MICRO STRAD
LA REVUE DES MICROS AMSTRAD

Mr Jacques LÖVI

Paris, le 7/7/86

[redacted]

Belgique

Madame, Monsieur,

Nous avons le plaisir de vous informer que nous avons retenu pour publication votre dossier enregistré sous la référence AMS 187

Cette publication aura probablement lieu (sans engagement de notre part) à la date approximative suivante:

☐ Dans le plus prochain numéro de MICROSTRAD.
☒ Dans l'un des 3 prochains numéros.
☐ Dans l'un des 6 prochains numéros.
☐ .....

Nous vous prions de bien vouloir nous signaler par retour du courrier si vous maintenez votre demande de publication, compte tenu du délai indiqué ci-dessus.

Attention, au cas où vous auriez envoyé un dossier semblable à une autre revue, il vous appartient de veiller à ce qu'une double publication n'ait pas lieu...

Nous attendons votre réponse. Utilisez de préférence pour cet usage le verso de cette lettre.

Bien amicalement,

La rédaction.

[redacted]

~~PJ: votre cassette ou disquette.~~

ÉLOI et Cie
Société en nom collectif au capital de 20 000 F
R.C. Paris B 328 451 521
Siège social
5 place du Colonel Fabien
75491 PARIS CEDEX 10
Tél (1) 240 22 01
Télex : LORDI 215 105 F

groupe tests publications

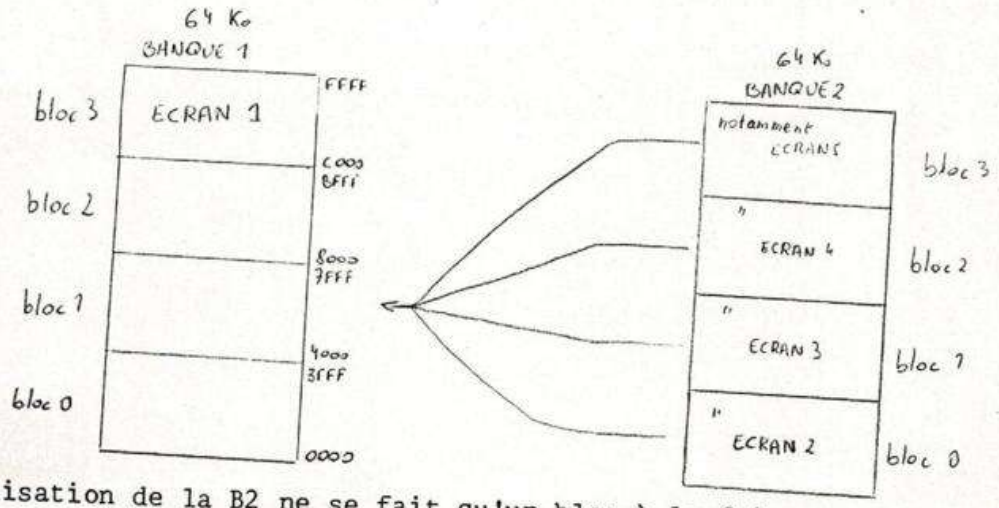
déjà envoyé à MS-Stead.

CPC-6128 : UN "DISQUE VIRTUEL" PAS SI VIRTUEL QUE CA... ET PAS SI DISQUE NON PLUS !

La firme AMSTRAD a jusqu'aujourd'hui incroyablement déprécié son CPC-6128 (~~cet article ne concerne pas les 464 et 664~~). En effet, dans son manuel d'utilisation, elle appelle les 64 Ko supplémentaires "DISQUE VIRTUEL" et fournit un programme générant des fontions d'extension RSX gérant ce D.V. (disque virtuel). Disque virtuel signifie qu'il ne s'agirait que d'une RAM de stockage; or, on peut, moyennant quelques connaissances, y loger des programmes tout à fait exécutables . Il y a donc bien 106 Ko programmables sous le clavier du 6128 et non pas 42Ko (64 Ko - 16 Ko écran - 6 Ko gestion système). A noter qu'en cas d'utilisation de la 2e banque RAM pour des programmes, les écrans 2 à 5 et les fichiers (BANKREAD, BANKWRITE ...) ne sont évidemment plus utilisables, mais on peut envisager l'utilisation des écrans 2 à 4 et de 16 Ko de programme en bloc 3 de la Banque 2 (b3B2), par exemple.

Une autre incroyable bévue de la firme AMSTRAD est le fait qu'elle ait protégé son utilitaire BANK MANAGER. Dès lors, comment savoir, si l'on désire utiliser la B2, si notre programme ne chevauchera pas les fonctions RSX, donc les effacera en tout ou en partie, puisque l'on ne sait pas où ces fonctions se situent en RAM. Donc, l'alternative serait de soit utiliser un programme assembleur, soit la B2; mais en aucun cas les deux ensemble... Heureusement que les fouineur sont là !...

Pour comprendre comment "ça marche là-d'dans", il faut d'abord parler d'architecture interne (hé oui...). Les 128 Ko de RAM sont divisés en 2 banques (B1 et B2) de 64 Ko chacune. Ces banques sont elles-même divisées en quatre blocs de 16 Ko chacun (b1,b2, b3 et b4).



L'utilisation de la B2 ne se fait qu'un bloc à la fois, ce qui amène une contrainte au programmeur - la seule - qui est qu'il faut structurer les programmes en 6, 7 parties de 16 Ko au maximum réparties dans les différents blocs de la B1 et de la B2.
Maintenant, le plus important : pour utiliser la B2, il faut effectuer ce qu'on appelle une commutation de mémoire, c'est-à-dire que l'on change la disposition interne des des huit blocs de 16 Ko. Pour effectuer cette commutation, on utilise le port d'adresse &7F00 auquel on envoie une des valeurs suivantes : &C0, &C4, &C5, &C6 ou &C7. Ces valeurs donnent les configurations suivantes :

&C0 : état de départ	b0B1 b1B1 b2B1 b3B1 - b0B2 b1B2 b2B2 b3B2
&C4 : échange de b0B2	b0B1 b0B2 b2B1 b3B1 - ...
&C5 : échange de b1B2	b0B1 b1B2 b2B1 b3B1 - ...
&C6 : échange de b2B2	b0B1 b2B2 b2B1 b3B1 - ...
&C7 : échange de b3B2	b0B1 b3B2 b2B1 b3B1 - ...

La configuration de la deuxième banque, après commutation, n'a aucune importance et il est d'ailleurs impossible (du moins je pense) de la déterminer avec certitude puisque les blocs de la banque 2 peuvent très bien être échangés avec le b1B1, ou simplement le "masquer" (je pencherais plutot pour cette dernière hypothèse, vu la rapidité de la commutation).

note : B1 = Banque 1 B2 = banque 2
b0 = bloc 0 b2 = bloc 2 ...

Pour ceux qui ne seraient pas encore initiés à l'adressage de ports, sachez que pour envoyer la valeur &C4 au port d'adresse &7F00, il faut effectuer l'instruction OUT &7F00, &C4.

Pour vérifier si je ne suis pas en train de divaguer, il vous suffit d'entrer un programme basic de plus de 16 Ko. en mémoire, de faire l'instruction OUT &7F00,&C4, et de lister le programme; vous constaterez à un certain moment que le listing se plante...

Vous trouverez ci-dessus un programme assembleur, et pour ceux qui n'ont pas de programme d'assemblage son chargeur basic. Ce programme réalise l'instruction RSX |SCREENSWAP,n et permet l'échange entre les écran 1 et n (n=2,3,4,5). La syntaxe est différente du |SCREENSWAP,n,n' du BANK MANAGER et ne permet pas l'échange d'écrans tels que pour n=2 et n'=3 par exemple.* En revanche, "mon" |SCREENSWAP et plus rapide que l'autre (87 unités TIME contre 99) et ne dure que 0,29 seconde.

A noter, et ce n'est pas le moins important que l'instruction OUT &7F00,&Cx (x=0,4-7) ne prend pas 1/300e de seconde puisque l'instruction suivante affiche zéro :

```
t=TIME:OUT &7F00,&C4:PRINT TIME-t
```

En fait, la commutation prend, en basic, en première approximation entre 0.37 et 0.52 (je pense 0.47±0.01) milliseconde, soit pratiquement rien du tout...

Tous les problèmes d'utilisation des 64Ko de la banque 2 devraient ainsi être éclaircis. Il ne vous reste donc plus qu'à écrire des programmes les utilisant... Bonne chance.

LöVi, Jacques.

BELGIQUE.)

* CE QUI EST D'AILLEURS TRES RAREMENT NECESSAIRE.

SCREENSWAP,n (ASSEMBLEUR Z80).

```
8CA0    10                                  ORG &8000
8000    20                          BUFF:   DEFS &04
8004    30    010E80                        LD BC,EXTCOM:
8007    40    210080                        LD HL,BUFF:
800A    50    CDD1BC                        CALL &BCD1
800D    60    C9                            RET
800E    70    1380                EXTCOM:   DEFW NENAM:
8010    80    C31E80                        JP SW:
8013    90    53435245454E535741  NENAM:    DEFM "SCREENSWA"
801C    100   D0                            DEFB "P"+&80
801D    110   00                            DEFB &0
801E    120   DD7E00                SW:     LD A,(IX+0)
8021    130   0602                          LD B,2
8023    140   B8                            CP B
8024    150   D8                            RET C
8025    160   0606                          LD B,6
8027  ! 165   B8                            CP B
8028    170   D0                            RET NC
8029    180   C6C3                          ADD A,&C3
802B    190   01007F                        LD BC,&7F00
802E    200   ED79                          OUT (C),A
8030    210   2100C0                        LD HL,&C000
8033    220   110040                        LD DE,&4000
8036    230   010040                        LD BC,&4000
8039    240   1A                  A:        LD A,(DE)
803A    250   EDA0                          LDI
803C    260   2B                            DEC HL
803D    270   77                            LD (HL),A
803E    280   23                            INC HL
803F    290   EA3980                        JP PE,A:
8042    300   3EC0                          LD A,&C0
8044    310   01007F                        LD BC,&7F00
8047    320   ED79                          OUT (C),A
8049    330   C9                            RET
```

CP 2.

CP PG

INITIALISER AVEC CALL &8000.

CHARGEUR BASIC.

```
10      ADR=&8000
20      DATA 01,0E,80,21,00,80,CD,D1,BC,C9,13,80,C3,1E,80
30      DATA 53,43,52,45,45,4E,53,57,41,D0,00,DD,7E,00,06
40      DATA 02,B8,D8,06,06,B8,D0,C6,C3,01,00,7F,ED,79,21
50      DATA 00,C0,11,00,40,01,00,40,1A,ED,A0,2B,77,23,EA
60      DATA 39,80,3E,C0,01,00,7F,ED,79,C9
70      MEMORY ADR-1
80      FOR I=ADR TO ADR+69
90      READ B$
100     POKE I,VAL("&"+B$)
110     NEXT I
120     CALL &8000        'Initialisation
130     DELETE 10-130
```