

Devoir surveillé

Semestre : 1 ☐ 2 ☒

Session : Principale ☒ Rattrapage ☐

Unité d'enseignement : UP embarqué

Module : Architecture des microcontrôleurs

Classe(s) : 2A, 2P

Nombre de pages : ENONCE (4 pages) + ANNEXE (1 pages)

Documents autorisés : ☐ Oui ☒ Non

Calculatrice autorisé : ☐ Oui ☒ Non

Date : 06/04/2024

Heure : 09H00

Durée : 1H

QCM (10 points) : Cocher la ou les bonne(s) réponse(s) :

1. Quand est-il possible d'accéder à un bit à partir de la banque "0" dans le mode d'adressage direct des PIC ?

- A. Uniquement lorsque le bit RPO du registre STATUS est à "zéro".
- B. Uniquement lorsque le bit RPO du registre STATUS est réglé sur "1".
- C. Uniquement lorsque le bit RPO du registre STATUS est utilisé avec les 7 bits inférieurs du code d'instruction.
- D. Aucune réponse

2. Le PIC 16F84 possède un registre de travail interne W :

- A. De taille 14 bits permettant de coder les instructions.
- B. De taille 13 bits, permet de stocker l'adresse de retour suite à l'exécution d'un sous-programme.
- C. De taille 8 bits, joue le rôle d'accumulateur pour le chargement des résultats intermédiaires.
- D. De taille 16 bits car le microcontrôleur est de famille mid range.

3. On a un programme qui nécessite 0.5 μ s (CI) pour exécuter une instruction. Déterminer la fréquence du microcontrôleur $F_{\mu c}$ et le temps nécessaire pour exécuter ce programme T_E , sachant que le nombre de cycle d'instruction NB= 250.

- A. $F_{\mu c} = 4 \text{ MHz}$; $T_E = 250 \mu s$
- B. $F_{\mu c} = 11 \text{ MHz}$; $T_E = 90 \mu s$
- C. $F_{\mu c} = 8 \text{ MHz}$; $T_E = 125 \mu s$
- D. Aucunes réponses

4. Quelle est la taille de la mémoire de programme qu'on a le droit de manipuler du pic16F84

- A. 2^{13} mots \times 14 bits
- B. 2^{10} mots \times 13 bits
- C. 1024 mots \times 14 bits
- D. 2^{10} mots \times 14 bits

5. Quelle est la fonction du registre PC (Program Counter) dans le microcontrôleur Pic 16F84?

- A. Stocker le résultat des opérations arithmétiques et logiques
- B. Stocker l'adresse de la prochaine instruction à exécuter
- C. Stocker les constantes numériques
- D. Stocker les données d'entrée

6. La réalisation d'une opération logique par le processeur de notre microcontrôleur pic16F84 a un effet sur :

- A. Le bit Carry
- B. Le bit Digit Carry
- C. Le bit Zéro
- D. Toutes les réponses sont correctes.

7. Nous avons le bout de code suivant :

```
Unsigned char A, B, Somme;  
A = 158 ;  
B = 98 ;  
Somme = A + B ;
```

Cochez-la ou les proposition(s) correcte (s) :

- A. C=0 ; Z=0 ; Somme = 256
- B. C=1 et Z=1 ; Somme = 0
- C. Elle n'affecte pas le registre STATUS
- D. C=1 et Z=1 ; Somme =256

8. Nous avons le bout de code suivant :

```
Unsigned char b=276, a=2;  
while(a<=5)  
{ a++;  
  b=b - a; }
```

Au bout de 4 itérations de la boucle while :

- A. Z =0, C=0, b= 258
- B. Z =1, C=1, b=2.
- C. Z =0, C=1, b=2
- D. Z =0, C=1, b= 258

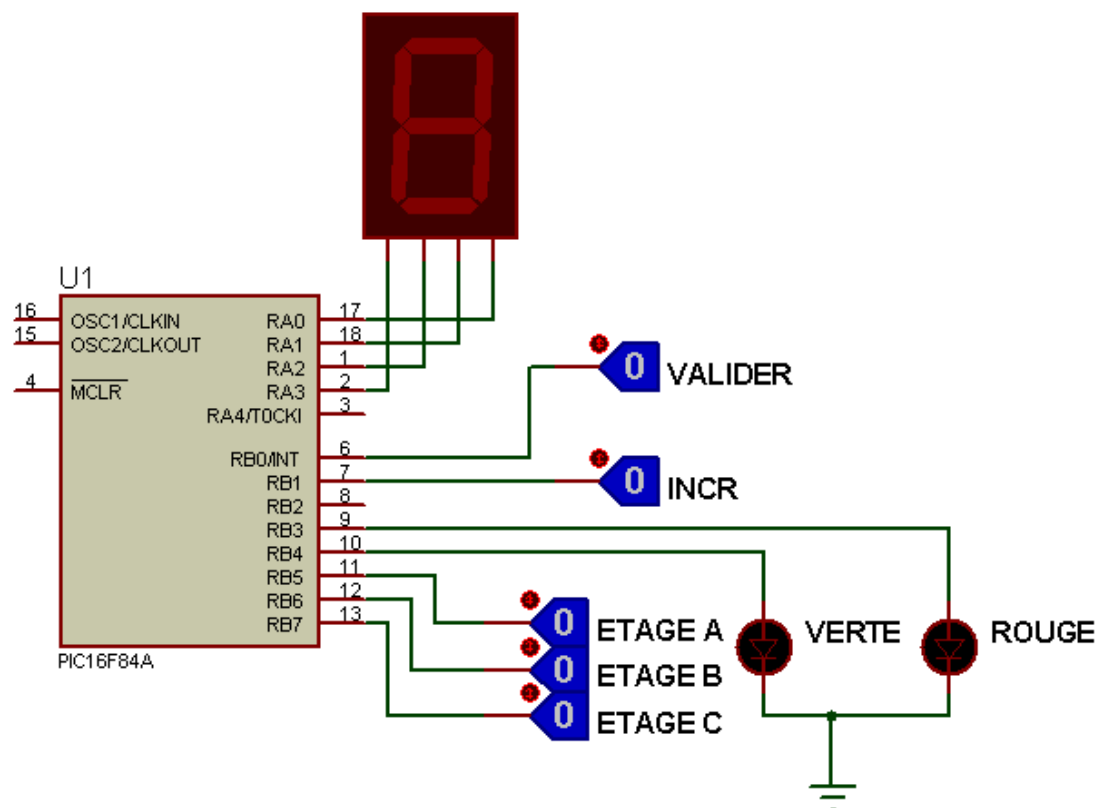
Problème (10 points) :

Dans une cabine d'ascenseur conçue pour accueillir jusqu'à 4 personnes, **la présence de ces 4 occupants est nécessaire pour démarrer**. Nous souhaitons réaliser un système de gestion intégré permettant aux utilisateurs de connaître le nombre de places disponibles avant de choisir l'étage souhaité.

Le système fonctionne à base d'un PIC16F84 et dispose de :

- Un bouton **INCRE** permettant d'incrémenter le nombre de personnes disponibles dans la cabine.
- Deux lampes **LEDs : LED_R & LED_V**, permettant d'indiquer l'état de la cabine (pleine ou pas encore).
- Trois boutons (**ETAGE_A**, **ETAGE_B** et **ETAGE_C**) permettant de sélectionner l'étage choisi.
- Un bouton **VALIDER** permettant de valider le choix de l'étage sélectionné.
- Un afficheur **BCD** indiquant :
 - Le nombre de personnes disponibles dans la cabine
 - L'étage sélectionné

Le montage ci-dessous décrit les composants du système :



Fonctionnement :

- Au départ, lorsque le système s'active, il affiche le nombre de personnes présentes dans la cabine. À ce stade, la valeur de la variable « **NbPers** » est **nulle**. La LED rouge s'allume tandis que la LED verte s'éteint, signalant ainsi qu'aucune personne n'est présente dans la cabine et qu'elle n'est pas encore pleine.
- Le bouton **INCRE** simule l'entrée d'une personne dans la cabine, ce qui entraîne l'incrément de la variable « **NbPers** » **qui est constamment affichée sur le BCD**. Si le nombre de personnes dans la cabine atteint la valeur 4, **la LED_V s'allume et la LED_R s'éteint**.
- Une fois que l'ascenseur est plein, l'une des personnes disponibles peut **sélectionner l'étage** désiré en **appuyant sur l'un des 3 boutons** (ETAGE_A, ETAGE_B et ETAGE_C). Dès qu'elle appuie sur l'un de ces boutons, cela déclenche l'affichage de la lettre correspondant à l'étage choisi. Notons que chaque appui sur un bouton envoie un signal logique de 1.
- Pour **valider ce choix**, il suffit d'appuyer sur le bouton (**VALIDER=0**). Cette action entraîne immédiatement le clignotement des deux LEDs verte et rouge 4 fois, d'une durée totale de 2 secondes. Ensuite, le système revient à son état de départ.

Travail demandé :

Réaliser le code C qui correspondent au fonctionnement du système tout en passant par les étapes suivantes :

- A. Les directives : **(1 point)**
 - Défines (0.5 point)
 - Macros (0.5 point)
- B. La fonction principale main : **(5 points)**
 - La configuration des entrées / sorties. (1 point)
 - La configuration de l'interruption. (1 point)
 - L'initialisation. (1 point)
 - Le programme principal. (2 points)
- C. La fonction d'interruption **(4 points)**
 - La sélection de l'étage (2 points)
 - La validation du choix (2 points)

NB : nous disposons d'une fonction **delay_ms()** qui assure les attentes.

BON TRAVAIL

Annexe

REGISTER 2-2: OPTION REGISTER (ADDRESS 81h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBPUP	INTEDG	T0CS	T0SE	PSA	PS2	PS0
bit 7							bit 0
bit 7	RBPUP : PORTB Pull-up Enable bit 1 = PORTB pull-ups are disabled 0 = PORTB pull-ups are enabled by individual port latch values						
bit 6	INTEDG : Interrupt Edge Select bit 1 = Interrupt on rising edge of RB0/INT pin 0 = Interrupt on falling edge of RB0/INT pin						
bit 5	T0CS : TMR0 Clock Source Select bit 1 = Transition on RA4/T0CKI pin 0 = Internal instruction cycle clock (CLKOUT)						
bit 4	T0SE : TMR0 Source Edge Select bit 1 = Increment on high-to-low transition on RA4/T0CKI pin 0 = Increment on low-to-high transition on RA4/T0CKI pin						
bit 3	PSA : Prescaler Assignment bit 1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module						
bit 2-0	PS2:PS0 : Prescaler Rate Select bits						
	Bit Value	TMR0 Rate	WDT Rate				
	000	1 : 2	1 : 1				
	001	1 : 4	1 : 2				
	010	1 : 8	1 : 4				
	011	1 : 16	1 : 8				
	100	1 : 32	1 : 16				
	101	1 : 64	1 : 32				
	110	1 : 128	1 : 64				
	111	1 : 256	1 : 128				

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE	EEIE	T0IE	INTE	RBIE	T0IF	RBIF
bit 7							bit 0
bit 7	GIE : Global Interrupt Enable bit 1 = Enables all unmasked interrupts 0 = Disables all interrupts						
bit 6	EEIE : EE Write Complete Interrupt Enable bit 1 = Enables the EE Write Complete interrupts 0 = Disables the EE Write Complete interrupt						
bit 5	T0IE : TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 interrupt 0 = Disables the TMR0 interrupt						
bit 4	INTE : RB0/INT External Interrupt Enable bit 1 = Enables the RB0/INT external interrupt 0 = Disables the RB0/INT external interrupt						
bit 3	RBIE : RB Port Change Interrupt Enable bit 1 = Enables the RB port change interrupt 0 = Disables the RB port change interrupt						
bit 2	T0IF : TMR0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow						
bit 1	INTF : RB0/INT External Interrupt Flag bit 1 = The RB0/INT external interrupt occurred (must be cleared in software) 0 = The RB0/INT external interrupt did not occur						
bit 0	RBIF : RB Port Change Interrupt Flag bit 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software) 0 = None of the RB7:RB4 pins have changed state						