

A. 題目

應試者必須在LCM 進階選單畫面的選項2 上呈現出"OPS"等文字，在週邊板背面前方共有四個CNY70光學感測器，這四個光學感測器依照前(F)後(B)左(L)右(R)的位置等距分別排列，這四個光學感測器的編號分別為編號U9 的前方(F)、編號U10 的後面(B)、編號U17 的左邊(L)、編號U11 的右邊(R)，當遮住任何一個光學感測器時都會在LCM 上顯示相對的狀態訊息；如果週邊電路板作為自走車使用時，則可以用來偵測地板上的黑色電工膠帶。應試者在參考電路圖完成設計之後，於進階選單狀態之下，按數字"2"鍵進入光學感測器功能選項，此時LCM畫面的的一列必須顯示"FBLR"等文字，而第二列則呈現不斷從四個光學感測器所立即讀取到的資料；如果以黑色物體遮住前方(F)位置的光學感測器、而其它感測器置於白色桌面上時，則在LCM 畫面上必須立即顯示如下資料：

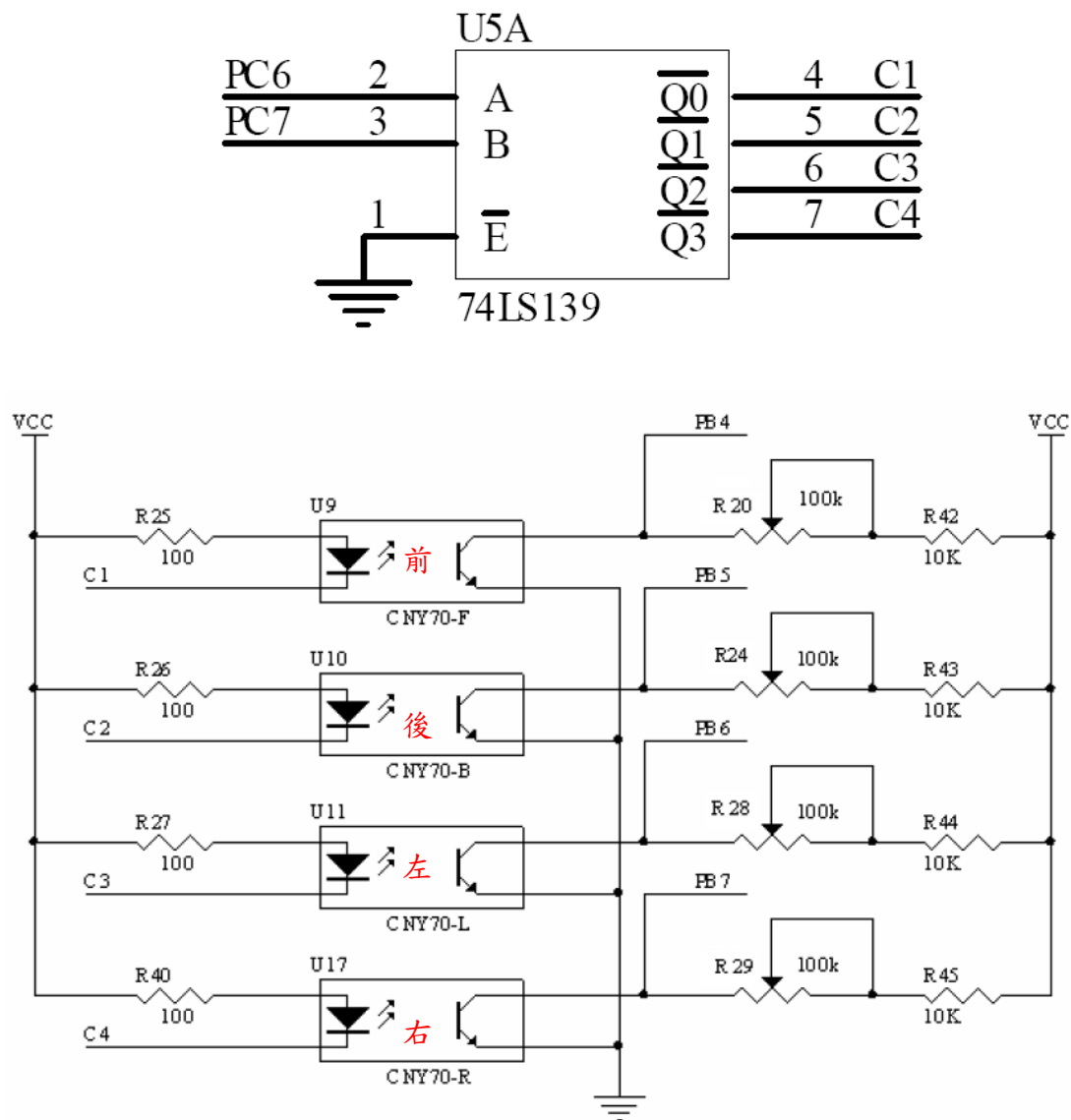
FBLR
1000

如果再以黑色物體遮住右邊(R)與左邊(L)位置的光學感測器，則在LCM 畫面上會隨即顯示如下資料：

FBLR
0011

當操作者按下"Mu"鍵後LCM 立即回到進階選單的畫面。

B. 參考電路及程式說明：



電路說明：

電路(參考題本第 14 頁)中的 CNY70 是由 74139 的輸出所控制，當 74139 的輸入 B(PC7)、A(PC6)依序從 00 變化到 11 時，輸出 Q0~Q3 同一時間會有一組輸出為 0，其餘為 1，控制 CNY70 同一時間只會有一組動作，兩者間的關係可參考下表：

B (PC7)	A (PC6)	Q3 (C4 右)	Q2 (C3 左)	Q1 (C2 後)	Q0 (C1 前)
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

假設當 C1 輸入為 0 時，U9 的紅外線 LED 發射訊號，當地板為白色時(測試時可用手遮住感測器代替)，可反射紅外線，光電晶體接收到紅外線導通，R20 和 R42 會有電流通過產生壓降，使

得 PB4 讀回的訊號為 0，如果是黑色物體遮住時（測試時可讓感測器下方無任何物品，沒辦法反射訊號代替），因紅外線被黑色物體吸收，光電晶體不導通，所以 PB4 讀回的訊號為 1，再同一時間，C2~C4 這三個紅外線 LED 發射因為沒有訊號，光電晶體不管是何種地面都不會導通，所以體讀回的值一定是 1

B (PC7)	A (PC6)	掃描位置	讀回值				16 進制
			PB7	PB6	PB5	PB4	
0	0	前	1	1	1	0(白) 1(黑)	e(白) f(黑)
0	1	後	1	1	0(白) 1(黑)	1	d(白) f(黑)
1	0	左	1	0(白) 1(黑)	1	1	b(白) f(黑)
1	1	右	0(白) 1(黑)	1	1	1	7(白) f(黑)

由上面表格可看出，不管是啟動哪個 CNY70 感測器，只要地面黑的，PB7~PB4 讀回的值就是 f，如果不是 f，就是地面是白的。

根據上述原理，可修改以下考場提供的副程式，這段副程式，最原始的功能，是將獨到的光感測器狀態，顯示於 P1，將它修改成可判斷讀到情形，分別顯示 0 或 1。

```
data_pro:
    anl    pb55, #0f0h    ;pb4-pb7
    mov    cny70, pb55    ;store to cny70
    mov    p1, cny70      ;display sensors status
    ret
```

可以改成

```
data_pro:
    anl    pb55, #0f0h    ;pb4-pb7
    mov    a, pb55        ;儲存 cny70 訊號到累加器
    cjne   a, #0f0h, show0 ;判斷讀回值(PB7~PB4)是否全是 1，
                           ; 如果不是等於(f0)，表示是白底，則跳去顯示"0"

    mov    abuf, #'1'
    call   wr_lcddat      ;黑底(全是 1)顯示 1
    ret

show0:
    mov    abuf, #'0'
    call   wr_lcddat      ;白底顯示 0
    ret
```

B. 參考程式：(原始檔名 sensor.a)

參考程式位於 OPS-cny70 資料夾內的 sensor.a

```

;-----8255 & sensors-----
ctrl55    equ    0x5f
pa55      equ    0x5e
pb55      equ    0x5d
pc55      equ    0x5c
pcbit     equ    0x5b
;
cny70     equ    0x55    ;store sensor's status
tdl70     equ    0x54    ;sensor hold delay time
;
a0_8255   equ    p2.0
a1_8255   equ    p2.1
;=====
=====
        org     00h
        mov     p1,#0xff ;if p1=>8 leds
;
sensor:
        mov     ctrl55,#0x82 ;pa:output(lcm),pb:input,pc:output
        call    wr_ctrl
scan_y70:
        mov     cny70,#0 ;initial sensor=0
;front
        ;74139=>q0=0
        mov     pcbit,#6 ;pc6=0
        call    cl_pc
        mov     pcbit,#7 ;pc7=0
        call    cl_pc
        call    y70_d1
        call    rd_pb ;read pb,data store at pb55
        call    data_pro ;data processing
;back
        ;74139=>q1=0
        mov     pcbit,#6 ;pc6=1
        call    st_pc
        mov     pcbit,#7 ;pc7=0
        call    cl_pc
        call    y70_d1
        call    rd_pb ;read pb,data store at pb55
        call    data_pro ;data processing
;left
        ;74139=>q2=0
```

只需拷貝這 2 行，其餘 8255 宣告部份原主程式已經有，不能重複宣告。

```

        mov     cbit,#6      ;pc6=0
        call    cl_pc
        mov     pcbit,#7     ;pc7=1
        call    st_pc
        call    y70_d1
        call    rd_pb        ;read pb,data store to pb55
        call    data_pro     ;data processing
;right                                ;74139=>q3=0
        mov     pcbit,#6     ;pc6=1
        call    st_pc
        mov     pcbit,#7     ;pc7=1
        call    st_pc
        call    y70_d1
        call    rd_pb        ;read pb,data store to pb55
        call    data_pro     ;data processing
;
        jmp     scan_y70
;-----data processing
data_pro:
        anl     pb55,#0f0h    ;pb4-pb7
        mov     cny70,pb55    ;store to cny70
        mov     p1,cny70     ;display sensors status
        ret
;-----sensor hold delay time
y70_d1:
        mov     tdl70,#20
        djnz    tdl70,$
        ret
;=====
=====
;*****write data to control register
wr_ctrl:                                ;ctrl register select=11
        setb    a0_8255
        setb    a1_8255
        mov     a,ctrl155
        movx    @r0,a         ;wr signal
        ret
;*****output data to 8255@pa
wr_pa:                                ;pa register select=00
        clr     a0_8255
        clr     a1_8255
        mov     a,pa55
        movx    @r0,a         ;wr signal

```

以下 8255 宣告部份原主程式已經有，不能重複宣告。

```

        ret
;*****input data from 8255@pb
rd_pb:                ;pb register select=01
        setb    a0_8255
        clr     a1_8255
        movx    a,@r0        ;rd signal
        mov     pb55,a
        ret
;*****output data to 8255@pc by single-bit model
st_pc:                ;pc register select=10
        setb    a0_8255        ;ctrl register select=11
        setb    a1_8255
        mov     a,pcbit
        setb    c
        rlc     a
        movx    @r0,a        ;wr signal
        ret
;
cl_pc:
        setb    a0_8255        ;ctrl register select=11
        setb    a1_8255
        mov     a,pcbit
        clr     c
        rlc     a
        movx    @r0,a        ;wr signal
        ret
;
        end

```

C. 光學感測器程式設計說明：

以下的步驟得背起來

步驟一：複製以下幾行到暫存器宣告區

```
cny70      equ      0x55      ;store sensor's status
tdl70      equ      0x54      ;sensor hold delay time
```

步驟二：將 scan_y70 開始後面的副程式，以下全部複製到考試程式的副程式區

scan_y70:

```
    mov      cny70,#0      ;initial sensor=0
;front                                ;74139=>q0=0 讀取前方感測器
    mov      pcbit,#6      ;pc6=0
    call     cl_pc
    mov      pcbit,#7      ;pc7=0
    call     cl_pc
    call     y70_d1
    call     rd_pb          ;read pb,data store at pb55
    call     data_pro       ;data processing
;back                                ;74139=>q1=0 讀取後方感測器
    mov      pcbit,#6      ;pc6=1
    call     st_pc
    mov      pcbit,#7      ;pc7=0
    call     cl_pc
    call     y70_d1
    call     rd_pb          ;read pb,data store at pb55
    call     data_pro       ;data processing
;left                                ;74139=>q2=0 讀取左邊感測器
    mov      cbit,#6       ;pc6=0
    call     cl_pc
    mov      pcbit,#7      ;pc7=1
    call     st_pc
    call     y70_d1
    call     rd_pb          ;read pb,data store to pb55
    call     data_pro       ;data processing
;right                               ;74139=>q3=0 讀取右邊感測器
    mov      pcbit,#6      ;pc6=1
    call     st_pc
    mov      pcbit,#7      ;pc7=1
    call     st_pc
```

這加上控制 LCD 游標移到第二
行開頭的命令

```

        call    y70_d1
        call    rd_pb      ;read pb,data store to pb55
        call    data_pro   ;data processing
;
        jmp     scan_y70
;-----data processing
data_pro:
        anl     pb55,#0f0h   ;pb4-pb7
        mov     cny70,pb55   ;store to cny70
        mov     p1,cny70    ;display sensors status
        ret
;-----sensor hold delay time
y70_d1:
        mov     tdl70,#20
        djnz    tdl70,$
        ret

```

這行改成 **ret**，不然程式會一直掃描感測器，跳不回主選單

這裡加上判斷式

延遲時間太短，請改成 **250**，以確定讀到正確結果

修改結果如下，紅字是修改後的部份

```

scan_y70:
        mov     cny70,#0    ;initial sensor=0
        call    line2       ;將游標移到第二行開頭
;front
        ;74139=>q0=0
        mov     pcbit,#6    ;pc6=0
        call    cl_pc
        mov     pcbit,#7    ;pc7=0
        call    cl_pc
        call    y70_d1
        call    rd_pb      ;read pb,data store at pb55
        call    data_pro   ;data processing
;back
        ;74139=>q1=0
        mov     pcbit,#6    ;pc6=1
        call    st_pc
        mov     pcbit,#7    ;pc7=0
        call    cl_pc
        call    y70_d1
        call    rd_pb      ;read pb,data store at pb55
        call    data_pro   ;data processing
;left
        ;74139=>q2=0
        mov     pcbit,#6    ;pc6=0
        call    cl_pc
        mov     pcbit,#7    ;pc7=1
        call    st_pc
        call    y70_d1

```

增加這一行

```

        call    rd_pb      ;read pb,data store to pb55
        call    data_pro   ;data processing
;right      ;74139=>q3=0
        mov     pcbit,#6   ;pc6=1
        call    st_pc
        mov     pcbit,#7   ;pc7=1
        call    st_pc
        call    y70_d1
        call    rd_pb      ;read pb,data store to pb55
        call    data_pro   ;data processing
;
        ret
;-----data processing
data_pro:
        anl     pb55,#0f0h   ;pb4-pb7
        mov     a,pb55       ;store to cny70
        cjne    a,#0f0h,show0 ;display sensors status
        mov     abuf,'#1'
        call    wr_lcddat    ;黑底顯示 1
        ret
show0:
        mov     abuf,'#0'
        call    wr_lcddat    ;白底顯示 0
        ret
;-----sensor hold delay time
y70_d1:
        mov     tdl70,#250
        djnz    tdl70,$
        ret

```

步驟三：修改 OPS_menu 選單顯示副程式內容

```

OPS_menu:
        mov     dptr,#OPS_scr1
        call    sho_lcm1
        mov     dptr,#scr2
        call    sho_lcm2
OPS_scan:
        call    scan_y70
        call    scan_key
        mov     a,keynum
        cjne    a,#0ah,OPS_scan      ;判斷是否為 Menu 鍵

```

ret

到此 OPS 功能已經完成，詳細內容可參考程式 5-41.a 或 5-42.a