



Année universitaire 2016/2017

NOM

Consignes relatives au déroulement de l'épreuve

A remplir obligatoirement par l'enseignant responsable du contrôle

Date : 16 novembre 2016

Devoir Module Bases des systèmes embarqués – Partie TP

Durée: 30mn

Professeurs responsables : François Joly – Yasmina Layouni

Documents : ☒ autorisés ☐ non autorisés

Si oui : type(s) de documents autorisés : Polycopié Fiche technique du 8051F020 uniquement

Calculatrices : ☒ autorisées ☐ non autorisées

Si oui : type(s) de calculatrices autorisées : alphanumériques

ASS

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

Les téléphones portables doivent être éteints pendant toute la durée de l'épreuve et rangés dans les cartables.

S'agissant de contrôle sans document, les trousseaux doivent être rangés dans les cartables.

Les cartables doivent être fermés et posés au sol.

Les oreilles des candidats doivent être dégagées.

Rappels importants sur la discipline lors 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.

Une absence non justifiée à un examen invalide automatiquement le module concerné.

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.

Toute fraude ou tentative de fraude est 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.

Merci d'indiquer sur chaque page votre nom et votre prénom

Devoir Session 1 - BSE - Partie TP

Durée recommandée : 30mn

Questions relatives au cahier des charges : « Module de surveillance de container »

Remarque : Les questions peuvent être traitées indépendamment les unes par rapport aux autres.

Contexte.

Un programmeur « débutant » a été chargé de générer un code de configuration du microcontrôleur, ainsi que des codes de configuration et d'utilisation d'un périphérique de transmission série asynchrone (pour la liaison Processeur Mesures / Processeur Transmission). Le code qui a été produit est présenté à la fin de ce document sur 1 page à détacher.

Nous allons vous demander de vérifier si ce code correspond bien au cahier des charges, et de procéder à des modifications si nécessaire.

1. Question – Fonction de configuration Timers_Init() – 4 points.

Quel reproche pourriez-vous faire sur la manière de coder cette fonction (plus particulièrement sur la manière d'initialiser CKCON, TMOD et TCON) ?

Ré-écrivez ces 3 lignes de code de manière plus « rigoureuse »

Le code utilise = pour affecter les valeurs. Cela signifie que l'on va modifier tout le registre, et pas seulement les bits qu'on souhaitait modifier.

CKCON |= 0x30;

TCON |= 0x40;

TMOD |= 0x02;

TMOD &= ~0x01; // si on souhaite mettre le timer en mode 2

2. Question – Configuration de la liaison asynchrone– 8 points. (5)

En analysant les fonctions `Timers_Init` et `UART_Init`, on peut constater que cette configuration ne respecte pas du tout le cahier des charges.

Indiquez en le justifiant quels sont les périphériques qui ont été configurés, quels sont les modes de fonctionnement, et quels sont les paramètres de transmission (vitesse, parité, nombre de bits etc....).

On a configuré l'UART0 en mode 1 (8 bits, variable Band Rate) avec un 6.5 de stop

On a activé les timers 1 et 2
timer 1: Mode 2: Compteur 8-6+ à rechargement automatique
Démarage à 0xFE0

timer 2: Utilisé pour l'horloge de l'UART0

Rechargement: 0xA3A3 = (41891)₁₀

On peut en déduire la vitesse de l'UART0: ^{Transmission}

$$\text{band rate} = \text{SYSCLK} / ((65536 - [\text{RCAP2H}, \text{RCAP2L}]) * 32)$$

avec $\text{SYSCLK} = 18,432 \times 10^6$

$$= 24,3 \text{ bauds}$$

3. Question – Gestion des interruptions– 4 points.

1,5

Si l'on décidait d'exécuter ce code tel quel, quelle remarque pourriez-vous faire sur la gestion des interruptions ?

Par d'ISR dans le code ? → et donc...
On ne configure pas les interruptions des timers

4. Question – Résultat de la compilation et de l'édition de liens– 4 points.

A l'issue des opérations de compilation de ce fichier et de l'édition de lien, on obtient les messages « Warnings » suivants :

```
Rebuild target 'Target 1'
assembling STARTUP.A51...
compiling Code_Exercice_TP_Session1.c...
Code_Exercice_TP_Session1.c(94): warning C206: 'CAN_Init': missing function-prototype
linking...
*** WARNING L16: UNCALLED SEGMENT, IGNORED FOR OVERLAY PROCESS
SEGMENT: ?PR? PUTCHAR?CODE_EXERCICE_TP_SESSION1
*** WARNING L16: UNCALLED SEGMENT, IGNORED FOR OVERLAY PROCESS
SEGMENT: ?PR?GETKEY?CODE_EXERCICE_TP_SESSION1
*** WARNING L1: UNRESOLVED EXTERNAL SYMBOL
SYMBOL: CAN_INIT
MODULE: Code_Exercice_TP_Session1.obj (CODE_EXERCICE_TP_SESSION1)
*** WARNING L2: REFERENCE MADE TO UNRESOLVED EXTERNAL
SYMBOL: CAN_INIT
MODULE: Code_Exercice_TP_Session1.obj (CODE_EXERCICE_TP_SESSION1)
ADDRESS: 0040H
Program Size: data=20.0 xdata=0 code=130
"code_session1" - 0 Error(s), 5 Warning(s).
Build Time Elapsed: 00:00:01
```

4

Expliquez les warnings L16, L1 et L2. Peut-on passer outre, ou pas ?

L16: une fonction n'est pas utilisée, on peut passer outre

L1: on fait référence à un symbole non reconnu.
cela va provoquer des problèmes à l'exécution

De même pour L2

```

1  #include "C8051F020.h"
2
3  void Reset_Sources_Init()
4  {
5      WDTCN      = 0xDE;
6      WDTCN      = 0xAD;
7  }
8  void Port_IO_Init()
9  {
10     XBR0        = 0x04;
11     XBR2        = 0x40;
12 }
13 void Oscillator_Init()
14 {
15     int i = 0;
16     OSCXCN      = 0x67;
17     for (i = 0; i < 3000; i++); // Wait 1ms for initialization
18     while ((OSCXCN & 0x80) == 0);
19     OSCICN      = 0x0C;
20 }
21
22 void Timers_Init()
23 {
24     CKCON        = 0x30; // Config T1M et T2M
25     TCON         = 0x40; // Config Timer1
26     TMOD         = 0x20; // Config Timer1
27     TL1          = 0xF0;
28     TH1          = 0xFE;
29     T2CON        = 0x14;
30     RCAP2L       = 0xA3;
31     RCAP2H       = 0xA3;
32 }
33 void UART_Init()
34 {
35     SCON0        = 0x70;
36     IE           = 0x10;
37 }
38
39 char putchar (char c) {
40     while (!TIO);
41     TIO = 0;
42     return (SBUF0 = c);
43 }
44
45 char getkey ()
46 {
47     char c;
48     while(!RIO);
49     c = SBUF0;
50     RIO = 0;
51     return (c);
52 }
53
54 void Init_Device(void)
55 {
56     Reset_Sources_Init();
57     Port_IO_Init();
58     Oscillator_Init();
59 }
60
61 //*****
62 // MAIN
63 //*****
64 main()
65 {
66     Init_Device();
67     Timer_Init();
68     UART_Init();
69     CAN_Init();
70     TIO = 1;
71     EA = 1;
72
73     while(1)
74     {
75     }
76 }
77
78 //*****
79 //
80 //*****

```