



Année universitaire 2020/2021

NOM :

PRENOM :

Consignes relatives au déroulement de l'épreuve

Date : 20 janvier 2022

Devoir Module Bases des systèmes embarqués – Session 1 - **Partie TP**

Durée: 1H00

Professeurs responsables : François Joly

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

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.

Devoir Session 1 – BSE - Partie TP

Durée de l'épreuve : 1H00

Contexte : Analyse de code.

Dans cette partie « TP » il va vous être demandé d'analyser un code qui a été produit pour répondre à la question 4 de l'examen « Cours ».

Le code à analyser est en annexe sur 3 pages. Ces 3 pages sont détachables pour vous faciliter la tâche.

Nous reproduisons ci-dessous, l'énoncé de cet exercice:

Fonctionnalité Comptage d'évènements (5 points).

Dans cet exercice, on se propose de mettre en place une solution de comptage d'évènements dans notre microcontrôleur.

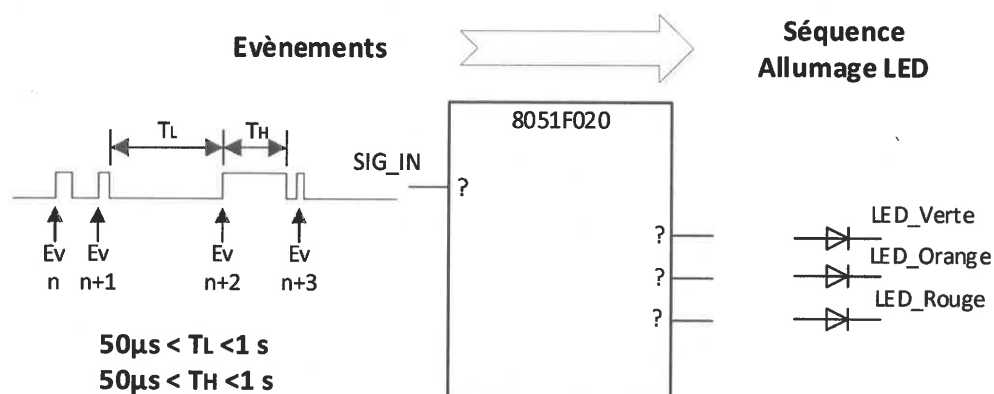
Ces « évènements » (le signal SIG_IN) arrivent au microcontrôleur sous forme d'impulsions positives de durée et de récurrence variables. Le niveau de tension est un niveau 0-3,3V.

En fonction du nombre d'évènements comptés, on pilotera 3 LED de couleur différentes : LED_Verte, LED_Orange et LED_Rouge. Le cycle de comptage sera le suivant :

- 0 à 4999 évènements LED_Verte Cligno - LED_Orange éteinte - LED_Rouge éteinte
- 5000 à 5999 évènement LED_Verte allumée - LED_Orange éteinte - LED_Rouge éteinte
- 6000 à 6999 évènement LED_Verte éteinte - LED_Orange Cligno - LED_Rouge éteinte
- 7000 à 7999 évènement LED_Verte éteinte - LED_Orange allumée - LED_Rouge éteinte
- 8000 à 8999 évènement LED_Verte éteinte - LED_Orange éteinte - LED_Rouge Cligno
- 9000 à 9999 évènement LED_Verte éteinte - LED_Orange éteinte - LED_Rouge allumée

En mode « Cligno » la LED s'allume pendant 50ms, puis s'éteint pendant 50ms.

Ce mode de comptage d'évènements se reproduit à l'infini. Après le 9999 ième évènement, on repart à zéro.



QUESTIONS (Chaque question vaut 1 point)

1. Qu'est censé contenir le fichier c8051F020.h ?

Ce fichier contient toutes les déclarations de SFR des registres de l'espace mémoire SFR (Registre de périphériques) ainsi que toutes les déclarations des bits des divers registres de l'espace SFR qui sont accessibles bit à bit et ceci pour le cas de l'ICP 8051F020.

2. Compte tenu des codes de configuration, quelle est la fréquence de l'horloge système (SYSCLK) obtenue ? Justifiez.

Code de config. relative à la SYSCLK \Rightarrow OSCICN: 0x07 (0000 0111)
 \rightarrow CLKSEL=0 \rightarrow Oscillateur interne utilisé comme SYSCLK
 \rightarrow IOSCN=1 \rightarrow OK interne valide
 \rightarrow IFCN 1-0 = 11 \rightarrow oscillateur interne à 16 MHz
SYSCLK = 16 MHz

3. Est-il possible d'avoir une horloge système plus précise ? Si oui, expliquez comment vous pourriez procéder et sur quelle partie du code vous agiriez.

possible de la SYSCLK actuelle : 12,5 à 19,2 MHz
 pour être plus précis : utilisation d'un oscillateur externe \Rightarrow soit 1 quartz, soit 1 oscillateur à quartz
 \rightarrow action sur OSCICN et OSCXCN.
 Procédure :
 - Désactiver l'oscillateur interne (agir sur OSCXCN)
 - Attendre 1-2 / Tester la stabilité de l'oscillateur (flag de OSCXCN)
 - Si stabilité alors basculer sur oscillateur externe (action sur OSCXCN)

4. Donnez les affectations de broches PIO liées à la configuration du CROSSBAR.

* BA0 = 0x06 = 0000 0100 \Rightarrow UART0 valide
 * BA1 = 0x07 = 0000 0101 \Rightarrow SYSCLK, TLEX, TRF, INT1E, INT0E valides
 * BA2 = 0x08 = 0000 1000 \Rightarrow CROSSBAR valide, TLEX, TRF valides

PO.0 \Rightarrow UART0 TX	PO.6 = TRF
PO.1 \Rightarrow UART0 RX	PO.7 = TLEX
PO.2 \Rightarrow INT0	PI.0 = SYSCLK
PO.3 \Rightarrow INT1	PI.1 = GPIO à partir de cette broche
PO.4 = TRF	
PO.5 = TLEX	

5. Quel est le rôle de la fonction Reset_Sources_Init dans ce code ?

Cette fonction initialise le chip de gauche.

6. Quelle est la différence entre une déclaration de type « bit » et une déclaration de type « sbit » ?

bit \Rightarrow déclaration d'une variable booléenne - cette variable sera placée dans le minimum accessible bit à bit de la RAM du 486. Son adresse sera déterminée par l'éditeur de bin.

sbit \Rightarrow fait référence à 1 bit particulier d'un registre accessible bit à bit de l'espèce SFR. Gard en tête une variable sbit on donne l'adresse à laquelle elle se trouve.

7. Expliquez la signification de la ligne « sfr16 RCAP2 = 0xca; » ? Si on l'omet, par quoi devra-t-elle être remplacée dans le code ?

\Rightarrow sfr16 permet de manipuler 2 registres d'adresse contigus de l'espèce SFR comme 1 variable 16 bits.

RCAP2 = 0xca; permet d'adresser les registres d'adresse 0xca et 0xcb comme 1 variable 16 bits.

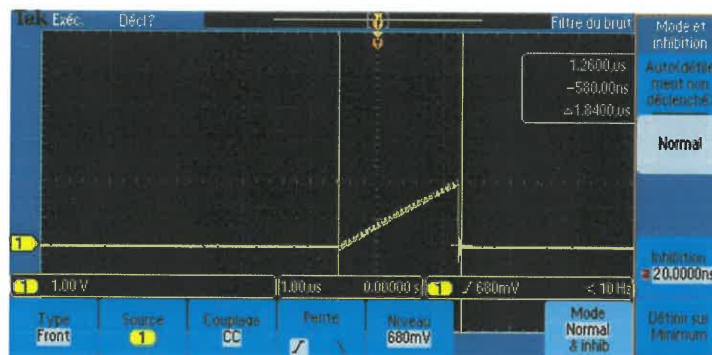
Sans cette déclaration on était alors manipulé séparément les registres d'adresse 0xca (RCAP2L) et 0xcb (RCAP2H).

8. Que se passe-t-il si on omet la ligne « EA = 1; » présente dans la fonction main ?

Si on omet « EA = 1 » alors les interruptions ne sont pas autorisées et l'application ne peut fonctionner correctement.

Rôle du bit EA : si EA = 1 alors les interruptions sont autorisées à la condition que localement elles le sont aussi.

9. On a cherché à visualiser le signal VISU_TRT_D sur un oscilloscope. L'allure du signal obtenu est la suivante : Est-ce normal ? Expliquer, et proposer une solution pour corriger le problème.



Ampli faible (1V) et allure de début d'exponentielle.
Le signal du signal \Rightarrow y a qd la broche n'est pas

configurée en push pull (elle est configurée en drain ouvert).

Solution : modifier la config de la broche et la placer en push pull.

\Rightarrow VISU_TRT_D sur la broche P2.5 \Rightarrow Actu sur P2.4 DOUT

\Rightarrow Code à joindre P2.4 DOUT |= (1 << 5);

Comme son nom semble l'indiquer, la fonction `Config_Timer_TimeBase` permet de configurer un périphérique Timer pour fonctionner en base de temps et donc produire des interruptions périodiquement.

10. Quel est le timer utilisé dans cette fonction `Config_Timer_TimeBase`?

Il s'agit du TIMER 2 (action sur Register `TRCON`, `RCAP2L`, `RCAP2H` (via `RCAP19`) et `TL2/TH2` (via `T2`)).

11. Dans quel mode est-il configuré ?

$CLK\phi = TCLK\phi = \phi$ ET $TRC = 1$ ET $CP/RCL = 0$ ET $CT2 = 0$ 16 bit Timer avec Auto-rechargement

12. Compte tenu de la configuration de ce Timer, exprimer la fréquence des interruptions en fonction de la fréquence de la `SYSCLK`

Mode TIMER
 $CLK\text{ Timer} = SYSCLK / 16$
 $RCAP = 0x\text{CBEB} = 52203$
 $F_{INT} = \frac{CLK\text{TIMER}}{RCAP} = \frac{SYSCLK / 16}{52203} = 65536 - 52203 = 13333$
 $F_{INT} = \frac{SYSCLK}{159996}$

Avec $F_s = 16\text{MHz}$, $\rightarrow F_{INT} = 100\text{Hz}$

13. Compte tenu de la configuration de ce Timer ; quel est l'évènement susceptible de déclencher une interruption ?

Mode Auto-reload
 \rightarrow le Timer 2 peut provoquer 1 int Timer 2 à cause de 2 évènements :
 - "CAPTURE" (Flag `EXT2`) \Rightarrow on a cette cfg `EXT2 = 0` \Rightarrow donc pas possible
 - OVERFLOW (Flag `TF2`)

Le seul évènement qui peut déclencher 1 interruption est l'overflow (Flag `TF2`)
 Parce que le Timer passe de sa valeur max (`0xFFFF`) à sa valeur de rechargement

14. Dans le code, quelle est la fonction d'interruption correspondant à la mise en œuvre de ce Timer ? Justifiez.

Le vecteur d'interruption pour le Timer 2 à le numéro 5

C'est donc la fonction `TIM2_C` qui est la fonction d'interruption attachée au Timer 2

On s'intéresse désormais à la fonction `Config_B`

15. Quel est le périphérique mis en œuvre dans cette fonction ?

Activa du `T4CON`, `RCAPL4` \Rightarrow on met en œuvre le `Timer 4`

16. Donner des précisions sur la configuration de ce périphérique et donc sur le fonctionnement attendu.

`T4CON = 0x01 = 0000 0010 \Rightarrow`

`RCCLK = TCLK = 0` // `FXEN4 = 0 \Rightarrow signaux T4EX ignorés`

`E/Th = 1 \rightarrow Mode Compteur`

`CP/AL4 \Rightarrow Mode auto-rechargement`

Mode Compteur 16 bits en auto-rechargement

il compte sans interruption de 65536 à 65535

\Rightarrow Comptage de 1000 événements itératifs

17. Ce périphérique va-t-il pouvoir produire des interruptions ? Justifiez.

`EIFE2 = xxx x1xx \Rightarrow interrupt-Timer 4 Activé`

Vecteur interruption 16 \Rightarrow la fonction d'interruption est dans la file T4T.D

18. Quelle est la caractéristique spécifique des fonctions `TRT_C` et `TRT_D` ? Pourquoi ne sont-elles pas appelées dans le code ? Que signifient les numéros 5 et 16 derrière les noms de ces fonctions ?

\Rightarrow Ce sont des fonctions d'interruption

Elles sont appelées automatiquement lorsque le périphérique lie à l'interruption fait une demande d'interruption

Les numéros (5 et 16) correspondent au numéro du vecteur d'interruption

ils permettent d'associer la fonction à 1 périphérique donné

19. Dans la ligne 171 « `if ((T4CON & (1<<6)) != 0) T4CON &= ~(1<<6);` » Expliquez avec des mots ce que fait ce code et à quoi il sert.

test non masquage du bit 6 de `T4CON` (`EXF4`)

Si ce bit est non nul, alors on le remet à zéro

20. Sur quelles broches du microcontrôleur sont connectées les LED verte, orange et rouge ainsi que le signal SIG_IN ?

LED_Verte \rightarrow P2.0

Orange : P2.1

Rouge : P2.2

SIG_IN \rightarrow Entrée de Comptage du Timer 4 \Rightarrow P0.6

21. BONUS - Expliquer le fonctionnement global de ce code en précisant le rôle des différents périphériques mis en œuvre.

Toute l'application fonctionne par Interruption

• Int_TIMER4 \Rightarrow et active à fin fin que P0 a compté 1000 événements

Cette interruption ne fait que positionner 1 drapeau par signal que P0 a compté 100 événements.

• INT_TIMER4 \Rightarrow Interruption Base de Temps : déclenchée toutes les 10ms.

\Rightarrow elle vérifie l'état du drapeau qui signale le comptage de 1000 événements.

et si ce dernier est à 1, gère les signaux d'allumage des LED.

par ailleurs cette interruption gère aussi le clignotement des LED (50ms).

22. BONUS - Aurait-il été possible de remplacer le timer configuré et utilisé par la fonction Config_Timer_TimeBase par un autre timer. Expliquer et justifier.

Gén. Malgré quel Timer on trouve Auto-rechargement capable de gérer des périodes de 10ms, avant ou après l'affaire \Rightarrow le Timer 3, et là aurait pu faire convenir. Mais Timer 4 est déjà utilisé \Rightarrow donc seul le Timer 3 avant ou convenir.

23. BONUS - Que devriez-vous changer pour avoir des cycles de comptage de 100 000 événements (0 à 49999, 50000 à 59999, , 90000 à 99999) ? Le comptage à 10^6 serait-il aussi possible ? Expliquer.

\Rightarrow Pour les cycles de 100 000 événements, on travaille par comptage de "paquets" de 10000 événements \Rightarrow le Timer 4 peut parfaitement compter 10^5 événements (au max il peut compter 2^{16} événements).

Par contre 10^6 , il aurait fallu à plus devoir gérer l'overflow du Timer 4.

Fin

ANNEXES pour Examen BSE TP 20/01/22

Pour faciliter votre travail,
cette feuille peut être
détachée.

Elle n'est pas à rendre à la
fin de l'examen.

```
1 //-----
2 // Examen_BSE_20-01-22.c
3 //-----
4 // AUTH: FJ
5 // DATE: 10/01/22
6 // Target: C8051F02x
7 // Tool chain: KEIL Microvision5
8 //
9 //-----
10 // Fichiers d'entête
11 #include<c8051F020.h>
12 //-----
13 // Prototypes de fonctions
14 //-----
15 void Oscillator_Init();
16 void Config_Timer_TimeBase(void);
17 void Config_B(void);
18 void Init_Device(void);
19 void Reset_Sources_Init();
20 void Port_IO_Init();
21 void Gestion_Etat_LED(unsigned char);
22 void Gestion_Cde_LED(void);
23 //-----
24 // Déclaration des MACROS
25 #define LED_ON 1
26 #define LED_OFF 0
27 #define ON 1
28 #define OFF 0
29 #define CLIGNO 2
30 //-----
31 //
32 sbit LED_Verte = P2^0;
33 sbit LED_Orange = P2^1;
34 sbit LED_Rouge = P2^2;
35 sbit VISU_TRT_C = P2^4;
36 sbit VISU_TRT_D = P2^5;
37
38 sfr16 RCAP2 = 0xca;
39 sfr16 T2 = 0xcc;
40 sfr16 RCAP4 = 0xe4;
41 sfr16 T4 = 0xf4;
42 //-----
43 // Variables globales
44 bit New_CP_1000;
45 unsigned char Etat_LED_Verte;
46 unsigned char Etat_LED_Orange;
47 unsigned char Etat_LED_Rouge;
48 //-----
49 // MAIN Routine
50 //-----
51 void main(void)
52 {
53     Init_Device();
54     Config_Timer_TimeBase();
55     Config_B();
56     Etat_LED_Verte = LED_OFF;
57     Etat_LED_Orange = LED_OFF;
58     Etat_LED_Rouge = LED_OFF;
59     EA = 1;
60     while(1);
61 }
62 //-----
63 //-----
64 void Reset_Sources_Init()
65 {
66     WDTCN = 0xDE;
67     WDTCN = 0xAD;
68 }
69 //*****
70 // Oscillator_Init()
71 //*****
72 void Oscillator_Init()
73 {
74     OSCICN = 0x07;
75 }
76 //-----
77 //-----
78 void Port_IO_Init()
79 {
80     XBR0 = 0x04;
81     XBR1 = 0xF4;
82     XBR2 = 0x58;
83
84     LED_Verte = LED_OFF;
85     LED_Orange = LED_OFF;
86     LED_Rouge = LED_OFF;
87     VISU_TRT_C = OFF;
88     VISU_TRT_D = OFF;
89 }
90
```

24/01/22


```

91
92 //-----
93 //-----
94 void Init_Device(void)
95 {
96     Reset_Sources_Init();
97     Port_IO_Init();
98     Oscillator_Init();
99 }
100 //*****
101 //*****
102 void Config_Timer_TimeBase(void)
103 {
104     CKCON &= ~(1<<5);
105     TF2 = 0;
106     EXF2 = 0;
107     RCLK0 = 0;
108     TCLK0 = 0;
109     CPRL2 = 0;
110     EXEN2 = 0;
111     CT2 = 0;
112     RCAP2 = 0xCBEB;
113     T2 = RCAP2;
114     TR2 = 1;
115     PT2 = 1;
116     ET2 = 1;
117 }
118 //*****
119 // void Config_B(void)
120 //*****
121 void Config_B(void)
122 {
123     T4CON = 0x02;
124     RCAP4 = 64536;
125     T4 = RCAP4;
126     T4CON |= (1<<2);
127     EIP2 |= (1<<2);
128     EIE2 |= (1<<2);
129 }
130 //*****
131 // void TRT_C(void)
132 //*****
133 void TRT_C (void) interrupt 5
134 {
135     static unsigned char CP_New_CP_1000;
136     static unsigned char CP_50ms;
137
138     VISU_TRT_C = 1;
139     if (TF2 == 1)
140     {
141         TF2 = 0;
142         if (New_CP_1000 == ON)
143         {
144             New_CP_1000 = OFF;
145             CP_New_CP_1000++;
146             if (CP_New_CP_1000 >= 10) CP_New_CP_1000 = 0;
147             Gestion_Etat_LED(CP_New_CP_1000);
148         }
149
150         CP_50ms++;
151         if (CP_50ms >= 5)
152         {
153             CP_50ms = 0;
154             Gestion_Cde_LED();
155         }
156     }
157     if (EXF2 == 1) EXF2 = 0;
158     VISU_TRT_C = 0;
159 }
160 //*****
161 // TRD_D
162 //*****
163 void TRT_D (void) interrupt 16
164 {
165     VISU_TRT_D = 1;
166     if ((T4CON & (1<<7)) != 0)
167     {
168         T4CON &= ~(1<<7);
169         New_CP_1000 = ON;
170     }
171     if ((T4CON & (1<<6)) != 0) T4CON &= ~(1<<6);
172     VISU_TRT_D = 0;
173 }
174
175
176
177
178
179
180

```

Pour faciliter votre travail,
cette feuille peut être
détachée.
Elle n'est pas à rendre à la
fin de l'examen.

```

181
182 //*****
183 // void Gestion_Etat_LED(unsigned char CP)
184 //*****
185 void Gestion_Etat_LED(unsigned char CP)
186 {
187     switch (CP)
188     {
189         case 0:
190         case 1:
191         case 2:
192         case 3:
193         case 4:
194             Etat_LED_Verte = CLIGNO;
195             Etat_LED_Orange = LED_OFF;
196             Etat_LED_Rouge = LED_OFF;
197             break;
198         case 5:
199             Etat_LED_Verte = LED_ON;
200             Etat_LED_Orange = LED_OFF;
201             Etat_LED_Rouge = LED_OFF;
202             break;
203         case 6:
204             Etat_LED_Verte = LED_OFF;
205             Etat_LED_Orange = CLIGNO;
206             Etat_LED_Rouge = LED_OFF;
207             break;
208         case 7:
209             Etat_LED_Verte = LED_OFF;
210             Etat_LED_Orange = LED_ON;
211             Etat_LED_Rouge = LED_OFF;
212             break;
213         case 8:
214             Etat_LED_Verte = LED_OFF;
215             Etat_LED_Orange = LED_OFF;
216             Etat_LED_Rouge = CLIGNO;
217             break;
218         case 9:
219             Etat_LED_Verte = LED_OFF;
220             Etat_LED_Orange = LED_OFF;
221             Etat_LED_Rouge = LED_ON;
222             break;
223         default:
224             Etat_LED_Verte = LED_OFF;
225             Etat_LED_Orange = LED_OFF;
226             Etat_LED_Rouge = LED_OFF;
227             break;
228     }
229 }
230 //*****
231 // void Gestion_Cde_LED()
232 //*****
233 void Gestion_Cde_LED()
234 {
235     if (Etat_LED_Verte == CLIGNO) LED_Verte = !LED_Verte;
236     else if (Etat_LED_Verte == LED_ON) LED_Verte = LED_ON;
237     else LED_Verte = LED_OFF;
238     if (Etat_LED_Orange == CLIGNO) LED_Orange = !LED_Orange;
239     else if (Etat_LED_Orange == LED_ON) LED_Orange = LED_ON;
240     else LED_Orange = LED_OFF;
241     if (Etat_LED_Rouge == CLIGNO) LED_Rouge = !LED_Rouge;
242     else if (Etat_LED_Rouge == LED_ON) LED_Rouge = LED_ON;
243     else LED_Rouge = LED_OFF;
244 }
245

```

Pour faciliter votre travail,
cette feuille peut être
détachée.
Elle n'est pas à rendre à la
fin de l'examen.