

Année universitaire 2021/2022

NOM :

PRENOM :

Consignes relatives au déroulement de l'épreuve

Date : 8 mars 2022

Devoir Module Bases des systèmes embarqués – Session 2 - **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 2 – BSE - Partie TP

Durée de l'épreuve : 1H00

Contexte : Analyse de code.

Dans cette examen de TP, Il va vous être demandé d'analyser un code qui a été produit pour répondre au cahier des charges suivant : « Générateur d'impulsions à durée programmée ».

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

Nous reproduisons ci-dessous, le cahier des charges de ce code:

Générateur d'impulsions à durée programmée

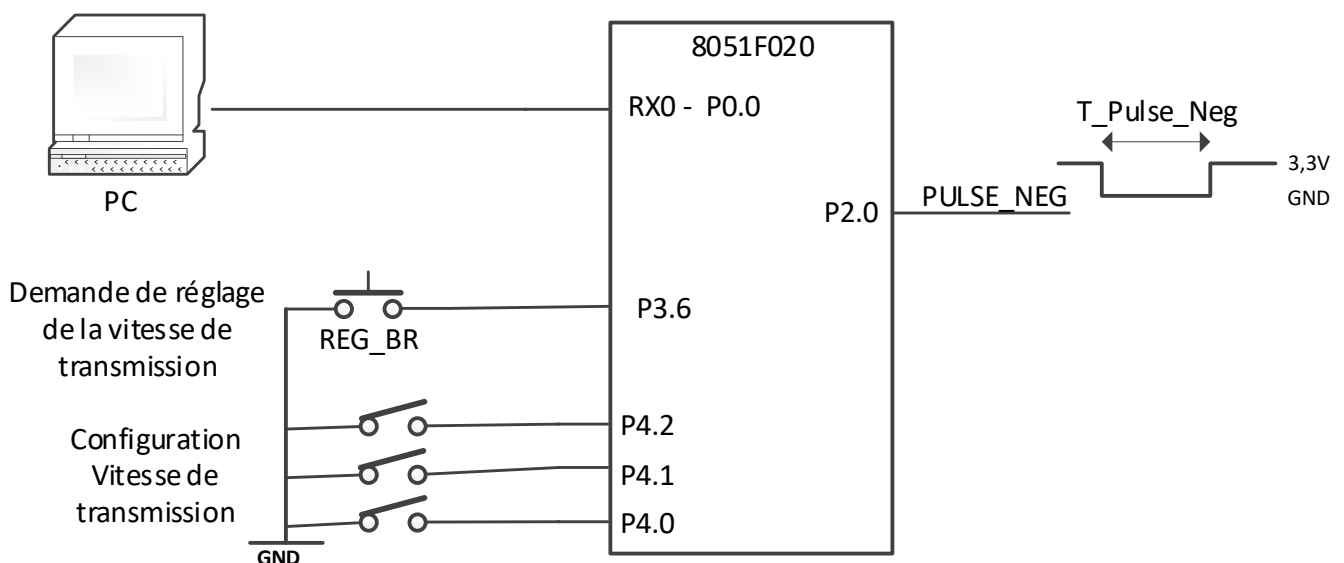
On souhaite, grâce au microcontrôleur, produire une impulsion négative PULSE_NEG (0/3,3V). La génération de cette impulsion sera déclenchée par un message produit par un terminal PC. Ce message contiendra de plus une information pour déterminer la durée de l'impulsion à produire.

Côté PC, on utilisera une liaison série asynchrone. La vitesse de transmission devra pouvoir être réglable au niveau du microcontrôleur grâce à 3 bits de configuration (interrupteurs câblés sur 3 broches du microcontrôleur) qui permettront un choix parmi 8 vitesses de transmission (115200,76800,57600,46080,38400,28800,19200,9600). Le réglage de la vitesse de transmission sera réalisé à chaque démarrage du microcontrôleur et à chaque action sur un bouton poussoir «Reg_BR ».

Le PC pour déclencher la génération de l'impulsion ne transmettra qu'un octet : CFG_PULSE. Dès la réception de cet octet, la génération de l'impulsion démarrera. Pendant toute la durée de la génération de l'impulsion, les éventuels messages envoyés par le PC ne seront pas pris en compte.

La durée de l'impulsion positive sera calculée à partir de la formule :

$$T_Pulse_Neg = (CFG_Pulse + 1) \text{ secondes}$$



6. (2 points) Dans la fonction Config_Timer2, on souhaite pouvoir configurer le timer2 pour obtenir des interruptions toutes les 25millisecondes. Sans changer une seule ligne de code, complétez la ligne RCAP2= ?; Justifiez votre calcul.

7. (1 point) Quel est le timer qui servira de source d'horloge pour le périphérique UART0 ? Justifiez

8. (2 point) Dans la fonction Config_Timer1, compte tenu du rôle attribué au Timer1, complétez en les justifiant les lignes TH1 = ?? et TL1 = ??;

9. (1 point) Expliquez le rôle de la fonction Config_ISR ?

.....

.....

.....

.....

.....

10. (2 points) Quels sont les évènements potentiellement susceptibles de provoquer l'exécution de la fonction ISR1 ? Mais, compte tenu du code, quel sera finalement le seul évènement qui provoquera l'exécution de la fonction ISR1 ?

.....

.....

.....

.....

.....

.....

11. (1 point) Dans la fonction ISR2, que fait la ligne de code « `RSTSRC = 0x10 ;` » ?

.....

.....

.....

.....

12. (1 point) Dans la fonction ISR2, la ligne de code « `VISU_ISR2 = OFF` » a-t-elle une utilité ? Pourquoi ?

.....

.....

.....

.....

.....

13. (2 points) Globalement, à quoi sert cette fonction ISR2 ? quel rôle remplit-elle par rapport à ce qui est demandé dans cet exercice ?

.....

.....

.....

.....

.....

.....

.....

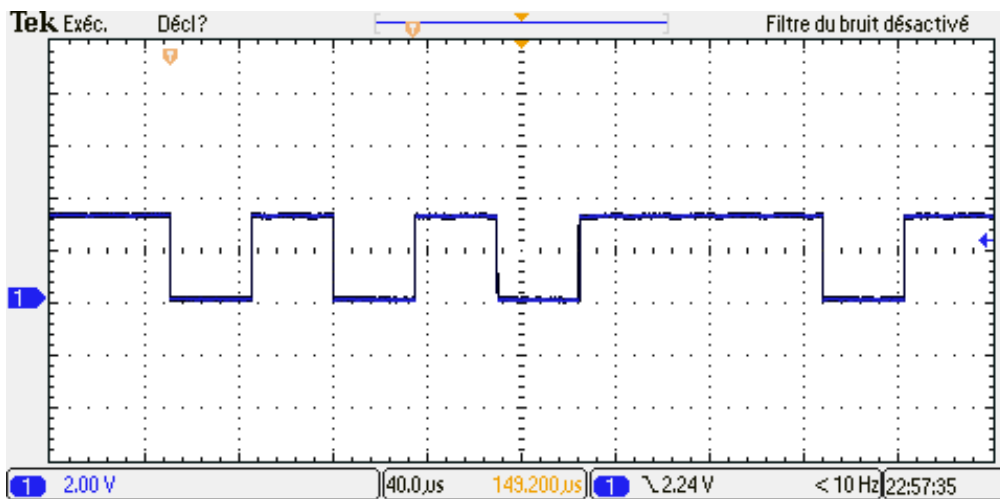
.....

.....

.....

14. (2 points) Quels sont les événements susceptibles de nous faire entrer dans la fonction ISR3 ?
Compte tenu de l'ensemble du code, risquons-nous de devoir traiter ces 2 événements ? Justifiez

15. (2 points) Lors de premier essais, après avoir branché la liaison série du PC sur le microcontrôleur, un premier octet est transmis par le PC. Cet octet transmis a été capturé par l'oscilloscope. Compte tenu de l'oscillogramme ci-dessous, déterminez la vitesse de transmission et la valeur de l'octet transmis. Remarque : la base de temps de l'oscilloscope est réglée sur 40µs/div



16. (1 point) Combien y a-t-il de fonctions d'interruptions dans ce code ? Compte tenu des diverses configurations, ces interruptions peuvent-elles s'interrompre l'une l'autre ? Justifiez ?

```

1 //-----
2 // Examen BSE Seconde-Session 08-03-22.c
3 //-----
4 // AUTH: FJ
5 // DATE: 21/02/22
6 // Target: C8051F02x
7 // Tool chain: KEIL Microvision5
8 //-----
9 // Fichiers d'entête
10 #include<c8051F020.h>
11 #include<c8051F020_SFR16.h>
12 #include<BSE_Main.h>
13 //-----
14 // Déclaration des MACROS
15 #define ON 1
16 #define OFF 0
17 #define SYSCLK 22118400L
18 //-----
19 // Déclarations Registres et Bits de l'espace SFR
20 sbit PULSE_NEG = P2^0;
21 sbit VISU_ISR1 = P2^3;
22 sbit VISU_ISR2 = P2^4;
23 sbit VISU_ISR3 = P2^5;
24 sfr CFG_BaudRate = 0x84;
25 //-----
26 // Variables globales
27 unsigned char CFG_PULSE = 1;
28 bit Pulse_Request = OFF;
29 const long int code tab_br[] = {9600,19200,28800,38400,46080,57600,76800,115200};
30 //-----
31 // MAIN Routine
32 //-----
33 void main (void) {
34
35     unsigned char i;
36
37     // Configurations globales
38     Init_Device();
39
40     // Configurations spécifiques
41     i = CFG_BaudRate & 0x07;
42     Config_Timer1(tab_br[i]);
43     Config_Timer2();
44     CFG_UART0();
45     Config_ISR();
46     // Fin des configurations
47
48     EA = 1; // Validation globale des interruptions
49
50     // Boucle infinie
51     while(1);
52 }
53 //-----
54 //-----
55 void Reset_Sources_Init()
56 {
57     WDTCN = 0xDE;
58     WDTCN = 0xAD;
59 }
60 //-----
61 //-----
62 void Port_IO_Init()
63 {
64     XBR0 = 0x04;
65     XBR1 = 0x94;
66     XBR2 = 0x44;
67
68     P0MDOUT |= (1<<0) | (1<<2);
69     P0MDOUT |= (1<<6);
70     P2MDOUT = 0xFF;
71     P74OUT = 0x00;
72     P3MDOUT &= ~(1<<6);
73     P3 |= (1<<6);
74     P4 = 0xFF;
75
76     PULSE_NEG = 1;
77     VISU_ISR1 = OFF;
78     VISU_ISR2 = OFF;
79     VISU_ISR3 = OFF;
80 }
81 //-----
82 //-----
83 void Oscillator_Init()
84 {
85     int i = 0; //Configuration de SYSCLK avec source externe à 22,1184Mhz
86     OSCXCN = 0x67;
87     for (i = 0; i < 3000; i++); // Wait 1ms for initialization
88     while ((OSCXCN & 0x80) == 0);
89     OSCICN = 0x08;
90 }

```

ANNEXES pour Examen BSE TP 03/22

Pour faciliter votre travail, ces feuilles de code peuvent être détachées. Elles ne sont pas à rendre à la fin de l'examen

```

91 //-----
92 // Initialisation globale du Microcontrôleur -
93 //-----
94 void Init_Device(void)
95 {
96     Reset_Sources_Init();
97     Port_IO_Init();
98     Oscillator_Init();
99 }
100 //*****
101 //*****
102 void Config_Timer2(void)
103 {
104     CKCON &= ~(1<<5);
105     TF2 = 0; // RAZ TF2
106     EXF2 = 0; // RAZ EXF2
107     RCLK0 = 0;
108     TCLK0 = 0;
109     CPRL2 = 0;
110     EXEN2 = 0;
111     CT2 = 0;
112     RCAP2 = ??;
113     T2 = RCAP2;
114     TR2 = 1;
115     PT2 = 1;
116     ET2 = 1;
117 }
118 //*****
119 //*****
120 void Config_Timer1(long int Baud_rate)
121 {
122     TR1 = 0;
123     TMOD |= (1<<5);
124     TMOD &= ~(1<<7) | (1<<6) | (1<<4);
125     CKCON |= (1<<4);
126     TH1 = ??;
127     TL1 = ??;
128     TR1 = 1;
129 }
130 //*****
131 //*****
132 //CFG_uart0
133 //
134 //*****
135 void CFG_UART0(void)
136 {
137     RCLK0 = 0;
138     TCLK0 = 0;
139     PCON |= (1<<7);
140     PCON &= ~(1<<6); // SSTAT0=0
141     SCON0 = 0x70;
142
143     RI0 = 0;
144     TI0 = 0;
145
146     ES0 = 1; // IE.4
147     PS = 0; // IP.4
148 }
149 //*****
150 //*****
151 void Config_ISR(void)
152 {
153     P3IF &= ~(1<<7);
154     P3IF &= ~(1<<2);
155
156     EIP2 &= ~(1<<4);
157     EIE2 |= (1<<4);
158 }
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180

```



```

181
182 //*****
183 // ISR
184 //*****
185 void ISR1 (void) interrupt 5
186 {
187     static bit Pulse_generation = OFF;
188     static unsigned int CP_Pulse;
189     static unsigned int Pulse_to_count;
190
191     VISU_ISR1 = 1;
192
193     if (TF2 == 1)
194     {
195         TF2 = 0;
196         if (Pulse_generation == ON)
197         {
198             Pulse_Request = OFF;
199             CP_Pulse++;
200             if (CP_Pulse == Pulse_to_count)
201             {
202                 PULSE_NEG = 1;
203                 Pulse_generation = OFF;
204             }
205         }
206         else if (Pulse_Request == ON)
207         {
208             Pulse_generation = ON;
209             CP_Pulse = 0;
210             Pulse_to_count = (CFG_PULSE + 1)*40;
211             PULSE_NEG = 0;
212         }
213     }
214
215     if (EXF2 == 1)
216     {
217         EXF2 = 0;
218     }
219     VISU_ISR1 = 0;
220 }
221
222 //*****
223 void ISR2 (void) interrupt 18
224 {
225     VISU_ISR2 = ON;
226     P3IF &= ~(1<<6);
227     RSTSRC = 0x10;
228     VISU_ISR2 = OFF;
229 }
230 //*****
231 void ISR3 (void) interrupt 4
232 {
233     VISU_ISR3 = 1;
234
235     if (RIO == 1)
236     {
237         RIO = 0;
238         CFG_PULSE = SBUF0;
239         Pulse_Request = ON;
240     }
241     if (TIO == 1)
242     {
243         TIO = 0;
244     }
245
246     VISU_ISR3 = 0;
247 }
248 //*****

```