ lực của bản thân, đồ án Điều Khiển Và Giám Sát Thiết Bị Điện Gia Đình đã hoàn thành đúng thời gian quy định, với những ưu khuyết điểm như sau:

### Ưu điểm:

- Điều khiển được thiết bị từ máy tính, tiện dụng khi có nhu cầu bật / tắt nhiều thiết bị trong thời gian ngắn.
- Có thể cài đặt thời gian bật / tắt thiết bị.
- Theo dõi và cảnh báo nếu nhiệt độ nằm ngoài khoảng giới hạn và tự động điều khiển thiết bị cài đặt trước.
- Lưu trữ dữ liệu quá trình.

### Khuyết điểm:

- Số lượng thiết bị được điều khiển giới hạn.
- Trạng thái thiết bị không tự động báo báo.
- Kiến thức tin học và viễn thông còn hạn chế nên việc áp dụng chưa triệt để.

## 2. Hướng Phát Triển Đề Tài:

- Mở rộng phần cứng, phần mềm để có thể điều khiển được nhiều thiết bị hơn.
- Chuyển sang công nghệ không dây, việc điều khiển sẽ thuận lợi hơn.