

INFO2 : INFORMATIQUE EMBARQUEE

TD3 : Accès aux registres – Entrées/Sorties Tout Ou Rien

1 Travail préparatoire

Lisez l'énoncé du TP1 jusqu'à la partie 2 incluse.

1. Par sécurité, au *reset* du PIC18F4520, toutes les broches d'entrées/sorties sont en entrée. Pourquoi à votre avis ?

2 Gestion au niveau octet des registres associés aux ports d'entrées/sorties

Dites comment vous feriez en langage C pour :

2. Au niveau du port B, déclarer les 5 bits RB7, RB6, RB4, RB3, RB2 en sortie et les 3 bits RB5, RB1, RB0 en entrée.
3. Déclarer l'intégralité du port C en sortie puis imposer sur celui-ci les 2 bits de poids le plus fort à 1 et les bits restant à 0.
4. Déclarer l'intégralité du port A en entrée et récupérer dans une variable `etatPortA` de type approprié la valeur du port A, imposée par exemple par 8 interrupteurs.
5. Déposer sur le port C l'état du port A.
6. Déposer sur le port C le complément à 1 du port A (inversion de chaque bit) en utilisant :
 - l'opérateur unaire de complémentation `~` ;
 - l'opérateur binaire OU EXCLUSIF `^`.

3 Gestion au niveau bit des registres associés aux ports d'entrées/sorties

3.1 En utilisant des masquages

Dites comment vous feriez en langage C pour que :

7. RD5 et RD0 soient des sorties, la direction des autres broches du port D restant fixe.
8. RD7, RD6, RD3 soient des entrées, la direction des autres broches du port D restant fixe.
9. RC4, supposée déjà configurée en sortie, ait son état qui s'inverse.
10. de façon permanente, si l'entrée RB2 est à 1, alors la fonction `declencheAction()` soit appelée.

3.2 En utilisant des masquages et des décalages pour accéder à l'état d'un bit

11. Cahier des charges

Ecrivez un programme complet pour que, de façon permanente, quand la broche RA3 du port A est à 1 (parce qu'on a appuyé sur un bouton poussoir), la led active haute connectée à RB7 s'allume. Sinon elle doit s'éteindre. (Faire un schéma électrique)

3.3 En utilisant des structures de bits

Dans **p18f4520.h**, le fichier d'en-tête du PIC18F4520, on trouve :

```
extern volatile near unsigned char PORTA;
extern volatile near union {
    struct {
        unsigned RA0:1;
        unsigned RA1:1;
        unsigned RA2:1;
        unsigned RA3:1;
        unsigned RA4:1;
        unsigned RA5:1;
        unsigned RA6:1;
        unsigned RA7:1;
    };
    struct {
        unsigned AN0:1;
        unsigned AN1:1;
        unsigned AN2:1;
        unsigned AN3:1;
        unsigned T0CKI:1;
        unsigned AN4:1;
        unsigned OSC2:1;
        unsigned OSC1:1;
    };
    struct {
        unsigned :2;
        unsigned VREFN:1;
        unsigned VREFP:1;
        unsigned :1;
        unsigned SS:1;
        unsigned CLKO:1;
        unsigned CLKI:1;
    };
    struct {
        unsigned :2;
        unsigned CVREF:1;
        unsigned :2;
        unsigned NOT_SS:1;
    };
    struct {
        unsigned :5;
        unsigned LVDIN:1;
    };
} PORTAbits;
```

Le port A est un octet (**unsigned char**) défini dans un fichier externe (**extern**) dont la valeur peut être écrasée entre deux appels (**volatile**).

La deuxième déclaration précise que **PORTAbits** est une union de structures anonymes de bits adressables. C'est parce que chaque bit d'un registre SFR peut avoir plusieurs usages qu'il y a plusieurs définitions de structures à l'intérieur de l'union.

Dans le cas présent, les bits du port A sont définis comme :

1^{ère} structure :

Port d'entrée/sortie GPIO parallèle (**RA7** à **RA0**).

2^{ème} structure :

Port d'entrées analogiques (5 entrées **AN4** à **AN0**) + Entrée d'horloge du timer 0 (**T0CKI**) + Entrées d'oscillateur (**OSC1** et **OSC2**).

3^{ème} structure :

Deux entrées de référence de tension du CAN (**VREFN** et **VREFP**) + Entrée de sélection du port série synchrone (**SS**) + Sortie d'horloge (**CLKO**) + Entrée d'horloge (**CLKI**).

4^{ème} structure :

Voir documentation du PIC18F4520.

5^{ème} structure :

Voir documentation du PIC18F4520.

Exemple d'utilisation utilisant la 1^{ère} structure :

```
n = PORTAbits.RA1; // la variable n reçoit l'état
                  // du bit n°1 du port A
```

12. Dans des programmes en langage C, dites comment, vous accéderiez à :

- la GPIO RA4 ;
- l'entrée d'horloge du *timer* 0.

13. Pourquoi le port A est-il décrit comme une union de structures de bits ?

14. Réécrivez le programme du cahier des charges de la partie précédente en utilisant les structures de bits. Que pensez-vous de la lisibilité obtenue ?

4 Ecriture d'un chenillard

15. Cahier des charges

Réaliser un chenillard sur le port D : RD0 est allumée, puis c'est au tour de RD1, puis de RD2, etc. Quand RD7 a été allumée, c'est à nouveau au tour de RD0 de l'être. Une variable image du port D sera utilisée et on réalisera des décalages sur celle-ci.