

■ Introduction :

Les fonctions d'entrées-sorties sont des fonctions de l'API du langage de programmation permettant de réaliser des entrées (lecture de données) et des sorties (écriture de données) depuis / vers un flux d'entrée-sortie (console, fichier, socket, ...).

Par exemple :

- lire depuis le clavier, écrire sur l'écran,
- lire depuis un fichier, écrire dans un fichier,
- ...

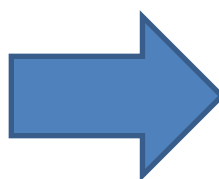
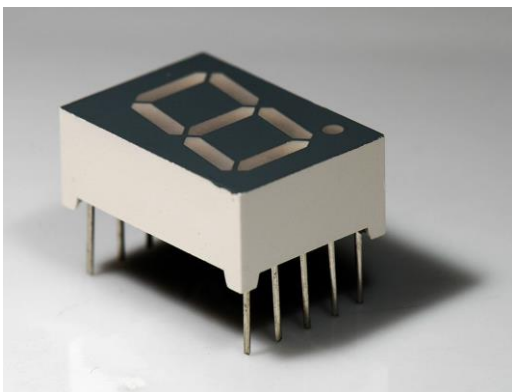
C'est pour faciliter les entrées et les sorties de données que des interfaces sont programmées. Elles doivent être simples et lisibles.

■ Partie I :

• LEDs à cathodes communes :

Dans les calculatrices, les jeux électroniques... il est nécessaire d'afficher les chiffres de 0 à 9. On utilise très souvent des afficheurs 7 segments.

- Afficheur 7 segments :



L'afficheur est composé de 7 LEDS (segments) notées a, b, c, d, e, f et g qui nécessitent, en fonction du type d'afficheur (anode commune ou cathode commune) une polarisation spécifique. Chaque segment peut avoir deux états :

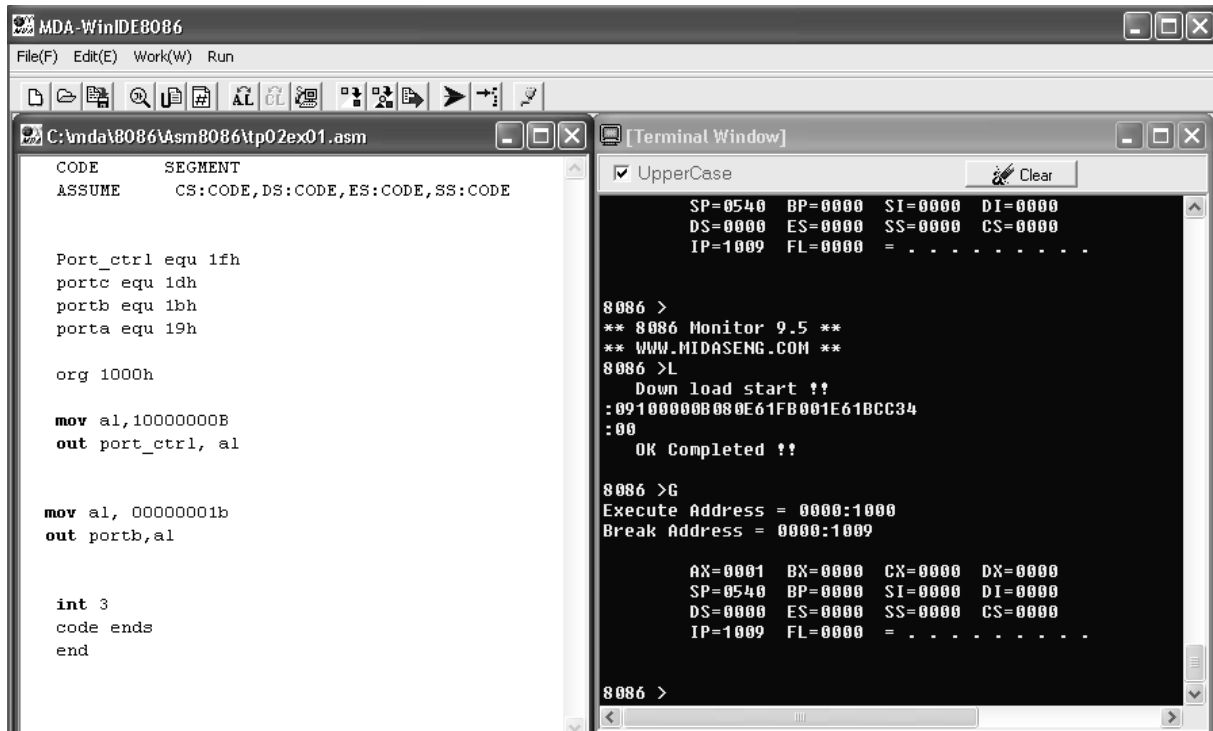
- 1 (allumé),
- 0 (éteint).

Les chiffres de 0 à 9 sont codés sur 4 bits A,B,C et D avec A bit de poids le plus faible et D bit de poids le plus fort (ex : 9 s'écrit DCBA=1001 en binaire).

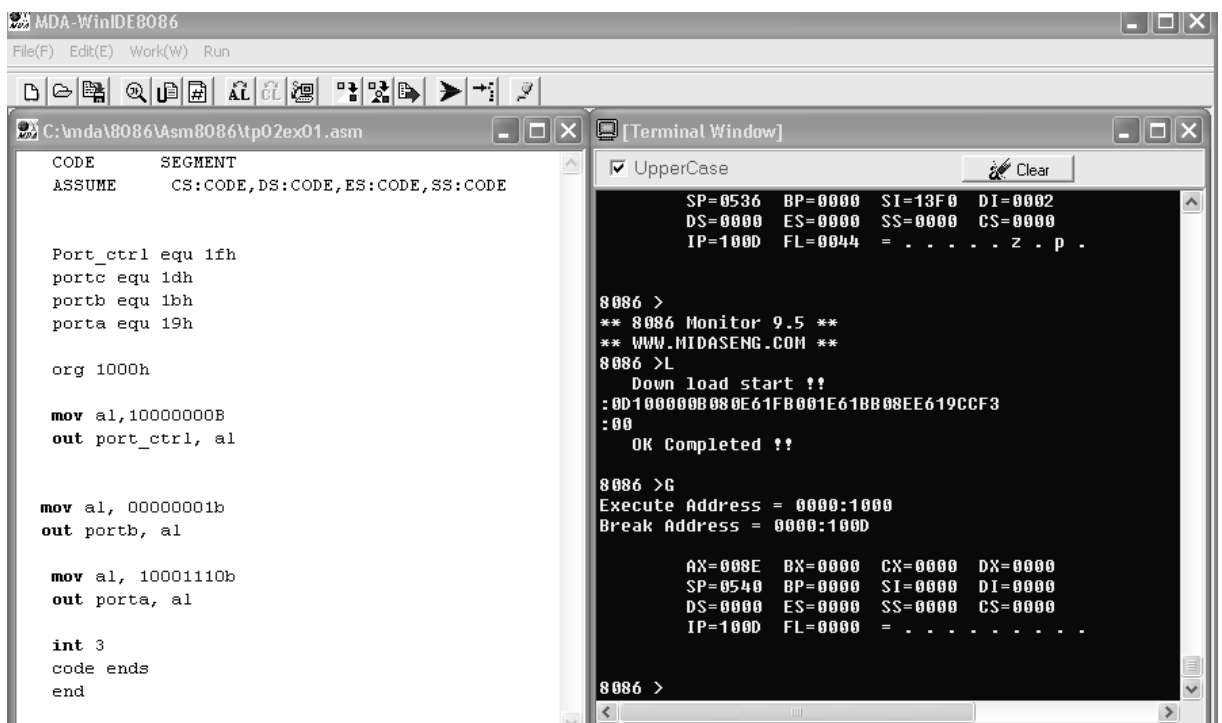
L'objectif du travail suivant est de réaliser le circuit (transcodeur) permettant de convertir un chiffre sur 4 bits en un affichage. Ceci revient à déterminer une fonction de A, B, C, D pour chaque segment a, b ,c ,d ,e ,f et g.

■ Travail demandé :

a- Le programme qui allume une des 0 LEDs a cathodes communes :



b- Le programme qui écrire sur l'afficheur 7 segments un des chiffres hexadécimaux :



■ Parti II :

• Présentation et définition :

On appelle couramment un afficheur à points, un DOT MATRIX ou alors un flip à DOT.

Pourquoi, en fait le système ici régit différemment les afficheurs, avant les afficheurs étaient limités de par leur possibilités d'affichage de Gadgets, quoi que, ils ce sont déjà bien amusés à faire des belles choses en Alphanumérique. Ouf voici qu'arrive l'afficheur à points, complexe, mais permet d'avoir des animations qui sont vraiment intéressantes (si l'on ne joue pas au flip, car en jouant on peut pas tout faire J).

Ici on est en fait en présence d'un système qui allume un point en faisant une intersection entre une ligne et une colonne.

Pour commander en fait électroniquement ceci , on commande la ligne de l'afficheur ainsi que sa colonne et on y applique une tension différente pour les deux, ce qui fait lors du balayage des lignes et de la persistance rétinienne que l'on voit un point ou un trait, qui est le résultat de l' ionisation du gaz .



■ Travail demandé :

a- le programme qui allume toutes la matrice en vert :

```
code    segment
assume  cs:code, ds:code, es:code,  ss:code
port_ctrl equ 1eh
portc equ 1ch
portb equ 1ah
porta equ 18h
org 1000h;
mov al,10000000b
out port_ctrl,al

mov al,11111111b
out portb,al
mov al,00000000b
out porta,al
mov al,11111111b
out portc,al
code    ends
end
```

b- le programme qui allume la5 em colonne en rouge :

```
code    segment
assume  cs:code, ds:code, es:code,  ss:code
port_ctrl equ 1eh
portc equ 1ch
portb equ 1ah
porta equ 18h
org 1000h;
mov al,10000000b
out port_ctrl,al

mov al,00000000b
out portb,al
mov al,11111111b
out porta,al
mov al,00010000b
out portc,al
code    ends
end
```

c- le programme qui allume la 4 em ligne en orange :

```
code    segment
assume  cs:code, ds:code, es:code,  ss:code
port_ctrl equ 1eh
portc equ 1ch
portb equ 1ah
porta equ 18h
org 1000h,
mov al,10000000b
out port_ctrl,al

mov al,11101111b
out portb,al
mov al,11101111b
out porta,al
mov al,11111111b
out portc,al
code    ends
end
```