

Liaison RS232, protocole XON/XOFF

Dans ce TP on développe une partie des pilotes permettant l'utilisation du port série du PC (RS232). On utilise ces pilotes pour mettre en oeuvre le protocole XON/XOFF de contrôle de flux lors d'une liaison série avec une console (HP ou ADM12+).

Fichier fourni : `UART_ba.cpp`

Matériel : Oscilloscope+sonde, console série (ADM12+ ou HP), TC/DOS

1. Mise en oeuvre de la console.

Le programme `UART_ba.cpp`, envoie l'alphabet 'a' à 'z' sur la ligne série de manière répétitive avec les paramètres de transmission suivants :

- 9600Bauds
- 8bits
- Parité impaire
- 1 Stop Bit

Connecter la console et le PC avec le câble série (3fils). DB9 coté PC, DB25 coté console (PORT1 pour HP, MODEM pour ADM12+). Configurer la console avec les paramètres précédents :

Console ADM12+

`Ctrl+Shift+SetUp` puis curseurs `↓↑` jusqu'aux champs à modifier, curseurs `←→` pour modifier les champs

Console HP

`SetUp` puis `NextScreen` jusqu'à l'onglet `PORT1`, modifier les champs nécessaires avec les flèches de curseur `↓↑←→`

Vérifier que l'alphabet s'affiche correctement sur la console. On veut vérifier ce qu'il se transfère sur la ligne série. Modifier le programme pour envoyer successivement les lettres 'a' puis 'c' puis 'a'etc. Par exemple :

```
send_car_uart('a');
delay(2);
send_car_uart('c');
delay(10);
```

Sur la ligne série, et en débranchant la console, visualiser à l'aide de l'oscilloscope le signal transmis.

Quelle est la durée d'un bit ? combien y a-t'il de bits émis pour 1 unique caractère? Quelle est la durée d'émission d'un caractère complet ? Quelles sont les niveaux correspondants à '0' et '1' ? A partir de l'oscillogramme, retrouver le caractère transmis et les bits de contrôle.

2. Modification des paramètres de la console.

A l'aide de la datasheet de l'UART 82C50, analyser et commenter les fonctions `void init_uart(void)` et `void send_car_uart(char)`. En particulier, donner les adresses et décrire l'utilisation des registres suivants : RBR, THR, LCR, LSR, DLL et DLM.

Détailler l'utilité de :

```
do{
    bit=inportb(ADR_LSR);
    bit=bit&0x20;
}while(bit==0);
```

Par exemple, commenter cette partie et vérifier que ça ne fonctionne plus.

Modifier les paramètres du programme d'émission et de la console pour avoir la vitesse maximale de transmission (19200 sur ADM12+ et 38400 pour HP).

Vérifier qu'au bout d'un « certain temps » l'alphabet ne s'affiche plus correctement. Pourquoi ?

3. Mise en place de la fonction de réception `char get_car_uart(void)`

En vous inspirant de la fonction d'émission, développer la fonction `char get_car_uart(void)` qui renvoie un éventuel caractère en provenance de la console.

Tester en affichant à l'écran du PC les caractères en provenance du clavier de la console.

4. Mise en place du contrôle de flux.

Lorsqu'on envoie un grand nombre de caractères à la console en rafale, celle ci ne peut les traiter assez rapidement, de sorte que « au bout d'un certain temps » certains caractères se perdent. Le protocole de contrôle de flux XON/XOFF permet de résoudre ce problème sur une liaison 3 fils :

- Quand le buffer de la console (configurée pour gérer XON/XOFF) dépasse un certain seuil, la console émet en retour vers le PC le caractère `XOFF=0x13=19` pour demander un arrêt de transmission.
- Lorsque la console est de nouveau disposée à recevoir des caractères, elle émet le caractère `XON=0x11=17` pour autoriser une reprise de transmission.

Mettre en place et tester le contrôle de flux coté PC.

Evaluer le plus précisément possible le nombre de caractères qu'on peut envoyer en rafale (sans contrôle de flux) sur votre consoles et pour les 2 vitesses de transmission les plus rapides

Conclure