

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, 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

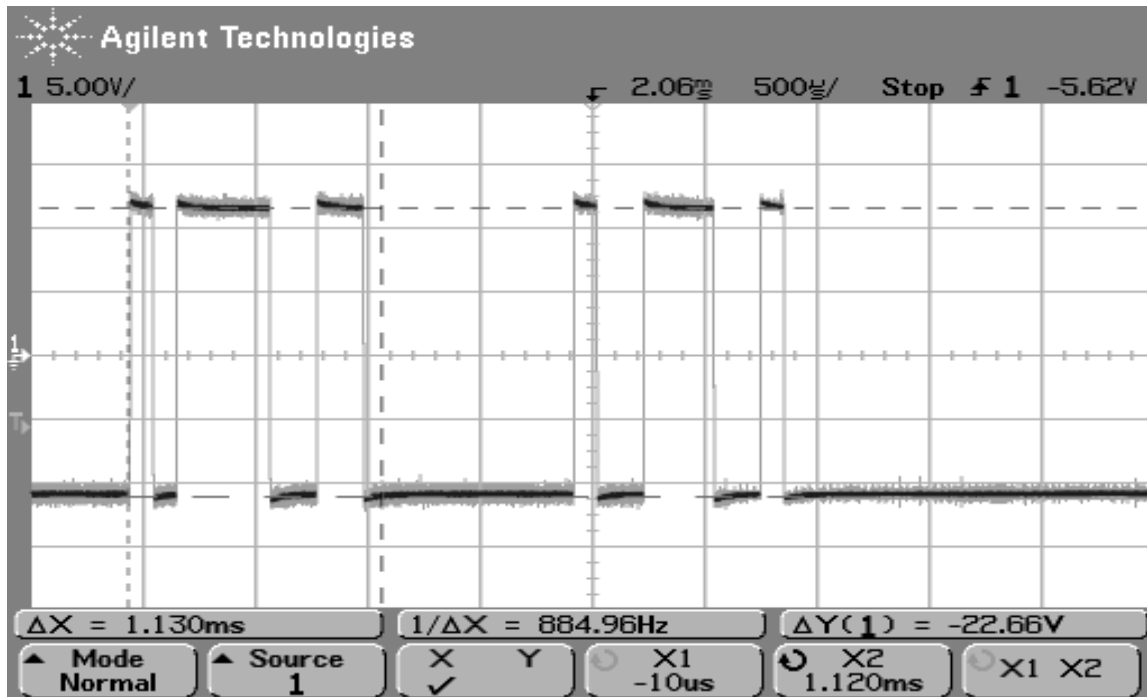
SetUp puis NextScreen jusqu'à l'onglet PORT1, modifier les champs nécessaires avec les flèches de curseur ↓↑←→

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.



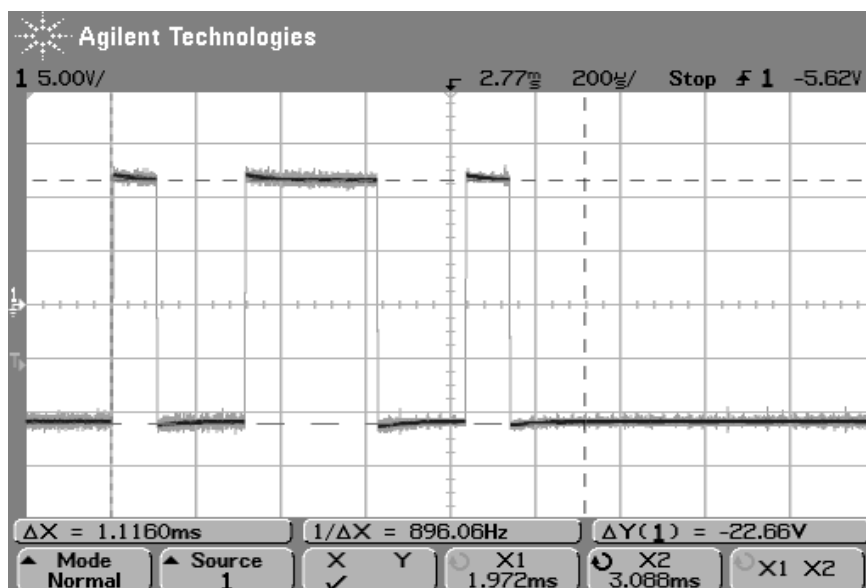
Ici on a 2 caractères : 'a' puis 'c', chaque caractère fait 11 bits :

Start +8bits+parité_impaire+stop soit 1.13ms, chaque bit fait $1/9600=104\mu\text{s}$ ce qu'on peut vérifier sur l'oscillogramme

Bit1 = -12V, bit0 = 12V, la norme RS 232 spécifie les niveaux +-V entre 3 et 25V (sur un PC c'est 12V, implémentation particulière de RS232) .

Entre 'a' et 'c' un bit (unique) change et donc le bit de parité aussi (cf niveau 12V sur le C !!), on ne distingue pas le bit de stop = -12V=bit1 mais il est la, on le verrait en supprimant le delay(2) du code

Remarquer que c'est le LSB qui part en premier (a gauche de l'oscillogramme donc....)



Caractère 'c' minuscule (code 97 = 0x61)

Remarquer qu'il y a aussi lms entre la fin de l'emission du 'a' et le début du 'c' mais qu'on a mis delay(2)...le delai est entre les 2 outportb() de send_car_uart() soit entre le debut du 'a' et le debut du 'c'....

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). Où sont configurées les caractéristiques de transmission (parité, 8bits etc...).

facile lire le corrigé fourni

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.

Evidement ca ne marche plus (si on enleve les delay) puisqu'on attend pas que le registre tampon de l'uart (THR) soit vide (bit THRE du registre LSR)...les caracteres s'ecrasent les uns sur les autres, sur une uart 16550 (celle qui est dans le PC mais qu'on utilise en mode compatibilité 82C50) il existe une fifo de 16 caracteres permettant d'éviter en partie ce probleme, on peut alors envoyer des blocs de 16 caracteres d'un coup sans vérifier l'état du bit THRE

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 ?

Saturation du buffer de la console cette fois ci..... voir XON/XOFF: La console ne peut afficher les caracteres aussi vite qu'ils arrivent, au bout d'un moment le buffer (j'ignore la taille mais autour de 1 ou 2 Ko ?) est saturé, les caracteres en provenance de la ligne série écrasent certains caractere bufferisés et non encore affiché. Les lettres de l'alphabet ne sont plus dans l'ordre et la console BIP (ADM12+ seulement)

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.

Attention on n'attend pas un caractere, on regarde si il y en a unc'est pour ca qu'il y a 2 prototype get_car_uart et wait_car_uart..... dans les 2 cas on lit de bit dataready de LSR mais dans un cas on attend un caractere...

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

4. Mise en place du controle 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 console et pour les 2 vitesses de transmission les plus rapides

Lire le code du corrigé....

L'alphabet s'affiche maintenant parfaitement. J'ai testé les résultats suivants :

ADM12+ :

-a 19200bauds : Le premier XOFF est arrivé apres 800 caracteres emis environ (0.4s, ensuite tous les 400 caracteres

-a 9600Bauds : Le premier XOFF est arrivé apres 7000 caracteres emis environ (7s), ensuite tous les 2500 caracteres

-en dessous de 9600 : l'adm12+ n'est jamais saturée, elle est donc capable d'afficher + de 480 caractères/seconde !!!

HP

On peut configurer 2 niveaux de buffer XON/XOFF 64 et 128 caractères.

Je pense que c'est les niveaux d'alerte de Buffer sur la console, si buffer>64 ou 128 caracteres alors la console envoie XOFF .

a 38400Bauds : XOFF apres 90 (pour une config 64), et 180 caracteres emis (pour 128) (ca confirme l'hypothèse précédente : XOFF des que 64 ou 128 caractères se trouvent en attente) .

J'en deduis qu'en $90 \times 0.2\text{ms} = 18\text{ms}$ la console a traité 26 caracteres et le buffer a atteint 64 caracteres, provoquant l'emission de XOFF. La console doit donc traiter environ $26\text{car}/18\text{ms} = 1500\text{car/seconde}$.

C'est approximatif mais ca correspondrait avec le fait qu'a 19200bauds (=1800car/sec) la console ne sature jamais)

Si erreur de parité le caractere ne s'affiche pas et sur la ligne en bas de la console on voit un '>' clignoter