

Site :  Luminy  St-Charles  St Jérôme  Cht-Gombert  Aix-Montperrin  Aubagne-SATIS

Sujet session de :  1<sup>er</sup> semestre -  2<sup>ème</sup> semestre -  Session 2

Durée de  
l'épreuve : 2h00.....

Examen de :  L1/ L2/ L3 -  M1/ M2 -  LP -  DU Nom diplôme : L3 SPI AGE

Code Apogée du module : SPI5U8 Libellé du module : Automatismes et Informatique industrielle

Document autorisé :  OUI -  NON (Poly de cours)

Calculatrices autorisées :  OUI -  NON

Aix-Marseille Université

**Contrôle des connaissances**  
**Automatisme et informatique industrielle**

**Documents: seul le photocopié de cours est autorisé**  
**Calculatrices autorisées**

**Cet examen de 2 heures est composé de 2 parties :**

- partie '*Informatique industrielle*' (10 points / approximativement 60min).
- partie '*Automatisme*' (8 points / approximativement 45min),

La *présentation* et le soin apporté à la rédaction de façon générale comptent pour 2 points.

**Vous rendrez votre sujet avec la copie d'examen**

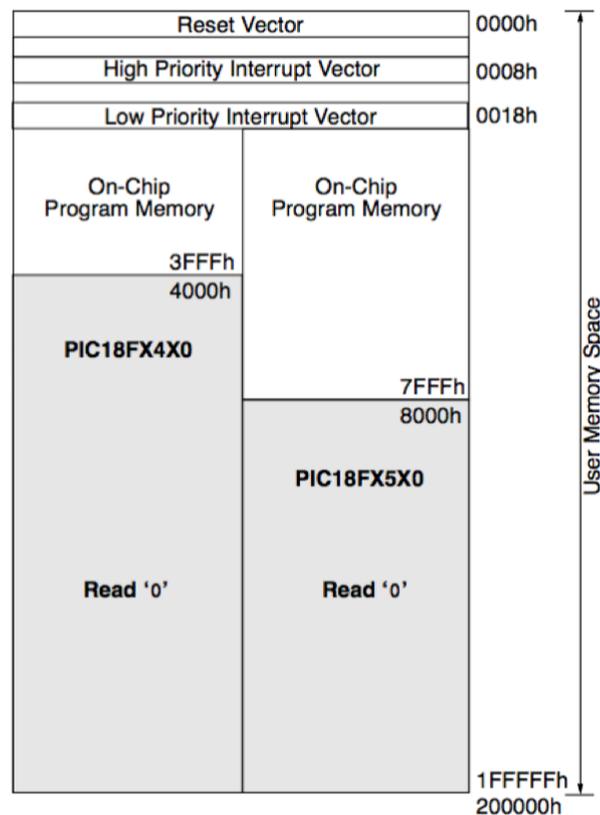
## Partie Informatique industrielle

### Questions de cours (3 pts)

**Question 1:** Expliquez le rôle du vecteur RESET dans la mémoire programme représentée *figure 1* ci-dessous.

**Question 2:** Expliquez le rôle du vecteur d'INTERRUPTION placé à l'adresse 0x0008 dans la mémoire programme représentée *figure 1* ci-dessous.

**Question 3:** Combien y a-t-il de cases mémoire dans la mémoire programme qui n'est pas 'Read 0', pour un PIC18F4520 ? Pour un PIC18F4420 ?



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

## Exercice 1 : programmation C (3 pts)

On souhaite configurer de façon simplifiée une interruption de débordement du TIMER0.  
Si le bit `TMR0ON` est à 1, il faut donner à `TMR0H` et `TMR0L` les valeurs `0x0B` et `0xDB` respectivement.

```
//-----Debut du programme-----  
int TMR0ON = 1;  
  
void main{  
  
    int TMR0H = 0x00;  
    int TMR0L = 0x00;  
  
    Config_Timer0(TMR0H,TMR0L, TMR0ON);  
    printf(TMR0H);  
    printf(TMR0L);  
}  
  
void Config_Timer0(int Valeur1, int Valeur2, int Valeur3) {  
  
    if (Valeur3 ==1)  
    {  
        Valeur1 = 0x0B ;  
        Valeur2 = 0xDB ;  
    }  
}  
  
//-----Fin du programme-----
```

Ce programme affiche deux valeurs qui sont `TMR0H` et `TMR0L`, après l'appel de la fonction `Config_Timer0`.

**Question 1:** Quelle est la variable globale dans ce programme ? Quelles sont les variables locales dans ce programme ?

**Question 2:** Quelles sont les valeurs de `TMR0H` et `TMR0L`, affichées par `printf` ? Justifiez brièvement.

**Question 3:** On vous demande de proposer une solution, la plus simple possible, pour que la modification de `TMR0H` et `TMR0L` soit prise en compte dans le programme principal : sur l'énoncé, faites les modifications adéquates, en barrant et réécrivant si nécessaire.

## Exercice 2 : Programmation d'un microcontrôleur en C (4 pts)

Considérons les fonctions suivantes, qui s'inspirent du programme chronomètre vu en TP :

La fonction bouton\_appui, qui boucle tant qu'on n'a pas appuyé sur un bouton poussoir puis levé le doigt du bouton; la fonction d'initialisation du vecteur d'interruption de priorité haute, et la fonction TMR0\_overflow qui est exécutée lorsqu'il y a débordement du TIMER0, qui permet de gérer l'incrément du chronomètre et son affichage. Ces fonctions sont présentées dans la page ci-après.

La valeur de MSB est 0x0B, la valeur de LSB est 0xDB.

**Question 1:** Ce programme comporte plusieurs erreurs. Sur l'énoncé, barrez les lignes correspondantes, et si nécessaire réécrivez une solution correcte, en justifiant brièvement votre choix.

**Question 2 :** La période  $T$  du Timer0 est définie de la façon suivante:

$$T = T_{Cy} * (max-min) * Prescaler$$

où  $T_{Cy} = 10^{-6}$  sec. *Prescaler* vaut 16, et *max* vaut  $(65535)_d$ .

Sachant que  $(0BDB)_h = (3035)_d$  quel est le temps écoulé entre chaque interruption de débordement du Timer0 ?

**Question 3 :** Faites un algorithme pour la fonction TMR0\_overflow.

Fonction d'appui sur bouton

```
void bouton_appui (void)
{
    // Tant que PORTXbits_bouton=1 (ie. bouton poussoir non actionne), on boucle...
    while( ! PORTXbits_bouton);

    // Tant que PORTXbits_bouton=0 (ie. bouton poussoir actionne), on boucle...
    while(PORTXbits_bouton);
}
```

Vecteur d'interruption et routine d'interruption

```
#pragma code Vecteur_interruption_priorite_haute = 0x256
// Directive pour spécifier l'adresse memoire ou debute la fonction

void Vecteur_interruption_priorite_haute (void)
{
    // va a la routine de gestion des interruptions de priorite haute
    _asm
        goto TMR0_overflow // instruction assembleur
    _endasm
}
#pragma code

#pragma interrupt TMR0_overflow
void TMR0_overflow(void)
{
    // RAZ flag it venant de TMR0
    INTCONbits.TMR0IF = 1;

    // RAZ TIMER 0
    TMR0H = MSB;
    TMR0L = LSB;

    // Increment d'une seconde et ajustement des minutes
    SEC ++;

    if(SEC == 65535){
        MIN ++;
        SEC=0;
    }

    lcd_affiche_chrono();
}
```

## PARTIE Automatismes

### **Exercice 1 (4 pts)**

On considère la photographie en *figure 2*. Il s'agit de la sonde Voyager 2. Son composant le plus proéminent est son antenne de 3.7 mètres de diamètre.



*Figure 2: sonde Voyager 2*

Les principaux instruments de mesure de Voyager 2 sont les suivants

ISS	Caméras
IRIS	Spectromètre infrarouge
UVS	Spectromètre ultraviolet
CRS	Analyse rayons cosmiques
LECP	Particules à faible énergie

Ces instruments sont montés sur une plateforme orientable, ce qui leur permet de viser une cible particulière. Par exemple, l'étude de Titan, un satellite de Saturne, a été un objectif majeur de Voyager 2.

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

- la partie opérative avec:
  - actionneurs, et
  - capteurs;
- la partie relation

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

## Exercice 2 (4 pts)

Cet exercice est fondé sur la maquette de domotique vue en TP. On s'intéresse à la boîte 'chauffage'. 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 est à valeurs entre 0 et 1.

La partie commande de la maquette de domotique est un automate crouzet. Il travaille sur des paquets de données de 10 bits, qui codent des valeurs entières (notées ici par exemple NUM).

### Question 1:

Le programme présenté en figure 3 permet-il d'utiliser le mode d'alimentation PWM ? Si non, proposez (de façon succincte) une solution.

### Question 2:

Le programme présenté comporte un trigger. Faites un schéma pour indiquer son mode de fonctionnement. A quelle sortie du programme (invisible sur la figure 3) est-il connecté ?

### Question 3:

Le coefficient alpha est transcrit sous la forme d'une valeur entière NUM dans l'automate.

Donnez la relation simple qui définit NUM à partir de la valeur d'alpha.

### Question 4:

En figure 4, deux montées en température sont représentées: c'est l'évolution de la température en fonction du temps. Quelles sont les valeurs de alpha, et de NUM, qui à chaque fois ont pu être utilisées pour effectuer ces montées en température ?

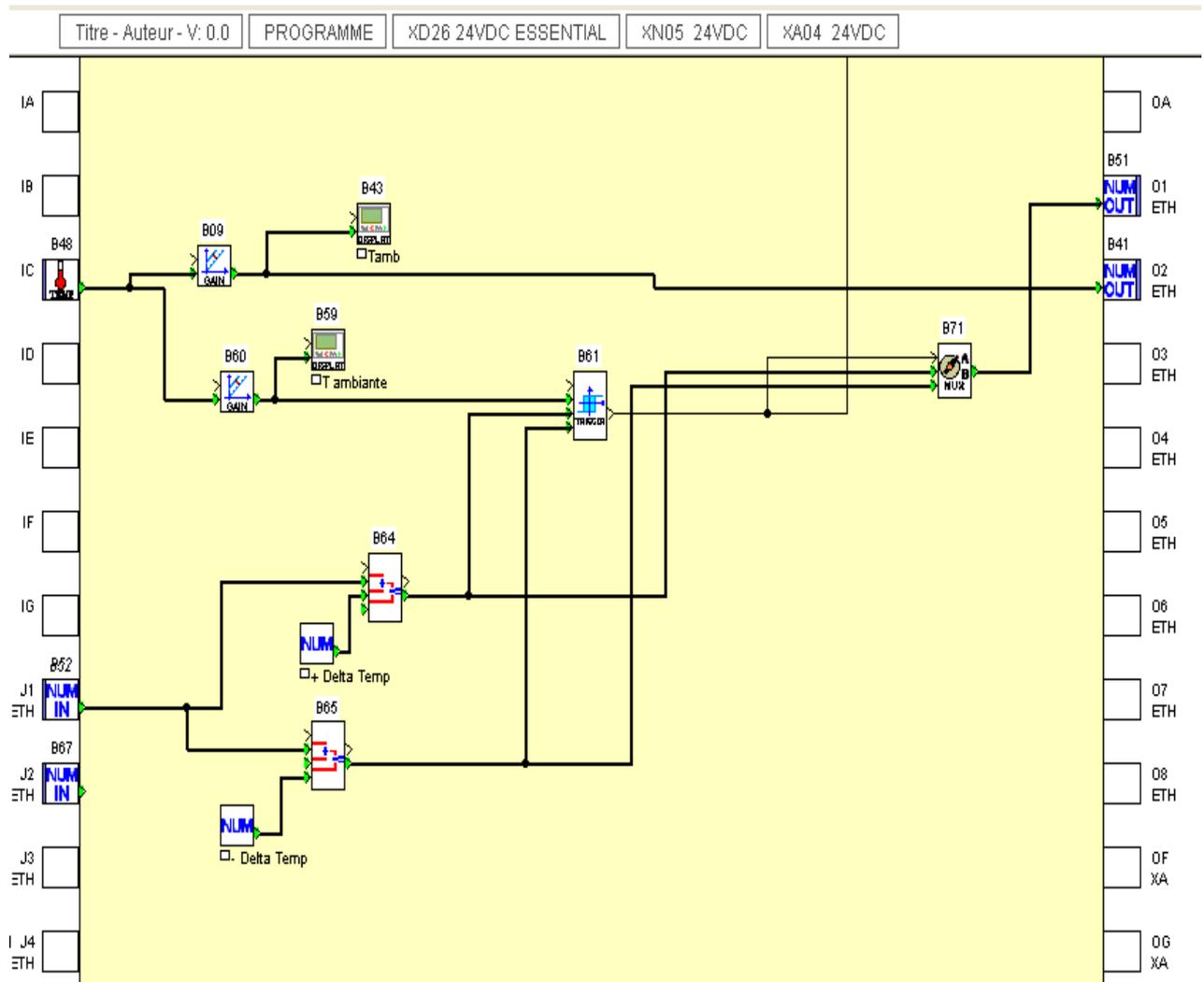
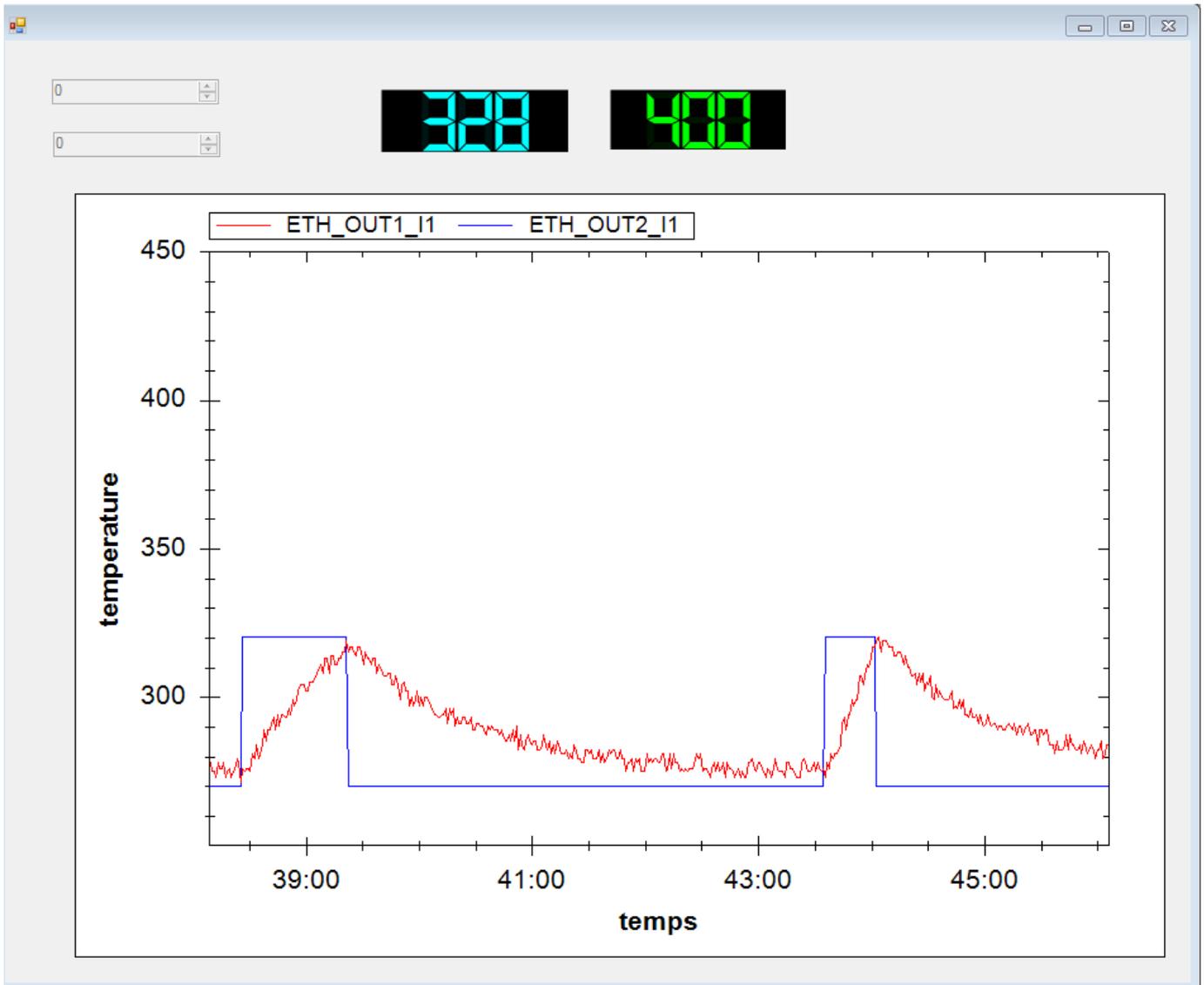


Figure 3: programme FBD pour régulation TOR



*Figure 4: deux montées en température*