

TÀI LIỆU THỰC HÀNH VI XỬ LÝ

THS. LƯU HOÀNG

Tài liệu dành cho sinh viên đại học và cao đẳng.

TRƯỜNG ĐẠI HỌC BÀ RỊA – VŨNG TÀU
VIỆN CNTT - ĐIỆN - ĐIỆN TỬ

MỤC LỤC

MỘT SỐ QUY ĐỊNH VÀ CÁC LỖI THƯỜNG GẶP	0
BÀI 1.1 ĐIỀU KHIỂN LED CHỚP TẮT.....	2
BÀI 1.2 ĐIỀU KHIỂN LED SÁNG VÀ TẮT DẦN.....	5
BÀI 1.3 ĐIỀU KHIỂN LED SÁNG DỒN.....	10
BÀI 2.1 HIỂN THỊ SỐ ĐẾM TRÊN LED 7 ĐOẠN.....	18
BÀI 2.2 ĐIỀU KHIỂN LED 7 ĐOẠN BẰNG PHƯƠNG PHÁP QUÉT	22
BÀI 3.1 ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC QUAY LIÊN TỤC.....	26
BÀI 3.2 ĐIỀU KHIỂN GÓC QUAY, CHIỀU QUAY, TỐC ĐỘ ĐỘNG CƠ BƯỚC.....	30
BÀI 4.1 CHƯƠNG TRÌNH HIỂN THỊ KÝ TỰ TRÊN LED MA TRẬN	34
BÀI 5.1 CHƯƠNG TRÌNH DELAY SỬ DỤNG TIMER	42
BÀI 5.2 CHƯƠNG TRÌNH ĐẾM GIỜ SỬ DỤNG NGẮT CỦA TIMER	45
BÀI 5.3 QUÉT PHÍM MA TRẬN VÀ HIỂN THỊ TRÊN LED 7 ĐOẠN.....	53
BÀI 5.4 ĐIỀU KHIỂN ĐÈN GIAO THÔNG.....	58
BÀI 6.1 MẠCH ĐIỀU KHIỂN LED ĐƠN	66
BÀI 6.2 MẠCH ĐIỀU KHIỂN LED 7 ĐOẠN.....	67
BÀI 6.3 MẠCH ĐIỀU KHIỂN LED MA TRẬN.....	68
BÀI 6.4 MẠCH QUÉT PHÍM MA TRẬN HIỂN THỊ TRÊN LED 7 ĐOẠN.....	69
BÀI 6.5 MẠCH ĐIỀU KHIỂN ĐÈN GIAO THÔNG.....	70
HƯỚNG DẪN MÔ PHỎNG BẰNG PROTEUS	71
HƯỚNG DẪN NẠP CHƯƠNG TRÌNH CHO VI ĐIỀU KHIỂN	75

MỘT SỐ QUY ĐỊNH VÀ CÁC LỖI THƯỜNG GẶP

Một số qui định về kết nối:

- Trên hệ thống có 4 port 0, 1, 2, 3 và các port có ghi trên hệ thống.
- Port 0, 1, 3 với các bit thứ 0 [LSB] đến bit thứ 7 [MSB] theo thứ tự từ trên xuống.
- Riêng port 2 thì ngược lại từ dưới lên. Trên bộ thí nghiệm đã có tên cho từng ngõ ra.
- Khi kết nối chú ý phải theo thứ tự bit 0 của port với bit 0 của đối tượng điều khiển.
- Khi bit thứ 0 đúng thì các bit còn lại sẽ đúng.
- Tất cả các chương trình trong hệ thống này đều được kiểm tra rất kỹ và viết đúng theo thứ tự kết nối trên.
- Nếu 1 yêu cầu nào đó không đúng thì hãy xem lại phần kết nối và chương trình.

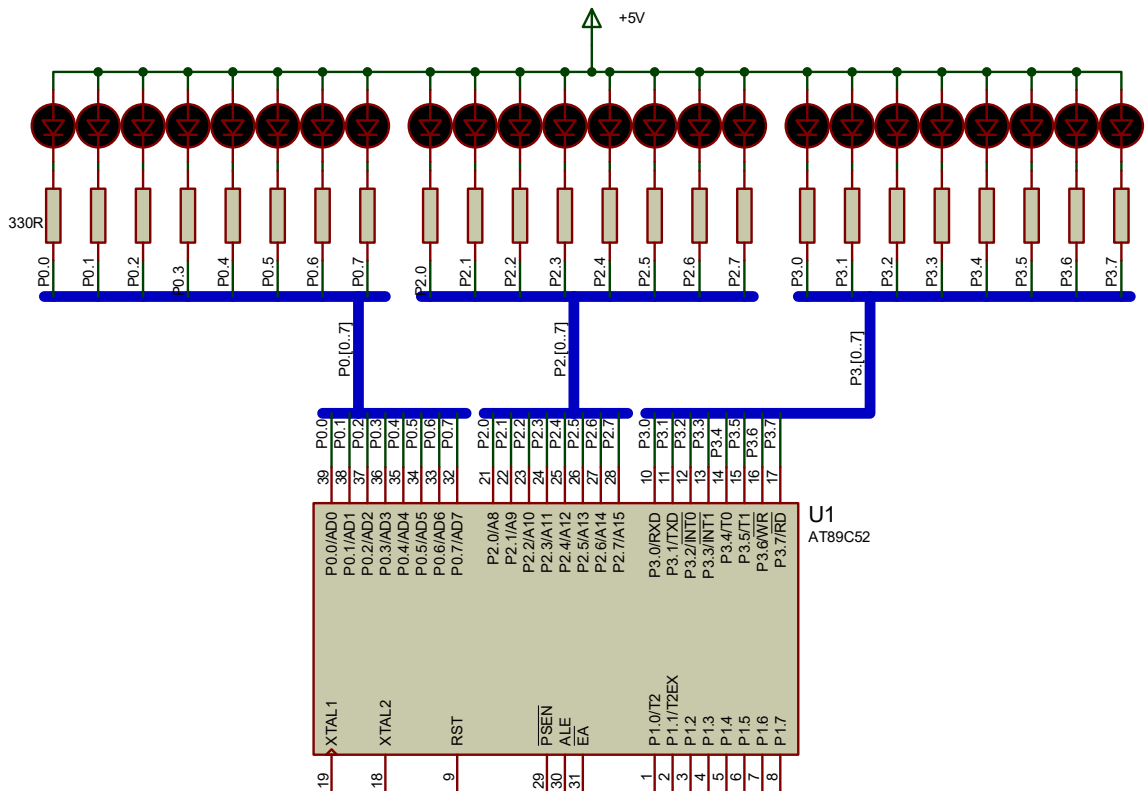
Chú ý: khi viết chương trình thường xảy ra các lỗi như sau:

- Số không “0” thường được đánh nhầm bằng chữ o.
- Sau lệnh end thì không có hàng hay một ký tự nào nếu không thì khi biên dịch chương trình sẽ thông báo có lỗi. Lỗi này có thể bỏ qua.
- Các nhãn trong chương trình phải đánh đúng như trong sách hướng dẫn.
- Các chú thích cho các lệnh thì phải nằm sau dấu chấm phẩy “;”. Có phần chú thích hay không có cũng được.
- Giữa lệnh và thanh ghi phải có khoảng trắng, giữa “org “ và địa chỉ phải có khoảng trắng.
- Hãy dùng nút tab để viết chương trình cho thẳng hàng để dễ xem và tìm lỗi nhanh chóng.
- Nếu đánh 1 chương trình nào đó trong tài liệu hướng dẫn mà chương trình chạy không đúng thì hãy xem kỹ lại có đánh đầy đủ tất các lệnh của chương trình hay chưa? Tất cả các chương trình trong tài liệu hướng dẫn đã được chạy thử và luôn luôn đúng.
- Khi bạn tự viết một chương trình thì sẽ có 1 số trường hợp chương trình không nạp được có thể do IC vi điều khiển hỏng thì bạn có thể nạp một chương trình nào đó đã chạy tốt, nếu vẫn không nạp được thì IC chắc chắn đã hỏng. Còn nếu nạp được và chạy tốt thì lỗi nằm ở chương trình mới viết và nguyên nhân có thể là chương trình bị xếp chồng bộ nhớ.
- Một trong những lý do làm cho chương trình nhưng không chạy là do bạn viết chương trình không bắt đầu tại địa chỉ 0000H,
- Hãy đọc kỹ các yêu cầu trước khi thực hành.

PHẦN 1:

ĐIỀU KHIỂN LED ĐƠN

(MỨC 0 LED SÁNG, MỨC 1 LED TẮT)



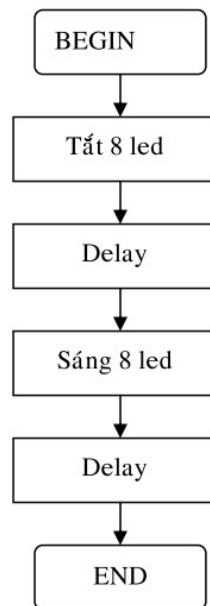
BÀI 1.1 ĐIỀU KHIỂN LED CHỚP TẮT

I. Mục đích yêu cầu:

- Nắm vững lệnh điều khiển xuất dữ liệu ra các port, biết cách viết chương trình con delay.
- Làm quen với phần mềm soạn thảo chương trình, cách hiệu chỉnh lỗi.

II. Trình tự thực hiện:

1. Giải thuật:



2. Kết nối mạch theo trình tự:

- Dùng bus dây (8 sợi) kết nối port 0 với một trong 3 PINHD của dãy 24 led đơn.

3. Khởi động phần mềm, tạo File mới, biên soạn chương trình sau:

```

;-----
;chương trình điều khiển 8 led chớp tắt kết nối với port 0
;-----
Org    0000h        ;khai báo địa chỉ bắt đầu của chương trình
rep:   mov    p0,#0ffh    ; nạp FF vào port0 để tắt 8 led
        lcall   delay      ; gọi chương trình con delay
        mov    p0,#0h     ; nạp 00 vào port0 để sáng 8 led
        lcall   delay      ; gọi chương trình con delay
        sjmp  rep         ; nhảy đến để làm lại từ đầu
;-----
;chương trình con delay
;-----

```

```

delay: mov  r6,#0ffh      ;nap hang so FFH vào thanh ghi r6
del:   mov  r7,#0ffh      ;nap hang so FFH vào thanh ghi r7
       djnz r7,$          ;giam r7 đi 1 và nhảy khi r7 khác 00
       djnz r6,del        ;giam r6 đi 1 và nhảy khi r6 khác 00
       ret                ;thoat khỏi chương trình con
       end

```

4. Lưu chương trình và biên dịch chương trình. Kiểm tra lỗi và hiệu chỉnh rồi biên dịch lại.
5. Nạp chương trình vào vi điều khiển.
6. Quan sát kết quả hiển thị của chương trình, nếu kết quả hiển thị không đúng yêu cầu đề bài thì phải quay lại chương trình chỉnh sửa và làm lại.

III. Chương trình mẫu:

```

;-----
;chương trình điều khiển port 16 led chớp tắt dùng port0, 2
;-----
       org    0000h
rep:   mov    p0,#0ffh      ;tat port0
       mov    p2,#0ffh      ;tat port2
       lcall  delay         ;delay
       mov    p0,#0h        ;sang 8 led
       mov    p2,#0h        ;sang 8 led
       lcall  delay         ;delay
       sjmp  rep            ;lam lai tu dau
;-----
;chương trình con delay
;-----
delay: mov    r6,#0ffh
del:   mov    r7,#0ffh
       djnz  r7,$
       djnz  r6,del
       ret
       end

```

IV. Bài tập:

1. Hãy xem chương trình mẫu điều khiển 16 led chớp tắt dùng 2 port 0 và 2 và hãy viết chương trình sáng tắt 3 port 0, 2 và 3.
2. Hãy viết chương trình sáng tắt xen kẽ 2 port: port0, port2.

3. Hãy viết chương trình sáng tuần tự: port 0, 2 và 3.
4. Hãy viết chương trình sáng xen kẽ từng led trên 3 port 0, 2 và 3.
5. Tìm hiểu thuật toán của chương trình DELAY. Từ chương trình mẫu, hãy tính toán để có chương trình delay với thời gian mong muốn.
6. Hãy cho biết ý nghĩa của lệnh: DJNZ, SJMP và LCALL.

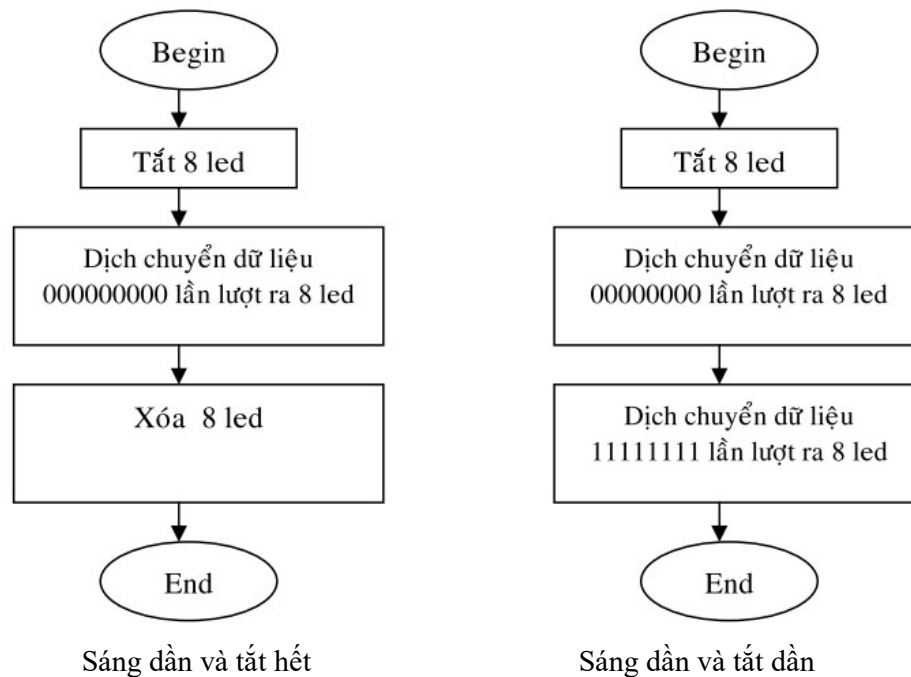
BÀI 1.2 ĐIỀU KHIỂN LED SÁNG VÀ TẮT DẦN

I. Mục đích yêu cầu:

Hiểu cách sử dụng lệnh xoay 8 bit, lệnh nhảy có điều kiện để thực hiện chương trình điều khiển led sáng dần, tắt dần.

II. Trình tự thực hiện:

1. Giải thuật:



2. Kết nối mạch theo trình tự :

- Dùng bus dây (8 sợi) kết nối port 0 với một trong 3 PINHD của dây 24 led đơn.

3. Khởi động phần mềm, tạo File mới, biên soạn chương trình sau:

```

;-----
;chương trình điều khiển port 0 sáng dần lên và tắt hết - cách 1
;kết nối port 0 đến 8 led bằng cấp 8 sợi
;-----

    org 0000h
rep:  mov  p0,#11111111b    ;tắt port 0
      lcall delay          ;gọi chương trình con delay
      mov  p0,#11111110b   ;sáng 1 led
      lcall delay          ;gọi chương trình con delay
      mov  p0,#11111100b   ;sáng 2 led
  
```



```

lcall delay          ;goi chuong trinh con delay
mov  p0,#11111000b  ;sang 3 led
lcall delay          ;goi chuong trinh con delay
mov  p0,#11111000b  ;sang 4 led
lcall delay          ;goi chuong trinh con delay
mov  p0,#11100000b  ;sang 5 led
lcall delay          ;goi chuong trinh con delay
mov  p0,#11000000b  ;sang 6 led
lcall delay          ;goi chuong trinh con delay
mov  p0,#10000000b  ;sang 7 led
lcall delay          ;goi chuong trinh con delay
mov  p0,#00000000b  ;sang 8 led
lcall delay          ;goi chuong trinh con delay
sjmp rep            ;sang het 8 led, lap lai tu dau
;-----
;chuong trinh con delay
;-----
delay: mov  r6,#0ffh
del:  mov  r7,#0ffh
      djnz r7,$
      djnz r6,del
      ret
      end

```

Trong lập trình có nhiều cách viết chương trình từ đơn giản dễ hiểu nhưng dài dòng đến chương trình phức tạp khó hiểu nhưng ngắn gọn tùy thuộc vào đối tượng nghiên cứu và đối tượng học. Ở đây trình bày luôn cả 2 cách viết.

Trong cách viết trên ta thấy chương trình dễ hiểu nhưng khá dài. Hãy cho chạy chương trình trên và xem cách viết thứ 2.

```

;-----
;chuong trinh dieu khien port 0 sang dan len va tat het – cach 2
;ket noi port 0 den 8 led bang cap 8 soi
;-----
      Org  0000h
rep:  mov  p0,#0ffh    ;tat port 0
rep1: lcall delay      ;goi chuong trinh con delay
      clr  c           ;lam cho bit C = 0

```

```

mov    a,p0        ;chuyen noi dung port0 vao thanh ghi A
rlc    a           ;xoay noi dung thanh ghi A sang trai
mov    p0,a        ;tra lai cho port0
jc     rep1        ;nhay ve de thuc hien tiep
sjmp   rep         ;quay lai tu dau
;-----
;chuong trinh con delay
;-----
delay: mov    r6,#0ffh
del:   mov    r7,#0ffh
      djnz   r7,$
      djnz   r6,del
      ret
      end

```

Giải thích:

Để led sáng dần lên ta phải dịch mức 0 vào thanh ghi A, mức 0 được chứa trong bit C, lệnh xoay thanh ghi A sang trái sẽ dịch mức 0 từ C vào bit A0 của thanh ghi A. Bit A7 sẽ dịch sang bit C.

Trong 8 lần dịch đầu tiên thì sau khi dịch, bit C luôn bằng 1. Nên ta dùng lệnh nhảy có điều kiện khi C = 1 thì nhảy để quay lại tiếp tục thực hiện.

Cho đến lần xoay thứ 9 thì C = 0 thì điều kiện không còn thỏa mãn nên lệnh nhảy có điều kiện thì lệnh nhảy “sjmp” mới được thực hiện để làm lại từ đầu.

4. Thực hiện các bước giống như các bài trước.

III. Chương trình mẫu:

1. Chương trình điều khiển port 0 sáng dần và tắt dần:

```

;-----
;chuong trinh dieu khien port 0 sang dan len va tat dan
;ket noi port 0 den 8 led bang cap 8 soi
;-----
      org    0000h
rep:   mov    p0,#0ffh    ;tat port 0
rep1:  lcall  delay       ;goi chuong trinh con delay
      clr    c           ;lam cho bit C = 0
      mov    a,p0        ;chuyen noi dung port0 vao thanh ghi A
      rlc    a           ;xoay noi dung thanh ghi A sang trai
      mov    p0,a        ;tra lai cho port0
      jc     rep1        ;nhay ve de thuc hien tiep khi C = 1

```

```

rep2: lcall  delay      ;goi chuong trinh con delay
      setb   c          ;lam cho bit C = 1
      mov   a,p0       ;chuyen noi dung port0 vao thanh ghi A
      rlc   a          ;xoay noi dung thanh ghi A sang trai
      mov   p0,a       ;tra lai cho port0
      jnc   rep2       ;nhay ve de thuc hien tiep khi C = 0
      sjmp  rep        ;quay ve lam lai tu dau
delay: mov   r6,#0ffh
del:   mov   r7,#0ffh
      djnz  r7,$
      djnz  r6,del
      ret
      end

```

2. Chương trình điều khiển port 0 và port 2 sáng dần và tắt dần:

```

;-----
;chuong trinh dieu khien port 0 va port 2 sang dan len va tat dan
;ket noi port 0 den 8 led bang cap 8 soi, ket noi tuong tu cho port 2
;-----
      Org   0000h
rep:   mov   p0,#0ffh   ;tat port 0
      mov   p2,#0ffh   ;tat port 2
rep1:  lcall  delay     ;goi chuong trinh con delay
      clr   c          ;lam cho bit C = 0
      mov   a,p0       ;chuyen noi dung port0 vao thanh ghi A
      rlc   a          ;xoay noi dung thanh ghi A sang trai
      mov   p0,a       ;tra lai cho port0
      mov   a,p2       ;chuyen noi dung port2 vao thanh ghi A
      rlc   a          ;xoay noi dung thanh ghi A sang trai
      mov   p2,a       ;tra lai cho port2
      jc    rep1       ;nhay ve de thuc hien tiep khi C = 1
rep2:  lcall  delay     ;goi chuong trinh con delay
      setb  c          ;lam cho bit C = 1
      mov   a,p0       ;chuyen noi dung port0 vao thanh ghi A
      rlc   a          ;xoay noi dung thanh ghi A sang trai
      mov   p0,a       ;tra lai cho port0

```

```

mov    a,p2        ;chuyen noi dung port2 vao thanh ghi A
rlc    a           ;xoay noi dung thanh ghi A sang trai
mov    p2,a        ;tra lai cho port2
jnc    rep2        ;nhay ve de thuc hien tiep khi C = 0
sjmp   rep        ;quay ve lam lai tu dau
delay: mov    r6,#0ffh
del:   mov    r7,#0ffh
      djnz   r7,$
      djnz   r6,del
      ret
      end

```

IV. Bài tập:

1. Dùng port 0 kết nối với 8 led, hãy viết chương trình điều khiển 1 led sáng và di chuyển từ trái sang phải.
2. Tương tự, hãy viết chương trình điều khiển 1 led sáng và di chuyển từ phải sang trái trên 3 port 0, 2 và 3.
3. Hãy viết chương trình điều khiển 3 port 0, 2, 3 sáng dần và tắt dần từ trái sang phải và từ phải sang trái.
4. Hãy viết chương trình điều khiển 3 port 0, 2, 3 sáng dần và tắt dần từ ngoài vào trong và từ trong ra ngoài.
5. Hãy cho biết ý nghĩa của các lệnh: RL, RLC, RR, RRC trong chương trình.
6. Cho biết ý nghĩa của các lệnh: JC, JNC, JB, và JNB.

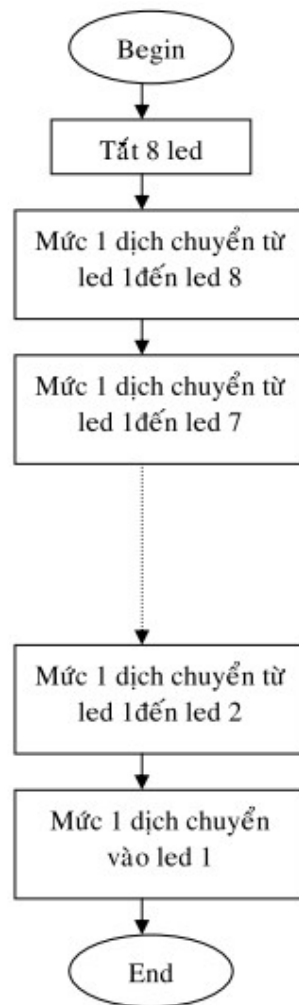
BÀI 1.3 ĐIỀU KHIỂN LED SÁNG DỒN

I. Mục đích yêu cầu:

Hiểu cách sử dụng lệnh xoay kết hợp với lệnh logic để thực hiện chương trình điều khiển led sáng dồn.

II. Trình tự thực hiện:

1. Giải thuật:



2. Kết nối mạch theo trình tự :

- Dùng bus dây (8 sợi) kết nối port 0 với một trong 3 PINHD của dây 24 led đơn.

3. Khởi động phần mềm, tạo File mới, biên soạn chương trình sau:

```

;-----
;chương trình điều khiển port 0 sang đơn và tắt hết - cách 1
;kết nối port 0 đến 8 led bằng cáp 8 sợi

```

```
;-----  
    org    0800h                ;khai bao dia chi luu tru vung du lieu  
data: db    11111111b  
      db    11111110b  
      db    11111101b  
      db    11111011b  
      db    11110111b  
      db    11101111b  
      db    11011111b  
      db    10111111b  
      db    01111111b  
;lan thu hai la 7 byte  
      db    01111110b  
      db    01111101b  
      db    01111011b  
      db    01110111b  
      db    01101111b  
      db    01011111b  
      db    00111111b  
;lan thu 3 la 6 byte  
      db    00111110b  
      db    00111101b  
      db    00111011b  
      db    00110111b  
      db    00101111b  
      db    00011111b  
;lan thu 4 la 5 byte  
      db    00011110b  
      db    00011101b  
      db    00011011b  
      db    00010111b  
      db    00001111b  
;lan thu 5 la 4 byte  
      db    00001110b  
      db    00001101b  
      db    00001011b
```

```

        db      00000111b
;lan thu 6 la 3 byte
        db      00000110b
        db      00000101b
        db      00000011b
;lan thu 7 la 2 byte
        db      00000010b
        db      00000001b
;lan thu 8 la 1 byte
        db      00000000b      ;byte du lieu cuoi cung = 00h
;-----
;bat dau chuong trinh chinh
;-----
        org    0000h      ;khai bao dia chi chuong trinh chinh
rep:    mov    dptr,#0800h ;nap dia chi luu du lieu vao thghi dptr
rep1:   clr    a
        movc  a,@A+dptr   ;lay du lieu tu bo nho dua vao A
        mov   p0,a        ;goi ra port 0
        lcall delay      ;goi chtr con delay
        inc   dptr        ;tang dptr len o nho ke
        cjne a,#00h,rep1  ;ktra co phai la byte ket thuc hay chua
        sjmp  rep         ;quay tro lam lai khi da het du lieu
;-----
delay:  mov    r6,#0ffh
del:    mov    r7,#0ffh
        djnz  r7,$
        djnz  r6,del
        ret
        end

```

Theo cách viết 1 ta hãy quan sát dữ liệu trong chương trình đã được sắp xếp theo đúng trình tự và chương trình chỉ thực hiện nhiệm vụ là di chuyển lần lượt các byte dữ liệu có trong bộ nhớ đem gửi vào A và sau đó gửi ra port 0.

Lệnh “data: db dữ liệu “ có chức năng nạp các byte dữ liệu vào vùng nhớ ROM có địa chỉ 0800H.

Byte cuối cùng là 00H là byte báo cho biết hết dữ liệu.

Dữ liệu viết dưới dạng số nhị phân cho dễ nhìn thấy và có thể viết dưới dạng số hex – khi đó chương trình sẽ ngắn hơn rất nhiều. Phần khai báo dữ liệu dưới dạng số hex như sau:

```

org    0800h        ;khai bao dia chi luu tru vung du lieu
data:  db    0ffh
      db    0feh,0fdh,0fbh,0f7h,0efh,0dfh,0bfh,07fh
      db    7eh,7dh,7bh,77h,6fh,5fh,3fh
      db    3eh,3dh,3bh,37h,2fh,1fh
      db    1eh,1dh,1bh,17h,0fh
      db    0eh,0dh,0bh,07h
      db    06h,05h,03h
      db    02h,01h
      db    00h

```

```

;-----
;chuong trinh dieu khien 8 led sang don va tat het dung port 0 – cách 2
;ket noi port 0 den 8 led bang 1 soi cap 8 soi

```

```

;-----
;đinh nghĩa các biến

```

```

      biendem     equ    30H    ;biên đếm
      bientam     equ    31h    ;biên tạm
      x0          equ    r0
      y0          equ    20h
      led0        equ    p0

```

```

;-----
;bat dau chuong trinh chinh

```

```

      org    0000h
      mov    led0,#0ffh
      lcall  delay
rep:   mov    x0,#11111111b    ;X0 luu trang thai ban dau
      mov    biendem,#08    ;bien dem so lan dich chuyen LAN DAU = 8
rep2:  mov    bientam,biendem ;chuyen biendem sang bientam
      mov    y0,#11111110b
rep1:  mov    a,y0
      anl   a,x0            ;lay x0 and voi y0 roi goi ra led0
      mov   led0,a         ;xuat ra led
      lcall  delay
      mov   a,y0

```



```

    rl    a
    mov  y0,a
    djnz bientam,rep1      ;giam bientam neu chua bang 0 thi quay ve lai
    mov  x0,led0          ;cat noi dung sau cung khi da dich chuyen 1 led
    djnz biendem,rep2     ;giam bien dem de xu li lan ke
    sjmp rep              ;nhay ve lam lai tu dau
;-----
delay: mov  r6,#250
del:   mov  r7,#250
       djnz r7,$
       djnz r6,del
       ret
       end

```

III. Chương trình mẫu:

Chương trình điều khiển 16 led sáng dần

```

;-----
;chuong trinh dieu kien 16 led sang don dung port 0, 2
;ket noi port 0 va port 2 den 16 led bang 2 soi cap 8 soi
;-----
;dinh nghia cac bien
    biendem    equ    30h
    bientam    equ    31h
    x0         equ    r0
    x1         equ    r1
    y0         equ    20h
    y1         equ    21h
    led0       equ    p0
    led1       equ    p2
;-----
;bat dau chuong trinh chinh
;-----
    org    0000h
    mov    led0,#0ffh
    mov    led1,#0ffh
    lcall  delay
rep:  mov    x0,#11111111b    ;X0 luu trang thai ban dau

```

```

        mov    x1,#11111111b    ;X1 lưu trạng thái ban đầu
        mov    biendem,#16     ;biên đếm số lần dịch chuyển lần đầu = 16
rep2:   mov    bientam,biendem  ;chuyển biendem sang bientam
        mov    y0,#11111110b
        mov    y1,#11111111b
rep1:   mov    a,y0
        anl    a,x0             ;lấy x0 and với y0 rồi gọi ra led0
        mov    led0,a          ;xuất ra led0
        mov    a,y1
        anl    a,x1             ;lấy x1 and với y1 rồi gọi ra led1
        mov    led1,a          ;xuất ra led1
        lcall  delay
        setb   c
        mov    a,y0
        rlc    a
        mov    y0,a
        mov    a,y1
        rlc    a
        mov    y1,a
        djnz   bientam,rep1     ;giảm bientam nếu chưa bằng 0 thì quay về lại

        mov    x0,led0         ;cat nội dung sau cùng khi đã dịch chuyển 1 led
        mov    x1,led1         ;cat nội dung sau cùng khi đã dịch chuyển 1 led
        djnz   biendem,rep2    ;giảm biên đếm để xử lý lần kế
        ljmp   rep             ;nhảy về làm lại từ đầu
;-----
delay:  mov    r6,#0ffh
del:    mov    r7,#0ffh
        djnz   r7,$
        djnz   r6,del
        ret
        end

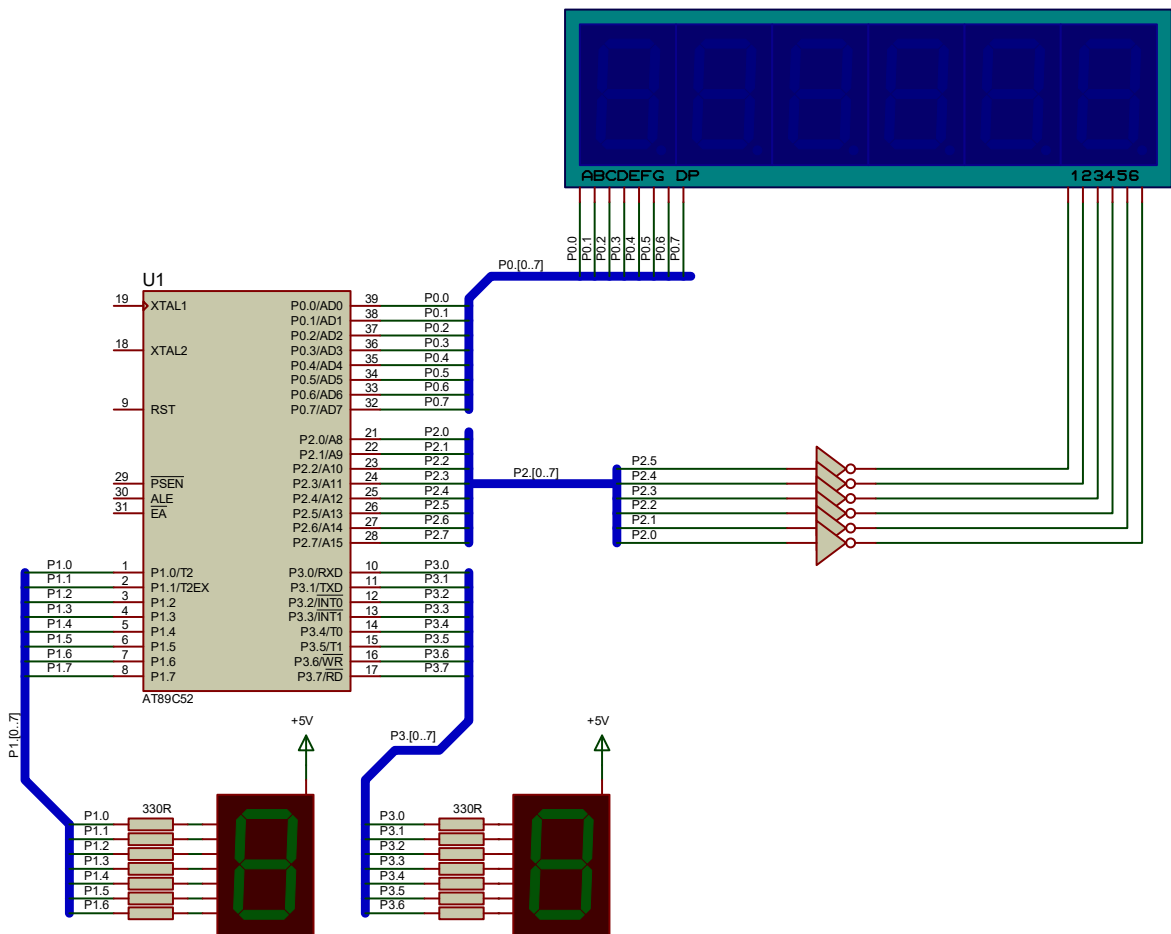
```

IV. Bài tập:

1. Hãy viết chương trình điều khiển 3 port: port0, port2, port3 sáng đèn.
2. Hãy viết chương trình sáng đèn 2 port 0 và 2 từ ngoài vào trong và từ trong ra ngoài.

3. Hãy viết chương trình sáng đèn 4 port 0, 1, 2 và 3 từ ngoài vào trong và từ trong ra ngoài.
4. Hãy cho biết ý nghĩa của lệnh: DB, EQU, ANL, MOVC và CJNE.

PHẦN 2: ĐIỀU KHIỂN LED 7 ĐOẠN (MỨC 0 LED SÁNG)



BÀI 2.1 HIỂN THỊ SỐ ĐẾM TRÊN LED 7 ĐOẠN

I. Mục đích yêu cầu:

Biết cách xây dựng được mã điều khiển led 7 đoạn. Điều khiển led 7 đoạn hiển thị theo yêu cầu bằng nhiều cách khác nhau.

II. Trình tự thực hiện:

1. Kết nối mạch theo trình tự :

- Dùng bus 8 dây kết nối port 3 với một trong led 7 đoạn bên phải trong nhóm 2 led.
- Chú ý: P3.0 kết nối với chân a.

2. Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
; chương trình hiển thị từ 0 đến 9 trên led 7 đoạn – cách 1
;-----
        org 0000h
rep:    mov     p3,#11000000b      ; ma so 0 (C0h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#11111001b     ; ma so 1 (F9h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#10100100b     ; ma so 2 (A4h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#10110000b     ; ma so 3 (B0h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#10011001b     ; ma so 4 (99h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#10010010b     ; ma so 5 (92h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#10000010b     ; ma so 6 (82h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#11111000b     ; ma so 7 (F8h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#10000000b     ; ma so 8 (80h)
        lcall  delay              ;goi chuong trinh con delay
        mov     p3,#10010000b     ; ma so 9 (90h)
        lcall  delay              ;goi chuong trinh con delay

```

```

    sjmp  rep                                ;sang het 8 led, lap lai tu dau
;-----
delay: mov  r5,#4
del1:  mov  r6,#0ffh
del2:  mov  r7,#0ffh
      djnz  r7,$
      djnz  r6,del2
      djnz  r5,del1
      ret
      end

```

3. Thực hiện các bước giống như trên và xem kết quả.

Giải thích :

Chương trình này lần lượt xuất các mã từ 0 đến 9 ra port 0 để hiển thị trên led 7 đoạn.

Từ p3.0 đến p3.6 lần lượt điều khiển các đoạn từ a đến f, p3.7 điều khiển dấu chấm (p). Ở đây ta sử dụng led 7 đoạn loại anode chung nên muốn điều khiển đoạn nào sáng bit điều khiển phải bằng 0.

Chương trình trên dễ hiểu nhưng khá dài, ta có thể tạo một vùng mã của led 7 đoạn trên ROM và lần lượt xuất ra hiển thị như chương trình dưới đây.

```

;-----
; chương trình hiển thị từ 0 đến 9 trên led 7 đoạn – cách 2
;-----
    org  0500h                                ;khai tạo vùng mã led 7 đoạn tại địa chỉ 0500h trên ROM
    db   0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
    org  0000h
    mov  dptr,#0500h                          ; nạp địa chỉ vùng nhớ lưu mã vào dptr
begin: mov  r2,#0                              ; nạp biến đếm vào r2
rep:   mov  a,r2
      movc a,@a+dptr                          ; lấy mã của số đếm vào a
      mov  p3,a                              ; xuất mã của số đếm ra hiển thị
      lcall delay                            ; gọi chương trình con delay
      inc  r2                                ; tăng biến đếm
      cjne r2,#10,rep                        ; kiểm tra biến đếm, chưa bằng 10 thì lặp lại
      sjmp begin                            ; hiển thị đến 9, lặp lại từ đầu
;-----
delay: mov  r5,#4
del1:  mov  r6,#250

```

```

del2:  mov  r7,#250
       djnz r7,$
       djnz r6,del2
       djnz r5,del1
       ret
       end

```

Giải thích :

Ở chương trình trên, các mã của led 7 đoạn được lưu vào vùng ROM ở địa chỉ từ 0500h đến 0509h. Các mã này được lần lượt lấy ra và hiển thị trên led 7 đoạn thông qua lệnh: MOVC. Phương pháp này có thể tận dụng được chính bộ nhớ Flash của vi điều khiển để lưu dữ liệu.

III. Chương trình mẫu:

```

;-----
; chương trình hiển thị từ 00 đến 99 trên 2 led 7 đoạn
; dùng bus 8 dây kết nối thêm port 1 với led 7 đoạn bên trái để hiển thị hàng chục
;-----
       org  0000h
       mov  dptr,#ledcode      ; nạp địa chỉ vùng nhớ lưu mã vào dptr
begin: mov  r2,#0              ; nạp biến đếm hàng đơn vị vào r2
       mov  r3,#0              ; nạp biến đếm hàng chục vào r3
rep:   mov  a,r2                ; nạp biến đếm đơn vị vào a
       movc a,@a+dptr          ; lấy mã của số đếm vào a
       mov  p3,a                ; xuất mã của số đếm ra hiển thị hàng đơn vị
       mov  a,r3                ; nạp biến đếm chục vào a
       movc a,@a+dptr          ; lấy mã của số đếm vào a
       mov  p1,a                ; xuất mã của số đếm ra hiển thị hàng chục
       lcall delay              ; gọi chương trình con delay
       inc  r2                  ; tăng biến đếm đơn vị
       cjne r2,#10,rep          ; kiểm tra biến đếm đơn vị, chưa bằng 10 thì lặp lại
       mov  r2,#0              ; bằng 10 thì nạp lại giá trị 0
       inc  r3                  ; tăng biến đếm chục
       cjne r3,#10,rep          ; kiểm tra biến đếm, chưa bằng 10 thì lặp lại
       sjmp begin              ; bằng 10 thì trở lại từ đầu
;-----
delay: mov  r6,#0ffh
del:   mov  r7,#0ffh

```

```
djnz r7,$  
djnz r6,del  
ret
```

```
-----  
;khởi tạo vùng mã led 7 đoạn trên ROM không cần xác định địa chỉ  
-----
```

ledcode:

```
db 0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h  
end
```

IV. Bài tập:

1. Hãy viết chương trình hiển thị theo số hex từ 0 đến F trên port 3.
2. Viết chương trình hiển thị từ 00 đến 36 rồi lặp lại.
3. Viết chương trình hiển thị từ 99 về 00.

BÀI 2.2 ĐIỀU KHIỂN LED 7 ĐOẠN BẰNG PHƯƠNG PHÁP QUÉT

I. Mục đích yêu cầu:

Biết cách điều khiển hiển thị được nhiều led 7 đoạn bằng phương pháp quét.

II. Trình tự thực hiện:

1. Kết nối mạch theo trình tự :

- Dùng bus 8 dây kết nối port 0 với ngõ vào dữ liệu của nhóm 6 led 7 đoạn. Chú ý: P0.0 kết nối với chân a.
- Dùng bus 8 dây kết nối port 2 với ngõ và chọn led. Led được chọn khi bit điều khiển bằng 0.

2. Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
; chương trình hiển thị số 0 trên led 7 đoạn thứ 3 của nhóm 6 led.
;-----
org 0000h
mov p2,#11111011b ;chọn led 7 đoạn thứ 3
mov p0,#0c0h ;xuất mã số 0 ra port 0
sjmp $ ;nhảy tại chỗ
end

```

3. Thực hiện các bước giống như trên và xem kết quả.

Giải thích :

Để hiển thị số 0 trên led 7 đoạn thứ 3, trước hết cần chọn led muốn hiển thị bằng cách cho bit điều khiển chọn led đó bằng 0 và xuất ra port điều khiển là port 2. Tiếp theo ta xuất dữ liệu cần hiển thị là mã của số 0 ra port 0.

Tương tự, các bạn hãy viết chương trình hiển thị cho các led còn lại.

Để hiển thị đồng thời trên 6 led, ta phải sử dụng phương pháp quét như ví dụ sau:

```

;-----
; chương trình hiển thị chuỗi HELLO- trên 6 led 7 đoạn.
; port 0 xuất dữ liệu, port 2 xuất tín hiệu chọn led.
;-----
org 0000h
rep: mov p2,#11111110b ;chọn led thứ 1
mov p0,#0bfh ;xuất mã hiển thị chuỗi '-'
lcall delay ;gọi chương trình con delay
mov p2,#11111101b ;chọn led thứ 2

```

```

mov    p0,#0c0h        ;xuất ma hien thi chu 'O'
lcall  delay           ;goi chuong trinh con delay
mov    p2,#11111011b   ;chon led thu 3
mov    p0,#0c7h        ;xuất ma hien thi chu 'L'
lcall  delay           ;goi chuong trinh con delay
mov    p2,#11111011b   ;chon led thu 4
mov    p0,#0c7h        ;xuất ma hien thi chu 'L'
lcall  delay           ;goi chuong trinh con delay
mov    p2,#11101111b   ;chon led thu 5
mov    p0,#86h         ;xuất ma hien thi chu 'E'
lcall  delay           ;goi chuong trinh con delay
mov    p2,#11011111b   ;chon led thu 6
mov    p0,#89h         ;xuất ma hien thi chu 'H'
lcall  delay           ;goi chuong trinh con delay
sjmp   rep             ;lap lai tu dau
;-----
delay: mov    r6,#6
del:   mov    r7,#250
      djnz   r7,$
      djnz   r6,del
      ret
      end

```

Giải thích :

Ở chương trình trên, để các led cùng hiển thị thì tất cả các led phải được quét lần lượt với tần số khoảng 50Hz thì mới không bị nhấp nháy. Như vậy thời gian để quét hết các led là: $1/50 = 20\text{ms}$. Thời gian này phải được chia đều cho 6 led, do đó thời gian để hiển thị mỗi led là: $20/6 = 3,34\text{ms}$. Ta chọn thời gian delay là 3ms.

III. Chương trình mẫu:

```

;-----
; chuong trinh hien thi chu HELLO chay tu phai sang trai tren led 7 doan
; port 0 xuất du lieu, port 2 xuất tin hieu chon led.
;-----
biendem    equ    r2    ;biến đếm
tocdo      equ    r3    ;tốc độ
org        0500h        ;khởi tạo vùng du lieu tai dia chi 0500h tren ROM
db         0ffh,0ffh,0ffh,0ffh,0ffh,0ffh ;vung ma de tat led
db         89h,86h,0c7h,0c7h,0c0h    ;vung ma chu HELLO

```

```

db    0ffh,0ffh,0ffh,0ffh,0ffh,0ffh
org   0000h
begin: mov  dptr,#0500h      ;nap dia chi vung nho luu ma vao dptr
repeat: mov  tocd0,#50      ;nap gia tri cho bien toc do
rep1:  mov  biendem,#0      ;nap 0 vao bien dem
       mov  p2,#0dfh        ;chon hien thi led 6
rep2:  mov  a,biendem        ;nap bien dem vao a
       movc a,@a+dptr        ;lay du lieu tren rom vao a
       mov  p0,a             ;xuat du lieu ra port 0
       lcall delay           ;goi chuong trinh con delay
       mov  p0,#0ffh        ;xoa hien thi de chong lem
       mov  a,p2             ;lay p2 vao a
       rr   a                ;xoay phai de chon led tiep theo
       mov  p2,a             ;tra lai p2
       inc  biendem          ;tang bien dem
       cjne biendem,#6,rep2  ;kiem tra bien dem, chua bang 6 thi lap lai
       djnz tocd0,rep1       ;bien toc do chua bang 0, quet lai tu dau
       inc  dptr              ;dich vung du lieu sang phai
       mov  a,dpl
       cjne a,#12,repeat
       sjmp begin            ;lap lai tu dau

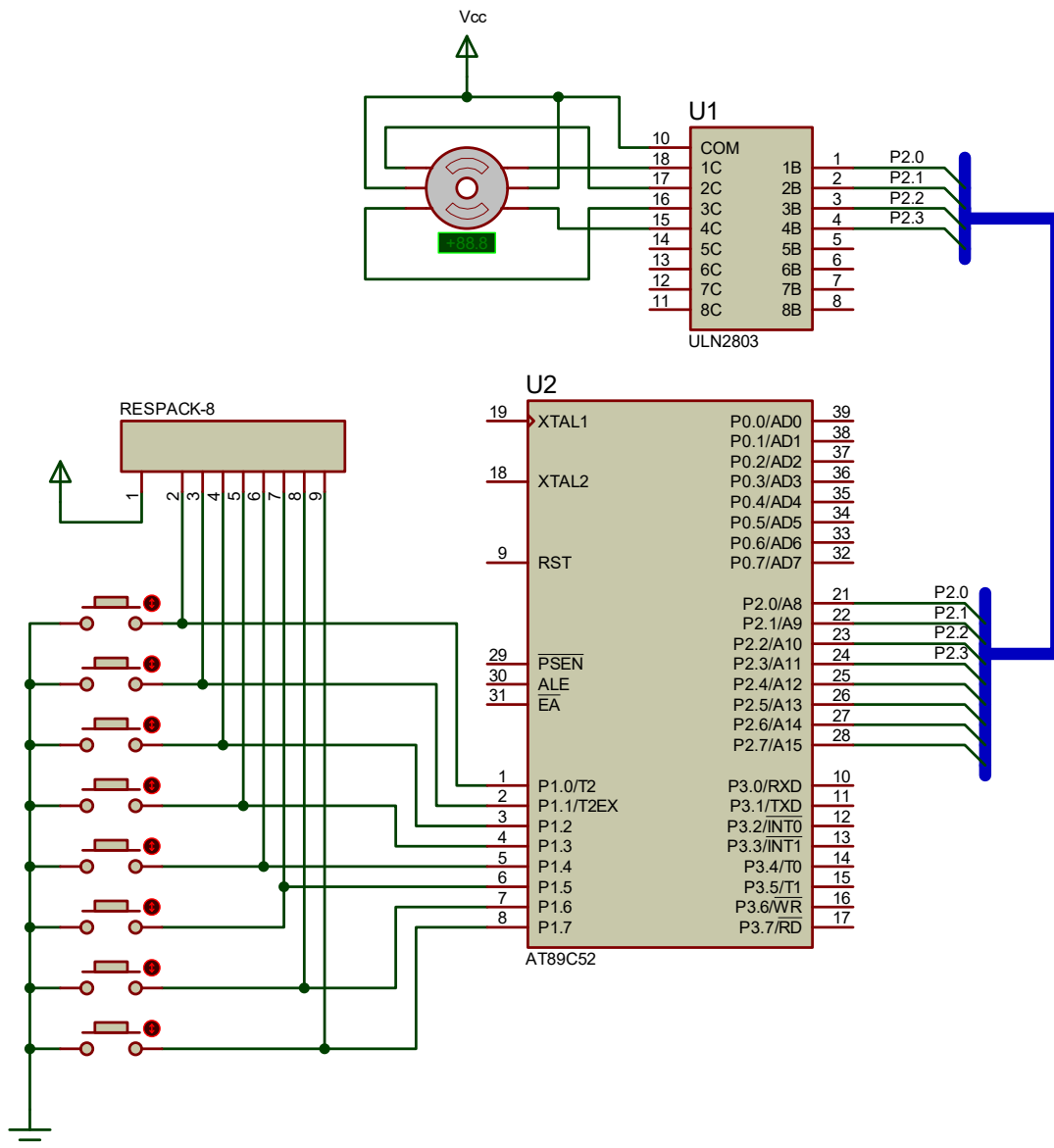
;-----
;chuong trinh con delay 3ms
;-----
delay: mov  r6,#6
del:   mov  r7,#250
       djnz r7,$
       djnz r6,del
       ret
       end

```

IV. Bài tập:

1. Hãy viết chương trình hiển thị từ 0 đến 5 trên 6 led 7 đoạn.
2. Viết chương trình hiển thị một chuỗi số từ 0 đến 9 chạy từ trái sang phải.
3. Viết chương trình hiển thị từ 00 đến 99 trên 2 led 7 đoạn ở giữa.

PHẦN 3: ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC



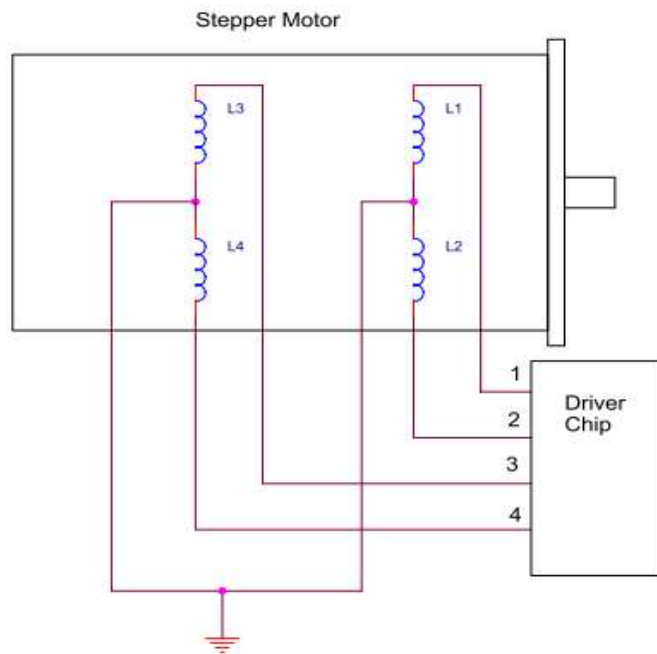
BÀI 3.1 ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC QUAY LIÊN TỤC

I. Mục đích yêu cầu:

Biết cách điều khiển động cơ bước ở chế độ full step và half step. Điều khiển quay thuận và quay nghịch. Nắm được nguyên tắc hoạt động của động cơ. Động cơ sử dụng trong bài này có 4 phase, 25 cực.

II. Trình tự thực hiện:

1. Nguyên tắc hoạt động:



- Điều khiển full step cách 1

STEP	L4	L3	L2	L1
1	L	L	L	H
2	L	L	H	L
3	L	H	L	L
4	H	L	L	L

- Điều khiển full step cách 2

STEP	L4	L3	L2	L1
1	L	L	H	H
2	L	H	H	L

3	H	H	L	L
4	H	L	L	H

- Điều khiển half step:

STEP	L4	L3	L2	L1
1	L	L	L	H
2	L	L	H	H
3	L	L	H	L
4	L	H	H	L
5	L	H	L	L
6	H	H	L	L
7	H	L	L	L
8	H	L	L	H

2. Kết nối mạch theo trình tự :

- Dùng bus 4 dây kết nối port 2 với A,B,C,D của khối động cơ bước.
- Chú ý: P2.0 kết nối với chân A.

3. Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
;chương trình điều khiển động cơ bước quay thuận liên tục full step
;dung bus 4 soi ket noi port 2 khi diu khien dong co buoc
;-----
    Org    0000h
begin: Mov    p2,#01h
        Lcall  delay
        Mov    p2,#02h
        Lcall  delay
        Mov    p2,#04h
        Lcall  delay
        Mov    p2,#08h
        Lcall  delay
        Sjmp   begin
delay:  mov    r6,#200
del:    mov    r7,#250

```

```

Djnz r7,$
Djnz r6,del
Ret
End

```

```

;-----
;chuong trinh dieu khien dong co buoc quay thuan lien tuc half step
;dung bus 4 soi ket noi port 2 khi dieu khien dong co buoc
;-----

```

```

Org 0000h
begin: Mov p2,#01h
      Lcall delay
      Mov p2,#03h
      Lcall delay
      Mov p2,#02h
      Lcall delay
      Mov p2,#06h
      Lcall delay
      Mov p2,#04h
      Lcall delay
      Mov p2,#0ch
      Lcall delay
      Mov p2,#08h
      Lcall delay
      Mov p2,#09h
      Lcall delay
      Sjmp begin
delay: Mov r6,#100
del:   Mov r7,#250
      Djnz r7,$
      Djnz r6,del
      Ret
      End

```

III. Chương trình mẫu:

Chương trình điều khiển động cơ bước quay thuận nghịch bằng 1 nút nhấn.

```

;-----
;chuong trinh dieu khien dong co buoc quay thuan, nghich lien tuc full step

```

;dung bus 4 soi ket noi port 2 khoi dieu khien dong co buoc

;dung bus 2 soi ket noi p3.0 va p3.1 voi 2 nut nhan

```

;-----
      Org    0000h
thuan: mov   p2,#01h
      Lcall  delay
      Mov    p2,#02h
      Lcall  delay
      Mov    p2,#04h
      Lcall  delay
      Mov    p2,#08h
      Lcall  delay
      Jb     p3.0,thuan    ;kien tra nut nhan
nghich: Mov   p2,#08h
      Lcall  delay
      Mov    p2,#04h
      Lcall  delay
      Mov    p2,#02h
      Lcall  delay
      Mov    p2,#01h
      Lcall  delay
      Jnb    p3.0,nghich
      Sjmp   thuan
Delay: mov   r6,#100
Del:   mov   r7,#250
      Djnz  r7,$
      Djnz  r6,del
      Ret
      End

```

IV. Bài tập:

1. Tìm hiểu nguyên tắc hoạt động và các phương pháp điều khiển động cơ bước.
2. Viết chương trình điều khiển động cơ bước quay nghịch full step.
3. Viết chương trình điều khiển động cơ bước quay nghịch half step.
4. Viết chương trình điều khiển động cơ bước full step với 2 nút nhấn điều khiển chiều quay kết nối với P3.0 và P3.1.

BÀI 3.2 ĐIỀU KHIỂN GÓC QUAY, CHIỀU QUAY, TỐC ĐỘ ĐỘNG CƠ BƯỚC

I. Mục đích yêu cầu:

Biết cách điều khiển động cơ bước chính xác về góc quay, chiều quay và tốc độ. Điều khiển start/stop và chiều quay bằng nút nhấn. Điều khiển tốc độ quay hiển thị trên led 7 đoạn.

II. Trình tự thực hiện:

1. Kết nối mạch theo trình tự :

- Dùng bus 4 dây kết nối port 2 với A,B,C,D của khối động cơ bước.
- Chú ý: P2.0 kết nối với chân A.

2. Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
;chương trình điều khiển động cơ bước quay thuận, nghịch liên tục half step
;dung bus 4 soi ket noi port 2 khoi dieu khien dong co buoc
;dung bus 2 soi ket noi p3.0 va p3.1 voi 2 nut nhan
;-----

        biendem    equ    r0           ;biên đếm
        chieuquay  bit    00h         ;chiều quay
        quaythuan  bit    p3.0        ;nút nhấn quay thuận
        quaynghich bit    p3.1        ;nút nhấn quay nghịch
        Org       0000h
        mov        dptr,#step_data
        mov        biendem,#0
repeat: jb        quaythuan,next
        setb       chieuquay
next:   jb        quaynghich,next2
        clr        chieuquay
next2: mov        a,biendem
        movc       a,@a+dptr
        mov        p2,a
        lcall      delay
        jnb        chieuquay,nglich
        inc        biendem
        cjne       biendem,#8,repeat

```

```

        mov    biendem,#0
        sjmp  repeat
nghich:dec    biendem
        cjne  biendem,#0ffh,repeat
        mov  biendem,#7
        sjmp  repeat
Delay: mov    r6,#20
Del:   mov    r7,#250
        Djnz  r7,$
        Djnz  r6,del
        Ret
Step_data: db 01h,03h,02h,06h,04h,0ch,08h,09h
        End
;-----
;chuong trinh dieu khien dong co buoc quay thuan, nghich lien tuc half step
;voi 2 nut dieu khien start/stop
;dung bus 4 soi ket noi port 2 khoi dieu khien dong co buoc
;dung bus 4 soi ket noi p3.0 den 3.3 voi 4 nut nhan
;-----
        biendem    equ    r0
        chieuquay  bit    00h
        quaythuan  bit    p3.0
        quaynghich bit    p3.1
        start      bit    p3.2
        stop       bit    p3.3
        Org        0000h
        mov    dptr,#step_data
        mov    biendem,#0
repeat: jb    stop,over
        jb    start,$
over:  jb    quaythuan,next
        setb  chieuquay
next:  jb    quaynghich,next2
        clr  chieuquay
next2: mov    a,biendem
        movc  a,@a+dptr

```

```

mov    p2,a
lcall  delay
jnb    chieuquay,ngich
inc    biendem
cjne   biendem,#8,repeat
mov    biendem,#0
sjmp   repeat
ngich:dec  biendem
        cjne   biendem,#0ffh,repeat
        mov    biendem,#7
        sjmp   repeat
Delay:  mov    r6,#20
Del:    mov    r7,#250
        Djnz   r7,$
        Djnz   r6,del
        Ret
Step_data:  db 01h,03h,02h,06h,04h,0ch,08h,09h
        End

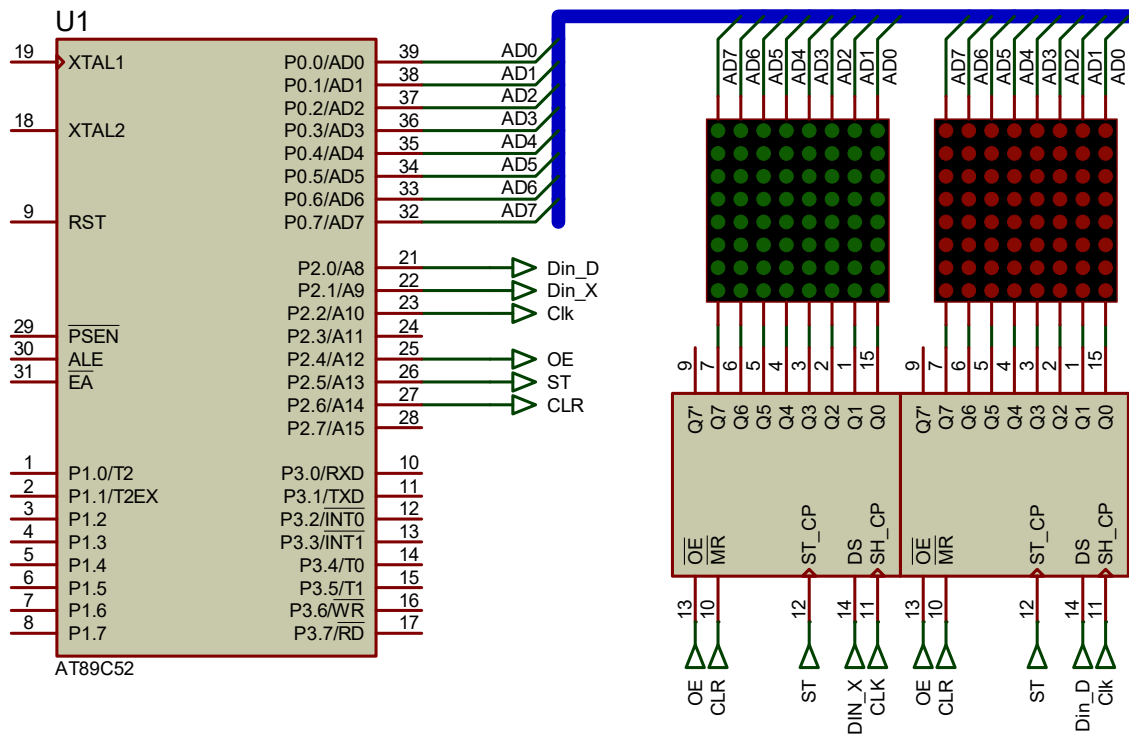
```

IV. Bài tập:

1. Viết chương trình điều khiển động cơ bước quay thuận 45^0 sau đó quay nghịch 90^0 và lặp lại.
2. Viết chương trình điều khiển động cơ bước quay thuận với tốc độ tăng dần sau mỗi vòng.
3. Viết chương trình điều khiển động cơ bước quay thuận nghịch với nút điều khiển start/stop và 2 nút điều khiển tốc độ hiển thị bằng led 7 đoạn.

PHẦN 4

ĐIỀU KHIỂN LED MA TRẬN



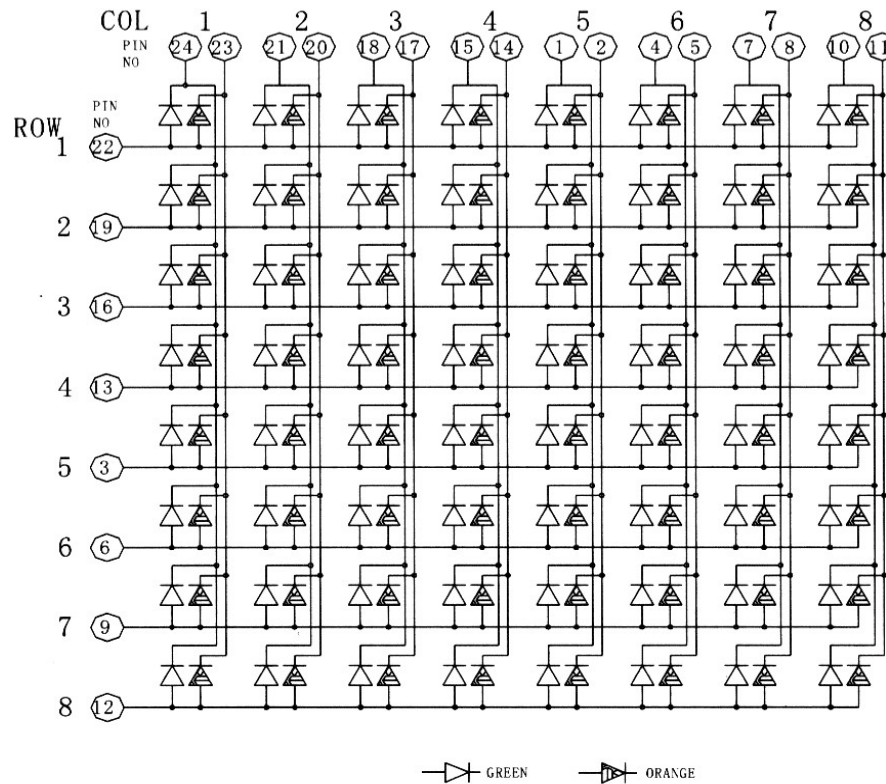
BÀI 4.1 CHƯƠNG TRÌNH HIỂN THỊ KÝ TỰ TRÊN LED MA TRẬN

I. Mục đích yêu cầu:

Nắm được nguyên tắc quét hiển thị trên led ma trận. Điều khiển hiển thị từng màu khác nhau trên led.

II. Trình tự thực hiện:

1. Sơ đồ nguyên lý led ma trận 8x8 2 màu:



2. Kết nối mạch theo trình tự :

- Dùng bus 8 dây kết nối port 0 với hàng của ma trận.
- Kết nối port 2 với connector điều khiển cột.

3. Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
;chương trình điều khiển led ma trận hiển thị chu A màu đỏ
;kết nối p0 với hàng của ma trận
;kết nối p2 cùng chiều với connects điều khiển cột (bên phải-phía dưới ma trận)
;phần cứng: quét led dùng ic ghi dịch 6b595
    
```

```
;ma chu xuất ra port 0
```

```
-----
    dind        bit        p2.0
    dinx        bit        p2.1
    clk         bit        p2.2
    oe          bit        p2.4
    st          bit        p2.5
    clr_        bit        p2.6
```

```
-----;chương trình chính
```

```
    org    0h
```

```
-----;khởi tạo 6b595
```

```
    clr    dind
    clr    dinx
    clr    clk
    setb   oe
    clr    st
    clr    clr_
    setb   clr_
```

```
-----
```

```
main: mov    dptr,#matranchua
```

```
    setb   dind
```

```
    mov    r2,#0h
```

```
rep:  mov    a,r2
```

```
    movc   a,@a+dptr
```

```
    mov    p0,a
```

```
    setb   clk
```

```
    clr    dind
```

```
    clr    clk
```

```
    setb   st
```

```
    clr    st
```

```
    clr    oe
```

```
    lcall  delay
```

```
    setb   oe
```

```
    inc    r2
```

```

    cjne  r2,#8,rep
    clr   clr_
    setb  clr_
    sjmp  main

;----- ;chương trình delay
delay:
    mov  r6,#5
del:   mov  r5,#250
       djnz r5,$
       djnz r6,del
       ret

;-----;dữ liệu của chữ A để hiển thị
matranchua:
    db   0f8h,24h,22h,21h,22h,24h,0f8h,00h
    end

;-----
;chương trình điều khiển led ma tran hiển thị chữ A màu đỏ + xanh
;ket noi p0 voi hang cua ma tran
;ket noi p2 cung chieu voi connects điều khiển cột
;ma chu xuất ra port 0
;-----
    dind      bit      p2.0
    dinx      bit      p2.1
    clk       bit      p2.2
    oe        bit      p2.4
    st        bit      p2.5
    clr_      bit      p2.6

;-----;bắt đầu chương trình chính
    org  0h

;-----;khởi tạo 6b595
    clr  dind
    clr  dinx
    clr  clk
    setb oe
    clr  st

```

```

    clr    clr_
    setb   clr_
;-----
main:  mov   dptr,#matranchua
       setb  dind
       setb  dinx
       mov   r2,#0h
rep:   mov   a,r2
       movc  a,@a+dptr
       mov   p0,a
       setb  clk
       clr   dind
       clr   dinx
       clr   clk

       setb  st
       clr   st

       clr   oe
       lcall delay
       setb  oe
       inc   r2
       cjne  r2,#8,rep
       clr   clr_
       setb  clr_
       sjmp  main
;-----;chương trình delay
delay: mov   r6,#5
del:   mov   r5,#250
       djnz  r5,$
       djnz  r6,del
       ret
;-----;dữ liệu ma trận chữ A
matranchua:
       db   0f8h,24h,22h,21h,22h,24h,0f8h,00h
       end

```


III. Chương trình mẫu:

Chương trình điều khiển điều khiển led ma trận chạy chữ A màu đỏ + xanh.

```

;-----
;ket noi p0 voi hang cua ma tran
;ket noi p2 cung chieu voi connects dieu khien cot
;-----
    dind    bit    p2.0
    dinx    bit    p2.1
    clk     bit    p2.2
    oe      bit    p2.4
    st      bit    p2.5
    clr_    bit    p2.6
    tocdo   equ    20h
    bd_cot  equ    21h
    goc     equ    40h
    dich    equ    4dh

;----- ;bắt đầu chương trình chính
    org     0000h

;-----;khởi tạo mã chữ trên Ram
    mov     40h,#00h
    mov     41h,#00h
    mov     42h,#00h
    mov     43h,#00h
    mov     44h,#00h
    mov     45h,#00h
    mov     46h,#00h
    mov     47h,#00h
    mov     48h,#0f8h
    mov     49h,#24h
    mov     4ah,#22h
    mov     4bh,#24h
    mov     4ch,#0f8h
    mov     4dh,#00h    ;dich
    mov     4eh,#00h
    mov     4fh,#00h

```

```

    mov    50h,#00h
    mov    51h,#00h
    mov    52h,#00h
    mov    53h,#00h
    mov    54h,#00h
;-----;khai tao 6b595
    clr    dind
    clr    dinx
    clr    clk

    setb   oe
    clr    st

    clr    clr_
    setb   clr_
;-----
main:
    mov    r1,#goc      ;khai tao goc
ht3:    mov    tocdo,#200 ;khai tao toc do
ht2:    mov    r0,01h    ;nhan goc
    mov    bd_cot,#8    ;khai tao bien dem so cot
    setb   dind         ;serial_data=1 do
    setb   dinx         ;serial_data=1 xanh

ht1:    mov    a,@r0
    mov    p0,a
;-----
    setb   clk
    clr    dind        ;chon cot do
    clr    dinx        ;chon cot xanh

    clr    clk
    setb   st
    clr    st
;-----
    clr    oe

```

```

        lcall  delay      ;hien thi cot
;-----
        setb  oe         ;chong lem
        mov   p0,#0h
;-----
        inc   r0         ;chon dia chi ke
        djnz  bd_cot,ht1 ;quet chua du 8 cot=> quet tiep
;-----
        clr   clr_       ;quet du => xoa thanh ghi dich
        setb  clr_
        djnz  tocdo,ht2  ;bien tocdo # 0 =>quet lai
;-----
        inc   r1         ;tang goc
        cjne  r1,#dich,ht3 ;goc=dich
        sjmp  main       ;ket thuc (lap lai tu dau)
;-----
delay:  mov   r6,#5
del:    mov   r5,#250
        djnz  r5,$
        djnz  r6,del
        ret
        end

```

IV. Bài tập:

1. Tìm hiểu phương pháp quét led ma trận và thanh ghi dịch .
2. Viết chương trình điều khiển led ma trận hiển thị các ký tự khác màu xanh.
3. Viết chương trình hiển thị hình trái tim đổi màu theo trình tự xanh, đỏ, xanh+đỏ.
4. Viết chương trình chạy chữ ‘DH BRVT’ bằng màu bất kỳ.

PHẦN 5

TIMER VÀ NGẮT

BÀI 5.1 CHƯƠNG TRÌNH DELAY SỬ DỤNG TIMER

I. Mục đích cầu:

Biết cách tính toán các thông số delay của timer để viết các chương trình delay chính xác.

II. Trình tự thực hiện:

1. Kết nối mạch theo trình tự :
 - Dùng bus 8 dây kết nối port 0 với một trong 3 PINHD của dây 24 led.
2. Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
; chương trình sang tắt port0 sử dụng timer làm bộ định thời delay 65536 micro giây
;-----
                org    0000h
                mov    tmod,#01        ; khởi tạo timer T0 mode 1 đếm 16 bit
                setb   tr0              ; cho phép timer 0 bắt đầu đếm xung
rep:            mov    p0,#00h
                lcall  delay            ; delay 65536 micro giây
                mov    p0,#0ffh
                lcall  delay
                jmp   rep
;-----
; chương trình con delay 65535 micro giây
;-----
delay:          clr    tf0              ; xóa cờ ngắt của timer 0
                mov    tl0,#0          ; nạp 0 vào TL0
                mov    th0,#0          ; nạp 0 vào TH0
                jnb   tf0,$            ; kiểm tra cờ tràn
                ret
                end

```

3. Thực hiện các bước giống như trên và xem kết quả.

Giải thích :

Bài sáng tắt port0 trên giống như bài đã làm trước đây chỉ khác là thay chương trình delay bằng một chương trình sử dụng timer để việc tính toán thời gian dễ dàng hơn.

Hàng lệnh đầu tiên trong chương trình chính là chọn mode làm việc cho timer T0 – hãy xem chương timer (timer T1 chưa sử dụng nên không cần quan tâm).

Với chương trình trên thì timer T0 sẽ đếm từ giá trị nạp ban đầu 0000H đến FFFFH và khi có thêm 1 xung nữa thì giá trị đếm sẽ là 10000H. Một cách đơn giản ta có thể xem con số 10000H được chia ra làm 2: số “1” được lưu trong cờ tràn TF0 và số “0000H” được lưu trong 2 thanh ghi TH0, TL0. Số xung đếm được là $10000H - 0000H = 10000$ (65536) xung và mỗi xung có chu kỳ 1 micro giây nên lượng thời gian mà timer T0 đếm được là 65536 micro giây.

Trong chương trình con có 2 lệnh nạp lại giá trị cho TH0 và TL0 của bài này là dư vì khi timer bị tràn thì nó tự động là cho 2 thanh ghi trên mang giá trị 0.

III. Chương trình mẫu:

Để đếm lượng thời gian nhỏ hơn ta hãy tham khảo bài mẫu dưới đây:

```

;-----
; chương trình sang tắt port0 sử dụng timer làm bỏ định thời delay 250 micro giây
;-----
        org     0000h
        mov     tmod,#01           ; khai tạo timer T0 mode 1 đếm 16 bit
        setb    tr0                ; cho phép timer 0 bắt đầu đếm xung
rep:     mov     p0,#00h
        lcall   delay              ; delay 250 micro giây
        mov     p0,#0ffh
        lcall   delay
        sjmp    rep
;-----
; chương trình con delay 250 micro giây
;-----
delay:   clr     tf0
        mov     tl0,#low(-250)     ; nạp 05 vào TL0
        mov     th0,#high(-250)   ; nạp FF vào TH0
del1:   jnb     tf0,del1
        ret
        end

```

Giải thích :

Lượng thời gian của bài này chỉ có 250 micro giây, vì giá trị ban đầu nạp cho timer là FF05H nên khi đếm đến 10000H kết quả xung đếm được $10000H - FF06H = 00FA$ (250) và mỗi xung có chu kỳ 1 micro giây nên lượng thời gian mà timer T0 đếm được là 250 micro giây.

Để khởi phải tính toán phức tạp ta có thể viết bằng hai lệnh như sau:

```
mov    tl0,#low(-250)    ; nạp 05 vào TL0
mov    th0,#high(-250)   ; nạp FF vào TH0
```

Khi biên dịch thì trình biên dịch tự động tính toán cho chúng ta.

Chú ý với các chương trình điều khiển led sáng với thời gian trễ nhỏ thì led sáng mờ nhưng không chớp tắt như các bài ta đã viết ở trên. Trong phần tính toán chúng ta chưa tính toán các lệnh trong chương trình con delay.

Muốn viết chương trình với các khoảng thời gian lớn hơn thì phải thêm thanh ghi ví dụ muốn viết delay 5 giây thì ta viết chương trình con delay $50.000 \mu s = 50ms$ và cho chúng thực hiện 100 lần – sau này chúng ta sẽ dùng timer để tạo ra các xung chính xác về thời gian cho các bài sau.

IV. Bài tập:

1. Hãy viết chương trình sáng tắt port 2 sử dụng timer làm bộ định thời delay 5 giây.
2. Hãy viết chương trình giống trên nhưng delay 10 giây.
3. Tương tự hãy viết chương trình delay 1 giờ.
4. Tìm hiểu nguyên tắc hoạt động của timer.
5. Lệnh nào cho phép timer chạy? Khi nào bit TF0 bằng 1?
6. Muốn khởi tạo timer0 ở mode 2, timer1 ở mode 1 ta thay đổi thanh ghi nào? Với giá trị bằng bao nhiêu?

BÀI 5.2 CHƯƠNG TRÌNH ĐẾM GIỜ SỬ DỤNG NGẮT CỦA TIMER

I. Mục đích yêu cầu:

Làm quen với cách viết chương trình đếm và sử dụng ngắt của timer để đếm chính xác về thời gian.

II. Trình tự thực hiện:

1. Kết nối mạch theo trình tự :

- Dùng bus 8 dây kết nối port 3 với led bên phải của nhóm 2 led 7 đoạn để hiển thị hàng đơn vị, port 1 với led bên trái để hiển thị hàng chục.

2. Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
;chuong trinh hien thi dem giay tu 00 den 59 tren 2 led 7 doan cua nhom 2 led.
;su dung ngat cua timer0 de dem chinh xac thoi gian
;-----

    biendem    equ    20h    ;dinh nghia cac bien
    giay_dvi   equ    21h
    giay_chuc  equ    22h

    org        0000h
    sjmp       start          ;nhay den chuong trinh chinh
    org        000bh          ;dia chi vecto ngat timer0
    ljmp       clock          ;nhay den chuong trinh phuc vu ngat timer0
start: mov     tmod,#01h      ;khoi tao timer0 mode 1
    mov       th0,#high(-50000)
    mov       tl0,#low(-50000)
    setb      et0             ;cho phep ngat timer0
    setb      ea              ;cho phep ngat toan bo
    setb      tr0             ;chay timer0

begin:
    mov       dptr,#ledcode   ;nap dia chi vung ma led vao dptr
    mov       a,giay_dvi      ;lay o nho chua giay don vi vao a
    movc     a,@a+dptr         ;lay ma led 7 doan tuong ung
    mov       p3,a             ;xuat ra hien thi don vi
    mov       a,giay_chuc     ;lay o nho giay chuc vao a

```



```

    movc    a,@a+dptr        ;lay ma led 7 doan tuong ung
    mov     p1,a             ;xuat ra hien thi hang chuc
    sjmp    begin

;-----
;chuong trinh con ngat timer0 de dem giay
;-----
clock:push    acc           ;cat thanh ghi a
    mov     tl0,#low(-50000) ;nap lai gia tri cho timer
    mov     th0,#high(-50000)
    inc     biendem         ;tang bien dem
    mov     a,biendem
    cjne   a,#20,finish
    mov     20h,#0
    inc     giay_dvi        ;tang giay dv
    mov     a,giay_dvi
    cjne   a,#0ah,finish
    mov     giay_dvi,#0
    inc     giay_chuc       ;tang giay chuc
    mov     a,giay_chuc
    cjne   a,#6,finish
    mov     giay_chuc,#0
finish:pop    acc
    reti

;-----
;khoi tao vung ma led 7 tren ROM
;-----
ledcode:
    db     0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
    end

```

Giải thích :

Để đếm giây chính xác ta sử dụng timer 0 ở mode1 hoạt động với 16 bit. Như vậy, nếu chu kỳ máy là 1us thì timer sẽ tràn sau $2^{16}\text{us} = 65536\text{us}$. Để đếm được 1 giây ta chọn tốc độ tràn của timer là 50000 và sử dụng một biến đếm để xác định số lần tràn. Với 20 lần timer tràn thì được 1 giây.

Hai biến `giay_chuc` và `giay_dvi` để đếm thời gian. Sau đó được giải mã sang mã led 7 đoạn và xuất ra hiển thị trên port 1 và port 3.

Tương tự, các bạn hãy viết chương trình đếm phút và giây trên 4 led 7 đoạn.

Để hiển thị đồng thời trên led, ta phải sử dụng phương pháp quét như ví dụ sau:

```

;-----
; chương trình hiển thị phút, giây trên 4 led 7 đoạn.
; port 0 xuất dữ liệu, port 2 xuất tín hiệu chọn led.
;-----
        biendem      equ    20h    ;định nghĩa các biến
        giay_dvi     equ    21h
        giay_chuc    equ    22h
        phut_dvi     equ    23h
        phut_chuc    equ    24h

        org          0000h
        sjmp        start          ;nhảy đến chương trình chính
        org          000bh          ;địa chỉ vectơ ngắt timer0
        ljmp        clock          ;nhảy đến chương trình phục vụ ngắt timer0
start:   mov         tmod,#01h      ;khởi tạo timer0 mode 1
        mov         th0,#high(-50000)
        mov         tl0,#low(-50000)
        setb        et0            ;cho phép ngắt timer0
        setb        ea            ;cho phép ngắt toàn bộ
        setb        tr0            ;chạy timer0
begin:   mov         r0,#40h        ;lấy địa chỉ đầu của ram hiển thị
        mov         r2,#11111110b  ;chọn led đầu tiên
rep:     mov         p2,r2
        mov         a,@r0
        mov         p0,a
        lcall        delay
        mov         p0,#0ffh
        inc         r0
        mov         a,r2
        rla
        mov         r2,a
        jb          acc.4,rep
        sjmp        begin
;-----
; chương trình còn giải mã giờ phút thành mã led 7 đoạn để hiển thị
;-----

```

```

decode: mov  dptr,#ledcode      ;nap dia chi vung ma led vao dptr
        mov  r0,#21h           ;nap dia chi cua vung du lieu gio phut giay
        mov  r1,#40h          ;nap dia chi vung ma hien thi
dec1:   mov  a,@r0
        movc a,@a+dptr
        mov  @r1,a
        inc  r0
        inc  r1
        cjne r0,#25h,dec1
        ret

```

```

;-----

```

```

;chuong trinh con ngat timer0 de dem gio phut giay

```

```

;-----

```

```

clock:  push  00h              ;cat thanh ghi r0
        push  acc              ;cat thanh ghi a
        mov  tl0,#low(-50000) ;nap lai gia tri cho timer
        mov  th0,#high(-50000)
        inc  biendem          ;tang bien dem
        mov  a,biendem
        cjne a,#20,finish
        mov  20h,#0
        inc  giay_dvi         ;tang giay dv
        mov  a,giay_dvi
        cjne a,#0ah,finish
        mov  giay_dvi,#0
        inc  giay_chuc        ;tang giay chuc
        mov  a,giay_chuc
        cjne a,#6,finish
        mov  giay_chuc,#0
        inc  phut_dvi         ;tang phut dv
        mov  a,phut_dvi
        cjne a,#0ah,finish
        mov  phut_dvi,#0
        inc  phut_chuc        ;tang phut chuc
        mov  a,phut_chuc
        cjne a,#6,finish

```

```

        mov     phut_chuc,#0
finish: lcall  decode           ;goi chuong trinh con giai ma hien thi
        pop     acc
        pop     00h
        reti
delay:mov    r6,#6
del:  mov    r7,#250
      djnz   r7,$
      djnz   r6,del
      ret

;-----
;khoi tao vung ma led 7 tren ROM
;-----
ledcode:
      db 0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
      end

```

III. Chương trình mẫu:

```

;-----
; chuong trinh hien thi gio, phut, giay tren 6 led 7 doan
; port 0 xuất dữ liệu, port 2 xuất tín hiệu chọn led.
;-----
      biendem     equ    20h    ;định nghĩa các biến
      giay_dvi     equ    21h
      giay_chuc    equ    22h
      phut_dvi     equ    23h
      phut_chuc    equ    24h
      gio_dvi      equ    25h
      gio_chuc     equ    26h
      org         0000h
      sjmp        start       ;nhay den chuong trinh chinh
      org         000bh       ;dia chi vecto ngat timer0
      ljmp        clock       ;nhay den chuong trinh phuc vu ngat timer0
start: mov        tmod,#01h    ;khoi tao timer0 mode1
      mov        th0,#high(-50000)
      mov        tl0,#low(-50000)

```

```

    setb    et0                ;cho phep ngat timer0
    setb    ea                ;cho phep ngat toan bo
    setb    tr0               ;chay timer0
begin: mov    r0,#40h        ;lay dia chi dau cua ram hien thi
    mov    r2,#11111110b    ;chon led dau tien
rep:  mov    p2,r2
    mov    a,@r0
    mov    p0,a
    lcall  delay
    mov    p0,#0ffh
    inc    r0
    mov    a,r2
    rl     a
    mov    r2,a
    jb     acc.6,rep
    sjmp   begin

;-----
;chuong trinh con giai ma gio phut giay thanh ma led 7 doan de hien thi
;-----
decode: mov    dptr,#ledcode    ;nap dia chi vung ma led vao dptr
    mov    r0,#21h            ;nap dia chi cua vung du lieu gio phut giay
    mov    r1,#40h           ;nap dia chi vung ma hien thi
decl: mov    a,@r0
    movc   a,@a+dptr
    mov    @r1,a
    inc    r0
    inc    r1
    cjne   r0,#27h,decl
    ret

;-----
;chuong trinh con ngat timer0 de dem gio phut giay
;-----
clock: push  00h                ;cat thanh ghi r0
    push  acc                  ;cat thanh ghi a
    mov    tl0,#low(-50000)    ;nap lai gia tri cho timer
    mov    th0,#high(-50000)

```

```

inc     biendem           ;tang bien dem
mov     a,biendem
cjne   a,#20,finish
mov     20h,#0
inc     giay_dvi          ;tang giay dv
mov     a,giay_dvi
cjne   a,#0ah,finish
mov     giay_dvi,#0
inc     giay_chuc        ;tang giay chuc
mov     a,giay_chuc
cjne   a,#6,finish
mov     giay_chuc,#0
inc     phut_dvi         ;tang phut dv
mov     a,phut_dvi
cjne   a,#0ah,finish
mov     phut_dvi,#0
inc     phut_chuc        ;tang phut chuc
mov     a,phut_chuc
cjne   a,#6,finish
mov     phut_chuc,#0
inc     gio_dvi          ;tang gio dv
mov     a,gio_chuc
cjne   a,#2,next
mov     a,gio_dvi
cjne   a,#4,finish
mov     gio_dvi,#0
mov     gio_chuc,#0
sjmp   finish
next:  mov     a,gio_dvi
       cjne   a,#0ah,finish
       mov     gio_dvi,#0
       inc     gio_chuc        ;tang gio chuc
finish: lcall  decode          ;goi chuong trinh con giai ma hien thi
       pop     acc
       pop     00h
       reti

```

```

delay:mov    r6,#6
del:  mov    r7,#250
      djnz   r7,$
      djnz   r6,del
      ret

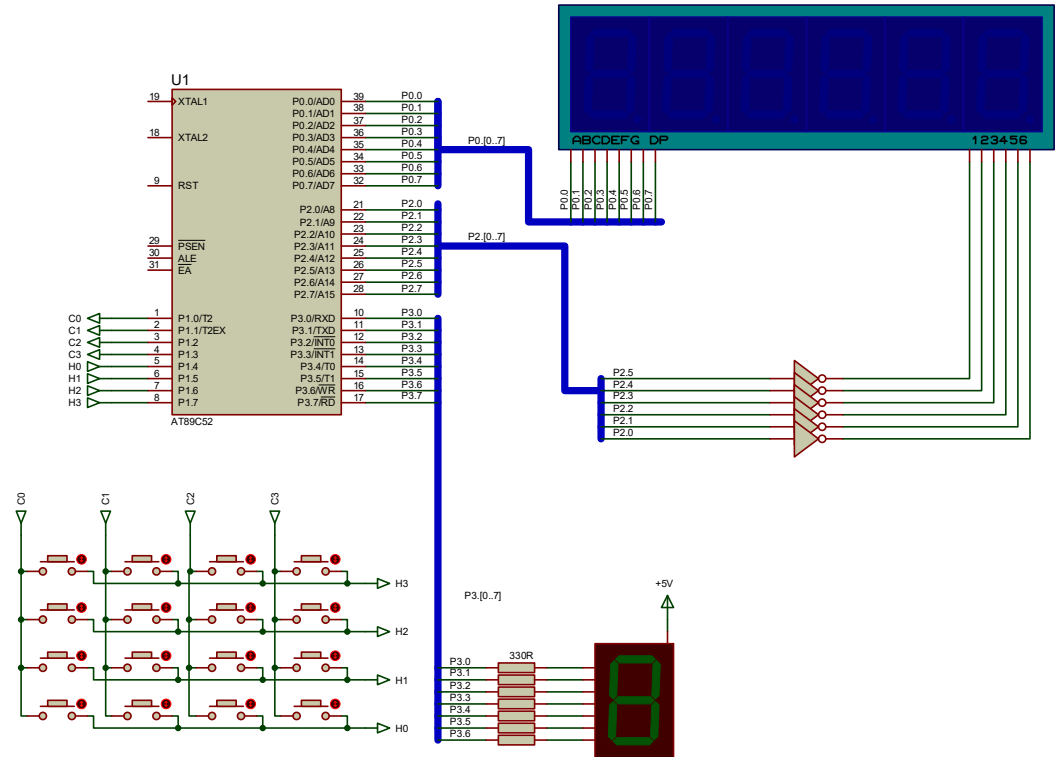
;-----
;khoi tao vung ma led 7 tren ROM
;-----
ledcode:
      db  0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
      end

```

IV. Bài tập:

1. Tìm hiệu hoạt động ngắt của timer.
2. Hãy viết chương trình đếm phút giây hiển thị trên led 7 đoạn số 1,2,5 và 6. Led 3,4 tắt.
3. Viết chương đồng hồ thể thao hiển thị phút, giây và phần trăm của giây hiển thị trên 6 led .
4. Tương tự bài trên, kết hợp với 2 nút nhấn kết nối với P1.0 và P1.1 để điều khiển Start/Stop và Reset.
5. Viết chương trình đồng hồ với các nút nhấn để chỉnh giờ, phút, giây.

BÀI 5.3 QUÉT PHÍM MA TRẬN VÀ HIỂN THỊ TRÊN LED 7 ĐOẠN



I. Mục đích yêu cầu:

Biết cách viết chương trình điều khiển quét phím, tạo mã phím và hiển thị mã phím. Khi nhấn một phím nào thì mã của phím đó hiển thị trên led.

II. Trình tự thực hiện:

1. Chức năng của phím hay nút nhấn hay contact:

Dùng để giao tiếp điều khiển giữa con người và thiết bị ví dụ như contact tắt mở bóng đèn và người sử dụng tác động đến contact để tắt mở thiết bị. Máy tính cộng trừ nhân chia thì chức năng của bàn phím là nhập các thông số vào máy và yêu cầu máy thực hiện các phép tính, tương tự bàn phím máy tính cũng vậy dùng để giao tiếp con người và máy.

Trong thực tế các thiết bị điều khiển lập trình đều có sử dụng nút nhấn, có thiết bị sử dụng ít nút nhấn như tivi, máy giặt, ..., có thiết bị sử dụng nhiều nút nhấn như bàn phím vi máy tính, điện thoại, bàn phím máy tính cộng trừ nhân chia,...

2. Nguyên lý:

Với nút nhấn thường hở thì khi ta nhấn thì sẽ ngắn mạch cho tín hiệu hay dòng điện chạy qua và khi không nhấn thì hở mạch sẽ ngắt tín hiệu hay ngắt dòng điện, còn nút nhấn thường

đóng thì ngược lại. Đối với tín hiệu số ta có thể xem “khi contact hở làm cho 1 ngõ vào ở mức logic 1 và khi contact nhấn làm thay đổi sang trạng thái mức logic 0 hoặc ngược lại”. Với cách kết nối kiểu như đã trình bày thì một contact phải sử dụng một đường tín hiệu giao tiếp. Nếu ứng dụng dùng vài chục phím thì theo cách này là không khả thi vì số lượng tín hiệu không đủ, nếu muốn thì phải dùng thêm IC giao tiếp.

Khi ứng dụng sử dụng nhiều phím thì ta nên kết nối theo dạng ma trận phím, với ma trận [m hàng, n cột] thì số phím bằng [n x m]. Ma trận 4 x 4 sẽ có 16 phím thì chỉ cần dùng 8 đường tín hiệu giao tiếp, ma trận 4 x 5 sẽ có 20 phím thì chỉ cần dùng 9 đường tín hiệu giao tiếp, ma trận 8 x 8 thì sẽ có 64 phím thì chỉ cần dùng 16 đường tín hiệu giao tiếp.

3. Giải thuật cho chương trình con quét phím dùng ma trận:

Với ma trận 4x4 thì phải có 4 ngõ vào và 4 ngõ ra, 4 ngõ vào thường đặt ở mức 1. 4 ngõ còn lại là 4 ngõ ra thường để xuất tín hiệu quét hay mã quét có 1 bit ở mức 0 các bit còn lại ở mức 1. Chương trình quét phím thường là chương trình con, chương trình chính sẽ gọi chương trình con quét phím để kiểm tra xem có phím nào bị nhấn hay không: nếu không có phím nào bị nhấn thì chương trình con quét phím kết thúc với 1 mã do người lập trình qui định, nếu có phím nhấn thì phải tiến hành thiết lập mã phím của phím đó (mỗi phím có 1 mã duy nhất) rồi tiến hành chống dội phím, sau đó có thể kết thúc quá trình chống dội bằng cách chờ buông phím hay hết thời gian qui định.

Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
; chương trình quét phím ma trận 4x4 và hiển thị mã phím trên led 7 đoạn
; dùng bus 8 sợi kết nối port 1 với ma trận phím. P1.0 kết nối với cột 0
;-----
        maphim      equ    20h    ; mã phím, sử dụng ở nhớ 20h lưu
        org         0000h
        mov         p3,#0bfeh    ; hiển thị dấu gạch ngang khi mỗi reset
begin:   mov         maphim,#0    ; mã phím nhấn = 0
        mov         p1,#0feh    ; chọn cột 0
rep:     mov         a,p1        ; lay p1 vào a
        anl         a,#0f0h     ; lay 4 bit cao
        cjne        a,#0f0h,next ; kiểm tra có phím nhấn không
        mov         a,p1        ; không có phím nhấn, lay p1 vào a
        rl          a           ; xoay a để chọn cột tiếp theo
        jnb         acc.4,begin  ; nếu qua cột thứ 4, lặp lại từ đầu
        mov         p1,a        ; nếu k, trả lại p1 để chọn cột tiếp theo
        mov         a,maphim    ; lay mã phím vào a
        add         a,#4        ; cộng 4 để qua mã phím của cột tiếp theo
        mov         maphim,a    ; tra dữ liệu cho mã phím
        sjmp        rep         ; nhảy trở lại kiểm tra phím cột tiếp theo
next:    rr          a
        jnb         acc.3,over

```

```

inc    maphim
sjmp  next
over:  mov    dptr,#ledcode      ;nap dia chi vung nho luu ma vao dptr
      mov    a,maphim
      movc  a,@a+dptr          ;lay ma cua so dem vao a
      mov    p3,a              ;xuat ma cua so dem ra hien thi
      sjmp  begin

;-----
;khoi tao vung ma led 7 doan
;-----
ledcode:
      db    0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
      db    88h,83h,0c6h,0a1h,86h,8eh
      end

```

III. Chương trình mẫu:

Chương trình quét phím ma trận và hiển thị mã của phím nhấn trên 6 led 7 đoạn dịch chuyển từ phải sang trái

```

;-----
;ket noi port 1 voi ma tran phim, P1.0 voi cot 0
;ket noi port 0 voi ngo vao du lieu led 7 doan, port 2 voi ngo vao chon led
;-----
      Ram_hien_thi    equ    30h
      Maphim          equ    20h
      org    0000h
      sjmp  start      ;nhay den chuong trinh chinh
      org    000bh      ;dia chi vecto ngat timer0
      ljmp  quet_phim   ;nhay den chuong trinh con quet phim
start:  mov    tmod,#01h ;khoi tao timer0 mode1
      mov    th0,#high(-5000)
      mov    tl0,#low(-5000)
      setb  et0          ;cho phep ngat timer0
      setb  ea           ;cho phep ngat toan bo
      setb  tr0          ;chay timer0
      mov    35h,#0f7h   ;hien thi dau gach ngang khi moi khoi dong
      mov    34h,#0f7h
      mov    33h,#0f7h

```

```

    mov    32h,#0f7h
    mov    31h,#0f7h
    mov    30h,#0f7h
quet_hien_thi:
    mov    r0,#ram_hien_thi    ;lay dia chi dau cua ram hien thi
    mov    r2,#11111110b      ;chon led dau tien
qht:  mov    p2,r2
    mov    a,@r0
    mov    p0,a
    lcall  delay
    mov    p0,#0ffh
    inc    r0
    mov    a,r2
    rl     a
    mov    r2,a
    jb     acc.6,qht
    sjmp   quet_hien_thi

;-----
;chuong trinh con quet phim
;-----
quet_phim:
    push  acc                ;cat thanh ghi a
    push  00h                ;cat thanh ghi r0
    mov   maphim,#0         ;ma phim nhan = 0
    mov   p1,#0feh          ;chon cot 0
rep:   mov   a,p1            ;lay p1 vao a
    anl   a,#0f0h           ;lay 4 bit cao
    cjne  a,#0f0h,next      ;kiem tra co phim nhan khong
    mov   a,p1              ;khong co phim nhan, lay p1 vao a
    rl    a                  ;xoay a de chon cot tiep theo
    jnb   acc.4,exit         ;neu qua cot thu 4, thoat
    mov   p1,a              ;neu k, tra lai p1 de chon cot tiep theo
    mov   a,maphim          ;lay ma phim vao a
    add   a,#4              ;cong 4 de qua ma phim cua cot tiep theo
    mov   maphim,a          ;tra du lieu cho ma phim
    sjmp  rep               ;nhay tro lai kiem tra phim cot tiep theo

```

```

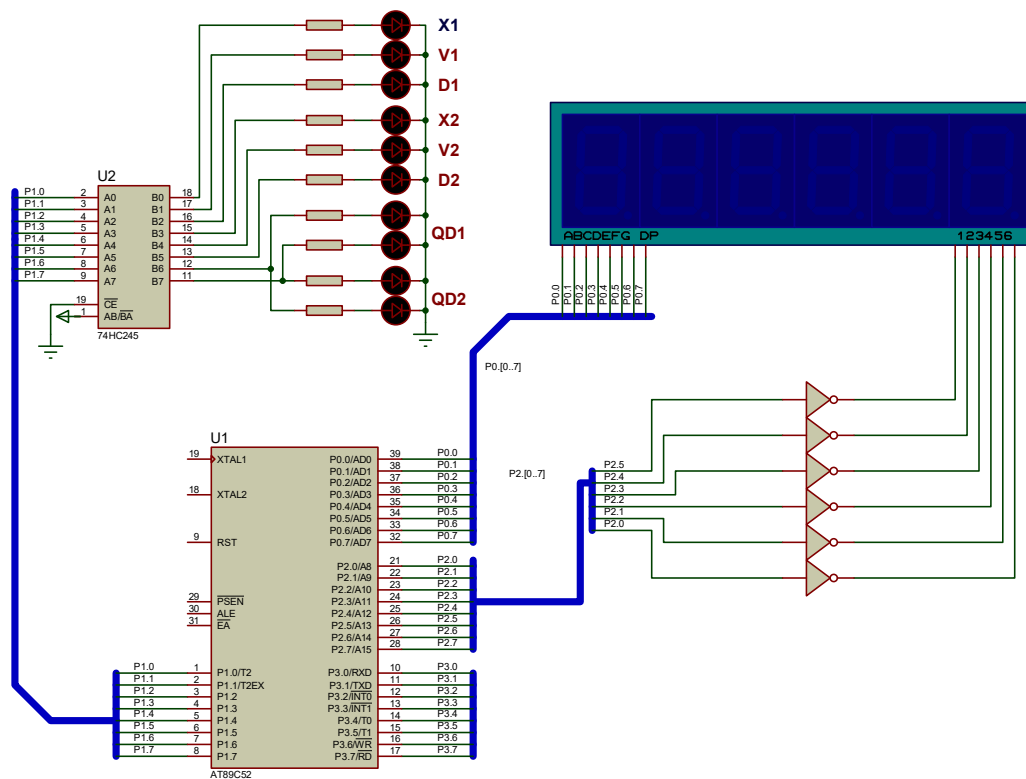
next:  rr    a
       jnb  acc.3,wait
       inc  maphim           ;tang ma phim
       sjmp next
wait:  mov  p1,#0f0h
       mov  a,p1             ;lay p1 vao a
       cjne a,#0f0h,wait    ;kiem tra phim con nhan khong
       mov  dptr,#ledcode   ;nap dia chi vung nho luu ma vao dptr
       mov  a,maphim
       movc a,@a+dptr       ;lay ma cua so dem vao a
       mov  35h,34h         ;dich du lieu qua trai
       mov  34h,33h
       mov  33h,32h
       mov  32h,31h
       mov  31h,30h
       mov  30h,a
exit:  pop  00h
       pop  acc
       reti
delay: mov  r6,#6
del:   mov  r7,#250
       djnz r7,$
       djnz r6,del
       ret
ledcode:
       db   0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
       db   88h,83h,0c6h,0a1h,86h,8eh
       end

```

IV. Bài tập:

1. Tìm hiểu nguyên tắc quét phím.
2. Tại sao ở bài mẫu, khi nhấn 1 phím thì led tắt. Hãy khắc phục tình trạng trên.
3. Kết hợp với các bài tập trước để viết chương trình đếm thời gian giờ, phút, giây và được hiệu chỉnh giờ bằng các phím nhấn ma trận.

BÀI 5.4 ĐIỀU KHIỂN ĐÈN GIAO THÔNG



I. Mục đích yêu cầu:

Biết cách viết chương trình điều khiển đèn giao thông, tính toán thời gian cho các đèn và hiển thị thời gian đếm ngược trên led 7 đoạn.

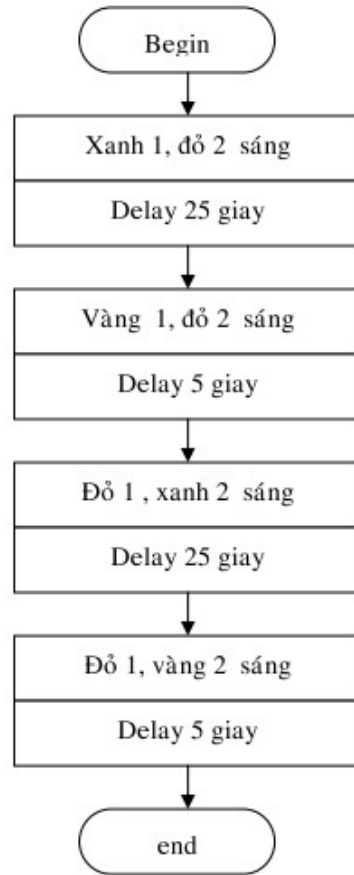
II. Trình tự thực hiện:

1. Yêu cầu:

Viết chương trình điều khiển đèn giao thông cho 1 ngã tư gồm Xanh1, Vàng1, Đỏ1, Xanh2, Vàng2, Đỏ2. Cho thời gian xanh sáng 25 giây, thời gian vàng sáng 5 giây, thời gian đỏ bằng thời gian xanh cộng thời gian vàng. Các đèn xanh vàng đỏ dùng led đơn để hiển thị và được điều khiển bằng port 1. Led sáng khi bit điều khiển bằng 1.

Đèn	Bit điều khiển	Thời gian
Xanh 1	P1.0	25s
Vàng 1	P1.1	5s
Đỏ 1	P1.2	
Xanh 2	P1.3	25s
Vàng 2	P1.4	5s
Đỏ 2	P1.5	

2. Giải thuật:



Khởi động phần mềm, tạo File mới và biên soạn chương trình sau:

```

;-----
;chương trình điều khiển đèn giao thông hiển thị bằng led đơn – cách 1
;dung bus 8 soi kết nối port 1 với dây led mô phỏng đèn giao thông
;-----

```

```

    Org    0000h
Begin: Mov  p1,#21h    ;xanh 1 đỏ 2
        Mov  r4,#25    ;nap thời gian delay vào biến đếm
Rep1:  Lcall delay1s
        Djnz r4,rep1
        Mov  p1,#22h    ;vang 1 đỏ 2
        Mov  r4,#5     ;nap thời gian delay vào biến đếm
Rep2:  Lcall delay1s
        Djnz r4,rep2
        Mov  p1,#0ch    ;xanh 2 đỏ 1
        Mov  r4,#25    ;nap thời gian delay vào biến đếm
Rep3:  Lcall delay1s

```

```

Djnz r4,rep3
Mov p1,#14h ;vang 2 đò 1
Mov r4,#5 ;nap thoi gian delay vao bien dem
Rep4: Lcall delay1s
Djnz r4,rep4
Sjmp begin
Delay1s: ;chuong trinh con delay1s
Mov r5,#8
Del1: mov r6,#250
Del2: mov r7,#250
Djnz r7,$
Djnz r6,del2
Djnz r5,del1
Ret
End

```

3. Thực hiện các bước giống như trên và xem kết quả.

Ở chương trình trên, ta thấy có 4 đoạn chương trình gần như nhau. Chỉ khác dữ liệu xuất ra port 1 và thời gian delay nạp vào biến đếm. Ta có thể rút gọn chương trình bằng vòng lặp và viết lại theo cách 2 như sau:

```

;-----
;chuong trinh dieu khien den giao thong hien thi bang led don – cach 2
;dung bus 8 soi ket noi port 1 voi dây led mo phong den giao thong
;-----
Org 0000h
Mov 20h,#21h ;xanh 1 đò 2
Mov 21h,#25 ;thoi gian delay
Mov 22h,#22h ;vang 1 đò 2
Mov 23h,#5
Mov 24h,#0ch ;xanh 2 đò 1
Mov 25h,#25
Mov 26h,#14h ;vang 2 đò 1
Mov 27h,#5
Begin:
Mov r0,#20h
Repeat: Mov a,@r0
Mov p1,a ;xuât ra điều khiển đèn

```

```

    Inc    r0
    Mov    a,@r0      ;lấy thời gian delay
    Mov    r4,a       ; nạp vào biến đếm
Rep:  Lcall delay1s
      Djnz r4,rep
      Inc  r0
      Cjne r0,#28h,repeat
      Sjmp begin
Delay1s:                ;chương trình con delay1s
      Mov  r5,#8
Del1:  mov  r6,#250
Del2:  mov  r7,#250
      Djnz r7,$
      Djnz r6,del2
      Djnz r5,del1
      Ret
End

```

III. Chương trình mẫu:

Chương trình điều khiển đèn giao thông kết hợp với led 7 đoạn để hiển thị thời gian đếm ngược trên led 1,2 và 5,6 của nhóm 6 led. Led 3,4 tắt.

```

;-----
;chương trình đèn giao thông có hiển thị thời gian đếm ngược
;kết nối port 1 với dãy led mô phỏng đèn giao thông
;kết nối port 0 với ngõ vào dữ liệu led 7 đoạn, port 2 với ngõ vào chọn led
;-----
    biendem    equ    20h    ;định nghĩa các biến
    bientrangthai equ    21h    ;trạng thái hiện tại của hệ thống
    thoigian1   equ    22h    ;thời gian hiển thị ở hướng 1
    thoigian2   equ    23h    ;thời gian hiển thị ở hướng 2
    xanh1       equ    15
    vang1       equ    5
    xanh2       equ    10
    vang2       equ    3

    org    0000h
    sjmp   start                ;nhảy đến chương trình chính

```



```

    org    000bh           ;dia chi vecto ngat timer0
    jmp    dengiaothong    ;nhay den chuong trinh phuc vu ngat timer0
;-----;khoi tao timer 0 dem thoi gian
start:  mov    tmod,#01h    ;khoi tao timer0 model
        mov    th0,#high(-50000)
        mov    tl0,#low(-50000)
        setb   et0         ;cho phep ngat timer0
        setb   ea         ;cho phep ngat toan bo
        setb   tr0        ;chay timer0
;-----;khoi tao cac gia tri ban dau
        mov    biendem,#0
        mov    bientrangthai,#0
        mov    thoigian1,#1
;-----;chuong trinh chinh thuc hien quet hien thi
begin:  mov    r0,#45h     ;lay dia chi cuoi cua ram hien thi
        mov    r2,#1111110b ;chon led dau tien
rep:    mov    p2,r2
        mov    a,@r0
        mov    p0,a
        lcall  delay
        mov    p0,#0ffh
        dec   r0
        mov   a,r2
        rl    a
        mov   r2,a
        jb    acc.6,rep
        sjmp  begin
;-----;chuong trinh ngat dieu khien den giao thong
dengiaothong:
        push  00h
        push  acc
        mov   tl0,#low(-50000)
        mov   th0,#high(-50000)
        inc   biendem      ;tang bien dem
        mov   a,biendem
        cjne  a,#20,finish

```

```

        mov    biendem,#0
        dec   thoigian1
        dec   thoigian2
        mov   a,thoigian1
        cjne  a,#0,dgt1
        sjmp  dgt2
dgt1:  mov   a,thoigian2
        cjne  a,#0,giaima
dgt2:  inc   bientrangthai
        mov   a,bientrangthai
        cjne  a,#1,dgt3
        mov   p1,#00100001b
        mov   thoigian1,#xanh1
        mov   thoigian2,#xanh1+vang1
        sjmp  giaima
dgt3:  cjne  a,#2,dgt4
        mov   p1,#00100010b
        mov   thoigian1,#vang1
        sjmp  giaima
dgt4:  cjne  a,#3,dgt5
        mov   p1,#00001100b
        mov   thoigian1,#xanh2+vang2
        mov   thoigian2,#xanh2
        sjmp  giaima

dgt5:  mov   p1,#00010100b
        mov   thoigian2,#vang2
        mov   bientrangthai,#0

giaima:mov  dptr,#ledcode
        mov   r0,#40h
        mov   a,thoigian1
        mov   b,#10
        div   ab
        movc  a,@a+dptr
        mov   @r0,a

```

```

inc    r0
mov    a,b
movc   a,@a+dptr
mov    @r0,a
inc    r0
mov    @r0,#0ffh
inc    r0
mov    @r0,#0ffh
inc    r0
mov    a,thoigian2
mov    b,#10
div    ab
movc   a,@a+dptr
mov    @r0,a
inc    r0
mov    a,b
movc   a,@a+dptr
mov    @r0,a
finish: pop    acc
        pop    00h
        reti
delay:  mov    r6,#6
del:    mov    r7,#250
        djnz   r7,$
        djnz   r6,del
        ret

ledcode: db    0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f8h,80h,90h
        end

```

IV. Bài tập:

1. Tìm hiểu nguyên tắc hoạt động của đèn giao thông.
2. Thay đổi thời gian khác nhau giữa hai hướng của giao lộ.
3. Từ bài mẫu, kết hợp với phím nhấn để chỉnh thời gian cho từng hướng của giao lộ.

PHẦN 6

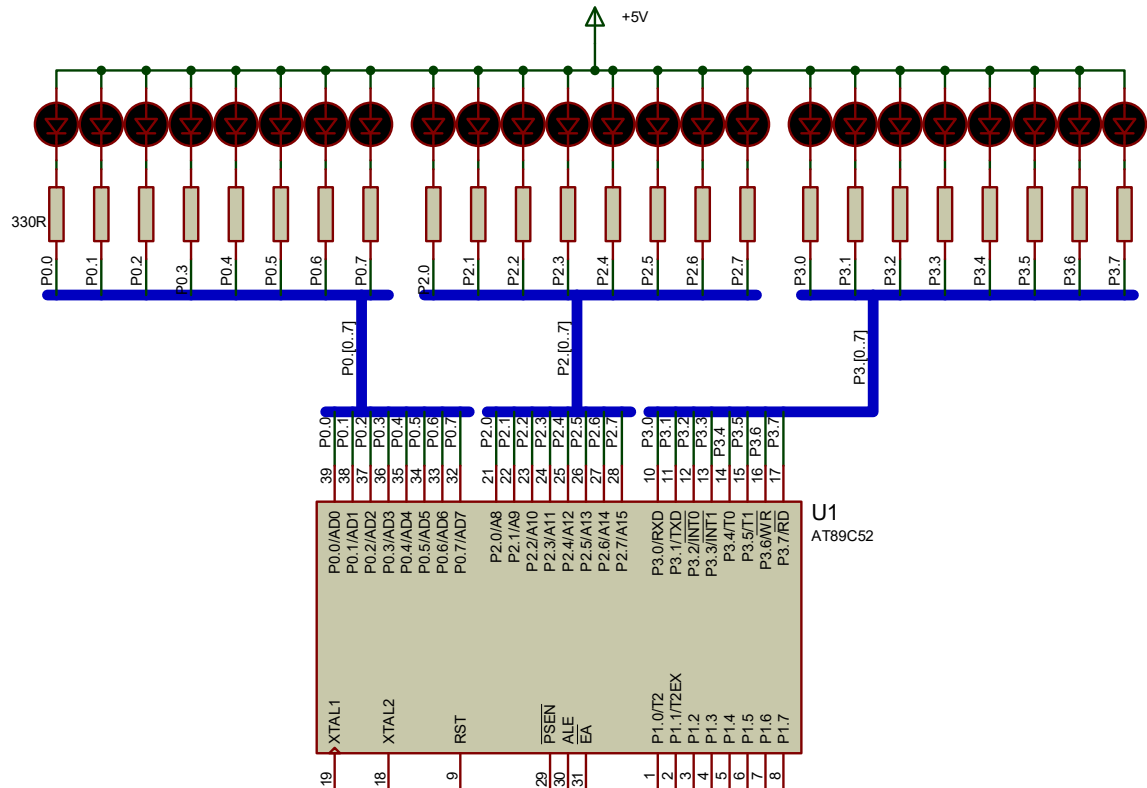
THI CÔNG MẠCH ỨNG DỤNG

BÀI 6.1 MẠCH ĐIỀU KHIỂN LED ĐƠN

I. Mục đích yêu cầu:

- Ứng dụng được vi điều khiển để điều khiển 16 led đơn theo các hiệu ứng đã học.
- Mở rộng sáng tạo thêm các hiệu ứng theo tư duy cá nhân.
- Thiết kế và thi công thành mạch ứng dụng hoàn chỉnh.

II. Sơ đồ nguyên lý:



III. Danh mục linh kiện vật tư cần:

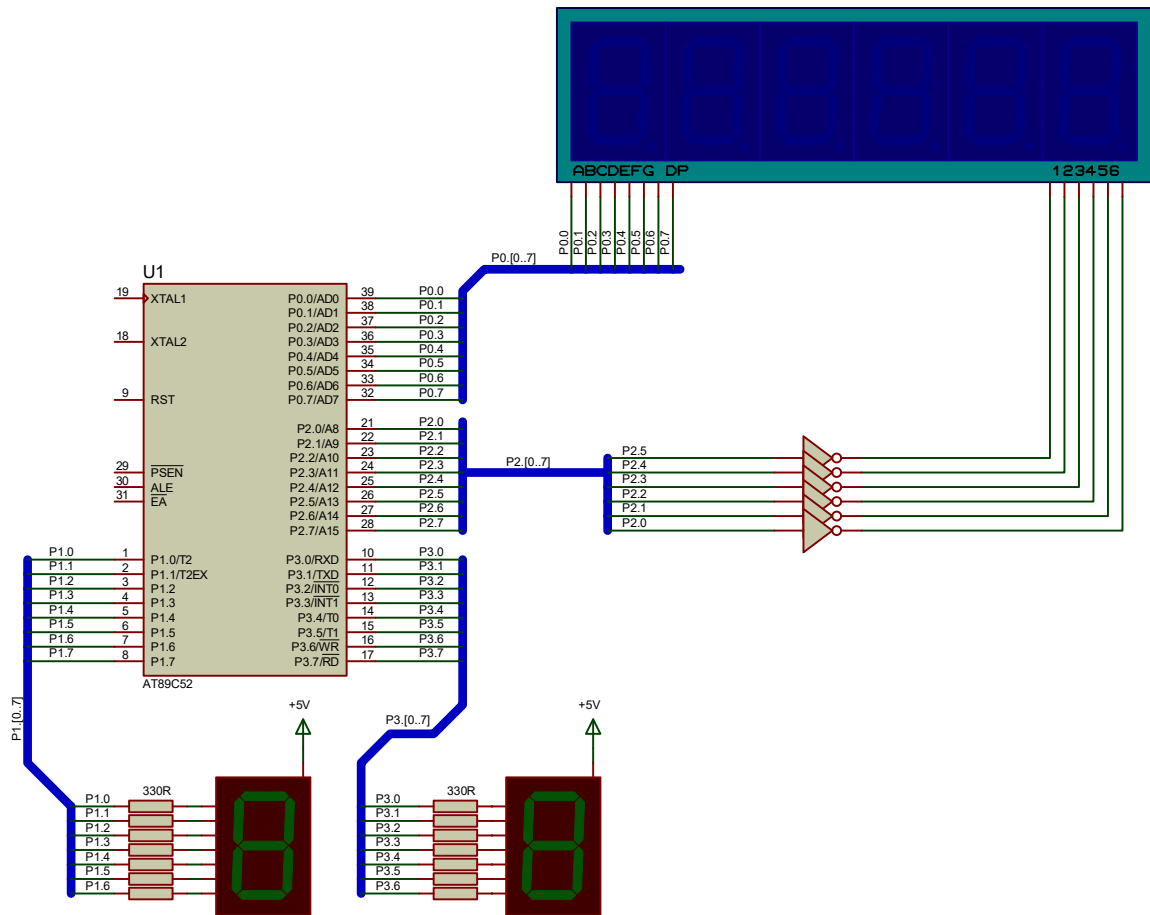
- Mạch in hoặc testboard hàn 6 x 10 cm: 1 miếng
- Chip 89S52: 1 con
- Thạch anh 12MHz: 1 con
- Tụ điện 33p: 2 con
- Tụ điện 104: 1 con
- Dây bus 8: 1 sợi
- Pinhead: 0.5 cây
- Chì hàn: 0.5 m
- Điện trở 330: 16 con
- Led 5 ly: 16 cái

BÀI 6.2 MẠCH ĐIỀU KHIỂN LED 7 ĐOẠN

I. Mục đích yêu cầu:

- Ứng dụng được vi điều khiển để điều khiển 2 led 7 đoạn theo các bài đã học.
- Mở rộng sáng tạo thêm các ứng dụng khác theo tư duy cá nhân.
- Thiết kế và thi công thành mạch ứng dụng hoàn chỉnh.

II. Sơ đồ nguyên lý:



III. Danh mục linh kiện vật tư cần:

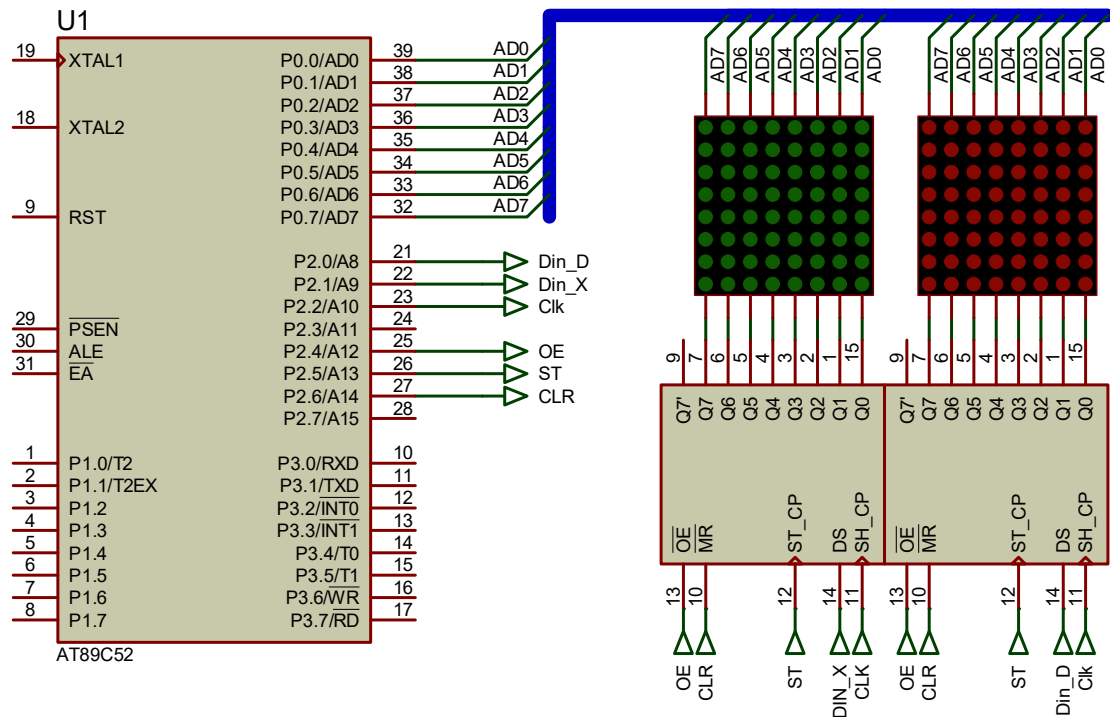
- Mạch in hoặc testboard hàn 6 x 10 cm: 1 miếng
- Chip 89S52: 1 con
- Thạch anh 12MHz: 1 con
- Tụ điện 33p: 2 con
- Tụ điện 104: 1 con
- Dây bus 8: 1 sợi
- Pinhead: 0.5 cây
- Chì hàn: 0.5 m
- Điện trở 330: 16 con
- Led 7 đoạn AC: 2 cái

BÀI 6.3 MẠCH ĐIỀU KHIỂN LED MA TRẬN

I. Mục đích yêu cầu:

- Ứng dụng được vi điều khiển để điều khiển 1 led ma trận 8x8 theo các bài đã học.
- Mở rộng sáng tạo thêm các ứng dụng khác theo tư duy cá nhân.
- Thiết kế và thi công thành mạch ứng dụng hoàn chỉnh.

II. Sơ đồ nguyên lý:



III. Danh mục linh kiện vật tư cần:

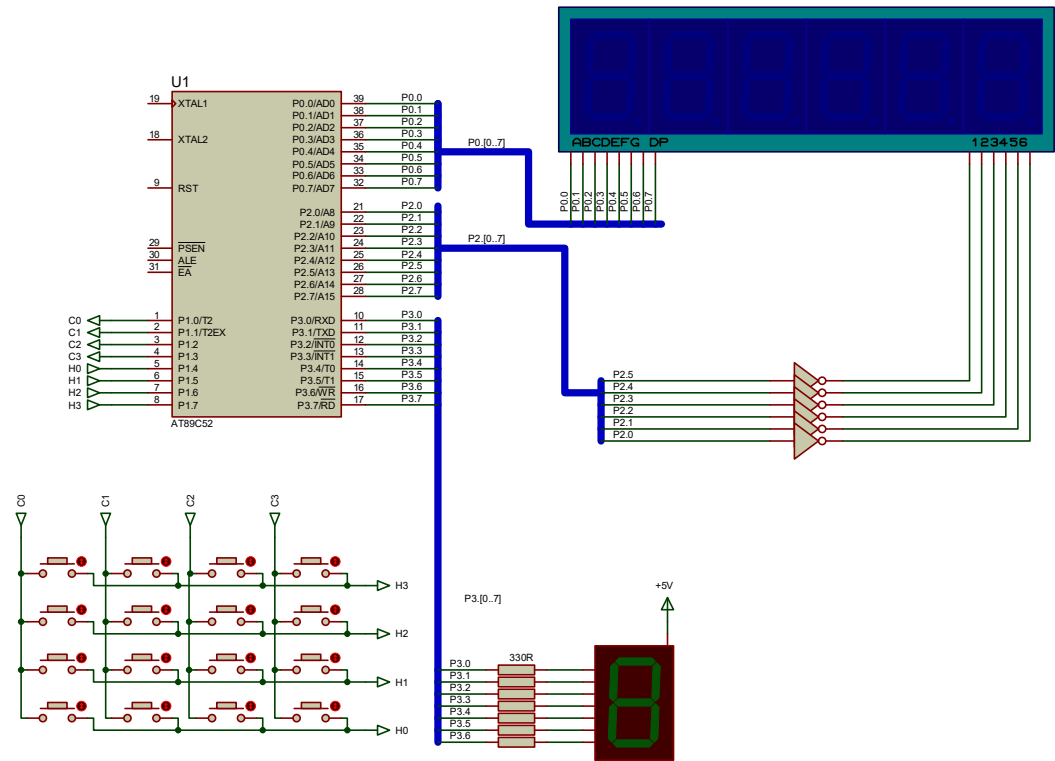
- Mạch in hoặc testboard hàn 6 x 10 cm: 1 miếng
- Chip 89S52: 1 con
- Thạch anh 12MHz: 1 con
- Tụ điện 33p: 2 con
- Tụ điện 104: 1 con
- Dây bus 8: 1 sợi
- Pinhead: 0.5 cây
- Chi hàn: 0.5 m
- Điện trở 150: 16 con
- Led ma trận 8x8 2 màu: 1 cái
- IC 5B595: 2 con
- IC TD 62783: 2 con

BÀI 6.4 MẠCH QUÉT PHÍM MA TRẬN HIỂN THỊ TRÊN LED 7 ĐOẠN

I. Mục đích yêu cầu:

- Ứng dụng được vi điều khiển để điều khiển quét phím ma trận hiển thị trên 1 led 7 đoạn theo các bài đã học.
- Mở rộng sáng tạo thêm các ứng dụng khác theo tư duy cá nhân.
- Thiết kế và thi công thành mạch ứng dụng hoàn chỉnh.

II. Sơ đồ nguyên lý:



III. Danh mục linh kiện vật tư cần:

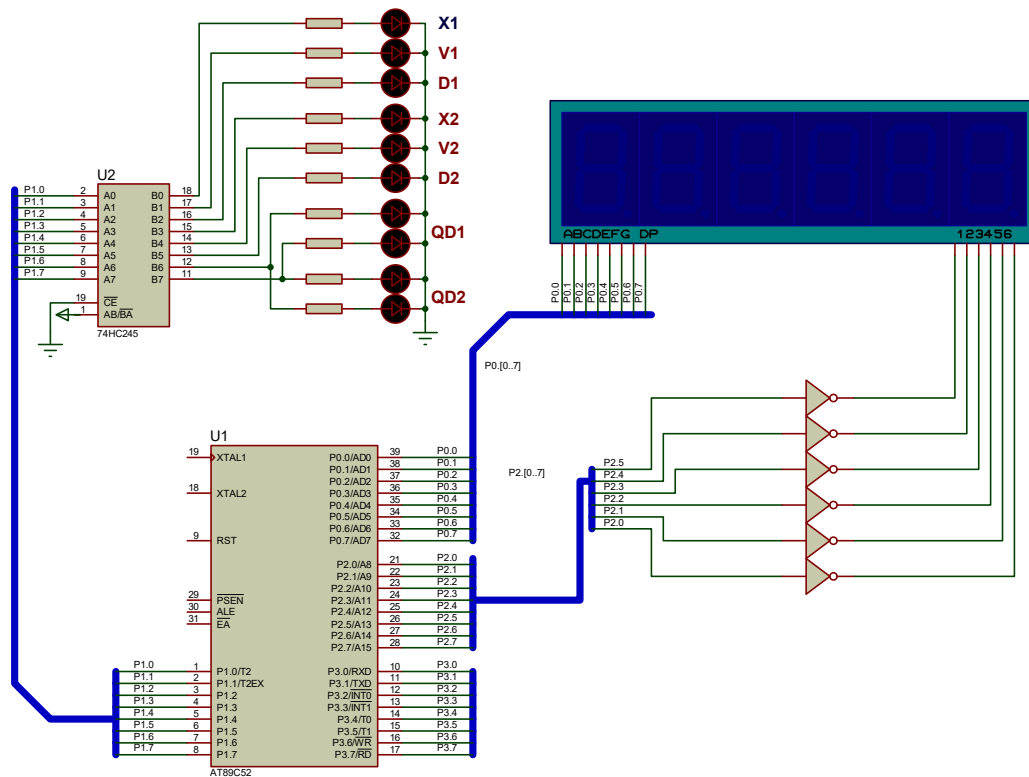
- Mạch in hoặc testboard hàn 6 x 10 cm: 1 miếng
- Chip 89S52: 1 con
- Thạch anh 12MHz: 1 con
- Tụ điện 33p: 2 con
- Tụ điện 104: 1 con
- Dây bus 8: 1 sợi
- Pinhead: 0.5 cây
- Chi hàn: 0.5 m
- Điện trở 330: 8 con
- Led 7 đoạn AC: 1 cái
- Nút nhấn 5x5x6: 8 cái

BÀI 6.5 MẠCH ĐIỀU KHIỂN ĐÈN GIAO THÔNG

I. Mục đích yêu cầu:

- Ứng dụng được vi điều khiển để điều khiển đèn giao thông có số đếm ngược theo các bài đã học.
- Mở rộng sáng tạo thêm các ứng dụng khác theo tư duy cá nhân.
- Thiết kế và thi công thành mạch ứng dụng hoàn chỉnh.

II. Sơ đồ nguyên lý:



III. Danh mục linh kiện vật tư cần:

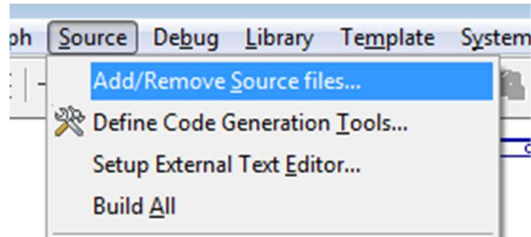
- Mạch in hoặc testboard hàn 6 x 10 cm: 1 miếng
- Chip 89S52: 1 con
- Thạch anh 12MHz: 1 con
- Tụ điện 33p: 2 con
- Tụ điện 104: 1 con
- Dây bus 8: 1 sợi
- Pinhead: 0.5 cây
- Chì hàn: 0.5 m
- Điện trở 330: 22 con
- Led 7 đoạn AC: 2 cái
- Led đơn 5 mm: 6 cái

HƯỚNG DẪN MÔ PHỎNG BẰNG PROTEUS

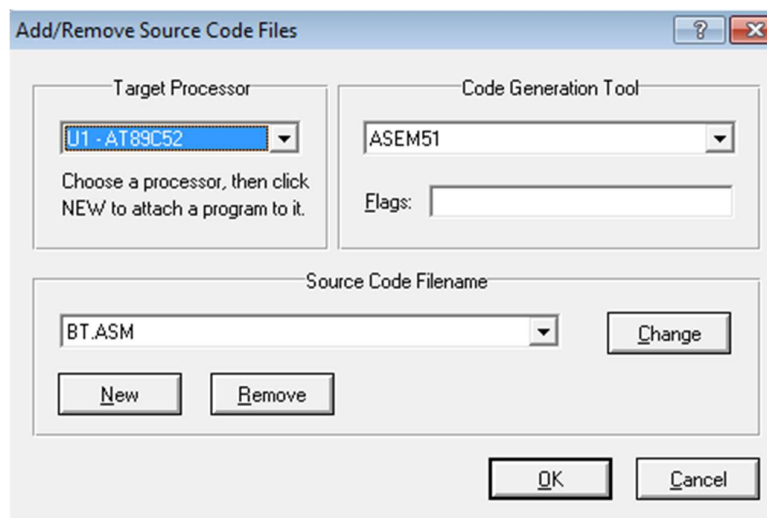
1. Tạo file ASM mới

1.1. Mở file mô phỏng trong thư mục TN VI XU LY

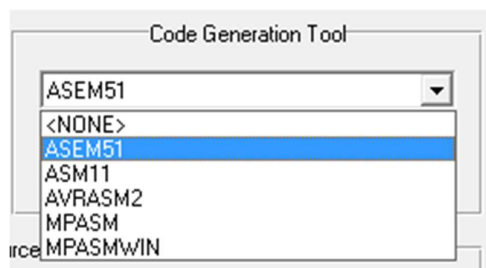
1.2. Vào **Menu > Source > Add/Remove Source files...**



1.3. Trong mục **Source Code Filename** nhấn **Remove** để xóa những source code cũ.



1.4. Trong mục **Code Generation Tool** chọn **ASEM51** là công cụ để biên dịch từ file .ASM sang file .HEX



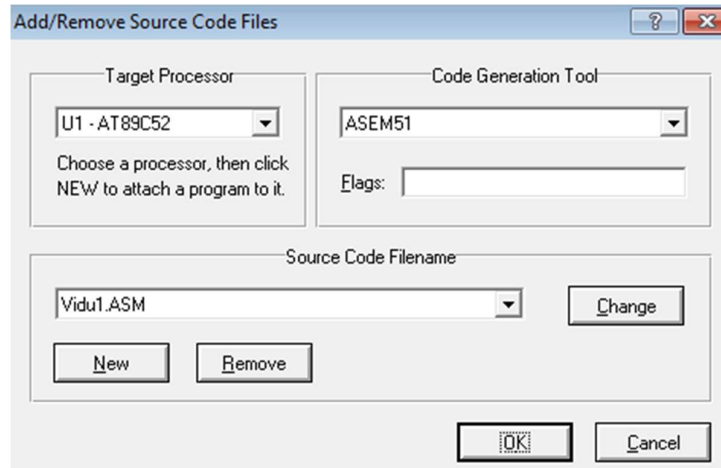
1.5. Trong mục **Source Code Filename** nhấn **Open** để mở file .ASM có sẵn hoặc tạo file mới.



1.6. Đầu tiên, các bạn tạo file mới tên **Vidu1** và lưu trong thư mục của nhóm. Thư mục này các bạn đặt trong ổ đĩa D, tên thư mục là tên lớp và tên nhóm.

Ví dụ tên thư mục: D:\DH14DD_NHOM1.

1.7. Nhấn **OK** để thoát ra.



2. Soạn thảo chương trình

2.1. Vào **Menu > Source** chọn file **Vidu1.ASM** cửa sổ soạn thảo chương trình sẽ mở ra.

2.2. Nhập vào đoạn chương trình như sau:

```

org    0000h                ;bat dau chuong trinh
rep:   mov    p0,#00000000b
       lcall  delay
       mov    p0,#11111111b
       lcall  delay
       sjmp  rep
delay: mov    r6,#0ffh
del:   mov    r7,#0ffh
       djnz  r7,$
       djnz  r6,del
       ret
end                ;ket thuc chuong trinh

```

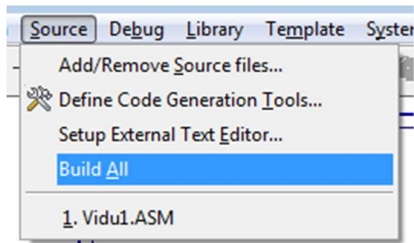
2.3. Chương trình được viết thành 4 cột, mỗi cột cách nhau bởi khoảng cách **Tab**.

- Cột thứ 1 để ghi các nhãn. Ví dụ : rep, delay, del, ...
- Cột 2 để ghi lệnh. Ví dụ: mov, lcall, sjmp, ...
- Cột 3 để ghi toán hạng. Ví dụ: p0,#00000000b ...
- Cột 4 để ghi chú nếu cần. Tất cả các phần ghi sau dấu chấm phẩy được xem là ghi chú và trình biên dịch sẽ bỏ qua.

2.4. Sau khi viết chương trình xong, bấm **Ctrl + S** hoặc nhấn nút **SAVE** để lưu lại.

3. Biên dịch

3.1. Vào **Menu > Build All** để biên dịch chương trình sang file .HEX



3.2. Khi biên dịch thành công sẽ xuất hiện thông báo sau:

```

BUILD LOG
Message
Building Vidu1.ASM
ASEM51 Vidu1.ASM /INCLUDES:"C:\Program Files\Lak
MCS-51 Family Macro Assembler ASEM-51 V1.3
no errors
Extracting debug data from Vidu1.LST...
Processed 54 lines.
Source code build completed OK.

```

3.3. Nếu chương trình bị lỗi sẽ xuất hiện thông báo ở những dòng bị lỗi, cần sửa và biên dịch lại.

```

BUILD LOG
Message
Building Vidu1.ASM
ASEM51 Vidu1.ASM /INCLUDES:"C:\Program Files\Labcenter Electronics\
MCS-51 Family Macro Assembler ASEM-51 V1.3
Vidu1.ASM(8): illegal statement syntax
Vidu1.ASM(10): symbol not defined
2 errors detected
ASEM.EXE reported error code (1L).
Extracting debug data from Vidu1.LST...
8:          del      mov      r7,#0ffh
          ^
          @@@@ illegal statement syntax @@@@
10: 0012 DE 00          djnz     r6,del
          ^
          @@@@ symbol not defined @@@@
Processed 60 lines.
Source code build FAILED with 1 error(s).

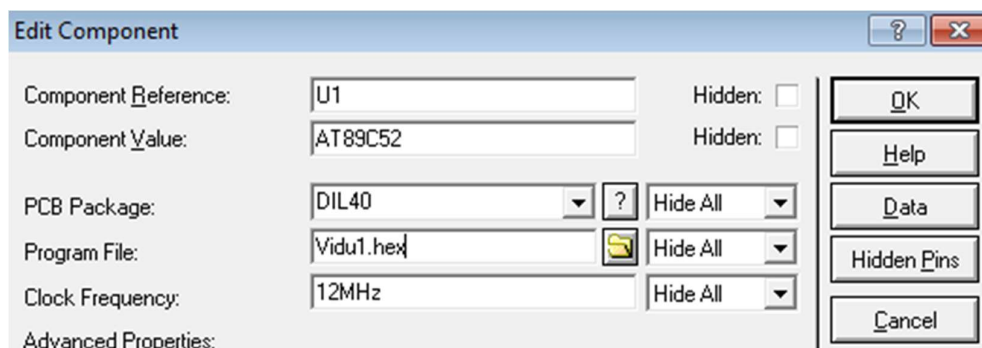
```

3.4. Sau khi biên dịch thành công sẽ xuất hiện file Vidu1.HEX trong cùng thư mục với file Vidu1.ASM. File này dùng để nạp xuống vi điều khiển hoặc dùng để mô phỏng.

4. Mô phỏng

4.1. Nhấp đúp vào ic 89C52 trong cửa sổ của Proteus, cửa sổ **Edit Component** xuất hiện.

4.2. Trong mục **Program File** mở file **Vidu1.HEX** để mô phỏng.



4.3. Mục **Clock Frequency** đặt **12MHz**.

4.4. Nhấn **OK** để thoát ra.

4.5. Nhấn nút **PLAY** để chạy mô phỏng.



4.6. Có thể mô phỏng từng bước bằng cách nhấn vào nút **STEP**. Cửa sổ mô phỏng theo step xuất hiện.

```

8051 CPU Source Code - U1
Vidu1.SDI
---- org      0000h                ;bat dau chuong trinh
0000 rep:     mov      p0,#00000000b
0003         lcall   delay
0006         mov     p0,#11111111b
0009         lcall   delay
000C sjmp    rep
000E delay:   mov     r6,#0ffh
0010 del:    mov     r7,#0ffh
0012         djnz   r7,$
0014         djnz   r6,del
0016         ret
000E         end                ;ket thuc chuong trinh
    
```

4.7. Có thể kết hợp với cửa sổ **CPU Register** và **CPU SFR Memory** trong Menu **Debug** để có thể quan sát trạng thái từng thanh ghi hoặc ô nhớ trong bộ nhớ khi chương trình chạy.

8051 CPU Registers - U1

PC	INSTRUCTION
0000	MOV P0,#00
ACC B DPTR SP CA-rs0-P	
00 00 0000 07 00000000	
R0 R1 R2 R3 R4 R5 R6 R7	
00 00 00 00 00 00 00 00	
P0 P1 P2 P3 SCON SBUF	
FF FF FF FF 00 00	
TMR0 TMR1 TMOD TCON PCON	
0000 0000 00 00 00	
IE IP TMR2 TCON RCAP	
00 00 0000 00 0000	

8051 CPU SFR Memory - U1

80	FF 07 00 00	00 00 00 00
88	00 00 00 00	00 00 02 00
90	FF 00 00 00	00 00 00 00
98	00 00 00 00	00 00 00 00
A0	FF 00 00 00	00 00 00 00
A8	00 00 00 00	00 00 00 00
B0	FF 00 00 00	00 00 00 00
B8	00 00 00 00	00 00 00 00
C0	00 00 00 00	00 00 00 00
C8	00 00 00 00	00 00 00 00
D0	00 00 00 00	00 00 00 00
D8	00 00 00 00	00 00 00 00
E0	00 00 00 00	00 00 00 00
E8	00 00 00 00	00 00 00 00
F0	00 00 00 00	00 00 00 00
F8	00 00 00 00	00 00 00 00

4.8. Nút **STOP** để dừng mô phỏng.

HƯỚNG DẪN NẠP CHƯƠNG TRÌNH CHO VI ĐIỀU KHIỂN

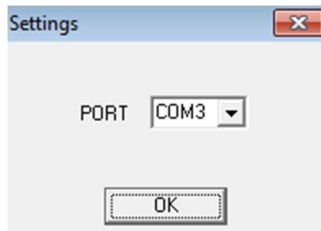
Có nhiều phần mềm nạp cho Vi điều khiển họ 8051 ứng với từng mạch nạp khác nhau. Ở đây chúng ta sử dụng 2 loại mạch nạp sau:

1. Mạch nạp 89S ISP PRO



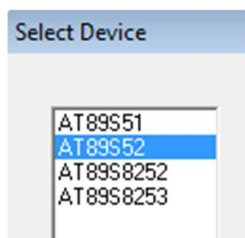
1.1 Mở phần mềm nạp **ISP_PROG v1.4** trên desktop.

1.2 Vào Menu **Option > Setting**. Trong cửa sổ **Setting** chọn PORT đang kết nối mạch nạp. Nhấn **OK** để thoát. (Phần này chỉ cài đặt lần đầu hoặc khi cắm mạch nạp qua cổng USB khác).

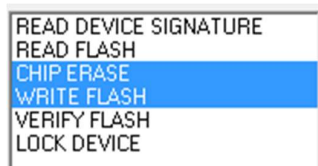


Chú ý: khi không kết nối được cần tắt chương trình nạp, rút dây usb rồi cắm lại và chạy lại chương trình.

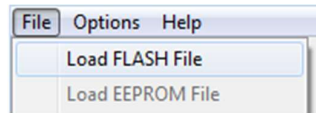
1.3 Vào Menu **Option > Devive**. Trong cửa sổ **Select Device** chọn **AT89S51** hoặc **AT89S51** theo chip lắp đặt trên Kit. Nhấn **OK** để thoát. (Phần này chỉ cài đặt lần đầu hoặc khi đổi chip loại khác).



1.4 Chọn mục **CHIP ERASE** và **WRITE FLASH** như hình.



1.5 Vào Menu **File** > **Load Flash** để chọn file HEX cần nạp.



1.6 Sau khi đã load file HEX, chỉ cần nhấn **RUN** để nạp xuống vi điều khiển. Chú ý là nguồn của Kit phải bật và đã kết nối cáp giữa mạch nạp và Kit. Sau khi nạp xong, chương trình sẽ được chạy ngay trên Kit.

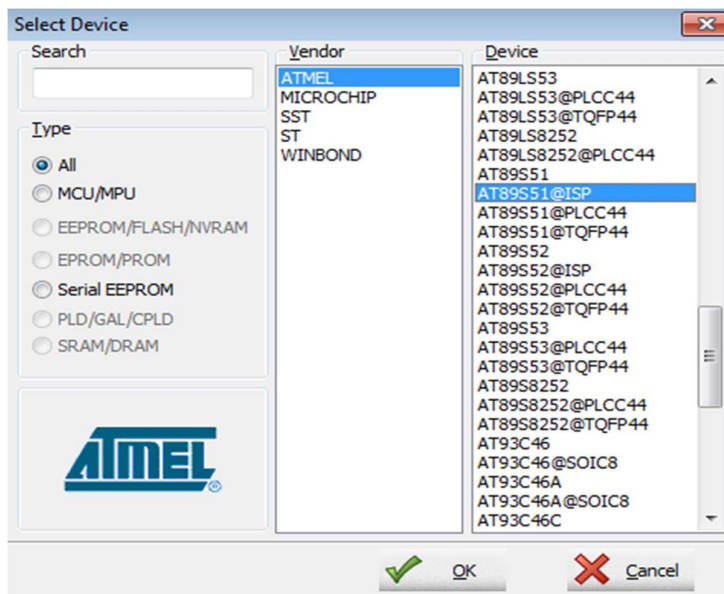
2. Mạch nạp SP200S Enhanced Programmer



2.1 Mở phần mềm nạp **WLPRO V2.20** trên desktop.

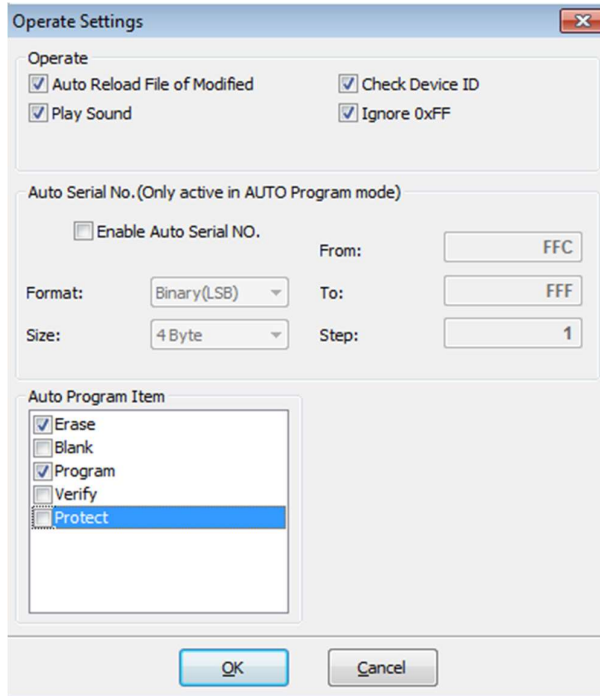
2.2 Vào Menu **Device** > **Select Device** để mở cửa sổ **Select Device**.


2.3 Chọn chip **AT89S51@ISP** hoặc **AT89S52@ISP** tùy theo chip đang được lắp trên kit. Sau đó nhấn **OK** để thoát (Phần này chỉ cài đặt lần đầu hoặc khi đổi chip loại khác).



2.4 Vào Menu **Operate** > **Setting** để mở cửa sổ **Operate Settings**.

2.5 Trong mục **Auto Program Item** chỉ check vào **Erase** và **Program** sau đó **OK** để thoát ra (Phần này chỉ cài đặt lần đầu hoặc khi đổi chip loại khác).



2.6 Vào Menu **File > Load File** hoặc **Ctrl+L** hoặc nhấn vào  trên thanh công cụ để mở file HEX cần nạp.

```
Device Model: ATMEL AT89S51@ISP
Load File: D:\LED DON\Vidu1.hex
File Format: IntelHEX
Load File Successful
```

2.7 Sau khi đã load file HEX, chỉ cần nhấn  **Auto** để nạp xuống vi điều khiển. Chú ý là nguồn của Kit phải bật và đã kết nối cáp giữa mạch nạp và Kit. Sau khi nạp xong, chương trình sẽ được chạy ngay trên Kit.