Chapitre 5

Interface parallèle d'une carte à µP « Le PIO-8255 »

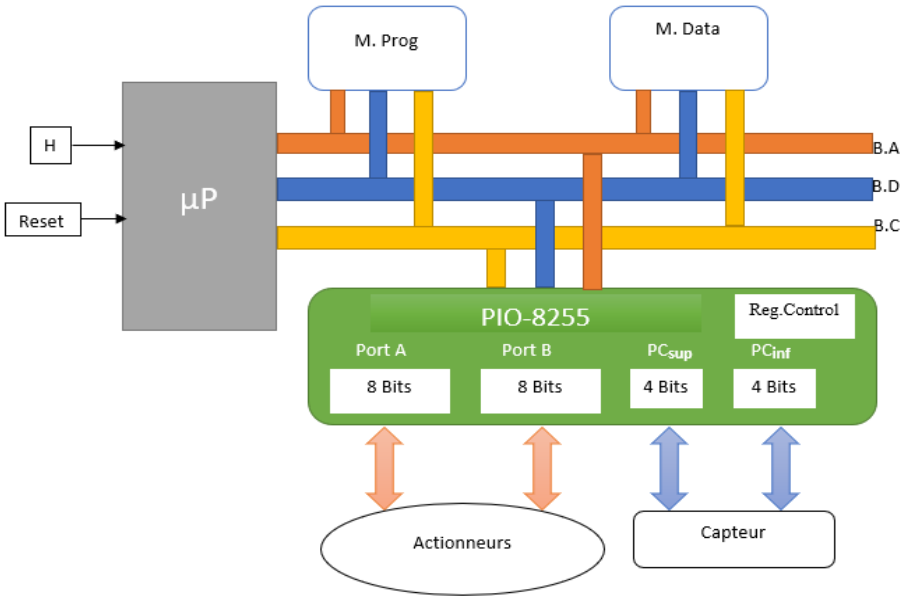
Pour dialoguer avec ses périphériques, l'unité centrale fait appel à trois principes:

1. ***Les échanges programmés :*** C'est le programme qui ordonne un accès à un élément périphérique quelconque.

2. ***Les interruptions :*** L'échange peut désormais avoir lieu de manière aléatoire, à n'importe quel instant, uniquement lorsque le périphérique le demande.

3. ***Le DMA :*** ou accès direct à la mémoire (*«Direct Memory Access»*). Ce mode d'échange privilégie la vitesse.

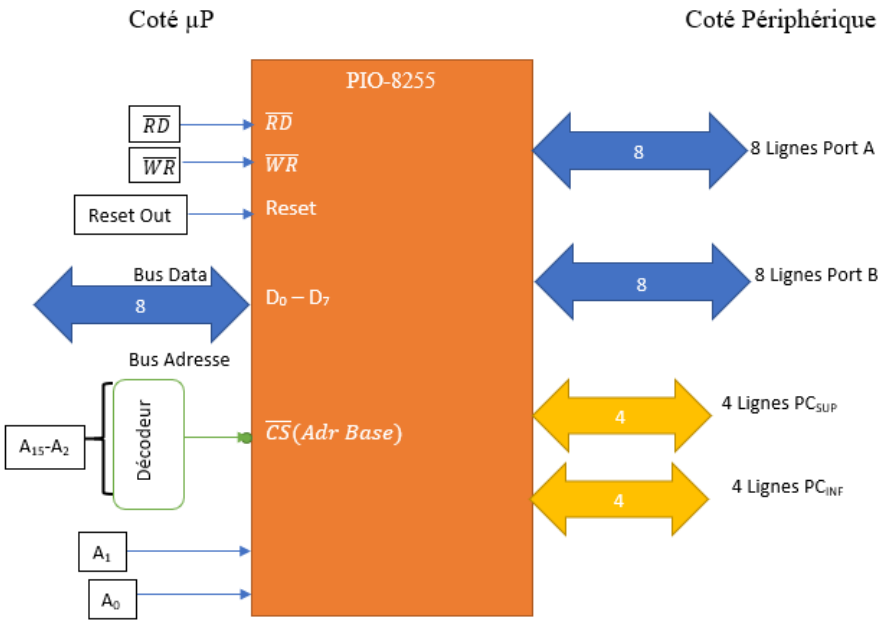
Chacune de ces méthodes offre des avantages et des inconvénients. C'est pourquoi elles sont complémentaires et non concurrentes.



I. Organisation du PIO-8255(parallèle Input-Out put) :

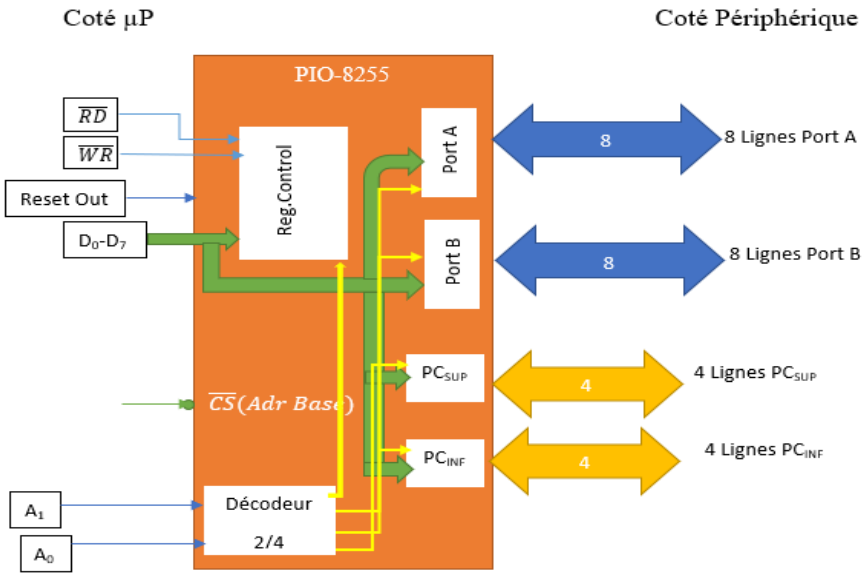
- Le PIO-8255 est un interface parallèle programmable, il reçoit des données et les transmet du/vers le µP au du/vers un ou plusieurs périphériques sur des bue de 8bits (de niveau TTL). Il dispose de 3 portes d'Entrées-Sorties sur 8 bits. Il est programmable ce qui signifie qu'on peut choisir l'état de chaque Port (en entrée/Sortie). Il dispose de 3 modes de fonctionnement :
  - Mode 0 : simple très utiliser.
  - Mode 1 : l'état des ports peut changer durant le programme.
  - Mode 2 : exploitation des Interruption des ports dans la programmation.

II. Structure externe du PIO-8255 :



- Le Bus de données 8 bits de niveau TTL relié aux bus donnés du PIO-8255.
- Les lignes d'adresses sont connectées à un décodeur dans l'une des sorties est connectée à l'entrée $\overline{CS}$ (Chip Select) pour fixer l'adresse de base du PIO (exemple : le cas du SDK : PIO-0 adr_base=40 et PIO-1 adr_base=50).
- Les lignes A_0 et A_1 sont utilisées pour sélectionner les registres et Port internes du PIO.
- Les lignes de contrôle : ($\overline{R}$, $\overline{W}$ et reset out) : sont utilisées pour synchroniser le PIO avec le µP est connecté au reset du PIO pour assurer la réinitialisation du système.

III. Structure Interne du PIO-8255 :



- Le PIO-8255 contient :
  - Le registre Port A : 8 bits de niveau TTL bidirectionnel programmable.
  - Le registre Port B : 8 bits de niveau TTL bidirectionnel programmable.

- Le registre Port C : divisé en 2 ports : C_{INF} et C_{SUP} de niveau TTL de 4 bits longueur programmable indépendamment.
- Un registre de contrôle : qui permet de programmer le mode de fonctionnement du PIO et le sens de transfert de différents registres(Port A, B, C_{INF} et C_{SUP})
- Un décodeur d'adresse (2/4) : qui permet de sélectionner les registres internes du PIO selon le tableau suivant :

- Exercice :
  - Nous voulons automatiser une machine industrielle qui contient : 3 capteurs analogiques, 2 actionneurs analogiques, 18 capteurs TOR (Tout Ou Rien) et 8 actionneurs TOR.
- Donner un synoptique de la carte à µP.

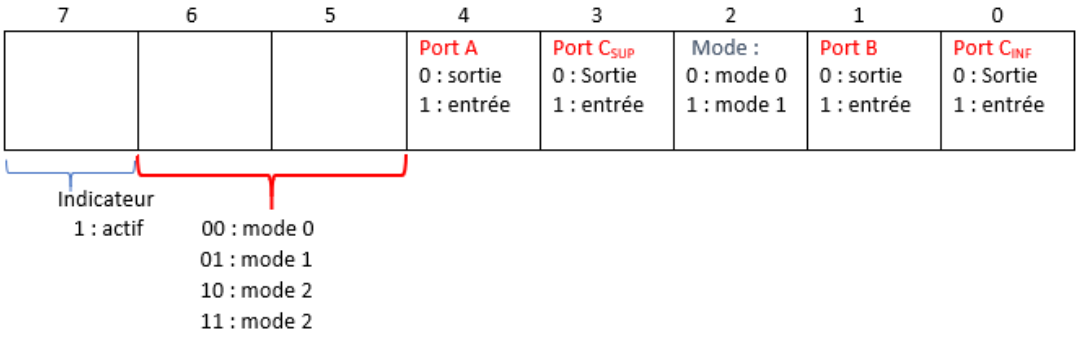
1. Tableau d'adressage des registres internes du PIO 8255 :

Type :	**A_1**	**A_0**	**$\overline{R}\overline{w}$**		**$\overline{CS}$**	**Opération**	**Adresse de Base : Adr**
Entrée	0	0	0	1	0	Port A Bus Data	Adr : Adresse Port A
	0	1	0	1	0	Port B Bus Data	Adr+1 : Adresse Port B
	1	0	0	1	0	Port C Bus Data	Adr+2 : Adresse Port C
Ecriture	0	0	1	0	0	Bus Data Port A	Adr
	0	1	1	0	0	Bus Data Port B	Adr+1
	1	0	1	0	0	Bus Data Port C	Adr+2
Autre	x	x	x	x	1	Bus de donnée relié au PIO à 3 états	
	1	1	0	1	0	Illégal	
	x	x	1	1	0	Bus de donnée à 3 états	

2. Adressage des registres Internes :

Adresse Base	**Registres**	**Exp : SDK 8085**	
		PIO-0	PIO-1
Adr	Port A	40	50
Adr+1	Port B	41	51
Adr+2	Port C	42	52
Adr+3	Registre Contrôle	43	53

3. Programmation du registre de contrôle « RC » :



Exp1 : programmer tous de PIO-1 du SDK-8085 en sortie :

```
MVI A, 80
OUT 53
```

Exp2 : programmer les 3 ports du PIO-1 en entrée :

```
MVI A, 9B
OUT 53
```

Exp3 : lire l'état du port B et écrire l'état sur Port A du PIO-1 :

```
      MVI A,82
      OUT 53
Etiq : IN 51
      OUT 50
      LXI D, 0400
      CALL OFFA
      JMP Etiq
      RST1
```

- Exercice :
  - Nous voulons réaliser un jeu de lumière sur les 8 LEDS de SDK-8085 :
- Les LEDS clignotent entre 00 et FF pendant 10 s puis clignotent entre AA et 55 pendant 10 autres seconde et le programme est bouclé. Avec une fréquence de 500ms.
- Solution :

Adresse :	Code Machine :	Assembleur :
7000	3E 82	MVI A , 82
7003	D3 53	OUT 53
7005	06 14	Etiq : MVI B, 14
7007	3E 00	Etiq1 : MVI A, 00
7009	D3 50	OUT 50
700B	11 00 01	LXI D, 0100
700E	CD FA 0F	CALL 0FFA
7011	3E FF	MVI A, FF
7013	D3 50	OUT 50
7015	11 00 01	LXI D, 0100
7018	CD FA 0F	CALL 0FFA

701B	05	DCR B
701C	C2 07 70	JNZ Etiq1
701F	06 14	MVI B, 14
7021	3E AA	Etiq2 : MVI A, AA
7023	D3 50	OUT 50
7024	11 00 01	LXI D, 0100
7027	CD FA 0F	CALL 0FFA
702A	3E 55	MVI A, 55
702C	D3 50	OUT 50
702E	11 00 01	LXI D, 0100
7031	CD FA 0F	CALL 0FFA
7034	05	DCR B
7035	C2 21 70	JNZ Etiq2
7038	C3 05 70	JMP Etiq

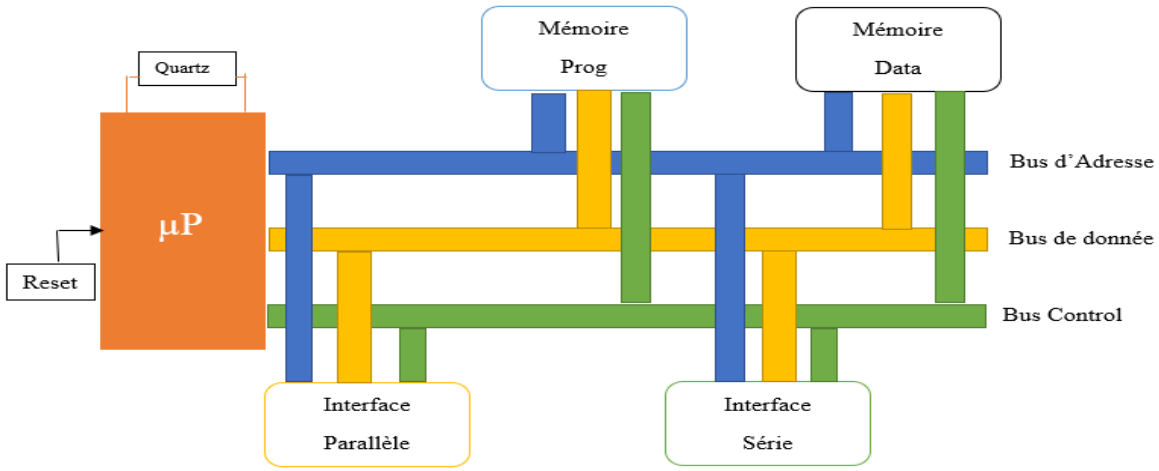
- ## Exercice 2 : (devoir n°03)

- Nous voulons intégrés le Kit-SDK-8085 dans l'automatisation d'une machine industrielle avec :

- 20 capteurs TOR.
- 18 actionneurs TOR.
- 3 capteurs analogiques.
- 2 actionneurs analogiques.

1- Faire le synoptique de la carte.
2- Donner des adresses de Base du PIO.
3- Ecrire un programme d'initialisation des ports.

Gestion des E/S dans une carte à µP :



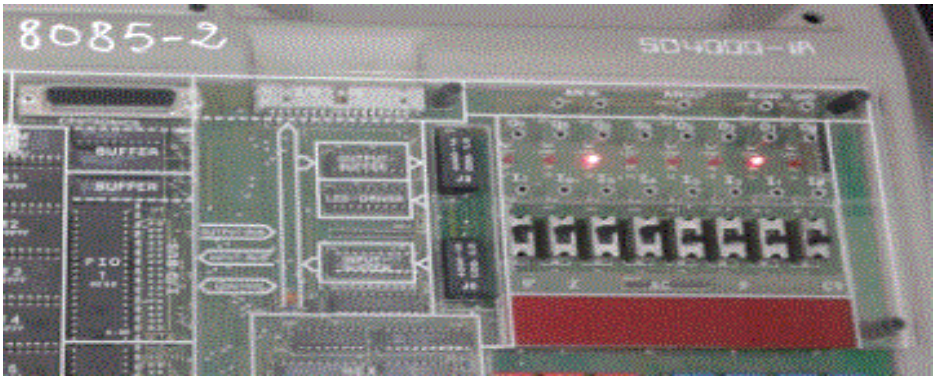
Périphérie : clavier, écran, imprimante, capteurs, actionneur, enregistreur.

- Les échanges entre µP et périphérique se font selon 3 principes :
  1- Mode programmé.
  2- Mode Interruption : transfert asynchrone des données.
  3- Mode DMA (Direct Memory Access).

Exemple : Ecrire un programme qui contrôle l'état des interrupteurs :

ADRESSE	ASSEMBLEUR	CODE MACHINE	COMENTAIRE

7000	MVI A ,82	3 E 82	A ← 82
7002	OUT 53	D3 53	Reg- control ←82
7004	etq : IN 51	DB 51	
7006	OUT 50	D3 50	Adr- port A←Acc
7008	JMP etq	C3 04 70	
7009	RST1	CF	Fin
700A	MVI A ,00 / AA	3 E 00 / AA	A ← 82 / AA
700C	OUT 50	D3 50	Adr- port A←Acc
700E	LXI D,0400	11 00 04	D=04 E=00
7011	CALL 0FFA	CD FA 0F	appel SP



Quelques exercices sur le PIO:

EX1 : ecrire un programme qui lit un ensemble de données dans l'espace mémoire 8000_800A et envoi e ces données sur le port A avec temporisation de 1s :

ADRESSE	ASSEMBLEUR	CODE MACHINE	COMENTAIRE
7000	MVI A ,82	3 E 82	A ← 82
7002	OUT 53	D3 53	Reg- control ←82
7004	LXI H ,8000	21 00 80	HL ← 8000
7007	MVI B,0A	06 0A	B ← 0A
7009	etq :MOV A ,M	7E	
700A	OUT 50	D3 50	Adr- port A←Acc
700C	LXI D,0400	11 00 04	D=04 E=00
700F	CALL 0FFA	CD FA 0F	appel
7012	INX H	23	H←H +1
7013	DCR B	05	B←B -1
7014	JNZ etq	C2 09 70	Si B≠0 va a etq
7017	RST 1	CF	Fin

EX2 : ecrire un programme qui lit l'etat dans interuption et range les données dans le block mémoire 8010-8020

ADRESSE	ASSEMBLEUR	CODE MACHINE	COMENTAIRE
7000	MVI A ,83	3 E 83	A ← 83
7002	OUT 53	D3 53	Reg- control ←83
7004	LXI H ,8010	21 10 80	HL ← 8010
7007	MVI B,0A	06 0A	A ← 0A
7009	etq :MVI A ,01	3 E 01	A ← 01
700B	OUT 50	D3 50	Adr- port A←Acc
700D	LXI D,0400	11 00 04	D=04 E=00
7010	CALL 0FFA	CD FA 0F	Appel SP
7013	MVI A,00	3 E 00	A ← 00
7015	OUT 50	D3 50	Adr- port A←Acc
7017	IN 51	DB 51	Acc←port
7019	MOV M,A	77	
701A	LXI D ,0400	11 00 04	D=04 E=00
701D	CALL 0FFA	CD FA 0F	appel SP
7020	INX H	23	H← H +1
7021	DCR B	05	B←B -1
7022	JNZ etq	C2 09 70	Si B≠0 va a etq
7025	RST 1	CF	Fin