

## Afisaj 4x7 segmente CBM 557

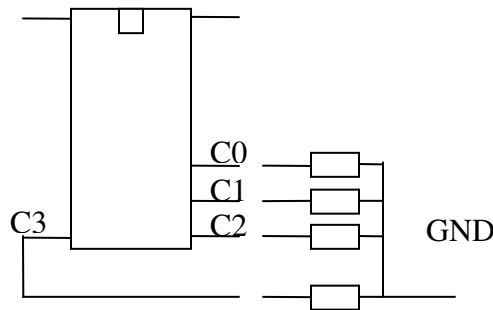
In functie de configuratie poate dispune de urmatoarele resurse hard:

- RS485
- 8 leduri
- 4 rele
- 1 intrare ADC
- butoane

Poate fi comandat prin RS485 de catre un automat programabil sau poate functiona independent, caz in care pe cele 4 cifre cu 7 segmente se va afisa valoarea de pe intrarea ADC.

Adresele pe RS485 rezervate afisajelor sunt de la D0÷DE, pentru afisajele care functioneaza impreuna cu un automat programabil. Pentru afisajele care se doresc a functiona independent se va atribui adresa DF.

Modul in care se stabileste adresa afisajului:



Cei 4 pini stabilesc adresa afisajului in felul urmatoar:

$$\text{adresa\_afisaj} = \text{D0h} + \text{C3C2C1C0}.$$

Un pin este 1 daca este conectat la masa prin rezistenta si 0 daca nu.

Astfel pentru afisajul independent vom avea toti pini conectati la masa iar adresa va fi:

$$\text{D0h} + \text{1111b} = \text{D0h} + \text{Fh} = \text{DFh}$$

Configurarea afisajului de catre automatul programabil se face in urmatoarul fel:

- se declara urmatoarele variabile de comunicare cu afisajul:  
ex. (cazul in care sunt 2 afisaje legate la automat prin RS485)

word:

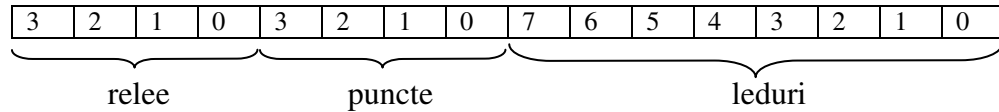
```
leduri1      down  addr_afisaj1
leduri2      down  addr_afisaj2
but1         up    addr_afisaj1
adc1         up    addr_afisaj1
but2         up    addr_afisaj2
adc2         up    addr_afisaj2
```

long:

```
num1 down  addr_afisaj1
num2 down  addr_afisaj2
```

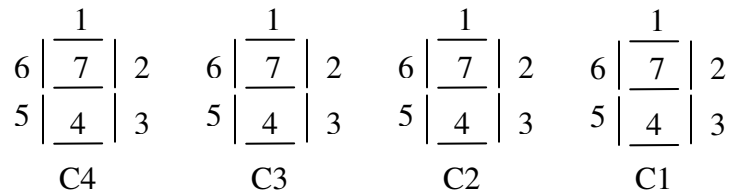
...

- variabila “**leduri**” controleaza cele 8 leduri, iesirile de relee(care se comanda si prin ADC) si punctele zecimale de pe afisajul cu segmente. Structura variabilei este urmatoarea :



- variabila “**num**” reprezinta numarul de afisat pe cele 4 cifre a cate 7 segmente. Pentru afisajul independent(fara automat programabil) acest numar va indica valoarea ADC-ului de pe afisaj. Pentru celelalte afisaje (cu automat programabil) el poate indica numere intre [-999;9999]. Daca se depasesc aceste limite exista urmatoarele cazuri:

1.  $9999 < \text{numar} \leq 65535$  – se va afisa “Err”
2.  $-65536 \leq \text{numar} < -999$  - se va afisa “-Err”
3.  $\text{numar} > 65536$  – se acceseaza segmentele de pe afisaj in felul urmatoare:



Valorile pentru care se acceseaza segmentele de pe afisaj sunt date in urmatoarul tabel:

obs:  $A = 2^{31} + 2^{15}$

Cifra	Valoare	Actiune
-	$2^{31}$	sterge tot afisajul
C1	$A + 2^0$	segment 1
C1	$A + 2^1$	segment 2
C1	$A + 2^2$	segment 3
C1	$A + 2^3$	segment 4
C1	$A + 2^4$	segment 5
C1	$A + 2^5$	segment 6
C1	$A + 2^6$	segment 7
C2	$A + 2^8$	segment 1
C2	$A + 2^9$	segment 2
C2	$A + 2^{10}$	segment 3
C2	$A + 2^{11}$	segment 4
C2	$A + 2^{12}$	segment 5
C2	$A + 2^{13}$	segment 6
C2	$A + 2^{14}$	segment 7
C3	$2^{16}$	segment 1
C3	$2^{17}$	segment 2
C3	$2^{18}$	segment 3
C3	$2^{19}$	segment 4
C3	$2^{20}$	segment 5
C3	$2^{21}$	segment 6



doreste a fi modificat(daca este una din primele 2 comenzi). Daca nu atunci nu se introduce nimic in variabila "num". Comanda care se doreste a fi executata se introduce in variabila "led", variabila care reprezinta ledurile.

Dupa introducerea unei comenzi trebuie sa treaca un anumit timp pana cand putem considera ca, comanda s-a executat. Acest timp este indeplinit printr-un ciclu de felul urmatoar:

```
wait:
    if(led<>but) jmp wait.
```

- pentru **comandarea iesirilor de rele** avem cate 2 comenzi pentru fiecare releu in parte, una dintre ele modificand tensiunea de limita pentru releu, iar cealalta modificand tensiunea de histereza pentru releu.

Comanda	0hBCD1	0hBCD2	0hBCD3	0hBCD4	0hBCD5	0hBCD6	0hBCD7	0hBCD8
Semnificatia	$U_{lim}$	$U_{hist}$	$U_{lim}$	$U_{hist}$	$U_{lim}$	$U_{hist}$	$U_{lim}$	$U_{hist}$
Releul	0	0	1	1	2	2	3	3

Pentru comenzile de la iesiri se procedeaza la fel ca si la ADC.

Comanda releelor conectate din exterior la iesiri se face prin cate un tranzistor care pentru starea "releu actionat " conduce spre "+5V".

In continuare vom da un exemplu de configurare a ADC-ului si a iesirilor de rele:

ex: ...

*word:*

```
leduri down d0
but up d0
adc up d0
```

*long:*

```
numar down d0
```

...

*code segment*

```
call config_adc ; procedura de configurare a parametrilor P0 si Coef
call config_relee ;procedura de configurare a pragurilor "Ulim" si "Uhist"
;pentru relee
```

...

*end segment*

*proc segment*

**proc config\_adc**

```
numar=125 ; se introduce in numar valoarea dorita pentru parametrul "Coef"
leduri=0hBCDA ; se da comanda pentru scrierea valorii din numar pentru
;parametrul "Coef"
```

*wait1:*

```
if(leduri<>but) jmp wait1 ;se asteapta sa se scrie valoarea
numar=0 ; se introduce in numar valoarea dorita pentru parametrul "P0"
leduri=0hBCDB ; se da comanda pentru "P0"
```

*wait2:*

```
if(leduri<>but) jmp wait2 ;se asteapta sa se scrie valoarea
```

```

call scrie_valori
ret
proc config_relee
numar=3000      ; "Ulim"
leduri=0hBCD1  ; comanda pentru scriere "Ulim" pentru releul 0
wait3:
if(leduri<>but) jmp wait3    ;se asteapta sa se scrie valoarea
numar=0        ; "Uhist"
leduri=0hBCD2  ; comanda pentru scriere "Uhist" pentru releul 0
wait4:
if(leduri<>but) jmp wait4    ;se asteapta sa se scrie valoarea
call scrie_valori
ret
proc scrie_valori
leduri=0hBCDE  ; comanda de pregatire scriere valori in FLASH
wait5:
if(leduri<>but) jmp wait5    ;asteptam
leduri=0hBCDF  ; comanda de scriere valori in FLASH
wait6:
if(leduri<>but) jmp wait6    ;asteptam
ret
end segment

```