

Aix-Marseille Université

Contrôle des connaissances

UE3 – Automatismes et informatique industrielle

CORRIGE

Partie Informatique industrielle

Questions de cours (3 pts)

Question 1: Le vecteur RESET est la case mémoire sur laquelle le programme démarre lorsque le microcontrôleur est mis sous tension.

Question 2: Le vecteur d'INTERRUPTION, placée à l'adresse 0x0008, est la case mémoire sur laquelle le programme est redirigé lorsqu'une interruption de priorité haute est déclenchée. Une telle interruption peut provenir par exemple du débordement d'un compteur tel le TIMER0

Question 3: pour un PIC18F4520, il y a (8000)h cases donc 32000 cases environ ; Pour un PIC18F4420 (4000)h cases donc 16000

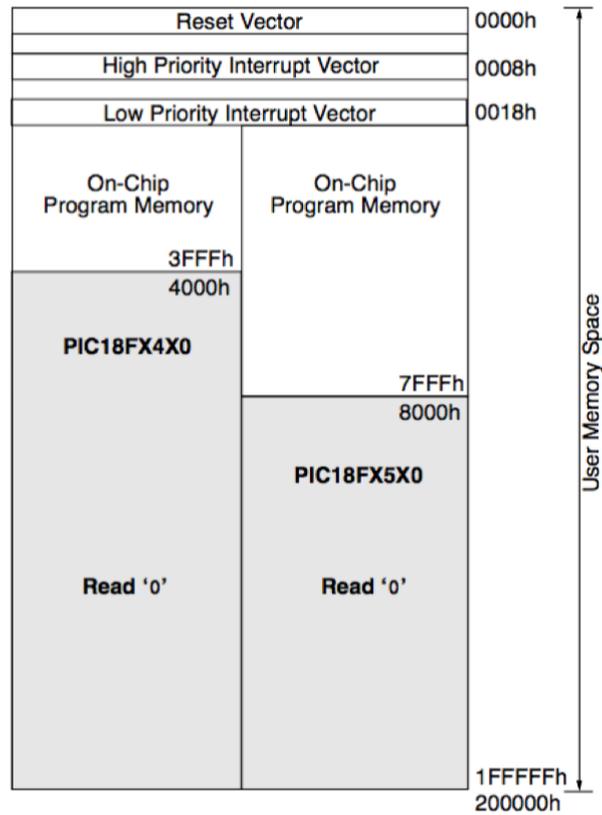


Figure 1 : organisation de la mémoire programme du PIC 18F4520.

Exercice 1 : programmation C (3 pts)

On considère le programme suivant :

```
//-----
int a = 2;

void main{

int b=2 ;
int c=3 ;
```

```
int* Pc=&c ;
Ma_Fonction(b, Pc)

printf(a);
printf(b);
printf(c);

}

void Ma_Fonction(int Valeur, int* Pointeur) {
Valeur = a+Valeur ; 4
*Pointeur = *Pointeur + a ; 5
a= Valeur ; 4
}
```

//-----

Ce programme affiche deux valeurs qui sont b et c, après l'appel de la fonction Ma_Fonction.

Question 1: Quelle est la variable globale dans ce programme ?

La variable globale est a

Question 2: Quelles sont les valeurs de a, b, c, affichées par les fonctions fprintf ?

a=4 modifié car variable globale; b=2 non modifié car variable locale ; c=5 modifié car passage par pointeur

Exercice 2 : Programmation d'un microcontrôleur en C (2 pts)

On considère le programme suivant :

```
//-----  
#include <delays.h>  
#include <p18f4520.h> // Définition des variables globales  
#pragma config WDT = OFF  
  
#define Enable                PORTDbits.RD6  
  
// Definition : masques pour affectation uniquement du bus data du LCD  
placé en RD0:RD3  
#define masque_LSB           0x0F  
#define masque_MSB           0xF0  
  
void lcd_ecrire_quartet(unsigned char quartet_dans_octet,  
unsigned char PORTD)  
{  
  
    quartet_dans_octet  &= masque_LSB;  
    PORTD                &= masque_MSB;  
    PORTD                |= quartet_dans_octet;  
  
    PORTD = 1;  
    Enable                = 1;  
    Delay1TCY();  
    Delay1TCY();  
    Delay1TCY();  
    Delay1TCY();  
    PORTD = 0;  
    Enable                = 0;  
  
}
```

Ce programme est censé modifier le PORTD en fonction de quartet_dans_octet, autoriser le transfert d'information sur le PORTD en faisant passer Enable à 1, puis ne plus autoriser le transfert d'information en faisant passer Enable à 0 (après un délai) .

L'opération &= signifie 'ET Logique' puis affectation ;

L'opération |= signifie 'OU Exclusif' puis affectation.

Question 1: Ce programme comporte plusieurs erreurs.

Barrez les lignes correspondantes, et si nécessaire réécrivez une solution correcte, en justifiant brièvement votre choix.

Question 2 : Supposons qu'avant exécution de la fonction `lcd_ecrire_quartet`, on ait :

```
PORTD= 0101 1100
```

Et

```
quartet_dans_octet = 0100 1010;
```

Que vaut `PORTD` après exécution de la fonction ?

```
PORTD= 0101 1010
```

Quartet de poids fort protégé, quartet de poids faible mis à jour avec le contenu du quartet de poids faible de `quartet_dans_octet`.

PARTIE Automatismes

Exercice 1 (3 points)

On considère la photographie en figure 1. Il s'agit d'une interface homme-machine tactile.



Figure 1: interface homme machine

Question 1: Dans ce système, déterminez ce qui joue le rôle de:

-la partie opérative avec:

-actionneurs :

Un rail qui saisit la boisson sélectionnée et la fait tomber

, et

-capteurs :

Des capteurs infrarouge au-dessus et sur les côtés de l'écran, qui permettent de définir l'emplacement du doigt de l'utilisateur.

;

- la partie relation :

L'écran, qui semble être un écran tactile mais qui en fait est associé à des capteurs infra-rouge sur les côtés.

Question 2: De quoi peut être constituée la partie commande ?

On peut penser à un automate programmable industriel à l'intérieur du distributeur

Exercice 2 (4 points)

PWM et programmation d'automate

Considérons la partie chauffage de la maquette de domotique étudiée en TP. Elle comporte une ampoule chauffante qui peut être alimentée par le biais d'une PWM (pulse width modulation, c'est-à-dire modulation de largeur d'impulsion), de rapport cyclique alpha. Ce coefficient varie entre 0 et 1.

La partie commande de la maquette de domotique est un automate cruzet. Il travaille sur des paquets de données de 10 bits.

Question 1:

Le programme présenté en ci-dessous permet-il d'utiliser le mode d'alimentation PWM ? Si non, proposez (de façon succincte) une solution.

Non, il faut brancher une PWM en OF XA

Question 2:

Le coefficient alpha est transcrit sous la forme d'une valeur entière NUM dans l'automate. Que choisir pour NUM si on souhaite 0.5 comme valeur de alpha ? Que vaut NUM quand alpha vaut 1 ?

Les valeurs de NUM sont comprises entre 0 et 1023. Pour alpha=0.5, on a NUM = 512. Pour alpha=1, on a NUM = 1023.

Considérons le graphe présenté ci-dessous : on affiche la température T dans la maquette en fonction du temps t. Donnez un exemple de couple de valeurs (alpha, NUM) que l'on a pu utiliser pour obtenir :

- premièrement la montée en température 1, -> **(1,1023) (chauffage rapide)**

- deuxièmement la montée en température 2. -> (0.5,512) (chauffage lent)

