	Institut Supérieur des Etudes Technologiques Département de Génie Electrique Licence appliquée en Automatismes et Informatique Industrielle	Année universitaire : 2011/2012 Semestre : 1 Date : 14/11/2011 Durée : 1h30
	Devoir Surveillé	
Matière : Système Temps Réel	Unité d'enseignement : Systèmes Automatisés 2	Classes : AI3.1
Documents : Non autorisés	Enseignant : Ali HMIDENE	Nb. Pages : 3

On se propose d'interfacer une carte électronique avec un PC via une liaison série RS232. La carte comporte un microcontrôleur PIC16F876, un capteur de température et un afficheur LCD (figure 1).

Le système comporte deux parties indépendantes :

- Acquisition de la température de façon continu à une fréquence de 5 Hz et envoi de la valeur numérique au format ASCII via la liaison série.
- Affichage des informations reçues à partir du PC sur un afficheur LCD.

La première partie comporte les tâches suivantes :

- Acquisition de la température ($C_1 = 70\mu s$)
- Conversion binaire-ASCII, on suppose que la valeur est codée au maximum sur 3 caractères ($C_2 = 30\mu s$)
- Transmission série de la valeur convertie avec un débit de 9600 bps (un caractère est codé sur 10 bits).

La deuxième partie comporte les tâches suivantes :

- Réception des informations à partir du PC (une information est codée sur 4 caractères en ASCII). Ces informations arrivent à une fréquence de 10 Hz.
 - Affichage de l'information reçue sur un afficheur LCD.
- 1) Calculez le temps d'exécution de la tâche de transmission série de la valeur de la température vers le PC (C_3).

$$C_3 = T_b \times Nb. \text{ de bits} \times Nb. \text{ de caractères} = \frac{1}{9600} \times 10 \times 3 = 3,12ms$$

- 2) Calculez le temps d'exécution de la tâche de réception des informations à partir du PC (C_4).

$$C_4 = T_b \times Nb. \text{ de bits} \times Nb. \text{ de caractères} = \frac{1}{9600} \times 10 \times 4 = 4,16ms$$

- 3) Calculez le temps d'affichage d'une information sur l'LCD (C_5). La procédure d'affichage devra être toujours précédée par l'effacement de l'LCD.

Temps d'effacement de l'LCD = 1,64 ms

Temps d'écriture d'un caractère sur l'LCD = 40 μs

$$C_5 = 40\mu s \times 4 + 1,64ms = 1,80ms$$

- 4) Etant donné l'organigramme de la figure 2. Montrez que cet organigramme ne respecte pas les contraintes temps réel du système.

Les informations reçus par le PC arrivent à des instants imprévisibles (événements asynchrones). Or cette tâche est traitée de manière séquentielle, elle est exécutée périodiquement après ($70\mu s + 30\mu s + 3,12ms + 1,8ms = 5,02ms$). Les caractères reçus seront écrasés avant l'exécution de la tâche de la réception.

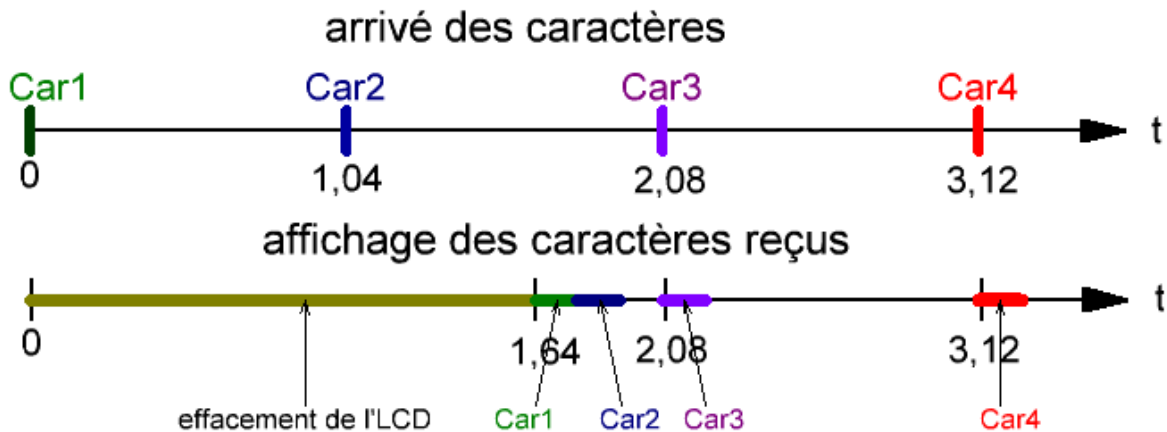
5) La partie réception et affichage des informations sur LCD est gérée par interruption.

```
#int_rda
Void isr_serie()
{
    Reception_Affichage_LCD()
}
```

L'organigramme de la fonction réception et affichage des informations sur LCD est donné par la figure 3. Les caractères reçus sont affichés en ligne, c.à.d. chaque caractère reçu est affiché immédiatement sur l'LCD.

a) Est-ce que l'organigramme proposé respecte les contraintes temps réel ?

L'arrivée du premier caractère déclenche l'exécution de la procédure Reception_Affichage_LCD. Ce caractère sera affiché sur l'LCD après 1,64ms (après effacement de l'LCD). Les autres caractères se succèdent à des intervalles de 1,04ms. Le chronogramme suivant illustre le temps d'arrivée et d'affichage de 4 caractères.



b) Ecrire le code de la procédure Reception_Affichage_LCD().

```
Reception_Affichage_LCD()
{
    n = 0 ;
    do{
        c = getc() ;
        if( n == 0)
            lcd_puc("\f") ;
        lcd_puc(c) ;
        n++ ;
    } while(n != 4) ;
}
```

c) Que peut-on dire si le débit de la liaison série était de 19200 bps ? Proposez une autre solution si les contraintes temps réel ne sont pas respectées.

Les caractères arrivent à des intervalles de 0,52ms ; le 4^{ème} caractère arrive au bout de 1,56ms après le 1^{er} caractère. Cet intervalle de temps est inférieur au temps d'effacement de l'LCD. Il faut donc faire sortir l'affichage de la routine d'interruption.

La partie acquisition est gérée par le timer 2 en mode interruption. La période du débordement du timer 2 étant fixée à 5ms.

6) Calculez les valeurs du prédiviseur, du postdiviseur et du registre PR2.

```
Prediv = 4 ; PR2 = 249 ; Postdiv = 5
```

7) Complétez la routine d'interruption ci-dessous :

```
#int_timer2
void isr_timer2()
{
    count-- ;
    If(count == 0)
    {
        Val = read_ADC() ;           // Fonction d'acquisition
        sprintf(string, "%u",val) ;   // Conversion binaire-ASCII
        print("%s",string) ;         // Transmission série
        count = 40 ;
    }
}
Void main()
{ Init() ;                          // procédure d'initialisation
  while(1) ;
}
```

8) Vérifiez la validité de cette démarche.

Le microcontrôleur PIC16F876 ne gère pas la priorité des interruptions. Lorsque les informations série arrivent pendant l'exécution de la routine d'interruption du timer2, il est fort probable de perdre quelques caractères.

9) Proposez une solution si les contraintes de temps ne sont pas respectées.

La transmission série à l'intérieur de la routine d'interruption peut gêner la réception des informations. Donc la transmission série et la procédure d'affichage doivent être placées dans le programme principal.

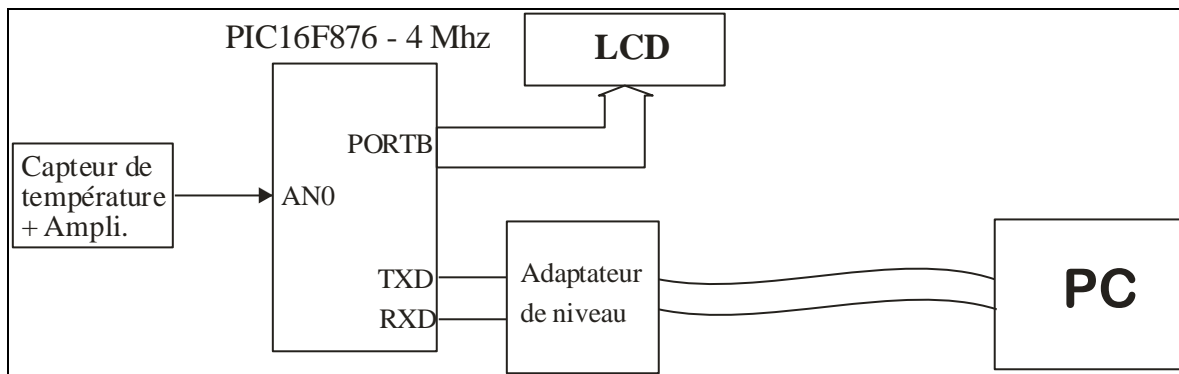


Figure 1

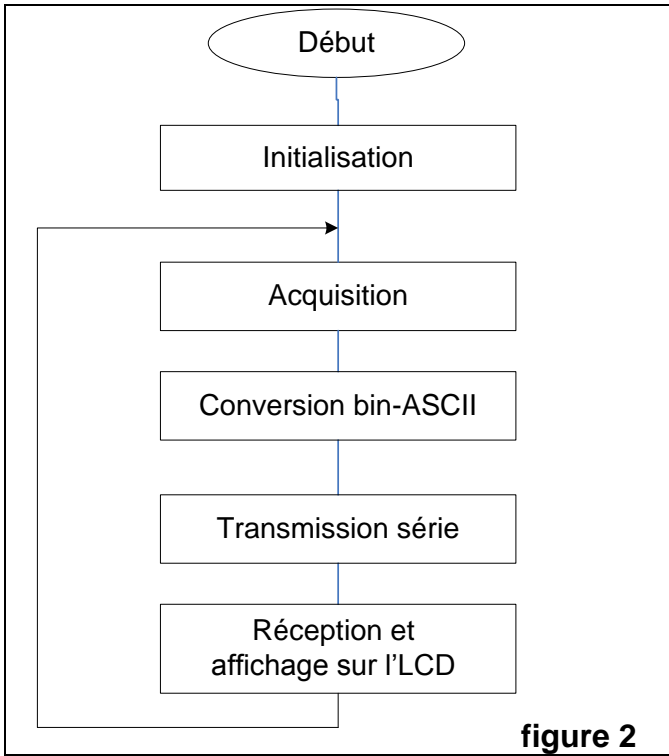


figure 2

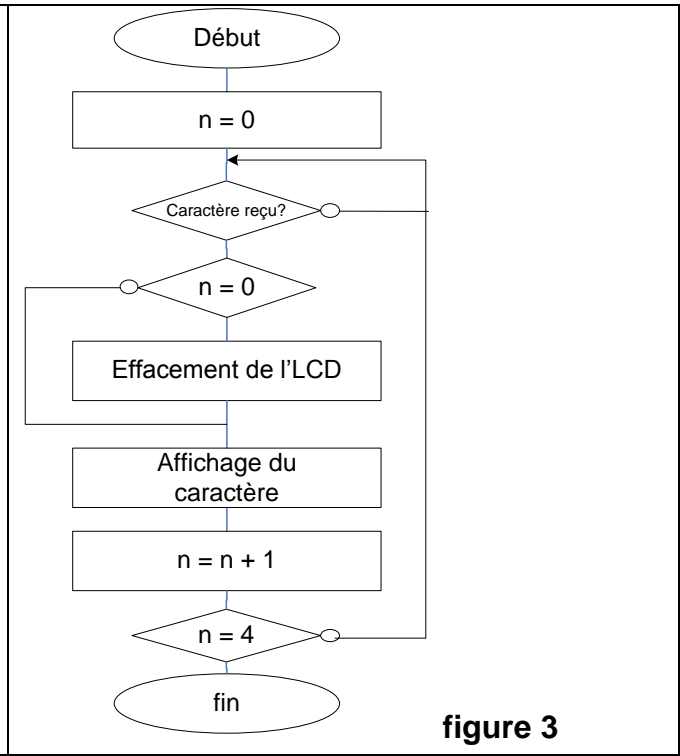


figure 3

ANNEXE

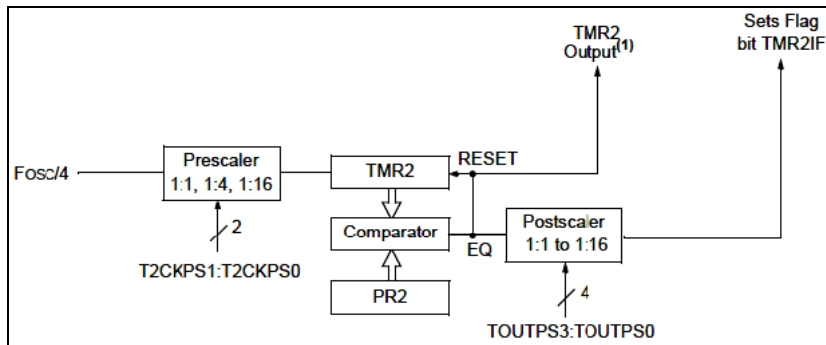


Schéma synoptique du timer 2

Registre PR2

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7:	Unimplemented: Read as '0'						
bit 6-3:	TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits						
	0000 = 1:1 Postscale						
	0001 = 1:2 Postscale						
	0010 = 1:3 Postscale						
	•						
	•						
	•						
	1111 = 1:16 Postscale						
bit 2:	TMR2ON: Timer2 On bit						
	1 = Timer2 is on						
	0 = Timer2 is off						
bit 1-0:	T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits						
	00 = Prescaler is 1						
	01 = Prescaler is 4						
	1x = Prescaler is 16						

Registre INTCON

INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
--------	-----	------	------	------	------	------	------	------

Registres PIR1 et PIE1

PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

Fonctions prédéfinies :

int read_ADC(void) : cette fonction renvoie la valeur convertie sur 8 bits

sprintf(chaine cible, chaîne de formatage, val) : convertie la valeur val en une chaîne de caractère.

putc(caractères) : envoie un caractères sur le port série

lcd_putc("\f") : efface l’LCD

lcd_putc(caractère) : écrit un caractère sur l’LCD