

TP 5 : programmation d'une application faisant intervenir l'interface parallèle**Le format des programmes rédigés en assembleur**

L'écriture des programmes est réalisée avec un outil fourni avec les cartes de développement *figure-II-3*. L'outil utilise MASM de Microsoft comme assembleur et Debugger. Les programmes assemblés par MASM sous la forme de fichiers objet (représentation binaire des instructions), sont convertis en fichiers *ABS* et transférés sur la mémoire vive de la carte de développement via une interface série RS232.

Nous avons expliqué préalablement que l'espace mémoire est de 64 ko, la limite imposée pour un segment. Ceci nous pousse à combiner sur la taille d'un seul segment les trois types : Code, Données et pile, l'ensemble ne peut dépasser la capacité de 64ko, ce qui oblige la définition d'un seul segment au niveau de l'assembleur.

```

-*****
;
; TP N°-- du --/--/---
; Titres du Travail Pratique, Exercice N°--
; Matière systemes embarques
-*****
; Début de définition du segment dit BK
BK          SEGMENT
    ASSUME  CS:BK,DS:BK,SS:BK
;
    ORG 1000H
;
; Instructions
;
BK  ENDS ; Fin du segment BK
END

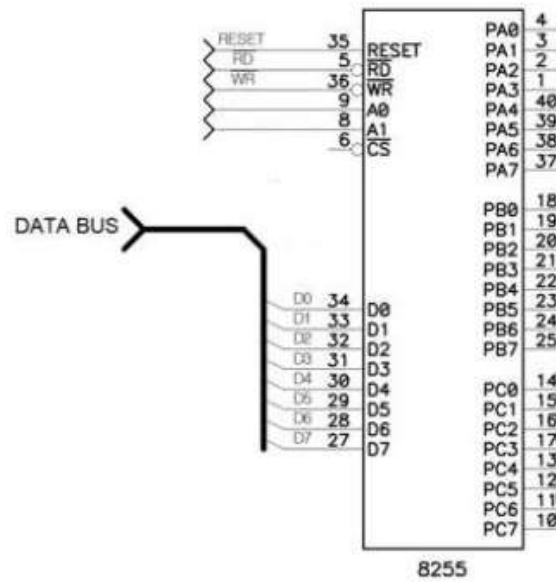
```

Les programmes doivent respecter :

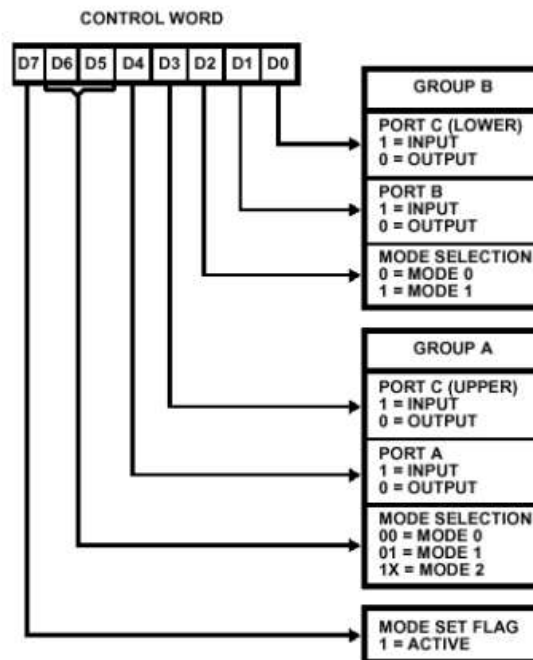
- *ASSUME CS:BK,DS:BK,SS:BK*: informe l'assembleur que les registres CS, DS et SS pointent à l'adresse du segment créé derrière l'identificateur *BK*.
- *ORG 1000H* : Une directive destinée à l'assembleur qui indique l'adresse mémoire de départ pour la séquence de code qui suit. Dans notre cas la première instruction exécutée. Cette valeur de l'adresse est imposée par le Firmware de la carte de développement.

Présentation de l'interface parallèle

L'interface parallèle intervient dans le cadre d'un échange simultané de plusieurs bits, un octet en général. Ce type d'interface présente l'avantage de la rapidité des transferts et facilité de la programmation. Chaque interface parallèle avant son utilisation doit être configurée en entrée, en sortie, bidirectionnelle, avec ou sans signaux de contrôle. Dans la carte de développement que nous utilisons, l'interface parallèle est matérialisée par le circuit 8255A (*figure-II-5*). Ce dernier possède trois ports de un octet chacun : PortA, PortB et PortC. Le mode de fonctionnement de l'interface parallèle est assuré via un registre de commande (*figure-II-6*)



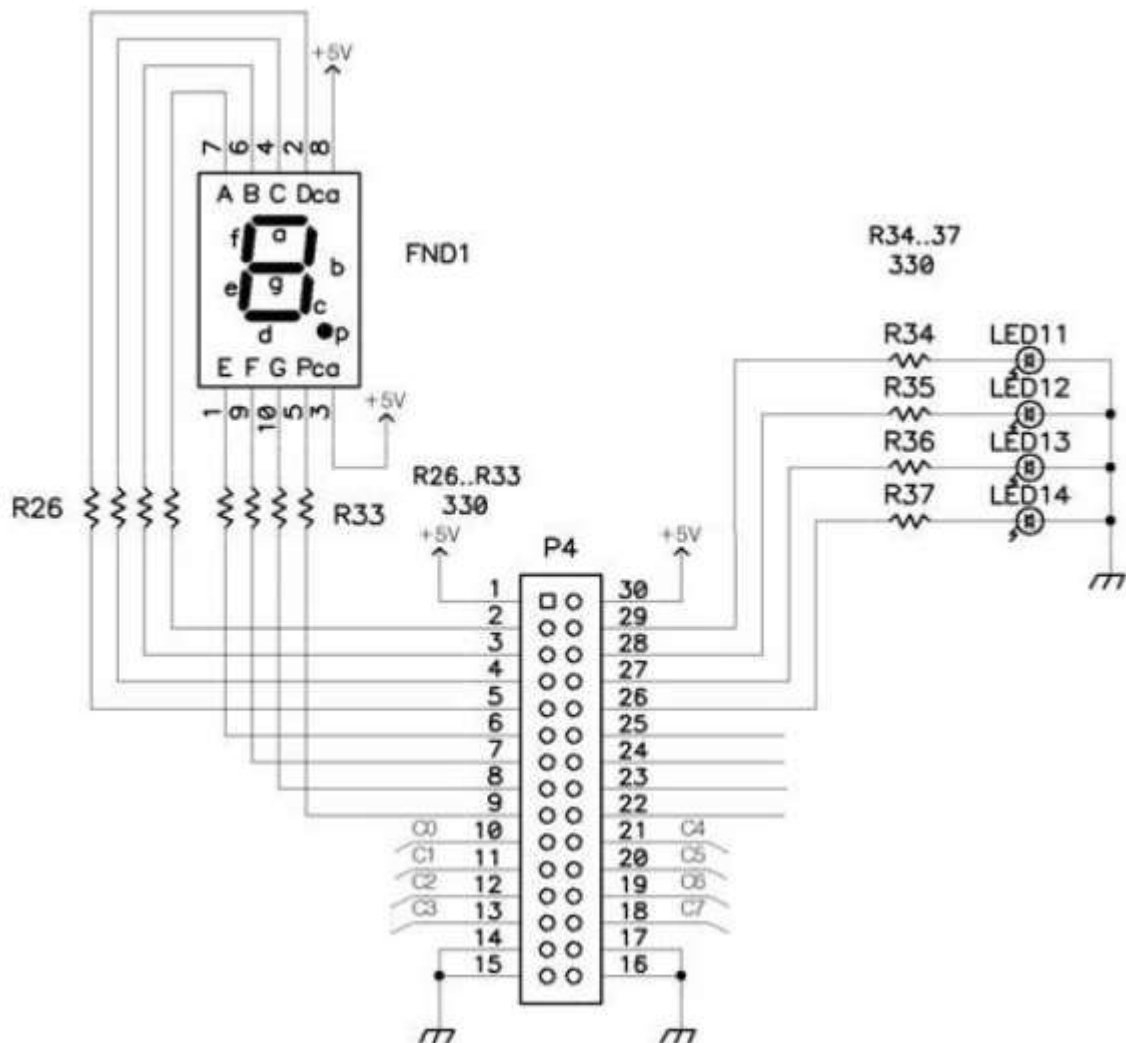
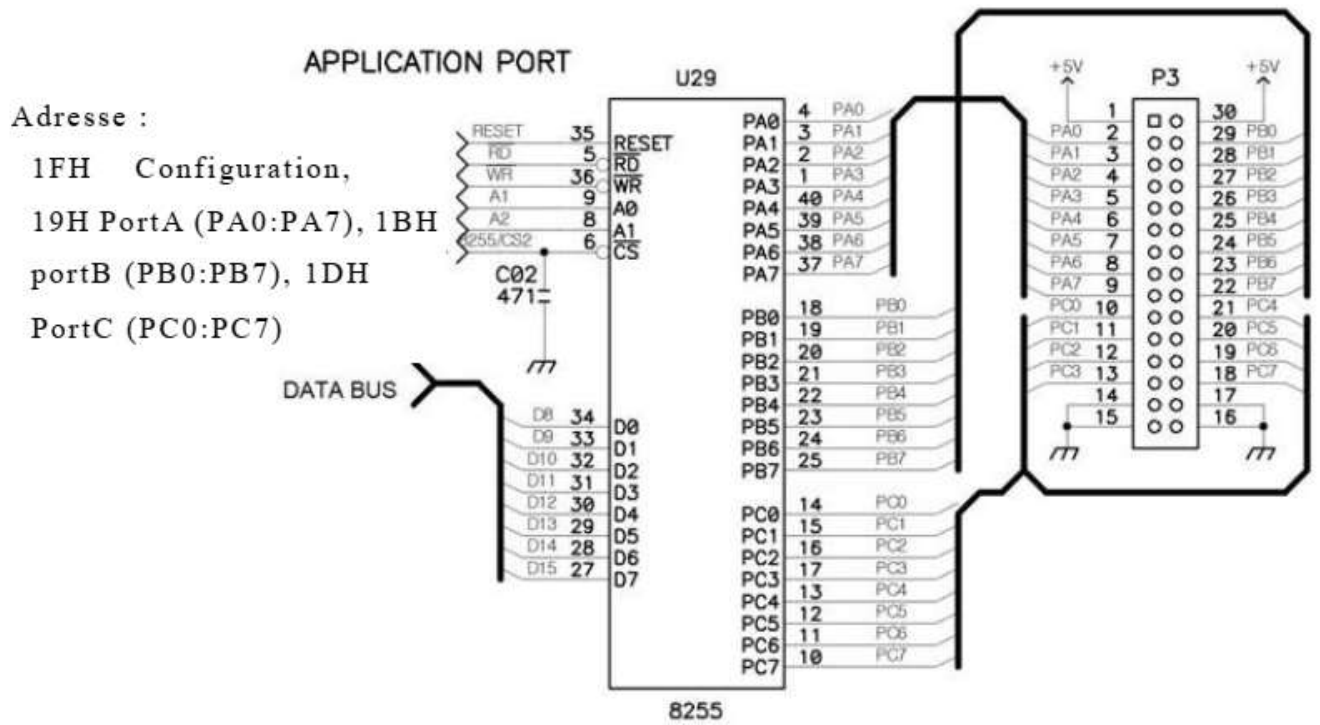
Circuit de l'interface parallèle 8255



Configuration du 8255.

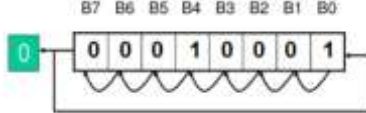
Description du dispositif :

Soit l'interface parallèle représentée avec le circuit ci-dessous.



Schématique Afficheur 7 segment et 4 LEDs interfacés avec le 8255A-CS2.
P4=P3.

Les instructions à utiliser pour ce travail sont énumérées sur le tableau suivant.

<i>Mnémonique</i>	<i>Signification</i>	<i>Nombre de cycles</i>
OUT Adresse, AL	Ecrire le contenu du registre AL vers l'interface spécifiée par l'Adresse.	14
IN AL, Adresse	Lire dans le registre AL le contenu de l'interface spécifiée par l'Adresse.	14
MOV Registre, Valeur	Copier la Valeur dans le Registre (adressage immédiat)	04
MOV Registre1, Registre2	Copier le Registre2 dans le Registre 1	03
ROL Registre		
NOP	No Opération	03
CMP Registre, Valeur	Effectue (Registre – Valeur), le résultat est perdu et actualise les flags : AF CF OF PF SF ZF	04
XOR registre1, registre2	XOR : registre1 = registre1 \oplus registre2	04
DEC Registre	Registre = Registre -1	03
JNZ étiquette JNE étiquette	IP= étiquette si ZF = 0	04 16 cas du jump
JLE étiquette JBE étiquette	IP = étiquette si CF=1 ou ZF =1	04 16 cas du jump
JMP étiquette	IP = étiquette.	11
{id_Adr} DB valeur	Une directive qui définit un octet dans la mémoire est l'initialise avec valeur. L'adresse de cette case est accessible via <i>id_Adr</i> .	/
Identificateur EQU valeur	Déclaration de constante, une directive pour l'assembleur.	/

Travail demandé:

1. Ecrire un programme qui permet d'allumer La LED verte et affiche 3 sur l'afficher 7 segments.
2. Ecrire un programme qui permet d'allumer Les LEDs rouges et affiche 6 sur l'afficher 7 segments.

Remarque

Les LEDs sont placées sur la carte :

LED11 : Rouge, anode connectée avec PB0,
LED12 : Verte, anode connectée avec PB1,
LED13 : Jaune, anode connectée avec PB2,
LED14 : Rouge, anode connectée avec PB3.