



T.P. N°2 microprocesseurs

Programmation des Entrée/Sorties du PIC 16F877

I. Aperçu

Le PIC 16F877 dispose de 33 Broches d'entrée/sortie (Fig 1). Chacun peut être exploité sous une ou plusieurs des formes suivantes :

- Entrée **et/ou** sortie numérique,
- Entrée analogique,
- Entrée **ou** sortie d'horloge,
- Entrée **ou** sortie de contrôle.
- Entrée de tension de référence.

La broche prend alors des noms différents selon les cas.

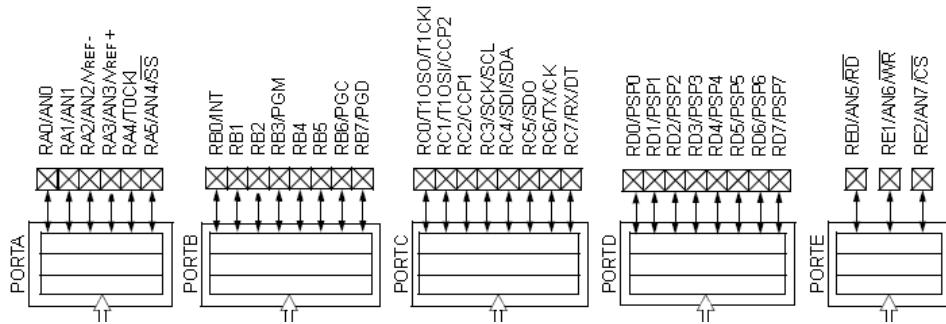


Fig.1

Par exemple La broche marquée **RA0/AN0** possède deux symboles. Chacun de ces symboles est associé à une forme d'utilisation particulière :

- RA0** : La broche porte ce symbole lorsqu'elle est utilisée comme entrée **ou** sortie numérique
AN0 : La broche porte ce symbole lorsqu'elle est utilisée comme entrée analogique

L'ensemble des broches sont regroupées sous forme de **Ports** en nombre de cinq (5).

Le **PortA** utilise 6 broches, Le **PortB** utilise 8 broches, Le **PortC** utilise 8 broches, Le **PortD** utilise 8 broches, Le **PortE** utilise 3 broches (Fig. 1)

A chaque port d'entrée/sortie est associé un seul registre de données et un ou deux registres de contrôle selon le cas (Tab 1).

- **Le registre de données** est un registre qui sert au transfert de données. **Chaque bit** du registre de données correspond à **une broche** du port d'E/S.
- Pour **mettre à 1 une broche** quelconque, il suffit de **mettre à 1 le bit correspondant** du registre du port.

Exemple 1 :

Si on veut mettre à 1 la **broche RA2** il suffit de mettre à 1 le **bit numéro 2** du registre **PORTA** (**adresse 0x05**)



Ce qui correspond à l'instruction suivante :

BSF 0x05, 2 ; mettre à 1 la broche RA2

Exemple 2

Si on veut lire toutes les broches du port B (adresse 0x06), il suffit d'écrire

MOVF 0X06, 0 ; transférer le port B dans le registre w

- **Le registre de contrôle ou de direction** est un registre qui sert à **indiquer le sens de transfert des données**. Chaque broche peut être alors configurée **soit en entrée soit en sortie**.
 - > Pour configurer **une broche quelconque en entrée** il faut **mettre à 1 le bit correspondant** du registre relatif au port concerné.
 - > Pour configurer **une broche quelconque en sortie** il faut **mettre à 0 le bit correspondant** du registre relatif au port concerné.

Port d'E/S	Registre Donnée	Adresse	Registre Contrôle (Direction)	Adresse	Autres registres	Adresse
PortA	PORATA	0x05	TRISA	0x85	ADCON1	0x9F
PortB	PORATB	0x06 , 0x106	TRISB	0x86 , 0x186	OPTION_REG	0x81
PortC	PORATC	0x07	TRISC	0x87	-	-
PortD	PORATD	0x08	TRISD	0x88	TRISE	0x89
PortE	PORATE	0x09	TRISE	0x89	ADCON1	0x9F

Tab. 1

Exemple 3 :

On veut mettre **la broche RA0 en entrée, la broche RA3 en sortie et la broche RB2 en entrée**.

Dans ce cas, il faut configurer les deux registres TRISA et TRISB.

Dans le registre TRISA (adresse 0x85) il faut **mettre à 1 le bit numéro 0 et à 0 le bit numéro 3**

Dans le registre TRISB (adresse 0x86) il faut **mettre à 1 le bit numéro 2**

On peut alors écrire les instructions suivantes

```
BCF 0x85, 0
BSF 0x85, 3
BSF 0x86, 2
```

Notes :

- **il faut veiller à modifier uniquement les bits relatifs aux broches concernés et ne pas toucher aux autres.**
- **Au démarrage du PIC, tous les registres sont remis à zéro. Toutes les broches sont donc configurées en sortie par défaut**

L'accès à un registre de données ou de contrôle peut se faire de deux manières différentes mais dans les deux cas on doit d'abord sélectionner la "bank" où se trouve le registre

1. **Accès à travers son adresse** : dans ce cas on utilise l'adresse du registre concerné (Tab. 1)

Exemple : on veut écrire 0xA0 dans le registre PORTA



```
MOVLW 0xA0 ; charge la donnée 0xA0 dans le registre w
MOVWF PORTA ; écrit le contenu du registre w dans le registre PORTA
```

2 . **Accès à travers son nom** : Dans ce cas on utilise le nom du registre au lieu de l'adresse.

Mais avant, il faut inclure le fichier **P16f877.inc** dans le fichier source à l'aide de la directive :

```
#include <p16F877.inc>
```

Exemple : on veut écrire 0xA0 dans le registre PORTA

```
#include <p16f877.inc> ; cette ligne est obligatoire
MOVLW 0xA0 ; charge la donnée 0xA0 dans le registre w
MOVWF PORTA ; écrit le contenu du registre w dans le registre PORTA
```

II. Travail à faire

II.1. Avant la séance (à remettre pendant la séance)

1. Réécrire le programme ci-dessous sur une feuille en remplaçant les adresses des registres utilisés par leur nom.
2. Ajouter un Commentaire à chaque instruction indiquant son objectif

II.2 Pendant la séance

3. Créer un projet Mplab ou ouvrir un projet existant.
4. Créer un nouveau fichier (Menu **File/New**).
5. Enregistrer le fichier (Menu **File/Save As...**) dans le dossier c:\Mplab, lui donner le nom TP2.asm.
6. Saisir le programme ci-dessous, le compiler (Menu **Project/Build All**), corriger les éventuelles erreurs, remplacer les adresses des registres par leur nom puis l'exécuter pas à pas (Touche F7 du clavier).
7. Vérifier l'exécution du programme :
 - a. En visualisant l'état des registres utilisés (Menu **View/watch**).
 - b. En visualisant l'état des entrées/sorties (Menu **View/Simulator Logic Analyzer**)

```
#include <p16f877.inc>
        BSF      0x03,5
        BCF      0X86,0
        BCF      0x03,5
deb     MOVLW    0xFF
        MOVWF    0x70
        BSF      0x06,0
att1   DECFSZ   0x70,1
        GOTO     att1
        MOVLW    0xFF
        MOVWF    0x70
        BCF      0x06,0
att2   DECFSZ   0x70,1
        GOTO     att2
        GOTO     deb
        END
```