

## Chương VI

# PHỤ LỤC

### 6.1 Hướng dẫn sử dụng KIT

Người học phải nắm vững bản đồ bộ nhớ của KIT sau:

Địa chỉ	Thiết bị	Ghi chú
0000h - 1FFFh	<b>EEPROM</b>	Chứa chương trình Monitor
4000h - 7FFFh	<b>RAM</b>	Chứa chương trình ứng dụng
8000h - 8003h	<b>8255 (1)</b>	<b>8000h</b> - PortA : Điều khiển LCD. <b>8001h</b> - PortB : Data bus của LCD. <b>8002h</b> - PortC : Nhận mã của bàn phím. <b>8003h</b> - CW : Thanh ghi điều khiển.
8004h - 8007h	<b>8255 (5)</b>	<b>8004h</b> - PortA: Điều khiển cấp nguồn cho các Anot chung của LED matrix (các hàng). <b>8005h</b> - PortB: Điều khiển đèn màu xanh của LED matrix (các cột xanh). <b>8006h</b> - PortC: Điều khiển đèn màu đỏ của LED matrix (các cột đỏ). <b>8007h</b> - CW: Thanh ghi điều khiển.
8008h - 800Bh	<b>8255 (2)</b>	<b>8008h</b> - PortA : Đường Data của ADC0809. <b>8009h</b> - PortB : Đường Data của DAC0808. <b>800Ah</b> - PortC : Điều khiển ADC0809. <b>800Bh</b> - CW : Thanh ghi điều khiển.
800Ch - 800Fh	<b>8255 (3)</b>	<b>800Ch</b> - PortA : Đầu vào Digital. <b>800Dh</b> - PortB : Điều khiển động cơ bước( <b>PB0-PB3</b> ) và động cơ một chiều ( <b>PB4 - PB5</b> ). <b>800Eh</b> - PortC : 4 đầu vào xung dạng nút ấn ( <b>PC4-PC7</b> ), 4 đầu vào cho các thiết bị ngoài như encoder ( <b>PC0 - PC3</b> ). <b>800Fh</b> - CW : Thanh ghi điều khiển.
C000h - C003h	<b>8255 (4)</b>	<b>C000h</b> - PortA : Hiển thị 8 LED đơn. <b>C001h</b> - PortB : Hiển thị 2 LED 7 thanh (trái). <b>C002h</b> - PortC : Hiển thị 2 LED 7 thanh (phải). <b>C003h</b> - CW : Thanh ghi điều khiển.
C004h – C007h	<b>8255-EX</b>	BUS mở rộng dự trữ, có thể gắn thêm 1 module khác

**Bảng 2.1** Sắp xếp bộ nhớ và cách phân địa chỉ cho các thiết bị ngoại vi của KIT

Sau khi đã nắm vững bản đồ bộ nhớ, một số lưu ý cần phải nhớ khi thí nghiệm với KIT như sau:

- Nếu nạp chương trình vào KIT để chạy ở chế độ "Chương trình người sử dụng", tức chạy độc lập với chương trình MONITOR quản lý KIT thì chương trình của người học phải bắt đầu ở địa chỉ 00H, tức bắt đầu chương trình bằng lệnh ORG 0. Sau khi KIT báo nạp hoàn tất thì ấn công tắc MODE SELECT để ở trạng thái hở (nhô lên), tiếp đó bấm nút reset để bắt đầu chạy chương trình của mình. ở trạng thái này thì tất cả các tài nguyên phần cứng và các thiết bị ngoại vi của KIT đều được phép sử dụng mà không có một giới hạn nào.
- Nếu nạp vào để chạy ở chế độ DEBUG - gỡ rối, chạy và xem kết quả các vùng nhớ sau mỗi lệnh thì:
  - o Chương trình phải bắt đầu ở vị trí 4000H (vị trí khởi đầu của bộ nhớ RAM). Thực hiện bằng lệnh ORG 4000H.
  - o Các vector ngắt phải được cộng thêm 4000H. Cụ thể như sau:

<i>Nguồn ngắt</i>	<i>Cờ</i>	<i>Địa chỉ vector ngắt ở chế độ bình thường</i>	<i>Địa chỉ vector ngắt ở chế độ DEBUG</i>
Reset hệ thống	RST	0000H	4000H
Ngắt ngoài 0	IE0	0003H	4003H
Bộ định thời 0	TF0	000BH	400BH
Ngắt ngoài 1	IE1	0013H	4013H
Bộ định thời 1	TF1	001BH	401BH
Port nối tiếp	RI or TI	0023H	4023H
Bộ định thời 2	TF2 or EXF2	002BH	402BH

- Ngoài ra người học còn không được sử dụng ngắt ngoài 1 đã được thiết kế để phục vụ chế độ DEBUG. Còn đối với ngắt ngoài 0 như ở chương 2 đã giới thiệu, phục vụ giao tiếp với bàn phím của KIT.
- Vùng nhớ RAM nội 32 byte từ 7FH-5FH không sử dụng, tuy nhiên 128 byte RAM nội được định địa chỉ gián tiếp còn lại và 32 K-Byte RAM ngoài được sử dụng thoải mái.

Các hướng dẫn cụ thể bằng hình ảnh của KIT sẽ được trình bày ở phần trợ giúp của chương trình chạy trên môi trường Windows đi kèm theo KIT, chương trình hỗ trợ biên dịch và dịch ngược các file chương trình dạng hợp ngữ, nạp và xóa bộ nhớ EEPROM của KIT và một số bài ví dụ của chương trình như bài 7, thí nghiệm về "Truyền thông nối tiếp"...

Sau đây sẽ là các khai báo mẫu của các chương trình chạy được trên KIT cũng như chương trình MONITOR và các ví dụ.

## 6.2 Các khai báo về địa chỉ các hằng và biến số sử dụng trong chương trình MONITOR và trong các ví dụ DEMO của KIT

```
; #####
;                                     CAC KHAI BAO
; #####

; Khai bao dia chi cua cac 8255
; 8255-1
LCD_CONTROL      EQU    8000H ; PA: Cac tin hieu dieu khien cua LCD 2408
                                ; PA0 = LCD_RS; PA1 = RW; PA2=E1; PA3 = E2
LCD_DATA          EQU    8001H ; PB: BUS data cua LCD 2408
KEYPAD            EQU    8002H ; D0-D6 lam bus nhan ma phim
CW1               EQU    8003H ; Tu dieu khien CW cua 8255-1
; 8255-5
MATRIX_LED_ANOD   EQU    8004H ; Cac hang`, tuc cac anod chung cua led ma tran
MATRIX_LED_G      EQU    8005H ; Cac cot xanh cua LED ma tran
MATRIX_LED_R      EQU    8006H ; Cac cot do cua LED ma tran
CW5               EQU    8007H ; Tu dieu khien CW cua 8255-5

; 8255-2
ADC_DB            EQU    8008H ; Bus du lieu cua ADC0809
```

```

DAC_DB      EQU      8009H  ; Bus du lieu cua DAC0808
ADC_CONTROL EQU      800AH  ; Cac tin hieu dieu khien ADC0809
CW2          EQU      800BH  ; Tu dieu khien CW cua 8255-2

; 8255-3
DIGITAL_INPUT EQU      800CH  ; Dau vao so
DC_STEPPER   EQU      800DH  ; Dieu khien dong co buoc (PB0-PB3),
                                ; dong co 1 chieu (PB4-PB7)
PULSE_INPUT  EQU      800EH  ; 4 dau vao xung = nut (PC4-PC7),
                                ; 4 tu cac nguon phat xung (PC0-PC3)
CW3          EQU      800FH  ; Tu dieu khien CW cua 8255-3

; 8255-4
LED_DON      EQU      0C000H ; Dieu khien hien thi 8 LED don
LED_7_DOAN_12 EQU      0C001H ; Dieu khien hien thi 2 LED 7 doan 1, 2 (ben trai')
LED_7_DOAN_34 EQU      0C002H ; Dieu khien hien thi 2 LED 7 doan 3, 4 (ben phai')
CW4          EQU      0C003H ; Tu dieu khien CW cua 8255-4

; Quy uoc cach dat ten hang` cua tu dieu khien nhu sau: CWx voi x la STT cua 8255
; 011 thu tu cac so la: PA, PB, PC; 0: dau ra, 1: dau vao
CW_011      EQU      10001011B
CW_000      EQU      10000000B
CW_100      EQU      10010000B
CW_101      EQU      10011001B

; ROM - RAM
EEPROM_ADDR EQU      0000H  ; Dia chi bat dau cua EEPROM - 8K - ket thuc o 1FFFF
RAM_ADDR    EQU      4000H  ; Dia chi bat dau cua RAM - 32K - ket thuc o 7FFFF

; Dia chi cac dong` cua LCD 2408
DONG1       EQU      0080H
DONG2       EQU      0098H
DONG3       EQU      00B8H
DONG4       EQU      00D8H

; Chan ngat ngoai o che do debug
CHAN_NGAT_INT0 EQU      P3.2

; Ma cac phim chuc nang
KEY_F1      EQU      16
KEY_F2      EQU      17
KEY_BS      EQU      18

```

```

KEY_ENTER      EQU      19

; RAM dành cho xy ly cua chuong trinh MONITOR
ProgRamAddr     EQU      7FH
E                EQU      ProgRamAddr    ; De chon lam viec voi nua man hinh 1 hay 2,
                                         tuong ung E =1, 2
KEY             EQU      ProgRamAddr - 1  ; Byte dem. cua ban phim, luu gia tri phim
                                         nhan ve, cap nhat bang ngat

So_Man_Hinh     EQU      ProgRamAddr - 2
LED7Doan1       EQU      ProgRamAddr - 3
LED7Doan2       EQU      ProgRamAddr - 4
LED7Doan3       EQU      ProgRamAddr - 5
LED7Doan4       EQU      ProgRamAddr - 6

; RAM lam vung dem cho qua trinh nap EEPROM tu PC
StartAddrH      EQU      ProgRamAddr - 8; Byte cao cua dia chi can ghi den' cua goi du lieu
StartAddrL      EQU      ProgRamAddr - 9; Byte thap cua dia chi can ghi den' cua goi du lieu

; #####
;          KET THUC CAC KHAI BAO
; #####

```

### 6.3 Chương trình quét bàn phím

**Chương trình này được nạp cho 1 chip vi điều khiển loại 20 chân của ATMEL là AT89C2051**

```

; #####
; Chuong trinh quet ban phim 4x4
; Su dung P1 de giao tiep voi ban phim
; P1.0 - P1.3 noi voi cac hang lan luot 1 2 3 4
; P1.4 - P1.7 noi voi cac cot lan luot 1 2 3 4
; #####

ORG 0000H

; Truoc het ta doc cac hang 1-4, cac phim chuc nang, da duoc noi dat san
    mov P3, #0FFH
    call Delay
    mov P3, #00H
    call Delay
    mov P3, #0FFH

Start:

```

```

mov P1, #11111111B      ; Dat tat ca len 1
mov A, P1
anl A, #00001111B      ; Lay 4 bit thap cua P1, noi voi hang
cjne A, #00001111B, Doc_Cot_Chuc_Nang ; Kiem tra de chac rang tat ca cac
                                     phim chuc nang duoc tha
sjmp Doc_Ban_Phim

Doc_Cot_Chuc_Nang:
mov P1, #11111111B      ; Dat tat ca len 1
mov A, P1
anl A, #00001111B      ; Che cac bit cot di
F1:
cjne A, #00001110B, F2
Wait:
mov A, P1
anl A, #00001111B
cjne A, #00001111B, Wait ; Doi tha nut
clr P3.7
nop
nop
nop
mov P3, #16
call Delay ; Chong rung
ljmp Tiep_tuc_quet ; Tiep tục quét phím

F2:
cjne A, #00001101B, BACK_SPACE
Wait1:
mov P1, #11111111B      ; Dat tat ca len 1
mov A, P1
anl A, #00001111B
cjne A, #00001111B, Wait1 ; Doi tha nut
mov P3, #17
call Delay ; Chong rung
ljmp Tiep_tuc_quet

BACK_SPACE:
cjne A, #00001011B, ENTER
Wait2:
mov P1, #11111111B      ; Dat tat ca len 1
mov A, P1
anl A, #00000100B

```

```

    cjne    A, #00000100B, Wait2      ; Doi tha nut
    mov     P3, #18
    call    Delay                    ; Chong rung
    ljmp     Tiep_tuc_quet

ENTER:
    cjne    A, #00000111B, Start
Wait3:
    mov     P1, #11111111B           ; Dat tat ca len 1
    mov     A, P1
    anl     A, #00001000B
    cjne    A, #00001000B, Wait3 ; Doi tha nut
    clr     P3.7
    mov     P3, #19
    call    Delay                    ; Chong rung

Tiep_Tuc_Quet:
    WAIT4: ; Doi phim duoc tha het
    mov     P1, #11111111B           ; Dat tat ca len 1
    mov     A, P1
    anl     A, #00001111B
    cjne    A, #00001111B, Wait4
    call    Delay
    mov     P3, #0FFH; Tat ngat, khoa tin hieu o ngo ra tin hieu o ngo ra
    ljmp     Start                  ; Tiep tục quét phím

;;;;;;;;;;;;;
;Khong co phim chuc nang nao duoc an, tim phim thuong`
Doc_Ban_Phim:
    mov     P1, #11110000B           ; Noi dat tat ca cac hang
    mov     A, P1
    anl     A, #11110000B
    cjne    A, #11110000B, Co_Phim_Duoc_An ; Neu co phim duoc an
    jmp     Start                  ; Neu khong, tiep tục giám sát

Co_Phim_Duoc_An:
    MOV     P1, #11111110B           ; Noi dat hang 1
    MOV     A, P1
    anl     A, #11110000B
    cjne    A, #11110000B, ROW1

    MOV     P1, #11111101B           ; Noi dat hang 2
    MOV     A, P1

```

```

ANL    A,#11110000B
cjne   A,#11110000B, ROW2

MOV     P1,#11111011B           ; Noi dat hang 3
MOV     A,P1
ANL     A,#11110000B
cjne    A,#11110000B, ROW3

MOV     P1,#11110111B           ; Noi dat hang 4
MOV     A,P1
ANL     A,#11110000B
cjne    A,#11110000B, ROW4

ROW1:   MOV     DPTR,#KCODE1
        call    FIND
        ljmp    Start

ROW2:   MOV     DPTR,#KCODE2
        call    FIND
        ljmp    Start

ROW3:   MOV     DPTR,#KCODE3
        call    FIND
        ljmp    Start

ROW4:   MOV     DPTR,#KCODE4
        call    FIND
        ljmp    Start

; #####
;           THU TUC TIM KIEM TRONG BANG
; #####

FIND:
        rlc     A                ; Quay lan luot cac cot len co nho' de ktra
        jnc     MATCH           ; Neu = 0, dung la cot dang co phim duoc an
        inc     DPTR            ; Neu = 1, tiep tục tăng con trỏ DL và quay để tìm ra cột chứa phím
                                dc an
        sjmp    FIND

MATCH:
        CLR     A
        MOVC    A,@A+DPTR       ; Lúc này A sẽ chứa các giá trị từ 0-15

```



```

        MOV     P3,A           ; xuất ra ngoài, do A < 16 nên bit D7 bao giờ cũng = 0,
                                ; ta tạm dùng để tạo ra ngắt ngoài cho CPU
WAIT5:  ; Doi phim duoc tha het
        mov     P1, #11110000B; Noi dat tat ca cac hang
        mov     A, P1
        anl     A, #11110000B
        cjne    A, #11110000B, WAIT5
        mov     P3, #0FFH      ; Tat ngat, khoa tin hieu o ngo ra
        call    Delay
        ret

; #####
;          KET THUC THU TUC TIM KIEM
; #####

DELAY:
TT0:    MOV     R2, #180
        MOV     R1, #72
TT1:    DJNZ     R1, TT1
        DJNZ     R2, TT1
        RET

;Giai ma
KCODE1:DB 3, 2, 1, 0
KCODE2:DB 7, 6, 5, 4
KCODE3:DB 11, 10, 9, 8
KCODE4:DB 15, 14, 13, 12

```

END

## 6.4 Chương trình điều khiển hiển thị LCD

```

; #####
;          THU TUC XUAT LENH RA LCD
; #####

Command_To_LCD:
        push    DPH
        push    DPL
        push    02H

        mov     R2, A           ; Luu A
        mov     A, #CW_011      ; Dat A ra, B,C dau vao
        mov     DPTR, #CW1
        movx    @DPTR, A        ; Xuat tu dieu khien
        ; Doc thanh ghi lenh

```

```

; PA3 PA2 PA1 PA0
; E2 E1 RW RS
mov    A, E
cjne   A, #2, Nua_MH1
mov    A, #00001010B
jmp    Xuat_Ra
Nua_MH1:
mov    A, #00000110B
Xuat_Ra:
mov    DPTR, #LCD_CONTROL
movx   @DPTR, A      ; Xuat th dk ra LCD
mov    DPTR, #LCD_DATA
Ready_CMD:
movx   A, @DPTR      ; Doc DB vao
jnb    ACC.7, Ready_CMD
; Dat lai PA, PB lam cong ra
mov    A, #CW_000    ; Dat A,B,C dau RA
mov    PTR, #CW1
movx   @DPTR, A
mov    A, R2
mov    DPTR, #LCD_DATA
movx   @DPTR, A      ; Xuat ra

; Chot du lieu bang cach dua xung tu H to L o chan E1
mov    DPTR, #LCD_CONTROL

mov    A, E
cjne   A, #2, Nua_MH1_C
mov    A, #00001000B    ; E2 = 1; E1 = 1, RW = 0; RS = 0
jmp    Chot
Nua_MH1_C:
mov    A, #00000100B    ; E2 = 0; E1 = 1, RW = 0; RS = 0
Chot:
movx   @DPTR, A      ; E2 =1; E1 = 0, RW = 0; RS = 0
nop
nop

mov    A, #00000000B
movx   @DPTR, A      ; Chot du lieu

pop    02H
pop    DPL

```

```

    pop    DPH
    ret

; #####

; KET THUC THU TUC XUAT LENH RA LCD
; #####

; #####

; THU TUC IN KI TU RA LCD
; INPUT = A (ASCII)
; #####

In_Ki_Tu:
    push   DPH
    push   DPL
    push   02H
    mov     R2, A          ; Luu A
    mov     A, #CW_011     ; Dat A ra, B,C dau vao
    mov     DPTR, #CW1
    movx    @DPTR, A       ; Xuat tu dieu khien
    ; Doc thanh ghi lenh
    ; PA3 PA2 PA1 PA0
    ; E2 E1 RW RS
    mov     A, E
    cjne    A, #2, Nua_MH1_InKT
    mov     A, #00001010B
    jmp     Xuat_Ra_InKT

Nua_MH1_InKT:
    mov     A, #00000110B

Xuat_Ra_InKT:
    mov     DPTR, #LCD_CONTROL
    movx    @DPTR, A       ; Xuat ra LCD
    mov     DPTR, #LCD_DATA

Ready_Chr:
    movx    A, @DPTR       ; Doc DB vao
    jnb     ACC.7, Ready_Chr
    ; Dat lai PA, PB lam cong ra
    mov     A, #CW_000     ; Dat A,B,C dau RA
    mov     DPTR, #CW1
    movx    @DPTR, A       ; Xuat tu dieu khien

    mov     A, R2 ; Lay lai gia tri luu ban dau, lenh
    mov     DPTR, #LCD_DATA
    movx    @DPTR, A       ; Xuat ra

```

```

; Chot du lieu bang cach dua xung tu H to L o chan E1
mov    DPTR, #LCD_CONTROL
mov    A, E
cjne   A, #2, Nua_MH1_Chot
mov    A, #00001001B
jmp    Chot_InKT
Nua_MH1_Chot:
mov    A, #00000101B
Chot_InKT:
movx   @DPTR, A ; E2 =0; E1 = 1, RW = 0; RS = 1
nop
nop
mov    A, #00000001B
movx   @DPTR, A ; Chot du lieu
pop    02H
pop    DPL
pop    DPH
ret

; #####
; KET THUC CHUONG TRINH IN KI TU RA LCD
; #####
; #####
; THU TUC IN CHUOI RA LCD
; INPUT = DIA CHI CUA CON TRO DU LIEU DPTR
; #####

Str_To_LCD:
push   ACC
Continue_To_Print:
clr    A
movc   A, @A+DPTR
jz     Done
call   In_Ki_Tu
inc    DPTR
mov    A, #06h ; Dich con tro sang phai
call   Command_To_LCD
sjmp   Continue_To_Print
Done:
pop    ACC
ret

; #####
; KET THUC THU TUC IN CHUOI RA LCD

```

```
; #####
; #####
;      THU TUC XOA LCD
; #####
```

CLS:

```
mov    E, #1
mov    A, #01H
call   Command_To_LCD
mov    E, #2
mov    A, #01H
call   Command_To_LCD
mov    E, #1
mov    A, #0EH
call   Command_To_LCD
mov    A, #0CH; Bat hien thi, tat con tro
call   Command_To_LCD
ret
```

```
; #####
;      KET THUC THU TUC XOA LCD
; #####
```

```
; #####
;      THU TUC IN SO HEXA
; #####
```

phex:

phex8: ; In so hexa 8 bit, input = ACC

```
push    02
push    acc
swap    a
anl     a, #0FH
add     a, #246
jnc     phex_b
add     a, #7
```

phex\_b:

```
add     a, #58
mov     R2, A ;Luu
mov     A, #0EH
call    Command_To_LCD
mov     A, R2
call    In_Ki_Tu
```

```

        pop     acc
phex1:
        push   acc
        anl    a, #0FH
        add    a, #246
        jnc    phex_c
        add    a, #7
phex_c:
        add    a, #58
        mov    R2, A ;Luu
        mov    A, #0EH
        call   Command_To_LCD
        mov    A, R2
        call   In_Ki_Tu
        pop    acc
        pop    02
        ret
phex16:      ; In so hexa 16 bit, input = DPTR
        push   acc
        mov    a, dph
        acall  phex
        mov    a, dpl
        acall  phex
        pop    acc
        ret

```

```

; #####
;      KET THUC THU TUC IN SO HEXA
; #####

```

## 6.5 Các ví dụ DEMO

```

; #####
;      CAC VI DU DEMO
; #####
DEMO1:      ; LED DON - 8255-4
        mov    DPTR, #LED_DON
DEMO1_1:    ; Nhap nhay cac LED xen ke
        mov    R5, #48
        mov    R6, #10101010B
Again_DM11:
        mov    A, R6
        movx   @DPTR, A
        cpl    A
        call   Delay_100ms
        mov    R6, A
        mov    A, KEY ; Neu co nut an tro ve
        cjne   A, #KEY_BSP, TiepDemol_1 ; Tiep
        tuc   chay cho het vong lap
        jmp    Exit_DEMO1 ; Neu co lenh thoat
        thi    tro ve luon
TiepDemol_1:
        djnz   R5, Again_DM11

```

```

; Neu co lenh thoat thi tro ve luon
DEMO1_2:      ; Chay LED tu trai sang phai
; Nghi mot chut
mov    A, #00H
movx   @DPTR, A
call   Delay_500ms

mov    R5, #48
mov    R6, #10000000B
Again_DM12:
mov    A, R6
movx   @DPTR, A
call   Delay_100ms
rr     A
mov    R6, A
mov    A, KEY ; Neu co nut an tro ve
cjne   A, #KEY_BSP, TiepDemol_2
; Tiep tục chạy cho hết vòng lặp
jmp     Exit_DEMO1
; Neu co lenh thoat thi tro ve luon
TiepDemol_2:
djnz   R5, Again_DM12

DEMO1_3:      ; Chay LED tu phai sang trai
; Nghi mot chut
mov    A, #00H
movx   @DPTR, A
call   Delay_500ms

mov    R5, #48
mov    R6, #10000000B
Again_DM13:
mov    A, R6
movx   @DPTR, A
call   Delay_100ms
rl     A
mov    R6, A
mov    A, KEY
; Neu co nut an tro ve
cjne   A, #KEY_BSP, TiepDemol_3
; Tiep tục chạy cho hết vòng lặp
mp     Exit_DEMO1

; Neu co lenh thoat thi tro ve luon
TiepDemol_3:
djnz   R5, Again_DM13
Exit_DEMO1:
mov    KEY, #0FFH
call   Restore_SCR
; Tra lai man hinh truoc'

ret
; Tro ve man hinh 4, chon vd DEMO
; -----
DEMO2:      ; LED 7 DOAN
mov    Led7Doan1, #0
mov    Led7Doan2, #0
mov    Led7Doan3, #0
mov    Led7Doan4, #0

DEMO2_1:
; Dem len
mov    A, #DONG2
call   Command_To_LCD

mov    DPTR, #MSG11
call   Str_To_LCD

mov    A, #DONG3
call   Command_To_LCD

mov    DPTR, #MSG12
call   Str_To_LCD

Dem_Len:
call   Display_Four_7Seg_Led
call   Delay_100ms
mov    A, KEY
cjne   A, #KEY_BSP, Sang_Dem_Xuong
mov    KEY, #0FFH
jmp     Exit_DEMO2

Sang_Dem_Xuong:
cjne   A, #1, TiepTucDemLen

```

```

mov     KEY, #0FFH
jmp     Exit_DEMO2_1

TiepTucDemLen:
mov     A, Led7Doan4
inc     A
mov     Led7Doan4, A
cjne    A, #0AH, Dem_Len
add     A, #6
anl     A, #00001111B
mov     Led7Doan4, A

mov     A, Led7Doan3
inc     A
mov     Led7Doan3, A
cjne    A, #0AH, Dem_Len
add     A, #6
anl     A, #00001111B
mov     Led7Doan3, A

mov     A, Led7Doan2
inc     A
mov     Led7Doan2, A
cjne    A, #0AH, Dem_Len
add     A, #6
anl     A, #00001111B
mov     Led7Doan2, A

mov     A, Led7Doan1
inc     A
mov     Led7Doan1, A
cjne    A, #0AH, Dem_Len
add     A, #6
anl     A, #00001111B
mov     Led7Doan1, A
jmp     Dem_Len

Exit_DEMO2_1:
; Thoat dem len, chuyen sang dem xuong
mov     KEY, #0FFH
mov     A, #DONG2
call    Command_To_LCD

mov     DPTR, #MSG13
call    Str_To_LCD

mov     A, #DONG3
call    Command_To_LCD

mov     DPTR, #MSG14
call    Str_To_LCD

DEMO2_2:      ; ---- Dem xuong'
call     Display_Four_7Seg_Led
call     Delay_100ms
mov     A, KEY
cjne    A, #KEY_BSP, Sang_Dem_Len
mov     KEY, #0FFH
jmp     Exit_DEMO2

Sang_Dem_Len:
cjne    A, #0, TiejTucDemXuong
mov     KEY, #0FFH
jmp     DEMO2_1

TiejTucDemXuong:
mov     A, Led7Doan4
dec     A
mov     Led7Doan4, A
cjne    A, #0FFH, DEMO2_2
mov     Led7Doan4, #9

mov     A, Led7Doan3
dec     A
mov     Led7Doan3, A
cjne    A, #0FFH, DEMO2_2
mov     Led7Doan3, #9

mov     A, Led7Doan2
dec     A
mov     Led7Doan2, A
cjne    A, #0FFH, DEMO2_2
mov     Led7Doan2, #9

```



```

mov    A, Led7Doan1
dec    A
mov    Led7Doan1, A
cjne   A, #0FFH, DEMO2_2
mov    Led7Doan1, #9

jmp     DEMO2_2

Exit_DEMO2:

    ret
; -----
DEMO3:      ; LED MA TRAN
DEMO3_1:
    mov    DPTR, #MATRIX_LED_ANOD
    mov    A, #0FFH
    movx   @DPTR, A

    mov    A, #11001100B
    mov    DPTR, #MATRIX_LED_R
    mov    R7, #32
Again_DM3_1: ; Nhay LED của các cột, xen kẽ
    movx   @DPTR, A
    cpl    A
    call   Delay_100ms
    djnz   R7, Again_DM3_1

DEMO3_2:
    mov    DPTR, #MATRIX_LED_ANOD
    mov    A, #0FFH
    movx   @DPTR, A

    mov    A, #11001100B
    mov    DPTR, #MATRIX_LED_R
    mov    R7, #32
Again_DM3_2: ; Chạy LED
    movx   @DPTR, A
    rr     A
    call   Delay_100ms
    djnz   R7, Again_DM3_2

DEMO3_3:
    mov    DPTR, #MATRIX_LED_ANOD
    mov    A, #0FFH
    movx   @DPTR, A

    mov    A, #11001100B
    mov    DPTR, #MATRIX_LED_R
    mov    R7, #32
Again_DM3_3: ; Chạy LED
    movx   @DPTR, A
    rl     A
    call   Delay_100ms
    djnz   R7, Again_DM3_3

DEMO3_4: ; Thử nghiệm với các cột
    mov    A, #0FFH
    mov    DPTR, #MATRIX_LED_R
    movx   @DPTR, A

    mov    A, #0FFH
    mov    DPTR, #MATRIX_LED_G
    movx   @DPTR, A

    mov    DPTR, #MATRIX_LED_ANOD
    mov    A, #01H
    movx   @DPTR, A

    mov    R7, #64
Again_DM3_4: ; Chạy LED
    movx   @DPTR, A
    rl     A
    call   Delay_100ms
    djnz   R7, Again_DM3_4
    ret

DEMO4:      ; BÀN PHÍM, LCD
    ret

```



```

        subb    A, #100
        jc      Nhiet_Do_Nho_Hon_100
Nhiet_Do_Lon_Hon_100:
        mov     A, Nhiet_Do
        dec     A ; Tru 1
        rl      A
        rl      A ; Nhan 4 de nhay cho dung
vi tri chuoi can tra trong bang
        mov     R6, #4
        mov     R0, A
        mov     DPTR, #LON_HON_100_DO
In_Nhiet_Do_1:
        movc    A, @A + DPTR
        call    In_Ki_Tu
        inc     R0
        mov     A, R0
        djnz    R6, In_Nhiet_Do_1
        jmp     In_Do_C

Nhiet_Do_Nho_Hon_100:
        mov     A, Nhiet_Do
        dec     A ; Tru 1
        rl      A ; Nhan 2 de nhay cho dung
vi tri chuoi can tra trong bang
        mov     R6, #2
        mov     R0, A
        mov     DPTR, #NHO_HON_100_DO
In_Nhiet_Do_2:
        movc    A, @A + DPTR
        call    In_Ki_Tu
        inc     R0
        mov     A, R0
        djnz    R6, In_Nhiet_Do_2
In_Do_C:
        mov     A, #' '
        call    In_Ki_Tu
        mov     A, #'d'
        call    In_Ki_Tu
        mov     A, #'o'
        call    In_Ki_Tu
        mov     A, #' '
        call    In_Ki_Tu

        mov     A, #'C'
        call    In_Ki_Tu
        mov     A, #' '
        call    In_Ki_Tu
        call    Delay_500ms
        call    Delay_500ms
        mov     A, KEY
        cjne    A, #KEY_ENTER, KT_Thoat_ADC
        mov     KEY, #0FFH
        jmp     DAC

KT_Thoat_ADC:
        cjne    A, #KEY_BSP, Tiep_Tuc_ADC
        mov     KEY, #0FFH
        call    Restore_SCR
        ret

Tiep_Tuc_ADC:
        jmp     ADC

; -----
----
DAC:
        mov     E, #2
        mov     A, #01
        call    Command_To_LCD

        mov     A, #DONG2
        call    Command_To_LCD

        mov     DPTR, #MSG38 ; 'Thi nghiem voi
DAC'
        call    Str_To_LCD

; DAC_DB EQU 8009H
; Bus du lieu cua DAC0808
        mov     A, #DONG3
        call    Command_To_LCD

        mov     DPTR, #MSG36
        call    Str_To_LCD

```

```

    mov     Ap_Ra, #10
Tang_Ap_DAC:
    mov     A, Ap_Ra
Dao_Gia_Tri:
    rrc     A
    mov     07H, C
    rrc     A
    mov     06H, C
    rrc     A
    mov     05H, C
    rrc     A
    mov     04H, C
    rrc     A
    mov     03H, C
    rrc     A
    mov     02H, C
    rrc     A
    mov     01H, C
    rrc     A
    mov     00H, C

    mov     A, 20H

    mov     DPTR, #DAC_DB
    movx    @DPTR, A
    mov     A, Ap_Ra
    add     A, #2
    mov     Ap_Ra, A
    call    Delay_500ms
    call    Delay_500ms
    call    Delay_500ms
    call    Delay_500ms
    call    Delay_500ms
    call    Delay_500ms
    call    Delay_500ms

    mov     A, Ap_Ra
    cjne    A, #254, KT_Thoat_Tang_Ap_DAC
    jmp     Exit_DEMO5

KT_Thoat_Tang_Ap_DAC:
    mov     A, KEY
    cjne    A, #KEY_BSP, Tang_Ap_DAC

    mov     KEY, #0FFH
Exit_DEMO5:
    call    Restore_SCR
    ret

DEMO6: ; STEPPER, DC MOTOR
;DIGITAL_INPUT EQU 800CH ; Dau vao so
;DC_STEPPER EQU 800DH ; Dieu khien dong co
    buoc (PB0-PB3), dong co 1 chieu(PB4-PB7

; PULSE_INPUT EQU 800EH ; 4 dau vao xung =
    nut(PC4-PC7), 4 tu cac nguon phat
    xung(PC0-PC3)

; CW3 EQU 800FH ; Tu dieu khien CW cua
8255-3

    mov     DPTR, #DC_STEPPER
    mov     R4, #00110011B
    mov     R7, #255
    mov     R6, #5
    mov     Nut1, #0
    mov     Nut2, #0
    mov     Nut3, #0
    mov     Nut4, #0

Quay_Thuan:
    mov     A, R4
    movx    @DPTR, A
    rl      A
    mov     R4, A
    call    Delay_7ms
    call    Delay_7ms
    call    Delay_7ms
    mov     A, KEY
    cjne    A, #KEY_BSP, Case_Enter_QT
    mov     KEY, #0FFH
    call    Restore_SCR
    ret

; MSG29: db 'ENTER=Test pulse input', 0
; MSG30: db 'ENTER=Test digital
input',0
Case_Enter_QT:
    mov     A, KEY

```

```

    cjne    A, #KEY_ENTER, Du_So_Buoc1
    mov     KEY, #0FFH
;Hien thi thong bao bam enter de quay
nguoc
    mov     E, #2
    mov     A, #DONG1
    call    Command_To_LCD

    mov     DPTR, #MSG29
    call    Str_To_LCD
    mov     R7, #255
    mov     R6, #5
    mov     R4, #00110011B
    mov     DPTR, #DC_STEPPER
    jmp     Quay_Nguoc

```

```

Du_So_Buoc1:
    djnz    R7, Quay_Thuan
    mov     R7, #255
    djnz    R6, Quay_Thuan

```

Quay\_Nguoc:

```

    mov     A, R4
    movx    @DPTR, A
    rr      A
    mov     R4, A
    call    Delay_7ms
    call    Delay_7ms
    call    Delay_7ms

```

```

    mov     A, KEY
    cjne    A, #KEY_BSP, Case_Enter_QN
    mov     KEY, #0FFH
    call    Restore_SCR
    ret

```

Case\_Enter\_QN:

```

    mov     A, KEY
    cjne    A, #KEY_ENTER, Du_So_Buoc2
    mov     KEY, #0FFH
;Hien thi thong bao bam enter de quay
TEST pulse input
    mov     E, #2

```

```

    mov     A, #DONG1
    call    Command_To_LCD

    mov     DPTR, #MSG30
    call    Str_To_LCD
    mov     DPTR, #DC_STEPPER
    mov     A, #0
    movx    @DPTR, A
    jmp     Pulse_Digital_Input_Test

```

Du\_So\_Buoc2:

```

    djnz    R7, Quay_Nguoc
    mov     R7, #255
    djnz    R6, Quay_Nguoc

```

Pulse\_Digital\_Input\_Test:

```

; MSG32:  db 'Digital input value:  ',0
    mov     E, #2
    mov     A, #DONG2
    call    Command_To_LCD

```

```

    mov     DPTR, #MSG31
    call    Str_To_LCD

    mov     A, #DONG3
    call    Command_To_LCD

    mov     DPTR, #MSG32
    call    Str_To_LCD

```

```

; DIGITAL_INPUT      EQU      800CH      ;
Dau vao so

; DC_STEPPER          EQU      800DH      ;
Dieu khien dong co buoc (PB0-PB3), dong co 1
chieu (PB4-PB7)

; PULSE_INPUT         EQU      800EH      ;
4 dau vao xung = nut (PC4-PC7), 4 tu cac nguon
phat xung (PC0-PC3)

; CW3                  EQU      800FH      ;
; Tu dieu khien CW cua 8255-3

```

Quet\_phim:

```

    mov     DPTR, #PULSE_INPUT
    movx    A, @DPTR

```

```

    cjne    A, #0FFH, Phim1

    jmp     Cap_nhat_gia_tri

Phim1:
    movx   A, @DPTR
    anl    A, #11110000B      ; Neu phim
    thu 1 dc an
    cjne   A, #01110000B, Phim2
Cho_Phim1_tha:
    mov    DPTR, #PULSE_INPUT
    movx   A, @DPTR
    anl    A, #11110000B
    cjne   A, #11110000B, Cho_Phim1_tha
    call   Delay_7ms
    call   Delay_7ms
    call   Delay_7ms
    inc    Nut1      ; Duoc tha roi thi tang
    Nut1 - chua tong so lan bam len
    jmp    Cap_nhat_gia_tri

Phim2:
    movx   A, @DPTR
    anl    A, #11110000B      ; Neu phim
    thu 2 duoc an
    cjne   A, #10110000B, Phim3
Cho_Phim2_tha:
    movx   A, @DPTR
    anl    A, #11110000B
    cjne   A, #11110000B, Cho_Phim2_tha
    call   Delay_7ms
    call   Delay_7ms
    call   Delay_7ms
    inc    Nut2      ; Duoc tha roi thi tang
    Nut1 - chua tong so lan bam len
    jmp    Cap_nhat_gia_tri

Phim3:
    movx   A, @DPTR
    anl    A, #11110000B      ; Neu phim
    thu 3 duoc an
    cjne   A, #11010000B, Phim4
Cho_Phim3_tha:
    movx   A, @DPTR
    anl    A, #11110000B
    cjne   A, #11110000B, Cho_Phim3_tha
    call   Delay_7ms
    call   Delay_7ms
    call   Delay_7ms
    inc    Nut3      ; Duoc tha roi thi tang
    Nut1 - chua tong so lan bam len
    jmp    Cap_nhat_gia_tri

Phim4:
    movx   A, @DPTR
    anl    A, #11110000B      ; Neu phim
    thu 4 duoc an
    cjne   A, #11100000B, Cap_nhat_gia_tri
Cho_Phim4_tha:
    movx   A, @DPTR
    anl    A, #11110000B
    cjne   A, #11110000B, Cho_Phim4_tha
    call   Delay_7ms
    call   Delay_7ms
    call   Delay_7ms
    inc    Nut4      ; Duoc tha roi thi tang
    Nut1 - chua tong so lan bam len

Cap_nhat_gia_tri:
Cap_nhat_gia_tri_digital_input:
    mov    A, #DONG3 + 28
    call   Command_To_LCD

    mov    DPTR, #DIGITAL_INPUT
    movx   A, @DPTR
    call   phex8      ; in ra gia tri cua
                      digital input

; MSG31: db 'B1: |B2: |B3: |B4: ',0
Cap_nhat_so_lan_bam_cac_nut:
    mov    A, #DONG2 + 3
    call   Command_To_LCD

    mov    A, Nut1
    call   phex8

```

```

mov    A, #DONG2 + 9
call   Command_To_LCD
mov    A, Nut2
call   phex8

mov    A, #DONG2 + 15
call   Command_To_LCD
mov    A, Nut3
call   phex8

mov    A, #DONG2 + 21
call   Command_To_LCD
mov    A, Nut4
call   phex8

mov    A, KEY
cjne   A, #KEY_BSP, Tiep_tuc
mov    KEY, #0FFH
call   Restore_SCR
ret

Tiep_tuc:
    jmp    Quet_Phim

ret

DEMO7:    ; RS232
call      Disable_SERIAL_INT
; Lam viec voi nua man hinh 2
mov       E, #2

mov       A, #01
call      Command_To_LCD ; Xoa LCD

mov       A, #DONG1
call      Command_To_LCD
; Select receive mode, bam nut 0
mov       DPTR, #MSG24
call      Str_TO_LCD

mov       A, #DONG2
call      Command_To_LCD

; Select receive mode, bam nut 1
mov       DPTR, #MSG25
call      Str_TO_LCD
mov       A, #0CH ; Giao con tro
call      Command_To_LCD

Cho_Phim_0_DM7:
    mov    A, KEY
    cjne   A, #0, Cho_Phim_1_DM7
    mov    KEY, #0FFH
    jmp    Receive_Mode

Cho_Phim_1_DM7:
    mov    A, KEY
    cjne   A, #1, Cho_Phim_BSP_DM7
    mov    KEY, #0FFH
    jmp    Send_Mode

Cho_Phim_BSP_DM7:
    mov    A, KEY
    cjne   A, #KEY_BSP, Cho_Phim_0_DM7
    mov    KEY, #0FFH
    call   Enable_SERIAL_INT
    ret

Receive_Mode:
    mov    A, #DONG3
    call   Command_To_LCD

    mov    DPTR, #MSG26
    call   Str_TO_LCD

    mov    A, #DONG4
    call   Command_To_LCD

    mov    A, #0FH ; Nhap nhay con tro
    call   Command_To_LCD
    clr    RI
    mov    R7, #23

Doi_Nhan_Du_Lieu:
    mov    A, KEY
    cjne   A, #KEY_BSP, Co_Du_Lieu_Den

```

```
mov     KEY, #0FFH
jmp     DEMO7

Co_Du_Lieu_Den:
    jnb     RI, Doi_Nhan_Du_Lieu
    mov     A, SBUF
    call    In_Ki_Tu

    djnz    R7, Chua_Het_Hang
    mov     R7, #24
    mov     A, #DONG4 ; Tro ve dau dong
    call    Command_To_LCD

Chua_Het_Hang:
    clr     RI
    jmp     Doi_Nhan_Du_Lieu

; -----
Send_Mode:
    mov     A, #DONG3
    call    Command_To_LCD

    mov     DPTR, #MSG27
    call    Str_To_LCD

Continue_Transfer:
    mov     DPTR, #MSG22
    call    Send_String

    mov     A, #13
    call    Send_Char

    mov     A, #10
    call    Send_Char

    mov     DPTR, #MSG23
    call    Send_String

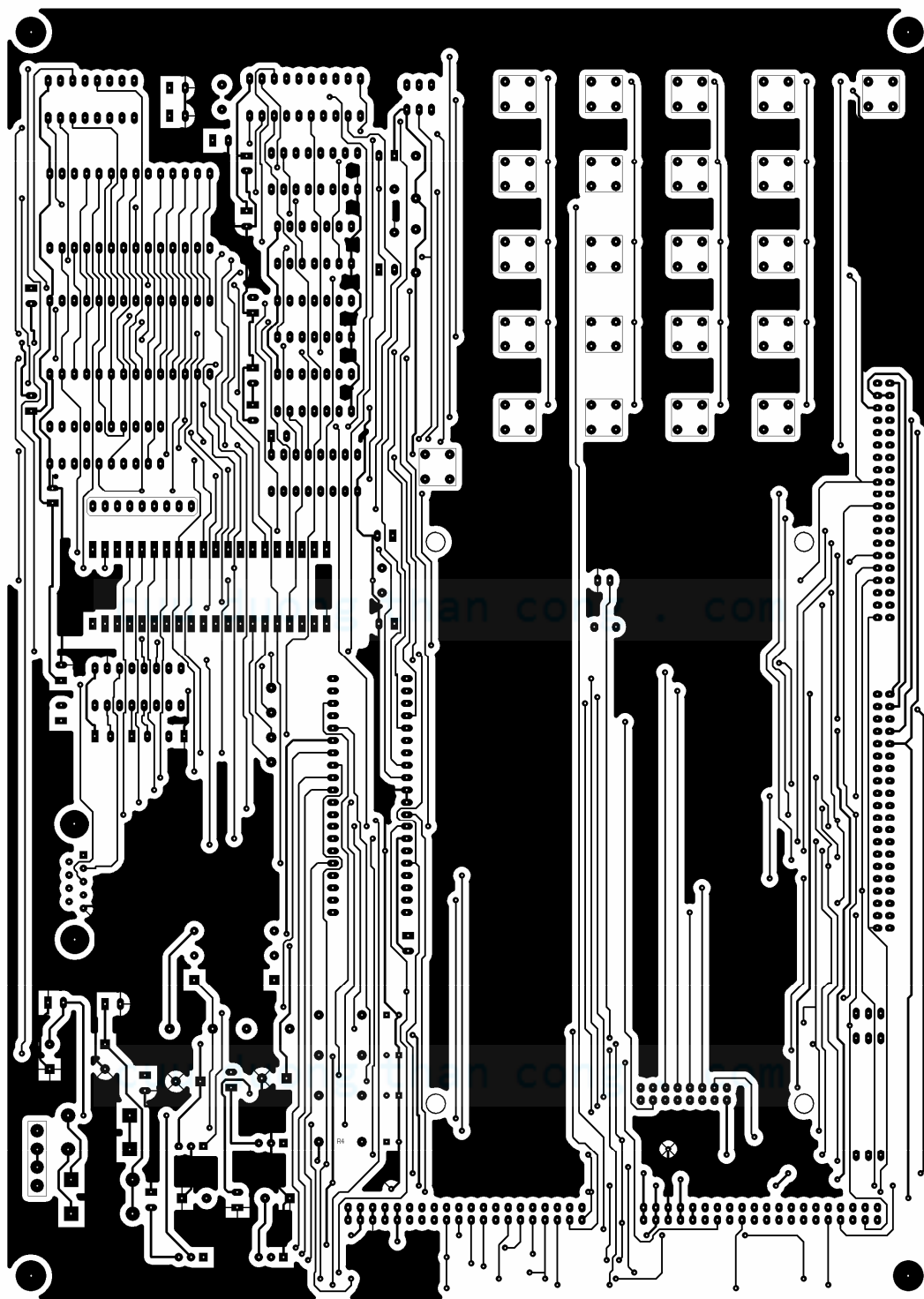
    mov     A, #13
    call    Send_Char

    mov     A, #10
    call    Send_Char

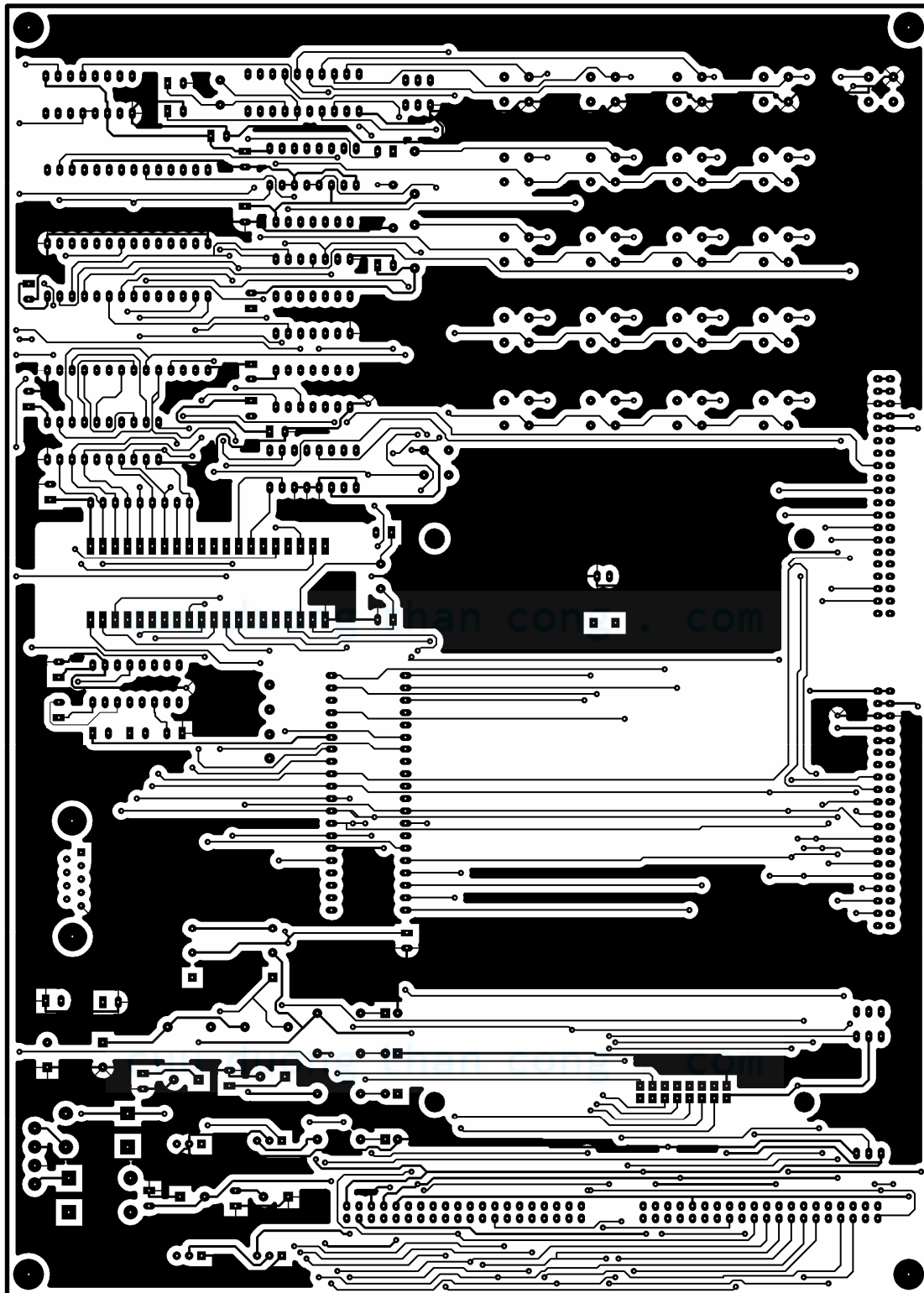
    call    Delay_500ms
    call    Delay_500ms
    call    Delay_500ms
    call    Delay_500ms
    mov     A, KEY
    cjne    A, #KEY_BSP, Continue_Transfer
    mov     KEY, #0FFH
    jmp     DEMO7
```



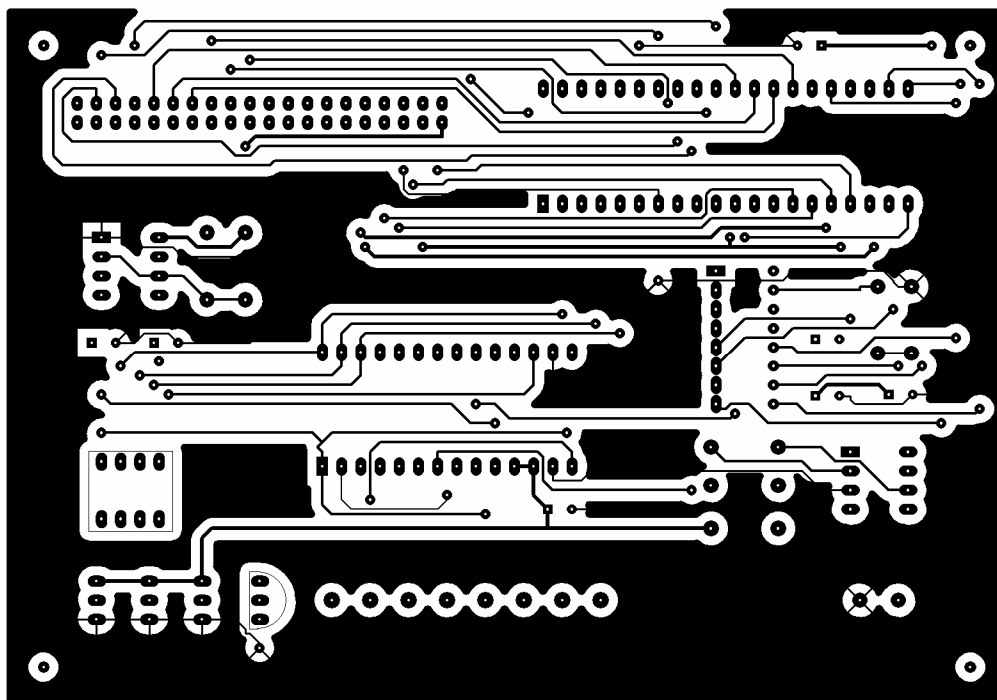
## 6.5 Sơ đồ mạch in của các board



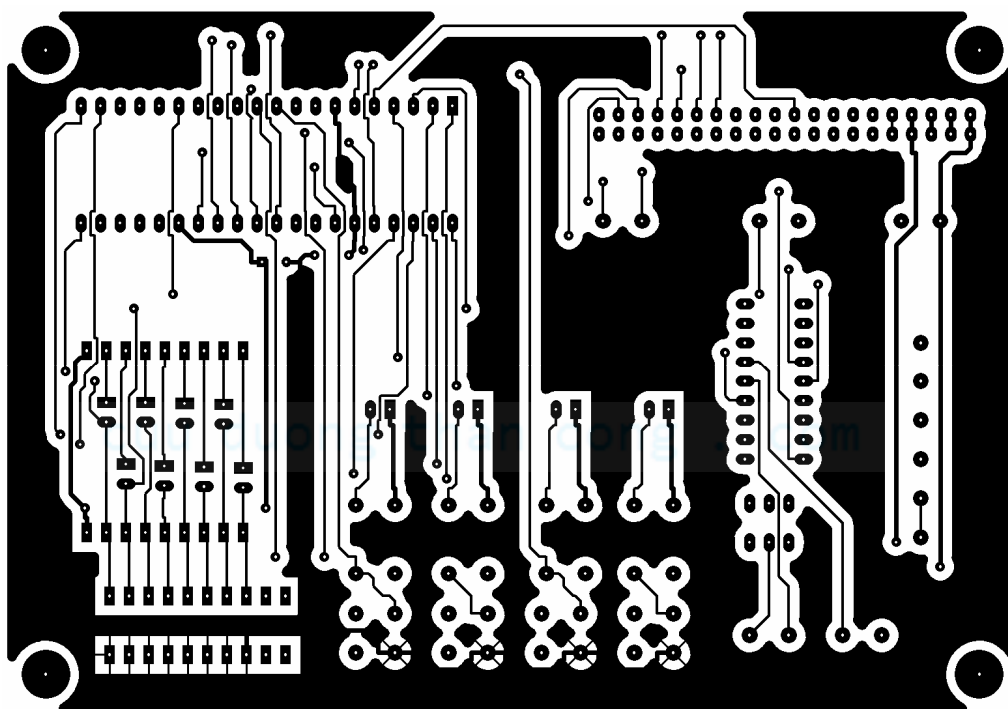
*Sơ đồ mạch in của BOARD CPU - mặt trên*



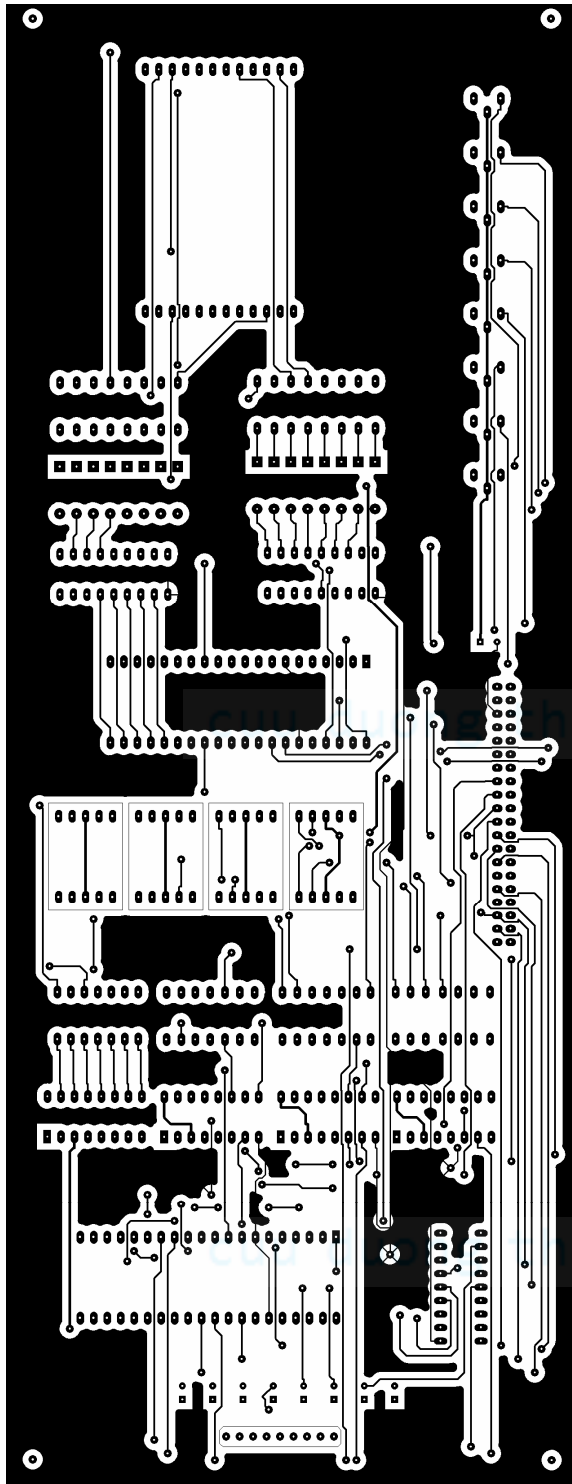
*Sơ đồ mạch in của BOARD CPU - mặt dưới*



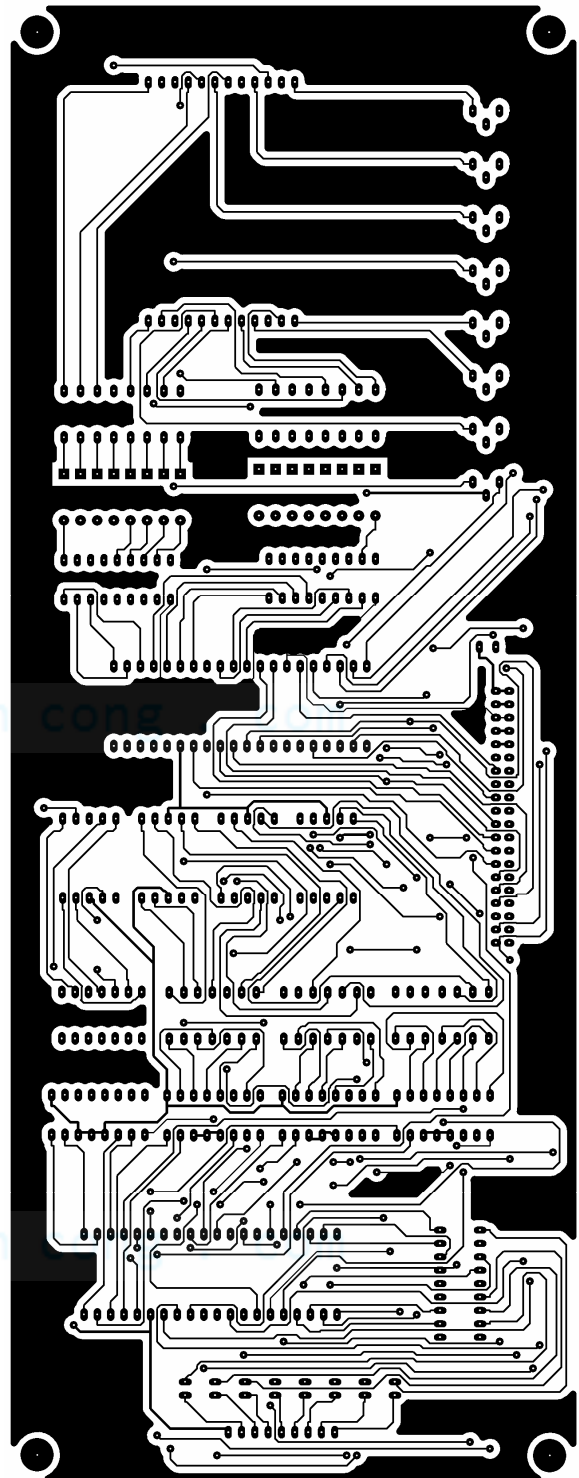
*Sơ đồ mạch in của BOARD ADC - mặt trên*



*Sơ đồ mạch in của BOARD xung số, động cơ bước- mặt dưới*



*Sơ đồ mạch in của BOARD LED  
đơn, LED 7 đoạn và LED ma trận  
(mặt trên)*



*(mặt dưới)*

**Tài liệu tham khảo**

1. Cấu trúc và lập trình họ vi điều khiển 8051 – Nguyễn Tăng Cường – Nhà xuất bản khoa học kỹ thuật – 2004
2. Họ vi điều khiển 8051 – Tống Văn On, Hoàng Đức Hải – Nhà xuất bản Lao động –Xã hội – 2001
3. Các tài liệu tham khảo khác dưới dạng html và dạng pdf đi kèm theo CD ở thư mục “Tai lieu tham khao”

cuu duong than cong . com

cuu duong than cong . com