

Atmos: 19200 bauds!

Le problème principal de l'Atmos/Oric-1 réside dans son manque de moyens de communication. A tel point que son successeur, le Téléstrat, a été présenté comme une "Borne de Communications"!

La simulation d'un port RS232 soft sur la prise d'imprimante est une bien précieuse réalisation. Cette simulation a été utilisée par exemple par LORITEL (Loriciciels® 1985), par ATMITEL (Th. Legal, Théoric n°27 page 28) et plus récemment par notre ami François L. (CEO-MAG n°43&44). Cependant le temps perdu à la conversion "soft" des octets en signaux série limite la vitesse de transmission à 1200 bauds.

J'ai tenté, hélas en vain, d'utiliser cette prise soi-disant Centronics et soi-disant bi-directionnelle pour mettre au point une communication parallèle à grande vitesse. Mais le clavier de l'Atmos se bloque dès que l'on branche une autre machine (même un Atmos) sur le port imprimante au moyen d'un câble parallèle direct ou croisé (type Laplink).

Reste la carte RS232 proposée par Frédéric Taraud dans Théoric n°14 page 43. Ici je dois remercier publiquement François B. qui a bien gentiment accepté de m'en fabriquer une. Cette carte utilise un UART 6551 qui effectue la conversion parallèle/série des octets présents sur le bus. La transmission se fait sur 3 fils: émission, réception et masse.

Le 6551 permet en théorie de transmettre à 19200 bauds, mais ce n'est pas toujours possible, car d'une part le câble de connexion peut ramasser des interférences et d'autre part l'Atmos est une machine relativement lente. Le premier problème est contournable avec un câble court (1 mètre) et blindé (la masse est une tresse qui entoure les deux autres conducteurs). Le deuxième problème est résolu par un petit programme en langage machine optimisé pour courir très vite. C'est ce programme qui fait l'objet de mon propos d'aujourd'hui.

Résultat: je suis parvenu sans difficulté à communiquer à la maximale de 19200 bauds (environ 1920 octets par seconde), quelle que soit la machine connectée à l'autre bout (ayant un port ou une carte RS232).

Le programme listé ci-dessous **IO.COM** est implanté de **#0900 à #09E2**. Il comporte les routines "Emission Bloc", "Réception Bloc", "Initialisation Carte" et "XTALK". Cette dernière routine permet de dialoguer avec une autre machine, les communications dans les deux sens étant simultanées. Les octets émis sont scrollés sur la ligne service, tandis que les octets reçus sont affichés à l'écran.

Pour envoyer un bloc d'octets, il faut charger ce bloc en mémoire à partir de \$0A00 (par exemple avec un LOAD"nom",A#\$0A00), puis DOKER l'adresse de fin du bloc en \$09FE (en BASIC: LOAD"nom",V:DOKE#9FE,(\$A00+ED-ST), c'est à dire une valeur égale à #0A00 + nombre d'octets à envoyer) et enfin faire CALL#0900.

Pour recevoir un bloc d'octets, il faut DOKER en \$09FE l'adresse de fin du bloc de stockage (en BASIC: INPUT NO: DOKE#9FE,#9FF+NO, c'est à dire une valeur égale à #0A00 + le nombre d'octets (NO) à recevoir soit #AAFF=43775 au maximum), faire CALL#0940 et enfin après retour au Ready, SAVE"nom",A#\$0A00,E(DEEK(#00)-1).

Prévoyez un HIMEM#08FF si vous utilisez un petit programme BASIC pour faire tout cela de manière plus conviviale.

Les "Labels" Lxxxx indiquent les points de branchement (adresses).

L0900 = Emission Bloc

LDA#\$00 adresse (adr) de
STA\$00 début du bloc à envoyer
LDA#\$0A soit \$0A00
STA\$01

LDA\$09FE adr fin bloc à envoyer
STA\$02 copié de \$09FE/09FF
LDA\$09FF
STA\$03
JSR\$09A0 initialise carte RS232

```

SEI          interdit interruptions
LDY#$00      index
L0918
LDA($00),Y      lit  octet  du
bloc
STA$BB80      affiche / ligne service
TAX           sauve octet dans X
L091E
LDA$0381      le registre d'émission
AND#$10       est-il vide?
BEQ$091E      si non, stand-by
STX$0380      si oui, écrit l'octet
INC$00        dans registre émission
BNE$092E      met à jour l'adresse
INC$01        de l'octet suivant
L092E
LDA$01        la fin du bloc
CMP$03        est-telle atteinte?
BCC$0918
BNE$093E      si oui, terminé
LDA$00
CMP$02
BCC$0918      si non, reboucle
BEQ$0918
L093E
CLI          autorise interruptions
RTS          et termine
L0940 = Réception Bloc
LDA#$00
STA$00        adr déb bloc à recevoir
LDA#$0A      soit $0A00
STA$01
LDA$09FE      adr fin bloc à recevoir
STA$02        copié de $09FE/09FF
LDA$09FF
STA$03
JSR$09A0      initialise carte RS232
SEI          interdit interruptions
LDY#$00
L0958
LDA$0381      le registre de réception
AND#$08       est-il plein?
BEQ$0958      si non, stand-by
LDA$0380      si oui, lit l'octet reçu
STA$BB80      l'écrit / ligne service
STA($00),Y    et dans le bloc
stockage
INC$00
BNE$096D      incrémente adr stockage
INC$01
L096D
LDA$01        tous les octets attendus
CMP$03        ont-ils été reçus?
BCC$0958
BNE$097D      si oui, terminé
LDA$00
CMP$02
BCC$0958      si non, reboucle
BEQ$0958
L097D

```

```

CLI          autorise interruptions
RTS          et termine
L097F = Préparation Ligne Service
LDX#$00
STX$BBA6      efface les deux
STX$BBA7      dernières cases à droite
LDA#$2E       code ASCII de "."
L0989
STA$BB82,X    copié dans les
36 cases
INX           centrales de la ligne
CPX#$24       (épargne les 2 cases à
BCC$0989      gauche & les 2 à droite)
RTS
L0992 = Scrolling Ligne Service
LDX#$00      lit un octet de la zone
L0994
LDA$BB83,X    des 36 cases et
le copie
STA$BB82,X    immédiatement à
gauche
INX           effectue 36 copies
CPX#$24       (épargne les 2 cases à
BCC$0994      gauche & les 2 à droite)
RTS
L09A0 = Initialisation Carte RS232
LDA#$00      soit 0000 0000
STA$0381      Reset
LDA#$0B       soit 0000 1011
STA$0382      sans parité, mode normal
LDA#$1F       soit 0001 1111
STA$0383      1 stop, 8 bits,
RTS           19200 bauds (#1E=9600,
#1C=4800, #1A=2400 & #18=1200 bds)
L09B0 = XTALK
SEI          interdit interruptions
JSR$097F      reset ligne service
JSR$09A0      initialise carte RS232
L09B7
LDA$0381      le registre de réception
AND#$08       est-il plein? (priorité)
BEQ$09C7      si non, teste émission
LDA$0380      si oui, lit octet reçu
JSR$CCD9      et l'affiche à l'écran
JMP$09B7      rebouclage forcé
L09C7
CLI          autorise interruption
JSR$EB78      teste touche clavier
SEI          interdit interruptions
BPL$09B7      reboucle si pas touche
PHA           empile touche pressée
L09CF
LDA$0381      le registre d'émission
AND#$10       est-il vide?
BEQ$09CF      si non, stand-by
JSR$0992      si oui, scroll l service
PLA           récupère touche pressée
STA$0380      écrit ds regist émission
STA$BBA5      et sur ligne service
JMP$09B7      rebouclage forcé

```

André C.