

## Sujet du TP 4

### Conditions de réalisation :

- Utilisation de microvision5 pour 8051 et de la carte de développement 8051F020DK.
- Utilisation d'une application « Terminal de commande » telle que « Putty » pour établir une liaison entre un microcontrôleur et un PC au travers d'une liaison série type RS232
- Documentation indispensable : Poly Fiche Technique du 8051F020 (extraits).
- Documentations utiles : Cours BSE
- Le point de départ est un projet Microvision BSE\_TP4 (téléchargeable sur le E-campus). Ce projet contient plusieurs fichiers source (4 fichiers dont `startup.A51`). Ce projet est 100% fonctionnel.

### Etape de préparation et de compréhension.

1. Installer le kit matériel 8051F020 constitué de la carte 8051, de son boîtier de débogage et de son bloc alimentation
2. Télécharger le projet BSE\_TP4, le désarchiver, le compiler et faire exécuter le code sur la carte. Vérifier que ce code exécute bien les tâches suivantes (cette application correspond à un corrigé du TP3):
  - Clignotement régulier de la LED branchée sur P1.6.
  - Application de comptage d'événements – Un signal SIG\_IN appliqué sur la broche P0.6 (T4) provoque un changement d'état sur le signal SIG\_OUT (broche P3.3) à chaque fois que l'on a compté XX événements (fronts montants) sur ce signal SIG\_IN. Déterminez la valeur de comptage XX.
  - Application fréquencemètre – à chaque seconde, la variable globale FREQUENCE est rafraîchie et contient une valeur correspondant à la fréquence du signal SIG\_IN.
  - Compte tenu des fonctions de configuration données quelle est la fréquence de l'horloge système SYSCLK ?



**Les signaux électriques échangés avec la carte 8051F020 sont des niveaux CMOS 0-3,3V !! Utilisation du +5V interdite !!**

### Etape de compréhension 1 – Configuration de l'UART0 – Test de fonctionnement de L'UART en transmission

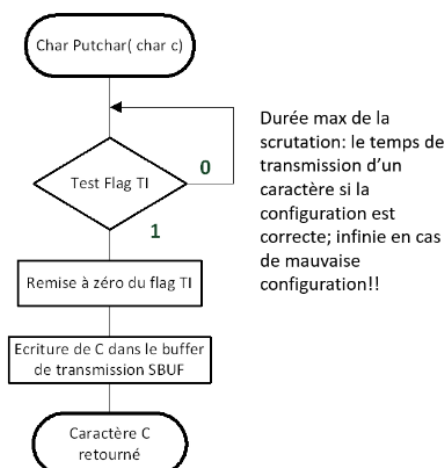
1. L'objectif de cette première étape est de coder les configurations de base de l'UART et de vérifier que cette configuration est fonctionnelle. Les paramètres de la liaison série à mettre en œuvre seront les suivants : 19200Bd, 8 bits, pas de parité, 1 stop bit. Les codes que vous produirez, ne doivent pas remettre en question les différentes tâches déjà exécutées.
2. Coder la fonction `Cfg_CLK_UART0` – Cette fonction est chargée de configurer l'horloge qui va piloter la cadence de transmission et de réception des informations sur l'UART0.
3. Coder la fonction `Cfg_UART0` – Cette fonction est chargée de configurer l'UART0 pour que ce périphérique puisse fonctionner selon les spécifications données dans le point1
4. Vérifier la configuration réalisée avec un test simplissime : dans le `while(1)` de la fonction `main`, faites exécuter en continu la séquence suivante : envoi d'un caractère sur la liaison série (toujours le même), suivi d'une temporisation de 10ms (une temporisation logicielle sera suffisante). Vérifier la bonne transmission du caractère sur le terminal de commande Putty et observer sur l'oscilloscope l'allure du caractère. Mesurer la durée de transmission du caractère (uniquement les 8 bits du caractère transmis)
5. Que se passe-t-il si on enlève la temporisation de 10ms ? Conclure.

### Etape de compréhension 2 – Test de fonctionnement de L'UART en réception de caractère – application d'écho de caractère

1. La transmission étant opérationnelle, il faut désormais vérifier que la réception de caractère sur l'UART fonctionne.
2. Cette vérification sera faite au travers d'un code placé dans le `while(1)` du `main`.
3. La séquence du code sera la suivante : A chaque fois qu'un caractère sera reçu par l'UART, cette dernière le retransmettra en incrémentant la valeur reçue de 1. Ainsi si le terminal de commande envoie le caractère 'T' (0x54), alors le caractère qui lui sera retourné sera le caractère 'U' (0x55).
4. Vérifier sur un oscilloscope la transmission et la réception de caractère. Mesurer la durée qui sépare la fin de la réception du caractère reçu et le début de la transmission du caractère émis par l'UART.

### Etape de compréhension 3 – Codage de la fonction Putchar.

1. Coder la fonction `char putchar(char)`. L'algorithme de cette fonction apparaît ci-contre :
2. Pour tester cette fonction, transmettez toutes les secondes la valeur de la fréquence mesurée par l'application (variable fréquence), à l'aide de la fonction `printf` (fonction de librairie) qui s'appuie elle-même sur votre fonction `Putchar`. Pensez à insérer des sauts de lignes pour mieux visualiser la valeur de la fréquence. Ce code sera placé dans le `while(1)` du `main`.
3. Exemple de chaîne transmise : **Mesure fréquence: 48200Hz**
4. Mesurer la durée de transmission de cette chaîne de caractères.

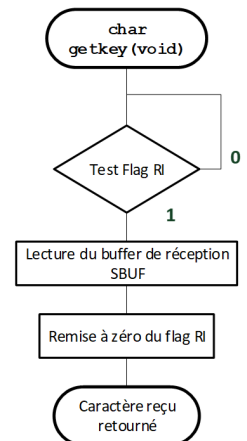
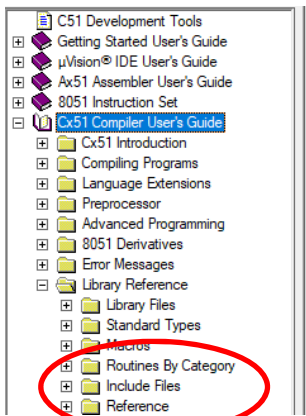


#### Etape de compréhension 4 – Codage de la fonction `_getkey`.

1. Coder la fonction `char _getkey ()`. L'algorithme de cette fonction, est donné la figure ci-contre :
2. Pour tester cette fonction, utilisez la fonction `gets` (fonction de librairie) (qui s'appuie sur votre fonction `_getkey`) pour lire une chaîne de caractères émise par le terminal de commande.
3. Modifiez la chaîne de caractères reçue en incrémentant de 1 toutes les valeurs de la chaîne et renvoyez-la vers le terminal. Par exemple, si vous avez envoyé « Essai » (0x45 0x73 0x73 0x61 0x69), le programme vous renvoie « Fttbj » (0x46 0x74 0x74 0x62 0x6A).

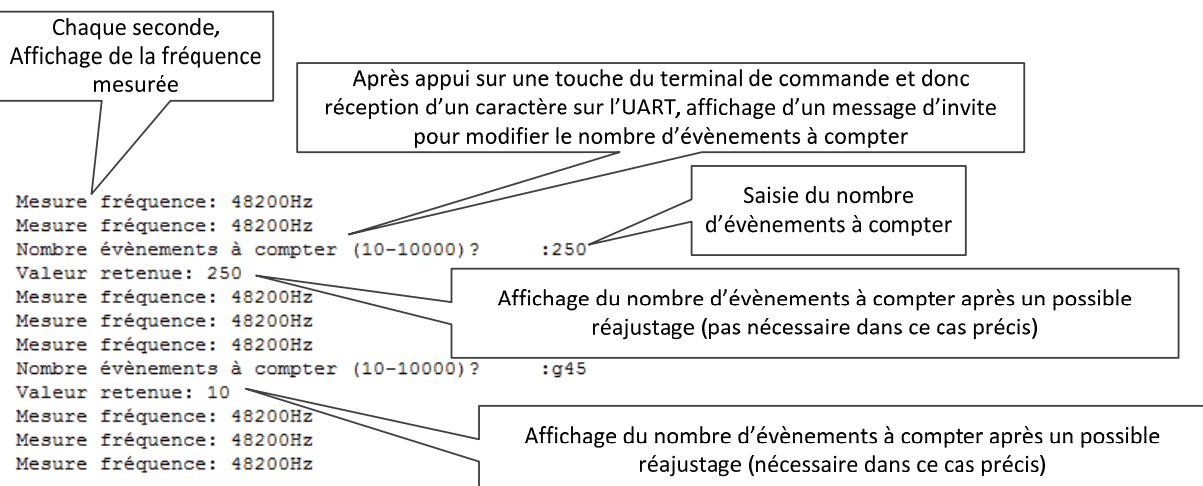


Pour les fonctions telles que `printf`, `gets`, qui sont des fonctions de la librairie standard (C51 Standard Library), il faut vous reporter sur la documentation de CE compilateur C51 (il y a quelques particularités)



#### Etape de synthèse 5 – Comptage d'événements pilotable.

1. L'objectif est de rendre configurable le comptage d'événements, depuis le terminal de commande. Dans l'application qui vous a été livrée pour ce TP, le comptage est réglable par décade (10 – 100-1000-1000) au travers de la lecture de l'état de 2 broches P2.5 et P2.6. On va supprimer cette possibilité et récupérer désormais la consigne de comptage d'événements à partir du terminal de commande.
2. La valeur transmise par le terminal de commande devra être comprise entre 10 et 10000. Si ce n'est pas le cas, on utilisera une valeur par défaut de 10.
3. Toutes les secondes on enverra vers le terminal de commande la fréquence mesurée (variable fréquence).
4. Par ailleurs, dès que l'on détectera la réception d'un caractère quelconque sur l'UART, on affichera un message d'invite pour modifier le nombre d'événements à compter. Ce nombre sera évalué (est-il dans la fourchette fixée ?) et sera réajusté à une valeur par défaut si nécessaire. On s'assurera ensuite que le comptage d'événements a bien été modifié dans l'application.
5. Exemple d'affichage :



**Validations - Sujet – TP4**
**Groupe :**
**Date :**
**Etudiant1 :**
**Etudiant2 :**

Etape Préparation	Validé par :	Heure :	Etape 1	Validé par :	Heure :
Validation 0 à 100%			Validation 0 à 100%		
<b>Contrôle du fonctionnement de l'application transmise</b> Nombre d'évènements SIG_IN comptés à chaque transition de SIG_OUT : _____ Valeur de Fréquence SYSCLK : _____  <b>Commentaires éventuels :</b>			<b>Test UART0 en transmission</b> Fonction Config_Clock_UART0 codée: Fonction Config_UART0 codée: Fonctionnement émission caractère en continu – OK ? Durée de transmission d'un caractère ? : _____  <b>Commentaires éventuels :</b>		

Etape 2	Validé par :	Heure :	Etape 3	Validé par :	Heure :
Validation 0 à 100%			Validation 0 à 100%		
<b>Test UART0 en réception :</b> Fonctionnement écho de caractère avec incrément – OK ? Mesure de la durée entre fin de réception de caractère et début de transmission du caractère en réponse : _____  <b>Commentaires éventuels :</b>			<b>Codage de putchar</b> Visu de la transmission de la chaîne de caractères toutes les secondes - OK ? Mesure de la durée de transmission de la chaîne de caractères : _____  <b>Commentaires éventuels :</b>		

Etape 4	Validé par :	Heure :	Etape 5	Validé par :	Heure :
Validation 0 à 100%			Validation 0 à 100%		
<b>Codage de _getkey</b> Visu de l'écho de chaîne de caractères « modifiée » - OK ?  <b>Commentaires éventuels :</b>			<b>Synthèse</b> Visu affichage en continu de la fréquence – OK ? Visu affichage de l'invite « modification comptage évènements » à chaque appui sur une touche – OK ? Visu contrôle de la valeur transmise – OK ? Contrôle de la prise en compte de la nouvelle valeur de comptage d'évènements – OK ?  <b>Commentaires éventuels :</b>		

Synthèse	Validé par :	Heure :		Validé par :	Heure :
Validation 0 à 100%			Validation 0 à 100%		
<b>Commentaires éventuels :</b>			<b>Commentaires éventuels :</b>		