

partie 1 : 7/10
2 : 7,25/20
14,25/20

Année 2015/2016

A remplir obligatoirement par l'enseignant responsable du contrôle

Date : 07 janvier 2016

Contrôle de : Architecture des ordinateurs

Durée : 2 heures

Professeur responsable : N.ABOUCHI

Documents : ☒ autorisés ☐ non autorisés

Si oui : type(s) de documents autorisés :

Calculatrices alphanumériques : ☒ autorisées ☐ non autorisées

REPONDRE SUR LE SUJET : ☒ OUI NON

LES TELEPHONES PORTABLES ET AUTRES APPAREILS DE STOCKAGE DE DONNEES NUMERIQUES NE SONT PAS AUTORISES

A l'attention des élèves : rappels importants sur la discipline des examens

La présence à tous les examens est strictement obligatoire ; tout élève présent à une épreuve doit rendre une copie, même blanche, portant son nom, son prénom et la nature de l'épreuve.

Toute absence non justifiée est sanctionnée par un zéro.

Toute fraude ou tentative de fraude avérée est sanctionnée par un zéro à l'épreuve et portée à la connaissance de la direction des études qui pourra réunir le Conseil de Discipline. Les sanctions prises peuvent aller jusqu'à l'exclusion définitive du (des) élève(s) mis en cause.

Toute suspicion sur la régularité et le caractère équitable d'une épreuve est signalée à la direction des études qui pourra décider l'annulation de l'épreuve; tous les élèves concernés par l'épreuve sont alors convoqués à une épreuve de remplacement à une date fixée par le responsable d'année.

Exercice 1 : Généralités sur le 8051 (4 pts) plusieurs réponses possibles, le point est affecté uniquement si toutes les réponses sont exactes.

Question 1 (1 pt) : Le microprocesseur 8051:

- ☐ Est un microprocesseur 8 bits
- ☒ Est un microprocesseur de type Harvard
- ☒ Dispose de deux registres d'adresses
- ☒ Dispose d'un espace mémoire pour le code séparé de l'espace mémoire pour les données
- ☒ Dispose d'un seul bus d'adresses

Question 2 (1 pt) : les registres internes du 8051

- ☐ Sont au nombre de 32 (R0, R1, R2, R3, R4,, R31)
- ☐ Sont localisés dans la mémoire code
- ☐ Sont localisés dans la RAM externe
- ☐ Sont localisés dans l'espace mémoire SFR
- ☒ Sont au nombre de 32 (4 fois [R0, R1, R2, R3, R4, R5, R6, R7])

1

Question 3 (1 pt) : gestion de la pile

- ☐ Les instructions CALL et JMP sont équivalentes
- ☐ L'instruction PUSH place un octet dans la pile et incrémente de 2 le registre SP
- ☐ L'instruction CALL provoque la mémorisation automatique du registre pointeur de pile (SP) dans la pile et incrémente de 2 le registre SP
- ☒ Le registre SP est automatiquement incrémenté d'un octet suite à l'instruction JMP
- ☐ La pile est forcément une mémoire de type EPROM

Question 4 (1 pt) : modes d'adressages

- ☒ Seul l'adressage indirect est possible pour les accès en mémoires de données externes
- ☒ L'adressage immédiat ne permet pas d'écrire dans la mémoire code
- ☒ L'instruction CALL utilise l'adressage direct
- ☒ L'instruction JMP utilise l'adressage direct
- ☒ L'écriture dans les registres R0 à R7 n'utilise pas forcément l'adressage immédiat

Exercice 2 : Notion de programmation du 8051F020 (6 points)

- Le 8051F020 utilise les deux registres pointeur de code PC et pointeur de données DPTR pour accéder aux mémoires externes. Expliquer par l'instruction MOVX @DPTR, A comment et quand est-ce que ces deux registres sont utilisés et pourquoi ?

DPTR est un registre sur 2 octets permettant l'accès à toutes les adresses de la mémoire externe (0000H à FFFFH).

L'instruction MOVX @DPTR, A permet de copier la valeur contenue dans A dans la case de pointeur d'adresse la valeur contenue dans le DPTR. Cette case se situe en mémoire externe.

- Que fait l'instruction MOV 00, #23H ? quels modes d'adressages sont utilisés pour les opérandes source et destination ?

Le registre situé à l'adresse 00H est R0.

L'instruction place dans la valeur 23H dans le registre R0.

#23H → mode d'adressage immédiat.

00 → mode d'adressage direct.

- Ecrire en assembleur 8051 les programmes suivants :

- Copier le contenu de la mémoire 1234H (Espace XDATA) dans le registre DPH.

MOV DPTR, #1234H

→ place la valeur 1234H dans le DPTR

MOV A, @DPTR

→ copie le contenu de la mémoire 1234H dans Acc

MOV DPH, A

→ copie la valeur de Acc dans DPH.

- Copier le contenu de la mémoire 0002H (Espace code) dans la mémoire 1234H (Espace XDATA).

MOV DPTR, #0002H

→ place la valeur 0002H dans le DPTR

MOV A, #00H

→ met l'accumulateur à 0

MOVC A, @A+DPTR

→ copie le contenu de la mémoire 0002H dans Acc

MOV DPTR, #1234H

→ place la valeur 1234H dans le DPTR

MOVX @DPTR, A

→ copie le contenu de Acc dans la mémoire 1234H

Gestion de la pile

Pour le programme suivant, représentez la pile (contenus et adresses) à 2 moments différents d'exécution du code (complétez les croquis ci-dessous).

Remarque: ce programme ne fait rien de censé, il ne sert qu'à nous permettre de vérifier votre compréhension de la mise en œuvre de la pile.

```

89  ;*****
90  Prog_biden:
91      mov sp,#40H
92      ; ...
93  bcl:   nop
94      call sp1
95      ; ...
96      jmp bcl
97  ;*****
98  sp1:   push ACC
99      push 10
100      ; ...
101      call sp2
102      ; ...
103      pop 10
104      pop ACC
105      ret
106  ;*****
107  sp2:   push DPH
108      push PSW
109      ; ....
110      pop PSW
111      pop DPH
112      ret
113  ;*****

```

Contenu de la pile et valeur de SP juste avant l'exécution de la ligne 109

Mémoire IDATA

Adresses: Contenus:

[illegible]

SP=

Contenu de la pile et valeur de SP juste avant l'exécution de la ligne 96

Mémoire IDATA

Adresses: Contenus:

[illegible]

SP=

; ... cette ligne signifie qu'il peut y avoir plus de code, mais qu'il n'affecte pas la pile

Indiquez le contenu de chaque élément de la pile sous la forme d'un nom de registre, d'une adresse, d'une partie d'adresse, etc....

Address	Content
40	94
41	95
42	ACC
43	10
44 44	102
45	DPH
46	PSW

$$\bar{A} \log \Rightarrow SP = 47$$

Addresses	contents
40	91
41	95
42	Acc
43	10
44	102
45	DPH
46	PSW
47	

$\hat{A}96 \Rightarrow SP = 40$

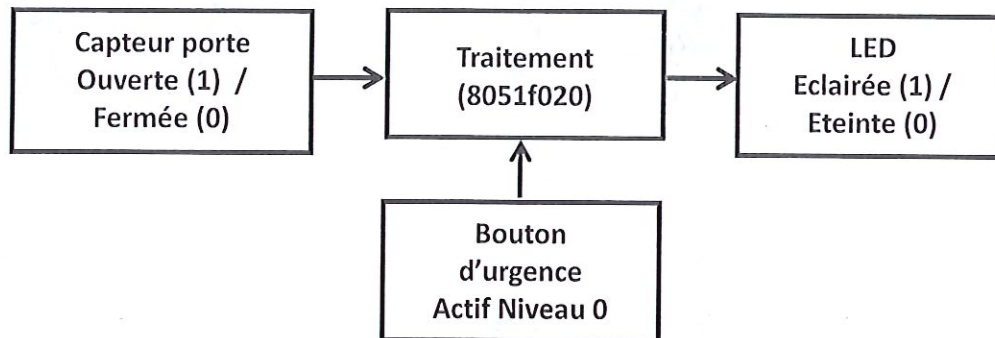
Problème : Codage en C et configuration de périphériques du 8051F020 (10 pts)

On souhaite utiliser un système basé sur le microcontrôleur 8051F020 pour surveiller l'accès à un local de stockage. L'état de la seule porte d'accès au local est en permanence affiché sur une LED. La LED est éclairée si la porte du local est ouverte et éteinte si la porte est fermée.

Le fonctionnement du local est le suivant :

- Le local est normalement fermé (LED éteinte),
- Si la seule porte d'accès au local est ouverte, la LED passe en mode éclairé,
- Si la porte reste ouverte pendant plus que 100 secondes, le LED clignote (100ms éteinte, 100 ms allumée)
- La LED reste en mode clignotement jusqu'à ce que la porte soit fermée.
- L'appui sur le bouton d'urgence place automatiquement la porte en position ouverte, par un mécanisme à ne pas considérer dans le cadre de ce problème, dans ce mode, la LED clignotera par les allumages brefs (100ms allumée, 1s éteinte).
- On sortira du mode « Urgence » par un redémarrage du système.

Schéma de principe de la réalisation :



Mise en œuvre :

- Le capteur d'état de la porte **Etat_porte** sera relié sur le port **P1.1**. Un niveau 0 signalera l'état « porte fermée »
- La LED **LED** sera reliée sur le port **P1.6**. Elle s'allumera avec un niveau haut
- Le bouton d'urgence **Urgence** sera relié sur l'entrée d'interruption **INT7 (P3.7)**. Ce bouton est actif sur un niveau bas.

L'objectif du problème est d'écrire un programme permettant de:

- a. scruter en permanence l'état de la porte
- b. mettre à jour l'état de la LED
- c. vérifier que la durée de la porte ouverte soit inférieure à 100 secondes.
- d. gérer les divers modes de clignotement de la LED
- e. gérer le mode « Urgence ».
- f. Mis à part le mode Urgence, toute la gestion du dispositif sera assurée par une interruption produite par un timer.

Questions :

Question 1 – Configuration globale du microcontrôleur (1 point)

Pour cette application, quels sont les éléments de configuration globale à gérer ?
Citez-les, expliquez. (Nous ne demandons pas de code dans cet exercice).

- Les horloges → pour avoir une base de temps de 10ms

- L'affectation des broches aux entrées sorties (GPIO).

↳ - capteur de porte sur P1.1

- LED sur P1.6

- Bouton d'urgence sur P3.7

Question 2 – Configuration des broches d'entrée-sortie et de l'entrée d'interruption INT7 (2 points)

Configuration de la broche Etat_Porte (P1.1) en entrée.

Expliquez ce que vous devez faire

Nommer les registres sur lesquels vous devez agir et pourquoi.

- configurer le port en Drain ouvert

- écrire un "1" sur le port

Il faut agir sur : P1MDOUT → mode Drain ouvert

- P1^1

→ écrire un "1" sur le port

Configuration de la broche LED (P1.6) en sortie. Cette broche fait allumer un LED lorsque que la broche délivre un niveau haut. (Courant LED = 15mA).

Expliquez ce que vous devez faire

Nommer les registres sur lesquels vous devez agir et pourquoi.

Il faut configurer le port P1.6 en mode Push Pull

Il faut agir sur P1MDOUT → mode Push Pull

Coder la fonction `Config_GPIO()`. Cette fonction assurera la configuration des broches P1.1, P1.6 et P3.7. Ce code ne devra pas altérer la configuration des ports autres que P1.1, P1.6 et P3.7.

`P1MDOUT |= 0x40;`

// P1.6 en mode Push Pull

`P1MDOUT &= 0xFE;`

// P1.1 en mode Drain ouvert

`P1^1 = 1;`

// Écrit "1" sur P1.1

`P3MDOUT &= 0x7F;`

// P3.7 en mode Drain ouvert

`P3^7 = 1;`

// Écrit "1" sur P3.7

Question 3 – Configuration d'un Timer (3 points)

Pour assurer la gestion du système, on fera en sorte de provoquer une interruption toutes les 10ms. Cette interruption sera produite par un Timer.

C'est dans ce programme d'interruption que l'on exécutera la lecture de l'état de la porte et le pilotage de la LED. Pour cet exercice, nous vous imposons l'utilisation du Timer3.

Expliquez le fonctionnement du timer dans cet exercice.

Le timer utilise l'horloge interne SYSClk pour incrémenter le registre TMR3 réparti sur 2 octets (TMR3L et TMR3H). Lorsque TMR3 vaut 0xFFFF, une interruption est générée sur TF3. Les registres TMR3L et TMR3H prennent respectivement les valeurs de TMR3RL et TMR3RH comme valeurs d'initialisation.

Inventoriez tous les registres sur lesquels vous allez devoir agir pour programmer le timer3 afin qu'il génère une interruption toutes les 10ms.

Pour chaque registre, indiquez son rôle.

TMR3CN → TF3 : bit interruption

TR3 : marche/arrêt du timer

T3XCLK : choix de l'horloge

T3M : division de l'horloge SYSClk au 12 ou pas.

TMR3RL → octet de poids faible de la valeur de réinitialisation

TMR3RH → octet de poids fort de la valeur de réinitialisation

TMR3L → octet de poids faible de la valeur de démarrage du timer

TMR3H → octet de poids fort de la valeur de démarrage du timer

EIE2 → ET3 : Autoriser l'interruption

Codez la fonction Config_Timer3(). Cette fonction configure le Timer3 pour générer une interruption toutes les 10ms (on considère que le 8051F020 fonctionne avec une fréquence SYSClk de 22,1184 MHz)

$$\frac{SYSClk}{12} = \frac{1843200 \text{ Hz}}{12} = 153600 \text{ Hz} \quad \frac{1}{153600} = 6.5 \mu s \quad \frac{10 \times 10^{-3}}{6.5 \times 10^{-6}} = 1538 \rightarrow \text{valeur de TMR3RL}$$

TMR3CN = 0x00; // TF3 = 0, TR3 = 0, T3M = 0, T3XCLK = 0

TMR3RL = 0x00;

TMR3RH = 0x48;

TMR3L = 0x00;

TMR3H = 0x48;

TMR3CN = 0x04; // TR3 = 1 → lance le timer

EIE2 = 0x01; // ET3 = 1 → autorise interruption timer 3.

Question 4 – Configuration de l'interruption externe INT7 (P3.7). (1 point)

Expliquez le mode retenu pour le fonctionnement de INT7.

Coder la fonction Config_INT7()

EIE2 = 0x20; // EX7 = 1 → Autorise les interruptions sur INT7

et P3IE ?

Question 5 – Programme d'interruption du Timer 3 (3 points)

6/35

C'est dans ce programme que toute la gestion de l'application est faite (la boucle infinie dans la fonction **main** est vide).

Ecrire sous forme de pseudo-code ou d'algorithme ce programme d'interruption.

```

if (P1^1 == 1 && cpt < 10000)
{
    P1^6 = 1;
    cpt++;
}

if (cpt >= 10000)
{
    if (cpt2 < 10)
    {
        cpt2++;
    }
    else
    {
        P1^6 = 1; // complémente l'état de la LED
        cpt2 = 0;
        cpt = 0;
    }
    if (P1^1 == 0)
    {
        P1^6 = 0;
    }
}

```

Programme interruption INT2:

```

if (cpt3 < 10)
{
    cpt3++;
}
else
{
    P1^6 = 1;
    cpt3 = 0;
}

```

Question 6 – Amélioration du dispositif. (Bonus 2 points)



Pour améliorer le dispositif, on souhaite envoyer l'information à distance à l'aide d'une liaison série asynchrone. Expliquez comment vous pourriez procéder. Citez les paramètres dont vous pourriez avoir besoin.

Quels sont les périphériques à mettre en œuvre pour cette application.

Expliquez les configurations à faire et comment vous comptez utiliser ce périphérique.

Il faudrait agir sur les registres : on

~~SCCR0~~ :

KKCON : Control de l'halage

TYOD : Choix du mode du Timer

TP1 : Flag du timer 1

RCLK : # Source de l'horloge de réception

TCLK : Source de l'horloge d'émission

PCON : Choix de la fréquence de transmission

SCCR0 : Registres de Contrôle de l'UART 0.

pour faire quoi ?