

Lezione 5

In questa sezione del corso, impareremo a pilotare display a sette segmenti. Come tutti sappiamo, la soluzione di inserire un display a sette segmenti per una qualsiasi indicazione, è, fra le tante possibili, una delle meno dispendiose e di sicuro effetto. Quando, alcuni anni fa, non si impiegavano i microcontroller per le piccole e medie applicazioni, per gestire i display erano necessari integrati dedicati, come quello visto lo scorso mese, in grado di convertire un numero binario in codifica a sette segmenti. In alcuni casi, venivano programmate delle PAL o delle GAL, per ottenere anche un minimo di ottimizzazione delle porte logiche. Con l'avvento dei microcontroller, gestire un display a sette segmenti è diventato semplicissimo, e, come vedremo, anche più di uno.

Controllo di un display

In figura 1, troviamo lo schema elettrico necessario per poter eseguire esperimenti con un display. Da notare che i numeri sui pin del display sono in successione come le corrispondenti lettere dei segmenti, ovvero il numero 1 corrisponde alla lettera "A", il numero 2 alla "B" e così via fino alla "F". Il punto decimale non è stato utilizzato.

Il display ha il positivo in comune, mentre i sette segmenti si accendono portandoli verso la massa. Le resistenze da 220 Ohm riducono l'assorbimento di ogni segmento. Il PIC quindi, per accendere uno dei segmenti, dovrà settare a 0 il pin corrispondente.

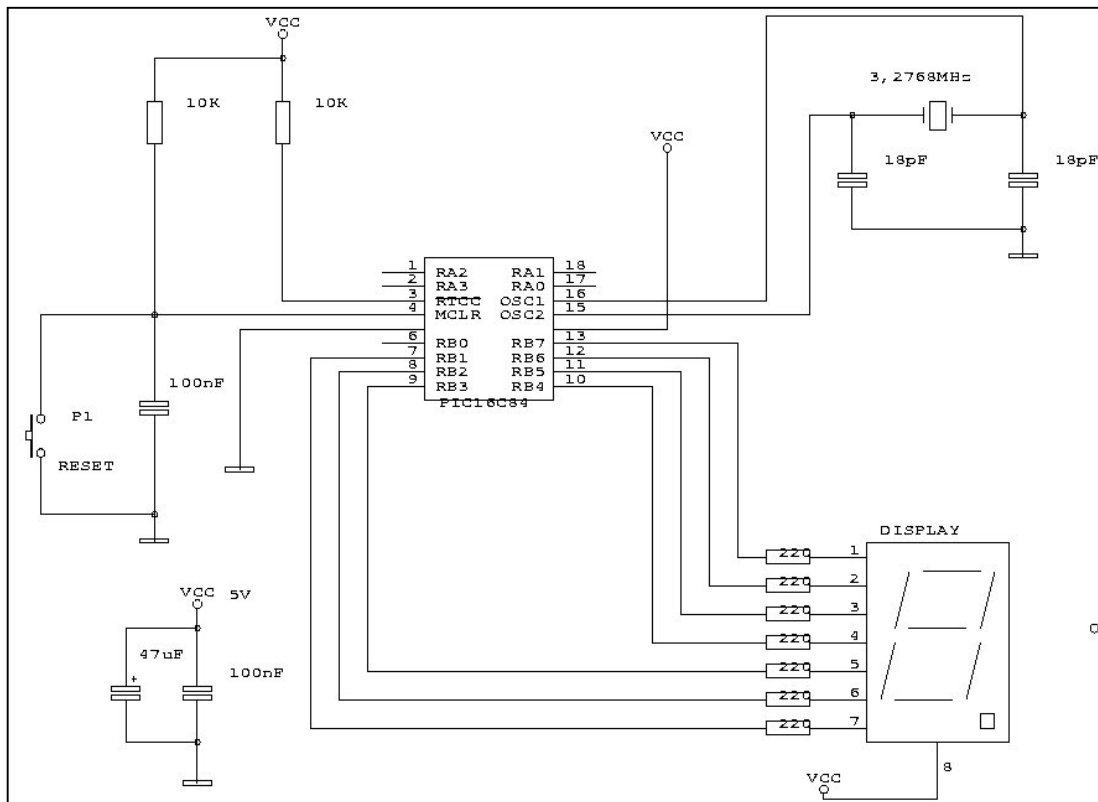


Figura 1. Schema elettrico per display a 7 segmenti

Il programma PROG08 visualizza i numeri da 0 a 9, in crescendo, con passi di un secondo.

```
TITLE 'PROG08: Controllo display 7 segmenti'
list F=INHX8M,P=16C84
;-----
RTCC EQU 01H ; Real Time Clock Counter
PCL EQU 02H ; Program Counter
STAT EQU 03H ; Registro di stato
PORTA EQU 05H ; Porta A
PORTB EQU 06H ; Porta B
INTCON EQU 0BH ; Registro abilitazione interrupt
TR_A EQU 85H ; Tris A
TR_B EQU 86H ; Tris B
OPTIO EQU 81H ; Registro OPTION
;-----
FR01 EQU 0CH ; Ritardo 1S
FR02 EQU 0DH ; Numero da visualizzare
;----- Definizione costanti per display -----
ZER_O EQU 003H ; Codifica numero 0: 00000011
UNO EQU 09FH ; Codifica numero 1: 10011111
DUE EQU 025H ; Codifica numero 2: 00100101
TRE EQU 00DH ; Codifica numero 3: 00001101
QUATT EQU 099H ; Codifica numero 4: 10011001
CINQUE EQU 049H ; Codifica numero 5: 01001001
SEI EQU 041H ; Codifica numero 6: 01000001
SETTE EQU 01FH ; Codifica numero 7: 00011111
OTTO EQU 001H ; Codifica numero 8: 00000001
NOVE EQU 009H ; Codifica numero 9: 00001001
;-----
ORG 0
goto START ;Reset vector
nop
nop
nop
goto INTERP ;Interrupt vector

;----- subroutine RITARDO 5 mS -----
; con frequenza 3,2768 MHz
;-----
DELA5M clrwdt ;
movf RTCC,0 ; Controllo se finiti 5mS
SKPZ ;
goto DELA5M ;
movlw .128 ;
movwf RTCC ;
return ; Ritorna dopo la CALL
;----- subroutine presentazione numero su display -----
DISPLA movf FR02,0 ; Copio FR02 in w
addwf PCL,1 ; Sommo w al program counter
retlw ZER_O ; Visualizza 0
retlw UNO ; Visualizza 1
retlw DUE ; Visualizza 2
retlw TRE ; Visualizza 3
retlw QUATT ; Visualizza 4
retlw CINQUE ; Visualizza 5
```

```

    retlw    SEI           ; Visualizza 6
    retlw    SETTE        ; Visualizza 7
    retlw    OTTO         ; Visualizza 8
    retlw    NOVE         ; Visualizza 9
;-----
; START PROGRAM
;-----
START    bsf        STAT,5      ; Seleziona SRAM banco 1
        movlw     b'0000'      ; RA out
        movwf    TR_A         ;
        movlw     b'00000000'   ; RB out
        movwf    TR_B         ;
        clrf     INTCON        ; Disabilita interrupt
        movlw     b'11000100'   ; Prescaler 1:32
        movwf    OPTIO        ; Copia W in OPTION
        bcf      STAT,5        ; Seleziona SRAM banco 0
        movlw     .128         ; Settaggio iniziale RTCC
        movwf    RTCC         ;
        movlw     b'0000'      ; Porta A off
        movwf    PORTA        ;
        movlw     b'11111111'   ; Disabilita segmenti display
        movwf    PORTB        ;
;----- main program -----
MAIN     clrf      FR02         ; Numero iniziale
MAIN1    call     DISPLA       ; Visualizza numero
        movwf    PORTB        ; su porta B
        movlw     .200         ; Conteggi da 5mS per 1S
        movwf    FR01         ;
LOOP     call     DELA5M       ; Subroutine ritardo 5mS
        decfsz   FR01         ; Testo se scaduto 1S
        goto     LOOP         ;
        incf     FR02         ; Incremento numero da visualizzare
        movlw     .10          ; Copio 10 in W
        xorwf    FR02,0       ; Testo se FR02 = 10
        btfss   STAT,2        ;
        goto     MAIN1        ; <> 10
        goto     MAIN         ; = 10
;-----
INTERP   ; Vettore interruzioni
        retfie   ;
;-----
        END

```

Dato che il programma ha le stesse inizializzazioni dei precedenti, ci soffermeremo soltanto sulle righe che introducono delle novità. La prima consiste nell'aver ridotto il fattore di divisione del prescaler da 128 a 32. In questo modo, otterremo attese di 5mS precisi. In più sono state eliminate le resistenze di pull-up sulla porta B, visto che questa serve solo da output.

Il numero da visualizzare viene mantenuto in FR02 ed inizialmente vale zero. Nel ciclo MAIN1, si richiama la subroutine DISPLA, che torna la codifica da copiare sulla porta B per visualizzare il numero contenuto in FR02 sul display. Vediamo come funziona: DISPLA copia il contenuto del registro FR02 in W, poi aggiunge tale numero al program counter. In questo modo, l'istruzione successiva, non sarà più quella immediatamente dopo, ma sarà calcolata sommando il numero da visualizzare al program counter.

Vediamo un esempio pratico supponendo che il numero contenuto nel registro FR02 sia "3". Allora, in W viene copiato un 3. L'istruzione ADDWF PCL,1 somma al program counter un 3. Poichè il program counter è il registro che punta sempre alla prossima istruzione da eseguire, questo significa che l'istruzione successiva non sarà quella immediatamente dopo la ADDWF, ma la quarta (la prima se avessi sommato 0). Questa istruzione fa uscire dalla subroutine caricando il valore corrispondente alla label TRE nel registro W. Al ritorno dalla subroutine, l'istruzione successiva è la MOVWF PORTB, che copierà il valore contenuto in w sulla porta B, facendo accendere i led corrispondenti al numero 3.

Le label da ZER_O a NOVE infatti, decodificano il numero che rappresentano nei sette segmenti del display. La label UNO, ad esempio, pone a zero il bit 5 ed il bit 6, corrispondenti proprio ai due segmenti B e C relativi al numero "1".

In questo modo, possiamo modificare il valore sul display in qualsiasi momento, ad esempio facendo corrispondere allo zero la lettera "A", all'uno la lettera "b" e così via, compatibilmente con la disponibilità del display. Quello che abbiamo ora visto, è il metodo più semplice ed efficace per implementare tabelle con i controller della Microchip.

Ma vediamo ora un'altra serie di istruzioni che confrontano due valori: abbiamo necessità di sapere quando il numero dei conteggi è giunto a 9 e successivamente a 10, poichè in questo caso dobbiamo resettare il contatore ripartendo da zero.

Dopo aver incrementato il contatore, copiamo il numero 10 nel registro W, poi ne facciamo lo XOR con il registro FR02, stando attenti a collocare il risultato in W (perchè se lo mettessimo in FR02, il contatore verrebbe corrotto perdendo il suo valore originale). Quello che ci interessa poi, è andare a vedere se il risultato è uguale o diverso da 0. Nel primo caso, il registro FR02 è caricato con il valore 10, nel secondo con un numero diverso da 10. Quindi, nel primo caso si resetta il contatore, nel secondo si prosegue con la visualizzazione del numero e successiva attesa di un secondo.

-continua.