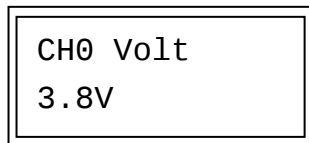


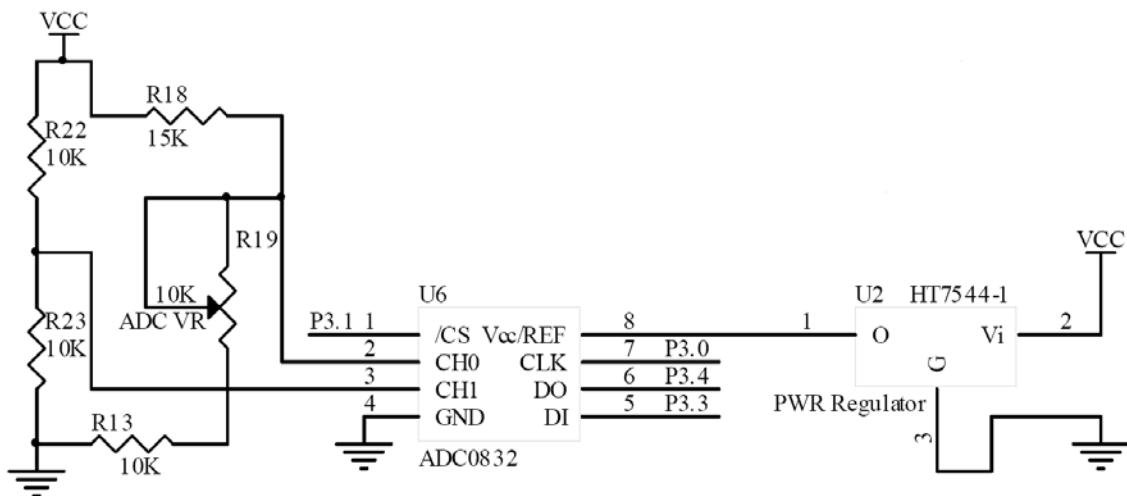
A. 題目

被指定到本題者必須在 LCM 進階選單的選項 3 上呈現出 "ADC" 等文字，應檢者必須利用週邊板上編號 U6 的 ADC0832 串列式 IC 將 CH0 接腳上的類比電壓 (電壓大小可由 10K 可變電阻來改變) 轉換成數位資料，由微控制器讀取處理後在 LCM 上顯示出電壓值 (到小數點第一位)。當系統在進階選單狀態時，若操作者按下數字鍵 "3" 則進入 ADC 功能，此時在 LCM 第一列的畫面隨即顯示出 "CH0 Volt" 等文字，而在第二列顯示出 CH0 接腳上的電壓值 (到小數點第一位，例如 3.8V)，由於連接到 ADC0832 的工作參考電壓值為 4.4V，應試者必須參考研究硬體電路圖之後，將轉換所得到的數位資料換算出真實的電壓值，再將正確數字顯示在 LCM 上，誤差的電壓值必須小於 0.2V (含) 以內 (加減 0.2V)，關於類比數位轉換功能的參考畫面如下所示：



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：

此電路 (參考題本第 15 頁) 中，HT7544 的功能在穩壓，產生 4.4V 的固定電壓輸出，做為 ADC0832 電源電壓和參考電壓，當輸入 CH0 或 CH1 輸入的類比電壓在 0~4.4V 間時，轉換後的數位值為 0~255，每變化一格所代表的電壓 = $\frac{4.4}{255} = 0.0173V$ 。在電路中，CH0 的輸入透過 2 個 10K 歐姆的電阻分壓，電壓 = $\frac{1}{2}V_{CC}$ ，所以轉換後的數位值應該是 128，CH0 的輸入透過 15K 電阻、10K 可變電阻和 10K 電阻分壓，所以 CH0 的輸入電壓變化如下：

$$\frac{10}{15+10+10}V_{CC} \leq CH0\text{電壓} \leq \frac{10+10}{15+10+10}V_{CC}$$

$$0.286V_{CC} \leq CH0\text{電壓} \leq 0.571V_{CC}$$

以 $V_{CC} = 5\text{ V}$ 為例(注意此 V_{CC} 為電源供應器的電壓，不是 ADC0832 的電源電壓)

$$1.4 \leq CH0\text{ 電壓} \leq 2.8 \quad (\text{只要顯示的數字不在這個範圍內，答案應該就是錯的})$$

HT7544-1, +4.4V Output Type

$T_a=25^\circ\text{C}$

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{IN}	Conditions				
V_{OUT}	Output Voltage Tolerance	6.4V	$I_{OUT}=10\text{mA}$	4.268	4.4	4.532	V
I_{OUT}	Output Current	6.4V	—	60	100	—	mA
ΔV_{OUT}	Load Regulation	6.4V	$1\text{mA} \leq I_{OUT} \leq 50\text{mA}$	—	60	150	mV
V_{DIF}	Voltage Drop	—	$I_{OUT}=1\text{mA}$	—	100	—	mV
I_{SS}	Current Consumption	6.4V	No load	—	2.5	5	μA
$\frac{\Delta V_{OUT}}{\Delta V_{IN} \times V_{OUT}}$	Line Regulation	—	$5.4\text{V} \leq V_{IN} \leq 24\text{V}$ $I_{OUT}=1\text{mA}$	—	0.2	—	%/V

HT7544 依照的資料手冊使用說明，要產生 4.4V 的穩定輸出，輸入電壓至少要比 4.4V 大一些，不然輸出電壓會小於 4.4V，請以實際量到的為準。

假設 $V_{CC} = 4.4\text{V}$

$$\text{每變化一格所代表的電壓} = \frac{4.4}{255} = 0.0173\text{V}$$

假設後 ADC 轉換後得到的數值為 n ，所代表的電壓為 x ，則 n 和 x 間的關係式

$$\frac{x}{n} = \frac{4.4}{255}$$

$$x = 0.0173\ n \quad (1)$$

若讀到的值是 n ，其實際的電壓 $x = 0.0173\ n$

因為 8051 組合語言不支援小數點的乘法，只提供 MUL AB 這個整數乘法指令

A	×	B	=	B		A
被乘數		乘數		高位元組		低位元組

在此先把(1)式兩邊時乘上 256，可得

$$256 \times 0.0173 \times 256 n \approx 4.4 n \quad (2)$$

為什麼要乘上 256？主要是希望乘完的結果會比原來大上 256 倍，捨去掉累加器 A(低位元組)那 8 位元數值，相當於除以 256，兩者互相抵消，此時留在的高位元組 B 內的數值就是我們所要的整數部分電壓，捨去的累加器 A 內的值，其實就是小數部份的電壓。

但是如果拿 4.4 去乘 n，4.4 仍然是帶有小數，可考慮兩邊再乘上 10 倍，(2)式兩邊時乘 10，可得

$$256 \times 10 \times = 44 n \quad (3)$$

此時留在累加器 A 的數值就是就是整數部分加小數點以下的一位數的電壓，如果這樣還不清楚，可以下範例為例：

假設 ADC0832 讀到的數值是 109 (01101101)

```
mov    a, #109
mov    b, #44
mul    ab
```

計算完的結果 b = 18，a = 188 (10111100)

對照(1)式

$$x = 0.0173 n = 0.0173 \times 109 = 1.8857 V$$

可知 b 的數值 18 代表 1.8V，捨掉的 A 數值 188，如果除以 256(滿 256 進位)，等於 0.734，也跟小數點第二位以後的數值 857 很接近，會有誤差，主要是當初(2)式中的 4.4，本身就是近似的結果，如果要再準一點，可考慮取四捨五入，判斷 a 值是否超過一半，如果 a 的最高位元是 1，可把 B 的值加 1，例如在此例中，a = 188(10111100)，所以 B 加 1 等於 19，LCM 顯示 1.9V。判斷程式寫法如下：

```
        jnb    acc.7, a1      ;判斷累加器最高位元是不是 1，如果不是跳至 a1
        inc    b              ;b 暫存器加 1
a1:     mov    a, b
```

因為考場題目只要求誤差小於±0.2V，所以以上 3 行程式可以不要背，不判斷四捨五入誤差並不大。

想把計算好的整數和小數點以下一位數電壓分開，則可以透過除法，將數值除以 10

```
mov    a, b
mov    b, #10
div    ab      ;執行 a/b = a(商) ... b(餘數)
```

若當初 $b = 19(1.9V)$ ，則執行完 $a = 1, b = 9$ ，可再分別將 a 和 b 的數值加上 $30H$ ，轉成 ACIII 碼，就可以顯示在 LCM 上。

B. 參考程式：(原始檔名 adc0832.a)

參考程式位於 ADC-adc0832 資料夾內的 adc0832.a

```
org 00h
;-----register define-----
s0832_cs      equ p3.1      ;low status
s0832_di      equ p3.3      ;low status
s0832_do      equ p3.4      ;high status
s0832_clk     equ p3.0      ;low status
;-----
temp_0832     equ 0x77      ;adc converter data
count_0832    equ 0x76
;-----

mov    p2,#0xff    ;if p2=>8 leds

clr s  0832_di     ;initial status clr
setb   s0832_cs    ;initial status setb
setb   s0832_do    ;initial status setb(notice)
clr s  0832_clk    ;initial status clr
;

call   wr_opcode   ;寫控制命令到 ADC0832
call   read_adc    ;讀取 8 位元轉換結果
mov    p2,temp_0832 ;OUTPUT DATA 資料輸出到 p2
end_0832:
jmp    end_0832    ;無窮回圈

;
;=====write operate code
wr_opcode:    ;opcode=110,start bit、sgl、/sign
    clr      s0832_cs    ;start bit=1
    setb     s0832_di
    setb     s0832_clk
    clr      s0832_clk
    clr      s0832_di
;
;                ;single-ended model
    setb     s0832_di
    setb     s0832_clk
    clr      s0832_clk
    clr      s0832_di
;
;                ;select channel-0
    clr      s0832_di
    setb     s0832_clk
```

暫存器定義區

初始設定

從這裡開始是副程式，以下全部複製到考試程式的副程式區

```

        clr    s0832_clk
        setb   s0832_di
        ret
;=====read adc data
read_adc:
        mov    count_0832,#8
        clr    a
adcbt:
        setb   s0832_clk
        clr    s0832_clk ;bit data output
        nop
        mov    c,s0832_do
        rlc    a
        djnz   count_0832,adcbt
        mov    temp_0832,a
        setb   s0832_cs
        ret
;*****
;++++++numerical transformation;256/3.2=80,data/80
        end

```

C. 類比數位轉換程式設計說明：

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

```
s0832_cs      equ p3.1      ;low status
s0832_di      equ p3.3      ;low status
s0832_do      equ p3.4      ;high status
s0832_clk     equ p3.0      ;low status
;-----
temp_0832     equ 0x77      ;adc converter data
count_0832    equ 0x76
```

暫存器定義區

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

步驟三：複製以下幾行到副程式區開頭，

(這幾行程式的功能在控制 ADC0832 讀取電壓，並將轉換結果存在暫存器 temp_0832)

```
clr s 0832_di ;initial status clr
setb s0832_cs ;initial status setb
setb s0832_do ;initial status setb(notice)
clr s 0832_clk ;initial status clr
;
call wr_opcode ;寫控制命令到 ADC0832
call read_adc ;讀取 8 位元轉換結果
mov p2,temp_0832 ;OUTPUT DATA ;將讀到的值以二進制方式顯示在 p2，
;這裡要將它改成顯示在 LCD
```

修改成以下副程式，並命名為 scan_ADC

scan_ADC:

```
call line2 ;游標移到第二行開頭
clr s 0832_di ;initial status clr
setb s0832_cs ;initial status setb
setb s0832_do ;initial status setb(notice)
clr s0832_clk ;initial status clr
;
call wr_opcode
call read_adc
```

```

mov    a,temp_0832    ;將讀取到的轉換結果複製到累加器 A 做計算
mov    b,#44          ;44 為 ADC0832 的 Vcc 電壓 4.4V，可參考前面說明
mul    ab              ;做數值換算
mov    a,b
mov    b,#10
div    ab              ;執行 a/b = a(商) ... b(餘數)
add    a,#30h          ;將整數部份轉成 ASCII 碼
mov    abuf,a
call   wr_lcddat       ;寫入 LCM
mov    abuf,#'.'
call   wr_lcddat       ;顯示小數點'.'
mov    a,b
add    a,#30h          ;將小數部份轉成 ASCII 碼
mov    abuf,a
call   wr_lcddat       ;寫入 LCM
mov    abuf,#'V'
call   wr_lcddat       ;顯示電壓符號'V'
ret

```

左邊兩個框內文字幾乎相同，直接複製過來就可以了，這段程式可參考數字輸入 NI 那題

步驟四：修改 ADC_menu 選單顯示副程式內容

ADC_menu:

```

mov    dptr,#ADC_scr1
call   sho_lcm1
mov    dptr,#scr2
call   sho_lcm2

```

ADC_scan:

```

call   scan_ADC        ;掃描 ADC，並將結果顯示於 LCM
call   scan_key
mov    a,keynum
cjne   a,#0ah,ADC_scan ;判斷是否為 Menu 鍵
ret

```

到此 ADC 功能已經完成，詳細內容可參考程式 5-5.a