La Commode

Le Magazine des Ordinateurs Commodore Atari et Oric

Les disques ATARI et ORIC

Les notes sur 64

Display List sur ATARI

Architecture de l'ATARI

Copie haute-résolution VIC

Le sommaire complet est en page 4

n° 11
Juillet 84
45 Francs

SPiD PRÉSENTE

LE N°2

LISTE DES POINTS DE VENTE

06000 - MAD'S - NICE - (93) 88.04.79
06210 - ÉVOLUTION 2000 - MANDELIEU - (93) 49.81.61
08600 - MICRO-BOUTIQUE JCR - GIVET - (24) 55.01.23
10000 - MICROPOLIS - TROYES - (25) 72.03.79
11000 - I ÉLEC VIDEO CLUB - CARCASSONNE - (68) 47.08.94
11000 - R 2 I INFORMATIQUE - NARBONNE - (68) 65.15.83
12000 - BASE 2 SOCODETI - RODEZ - (65) 42.50.05
13004 - ALLIANCE - MARSEILLE - (91) 86.35.99
13005 - ELP INFO - MARSEILLE - (91) 94.91.13
13006 - MD SYSTÈME - JCR BOUTIQUE - MARSEILLE - (91) 37.62.33
13200 - LUDO - ARLES - (90) 96.79.03
14000 - OMB-VASSARD TILLIETTE - CAEN - (31) 93.48.09
16000 - S.A. LHOMME - ANGOULÊME - (45) 92.27.37
18000 - AVENIR INFORMATIQUE - BOURGES (48) 65.16.57
19100 - MICROMATIC - BRIVE - (55) 87.15.17
19100 - INFORMATIC 19 - BRIVE - (55) 87.77.08
21000 - O.M.G. MICRO LEADER - DIJON - (80) 30.12.70 +
24100 - MICRO CYRANO INFORMATIQUE - BERGERAC - (16) 56.06.06.12 +
25206 - ITA MONTBELIARD - MONTBELLIARD CEDEX - (81) 94.50.65
26000 - DOMICA - VALENCE - (75) 41.14.75
26500 - ECA ÉLECTRONIQUE - BOURG-LES-VALENCE - (75) 42.68.88
29000 - L'ORDINATEUR 29 - QUIMPER - (98) 95.92.70
30000 - DISCOUNT INFORM. SERVICE - NIMES - (66) 23.74.21
31000 - MICRO DIFFUSION - TOULOUSE - (61) 22.81.17
33000 - MICRO DIFFUSION - BORDEAUX - (56) 81.11.99
33800 - ETS COCA - BORDEAUX - (61) 92.91.78
34006 - PIB - JCR BOUTIQUE - MONTPELLIER - (67) 58.84.37
34200 - BUREAU ORGANISATION - SÈTE - (67) 74.34.10
34500 - MARCELLEC - BÉZIERS - (67) 31.37.65
37170 - LIM - CHAMBRAY-LES-TOURS - (47) 27.29.00
38500 - MICRO AVENIR - VOIRON - (76) 65.72.55
39000 - MICRO 39 - JEAN-PIERRE-ANDRÉ - LONS-LE-SAUNIER (84) 24.45.39
41500 - T.I.M. - MER - (54) 81.62.47
42000 - DÉTROIT INFORMATIQUE - SAINT-ÉTIENNE - (77) 33.58.59
42100 - SAINT-ÉTIENNE COMPOSANTS - SAINT-ÉTIENNE (77) 33.50.14
42300 - MICRO SYSTÈME RHONE-ALPES - ROANNE - (77) 68.67.99 +
44100 - SILICONE VALLÉE - NANTES - (40) 73.21.67
45000 - TÉLÉPHONIE BIS - ORLÉANS - (38) 54.34.34
47000 - JULIEN ÉLECTRONIQUE - AGEN - (58) 66.55.64
49000 - TEMPS X - ANGERS - (41) 88.95.07
49300 - CHOLET INFORMATIQUE - CHOLET - (41) 46.02.40
54000 - SÉREC - NANCY - (8) 332.12.60
56000 - L'ORDINATEUR 56 - VANNES - (97) 42.52.20
56100 - L'ORDINATEUR 56 - LORIENT - (97) 64.52.54
57504 - ARGO INFORMATIQUE - SAINT-AVOLD - (87) 92.54.84 +
57800 - CMI - FREYMING MERLEBACH - (87) 81.14.89
59000 - ETS BOULANGER - LILLE - (20) 54.98.75
59000 - BÉCY INFORMATIQUE - LILLE - (20) 92.33.06
59400 - MICROSHOP - CAMBRAI - (27) 81.98.09 +
59500 - CID - DOUAI - (27) 88.47.20
59800 - M.B.D.C. - LILLE - (20) 57.91.87
60108 - QUENEUTTE - CREIL - (4) 425.04.26
60200 - LARDET S.A. - COMPIÈGNE - (4) 423.07.86
63000 - IMPACT - CLERMONT-FERRAND - (73) 92.17.55
64110 - ESPACE MICRO 64 - BAYONNE - (59) 59.41.55
64600 - INFORMATIQUE BASCO LANDAISE - ANGLET - (59) 31.96.05
66000 - SÉRIE INFORMATIQUE - PERPIGNAN - (68) 34.00.11
67150 - ETS A FRITSCH - ERSTEIN - (88) 98.03.51
68000 - E.I.B. - COLMAR (89) 23.68.35
69003 - B.I.M.P. - LYON (7) 860.84.27
69400 - MICRO INFORM.BEAUJOLAISE - VILLEFRANCHE-S/SAONE - (74) 68.44.92
70000 - ÉLECTRO BOUTIQUE - VESOUL - (84) 76.49.52 +
71100 - AVENIR ÉLECTRONIQUE - CHALON/SAONE - (85) 48.73.35
71400 - C.H.B. ÉLECTRONIQUE - AUTUN - (85) 52.70.26
72000 - MICROTIQUE AESCULAPPLE - LE MANS - (43) 24.97.80
73100 - L'ORDINATEUR - AIX-LES-BAINS - (79) 88.19.07
74102 - D.S.A. MICRO - ANNEMASSE - (50) 38.31.40
75001 - VIDEO SHOP - PARIS - (1) 296.93.95
75005 - HACHETTE - PARIS - 633.84.68
75006 - DURIEZ S.A. - PARIS - 329.05.60
75008 - ÉNERGY 8 - PARIS - 293.41.33
75009 - LE JEU ÉLECTRONIQUE - PARIS - 526.62.93 / 874.43.20
75009 - LPS BUREAU - PARIS - 878.26.45
75009 - J.C.R. ÉLECTRONIQUE - PARIS - 282.19.80
75010 - GÉNÉRAL VIDEO - PARIS - 206.50.50
75010 - LOGIC STORE - PARIS - 206.72.28
75011 - COCONUT INFORMATIQUE - PARIS - 355.63.00
75011 - P.I.T.B. - PARIS - 254.38.01
75012 - ELLIX - PARIS - 307.65.58
75014 - MIDEF - PARIS - 539.98.68
75015 - J.C.S. COMPOSANTS - PARIS - 355.96.22
75015 - ILLEL CENTRE - PARIS - 554.97.48
75016 - PENTASONIC - PARIS - 524.23.16
75016 - ANTIGONE - PARIS - 743.13.41
76600 - MICRO MAX - LE HAVRE - (35) 41.77.47
76600 - V.P.C. BUREAU - LE HAVRE - (35) 42.49.21
76600 - L'ORDINATEUR - LE HAVRE - (35) 21.54.55
77000 - EPSILON - MELUN - 437.51.95
80000 - LOGIC - AMIENS - (22) 95.54.84
83000 - P.S.I. ÉLECTRONIQUE - TOULON - (94) 93.11.20
86011 - LISTE INFORMATIQUE - POITIERS CEDEX - (49) 41.43.86
87000 - MICROLIM - LIMOGES - (55) 34.10.12 +
89100 - MINI LOISIRS - SENS - (86) 64.41.91
89100 - LASOBIKOR YONNE - SENS (86) 64.51.26 +
91210 - VIDEOTRONIC - DRAVEIL - 940.28.30
92100 - AXIOME - BOULOGNE - 604.02.21
92100 - OLIG - BOULOGNE BILL. - (1) 605.05.59
94100 - DIXMA - SAINT-MAUR - 885.98.22
98000 - MICROTEK 2 - MONACO - (93) 30.67.67 +
88002 - A.V.M. - ÉPINAL (29) 82.14.97

SUCCÈS OBLIGE

Le deuxième d'une longue série de guide des logiciels.
Plus d'un tiers de nouveautés.

AU SOMMAIRE :

— Une sélection de 416 programmes en Anglais ou en Français pour :
APPLE - ATARI - COMMODORE V20 et C64 - EPSON HX 20 - ORIC 1 et ORIC ATMOS - IBM PC - SINCLAIR ZX81 et SPECTRUM TRS 80 - THOMSON TO 7 - HECTOR.

— Les fiches techniques de chaque programme comprenant :
La description précise du programme.
Son prix moyen constaté.
Sa compatibilité avec tel ou tel micro.

— En plus vous trouverez :
Des conseils pour choisir et acheter le programme que vous cherchez.
Des index pour trouver facilement ce que vous cherchez.

EN VENTE 15 F CHEZ VOTRE DISTRIBUTEUR OU 15 F + 5 F DE PORT EN RENVOYANT LE COUPON CI-DESSOUS.

SPiD
LA HAUTE FIABILITÉ

BON DE COMMANDE A RENVOYER A **SPID** - 39, RUE V.-MASSÉ - 75009 PARIS

Je désire recevoir le "GUIDE DES LOGICIELS" Printemps 1984
Je joins 20 F en chèque (15 F+ 5 F de port) en règlement.

Nom .....

Adresse .....

Code et ville .....

Editorial

De Profondis ! Vous avez en main le dernier numéro de La Commode. Raison de plus pour le lire avec attention. Mais La Commode ne disparaît pas complètement. En fait, elle éclate en trois revues consacrées chacune à une marque : Commodore, Atari et Oric. La revue consacrée aux Commodore s'appelle 64 Magazine. Son premier numéro paraîtra à la date où serait paru le n° 12 de La Commode, donc pour le SICOB. Nous nous excusons de demander un peu de patience à nos amis Ataristes et Oriciens : leurs revues sont à l'étude mais il nous est impossible d'en lancer trois simultanément.

Daniel-Jean DAVID

N.B. : Nos abonnés recevront personnellement une circulaire leur donnant toutes informations sur leur cas.

Jouez la bonne carte avec SEDERMI

SEDERMI (société éditrice de La Commode) présente un ensemble de cartes d'extensions pour le VIC, caractérisées par un rapport qualité/prix sans concurrence. En préparation, systèmes de cartes pour 64 et ORIC.

Ce sont :

a — Les cartes EXTVIC-BUS 4/EXT64-BUS 4
qui permettent de connecter simultanément 4 cartouches ou extensions à votre VIC/64, par exemple VIC-MON, PROGRAMMER'S AID, SUPER EXPANDER et une extension mémoire, ou toute autre combinaison.
La carte EXTVIC-BUS 4 existe en 3 options :
— N La carte montée et testée, avec alimentation par le VIC ; c'est l'option normale.
— E Comme N, mais alimentation extérieure (non fournie). Cette option est utile pour certaines extensions d'entrées-sorties et applications particulières.
— S Permettant, grâce à 4 interrupteurs d'activer la cartouche voulue et ainsi d'avoir 4 cartouches de jeu ou Super Expander montées en permanence.

La carte EXT64-BUS 4 n'existe qu'en option S.

b) Les cartes EXT...-VIA/PIA 2 (pour VIC, 64, ORIC)
qui procurent 40 lignes d'entrées-sorties supplémentaires (idéal pour commander des appareils domestiques, des trains électriques etc.).

Chaque carte contient deux PIA 6821 ou VIA 6522 (qui procurent en plus 4 temporisateurs programmables). Il y a 4 adresses de base au choix (à préciser à la commande). On peut connecter simultanément 4 cartes analogues à un VIC ou à un 64, 2 à un ORIC.

c — La carte EXTVIC-EPROM 4
qui contient 4 supports d'EPROM 2532, ce qui vous permet d'avoir vos programmes permanents figés en EPROM et disponibles dès la mise sous tension.
Chaque EPROM peut être activée ou inhibée par interrupteur et être implantée au choix en 4 adresses.

d — L'interface ORIC-JOY 1 qui permet de connecter un manche à balais Commodore, Atari ou compatible à votre ORIC pour 70 F.

e — L'interface magnéto qui permet de connecter un magnéto ordinaire à votre Commodore. La version A ne commande pas le moteur, la version B (recommandée pour le 64) le commande.

En préparation, cartes relais, programmateur d'EPROM, etc.

BON DE COMMANDE

à adresser avec le règlement à SEDERMI, 28, rue Vicq d'Azir, 75010 Paris
(délai à prévoir : un mois environ)

Nom : .....
Adresse : .....
.....

.... ORIC JOY1	(70 F TTC) :	:F TTC	
.... EXT64-BUS4-S	(480 F TTC) :		
.... EXTVIC-BUS4-S	(480 F TTC :	:	règlement par
..... EXTVIC-BUS4-N	(390 F TTC) :	:	CCP : 0
..... EXTVIC-BUS4-E	(390 F TTC) :	:	CB : 0
..... EXTVIC-PIA2	(690 F TTC) :	:	mandat : 0
..... EXTVIC-VIA2	(730 F TTC) :	:	
..... EXT64-PIA2	(690 F TTC) :	:	
..... EXT64-VIA2	(730 F TTC) :	:	
..... EXTORIC-PIA2	(690 F TTC) :	:	
..... EXTORIC-VIA2	(730 F TTC) :	:	
..... EXTVIC-EPROM4	(310 F TTC) :	:	
..... INT-MAGNETO A	(120 F TTC) :	:	
..... INT-MAGNETO B	(170 F TTC) :	:	
TOTAL		F TTC	

Sommaire n° 11

Courrier des lecteurs

L'adresse 50003

1 / Je possède 8 mémoires statiques 8116. Peut-on les placer à la place des 4116 disponibles ?

2 / A quoi sert l'adresse PEEK (50003) que je retrouve sur des programmes divers (de PET)

Je voudrais aussi que vous me communiquiez l'adresse de votre boutique.

Je félicite l'équipe pour sa compétence. Je considère que votre revue est une grande source d'astuces et une source de développement sûre et intéressante.

Laurent HADVAD
94300 VINCENNES

1 / Nous ne connaissons pas la mémoire 8116. Il est possible qu'elle soit compatible avec la 4116 mais, à ce moment, elle ne serait pas statique. Donc, prudence.

2 / L'adresse 50003 contient 0 sur PET, 1 sur CBM 3000, 160 en BASIC 4, elle sert à discriminer le modèle depuis un programme.

3 / Nous n'avons pas de boutique. LA COMMODE est en vente chez les distributeurs PSI ou par correspondance.

*
* *

Supprimer le curseur

Je souhaiterais savoir comment à l'aide d'un POKE supprimer le curseur tout en pouvant continuer à écrire.

Pourriez-vous publier une table complète d'adressage pour le VIC (et pour d'autres CBM).

Arlette et Lucien DIETRICH
67000 STRASBOURG

Ce n'est pas un POKE qui empêche le curseur lors d'INPUT. Il faut faire les entrées par GET (et alors un POKE 204,0 fait clignoter le curseur)

*
* *

La télé qui déforme

J'ai trouvé moi-même la réponse à ma question sur la déformation de l'affichage du téléviseur. Elle n'est pas inhérente au codage, mais à la fréquence du quartz du VIC 20. En effet, il suffit de retoucher le réglage de ce que je suppose être une self pour voir disparaître le phénomène (probablement une différence entre la fréquence de rafraîchissement du VIC et la fréquence interne balayage-lignes du téléviseur.

M. Alain NOURY
91300 MASSY

*
* *

Problèmes disques

Possesseur d'un CBM 3032 associé à une imprimante 4022 et à une unité double disque 4040, il m'arrive d'avoir des ennuis avec cette dernière.

Je m'empresse de dire que ce n'est pas la règle générale, mais il arrive que, sur un disque contenant des programmes, il ne soit pas possible de retrouver ceux-ci.

Ceci apparaît au moment où on étend un fichier (programme ou séquentiel) d'une quantité assez grande de données, alors même que le disque n'est pas rempli à moitié. En somme, il semble qu'il y ait chevauchement des données sur le disque.

Pouvez-vous me dire s'il est possible que je me livre à une fausse manoeuvre à un moment donné, ou s'il y a des précautions spéciales à prendre dans ce cas ?

De plus, le répertoire affiche toujours les programmes devenus impossibles à charger...

Je vous remercie d'avance des conseils que vous pourriez me donner à ce sujet, et vous félicite pour la tenue de votre revue dont je suis un fidèle abonné.

Marcel DURVAUX
B 6110 MONTIGNY-LE-TILLEUL

Le système disque gère les données sous forme de liste chaînée et il lui arrive de s'emmêler les pédales dans les chaînages. Nous n'y connaissons pas de remède sauf:

- éviter les disques trop pleins,
- éviter les scratch et les save avec remplacement (@)
- éviter les rename.

*
* *

Boite à questions

Je me permets de vous féliciter pour votre n°5. Quel progrès et quelle qualité (= utilité) des programmes et idées ! Je crois que vous arrivez très vite à quelque chose d'essentiel. A quand une parution plus fréquente ?

Je vous propose quelques idées:

1 - Mettre le feedback derrière une page de publicité et non d'articles; ça va, cette fois, l'article n'était pas pour "mon" ordinateur, mais je suis sûr que certains vont hésiter à couper une page de valeur !

2 - S'il vous plait !!! Editez une table de correspondance entre les zones mémoires CBM - PET -- VIC, CBM - PET c'est presque complet, mais VIC reste derrière avec des programmes inadaptables.

A ce propos, je peux vous proposer la gestion de fichier pour VIC "avec un tri des DATA". Si ça vous intéresse. De toutes façons, les conversions de POKE sont ici très simples :

POKE	VIC	VIC
WAIT 158 =	198; WAIT 152 =	653

POKE 623 = 631; POKE 624 = 632
POKE 826 = 1 (par exemple).

Mais quelle envie me reste-t-il de pouvoir adapter les programmes donnant de nouvelles instructions !

3 - Pourriez-vous créer une rubrique "Boîte à questions" où les lecteurs vous enverraient leurs problèmes. Ceux-ci pourraient être répondus par vous ou par d'autres lecteurs dans les numéros suivants.

Pour vous aider à commencer, voici celles que j'ai :

a) Comment peut-on écrire de droite à gauche de l'écran de façon rapide. (J'ai un programme mais en BASIC on ne va pas vite). Ceci est indispensable pour l'écriture dans certaines langues. Y aurait-il un POKE par là-bas derrière qui ferait le travail vite et bien ?

b) Comment peut-on "diriger" de la mémoire en cartouche vers certaines adresses ?
Ex. : la cartouche 16K met tout en bloc. Je suis intéressé d'en mettre en $A000 (et oui ! VIC REVEALED donne une bonne liste d'instructions nouvelles à placer à cette adresse). Mais comment sans "fer à souder" pourrait-on réaliser cet exploit ?

c) Peut-on redéfinir les caractères en 5x7 (au lieu de 8x8) et réutiliser les "morceaux d'octets vides" pour obtenir plus de caractères par ligne ?

d) L'imprimante VIC a-t-elle, oui ou non, les accents ? Les versions semblent diverger. Peut-on, oui ou non, avoir en sortie imprimante un texte accentué, si ainsi ont été redéfinis les caractères dans le générateur en RAM ? -

e) Les prix vont-ils baisser substantiellement avec la sortie du VIC 64 ? Les éléments communs (accessoires) vont-ils aussi baisser ?

Bon, ma dernière question à propos de l'EXTENSION EXTVIC-BUS 4 Je suis très, très intéressé, mais n'y connaissant rien en électronique, quelle est la différence (et l'utilité) entre BUS4-N et BUS4-E . Y a-t-il avantage à avoir une alimentation extérieure (et si oui, quelle doit-elle être et où se la procurer) ? Quel avantage ou danger avec l'alimentation directe sur le VIC ? Et, enfin, à part le chassis et le prix, quelle est la différence entre votre extension et celle de COMMODORE ? J'attends vivement vos conseils pour pouvoir me décider et (enfin!) pouvoir utiliser mes diverses extensions simultanément ! Ecrivez vite s'il vous plait !

Voilà, j'espère de tout coeur que ces idées pourront collaborer à faire une revue toujours supérieure.

Francis ESTEVE
38290 VILLEFONTAINE

1 - Suggestion retenue (cf n°5)

2 - cf. n°6,page 35. Votre programme de GESTION DE FICHIER nous intéresse.

3 - Bonne suggestion aussi mais la périodicité de trois mois ne facilite pas.

a - à part POKE dans la mémoire d'écran ou le langage machine, il n'y a rien.

b - les cartouches 8K ont des interrupteurs qui permettent de "diriger" les adresses sans fer à souder (mais avec ouverture de la cartouche). Si les 16K n'ont pas les interrupteurs, il faut jouer du fer à souder. Si vous ne voulez pas ouvrir la cartouche, on peut jouer du fer à souder sur une carte EXT VIC BUS4.

c - NON. La cassette qui fournit 40 col. fait comme si elle traçait chaque caractère en tant que courbe haute résolution.

d - L'imprimante VIC a les accents, à condition de les programmer comme caractères graphiques.

e - Les prix ont effectivement baissé.

4 - La carte la plus normale est la N. La E ne sert que pour des applications bien particulières d'entrées-sorties.

*
* *

Adaptation des programmes graphiques

Je viens juste de recevoir mon VIC 20 et une extension 16K. Par ailleurs, pour m'initier au BASIC, j'ai lu le livre de Daniel-Jean DAVID: "La découverte du VIC" et je me suis intéressé au graphisme haute résolution. Mais le programme de tracé de courbes de Daniel-Jean DAVID se limitant à un VIC de base (3,5K de mémoire), pourriez-vous m'indiquer un programme adapté à 16K, qui me permettrait d'avoir le maximum de points sur l'écran (176x184), ou tout-au-moins les "POKAGES" de la ligne 10,20, les valeurs de E,G, de CO, et la valeur maximale de N (cf. test de la ligne 2000). (Le programme de tracé de courbes est à la page 124-125 de "La découverte du VIC")

Par ailleurs, pourriez-vous m'indiquer s'il existe une façon de redimensionner un tableau dans un programme afin d'éviter le message d'erreur : REDIM'D ARRAY. Si oui, laquelle ?

Adam BURE
76133 MANEGLISE

1 - La conversion des programmes haute résolution pour VIC étendu a été traitée en détail dans LA COMMODE n°8. Sinon, l'article du n°3,p. 31 devrait vous aider sachant qu'il faut envoyer le début de BASIC en $2000 par POKE 8192,0: POKE 44,32 : CLR (avant d'entrer votre programme).

2 - Pour redimensionner un tableau, faire CLR avant, mais attention, cela supprime toutes les variables.

*
* *

Comment connecter beaucoup de cartouches

J'aimerais connaître les modalités de connexion (si cette opération est réalisable) de votre carte EXT VIC - BUS4 pour plus de 4 cartouches :

VIC MON
PROGRAMMER'AID
SUPER EXPANDER + 3K
SCREEN MASTER
RAM 8
RAM 16
soit les 32 K RAM maximum.

Est-il possible d'utiliser avec une alim. 5V extérieure deux de vos cartes EXT VIC - BUS4 en parallèle pour obtenir 8 connexions ?

Gilles BROCARD
93250 VILLEMOMBLE

On peut connecter une deuxième BUS4 sur la première ce qui permet 7 extensions, mais même en alim. extérieure, attention à la fatigue du BUS. Dans les connexions que vous proposez, SUPER EXPANDER et SCREEN MASTER sont incompatibles : il faut un modèle S dont les interrupteurs permettent de désactiver l'une ou l'autre cartouche.

*
* *

Problèmes graphiques sur VIC

J'aimerais avoir quelques renseignements : je suis l'heureux possesseur d'un VIC 20 et d'une extension 16 K .

Dans le cas d'un VIC version de base :
- Par quels moyens, après avoir déplacé l'adresse du générateur de caractères en 5120, ce qui permet de redéfinir 320 caractères programmables, peut-on afficher sur l'écran ces 320 caractères, compte tenu que la valeur maximum autorisée dans un POKE est de 255 ?

Dans le cas d'un VIC et d'une extension 16K :
- Quels sont les moyens de faire de la haute résolution et de redéfinir un jeu de caractères programmables?

- Quelles valeurs doit-on mettre en 36869 (registre adresse position de mémoire écran et position du générateur de caractères) ?

Dans l'attente de votre réponse, sachez que je trouve votre revue passionnante, bourrée d'astuces et de renseignements très intéressants.

- Envisagez-vous une série d'articles sur le langage machine (initiation et perfectionnement) ?

- Continuez comme cela, à nous donner quantités d'articles d'intérêt général comme "Qui a dit que BASIC n'était pas structuré ?", ou d'articles plus spécialisés même s'ils ne correspondent pas toujours à notre propre matériel, mais qui, malgré tout, sont toujours bénéfiques.

Didier BENEY
71130 GUEUGNON

Vous ne pouvez pas avoir 320 caractères programmables. Il faut se limiter à 255, c'est pourquoi pour la HR en mode statique, on passe à 16 lignes élémentaires par maille (Cf. LA COMMODE n°5,p.36)

Avec l'extension 16K, faire :
G = 6144 : Po 36869,254

et mettre le début BASIC en $2000 par

Po 44,32 : Po 46,32 : Po 48,32 : Po 50,32: CLR.

La série sur l'initiation au L.M. a commencé au n°9. Nous pensons, comme vous, que, même pour un autre matériel, les articles peuvent être bénéfiques. Donc, Commodoristes, lisez les articles sur ORIC et ATARI, et vice-versa.

*
* *

Encore les touches de fonction

Très intéressé par votre revue LA COMMODE et par les livres que vous avez édités sur le VIC 20, je souhaiterais vous voir répondre aux questions suivantes :

1) Avec une extension 16 Ko, je désire placer le générateur de caractères tout en bas des RAM . La méthode est claire avec le VIC de base et le générateur placé avant la mémoire écran. Avec le 16 Ko, comment opérer ?

2 - Comment programmer les touches fonctions 1 à 8 pour pouvoir les utiliser pour la rédaction d'un programme ? (La réponse faite à un lecteur, dans le n° 6 de LA COMMODE, indique que la réponse se trouve dans la rubrique "Vic à Brac", je n'ai rien trouvé dans la page indiquée...)

3 - Comment programmer AUTO-RUN avec le VIC 20 ?

Etienne CIMETIERE
14000 CAEN

1) La technique la plus simple est de repousser le début de BASIC au-dessus de $2000 par :

```
POKE 8192,0 : POKE 44,32 : NEW
```

2) Pour déceler les touches de fonction faire PEEK (203) et PEEK (653) :

Touche	PEEK(203)	PEEK(653)
F1	39	0
F2	39	1
F3	47	0
F4	47	1
F5	55	0
F6	55	1
F7	63	0
F8	63	1

3) Rendre un programme AUTO-RUN a été traité dans le n° 9.

*
* *

Les extensions du VIC

Je n'ai qu'une vague idée de ce que le VIC peut recevoir comme extensions :

1 - Si je vous commande une carte EXTVIC - BUS4 peut-on mettre 2 cartouches de 16K, un SUPER EXPANDER et un PROGRAMMER'S AID (ce qui porterait la mémoire à :
16 + 16 + 3 + 3,5 = 38,5 K ...

2 - Je souhaite trouver une solution économique pour augmenter le nombre de colonnes à l'écran (27 lignes sur 22 colonnes étant juste). Quel est le meilleur moyen ?

3 - Pouvez-vous me faire parvenir un tarif et/ou un catalogue de tout ce qui est actuellement disponible pour le VIC 20, soit distribué par PROCEP, soit par vos soins.

J'ai lu avec intérêt la plupart de vos articles, mais mes faibles connaissances de la programmation ne me permettent pas de très bien assimiler cette richesse de renseignements. J'aurais souhaité avoir plus d'explications sur les points suivants :

- La cassette UTIL VIC remplace-t-elle une cartouche déjà citée plus haut ?

- Les cartes EXTVIC-BUS4 (laquelle faut-il, la N ou la E ?)
- EXTVIC-VIA2 : A quoi sert-elle

exactement ? Personne n'a pu me dire quoi que ce soit (il est vrai que je ne connais pas grand monde). S'agit-il de l'extension de 40 lignes à l'écran ?

Quant à EXTVIC-EPROM4 c'est pour moi l'obscurité totale.

Fernand BOUGON
7614 PETIT-QUEVILLY

1 - On peut mettre 2 cartouches 16K sur une EXTVIC-BUS4 mais elles auront des adresses en recouvrement et ne vous donneront que 24K. Ou alors il faut changer un jumper ou un interrupteur mais alors on prend la place de SUPER EXPANDER.

2 - Cette question est reprise dans ce numéro.

3 - Voyez la description de nos produits. SUPER EXPANDER et PROGRAMMER'S AID sont distribués par PROCEP, non par nous;

4 - La cassette UTIL-VIC a certaines fonctions de PROGRAMMER'S AID et de SUPER EXPANDER, de façon simplifiée, mais bien moins chère. Elle ne les remplace pas. L'EXTVIC-BUS4 qui suffit à presque toutes les applications est la N . La E est pour les bricoleurs; EXTVIC-VIA2 rajoute des entrées-sorties (Ex: pour un train électrique). Voyez une application dans ce numéro.

*
* *

Problèmes de la 2031

LA COMMODE c'est encore mieux que ce que je pensais ! En effet, je m'attendais à n'avoir la réponse à mes questions sur la 2031 que dans le prochain numéro, c'est-à-dire sans doute en juillet. Eh non, la rédaction prend la peine de vous rédiger une réponse personnelle et manuscrite. Merci donc !

Si vous éditez une réponse concernant ma lettre, indiquez aux novices comment entrer les chiffres voulus. Je butais, en effet, sur cette question; grâce à votre réponse, l'illumination est venue : il faut entrer avec CHR$(X)=PAR EX. X=18 (decimal), donnera 12(hexa) sur la disquette.

Il semble, par contre, impossible de corriger le "BUG" de la 2031. L'octet en question correspond à 0; or, le pointeur minimum est un. La meilleure solution pour utiliser le programme DISPLAY serait sans doute d'incrémenter la piste demandée quand on a affaire à la piste 18.

Pour l'EDEX, la notice est en accord avec votre réponse, mais l'inconvénient signalé est que si vous lancez un programme où figure la commande "SYS46000" il n'est plus possible de revenir au "dos". Il est donc souhaitable de rédiger et tester les programmes sans introduire cette commande ni les PRINT USING,etc.., pour pouvoir manipuler sans souci.

J'ai eu depuis ma lettre une autre surprise avec les disquettes : j'ai acheté sur les conseils du vendeur de SDAI 32,avenue Garibaldi à Limoges, des MEMOREX, le fin du fin ! Et chers ! Mal m'en a pris sur les 5 disquettes : 2 donnent une fois sur deux READ ERROR 20, les 3 autres refusent tout formattage. J'ai demandé au vendeur de les tester, son lecteur (un GOUPIL) n'a pas non plus pu les digérer. Quelques jours après, ce monsieur m'a dit que c'était son lecteur qui était en cause. Curieux, n'est-ce-pas ? Il ne pouvait rien pour moi, même pas me donner l'adresse de la maison MEMOREX !! Je n'ai pratiquement pas eu de problèmes avec VERBATIM ni 3M; y a-t-il une explication (Ces disquettes paraîssent plus rigides que les autres. J'ai acheté les 5 disquettes au détail, peut-être est-ce un tort?). Quant à READ ERROR 20, j'aimerais en pénétrer les mystères ?

J.BOUTIQUE
87000 LIMOGES

LA COMMODE essaie de répondre directement aux questions pour éviter d'attendre un trimestre, mais cela nous est de plus en plus difficile.

Pour les disquettes, dont acte. Que nos lecteurs ne manquent pas de nous signaler les résultats de leurs expérimentations, pour que cela profite à d'autres.

*
* *

Bug sur le 2031

Comme l'indique B. MICHEL dans le livre de BCM p.195, j'ai voulu tenter l'expérience de récupérer un fichier effacé. Mon matériel : CBM 3008 monté à 32K, EDEX, FLOPPY 2031. Cela a été l'occasion de quelques découvertes et interrogations :

1 - La piste 18 secteur 0 indique pour les deux premiers octets 11 01 (Décimal : 17 01). D'après le manuel de la 2031, ce devrait être 12 01 (La prochaine piste étant effectivement 18 01 en décimal). Cela entraîne des difficultés lors de l'utilisation du programme DISPLAY T&S fourni sur la disquette de démonstration. On ne peut utiliser le "Voulez-vous la prochaine piste & secteur" en ce qui concerne la piste 18, l'erreur se reproduisant sur tous les secteurs. Est-ce la même chose pour toutes les 2031 ?

2 - Comment remplacer l'octet d'entête du fichier (00 puisqu'effacé) par, par exemple, 82 (hexadécimal) qui signifie programme. J'ai tenté d'utiliser le programme du volume 1 de "La Pratique du PET/CBM" de D.J. DAVID (Exercice 3-21). Cela semble convenir pour atteindre l'octet en question (piste 18 secteur 1 pointeur 3 par ex). Mais comment écrire 82 (130 décimal) ? Il faudrait sans doute un programme en langage machine.

3 - Comment comprendre le code des secteurs libres piste 18 secteur 0 pour les 35 pistes c'est clair, 4 octets par piste à partir du 5éme octet. Le premier octet indique le nombre de secteurs libres ? L'utilisation bit par bit, 0 ou 1, ne correspond pas à la réalité pour le 2éme et le 4éme octet ?

4 -Quand je charge le DOS et que j'exécute certains programmes (sous EDEX semble-t-il) il est impossible de revenir aux commandes du DOS (SYNTAX ERROR). Que se passe-t-il ?

J'attends toujours avec impatience le prochain numéro de LA COMMODE. C'est parfois un peu long, mais le retard est vite pardonné, car les découvertes sont toujours renouvelées. Je constate d'ailleurs en reprenant les premiers numéros un certain progrès dans ma capacité de comprendre des difficultés qui me semblaient inaccesibles . Seriez-vous pédagogues, performance pour des spécialistes de l'informatique dont le moins qu'on puisse dire, c'est qu'ils ne sont pas, en général, très soucieux de clarté dans leur message vers les personnes.(En direction de leur machine ça va sans doute mieux?) .

Jacques BOUTIQUE
87000 LIMOGES

1 - Il semble qu'il y ait un bug dans la 2031. On peut probablement rétablir l'octet à 18 comme en 2-

2 - Pour réécrire l'octet voulu, utiliser la commande U2 une fois que la bonne valeur a été mise dans le secteur en mémoire. Pas besoin de langage machine (cf. "La Pratique du VIC", vol. 1, p. 90, ou "La Pratique du CBM" vol. 1, p. 75).

3 - La représentation doit être exacte : 1 = libre, 0= occupé ou inexistant sur cette piste. On trouve, dans l'ordre, les 3 octets 76 54 32 10 15 14 13 12 11 10 9 8 et 23 22 21 20 19 18 17 16 . Pour la piste 18 on a 11 10 11 00 11 11 11 11 00 00 01 11 soit ECFF 07.

4 - EDEX prend le contrôle de CHRGET de façon incompatible avec le DOS. Il faut charger et lancer DOS support avant de lancer EDEX.

*
* *

Pauvreté des accessoires du VIC

Après de nombreux tâtonnements sur le VIC, on s'aperçoit que les espaces mémoire maximum disponibles sont les suivants :

de 4608(dec) à 32767(dec):BASIC:28K

de 1024(dec) à 4095(dec): Langage machine : 3 K

et, ce dont on ne parle que très peu :

de 40960(dec) à 49151(dec) : Langage machine : 8K

Nous avons donc un total disponible de 38K ! Il est vraiment dommage que l'espace mémoire n'ait pas été construit de façon à tout utiliser même en BASIC !

Tout ceci, ainsi que la façon de commuter les 8K RAM aux différentes adresses grâce à leurs interrupteurs internes (étudiés pour) n'est expliqué nulle part, et aucune publication ne donne des indications exactes.

De la même façon, les incompatibilités entre les différentes cassettes (RAM, SUPER EXPANDER, PROGRAMMER'S AID) ne sont que très rapidement évoquées.

Ceci dit, lors d'un récent voyage à Paris, j'ai essayé de me documenter sur les nouveautés sur le VIC. Quelle pauvreté ! Quand on lit ce qui se passe chez nos voisins anglais ou allemands, on peut se poser des questions sur les compétences de PROCEP, qui, sous l'excuse de "francisation" (de quoi? des modes d'emploi ?) commercialise du matériel avec des mois de retard; que dire aussi de tous les matériels adaptables, très peu présents en France ?

Prenons un exemple : sur la place de Paris, au début mars, rares étaient les distributeurs qui avaient plus d'une ou deux cassettes COMMODORE (que ce soit RAM ou jeux). Dans ces conditions, l'acheteur éventuel ira certainement vers d'autres matériels.

J'apprécie beaucoup votre revue, qui m'a tiré plus d'une épine du pied et m'a permis de mieux exploiter mon matériel dont je suis très content, par ailleurs.

Michel PILOT
TEMARA (Maroc)

Vos remarques sont justifiées: il n'y a, au maximum, que 27K pour BASIC. Les 3K ne sont utilisables que par POKE ou en LM. Les 8K de $A000 à $BFFF sont le plus souvent utilisés pour des extensions ROM (ex. SUPER EXPANDER). quand on arrive à se les procurer. On explique dans le n° 8 comment commuter les cartouches de RAM. Nous approuvons votre remarque sur la pauvreté de ce qu'on peut se procurer pour le VIC. Mais c'est surtout un problème de distributeurs qui ne font pas l'effort de se tenir au courant de ce qui existe et de suivre leurs approvisionnements; beaucoup n'ont même pas LA COMMODE !

*
* *

Tout vient à point

Je constate que je ne suis pas seul à attendre du matériel Commodore. Depuis février, j'attends pour mon VIC: le magnéto, Programmers' aid, Super Expander.

Votre revue est très bien faite, et pour ce qui est du retard, bah!! c'est bien peu de chose: il suffit d'avoir un ordinateur Commodore pour savoir ce que c'est l'attente... on s'en va vers des délais d'un an...

P.S.: un petit reproche: c'est dommage d'être obligé de découper "La Commode" pour commander vos produits; ne pourriez-vous pas faire une page à part pour les bons de commande?

Roger BENARD
10000 TROYES

Voici un lecteur qui prend les choses du bon côté ! Ah si tous les clients de Procep étaient comme cela; ils pourraient en faire encore moins !

Remarque: Inutile de découper les bons de commande dans notre revue: vous pouvez écrire sur papier libre.

*
* *

Atmosphère

Bravo à La Commode pour la qualité et la variété des articles. De plus vous critiquez sans complaisance les différents systèmes, tout en apportant des solutions, c'est ce que j'apprécie énormément.

Par ailleurs, vous avez une très bonne idée de commencer un cours de langage machine. Possesseur d'un ORIC ATMOS, je suis évidemment tout particulièrement intéressé par les articles le concernant.

Permettez moi de vous poser quelques questions:

1) On trouve dans toutes les revues des programmes de copie d'écran Haute Résolution pour ORIC sur imprimante Seikosha GP 100. Qui pourrait me faire un programme de copie d'écran Haute Résolution Graphique pour l'imprimante Oric 4 couleurs ? ou alors faut-il me résigner à changer d'imprimante ?

2) Comment faire avec l'ORIC ATMOS pour tracer en Haute Résolution des arcs de cercle de longueur et courbures diverses ?

3) A quand la suite de l'excellent livre de Monsieur D.J. David " A la découverte de l'ORIC" aux éditions PSI ? Livre qui permettrait l'approfondissement du BASIC et de l'ORIC ATMOS.

4) La cassette TELECRAN pour ORIC ATMOS est elle disponible ?

Pierre ANGELI
83000 TOULON

1) Un programme de recopie d'écran sur imprimante analyse l'image point par point et imprime les points voulus. C'est très défavorable à la MCP40 qui fait plutôt des lignes: il vaut mieux faire un tracé prévu au départ pour la MCP40.

2) Il faut définir l'arc et faire les tracés:
FOR I= a TO b: X=XC+R*SIN(I):
Y=YC+YC+R*COS(I): DRAW X,Y,..:NEXT

3) La "pratique de l'ORIC" est prévu pour bientôt.

4) La cassette Télécran est disponible, au délai de traitement de la commande près.

*
* *

Questions sur l'ORIC

Je suis possesseur d'un ORIC V1.0 depuis quelques mois et il se trouve que je viens de faire connaissance des subtilités (ou problèmes) de la bêbête.

1) Il existe plusieurs ordinateurs dont le microprocesseur est un 6502, mais les possibilités sont différentes de l'un à l'autre. Par exemple sur certains il existe des "sprites", mais pas sur l'ORIC, pourquoi ? Pourquoi y-a-t-il des différences de possibilités ?

2) Lorsque l'on désassemble la ROM V1.0 on trouve des fonctions basic telles que INVERSE, mais lorsque l'on travaille en basic, il ne connait pas cette fonction: que s'est-il passé ?

3) Comment sont répartis les différents octets (48+16) ?

4) Lorsque nous sommes en Haute Résolution pour dessiner un objet, il faut obligatoirement le cerner d'attributs de couleur papier pour que les objets passant à proximité ne prennent pas sa couleur. De plus si deux de ces objets voyagent ensemble l'attribut de l'un masque la moitié de l'autre.

5) Il n'est pas possible de mélanger plusieurs couleurs, car nous sommes tributaires d'un affichage en matrice 6x6 pixels. Certains disent qu'il serait possible d'afficher pixel par pixel. Un livre devrait sortir la-dessus.

6) Peut-on remédier à la lenteur de calcul de l'ORIC ?

7) Quelles sont les différences valables entre ROM V1.0 et ROM Atmos?

8) Je possède la cassette ORIC base mais je ne comprends rien aux termes tels que label, champs ..etc... et de ce fait je suis bien incapable de créer un fichier.

9) Existe-t-il un compilateur ?

10) A quoi correspondent les deux réglages situés sous le boîtier ? Que se passerait-il s'ils étaient touchés ?

11) Existe-t-il des lecteurs de disquettes 5 pouces car on nous présente actuellement des lecteurs 3 pouces; la disquette 3 pouces est plus chère que la 5 pouces. D'autre part qu'est-l advenu de la maison qui fabriquait le lecteur CYBORG ? Depuis qu'on a appris que cette maison devait commercialiser un lecteur adaptable à tous les micros, on n'a plus entendu parler d'elle.

12) Peut-on réaliser de la synthèse d'image, ou digitaliser celle-ci, avec un magnétoscope, une caméra vidéo et un ordinateur, sinon que peut-on faire avec tout cela ? Comment branche-t-on un ordinateur sur un magnétoscope sachant que la sortie de l'ordinateur en UHF est en PAL.

Patrick DELIAS
45600 Sully sur Loire

1) Les "sprites" ne sont pas créés par le 6502 mais par le processeur graphique. Celui de l'ORIC n'en a pas. On peut cependant créer des effets analogues avec les caractères créés par l'utilisateur, voir "La Découverte de l'ORIC" (PSI) p.112.

2) Les fonctions INVERSE et NORMAL n'ont en fait pas été mises.

3) Voir La Commode n° 9 p.47.

4) Ce que vous dites est vrai et résulte de la gestion des attributs.

5) On peut changer de couleur pixel par pixel le long d'une verticale. Pour une horizontale, c'est par blocs de 6 pixels. Voir "La Découverte de l'Oric" p.129.

6) On subit les inconvénients d'un interpréteur Basic encore que celui de l'Oric soit parmi les plus rapides. Dans La Commode n°1, il y a un article sur "accélérer vos programmes Basic".

7) Voir article "le point sur l'ATMOS" dans notre n°10.

8) Ces questions seront traitées dans le livre "La pratique de l'ORIC" de D.J. DAVID aux éd. du PSI.

9) Pas encore à notre connaissance.

10) Ce sont des règlages fins pour la télé: l'image disparait s'ils sont déréglés.

11) CYBORG a toujours ses disques en préparation mais leur délai leur ôte beaucoup de crédibilité.

12) Le plus simple est d'avoir un magnétoscope PAL et de lui envoyer le signal d'antenne de l'ORIC. Sinon, il faut prendre les sorties RVB de l'ORIC et les envoyer sur un codeur SECAM, ou alors avoir un magnétoscope qui ait les entrées RVB (peut être que l'entrée caméra le permet).

* *
*

Ronflement de l'Atari

Je viens d'acquérir un ATARI 800XL et ne parviens pas à empêcher un ronflement sur mon téléviseur portable couleur TELEAVIA. Le service après-vente du commerçant (SERAP-Levallois) a pris contact avec le fabricant : sans solution pour l'instant...

André BLAD
93200 SAINT DENIS

La connexion des ATARI par la prise d'antenne semble poser des problèmes. On les évite probablement en se branchant par la prise Péritel. Voir le schéma dans ce numéro.

* *
*

Les joies du langage machine

1) Dans le n°8 p.28: les ficelles de PEThalberg, comment arrêter et reprendre le listing d'un programme. J'ai essayé sur 4016 en changeant INTNOR et VECT et ça se plante. Pourriez-vous expliquer plus en détail ce programme, en particulier ce qui se passe en E810,E811 ?

2) Sur ORIC, pour déplacer un dessin j'ai besoin de deux octets consécutifs en page zéro pour l'adressage indirect. J'ai choisi "au pif" $50 et $51: plantage ! à quoi correspondent ces adresses ?
Y-a-t-il des octets de disponibles en page zéro, quitte à les sauvegarder ?

Help ! help ! Qui peut m'aider ?

Hubert RINALDI
KENITRA (MAROC)

1) Sur 4000, c'est la rangée 6 et non 8 qui intervient. D'autre part c'est INC $E812 d'où la fin du programme pour 4000:

```
0347  A9 06 8D 10 E8 EE 12 E8
034F  D0 F6 4C 55 E4 ....
```

2) Les adresses $50, 51 sont utilisées par l'ORIC. En revanche (voir l'article dans notre n° 10), les 11 premiers octets ($0 à $0B) sont utilisables.

BON DE COMMANDE LOGICIELS

A envoyer accompagné du règlement à :
SEDERMI, 28, rue Vicq d'Azir, 75010 PARIS.

NOM : .....
Adresse : .....
.....

Je désire les logiciels suivants :

nom (spécifier si compilé)	machine (configuration)	support (préciser le type)	prix
----------	----------	----------	-------
. . . . .	. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .	. . . . .
			----------
		total:	. . . . .

signature:

La pendule magique

La plupart d'entre vous connaissent l'horloge interne des ordinateurs COMMODORE accessible depuis le BASIC sur la variable TI$. Sa mise à l'heure est aisée et elle se met automatiquement sous le format heure, minute, seconde. Cependant, elle souffre d'un gros défaut : elle ne supporte pas les échanges avec les périphériques comme les cassettes ou des disquettes. Lors d'un chargement ou d'une sauvegarde, sa valeur n'est plus mise à jour par le système, plus précisément par la routine d'interruption qui l'incrémente régulièrement d'un soixantième de seconde. Pour pallier cet inconvénient, les concepteurs du C 64 ont introduit une seconde horloge indépendante du BASIC et synchronisée sur les cycles du secteur. Elle n'est plus accessible depuis le BASIC mais son format DCB (décimal codé binaire) facilite grandement sa traduction. Cette pendule occupe quatre adresses mémoires consécutives débutant en 56328 et représentant respectivement les dixièmes de seconde, secondes, minutes et heures. Nous allons décrire ces registres et ceux qui en dépendent :

Adresse 56331 ($DC0B) : le bit 4 représente la dizaine de l'heure 0 ou 1, les bits 0-3 le chiffre des unités entre 0 et 9. Le bit 7 passe à 1 lorsque l'heure franchit 12:00 (après-midi), elle repart alors à 00:00.

Adresse 56330 ($DC0A) : pour les minutes, les bits 4-6 représentent les dizaines et les bits 0-3 les unités. Le bit 7 est toujours nul.

Adresse 56329 ($DC09) : identique à 56330 mais pour les secondes.

Adresse 56328 ($DC08) : les bits 0-3 comptent les dizièmes; les bits 4-7 sont nuls.

Il y a aussi d'autres adresses concernées. Tout d'abord comme la pendule est synchronisée sur la fréquence du secteur et que celui-ci oscille à 50 Hertz contre 60 aux ETATS-UNIS, il s'ensuit qu'elle devrait retarder (vérifiez le, c'est une bonne gymnastique de l'esprit). Heureusement, le cas a été prévu et le bit 7 de l'adresse 56334 doit être mis à 1 pour le 50 Hz et à 0 pour le 60 Hz.

La pendule possède aussi une alarme ! Pour la modifier, la procédure est identique au réglage de la pendule, mais le bit 7 de l'adresse 56335 doit au préalable être mis à 1. On ne peut lire sa valeur et elle génère une interruption lorsque qu'elle coïncide avec la valeur de la pendule.

Le programme ci-contre vous permettra de régler l'heure et de la voir affichée en permanence en haut à gauche de l'écran. On l'élimine par RUN/RESTORE et on la rappelle par SYS 49152. Ce programme comporte deux parties : l'une en langage machine à rentrer par un moniteur et l'autre en BASIC pour faciliter son réglage. 'F3/F4' changent l'heure, 'F5/F6' les minutes, 'F7/F8' les secondes et 'F1' la fait repartir. En effet, toute modification de l'heure arrête la pendule et seule une écriture dans le registre des dixièmes la fait repartir.

Pour terminer, qu'en est-il de la précision ? Eh bien, c'est assez décevant car elle retarde de plusieurs minutes par heure. Les américains ont peut-être plus de chan-

ce? Sans doute ce défaut a-t-il été corrigé sur les derniers modèles, ceux de la troisième génération, le mien appartenant à la seconde ?

Hervé LE MARCHAND

```
  5 ! CODE SOURCE PAR L'ASSEMBLEUR MIKRO
 10 *=$C000
 40 ECRAN          =   $0400
 50 COULEUR        =   $D800
 60 DIXIEME        =   $DC08
 70 SECONDE        =   $DC09
 80 MINUTE         =   $DC0A
 90 HEURE          =   $DC0B
100 INITOD         LDA $0314       ! SAUVE L'ANCIEN VECTEUR
110                STA INTER
120                LDA $0315
130                STA INTER+1
140                SEI
150                LDA #<WEDGETOD  ! ET LE REMPLACE PAR LE NOUVEAU
160                STA $0314
170                LDA #>WEDGETOD
180                STA $0315
190                CLI
200                RTS
210 INTER          BYT 0,0
220 WEDGETOD       LDX #0          ! ON RENTRE ICI
230                LDA HEURE
240                AND #$7F        ! ELIMINE AM/PM
250                JSR FORMAT
260                LDA #':+$80
270                JSR METECRAN
280                LDA MINUTE
290                JSR FORMAT
300                LDA #':+$80
310                JSR METECRAN
320                LDA SECONDE
330                JSR FORMAT
340                LDA #':+$80
350                JSR METECRAN
360                LDA DIXIEME
370                ORA #'0+$80
380                JSR METECRAN
390                JMP (INTER)     ! ET ON SORT LA
400 FORMAT         TAY
410                LSR A
420                LSR A
430                LSR A
440                LSR A           ! CHIFFRE DES DIZAINES DANS A
450                JSR METECR      ! QUE L'ON VA AFFICHER
460                TYA
470                AND #$0F        ! AVEC LE CHIFFRE DES UNITES
480 METECR         ORA #'0+$80     ! EN CONTRASTE INVERSE
490 METECRAN       STA ECRAN,X     ! EN HAUT A GAUCHE DECALE DE X
500                LDA 646
510                STA COULEUR,X   ! AVEC LA COULEUR D'ECRITURE
520                INX
530                RTS             ! ET ON REVIENT
```

```
C000 AD 14 03 LDA $0314     C020 29 7F    AND #$7F      C048 6C 19 C0 JMP ($C019)
C003 8D 19 C0 STA $C019     C022 20 4B C0 JSR $C04B     C04B A8       TAY
C006 AD 15 03 LDA $0315     C025 A9 BA    LDA #$BA      C04C 4A       LSR
C009 8D 1A C0 STA $C01A     C027 20 58 C0 JSR $C058     C04D 4A       LSR
C00C 78       SEI           C02A AD 0A DC LDA $DC0A     C04E 4A       LSR
C00D A9 1B    LDA #$1B      C02D 20 4B C0 JSR $C04B     C04F 4A       LSR
C00F 8D 14 03 STA $0314     C030 A9 BA    LDA #$BA      C050 20 56 C0 JSR $C056
C012 A9 C0    LDA #$C0      C032 20 58 C0 JSR $C058     C053 98       TYA
C014 8D 15 03 STA $0315     C035 AD 09 DC LDA $DC09     C054 29 0F    AND #$0F
C017 58       CLI           C038 20 4B C0 JSR $C04B     C056 09 B0    ORA #$B0
C018 60       RTS           C03B A9 BA    LDA #$BA      C058 9D 00 04 STA $0400,X
C019 00       BRK           C03D 20 58 C0 JSR $C058     C05B AD 86 02 LDA $0286
C01A 00       BRK           C040 AD 08 DC LDA $DC08     C05E 9D 00 D8 STA $D800,X
C01B A2 00    LDX #$00      C043 09 B0    ORA #$B0      C061 E8       INX
C01D AD 0B DC LDA $DC0B     C045 20 58 C0 JSR $C058     C062 60       RTS
```

```
C000 AD 14 03 8D 19 C0 AD 15
C008 03 8D 1A C0 78 A9 1B 8D
C010 14 03 A9 C0 8D 15 03 58
C018 60 00 00 A2 00 AD 0B DC
C020 29 7F 20 4B C0 A9 BA 20
C028 58 C0 AD 0A DC 20 4B C0
C030 A9 BA 20 58 C0 AD 09 DC
C038 20 4B C0 A9 BA 20 58 C0
C040 AD 08 DC 09 B0 20 58 C0
C048 6C 19 C0 A8 4A 4A 4A 4A
C050 20 56 C0 98 29 0F 09 B0
C058 9D 00 04 AD 86 02 9D 00
C060 D8 E8 60 A9 31 8D 14 03
```

```
10 REM *** PENDULE C-64 HERVE LE MARCHAND ***
20 PRINT"□■■■■ REGLER LA PENDULE AVEC LES TOUCHES"
30 PRINT"■  PROGRAMMABLES 'F1' JUSQU'A 'F8'"
50 SYS 49152:REM INITALISE LE WEDGE
60 POKE 56334,129:REM SECTEUR A 50 HERZ
70 POKE 56335,PEEK(56335) AND 127 : REM REGLAGE DE LA PENDULE ( # ALARME )
80 HE=56331:MN=HE-1:SE=HE-2:DI=HE-3:REM HEURE, MINUTE, SECONDE
90 POKE646,1:POKE650,128
95 :
100 GET R$:IF R$="" GOTO 100
110 R=ASC(R$):IF R<133 OR R>140 GOTO 100
120 IF R=133 THEN POKE 56328,0:POKE 646,1:END:REM FAIT DEMARRER LA PENDULE
130 IF R=137 THEN POKE HE,0:POKE MN,0:POKE SE,0:GOTO 100:REM INITIALISE
140 X=(R=138)-(R=134):IF X=0 GOTO 160
150 POKE 646,14:Y=PEEK(HE) AND 31:GOSUB300:IF Y>-1 AND Y<19 THEN POKE HE,Y
160 X=(R=139)-(R=135):IF X=0 GOTO 180
170 POKE 646,10:Y=PEEK(MN):GOSUB300:IF Y>-1 AND Y<90 THEN POKE MN,Y
180 X=(R=140)-(R=136):IF X=0 GOTO 200
190 POKE 646,13:Y=PEEK(SE):GOSUB300:IF Y>-1 AND Y<90 THEN POKE SE,Y
200 GOTO 100
210 :
300 REM CONVERSION DES DONNEES
310 L=Y AND 15:H=INT(Y/16):L=L+X:IF L=10 THEN L=0:H=H+1
320 IF L<0 THEN L=9:H=H-1
330 Y=16*H+L:RETURN
```

QUIZ

Voici un quiz destiné à vérifier votre connaissance de votre COMMODORE mais la plupart des questions (marquées *) s'appliquent à l'ORIC et à l'ATARI aussi. Réponses au prochain numéro.

1(*). Un programme se résume à

```
10  X = X + 1 : GOSUB 10
```

Au bout de très peu de temps, ce programme s'arrête et affiche un message d'erreur: OUT OF MEMORY alors qu'il reste plus de 31 K de libres.

. Pourquoi a-t-on ce message?

. Quelle est alors la valeur de X ?

. Quel renseignement apporte la valeur de X à l'arrêt ?

2(*). On modifie ce programme en remplaçant le GOSUB par un GOTO. Cette fois le programme ne s'arrêtera que lorsque X aura atteint la valeur limite acceptable pour une variable.

Dans combien de temps le programme cessera-t-il de tourner pour annoncer "OVERFLOW ERROR IN 10"

3(*). On modifie encore un peu le programme :

```
10 X = 2 * X : GOTO 10
```

Dans combien de temps le programme cessera-t-il de tourner pour annoncer que X a atteint la valeur-limite ?

4(*). On ajoute au programme précédent la ligne :

```
5 X = 1
```

Au bout de combien de temps le programme cessera-t-il de tourner ?

Question subsidiaire : comment peut-on mesurer cette durée en limitant le mieux possible les erreurs de manipulation ?

5(*). Un programme se limite à

```
10 DATA A, B, C, D : GOTO 10
```

Que va-t-il se passer si l'on fait RUN ?

6(*). Même question en remplaçant le mot DATA par le mot REM.

7(*). On a le programme suivant :

```
10 DATA 5 : DATA 16
20 READ X : READ Y
```

Pourra-t-on lire Y ou bien aura-t-on le message OUT OF DATA?

8(*). On a le programme suivant :

```
10 FOR I = 1 TO 10
20 DATA 5
30 READ A% (I)
40 NEXT
```

Aura-t-on ou non le message "OUT OF DATA" ?

9(*). Dans un programme, on a :

```
10 IF A$ THEN PRINT "OUI"
```

Si l'on fait RUN, verra-t-on s'écrire "OUI" ?

Même question si le programme commence par 5 A$ = ""

10. Est-il possible de faire un INPUT sans point d'interrogation ? (Un vrai INPUT et non une imitation fabriquée avec une série de GET)

11(*). On a la ligne suivante :

```
10 GOTO 100 : REM ON SAUTE A LA FIN DU PROGRAMME.
```

Aura-t-on des ennuis si on supprime le mot REM ?

12(*). Quel est l'intérêt d'écrire

```
DIM A$ (3)
```

13 . Quel message d'erreur aura-t-on si l'on écrit (Vrai aussi sur ORIC)

```
10 A$(1) = "TOTO"
20 DIM A$(10)
```

14 . Que manque-t-il au programme suivant pour qu'il marche (Vrai aussi sur ORIC) :

```
10 RACINE(X) = SQR(X)
20 X = 25
30 PRINT RACINE(X)
```

15 . Vous savez qu'en temps normal PEEK (52) + 256 * PEEK (53) = 32768 et vous avez peut-être quelquefois modifié le contenu des cases 52 et 53 pour abaisser la frontière entre la zone de travail Basic et l'écran. Pensez-vous que si pour une fois vous augmentiez le contenu de la case 53, vous pourriez voler quelques octets à la mémoire d'écran et en faisant PRINT FRE(0) obtenir enfin un nombre supérieur à 31743 ?

René BASSABER

Pour avoir sur imprimante le catalogue de vos disquettes ORIC

Rien dans le système Oric ne semble permettre cette chose pourtant essentielle: garder une copie durable du catalogue de vos disquettes.

Le petit programme ci-dessous donne la solution. L'ordre !DIR peut se mettre dans un programme et, lorsqu'il est fini, on passe à la suite du programme. Donc si cette suite est une copie écran-imprimante, vous aurez votre catalogue sur imprimante...

... ou plutôt la dernière page car le catalogue est en plusieurs pages. Tant qu'on n'est pas en dernière page on vous dit "press RETURN to continue". L'astuce est que si, lorsque vous avez une page intermédiaire, au lieu de taper RETURN vous tapez sur la touche ESC, le programme sort du !DIR et fait la suite qui n'est autre que la copie d'écran.

Le problème est qu'il va falloir faire ça pour chaque page. Faites RUN. Il s'affiche la 1ère page. Faites ESC. La 1ère page se copie. Refaites RUN. Il s'affiche la 1ère page. Faites RETURN. Il s'affiche la 2ème page. Faites ESC. La 2ème page se copie. Et caetera.. et caetera... mais il est rare qu'il y ait plus de 2 pages sur le catalogue.

L'instruction 20 fait remonter le curseur, sinon les lignes du haut de l'écran disparaîtraient.

```
10 !DIR O
20 PRINTCHR$(11);CHR$(11);
30 FORY=0TO26
40 FORX=1TO38:C=SCRN(X,Y)
50 LPRINTCHR$(C);:NEXT:LPRINT
60 NEXT
```

Honoric de BALSA

Architecture de l'Atari

Nous commençons par les quelques notions sur l'architecture de l'Atari qui sont indispensables pour comprendre comment fonctionne l'affichage de cette machine. Il faut noter que cette architecture est la même pour les anciens modèles 400 et 800 et pour les nouveaux 600XL et 800XL.

Comme pour les VIC, 64, ORIC et d'autres, l'intelligence du microprocesseur 6502 est secondée par celle d'un second processeur, le processeur graphique. Mais, alors que dans le VIC il y a un seul boîtier secondaire (le 6561), que dans le 64 ou l'ORIC il y a un processeur secondaire (le 6567, ou l'ULA) et un synthétiseur sonore (le 6581 ou le GI AY3-8192), il y a dans l'Atari trois boîtiers annexes: ANTIC, le GTIA/CTIA et "POKEY".

* POKEY s'occupe principalement du son, mais il comporte en plus des ports d'entrée-sortie. Atari n'a publié aucune explication des noms de ses boîtiers-maison. Pour POKEY, c'est simplement qu'on écrit souvent dans ses registres internes par des POKE.

* GTIA est un boîtier d'interface télé. Nous pensons que son nom veut dire "adaptateur d'interface télévision général" (General Television Interface Adaptor), mais c'est une conjecture. Les modèles les plus anciens étaient équipés de CTIA (Color Television Interface...) aux possibilités un tout petit peu plus restreintes (quelques modes en moins), mais, à notre avis, tous les Atari 400 et 800 non XL livrés en France avaient déjà le GTIA. Tous les XL ont le GTIA.

* ANTIC est aussi consacré à la télé. Nous pensons - et c'est encore une conjecture, et de toutes façons, c'est sans importance pratique - que son nom veut dire "nouveau boîtier Atari d'interface télé" (Atari New Television Interface Chip). Pendant qu'on en est dans les noms, savez vous qu'Atari est un cri de guerre japonais qui a donné naissance à un jeu, lequel a été un des premiers jeux à être informatisé. On aura bientôt les ordinateurs Banzaï.

La formation d'images est donc répartie entre ANTIC et GTIA. Le GTIA hérite des tâches subalternes de construction des signaux analogiques, de synchro, de couleur, etc... tandis qu'ANTIC se charge des tâches nobles d'organisation de l'image.

L'encadré à la fin de cet article donne la liste des registres des boîtiers d'entrée-sortie. L'adresse entre parenthèses après certains registres est ce qu'on appelle l'adresse du registre fantôme associé. C'est en fait une adresse mémoire. Le contenu de certains registres est en effet très transitoire: à des instants précis le système recopie dans le registre hardware le contenu du registre fantôme associé et on a plutôt intérêt, dans les programmes, à écrire dans les registres fantômes plutôt que dans les vrais registres. Pour chaque registre on a indiqué le nom symbolique qui lui est habituellement donné dans la littérature. Un second encadré donne la carte d'implantation mémoire des XL montrant comment, par recouvrements, on arrive à dépasser la limite des 64 K.

D-J DAVID

Les registres d'E/S de l'Atari

L = lecture E = écriture

GTIA/CTIA

D000	53248	E	HPOSP0	position horizontale joueur 0
		L	M0PF	collision missile 0 - écran
D001	53249	E	HPOSP1	position horizontale joueur 1
		L	M1PF	collision missile 1 - écran
D002	53250	E	HPOSP2	position horizontale joueur 2
		L	M2PF	collision missile 2 - écran
D003	53251	E	HPOSP3	position horizontale joueur 3
		L	M3PF	collision missile 3 - écran
D004	53252	E	HPOSM0	position horizontale missile 0
		L	P0PF	collision joueur 0 - écran
D005	53253	E	HPOSM1	position horizontale missile 1
		L	P1PF	collision joueur 1 - écran
D006	53254	E	HPOSM2	position horizontale missile 2
		L	P2PF	collision joueur 2 - écran
D007	53255	E	HPOSM3	position horizontale missile 3
		L	P3PF	collision joueur 3 - écran
D008	53256	E	SIZEP0	taille joueur 0
		L	M0PL	collision missile 0 - joueur
D009	53257	E	SIZEP1	taille joueur 1
		L	M1PL	collision missile 1 - joueur
D00A	53258	E	SIZEP2	taille joueur 2
		L	M2PL	collision missile 2 - joueur
D00B	53259	E	SIZEP3	taille joueur 3
		L	M3PL	collision missile 3 - joueur
D00C	53260	E	SIZEM	taille des missiles
		L	P0PL	coll. joueur 0 - joueur
D00D	53261	E	GRAFP0	forme joueur 0
		L	P1PL	coll. joueur 1 - joueur
D00E	53262	E	GRAFP1	forme joueur 1
		L	P2PL	coll. joueur 2 - joueur
D00F	53263	E	GRAFP2	forme joueur 2
		L	P3PL	coll. joueur 3 - joueur
D010	53264	E	GRAFP3	forme joueur 3
		L	TRIG0	gachette joystick 0 ($284,644)
D011	53265	E	GRAFM	forme des missiles
		L	TRIG1	gachette joystick 1 ($285,645)
D012	53266	E	COLM0	couleur joueur et missile 0 ($2C0,704)
		L	TRIG2	gachette joystick 2 ($286,646)
D013	53267	E	COLM1	couleur joueur et missile 1 ($2C1,705)
		L	TRIG3	gachette joystick 3 ($287,647)
D014	53268	E	COLM2	couleur joueur et missile 2 ($2C2,706)
		L	PAL	0 si PAL, 14 si NTSC
D015	53269		COLM3	couleur joueur et missile 3 ($2C3,707)
D016	53270		COLPF0	couleur écran 0 ($2C4,708)
D017	53271		COLPF1	couleur écran 1 ($2C5,709)
D018	53272		COLPF2	couleur écran 2 ($2C6,710)
D019	53273		COLPF3	couleur écran 3 ($2C7,711)
D01A	53274		COLBK	couleur fond ($2C8,712)
D01B	53275	E	PRIOR	priorités sur l'écran ($26F,623)
D01C	53276	E	VDELAY	délai vertical
D01D	53277	E	GRACTL	contrôle objets
D01E	53278	E	HITCLR	vide registres collisions
D01F	53279		CONSOL	image des touches OPTION, SELECT et START (pas HELP).

POKEY

D200	53760	E	AUDF1	frequence canal sonore 1	
		L	POT0	potentiomètre (raquette) 0	($270,624)
D201	53761	E	AUDC1	contrôle canal 1	
		L	POT1	pot. 1	($271,625)
D202	53762	E	AUDF2	fréquence canal 2	
		L	POT2	pot. 2	($272,626)
D203	53763	E	AUDC2	contrôle canal 2	
		L	POT3	pot. 3	($273,627)
D204	53764	E	AUDF3	fréquence canal 3	
		L	POT4	pot. 4 (pas sur XL)	($274,628)
D205	53765	E	AUDC3	contrôle canal 3	
		L	POT5	pot. 5 (pas sur XL)	($275,629)
D206	53766	E	AUDF4	fréquence canal 4	
		L	POT6	pot. 6 (pas sur XL)	($276,630)
D207	53767	E	AUDC4	contrôle canal 4	
		L	POT7	pot. 7 (pas sur XL)	($277,631)
D208	53768	E	AUDCTL	audio contrôle	
		L	ALLPOT	validité potentiomètres	
D209	53769	E	STIMER	démarrage temporisateurs	
		L	KBCODE	code clavier	($2FC,764)
D20A	53770	E	SKREST	reset du registre d'état série	
		L	RANDOM	nombre aléatoire	
D20B	53771	E	POTGO	démarrage séquence lecture des potentiomètres	
D20C	53772	inutilisé			
D20D	53773	E	SEROUT	sortie sur port série	
		L	SERIN	entrée sur port série	
D20E	53774	E	IRQEN	activation des interruptions	
		L	IRQST	état des demandes d'interruptions	
D20F	53775	E	SKCTL	contrôle port série	($232,562)
		L	SKSTAT	état port série	

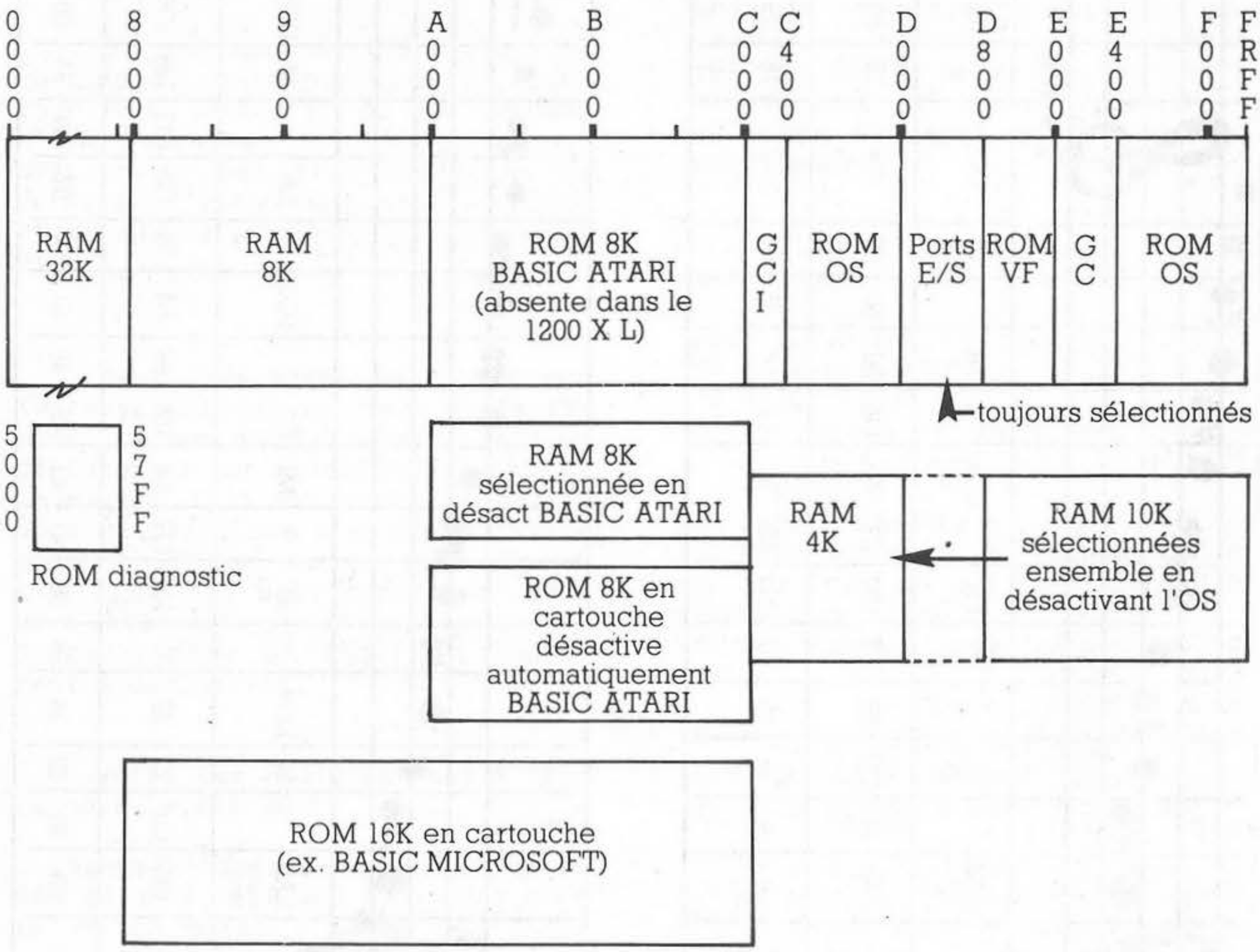
ANTIC

D400	54272	E	DMACTL	commande DMA	($22F,559)
D401	54273	E	CHACTL	mode caractère	($2F3,755)
D402-3	54274-5		DLISTL/H	pointeur (bas/haut) vers la display list	($230-231 ,560-561)
D404	54276	E	HSCROL	autorise déroulement horizontal	
D405	54277	E	VSCROL	autorise deroulement vertical	
D406	54278	inutilisé			
D407	54279	E	PMBASE	octet haut de l'adresse de base des joueurs et missiles	
D408	54280	inutilisé			
D409	54281	E	CHBASE	départ du générateur de caractères (octet haut)	($2F4,756)
D40A	54282	E	WSYNC	attente synchro V	
D40B	54283	L	VCOUNT	n° de ligne de balayage / 2 va de 0 à 155 en PAL	
D40C	54284	L	PENH	pos. horizontale du light pen	($234,564)
D40D	54285	L	PENV	pos. verticale du light pen	($234,564)
D40E	54286	E	NMIEN	autorise les interruptions NMI	
D40F	54287	E	NMIRES	RAZ des NMI	
		L	NMIST	état des NMI	

PIA

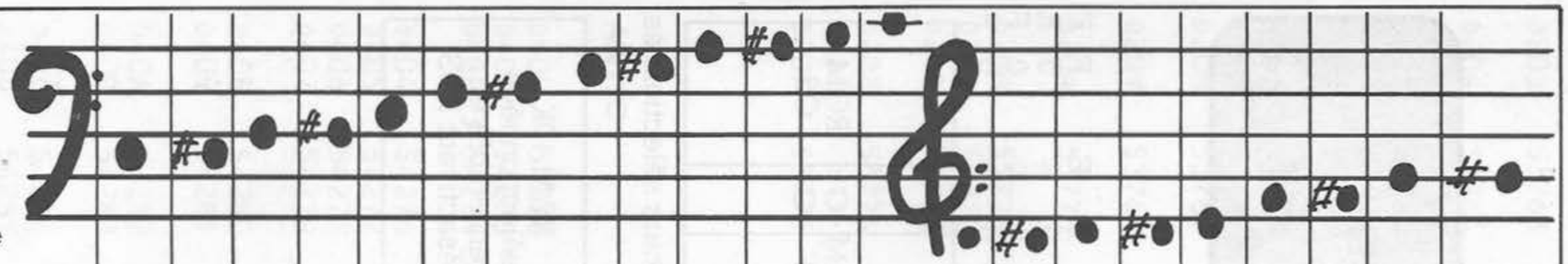
D300	54016	PORTA	joysticks 0-1	($278-279,632-633)
			paddles 0-3	($27C-27F,636-639)
D301	54017	PORTB	joysticks 2-3 (pas sur XL)	($27A-27B,634-635)
			paddles 4-7	($280-283,640-643)
D302	54018	PACTL	contrôle port A, moteur cassette : POKE 54018,52 démarre ; POKE 54018,60 arrête	
D303	54019	PBCTL	contrôle port B	

Recouvrements d'adresses dans les ATARI XL

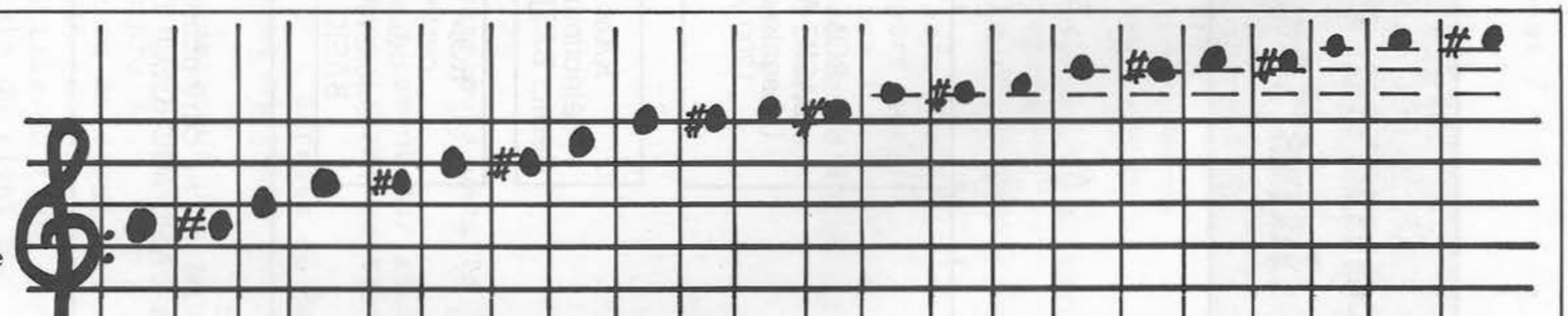


GG : générateur de caractères
GCI : générateur de caractères internationaux
VF : virgule flottante

Tableau des notes sur 64



Note																					
Nom	DO2 C		RE D		MI E	FA F		SOL G		LA A		SI B	DO3		RE		MI	FA		SOL	
Freq basse	140	14	152	42	197	105	23	208	146	96	60	35	24	36	50	86	140	211	47	160	37
Freq haute	8	9	9	10	10	11	12	12	13	14	15	16	17	18	19	20	21	22	24	25	27



Note																					
Nom	LA3 diapason		SI	DO4		RE		MI	FA		SOL		LA		SI	DO5		RE		MI	FA
Freq basse	195	121	72	52	61	100	173	24	168	96	64	76	135	243	146	106	123	202	91	50	83	193
Freq haute	28	30	32	34	36	38	40	43	45	48	51	54	57	60	64	68	72	76	81	86	91	96

Défilement latéral

Le défilement vertical et vers le haut est indispensable à tout éditeur d'écran. Il n'en est pas de même des autres types de défilement. Voici 2 programmes en assembleur permettant d'obtenir le défilement à gauche et le défilement à droite.

En réponse à l"article "17 questions" de la rubrique "Le courrier des lecteurs" de LA COMMODE n° 8, question n° 13, on répondait qu'il est très difficile de déplacer l'écran de gauche à droite et de droite à gauche :il n'en est rien !

Deux programmes en assembleur (vitesse oblige) tournent parfaitement sur mon 64. Le premier fait un deplacement en scrolling gauche et le second fait un scrolling droit. Vous me direz que c'est plus facile sur le 64 que sur le VIC 20; eh bien non, les deux programmes peuvent tourner sur n'importe quelle "bécane" 6502 en modifiant seulement 4 paramètres.

Voici les modifications à faire pour le VIC 20 :

. Changer l'adresse de début de la RAM écran : de $0400 (en 1024) pour le 64 on passe à $1E00 (ou 7680) pour le VIC 20

. Changer l'adresse de début de la RAM couleur : de $D800 (ou 55296) pour le 64 on passe à $9600 (ou 38400) pour le VIC 20

. Changer la taille des lignes : de $27 et $28 (40 colonnes) pour le 64, on passe à $15 et $16 (22 colonnes) pour le VIC 20

. Changer le nombre de lignes: de $19 (25 lignes) pour le 64 on passe à $17 (23 lignes) pour le VIC 20.

Ces modifications apparaîssent en noir et entre parenthèses dans la marges réservée aux commentaires; il suffit de mettre le chiffre hexadécimal qui est dans la marge à la place de celui qui est dans le programme.

L'algorithme de ces programmes est simple.

Pour les scrolling gauche :
1) initialisation
2) début ligne
3) on prend le caractère suivant que l'on met là où on est (en ($FB) , Y)
4) pareil pour la couleur
5) incrémentation n° colonne
6) fin ligne ? oui ->7 non ->3
7) saut de ligne
8) fin écran ? oui ->Fin non ->2

Ce programme est stocké à partir de 49152 ($C000). Sur le VIC, il sera stocké à partir de 828.

La possibilité de changer les 4 paramètres permet non seulement de le faire marcher sur un bon nombre de 6502 mais aussi de faire des scrollings partiels en définissant donc des fenêtres.

Philippe VAGNER

SCROLLING GAUCHE :

```
        LDA  #$04        ; initialisation début mémoire   (1E)
        STA   $FC        ; couleur $0800 ($FD et $FE)
        LDA  #$00        ; écran $0400 ($FB et $FC)
        STA   $FB        ;
        STA   $FD        ;
        LDA  #$08        ;                                (96)
        STA   $FE        ;
        LDX  #$00        ;
BCLLGH  LDY  #$00        ; boucle ligne
BCLCOL  INY              ; boucle colonne
        LDA  ($FB),Y     ; scrolling caractère
        DEY              ;
        STA  ($FB),Y     ;
        INY              ; scrolling couleur
        LDA  ($FD),Y     ;
        DEY              ;
        STA  ($FD),Y     ;
        INY              ;
        CPY  #$27        ; fin de ligne?                  (15)
        BNE  BCLCOL      ; si non, on va en boucle colonne
        LDA  #$20        ; dernier caractère à blanc
        STA  ($FB),Y     ; ($20 = 32)
        CLC              ;
        LDA   $FB        ; saut de ligne ($28 = 40)
        ADC  #$28        ; pour caractères                (16)
        STA   $FB        ;
        LDA  #$00        ;
        ROL   A          ;
        ADC   $FC        ;
        STA   $FC        ;
        CLL              ;
        LDA   $FD        ; saut de ligne pour couleur
        ADC  #$28        ;                                (16)
        STA   $FD        ;
        LDA  #$00        ;
        ROL   A          ;
        ADC   $FE        ;
        STA   $FE        ;
        INX              ;
        CPX  #$19        ; Fin d'écran                    (17)
        BNE  BCLLGH      ; si non on va en boucle ligne
        RTS              ; fin
```

SCROLLING DROIT :

```
        LDA  #$04        ; mêmes commentaires que pour le (1E)
        STA   $FC        ; programme de scrolling gauche
        LDA  #$00
        STA   $FB
        STA   $FD
        LDA  #$08                                        (96)
        STA   $FE
        LOX  #$00
BCLLGN  LDY  #$27                                        (15)
BCLCOL  DEY
        LDA  ($FB),Y
        INY
```

```
STA  ($FB),Y
DEY
LDA  ($FD),Y
INY
STA  ($FD),Y
DEY
BNE  BCLCOL
LDA #$20
STA  ($FB),Y
CLC
LDA  $FB
ADL  #$28                    (16)
STA  $FB
LDA  #$00
ROL  A
ADL  $FD
STA  $FC
CLC
LDA  $FD
ADL  #$28                    (16)
STA  $FD
LDA  #$00
ROL  A
ADC  #$FE
STA  #$FE
INX
CPX  #$19                    (17)
BNE  BCLLGH
RTS
```

Vic à brac

Compacter plusieurs lignes

Il est toujours ennuyeux de devoir retaper les programmes à compacter. Voici une petite astuce qui peut vous faire gagner du temps et du calme:

```
ex:  10 PRINT A$
     20 A=A+2
     30 GOTO 10
```

Il est fatiguant de réécrire sur la ligne 10 les lignes 20 et 30 (surtout si celles-ci sont longues ou compliquées, cas de POKE).

Il suffit de rajouter des espaces en 20 et en 30:

```
10 PRINT A$
20 :        :A=A+2
30 :                :GOTO10
```

Pour que les espaces soient acceptés, mettre un ":" au début puis les espaces et un autre ":" pour le compactage.

Faire alors: LIST 30
ce qui donne:

```
30 :                :GOTO10
READY
```

remonter sur le LIST et mettre LIST20 qui donne:

```
20 :        :A=A+2:GOTO10
```

remonter de nouveau sur le LIST et mettre LIST10 qui donne:

```
10 PRINT A$:A=A+2:GOTO10
```

L'astuce est simplement de mettre le nombre de blancs nécessaires pour que 2 lignes ne se chevauchent pas... et surtout de bien remonter sur le LIST écrit à l'écran, pour que la ligne antérieure s'imprime juste devant celle déjà à l'écran.

Il vous reste bien sur à supprimer les lignes devenues inutiles.

BASIC EN 7168 SUR VIC 16K

```
10 POKE44,28:POKE46,28:POKE48,28
20 POKE43,1:POKE45,3:POKE47,3:POKE49,3
30 POKE28*256,0:POKE29*256,0:POKE30*256,0
40 POKE198,7:POKE631,76:POKE632,207:POKE633,13
50 POKE634,82:POKE635,213:POKE636,13:NEW
```

ACCENTS SUR VIC 16K

```
15 PRINT"■■■ ATTENDEZ UN INSTANT"
20 FORI=5120TO7167
30 POKEI,PEEK(29696+I)
40 NEXT
30 POKE36869,205
100 FORI=1TO23
110 READA
120 FORJ=ATOA+7
130 READB
140 POKEJ,B
150 NEXT
170 NEXT
180 PRINT"■■■  ■■ ♠■└▔ ▔▔♥ ♠▔▔▔/■ ♥"
181 PRINT"■■ ┌═♠  ┐═♥  └═♦  ┘═♠"
182 PRINT"■■ ┴═▔  ┬═_  ▄═▔  ▌═_"
183 PRINT"■■ ▄═\  ***  ▌═/  ▐═\"
184 PRINT"■■ ▄═┌  ***  ▄═┐  ▐═└"
185 PRINT"■■ ▔═,  ▔═,  │═│  ▌═\"
186 PRINT"■■ ▔═—"
187 PRINT"■■ ▰═£  ◤=£+SHIFT"
188 PRINT"■■ π=PI  ↑=FLÈCHE HAUTE"
190 POKE198,7:POKE631,76:POKE632,207:POKE633,13
191 POKE634,82:POKE635,213:POKE636,13
192 END
500 DATA6000,32,16,56,4,60,68,58,0:REM ---- A
510 DATA6016,4,8,56,4,60,68,58,0
520 DATA5992,24,36,56,4,60,68,58,0
530 DATA6120,36,0,56,4,60,68,58,0
540 DATA6024,4,8,60,66,126,64,60,0:REM ---- E
550 DATA6032,32,16,60,66,126,64,60,0
560 DATA5984,24,36,60,66,126,64,60,0
570 DATA6104,36,0,60,66,126,64,60,0
580 DATA5904,6,8,0,24,8,8,28,0:REM -------- I
590 DATA5896,24,36,24,8,8,8,28,0
500 DATA5944,36,0,24,8,8,8,28,0
510 DATA6088,4,8,60,66,66,66,60,0:REM ----- O
520 DATA6008,24,36,60,66,66,66,60,0
530 DATA6064,36,0,60,66,66,66,60,0
540 DATA6080,4,8,82,66,66,70,58,0:REM ----- U
550 DATA6072,32,16,74,66,66,70,58,0
560 DATA6048,24,36,0,66,66,70,58,0
570 DATA6056,36,0,66,66,66,70,58,0
580 DATA5952,60,0,92,98,66,66,66,0:REM ---- CEDILLE
590 DATA5960,30,64,98,90,70,66,66,0:REM --- ENE
700 DATA6112,0,0,60,66,72,82,60,64
710 DATA5360,8,0,8,8,8,8,8,0:REM ---------- ? RENVERSE
720 DATA5872,8,0,8,16,32,36,24,0:REM ------ ! RENVERSE
```

Francis ESTEVE

Premier contact avec les disques ATARI et ORIC

Alors que les disques du 64 sont apparus depuis longtemps puisque ce sont les mêmes que pour le VIC, ceux de l'ORIC ne sont apparus que récemment. Ceux de l'ATARI (810) existent depuis longtemps, mais un nouveau modèle (1050) à l'esthétique XL est apparu récemment.

Le but de cet article n'est pas d'expliquer en détail les instructions disque, mais d'insister sur quelques points importants quelquefois négligés par les débutants. Ces points sont d'ailleurs les mêmes sur tous les systèmes.

Connexion

Atari

Les disquettes se connectent par le connecteur périphérique (trapézoïdal). Comme le connecteur est double, on peut enchaîner les connexions. On peut mettre jusqu'à 4 unités qui auront pour noms de périphérique D1:, D2:, D3: et D4:, par défaut D: désigne l'unité 1. Le numéro se détermine par des petits interrupteurs sur l'unité.

Oric

Les disquettes se montent sur le bus d'extension par un câble plat. Le connecteur d'extension est répété 2 fois ce qui permet d'autres connexions. On peut connecter jusqu'à 4 disquettes (0 à 3). La disquette 0 est dite disquette maîtresse. Les autres sont dites esclaves; elles sont d'un type différent: la maîtresse a un bouton de Reset rouge. Nous pensons que, pour le moment, Oric ne livre que les disquettes maîtresses. La disquette est livrée avec une alimentation qui alimente en outre l'unité centrale et une 2ième disquette. C'est un bon point, alors que l'Atari multiplie les boîtiers d'alimentation.

Media

Grande différence entre les deux: Atari utilise les disquettes 5 pouces 1/4 bien connues. Oric utilise les micro-disquettes 3 pouces. C'est un bon point pour l'encombrement mais les disquettes vierges sont chères et plus difficiles à trouver: nous avons eu nous-mêmes beaucoup de problèmes pour nous en procurer.

Compatibilité

Atari

L'unité 1050 peut fonctionner sous deux densités (simple ou double) sous DOS 3. L'unité 810 fonctionne en simple densité sous DOS 2. En simple densité on a 87 K disponibles (catalogue déduit). En double densité on a 127 K par disquette.

L'unité 1050 peut lire les disquettes simple ou double densité. Les disquettes qu'elle a formatées et écrites en simple densité (c'est une option de la commande de formatage I) peuvent être lues par l'unité 810. Enfin, sur 1050 sous DOS 3, on peut convertir les fichiers DOS 2 en DOS 3.

Oric

Les disquettes utilisables sont des Hitachi FC1 ou FC2. Chaque disquette a deux faces. L'unité Oric n'ayant qu'une tête de lecture écriture, il faut retourner la disquette pour utiliser la 2ième face. Il y a 159 K par face, de capacité.

Il faut noter que l'unité de

disquette est connectable tant à l'Oric-1 que l'ATMOS.

Les commandes disque

Les commandes disque sont de deux sortes:

1- gestion de la disquette, soit globalement soit au niveau d'un fichier.

2- lecture-écriture de données dans les fichiers.

Sur Atari, les commandes de type 1 sont toutes accessibles par un menu, ce qui est très pratique. Les instructions de lecture-écriture sont des cas particuliers des instructions d'E/S Basic dans lesquelles on spécifie le périphérique D:.

Sur Oric, toutes les commandes sont des instructions supplémentaires de Basic (elles commencent par !) utilisables en mode direct ou programmé.

Commandes de gestion.

Globales

Formatage : Toute disquette vierge doit subir un formatage. C'est la commande I (Init disk) sur Atari et !FORMAT sur Oric. Sur Atari, on a l'option de la densité.

Duplication : Permet de dupliquer une disquette sur une autre. Lorsque vous n'avez qu'une unité, le système vous indique sur l'écran qu'elle disquette source ou destination monter (il y a plusieurs montages et démontages). Sur Atari, c'est la commande Duplicate (D), sur Oric c'est !BACKUP.

Remarque fondamentale

A la différence des Commodore, tant les unités Atari que Oric n'ont pas le système d'exploitation (DOS) en ROM. Celui-ci réside sur la disquette livrée avec l'unité.

La première chose que vous devez IMPERATIVEMENT faire, est de dupliquer cette disquette en plusieurs exemplaires, avant tout autre essai de manipulation de la disquette. Il faut donc vous procurer des disquettes vierges avant de mettre votre unité sous tension.

Catalogue : Permet de lister les fichiers présents sur une disquette (ex. ci-dessous).

Oric :

```
O-System B    Directory  Page 1

  HELP.COM    7     HELP.01    4
  HELP.03     4     HELP.02    4
  HELP.04     4     HELP.05    4
  HELP.06     4     HELP.07    4
  HELP.08     4     HELP.09    4
  HELP.10     4     HELP.11    4
  HELP.12     4     HELP.13    4
  HELP.14     4     HELP.00    4
  HELP.16     4     HELP.15    4
  HELP.17     4     HELP.18    4
  HELP.19     4   SQUARE.CHS   4
 HELP.07A     4    SLANT.CHS   4
SYSTEM.DOS   45   DESIGN.COM   5
   SYS.COM    9      OLD.COM   1
   STD.CHS    4     BOLD.CHS   4
BOOTUP.COM    1     TELE.DAT   3
  TELE.COM    7     DEMO.COM   2
 ORIC1.LGO   10     ATMOS.LGO  13
 MICRO.LGO    7     TEXT.01    4
  TYPE.COM    1

press RETURN to continue >>
O-System B    Directory  Page 2
215 Used, 421 Free, Out of 636
```

Atari :

```
Filename     Size

FMS      SYS 004
KCP      SYS 001
KCPOVER  SYS 005
COPY     UTL 005
DUPDISK  UTL 004
INIT     UTL 006
CONVERT  UTL 005
HELP     UTL 002
HELP     TXT 012
HANDLERSSYS 001

042 FREE BLOCKS
```

Sur Atari, c'est la commande F (File index). Elle donne comme choix le périphérique de sortie utilisé, donc il est très facile d'avoir le catalogue sur imprimante au lieu de l'écran.

Sur Oric, c'est la commande !DIR. Elle donne le catalogue à l'écran, page par page (on fait RETURN pour passer à la page sui-

vante). Rien n'est prévu pour avoir le catalogue sur imprimante. Heureusement La Commode est là: voyez le petit article de ce numéro qui permet de vaincre cette difficulté.

Commandes sur fichiers

copie d'un ou plusieurs fichiers. C'est C sur Atari, !COPY sur ORIC.

effacement d'un ou plusieurs fichiers. C'est E sur Atari, !DELETE sur ORIC.

protection/déprotection d'un fichier. L'un et l'autre système offre plusieurs options.

changement de nom R (Atari) !RENAME (ORIC).

Chacun des systèmes a quelques autres commandes que nous ne détaillerons pas ici. Notons que chacun a une commande HELP qui fait afficher des informations pouvant nous aider.

Commandes d'écriture/lecture de données

Sauvegarde/Chargement d'un programme

Sur Atari, on emploie SAVE ou LOAD avec comme périphérique D:. Ex. SAVE "D:PROG". L'Atari a aussi le couple LIST/ENTER.

Sur ORIC on emploi !SAVE et !LOAD. Nous ne détaillons pas ici les différentes options possibles.

Lecture-écriture de données

Sur Atari, les commandes sont les commandes habituelles Basic, avec D: comme périphérique.

Un fichier doit d'abord être ouvert par OPEN.

Ex. Atari OPEN #2,8,0,"D1:FICH.DAT"
Oric !OPEN "O-FICH.DAT",W

On doit ensuite lire ou écrire, sur Atari par INPUT#/PRINT# ou GET#/PUT#, sur ORIC par !GET/!PUT

Enfin on doit fermer le fichier par CLOSE/!CLOSE

L'ORIC a en plus les ordres !STORE et !RECALL qui permettent de stocker un tableau entier et de le retrouver. En revanche l'ORIC n'a, dans sa version actuelle aucune provision pour l'accès direct. L'Atari, lui, permet l'accès direct grâce aux instructions NOTE# et POINT#.

Quelques difficultés des disquettes Oric

* problèmes hardware

Un certain nombre de revendeurs nous ont dit que les disquettes Oric "grillaient" par excès d'échauffement. Nous avons pu constater un échauffement important, mais sans catastrophe jusqu'à maintenant pour notre unité. Nous pensons que le problème vient du boîtier d'alimentation qui donne des tensions non régulées trop grandes, ce qui fatigue les régulateurs. Il est probable que ce problème est atténué lorsqu'on connecte une 2ième disquette. D'autre part, nous conseillons de ne pas alimenter l'unité centrale par le boîtier d'alimentation disquette mais de coserver l'ancienne alimentation. Celle-ci donne une tension non régulée déjà un peu trop élevée mais un peu moins que le boîtier disquette.

* problème cassette

On peut constater que, en présence de l'unité de disquette, l'ORIC-1 ou l'ATMOS n'arrive pas à lire une cassette. C'est plutôt fâcheux car la première chose que souhaite faire celui qui vient d'acheter une unité de disquette, c'est de copier sur disquette les logiciels qu'il a sur cassette (attention, des logiciels achetés peuvent ne pas être copiables pour raison de copyright) !

Le problème est d'arriver à lire le programme sur cassette hors de la présence de la disquette puis de brancher celle-ci sans que cela n'efface le programme en mémoire. La manip. qui permet d'y arriver est décrite dans ce numéro.

D.-J. DAVID

La loi uniforme

Uniforme, mais pas ennuyeuse.

I - Présentation

En statistiques, il existe plusieurs types de lois, les lois Normale, Binomiale, Uniforme pour ne citer que celles-ci.

Le programme présenté calcule la LOI UNIFORME.

II - Caractéristiques du Programme.

Langage..... Basic

Capacité mémoire.. 16 K

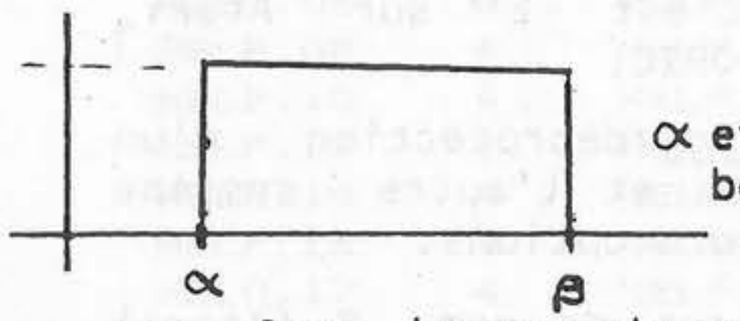
Machine..... CBM 3000

III - Organisation du Programme.

En observant un phénomène physique ou autre, on est amené a relever certaines valeurs, et l'on peut vouloir les analyser statistiquement.

Chaque phénomène peut répondre à une certaine distribution des résultats.

Pour le cas du programme, il s'agit d'une loi du type Uniforme dont la distribution est la suivante:



α et β sont les bornes de la répartition

Dans le cas du programme, il faut que le phénomène réponde à certaines conditions:

1. Les mesures doivent être organisées en classes de largeur constante.

2. A chaque classe doit correspondre un nombre de réalisations de la variable.

A partir de là, le programme calcule les probabilités mathématiques, l'espérance mathématique, les bornes de la répartition et le coefficient déterminant si la loi est suivie correctement.

Ce programme est assez spécifique, mais peut rendre certains services.

A l'intérieur du programme, est inclus un exemple d'utilisation qui permet d'en voir le fonctionnement, et de visualiser un type de tableau de résultats obtenus.

J. BRUGEASSOU

Lisez La Commode

```
0 R0$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒PRESSEZ [ 1 ] POUR UNE AUTRE LOI ■"
1 PRINT"▒"
2 COL$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
3 LNE$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
4 R1$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒PRESSEZ [ 1 ] POUR LE TABLEAU DE VALEUR ■"
5 DEFFNPR(X)=(INT(X*1000+0.5))/1000
6 :
7 REM----------------------------
8 REM      PRG   STATISTIQUE
9 REM      (C) BRUGEASSOU J
10 REM       LE 19/09/1981
11 REM        CBM 2001 8K
12 REM  VERSION 1.0   4271 OCTETS
13 REM----------------------------
14 :
15 PRINT"────────────────────────────────────▒▒"
16 PRINT"    CALCUL STATISTIQUE
17 PRINT"▒▒────────────────────────────────────"
18 PRINT" LOI UNIFORME "
19 GOSUB122
20 IFFLG=1THENMX=0:GOTO32
21 INPUT"▒▒ NOMBRE D'EXPERIENCES ";NE
22 INPUT" NOMBRE DE CLASSES     ";NC
23 INPUT" LARGEUR DES CLASSES   ";LC
24 DIMMI(NC),INF(NC),SUP(NC),PX(NC),X(NC),P(NC)
25 FORI=1TONC
26 PRINT"▒▒▒▒"
27 PRINT" VALEUR POUR LA CLASSE "I;:INPUTMI(I)
28 PRINT:PRINT
29 PRINT" VALEUR BORNE INF("I")     ";:INPUTINF(I)
30 SUP(I)=INF(I)+LC
31 NEXT:MX=0
32 FORI=1TONC
33 PX(I)=MI(I)/NE:X(I)=(SUP(I)+INF(I))/2
34 MX=MX+(X(I)*PX(I))
35 NEXT
36 DX=0
37 FORI=1TONC
38 DX=DX+(((X(I)-MX)*(X(I)-MX))*PX(I))
39 NEXT
40 Z0=SQR(12):Z1=2
41 SX=SQR(DX)
42 BETA=((Z0*SX)+(Z1*MX))/Z1
43 ALPHA=(Z1*MX)-BETA
44 A1=1/(BETA-ALPHA)
45 P(1)=(SUP(1)-ALPHA)*A1:P(NC)=(BETA-INF(NC))*A1
46 FORI=2TONC-1
47 P(I)=(SUP(I)-INF(I))*A1
48 NEXT
49 KHI2=0
50 FORI=1TONC
51 A=NE*P(I)
52 A=((MI(I)-A)*(MI(I)-A))/A
53 KHI2=KHI2+A
54 NEXT
55 R=NC-3
56 REM   RESULTATS
```

```
57 PRINT"⊐  ⊠CARACTERISTIQUES DE LA REPARTITION■"
58 PRINT" -------------------------------------"
59 PRINT:PRINT
60 PRINT" BORNES DE LA REP."
61 PRINT"
62 PRINT" .................ALPHA="FNPR(ALPHA)
63 PRINT" ..................BETA="FNPR(BETA)
64 PRINT
65 PRINT" ESPERENCE MATHEMATIQUE "FNPR(MX)
66 PRINT"
67 PRINT" VARIANCE               "FNPR(DX)
68 PRINT"
69 PRINT" ECART TYPE             "FNPR(SX)
70 PRINT"
71 PRINT" VALEUR DU KHI2         "FNPR(KHI2)
72 PRINT"
73 PRINT" NB. DE DEGRE DE LIBERTE"R
74 PRINT"
75 PRINTR1$
76 GETA$:IFA$=""THEN76
77 IFA$="1"THEN79
78 GOTO113
79 PRINT"⊐"
80 IFNC>9THEN111
81 PRINT" ⊠CLASSE VALEURS  PROBA  X TILDA PROBA "
82 PRINT"┌─────┬─────┬─────┬─────┬─────┐
83 PRINT"│     │     │     │     │     │
84 PRINT"├─────┼─────┼─────┼─────┼─────┤
85 PRINT"│     │     │     │     │     │
86 PRINT"├─────┼─────┼─────┼─────┼─────┤
87 PRINT"│     │     │     │     │     │
88 PRINT"├─────┼─────┼─────┼─────┼─────┤
89 PRINT"│     │     │     │     │     │
90 PRINT"├─────┼─────┼─────┼─────┼─────┤
91 PRINT"│     │     │     │     │     │
92 PRINT"├─────┼─────┼─────┼─────┼─────┤
93 PRINT"│     │     │     │     │     │
94 PRINT"├─────┼─────┼─────┼─────┼─────┤
95 PRINT"│     │     │     │     │     │
96 PRINT"├─────┼─────┼─────┼─────┼─────┤
97 PRINT"│     │     │     │     │     │
98 PRINT"├─────┼─────┼─────┼─────┼─────┤
99 PRINT"│     │     │     │     │     │
100 PRINT"└─────┴─────┴─────┴─────┴─────┘"
101 PRINT" ⊠SOMMES■▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
102 FORI=1TONC
103 COL=2:LNE=2+2*I
104 GOSUB117:PRINTITAB(10)FNPR(MI(I))TAB(16)FNPR(PX(I))TAB(24)FNPR
    (X(I))
105 GOSUB117:PRINTTAB(32)FNPR(P(I))
106 Z3=Z3+MI(I):Z4=Z4+PX(I):Z5=Z5+P(I)
107 FORT=1TO500:NEXT
108 NEXT
109 LNE=22:COL=2:GOSUB117:PRINT"⊠"TAB(10)FNPR(Z3)TAB(16)FNPR(Z4)
    TAB(32)FNPR(Z5)
110 GOTO113
111 PRINT"JE NE PEUX FAIRE CELA POUR UNE LOI DE      PLUS "
112 PRINT"DE 9 CLASSES "
113 PRINTR0$
114 GETA$:IFA$=""THEN114
115 IFA$="1"THENRUN
116 PRINT"⊐":END
```

```
117 REM
118 PRINT"■";
119 IFCOL>1THENPRINTLEFT$(COL$,COL-1);
120 IFLNE>1THENPRINTLEFT$(LNE$,LNE-1);
121 RETURN
122 REM EXPLICATION
123 REM-------------
124 PRINT" ──────────■■■■"
125 PRINT" CE PROGRAMME PERMET DE CALCULER LA STA--TISTIQUE D'EVENEMENT
 ]                                                        SUIVANT ";
126 PRINT" 1 LOI DE  TYPE UNIFORME
127 FORI=1TO5000:NEXT
128 PRINT" ■■VOULEZ VOUS UN EXEMPLE ■■UI "
129 GETA$:IFA$=""THEN129
130 IFA$="0"THEN132
131 RETURN
132 READNE,NC,LC
133 FORI=1TO8
134 READMI(I),INF(I)
135 SUP(I)=INF(I)+LC
136 NEXT
137 PRINT"■ETUDE D'UN ALTIMETRE "
138 PRINT"──────────────────"
139 PRINTNE" EXPERIENCES
140 PRINTNC" CLASSES
141 PRINTLC" LARGEUR DES CLASSES
142 PRINT"──────────────────────────────────────────"
143 FORI=1TONC
144 PRINT"CLASSE "I"MESURE "MI(I);
145 PRINT" INF("I")"INF(I)
146 PRINT"──────────────────────────────────────────"
147 NEXT
148 PRINT"UNE TOUCHE POUR LANCER LE CALCUL"
149 GETA$:IFA$=""THEN149
150 FLG=1:PRINT"■":RETURN
151 DATA400,8,10
152 DATA21,20,72,30,66,40,38,50,51,60,56,70,64,80,32,90
```

Adaptation à d'autres machines

Le programme passe tel quel sur 64. Au contraire, pour le VIC, toutes les impressions seraient à changer à cause de la largeur d'écran plus réduite.

Pour l'ORIC, seuls posent problème les caractères mouvements de curseur incorporés un peu partout (lignes 0, 1, 2, 3, 4, 15, 17, 21, 26, 57, 116, 118, 124, 128, 137 et 150) (excusez les oublis).

Le coeur signifie vidage écran et il est à remplacer par CHR$(12), le S est à remplacer par CHR$(30), le Q (->) par CHR$(10), le |(<-) par CHR$(9).

Le R qui signifie contraste inversé et le trait qui signifie retour au contraste normal peuvent être omis. Les caractères graphiques qui permettent de former le tableau (lignes 82 à 101) peuvent être simulés par - I et *. Maintenant, au prix d'une ré-écriture, les positionnements peuvent être remplacé par des PLOT : COL$ et LNE$ n'interviennent qu'en 119 et 120.

Dans l'ATARI, les caractères CTRL mouvement de curseur peuvent être incorporés dans une chaîne à condition de les précéder de ESC, mais là aussi, il est peut-être

préférable d'utiliser POSITION. N'oubliez pas de dimensionner toutes les variables chaînes de caractères. En 119 écrire :

```
......PRINT COL$ (1, COL-1) ;
```

Même chose en 120.

L'absence de DEFFN crée une difficulté. La ligne 5 doit être supprimée et partout où FNPR est employée, remplacer par l'expression explicite. Donc en 62, 63, 65, 67, 69, 71, 104, 105, 109 écrire par exemple :

```
62 PRINT ".... ALPHA=" ; INT (ALPHA
 * 1000 + 0.5) / 1000
```

104, 105 et 109 sont à réécrire car TAB n'existe pas: le plus simple est d'imprimer une chaîne d'espaces entre guillemets et ajuster les nombres d'espaces; Ex. :

```
105 GOSUB 117 : PRINT "          "
; INT (P(I) - 1000 + 0.5)/1000
```

La dernière difficulté vient des GET lignes 76, 114, 129 et 149. Ici, il suffit de les remplacer par INPUT. Cela imposera de terminer les réponses par "RETURN" ce qui n'est pas gênant dans ce contexte.

Pierre-Etienne THALBERG

Magnéto Ordinaire

La connexion d'un magnéto ordinaire devrait -sur le papier- procurer une certaine économie par rapport au magnéto Commodore ou Atari. Oric a décidé d'office de faire appel à un magnéto ordinaire. La Commode a publié (dans le numéro 3, épuisé : photocopie possible pour nos abonnés, coût 18 francs) un schéma d'interface Commodore - magnéto ordinaire.

Eh bien SEDERMI commercialise cette interface pour 120 F modèle A, et 170 F modèle B (ce dernier modèle gère le moteur ; il est fortement conseillé pour le 64 qui comme vous le savez interrompt la lecture lorsqu'il vient de trouver le nom de fichier: il ne faut pas que la bande défile pendant ce temps). Vous pouvez commander par simple lettre au 28 rue Vicq d'Azir.

Mais l'utilisation d'un magnéto ordinaire pose quelques problèmes pour trouver le réglage optimum. En principe, il faut être près du maximum des aigus et de la puissance.

Sur ORIC , mettez vous au maxi de puissance. Si le système ne trouve pas de fichier, diminuez un peu . Si le système trouve le fichier mais ne charge pas convenablement (le listing se termine par une flopée de UUUU...) diminuez encore un peu. Si vous avez le diagnostic " Load Aborted " ou "Errors Found ", c'est que vous avez trop diminué. Maintenant, sur ATMOS, le diagnostic ne prouve rien: vérifiez le programme lui-même.

Un bon magnéto

Nous avons repéré un modèle de magnéto qui nous semble donner d' excellents résultats tant avec l' ORIC qu'avec des Commodore (via notre interface). Il a même une fois pu lire sur 64 un programme qui donnait LOAD ERROR avec le propre magnéto Commodore. Il s'agit - réclame non payée - du PHILIPS DATA RECORDER 6600/30P. Attention ne vous laissez pas coller un 6600 tout court. Le 30P est, manifestement, spécialement conçu pour enregistrer des données.

Jean RENAUD
Pierre-Etienne THALBERG

64 à brac

Chargement accéléré des programmes sur cassettes pour le VIC ou le C-64

Une des raisons de la lenteur du chargement à partir des cassettes est la répétition de cette opération. La routine LOAD charge le programme une première fois puis compare ce qu'il y a maintenant en mémoire avec la seconde copie du programme sur la cassette . Si les deux ne sont pas identiques, le message "Load error" est généré. Le mieux est ici l'ennemi du bien car si la 1ère version est correcte et la seconde mauvaise, il y aura un "Load error" bien que le programme en mémoire soit correct. On peut seulement le lister. Si on édite une ligne ou si on fait RUN, le système se plante aussitôt.

La méthode suivante va vous permettre de ne charger que la première version et de la faire marcher. Vous pouvez ainsi récupérer quelques programmes réticents et surtout diviser le temps de chargement par deux, ce qui est appréciable pour le 64 et ses programmes marathons.

Bon, mettez un programme quelconque et tapez LOAD suivi de Return. Lorsque vous pensez avoir dépassé la moitié du temps de chargement, pressez stop. Faites ensuite:

```
POKE 45,PEEK(174)
:POKE 46,PEEK(175) :CLR
```

et vous pouvez maintenant éditer ou exécuter votre programme.

Pour vous aider à déterminer la moitié du chargement voici quelques trucs:

après le break, regardez PEEK(167); si c'est 2, c'est la première version et si c'est 1, c'est la seconde version qui était en cours de chargement. Vous pouvez aussi déterminer votre éloignement par rapport à la moitié en regardant le pointeur (173,174).
Soit P0 = PEEK(173)+256*PEEK(174) sa valeur.

Les adresses de début et de fin de votre programme sont respectivement DE = PEEK(193) + 256 * PEEK(194) et FI = PEEK(174) + 256 * PEEK(175).

Lors du chargement de chaque version, P0 parcourt l'intervalle entre DE et FI.

Pour optimiser votre chargement, repérez la valeur du compteur du magnétophone pour lequel un break donne peek (167) = 1 et P0 le plus proche possible de DE (souvent 2049).

Voilà c'est simple et çà marche.

DIVERS SUR LE C-64

Sauvegarde sur disquettes

- On connaît le problème de la synchronisation entre le 64 et son lecteur de disquette. C'est l'unique raison pour laquelle l'unité 1540 du VIC est incompatible avec le 64. On peut cependant charger et sauver un programme en inhibant la gestion d'écran par un POKE 53265,11 avant toute opération. Le problème n'est pas tout à fait réglé avec la I541 car si des sprites sont actifs, il y a parfois des problèmes (1 fois sur 10 en moyenne). Donc éteignez les tous par POKE 53269,0 auparavant.

- Un problème plus grave est le bug du DOS concernant la commande @ (replace). De temps en temps, elle

fait n'importe quoi en remplaçant le programme désiré, non par celui en mémoire, mais par un programme quelconque du répertoire. Résultat: vous n'avez plus aucune trace de votre programme! Ne l'utilisez jamais , sauf quand le programme remplaçant est de même longueur exactement. Je dis cela pour ceux qui travaillent en FORTH et qui ne peuvent s'en passer. Là, il n'y a pas de problèmes.

Sauvegarde d'un bloc en langage machine sans moniteur

Soit à sauver les octets compris entre les adresses A et B avec

```
A = LA+256*HA et B = LB+256*HB
SAVE "TITRE",1,1 (return puis stop)
POKE 780, 253
POKE 253,LA : POKE 254,HA
POKE 781,LB+1 : POKE 782,HB
SYS 62941          ($F5DD)
```

Voir un traitement plus complet de ce problème dans l'article spécifique de ce numéro.

Imprimer l'adresse de début d'un programme d'une disquette

```
10 OPEN 1,8,2, "TITRE, P,R"
20 GET   1, A$, B$
30 PRINT ASC ( A$ + CHR$(0) ) +
256*ASC ( B$ + CHR$(0) ): CLOSE 1
```

Déplacer rapidement une zone mémoire vers une autre

Le basic est trop lent pour cela. Par exemple si vous voulez recopier le basic en RAM pour le modifier vous feriez:

```
FOR I = 160*156 TO 192*256: POKE I,
PEEK (I): NEXT.
```

Cela dure 40 secondes.

On peut remplacer cela par:

```
POKE 88,0 : POKE 89, 192
POKE 90,0 : POKE 91, 192
POKE 781, 32+1  (nombre de pages à
translater)
SYS 41971
```

même résultat 0,5 seconde après.

Maintenant faites POKE 1, 54 et amusez vous.

A quelle génération appartient votre C-64 ?

- Allumez votre C-64 et passez en écriture noire (CTRL 0), effacez l'écran (SHIFT CLR) et faites POKE 1600,1:

si un A blanc apparait, il est de la 1ère génération
Si rien n'apparait, il est de la 2ème génération
Si un A noir apparait, il est de la 3ème génération.

En effet, selon le cas, la commande SHIFT CLR efface l'écran et initialise la carte couleur($D800-) avec du blanc, la couleur du fond ou la couleur de l'écriture.

Hervé LE MARCHAND.

Lisez La Commode

MICRO APPLICATION : NOUS PRENONS LE LOGICIEL AU SERIEUX

Vous possédez un Commodore 64. Utilisez-le à fond. MICRO APPLICATION vous en donne maintenant la possibilité, grâce à sa gamme complète de programmes en français*.

Créez :

PAINTPIC : Un programme révolutionnaire pour dessiner, peindre et colorier à l'écran. Va au devant de votre imagination. Un logiciel indispensable à tous.
SYNTHY 64 : Utilisez à fond les capacités musicales de votre ordinateur. Permet la composition et l'exécution de partitions musicales polyphoniques.

Développez :

ZOOM PASCAL : Le langage le plus populaire après BASIC. Programmation structurée. Comprend un éditeur, compilateur et traducteur.
TRI FORTH : Le langage du futur : développement efficace et rapide.
ARROW : Le langage de la machine : comprend un assembleur, un éditeur et un accès cassette accéléré.

Jouez :

MICRO APPLICATION, c'est le logiciel au sérieux mais c'est aussi la détente. Des jeux pour se distraire...
STAR CRASH, POKER, SKIER, POOL, TROBOTS, COSMIC SPLIT...
... et pour réfléchir :
DAEDALUS, SUPER DAEDALUS, LOGIK...
MICRO APPLICATION : Une gamme de jeux pour réfléchir en s'amusant et s'amuser en réfléchissant.

Tous nos programmes en français existent sur disquette, cassette ou cartouche.

Notre catalogue vous permettra d'en savoir plus sur les prix et les caractéristiques de tous nos programmes.

MICRO APPLICATION
147, avenue Paul Doumer
92500 RUEIL MALMAISON FRANCE
Tél. (1) 732.92.54 · Telex MA 205 944 F

SOFTWARE

* à partir de 95 FF TTC

Je désire recevoir sans engagement le catalogue gratuit de l'ensemble de vos programmes.

Nom .....

Adresse .....

.....

C. Postal ..... Ville .....

CO

Simulation de saisie sur ORIC

Dans le n° 5 de La Commode D.J. DAVID a, sous le titre "Le Tampon Clavier", démonté le mécanisme de la saisie simulée, bien connue des utilisateurs de modèles Commodore sous le nom de clavier dynamique.

Il en a montré les possibilités, notamment l'auto-modification d'un programme par insertion automatique de lignes.

Et l'ORIC ?

Peut-on arriver au même résultat sur ORIC ?

Ce n'est pas à priori évident, pour deux raisons.

La première est liée au principe même de le saisie simulée. On génère par programme une ligne d'instruction BASIC et on l'affiche sur l'écran. Sur les Commodore il suffit alors d'amener le curseur sur la dite ligne et de faire RETURN pour qu'elle soit aussitôt et entièrement placée dans le tampon d'entrée,puis exploitée.

Pas question d'en faire autant sur l'ORIC. Son éditeur d'écran, très sommaire, oblige vous le savez à repasser laborieusement le curseur sous tous les caractères avec CTRL-A, pour une prise en compte effective. Il est cependant assez facile de pallier ce point faible.

La deuxième raison qui nous interdit une simulation automatique de saisie est due au fait que l'ORIC n'est pas pourvu d'un tampon clavier capable de stocker plusieurs octets. Donc impossible de mettre à l'avance par des POKE les mouvements de curseur et le RETURN servant précisemment à commander la saisie de la ligne présente à l'écran. Il existe seulement une unique case mémoire (#2DF) appelée mémoire clavier qui contient le code ASCII+128 de la dernière touche enfoncée. C'est maigre mais pas sans intérêt comme nous le verrons.

Saisie d'une ligne à l'écran.

Supposons une ligne BASIC affichée à un emplacement déterminé sur l'écran, tout en haut par exemple. L'adresse écran du premier caractère est dans ce cas #BBAA. Rappelons au passage que cette ligne BASIC peut s'étendre en fait sur deux lignes d'écran successives.

Un petit programme, AUTOLM, de 47 octets, listing n° 1, va nous permettre de lire les codes des caractères de la ligne sur l'écran, de les transférer dans le tampon d'entrée, d'en marquer la fin par un code 0, et puis d'appeler l'interpréteur BASIC après avoir mis les pointeurs voulus sur l'adresse de début du tampon. Et le tour est joué !

Deux cas peuvent alors se présenter:

1. La ligne commençait par un numéro. Elle est alors insérée dans le programme BASIC existant éventuel.

2. Les instructions n'étaient pas précédées d'un numéro. Elles sont alors purement et simplement exécutées, comme en mode direct.
Si! Vous allez voir, cela peut aussi être très utile.

Création de lignes : AUTODATA

La création de lignes de DATAs est une application typique. Pour

peu qu'il soit nécessaire de mettre sous cette forme un nombre important de codes mémoire (redéfinition de caractères, programme en langage machine, ...),le gain de temps est important et surtout on élimine les risques d'erreurs.

AUTODATA, voir listing n° 3, associé à AUTOLM ne permet pas une saisie 100% automatique en raison de l'absence de tampon clavier. Nous n'en sommes toutefois pas loin puisqu'il suffit de maintenir un doigt sur RETURN pour déclencher la création d'une nouvelle ligne, et ce à cadence élevée; guère plus d'une seconde par ligne !

Commentaires :

- Les lecteurs disposant d'un moniteur entreront directement le programme en langage machine. Les autres utiliseront le listing 2 et une boucle classique de chargement:
FOR I=#490 TO #4BF: READ A
: POKE I,A: NEXT
L'adresse de début a été choisie pour permettre un groupement avec la routine APPEND, décrite dans ce numéro, mais elle peut être différente, le programme étant relogeable.

- La ligne 0 d'AUTODATA est très importante. C'est dans celle-ci que seront stockés les trois paramètres: adresse de début, incrément et n° de ligne utilisés à chaque nouvel appel. On sait en effet que la création- ou la modification- d'une ligne, entraine la perte de toutes les variables.

Attention ! Il y a un espace entre DATA et le premier zéro.

- Si vous avez ajouté AUTODATA à un programme existant, il faut une ligne supplémentaire:
1 GOTO 60000

- Il est prévu 8 DATAs par ligne créée. Vous pouvez aller jusqu'à 16 (avec 3 chiffres écran). Pour cela modifiez le début des lignes suivantes:
60025 FOR J=D TO D+15: ....
60030 D=D+16: ....

- En 60040, juste avant l'appel du sous-programme en LM, on remarquera le forçage en #2DF, embryon de tampon clavier, du code 161 (128+33), c'est à dire de "!". C'est l'astuce qui permet de relancer le programme par un simple appui sur RETURN, l'adresse affectée à la fonction ! étant #C98D, celle de RUN.

- La sortie se fait par CTRL-C. En cas de reprise remettre si nécessaire les trois DATA de la première ligne à zéro.

Exécution immédiate de la ligne saisie

Il y a là beaucoup plus d'applications envisageables qu'il n'y paraît. On peut par exemple en cours d'exécution d'un programme saisir une expression mathématique qui sera évaluée (simulation de la fonction EVAL de certains BASIC sophistiqués).

Application plus simple, mais bien utile, l'examen de variables ou du contenu d'adresses mémoire, lors de la mise au point d'un programme. Normalement il faut à chaque fois taper une ligne d'instructions adéquates. C'est vite fastidieux. Voici mieux si vous avez chargé AUTOLM, listing 1, avec comme adresse de début de ligne à lire #BF92 (POKE #491,146
: POKE #495,191):

- Réservez les deux lignes en bas d'écran par POKE 623,25. La première commence précisément en #BF92.

- Placez par PLOT 1,25,"....." et si nécessaire par PLOT 1,26,"....." vos instructions dans ces lignes. Vous pourrez les modifier par la suite à votre gré.

- N'oubliez pas de terminer par "\" (barre oblique inversée). Ce n'est pas strictement indispensable, mais pratiquement nécessaire.

- Pour terminer:
DOKE #2F5,#490

Et maintenant il vous suffira, soit de mettre aux endroits voulus de votre programme l'instruction "!", soit après un arrêt de taper ! <RETURN>, pour obtenir l'affichage désiré ou encore la reprise de l'exécution avec des paramètres définis.

Exemples:

```
?CHR$(30);A;B%,C$\
FORI==XXX TO YYY:?PEEK(I);:NEXT\
A=3:X=PI/2:GOTO1250\
```

etc..., toutes les instructions BASIC étant permises.

Il est même possible de boucler, mais pas indéfiniment, en terminant par ...:!\.

Conclusion (provisoire)

La simulation de saisie offre d'intéressantes possibilités, loin d'être limitées aux exemples décrits.

Et comme rien ne vaut un exercice pratique pour bien assimiler un concept, nous vous proposons de résoudre pour l'ORIC le dernier cas traité dans l'article de DJD: entrée en cours de programme de la definition d'une fonction DEFFN..., qui sera placée dans une ligne d'instruction. A vous de jouer !

Jean RENAUD

Modifications pour l'ATMOS

listing 1: à la fin 4C C1 C4 (JMP C4C1) au lieu de 4C D1 C4

listing 2: dernière ligne:
150 DATA 169,0,133,234,76,193,196,0

listing 3: ligne 60050 :
60050 .....:DOKE #2F5,#C9BD

```
I 490-4BF
0490:   A9  AA      LDA #$AA      }  adresse début ligne
0492:   85  00      STA $00       }  à lire mise en 0,1
0494:   A9  BB      LDA #$BB      }
0496:   85  01      STA $01       }
0498:   A0  00      LDY #$00
049A:   A2  00      LDX #$00
049C:   B1  00      LDA ($00),Y     lecture ligne
049E:   C9  5C      CMP #$5C      "\"? (fin de ligne)
04A0:   F0  0E      BEQ $04B0
04A2:   95  36      STA $36,X       entrée dans tampon
04A4:   E8          INX
04A5:   C8          INY
04A6:   C0  26      CPY #$26        38 car. ? (1ère ligne)
04A8:   D0  02      BNE $04AC
04AA:   C8          INY             on saute les 2 colonnes
04AB:   C8          INY             de gauche
04AC:   C0  4D      CPY #$4D        77-2 car. ?
04AE:   D0  EC      BNE $049C
04B0:   A9  00      LDA #$00        code 0 en fin d'entrée
04B2:   95  36      STA $36,X
04B4:   A9  35      LDA #$35      }
04B6:   85  E9      STA $E9       }  adresse tampon
04B8:   A9  00      LDA #$00      }  dans pointeurs
04BA:   85  EA      STA $EA       }
04BC:   4C  D1  C4  JMP $C4D1       --> BASIC
04BF:   00          BRK                (CHARGET)
```

Listing 1

```
100 DATA 169,170,133,0,169,187,133,1
110 DATA 160,0,162,0,177,0,201,92
120 DATA 240,14,149,54,232,200,192,38
130 DATA 208,2,200,200,192,77,208,236
140 DATA 169,0,149,54,169,53,133,233
150 DATA 169,0,133,234,76,209,196,0
```

Listing 2

```
0 DATA 00000,00,00000
10 :
20 :
60000 REM AUTODATA
60010 READ D,I,L:IFD+I+L=0THENGOSUB60050'
60020 PRINTCHR$(30)MID$(STR$(L),2)" DATA
 ";
60025 FORJ=DTOD+7:PRINTMID$(STR$(PEEK(J)
),2)CHR$(44);:NEXT:PRINTCHR$(8)"\"
60030 D=D+8:L=L+I:POKE616,3:PRINT:PRINT"
Adresse debut ligne suivante: "D;
60040 GOSUB60090:POKE#2DF,161:CALL#490
60050 PAPER0:INK2:CLS:POKE#BC21,3:DOKE#2
F5,#C98D
60060 FORJ=1TO8:PRINT:NEXT:INPUT"Adresse
 decimale de debut ";D
60070 PRINT:INPUT"Numero de 1ere ligne D
ATA ";L
60080 PRINT:INPUT"Increment numeros de l
ignes ";I
60090 N$=RIGHT$("0"+MID$(STR$(I),2),2):M
=#50C:GOSUB60120
60100 N$=RIGHT$("0000"+MID$(STR$(D),2),5
):M=#506:GOSUB60120
60110 N$=RIGHT$("0000"+MID$(STR$(L),2),5
):M=#50F
60120 FORJ=1TOLEN(N$):POKEM+J,ASC(MID$(N
$,J,1)):NEXT:RETURN
```

Listing 3

Le mode RETURN automatique sur ATARI

Nous avons montré (dans La Commode n° 4) comment le tampon clavier permettait aux Commodore de simuler des frappes clavier par programme. Un article de ce numéro montre comment faire sur ORIC. Cet article doit permettre à nos amis Ataristes de ne pas être en reste.

Supposons que nous ayons à simuler la frappe de quelques lignes (par exemple des lignes de programme à ajouter). Il faut obéir aux étapes suivantes:

1- vider l'écran (par ?"ctrl clear" ou ?CHR$(125)

2- taper les lignes voulues vers le milieu de l'écran, disons entre les lignes (d'écran) L1 et L2. Les lignes BASIC doivent avoir leurs n°s en tête.

3- faire POSITION 0,0:POKE 842,13 (met en marche le mode RETURN automatique)

4- faire POSITION 2,L3:PRINT "CONT" Il faut L3 > L2.

5- faire POSITION 2,L0:STOP
Il faut L0 > L1

6- POKE 842,12 remet en mode normal.

Faites tourner l'exemple ci-dessous:

```
10 ? CHR$(125)
15 POSITION 2,10
20 ? "100?X"
30 ? "110?Y"
40 ? "120?Z"
50 POSITION 0,0:POKE 842,13
60 POSITION 2,15:? "CONT"
70 POSITION 2,5:STOP
80 POKE 842,12
```

En faisant LIST après l'exécution vous verrez qu'il contient maintenant les lignes 100, 110 et 120 (ce qui explique les trois 0 imprimés).

Applications

Suppression en masse

Une première application est de supprimer toutes les lignes dans un intervalle: il suffit d'imprimer les numéros de ligne. Le programme ci-dessous supprime les lignes de numéros compris entre les limites spécifiées.

```
10 ? "DEB,FIN";:INPUT D,F
20 FOR I=D TO F STEP 20
30 ? CHR$(125):POSITION 2,3
35 K9=F-I:IF K9>19 THEN K9=19
40 FOR J=0 TO K9:? I+J:NEXT J
50 POSITION 0,0:POKE 842,13
60 POSITION 2,23:? "CONT"
70 POSITION 2,0:STOP
80 POKE 842,12
90 NEXT I
```

Les autres applications possibles consistent à fabriquer les lignes de DATA correspondant à une routine en langage machine ou à entrer une fonction sur le moment pour l'intégrer ou tracer sa courbe représentative. Voici un programme d'intégration général.

Rita A.

```
10 ? "DONNEZ LA FONCTION A INTEGRER SOUS
 LA FORME Y=F(X)"
20 DIM F$(30)
30 INPUT F$
40 ? CHR$(125):POSITION 2,10
50 ? "1000";F$
60 POSITION 0,0:POKE 842,13
70 POSITION 2,15:? "CONT"
80 POSITION 2,5:STOP
90 POKE 842,12:? CHR$(125)
100 ? "BORNES,NB DE PAS"
110 INPUT A,B,N
120 H=(B-A)/N
130 X=A:GOSUB 1000:S=Y
140 X=B:GOSUB 1000:S=(S+Y)/2
150 FOR X=A+H TO B STEP H
160 GOSUB 1000:S=S+Y:NEXT X
170 S=H*S
180 ? "INTEGRALE = ";S
190 END
1000 Y=3*SIN(X)
1010 RETURN
```

Les logiciels La Commode

Au catalogue : UTIL-VIC-KIT de H. LE MARCHAND, un programme qui ajoute des fonctions au BASIC de votre VIC:
SET (établit les couleurs écran et cadre) CLEAN (vide l'écran)
PLOT (trace un point en haute résolution) JOYO (lit les JOYSTICKS)
LOOK (recherche une chaîne) KEY (affecte une touche de fonction)
PAUSE et bien d'autres.
En bref, un résumé de Programmer's Aid et Super Expander pour 80 F.

ASSEMBLEUR UNIVERSEL de J. RENAUD
Un éditeur-assembleur complet. Traitement des opérantes symboliques, sauvegarde de votre texte sur disque ou cassette.
Pour tous les COMMODORE (VIC à partir de 16 K) et ORIC; ATARI en préparation
prix : 180 F. (+30 F sur disquette) (+50 F version compilée sur 64)

BASICOIS de H. LE MARCHAND, (pour VIC de base ou étendu)
Si l'anglais est un obstacle à votre compréhension de l'informatique, BASICOIS est fait pour vous. Prix 150 F.
MOTS CLES : ABS, ET, CODE, ATN, CAR$, FERME, EFF, CMD, CONT, COS, DONNEES, DEF, DIM, EXP, SI, DEMANDE, ENT, FN, POUR, LBR, FRAPPE, APPEL, VA-AU, GAU$, LONG, SOIT, VOIR, RAPPEL, LOG, MI$, VIDE, REPETE, NON, SUR, OUVRE, OU, MEM, METS, POS, AFFICHE, LIS, REM, REPRENDS, RETOUR, DROITE$, ALE, FAIS, RANGE, SGN, SIN, EXP, RCN, PAS, HALTE, SEQ$, SYS, TAB, TAN, ALORS, JUSQUE, USR, VAL, VERIFIE, ATTEND .

TELECRAN de D.J. DAVID
(Pour VIC, 64, ATARI; ORIC en préparation)
Le jeu bien connu de TELECRAN, mais là, le trait peut être interrompu et il y a des traits diagonaux. Votre dessin peut être sauvegardé sur disque ou sur cassette. Prix : 80 F.(+50 F version compilee 64).

SUPER-COMBAT de J.P. MORARD
(pour VIC de base)
Un jeu d'envahisseurs très rapide bien qu'en Basic. Rien que l'examen du listing est très instructif. Prix : 80 F.

BD64 de D.J. DAVID
Gestion d'une base de données sur VIC ou 64 équipé de sa 1541.
Prix: 290F (disquette seulement).

BON DE COMMANDE en page 16.

Sauvegarde et chargement d'une zone mémoire

Le sauvegarde et le chargement d'une zone mémoire dont on donne les adresses extrèmes est une chose évidente sur ORIC où il suffit de faire un

CSAVE, A début, E fin

Ceci est utile pour sauvegarder un programme en langage machine, une table ou un dessin sur écran.

Sur COMMODORE ou ATARI, on a toujours la possibilité de stocker chaque octet de la zone en tant que donnée :

```
COMMODORE

1000 OPEN 2,8,2,"0:nom,S,W"
1010 FOR I = début TO fin:
     A = PEEK(I)
1020 PRINT#2,CHR$(A);
1030 NEXT
1040 CLOSE 2

ATARI

1000 OPEN #2,8,0, "D: nom"
1010 FOR I : début TO fin :
     A = PEEK(I)
1020 PRINT#2,CHR$(A);
1030 NEXT I
1040 CLOSE#2
```

Ce qui est ennuyeux avec ce procédé, c'est qu'on obtient un fichier beaucoup plus long à lire qu'un fichier programme. Sur COMMODORE, nous savons constituer un fichier programme. Nos investigations se poursuivent sur ATARI et nous ne manquerons pas de faire part de leur résultat aux lecteurs de La Commode.

SAUVEGARDE D'UNE TABLE EN FICHIER PROGRAMME SUR COMMODORE

L'avantage de cette méthode est que le fichier est rechargable par un simple LOAD Basic :

LOAD "nom", 8 ou 1,1 (le ,1 final est indispensable !)

Comme pour la musique, il y a en fait deux méthodes, la bonne et la mauvaise.

Commençons par la mauvaise. Nous la citons parce qu'elle est instructive et aussi pour qu'il n'y ait pas que de bonnes choses dans La Commode : cela finit par faire de la concurrence déloyale aux autres revues !

La technique consiste à employer le SAVE de Basic. Oui, mais il faut lui fournir les adresses de début et de fin. Pas difficile ! On met les adresses dans les pointeurs début et fin Basic et le système croit qu'il sauve un programme:

```
POKE 43,DB : POKE 44,DH :
POKE 45,FB : POKE 46,FH
SAVE "nom", P
```

avec DB = adresse de début basse,...
FH = adresse de fin haute,
P=1 ou 8

En quoi cette méthode est-elle mauvaise ?

Eh bien, c'est qu'elle détruit les pointeurs fondamentaux de BASIC. Donc, si l'opération est faite depuis un programme Basic, les pointeurs doivent être sauvegardés - attention, pas dans des variables : dans des cases mémoire tranquilles, car les variables sont détruites - et restaurés. Quant au pendant de la méthode, le LOAD, il ne peut être employé à l'intérieur d'un programme Basic puisqu'après, le système donne le contrôle à un programme Basic qu'il croit commencer à des adresses fausses.

LA BONNE METHODE

Elle consiste à utiliser la routine du noyau du système d'exploitation (le "KERNAL") qui fait la sauvegarde. Elle est en $FFD8. Elle s'utilise comme suit :

- mettre aux adresses ZZ et ZZ+1 de la page zéro l'adresse de début de la zone à sauver. Charger A avec ZZ.
- mettre en X (partie basse) et Y (partie haute) l'adresse de fin de la zone (en fait, l'adresse +1, comme d'habitude).
- appeler $FFD8 : donc

```
SAUV    LDA   #<début
        STA   ZZ
        LDA   #>début
        STA   ZZ +
        LDA   #<ZZ
        LDX   #<fin
        LDY   #>fin
        JSR   $FFD8 ; sauvegarde
```

Mais auparavant, il faut avoir appelé les routines préparatoires $FFBA et $FFBD qui préparent le fichier logique et son nom :

```
PREP    LDX   #numéro logique
        LDA   #périphérique (1 ou 8)
        LDY   #adresse secondaire (0
              pour chargement,1 pour
              sauvegarde)
        JSR   $FFBA    ;prépare fi-
              fichier logique
        LDA   #longueur du nom
        LDX   #< NOM   ;partie basse
              de l'adresse où com-
              mence le nom
        LDY   #>NOM    ;partie haute
        JSR   $FFBD    ;prépare nom
```

Si l'on effectue cette opération depuis l'intérieur d'un programme BASIC, cette dernière séquence peut être remplacée par une instruction OPEN:

OPEN 1, U, X, F$ où U = 1 ou 8 ;
X = 0 pour chargement, 1 pour sauvegarde et F$ est le nom du fichier.

On fait ensuite SYS adresse de SAUV puis CLOSE 1.

De même pour le chargement, on appelle la routine $FFD5 ce qui ne dérange pas les pointeurs BASIC. Elle s'appelle comme suit :
- mettre en A : 0 pour un chargement, 1 pour une vérification
- si le fichier est ouvert avec l'adresse secondaire 0, 1 ou 2, le contenu de X et Y est indifférent : le chargement se fera à l'adresse lue sur le fichier. Si l'adresse secondaire est 3, le chargement se fait à l'adresse spécifiée par X (partie basse) et Y (partie haute).
- appeler $FFD5 donc :

```
CHARG   LDA   #0
        LDX   #<début
        LDY   #>début
        JSR    $FFD5    ;charge
```

Bien sûr, cela doit être précédé par une séquence PREP comme ci-dessus, mais en cas d'appel par BASIC il suffit de la remplacer par un OPEN.

Dans le programme ci-dessous, on charge les séquences SAUV et CHARG en 32768 et suivantes (cas d'un 64, mais l'adaptation au VIC est facile) par des DATA. En 500, sauvegarde, en 600 chargement.

```
400 FOR I = 0 TO 19 : READ A :
     POKE 32768 + I,A = NEXT :
     RETURN
410 DATA 169, 251, 166, 254, 164,
     255, 32, 216, 255, 96
420 DATA 169,0, 166, 254, 164, 255
     32, 213, 255, 96
500 INPUT "CASS (C) OU DISQ (D),
     NOM"; C$, F$ : U = 1
510 IF C$= "D" THEN U = 8 : F$ =
     "0:" + F$
520 OPEN 1, U, 1, F$
530 INPUT "DEBUT, FIN"; D, F
540 DH = INT (D/256) :
     DB = D - 256*DH
550 FH = INT (F/256) :
     FB = F - 256*FH
560 POKE 251, DB : POKE 252, DH :
     POKE 254,FB : POKE 255,FH
570 SYS 32768 : CLOSE 1 : RETURN
600 INPUT "CASS (L) OU DISQ (D),
     NOM"; C$, F$ : U = 1
610 IF C$ = "D" THEN U = 8 :
     F$ = "0 : "+F$
620 OPEN 1, U, 0, F$
630 SYS 32778 : CLOSE 1 : RETURN
```

ADAPTATION

Les routines sont valables pour VIC et 64. Pour les autres CBM, il faut mettre la longueur du nom, le n° logique, l'adresse secondaire, le n° de périphérique en $D1, D2, D3, D4; l'adresse du nom en $DA,DB, l'adresse de début en $FB,FC, l'adresse de fin en $C9,CA et faire JRS $F6A4 (BASIC 4.0 : $F6E3).

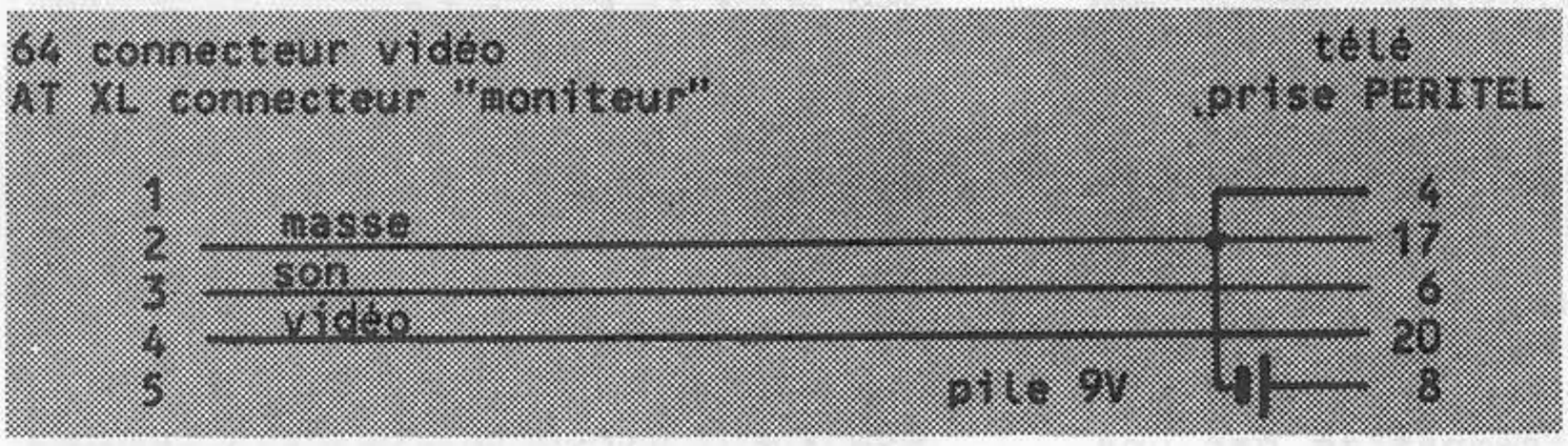
Pour le chargement, c'est JSR $F322 (BASIC 4.0 : $F356).

Daniel-Jean DAVID

Cordon PERITEL pour votre ATARI (et aussi pour votre 64)

Incroyable ! Le schéma de cordon Péritel que nous avions donné dans notre n° 8 (hélas épuisé - que de demandes nous avons eues !), eh bien il est valable aussi pour les ATARI XL. (N.B.: ceci est valable pour les 64 et les ATARI PAL).

Voici donc ce schéma. La connexion de la pile 9V n'est nécessaire que si votre télé n'a pas de position "vidéo" commandée depuis le tableau de bord. Si votre télé est seulement SECAM vous aurez une image en noir et blanc.



Les autre signaux (1 et 5) ne sont pas compatibles entre Atari et 64. Les nouveaux 64, notamment le SX ont une prise DIN 8 broches mais la prise mâle DIN 5 broches y entre et elle est compatible. Voici les brochages:

DIN 5 br / 8 br	64	Atari
1	lumin / sync	luminance
2	masse	masse
3	sortie son	sortie son
4	vidéo composite	vidéo composite
5	entrée son	chroma
6	chroma	n.a.
7	n.c.	n.a.
8	n.c.	n.a.

Pierre Etienne THALBERG

Sauvegarder une cassette sur disque ORIC

L'incompatibilité du "microdisk" avec les cassettes n'est pas son moindre défaut. Comment étendre aux logiciels que vous avez dans votre cassettothèque les avantages du chargement plus rapide (et plus fiable) des disquettes ?

Eh bien il y a une manip a faire. Pas très simple mais pas trop difficile tout de même. Suivez la exactement.

1- Chargez le SED (Système d'Exploitation Disque), donc opération habituelle: connectez l'unité de disque et l'Oric, mettez sous tension, appuyez sur le RESET rouge de l'unite de disque et introduisez la disquette.

2- Vous devez avoir Ready et le curseur. Tapez CALL #E6CA (ATMOS : #E76A)

3- Débranchez la disquette en tirant d'un coup sec le connecteur câble plat derrière l'Oric. Attention, ne mettre hors tension ni la disquette ni l'unité centrale.

4- Vous n'avez plus le curseur. Rétablissez-le en faisant RESET (c'est le faux RESET) en dessous de l'unité centrale. Branchez le magnéto. Chargez le programme voulu par le CLOAD convenable.

5- Tapez à nouveau CALL #E6CA (ATMOS: #E76A). Débranchez le magnéto (en fait, ce débranchement est facultatif).

6- Avec précautions, mais fermeté, remettez le connecteur câble plat de la disquette à l'arrière de l'ORIC.
Faites à nouveau un "faux" RESET: bouton caché en dessous de l'unité centrale. SURTOUT PAS le bouton rouge de l'unité de disques.

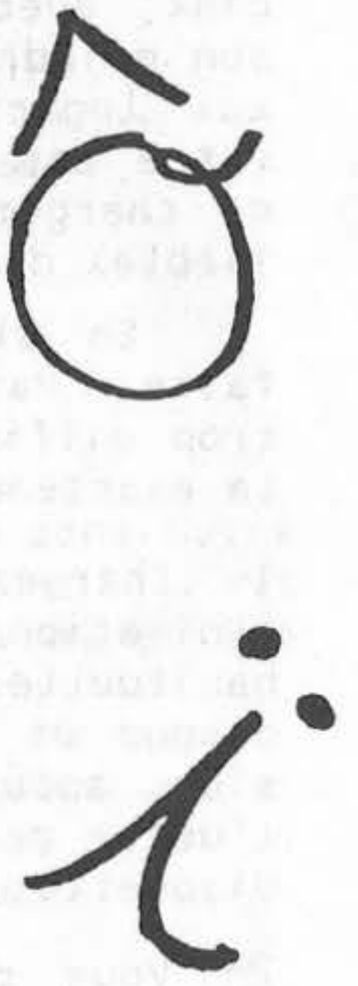
7- Vous êtes prêt maintenant à sauver le programme sur disque par !SAVE "nom".

Et voilà! Cette procédure est à suivre exactement. Il n'y a jamais à mettre hors tension ni l'unité centrale ni le microdisque. Il n'y a qu'un RESET rouge à l'étape 1. Sinon, aux étapes 4 et 6, c'est le RESET en dessous de l'unité centrale.

Emmanuel PUCEVEND
et Honoric de BALSA

Caractères accentués sur VIC

```
10 POKE56,24:POKE54,24:I=34816
20 FOR J=6144 TO 7167
30 POKE J,PEEK(I):I=I+1
40 NEXT J:PRINT "J":POKE36869,254
50 DIM A(127)
60 FOR I=0 TO 127
70 READ A(I):POKE 6992+I,A(I)
80 NEXT I
1000 DATA 48,24,56,4,60,68,58,00
1010 DATA 24,36,56,4,60,68,58,00
1020 DATA 36,00,56,4,60,68,58,00
1030 DATA 12,24,60,66,126,64,60,00
1040 DATA 48,24,60,66,126,64,60,00
1050 DATA 24,36,60,66,126,64,60,00
1060 DATA 36,00,60,66,126,64,60,00
1070 DATA 24,36,24,8,8,8,28,00
1080 DATA 20,00,24,8,8,8,28,00
1090 DATA 24,36,00,60,66,66,60,00
1100 DATA 36,00,60,66,66,66,60,00
1110 DATA 24,36,00,66,66,70,58,00
1120 DATA 36,00,66,66,66,70,58,00
1130 DATA 00,00,60,66,64,60,8,24
1140 DATA 00,00,62,65,65,65,62,00
1150 DATA 00,00,124,130,254,128,124,00
```



Mode d'emploi

Rentrez le programme et exécutez-le: le VIC passe en mode texte. Vous pouvez maintenant effacer le programme par NEW et travailler avec votre VIC comme d'habitude. Veillez simplement à rester en mode texte. Le générateur de caractères que le programme a placé en mémoire vive contient les caractères spéciaux utilisés dans notre langue. Les caractères alphanumériques habituels sont inchangés. Seuls quelques caractères graphiques obtenus par la touche "C=" sont remplacés par ces caractères spéciaux. Vos commentaires n'ont-ils pas meilleure allure ainsi ? Le programme tel qu'il est est sans doute perfectible, mais il a l'avantage de ne consommer aucune place mémoire puisque l'on peut l'effacer. Le générateur de caractères, quant à lui, mobilise à peine plus d'1K octet. Il reste, en effet, 2181 octets sur un VIC sans extension. Les différents caractères disponibles sont :

N.B. C= N signifie la touche "COMMODORE" puis la touche "N"

C= N : à	C= R : î
C= Q : â	C= W : ô
C= D : ä	C= H : ö
C= Z : é	C= J : û
C= S : è	C= L : ü
C= P : ê	C= Y : ç
C= A : ë	C= V : o } les 2
C= E : ï	C= O : e } ensemble formant le oe

Nota : ces nouveaux caractères ne sont pas compris par l'interpréteur.

Sébastien ANDALORO

Des listes plus lisibles

Voici un même court programme - fictif - avec sa liste telle qu' elle apparaît normalement à l'écran et telle qu'elle devient après le traitement proposé. Sans commentaires ! Et la place occupée en mémoire est rigoureusement la même.

```
1000 REM * DEMO *
1010 REM*
1020 PRINT"PROGRAMME FICTIF"
1025 REM*
1030 REM * CALCUL DE LA VITESSE *
1035 REM*
1040 A=SQR(PI)/1.234
1045 PRINTA,A*A
1050 END
```

```
      * DEMO *

1020 PRINT"PROGRAMME FICTIF"

      * CALCUL DE LA VITESSE *

1040 A=SQR(PI)/1.234
1045 PRINTA,A*A
1050 END
```

La transformation est basée sur l'exploitation de la méthode de stockage des programmes dans la mémoire de l'ORIC, décrite dans le numéro précédent, page 36.

Comment faire ? C'est très simple : ajoutez à votre programme les lignes suivantes que vous détruirez ou non après avoir fait
RUN 60000

```
60000 IF PEEK(#505)=157 THEN POKE#506,14
60010 D=DEEK(#501)
60020 REPEAT
60030 IF PEEK(D+4)=157 THEN POKE D+5,14
60040 D=DEEK(D)
60050 UNTIL D=0
```

Si on a lu - et assimilé - l'article précité, le mystère est vite levé. On regarde à chaque ligne si le cinquième octet a pour valeur 157 - code de REM -. Si oui, l'octet suivant est remplacé par 14, code d'effacement de la ligne sur laquelle se trouve le curseur. De ce fait, le numéro de ligne et le mot REM n'apparaîssent pas au listage.

Si REM est seulement suivi d'un unique caractère - * par ex. - on a alors une ligne vide, d'espacement.

Et aucune perturbation n'est apportée au bon déroulement du programme par ce tour de passe-passe.

Quant à l'impression de la liste, elle se fait normalement, tout au moins sur la MCP 40- 4 couleurs. Par contre, sur les GP 50, 80 ou 100 le code 14 est celui qui fait imprimer en double largeur. Il faut donc l'éliminer. Heureusement, on retrouve la liste d'origine avec le même sous-programme dans lequel en fin des lignes 60000 et 60030 on remplace 14 par 42 par ex. (42 = code de *)

Les perfectionnistes pourront faire mieux encore avec à l'écran des remarques d'une couleur différente, donc se détachant encore plus.

Il faut alors modifier les 3 octets qui suivent REM, avec dans l'ordre, 14 (comme ci-dessus), 27 (ESCAPE), xx (code ASCII d'une lettre : A = rouge, par ex.) :
POKE D + 5,14 : POKE D + 6,27 : POKE D + 7,xx .

Enfin, il est naturellement possible, toujours sur le même principe, de faire disparaître au listage tout ou partie de certaines lignes, histoire de rendre ses programmes moins accessibles. D'autres codes que 14, 12 et 30 peuvent être aussi utilisés.

Jean RENAUD

Contrôle des DATA

Les deux articles qui suivent, dans des registres différents, illustrent deux possibilités de contrôle de l'instruction DATA du BASIC.

Chacun des deux utilise un pointeur différent: il n'y en a que deux.

Dans le programme du docteur Jacques SAGLIER, le pointeur du n° de ligne de DATA en cours (adresse décimale 63-64 ; CBM64: même adresse; CBM: 60-61; ORIC: 174-175) est utilisé pour contrôler le volume du prélude n° 1 joué par ce programme. Ceci est réalisé entre autre par la ligne 152 qui utilise une propriété classique du BASIC qui permet d'utiliser une expression logique comme opérande d'une opération arithmétique : -1 est la valeur arithmétique de "vrai", et 0 celle de "faux". Il est bien sûr possible de réaliser ce contrôle, fonction de la position de l'interprétation dans la "partition", par d'autres moyens, cependant ce programme permet de bien voir comment le contrôle d'avancement des DATA peut être fait à l'aide de ce premier pointeur.

Dans l'article de Bernard PETRISOT, c'est le pointeur de l'adresse en cours des DATA - et non le n° de ligne du programme - qui est utilisé et manipulé. Il s'agit en fait de "gammes" (restons dans la musique) autour de la manipulation de ce pointeur.

Daniel TRECOURT

PRELUDE

```
10 :REM:PRELUDE:J.SAGLIER 1983
20 PRINT"■■■■■■■PRELUDE EN UT MAJEUR"
25 PRINT"■■■ DU■PREMIER LIVRE DU ":PRINT"■■■VIC 20 BIEN TEMPERE"
30 PRINT"■■■■■■■■■■■■■■(J.S.BACH)"
32 :
35 :REM:INITIALISATIONS
36 :
37 :
100 V=36878:S1=V-4:S2=V-3:S3=V-2:G=0:M=0:T=100:I=0:J=0:VV=6
102 K=0:L=0:LD=0:DIMA(2)
105 :
106 :
110 DEFFNF(I)=256*PEEK(I)+PEEK(I-1)
112 :
113 :
114 :
150 LD=FNF(64):REM* POINTEUR LIGNE DATA
151 :
152 VV=-(6*(LD<535)+12*(LD>540)+6*(LD>555)+15*(LD=>650))
155 READG:IFG=-1THEN350
160 READM
```

```
170 FORJ=0TO2:READA(J):NEXT
180 FORI=1TO2
190 POKEV,VV:POKES1,G
200 FORJ=1TOT:NEXTJ:POKES1,0
210 POKES2,M
220 :FORJ=1TOT:NEXTJ:POKES2,0
230 :FORJ=1TO2
240 ::FORK=0TO2
250 :::POKES3,A(K)
260 ::::FORL=1TOT:NEXTL
270 :::POKES3,0
280 ::NEXTK
290 :NEXTJ
300 :POKEV,0
310 :
320 NEXTI
330 GOTO150
350 POKEV,15:FORI=1TO2
355 :POKES1,128:FORK=1TOT:NEXT
356 :POKES1,0:POKES2,128:FORK=1TOT:NEXTK
358 :POKES2,0
360 :FORJ=1TO14
365 ::READK:POKES3,K:FORK=1TOT:NEXTK
370 ::T=T+2:NEXTJ
375 NEXTI
380 READG,M,K:POKES1,G:POKES2,M:POKES3,K
385 FORI=1TOT*5:NEXT
390 FORI=15TO0STEP-.05:POKEV,I:NEXT
392 :
395 :END
400 :
500 DATA 222,202,211,222,229
505 DATA 222,196,216,225,230
510 DATA 220,196,211,225,230
515 DATA 222,202,211,222,229
520 DATA 222,202,215,229,235
525 DATA 222,196,208,215,225
530 DATA 220,196,211,225,233
535 DATA 220,189,203,211,222
540 DATA 215,189,203,211,222
545 DATA 196,176,196,208,222
550 DATA 211,185,196,211,220
555 DATA 211,181,202,211,224
560 DATA 205,176,196,215,225
565 DATA 205,172,196,205,220
570 DATA 202,166,189,211,222
575 DATA 202,156,176,189,205
580 DATA 196,156,176,189,205
585 DATA 166,137,166,185,205
590 DATA 189,150,166,189,202
595 DATA 189,166,181,189,202
600 DATA 156,156,176,189,202
605 DATA 161,128,176,189,199
610 DATA 166,143,185,189,199
615 DATA 171,156,185,189,196
620 DATA 166,156,166,185,196
630 DATA 166,150,166,189,202
635 DATA 166,137,166,189,205
640 DATA 166,137,166,185,205
645 DATA 166,143,176,189,208
650 DATA 166,150,166,189,211
655 DATA 166,137,166,189,205
660 DATA 166,137,166,185,205
665 DATA 128,128,166,181,202
670 DATA -1,156,176,189,205,189,176,189,176,156,176,156,137,156,137
675 DATA 211,220,225,230,225,220,225,220,211,220,196,205,202,196
680 DATA 222,222,222
```

Dr Jacques SAGLIER

A PROPOS DE L'INSTRUCTION RESTORE

Le VIC 20 ne possède pas l'instruction RESTORE x (où x est le no de ligne) mais il possede un pointeur vers les zones de DATA en cours. Ce pointeur se trouve aux adresses 65 et 66 (65 octet de poids faible et 66 celui de poids fort).

Sur CBM64, ce pointeur est à la même adresse; CBM: 62-63; ORIC: 176-177.

. PRINT PEEK (65), PEEK (66) nous permet d'obtenir le contenu décimal de chacune de ces adresses (a et b par exemple)

. a + 256 * b nous permet de trouver l'adresse de la mémoire vive qui nous intéresse.

. En modifiant le contenu a et b de ces adresses 65 et 66, nous pouvons donc renvoyer le pointeur des DATA vers une ligne qui nous intéresse, le seul problème étant bien sûr de trouver a et b.

. Le petit programme suivant va illuster de façon très simple la méthode que j'utilise :

```
10  FOR I = 1 TO 6 : READ F :
    PRINT I, F : NEXT
20 PRINT PEEK (65), PEEK (66) ::
30 FOR I = 1 TO 6 : READ F :
    PRINT I, F : NEXT
40 PRINT PEEK (65), PEEK (66) ::
50 FOR I = 1 TO 6 : READ F :
    PRINT I, F : NEXT
100 DATA 1,1,1,1,1,1
110 DATA 2,2,2,2,2,2
120 DATA 3,3,3,3,3,3
```

A l'exécution, on voit :

- une série de 1 (ligne 10)
- 118 et 16 (contenu des adresses 65 et 66)
- une série de 2 (ligne 30)
- 135 et 16
- une série de 3 (ligne 50)

Expliquons maintenant le contenu des adresses 65 et 66

ligne	20	40
adr 65	118(a)	135(a)
adr 66	16(b)	16(b)
adr M.E.V.	4214	4231
contenu de cette M.E.V.	0 (fin ligne 100)	0 (fin ligne 110)

Le zéro de l'adresse 4214 est l'indication de fin de ligne 100 (qui vient d'être lue) tandis que celui de l'adresse 4231 est l'indicateur de fin de ligne 110.

Si je veux donc faire lire 2 fois la ligne 110, je devrai placer le pointeur des "DATA" en fin de ligne 100, ce que je vais faire par

```
40 POKE 65,118 : POKE 66,16
    (19 octets)
40 PRINT PEEK (65), PEEK (66) ::
    (19 octets également)
```

L'utilisation de deux : permet d'avoir le même nombre d'octets dans chaque ligne et d'éviter de nombreux calculs.

A l'exécution, j'obtiendrai une série de 1, une série de 2 puis une nouvelle série de 2.

Je peux aussi modifier la ligne 20 POKE 66,135 : POKE 66,16 et j'obtiendrai une série de 1 puis de 3 et enfin de 2.

Il va de soi :

- que l'apport de toute nouvelle instruction BASIC entraînera un deplacement de l'adresse des "DATA" donc une modification des valeurs a et b,

- que je ne place les instructions PRINT PEEK (65), PEEK (66) :: qu'aux endroits nécessaires.

Bernard PETRISOT

APPEND pour ORIC (Chaînage de 2 programmes Basic)

Le programme ci-dessous charge une routine en langage machine permettant de mettre bout à bout deux programmes BASIC avec chaînage correct des lignes.

Le mode d'emploi à respecter scrupuleusement est le suivant :

- Entrer APPEND - Une erreur de frappe éventuelle sera signalée après lancement par RUN. En faire une copie.

- Charger le deuxième des programmes à joindre, c'est-à-dire celui dont les numéros de lignes sont les plus élevés.

- En mode direct taper CALL #400 et appuyer sur RETURN.

- Charger maintenant le premier des programmes à réunir, qui a donc les numéros de lignes les plus faibles.

- En mode direct taper CALL #427 et appuyer sur RETURN.

Les deux programmes n'en font maintenant plus qu'un, comme on pourra s'en assurer par LIST.

Attention ! Toutes autres opérations que celles décrites sont à prohiber. Surtout pas de RUN entre les chargements.

L'opération peut être renouvelée, la routine en langage machine étant hors zone BASIC. A noter qu' elle est entièrement relogeable.

Enfin, le chaînage de deux programmes avec des numéros de lignes identiques est tout-à-fait possible, mais sans grand intérêt, le deuxième programme n'étant normalement pas accessible.

Jean RENAUD

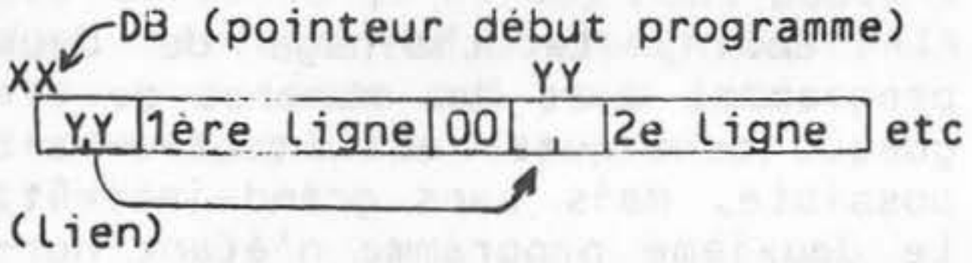
```
100 REM * APPEND *
110 FOR I=#400 TO #489
120 READ A:POKE I,A
130 SC=SC+A:NEXT
140 IF SC<>19041THENPRINT"ERREUR!"
150 END
1000 DATA 165,167,133,130,165,166,133,12
9,160,0,165,129,208,2,198,130
1005 DATA 198,129,177,156,145,129,165,15
6,208,2,198,157,198,156,208,234
1010 DATA 165,157,201,5,208,228,96,160,0
,56,165,129,229,156,165,130
1015 DATA 229,157,144,59,165,156,208,2,1
98,157,198,156,177,156,208,244
1020 DATA 56,165,156,233,1,176,2,198,157
,133,156,177,129,145,156,230
1025 DATA 156,208,2,230,157,230,129,208,
2,230,130,165,167,197,130,208
1030 DATA 234,165,166,197,129,208,228,32
,111,197,198,156,76,58,199,169
1035 DATA 118,160,4,76,237,203,80,65,83,
32,65,83,83,69,90,32
1040 DATA 68,69,32.80,76,65,67,69,33,0
```

Sauvetage

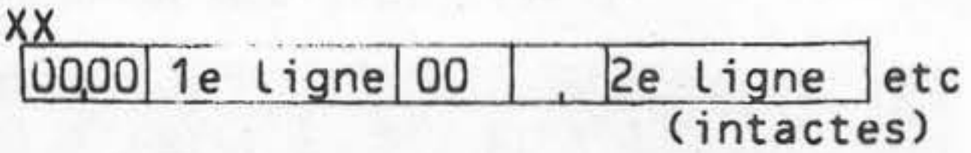
Avez-vous déjà tapé NEW alors qu'en fait vous voulez garder votre programme? Ce petit programme qui exploite la façon dont un programme Basic est rangé en mémoire (voir LA COMMODE n° 9) est fait pour vous.

En fait, lorsque vous faites NEW, votre programme n'est pas détruit. Seul le lien de la première ligne est remis à zéro, ce qui assure que le système croit que le programme n'existe plus.

Ex:



Après NEW, on a :



Pour annuler l'effet d'un NEW, il suffit donc de rétablir l'adresse YY en XX, XX+1.

Mais YY n'est autre que l'adresse de la case qui suit le 00 qui marque la fin de la 1ère ligne. D'où un programme très simple : on examine les octets successifs de la ligne jusqu'à trouver 0. A ce moment, le contenu de l'index donne la partie basse de l'adresse cherchée. La partie haute de YY est égale à la partie haute de XX car la ligne fait moins de 80 caractères; elle est en DB+1.

```
     DB = $2B ; 64 et VIC.C'est $9A
          sur ORIC
     *  = $C000 ; sur 64. Sur VIC
          mettre $033C, sur ORIC
          $0400
     LDY  #$03   ;  pour  commencer
          après le n° de ligne
BCLE INY
     LDA (DB),Y; octet de la ligne
     BNE BCLE
     INY
     INY  ; fabrique l'adresse YY
     TYA  ; partie basse en A
     LDY =$00
     STA (DB),Y ; met la partie
          basse en XX
     INY
     LDA DB,Y    ; la partie haute
          de XX
     STA (DB),Y ; est en DB+1
     RTS
```

Il suffit alors de faire SYS 49152 (64) / 828 (VIC) ou CALL #400 (ORIC) pour récupérer votre programme.

Daniel-Jean DAVID

Lisez La Commode

Répétition automatique

Je ne puis laisser passer sans réagir la réponse faite à J.C. RAIMBAULT: Il est parfaitement possible d'assurer la compatibilité de l'EDEX avec une Répétition Automatique des touches : à preuve le programme suivant que j'utilise quotidiennement (ou presque !) précisément sur 4032 petit écran :

Suppression par SYS 948
Remise en route après un SAVE
sur disquette par SYS 936
La ligne 130 actionne l'EDEX
La ligne 180 actionne REPKEY
et efface le programme BASIC
Le programme L.M est dans le tampon 2éme cassette, dans les zones d'adresses libres (les dix 0 de la ligne 320 se sont révélés nécessaires par expérience !)
Le DOS SUPPORT est compatible avec l'ensemble, à condition d'être chargé en premier.

J. PIERRAT

NDLR : Monsieur J.C. RAIMBAULT nous a envoyé un programme semblable .

```
100 REM J.PIERRAT VERSAILLES 21/06/82
110 REM:PROGR.[REPKEY 4000]
120 REM ==================
130 PRINTCHR$(147): SYS40960:REM EDEX
140 PRINT:PRINT:PRINTSPC(13)"REPKEY 4000"
150 REM ADAPTE D'UN ARTICLE  DE    L'     ORDINATEUR INDIVIDUEL N. 15 MARS 80
160 REM PROGRAMME DANS TAMPON 2EME  CASSETTE ,DE 898 A 958,UTILISE AUSSI 897.
170 D=898:REM=130,3 ADRESSE DE DEPART
180 READB:IFB<0THEN SYS936:NEW:END
190 POKED,B:D=D+1:GOTO180
200 DATA165,151  :REM LDA ABS 151
210 DATA201,255:REM CMP 255
220 DATA240,16:REM BEQ CLR
230 DATA238,129,03:REM INC CPT 897
240 DATA173,129,03:REM LDA ABS CPT
250 DATA201,15:REM CMP 15
260 DATA208, 9:REM BNE OUT
270 DATA169,255:REM REP)LDA 255
280 DATA133,151   :REM STA ABS 151
290 DATA169,08:REM LDA 8
300 DATA141,129,03:REM CLR)STA ABS CPT
310 DATA76, 79,228:REM OUT)JMP 58447
320 DATA 0,0,0,0,0,0,0,0,0,0
330 DATA 120:REM SEI
340 DATA169,130:REM LDAIM 130
350 DATA133,144:REM STA(0),144
360 DATA169,03:REM LDAIM 03
370 DATA133,145:REM STA(0),145 $0382=898=3$256+144
380 DATA88:REM CLI
390 DATA96:REM RTS
400 DATA234:REM NOP
410 DATA120:REM SEI
420 DATA169,79:REM LDAIM 79
430 DATA133,144:REM STA(0),144
440 DATA169,228:REM LDAIM 228
450 DATA133,145:REM STA(0) 145:RETABLIT VECTEUR INTERRUP.$E44F=228$256+79
460 DATA88:REM CLI
470 DATA96:REM RTS
480 DATA-1
490 REM
500 REM POUR UTILISER REPKEY:SYS936
510 REM
520 REM POUR ANNULER :SYS 948
530 REM RESPECTER L'ORDRE DE CHARGEMENT: DOS SUPORT,EDEX,REPKEY
```

Utilisation de deux magnétos avec un VIC

Vous vous souvenez de l'esquisse de manip que nous donnions dans le n° 4. (Ce numéro est épuisé, mais il existe un service photocopie pour les abonnés). Eh bien, ces idées marchent comme nous l'ont prouvé quelques lettres de lecteurs. Les seuls problèmes ont été dus au fait que, lorsqu'on fait le CLOSE du magnéto en écriture, ce magnéto doit être activé : en effet ce CLOSE produit éventuellement une écriture sur le magnéto.

Deux lecteurs nous ont envoyé des compléments sur cette manip, qui, soit dit en passant, est applicable aussi au 64. M. ROUSSET nous fournit un système complet de mise à jour de fichier. M. BUTLER nous envoie une routine en langage machine qui active l'un ou l'autre magnéto en fonction de la touche F1 ou F3 et affiche sur l'écran quel magnéto est actif.

Transfert et correction de données avec deux magnétos pour VIC 20

Point de départ :

Article paru dans LA COMMODE" n° 4, p.41-44. Il a été conservé la partie électronique et électro-mécanique ainsi que certaines données relatives à l'activation des magnétophones, des zones mémoires, etc...

Principe :

- Lire la bande magnétique du magnétophone n°1 (les données s'inscrivent dans la zone tampon du magnétophone : de 828 à 1019)

- Copie de ce tampon T7168 dans une zone mémoire (protégée par POKE 56,28) de 7168 à 7359.

- Remise à zéro du tampon magnétophone

- Copie du tampon T7168 dans un tableau d'où la possibilité de modifier très aisément le contenu de chaque case de ce tableau

- Ecriture des données du tableau sur la bande magnétique du magnétophone n° 2

- Retour au magnétophone n° 1 pour lecture d'un nouveau tampon de données, etc..,

En réalité bien plus complexe, car, par GET, il faut saisir les données caractère par caractère ; il faut des "retour chariot" (CHR$(13)) à la bonne place. Les magnétophones ont tendance à tourner n'importe quand d'où la redondance des ordres imposés par POKE pour les pointeurs, les arrêts ou démarrages moteur, les commandes de lecture ou d'écriture de bande. En outre, il faut prévoir des temporisations car le relais électromécanique est infiniment plus lent que les électrons dans le microprocesseur (même commandé en BASIC).

Les données :

Elles doivent être "formatées". L'unité de base du transfert est le tampon magnéto soit 192 octets (en réalité, seulement 191 seront disponibles car en 828 il n'y a pas de donnée : essayez PEEK (828). En outre, l'octet 192 devra être un retour chariot imposé de manière explicite sinon il y aura concatenation avec le tampon suivant.Il faut donc partager les 192 octets en un nombre de lignes de

192/nL caractères (l'exemple choisi comporte 12 lignes : 11 de 16 caractères et la dernière de 15) mais, à l'écran, cela ne se remarque pas si l'on fait :

```
Lignes 1 à 11 :
 donnée de 14 caractères + CHR$(32)
           + CHR$(13) ;  soit: 176
Ligne 12 : donnée de 14 caractères
           + CHR$(13) ;  soit:  15

           au total: 176 + 15 = 191
```

Le formatage doit être programmé et bien protégé de la malfaçon (surtout le dépassement en nombre de caractères et la mauvaise place des "retour chariot") .

Exemple.

```
-----------------------------------
lignes  colonnes
-----------------------------------
        1  2  3        14  15  16

1       A  A  A ..... A (32)(13)
2       B  B  B ..... B (32)(13)
"       "  "  "       "  "   "
"       "  "  "       "  "   "
"       "  "  "       "  "   "
"       "  "  "       "  "   "
"       "  "  "       "  "   "
11      P  P  P ..... P (32)(13)
12      Q  Q  Q ..... Q (13)
-----------------------------------
```

Le programme a été testé sur plusieurs tampons successifs puis la bande magnétique du magnétophone 2 a été, à son tour, mise en lecture, modifiée et recopiée à nouveau sans difficulté (la lecture se fait aussi bien par INPUT que par GET). A partir de ce squelette de programme, on peut envisager toutes les opérations de création, modification, ajout, recherche par mot-clé, etc.., en séquentiel, bien sûr. Evidemment, ce n'est pas de la T.G.V et il sera bon de prévoir un signal sonore pour avertir l'opérateur plus ou moins engourdi !!!

André ROUSSET

Adaptation aux extensions mémoire

Comme la mémoire disponible d'un VIC 20 en configuration de base est trop juste pour accueillir un programme complet de manipulation de fichier, l'utilisation avec des extensions est à prévoir. Voici les adaptations proposées :

VIC de base ou extension 3K:

```
Réservation mémoire:  POKE 56,29
début tampon       :  7424
fin tampon         :  7615
pointeur           :  7616
```

VIC + extension 8K:

```
Réservation mémoire:  POKE 56,63
début tampon       :  16 128
fin tampon         :  16 319
pointeur           :  16 320
```

VIC + extension 16K:

```
Réservation mémoire:  POKE 56,127
début tampon       :  32 512
fin tampon         :  32 703
pointeur           :  32 704
```

```
 10    PRINT "CLR" : POKE 56,28 : CLR
 20    POKE 37138,PEEK(37138) OR 32
 30    GOSUB 1100 : POKE 7552,0
 40    DIM A $ (16,12)
 50    GOSUB 1300 : GOSUB 1400 : GOSUB 1200 : PRINT
 60    OPEN 2,1,1,"xxx2"
 70    GOSUB 1100 : PRINT
 80    OPEN 1,1,0,"xxx1"
 90    POKE 192,0 : POKE 63680,169
100    GET # 1, A$
110    IF PEEK (166) = 190 THEN 140
120    IF A$ = "$" THEN 140
```

```
130     GOTO 100
140     PRINT : PRINT
150     GOSUB 1500
160     POKE 166,0
170     GOSUB 1300
180     POKE 64776,173 : FOR I = 1 TO 500 : NEXT
190     GOSUB 1200 : PRINT
200     I = 0
210     IF PEEK (7168) = 13 THEN I = I + 1
220     POKE 829,13
230     FOR L = 1 TO 12
240     FOR C = 1 TO 16
250     A$(C,L) = CHR$(PEEK(7168 + I))
260     PRINT A$(C,L);
270     IF A$(C,L) = "$" THEN PRINT #2, A$(C,L) : GOTO450
280     I = I + 1 : NEXT C
290     PRINT
300     B$ = "" : C$ = ""
310     PRINT "CORRECTION O/N";: INPUT C$ : IF C$ <> "O" THEN 380
320     PRINT "NOUVELLE DONNEE" : INPUT B$ : IF LEN(B$) > 15 THEN 320
330     B$ = B$ + CHR$ (13)
340     FOR C= 1 TO 16
350     A$ (C,L) = MID$(B$,C,1)
360     NEXT C
370     PRINT
380     FOR C = 1 TO 16
390     IF PEEK(166)=191 THEN A$(C,L)=CHR$(13):PRINT#2,A$(C,L):POKE63715,32:GOTO 430
400     PRINT #2, A$(C,L);
410     NEXT C
420     NEXT L
430     GOSUB 1300 : GOSUB 1400 : GOSUB 1100 : POKE 166,190 : PRINT
440     GOTO 90
450     CLOSE 2 : GOSUB 1100 : CLOSE 1 : PRINT "TERMINE" : END
1100    POKE 37136,PEEK(37136)OR32
1110    RETURN
1200    POKE 37136,PEEK(37136)AND223
1210    RETURN
1300    FOR I = 828 TO 1019
1310    POKE I,0
1320    NEXT I
1330    RETURN
1400    POKE 7552,0
1410    FOR I = 7168 TO 7359
1420    POKE I,0
1430    NEXT I
1440    RETURN
1500    POKE 7552,PEEK(166)
1510    FOR I = 0 TO 191
1520    POKE 7168 + I, PEEK(829 + I)
1530    NEXT I
1540    POKE 7359,13
1550    RETURN
```

André ROUSSET
TOULOUSE

Programme de choix magneto par les touches de fonction

Ce programme a été fait sur un VIC de base, il serait préférable de l'utiliser sur un VIC de même configuration. Il apparaît en haut à droite de l'écran le numéro du magnéto qui est activé.

La touche F1 active le magnéto 1 et la touche F3 active le magnéto 2

Après RUN/STOP + RESTORE, les touches n'agissent plus, il suffit de faire SYS 6578 pour qu'elles soient de nouveau actives.

Olivier BUTLER

Adresse début = 6400 = $1900

Variable A = PEEK (251)
+ PEEK (252) * 256

Variable B = PEEK (253)
= PEEK (254) * 256

```
., 1900 LDA $CB
., 1902 CMP #$27
., 1904 BEQ $190D
., 1906 CMP #$2F
., 1908 BEQ $1948
., 190A JMP $1980
., 190D LDA $9110
., 1910 ORA #$20
., 1912 STA $9110
., 1915 LDA $A6
., 1917 STA $1D81
., 191A LDA $1D80
., 191D STA $A6
., 191F LDA #$3C
., 1921 STA $FB
., 1923 LDA #$03
., 1925 STA $FC
., 1927 LDA #$C0
., 1929 STA $FD
., 192B LDA #$1C
., 192D STA $FE
., 192F JSR $19A4
., 1932 LDA #$00
., 1934 STA $FB
., 1936 LDA #$1C
., 1938 STA $FC
., 193A LDA #$3C
., 193C STA $FD
., 193E LDA #$03
., 1940 STA $FE
., 1942 JSR $19A4
., 1945 JMP $1980
., 1948 LDA $9110
., 194B AND #$DF
., 194D STA $9110
., 1950 STA $A6
., 1952 STA $1D80
., 1955 LDA $1D81
., 1958 STA $A6
., 195A LDA #$3C
., 195C STA $FB
., 195E LDA #$03
., 1960 STA $FC
., 1962 LDA #$00
., 1964 STA $FD
., 1966 LDA #$1C
., 1968 STA $FE
., 196A JSR $19A4
., 196D LDA #$C0
., 196F STA $FB
., 1971 LDA #$1C
., 1973 STA $FC
., 1975 LDA #$3C
., 1977 STA $FD
., 1979 LDA #$03
., 197B STA $FE
., 197D JSR $19A4
., 1980 NOP

., 1981 LDA $FB
., 1983 CMP #$00
., 1985 BEQ $198E
., 1987 LDA #$B2
., 1989 STA $02FF
., 198C JMP $1994
., 198F LDA #$B1
., 1991 STA $02FF
., 1994 LDA $02FF
., 1997 STA $1E15
., 199A LDA #$06
., 199C STA $9615
., 199F JMP $EABF
., 19A2 BRK
., 19A3 BRK
., 19A4 LDY #$00
., 19A6 LDA ($FB),Y
., 19A8 STA ($FD),Y
., 19AA INY
., 19AB CPY #$C0
., 19AD BNE $19A6
., 19AF RTS
., 19B0 BRK
., 19B1 BRK
., 19B2 SEI
., 19B3 LDA #$00
., 19B5 LDX #$19
., 19B7 STA $0314
., 19BA STX $0315
., 19BD CLI
., 19BE LDA #$19
., 19C0 STA $38
., 19C2 LDA $9112
., 19C5 ORA #$20
., 19C7 STA $9112
., 19CA LDA #$00
., 19CC STA $1D80
., 19CF STA $1D81
., 19D2 RTS
., 19D3 ADC $F7EF,Y
```

Quelques trucs pour CBM 3000-EDEX

Mettre EDEX en sommeil

En règle générale, EDEX est une aide appréciable pour les programmes. Mais parfois il devient néfaste, soit à cause de la répétition rapide des touches, soit à cause des mots-clés abrégés par aX (Tentez de mettre un a dans un INPUT), soit enfin de sa gestion dictatoriale de I.R.Q. dont votre propre programme en code machine a besion lui aussi.

Voici un petit programme qui permet par un simple SYS de mettre EDEX en sommeil, puis par un autre de le réveiller à volonté, tout en restant dans l'état initial choisi (pas à pas, pas de AUTO,...).

La seule restriction est de ne pas toucher au bloc-note d'EDEX qui s'étend de $3D8 à $3FB.

Mode d'emploi

* Charger éventuellement le DOS,
* Activer EDEX de manière usuelle par SYS46000,
* Pour le mettre en sommeil SYS 850, et si à ce moment vous êtes sous DOS, les ordres DOS refonctionnent normalement avec @.
* Pour réveiller EDEX: SYS900, l'écran n'est pas modifié et DOS est intact.

Le programme ne comporte volontairement que des sauts relatifs, il est donc translatable en mémoire.

ON-OFF EDEX

```
033A            1 !SUPPRIME PROVISOIREMENT EDEX
033A            2 ! APPEL PAR SYS 850
033A            3 ! TOTALEMENT TRANSLATABLE
033A            4 ! PEUT ETRE MIS EN ROM
0090            5       IRQ = $90
0070            6       CHAR = $70
03D8            7       DOS = $3D8 ! IMAGE DOS
0352            8 * = 850
0352 78         9       SEI ! PRUDENCE
0353 A92E      10       LDA #$2E
0355 8590      11       STA $90
0357 A9E6      12       LDA #$E6
0359 8591      13       STA IRQ + 1
035B 38        14       SEC !REMET CHARGOT
035C ADD803    15       LDA DOS
035F E917      16       SBC #$17 !RETOURNE AU DEBUT
0361 8571      17       STA CHAR + 1
0363 ADD903    18       LDA DOS + 1
0366 C9B0      19       CMP #$B0 ! SI PAS DE DOS ACTIF ON A UN SAUT DANS EDEX
0368 D00E      20       BNE SUIT
036A A9E6      21       LDA #$E6
036C 8570      22       STA CHAR ! PAS DE DOS REMET NORMAL
036E A977      23       LDA #$77
0370 8571      24       STA CHAR + 1
0372 A9D0      25       LDA #$D0
0374 8572      26       STA CHAR + 2
0376 D004      27       BNE FIN ! SAUT INCONDITIONNEL
0378 E9FF      28 SUIT  SBC #$FF ! CAR EN BINAIRE NON SIGNE
037A 8572      29       STA CHAR + 2
```

```
037C 58       30 FIN  CLI
037D 60       31      RTS
037E          32 !****** RELANCE EDEX******
037E          33 ! APPEL PAR SYS 900
0384          34 * = 900
0384 78       35      SEI
0385 A94C     36      LDA #$4C  ! ON RECONSTRUIT
0387 8570     37      STA CHAR
0389 A945     38      LDA #$45  ! POINT D'ENTREE D'EDEX
038B 8571     39      STA CHAR + 1
038D A9B0     40      LDA #$B0
038F 8572     41      STA CHAR + 2
0391 20B1BA   42      JSR $BAB1 ! CONTIENT SEI A LA FIN
0394 60       43      RTS
0395          44 .END
```

```
50000 POKE1,82:POKE2,3:REM-------------------PROG.MACH.  ON  EDEX(SYS 850)
50010 DATA120,169,46,133,144,169,230,133,145,56,173,216,3,233,23,133
50020 DATA113,173,217,3,201,176,208,14,169,230,133,112,169,119,133,113
50030 DATA169,208,133,114,208,4,233,255,133,114,88,96.
50040 FORI=850TO893:READA:POKEI,A:NEXT:RETURN
```

```
60000 POKE1,132:POKE2,3:REM-------------------PROG.MACH.OFF  EDEX(SYS 900)
60010 DATA120,169,76,133,112,169,69,133,113,169,176,133,114,32,177,186,96
60020 FORI=900TO916:READA:POKEI,A:NEXT:RETURN
```

```
0350 57 F7 78 A9 2E 85 90 A9
0358 E6 85 91 38 AD D8 03 E9
0360 17 85 71 AD D9 03 C9 B0
0368 D0 0E A9 E6 85 70 A9 77
0370 85 71 A9 D0 85 72 D0 04
0378 E9 FF 85 72 58 60 E7 77
0380 1C 9E C1 9E 78 A9 4C 85
0388 70 A9 45 85 71 A9 B0 85
0390 72 20 B1 BA 60 CA CB 98
```

EDEX et DOS

Les utilisateurs de l'unité de disque utilisent très souvent un programme "DOS SUPPORT" qu'il est très agréable d'avoir comme premier programme sur disque pour l'appeler par LOAD "*",8:RUN

Mais avec EDEX, il faut penser ensuite à faire SYS 46000 pour conserver les commandes disques et bénéficier d'EDEX.

Pourquoi ne pas tout faire en une fois ?

Tenter de modifier à l'écran le programme de DOS décale tout le programme du DOS qui ne fonctionne plus. Voici une méthode (valable pour BASIC 2 - PET2001 et CBM3000).

* SYS 1024: accès au moniteur
* M 0400-0410
* Modifier les octets de 0401 à 0416
* S"N-DOS",08,0400,06B0

Vous disposez maintenant d'un nouveau moniteur N-DOS qui met automatiquement EDEX en service, et qui, en prime tient moins de place sur disque (seulement 3 secteurs).

EDEX DOS

```
033A             1 !CREE UNE NOUVELLE EN TETE AU DOS POUR EDEX AUTOMATIQUE
0400             2 * = $400
0400 00          3 .BYTE $0
0401 1404        4 .WORD FIN
0403 0000        5 .WORD $0
0405 9E          6 .BYTE $9E ! SYS DOS
0406 313633      7 .TEXT '1639'
040A 3A          8 .TEXT ':'
040B 9E          9 .BYTE $9E ! SYS EDEX
040C 343630     10 .TEXT '46000'
0411 3A         11 .TEXT ':'
0412 A2         12 .BYTE $A2  ! NEW
0413 00         13 .BYTE $0
0414 0000       14 FIN .WORD $0
0416            15 .END

B*
     PC  IRQ  SR AC XR YR SP
.;   0401 E62E 32 04 5E 00 F4
.
.:   0400 00 14 04 00 00 9E 31 36
.:   0408 33 39 3A 9E 34 36 30 30
.:   0410 30 3A A2 00 00 00 00 00

 0  SYS1639:SYS46000:NEW
```

Encore un "truc" : lecture des cassettes du VIC 20 ou du CBM 64 **sans POKE !**

Si vous avez un CBM avec EDEX, il vous suffit de faire APPEND pour pouvoir lire et lister toute cassette en BASIC d'un CBM 64 ou d'un VIC 20. Pas de POKE, et pas de perte de mémoire vive.

J.M. DECONINCK

ORIC A BRAC

LM + Basic : *encore plus simple !*

A condition que la partie en langage machine de votre programme BASIC soit logée entre les adresses #400 à #4FF, il suffit à l'enregistrement de faire:

CSAVE"PROG",A#400

C'est tout !

Au chargement, rien de plus que le CLOAD"" habituel suffit, et le programme se lance normalement par RUN.

Explication: Au chargement le pointeur de début de BASIC (#501) n'est pas modifié quelle que soit l'adresse de début d'enregistrement.

Remargue: Si un tiers recopie le programme par un CSAVE normal, la partie en langage machine ne sera naturellement pas sauvegardée.

Jean RENAUD

Copie haute-résolution de l'écran VIC 20

Configuration

Ce programme réalise la copie de l'écran en haute résolution obtenue avec la cartouche Super-Expander en graphic 2 ou 3.

Il a été mis au point avec une Seikosha GP-100VC. Il doit aussi fonctionner avec les 1515 ou 1525.

L'imprimante doit avoir comme n° de périphérique le n° 4.
LA ROUTINE SE CHARGE D'OUVRIR LE FICHIER IMPRIMANTE ET DE LA METTRE EN MODE GRAPHIQUE: inutile donc d'"envoyer" OPEN4,4:PRINT 4,CHR$(8)

La routine est implantable dans les 3K invisibles pour le VIC20 étendu de 8 ou 16 K.

La routine devra être relogée pour un VIC 20 de base.

Quelques remarques générales

En mode graphique 2 ou 3 avec Super-Expander l'écran est organisé en 20 colonnes de 160 octets, soit une résolution de 160x160=256000 pixels.

La mémoire écran occupe donc (160x160)/8 soit 3200 octets implantables de 4096 ($1000) à 7296.

L'octet destiné à l'imprimante en mode graphique se compose de 7 bits significatifs, le 8ème bit (de poids fort) est forcé à 1 pour que l'octet soit reconnu par l'imprimante comme caractère graphique.

Pour faire une copie de l'écran il suffit donc de prendre les octets de la MEV d'écran 7 par 7, puis avec un masque on forme autant d'octets imprimante (dont le bit de poids fort est toujours à 1). Ainsi en terme de bilan mémoire, 7 octets "écran" donnent lieu à la création de 8 octets "imprimante".

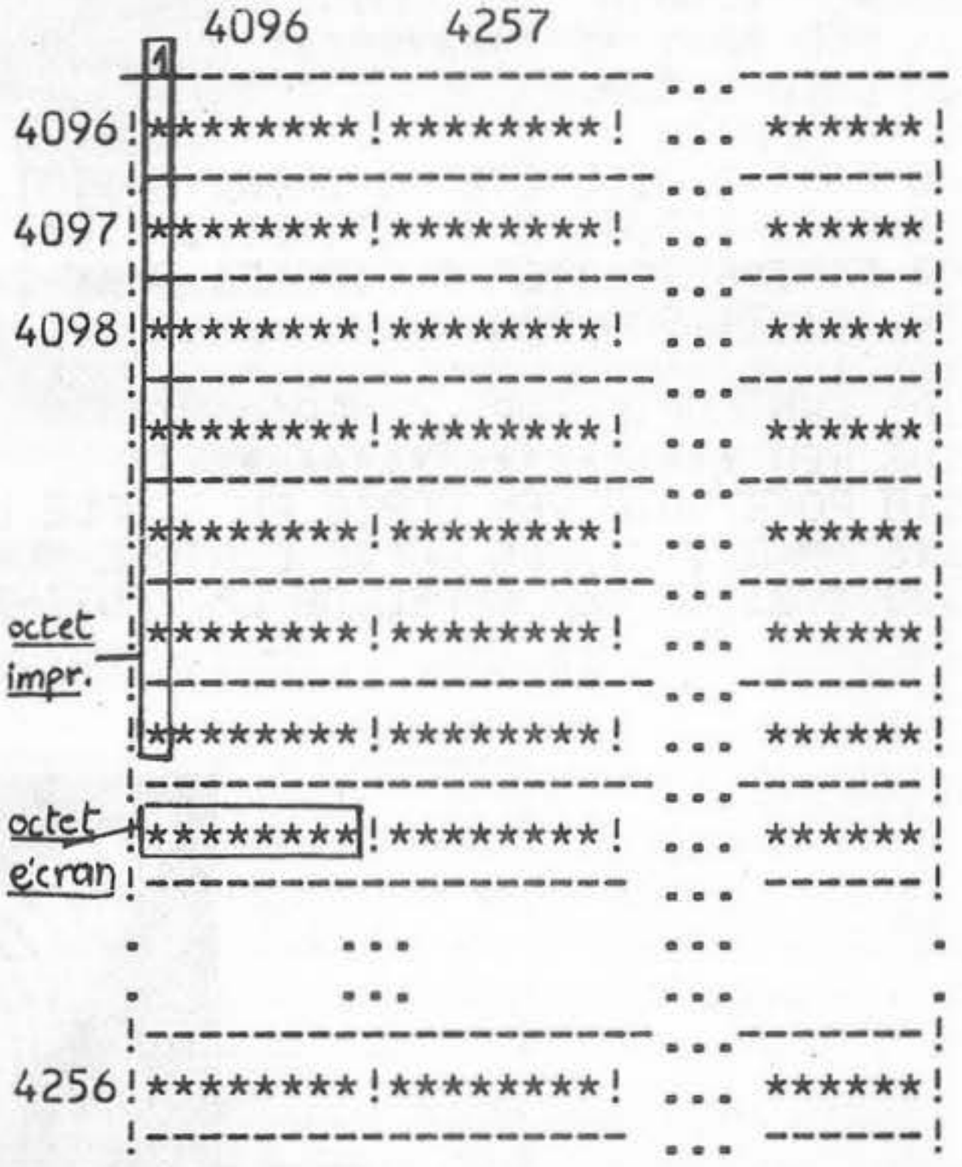


Schéma de l'organisation de la mémoire d'écran en graphic 2 ou 3

Mode d'emploi

Cette routine a été implantée dans les 3K de Super-Expander qui ne peuvent servir au BASIC avec les cartouches de 8 ou 16 K.

ATTENTION! on peut la reloger ailleurs à condition de recalculer les JMP et autres adresses, mais pas dans le tampon cassette car cette routine l'utilise.

Registres de communication

La copie peut être exécutée en simple largeur ou en double largeur suivant le contenu du registre 780.

La copie peut être faite à gauche de la feuille ou centrée suivant le contenu du registre 781.

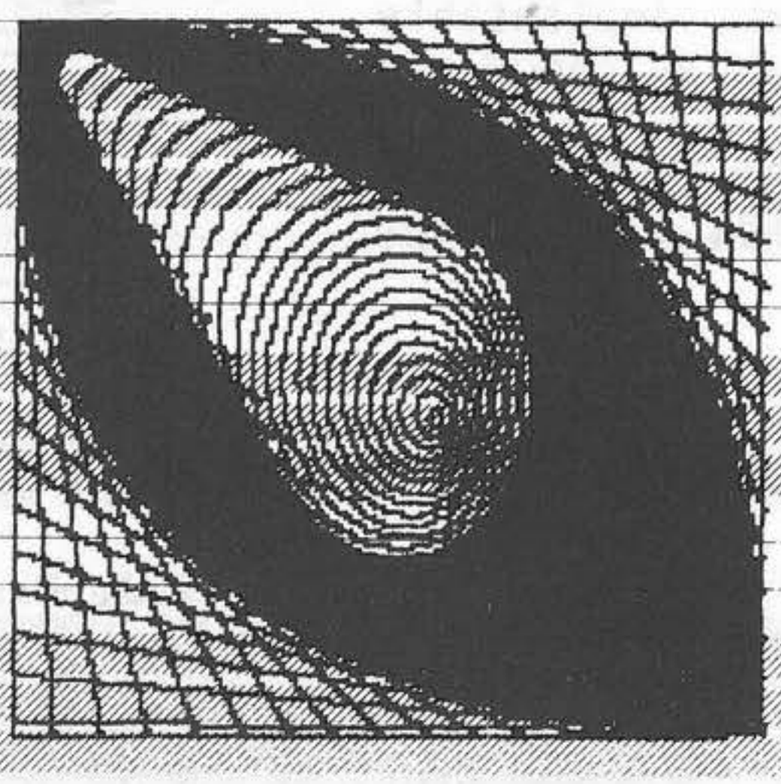
POKE 780,1 fait imprimer en double largeur.

POKE 781,1 fait centrer la copie sur la feuille.

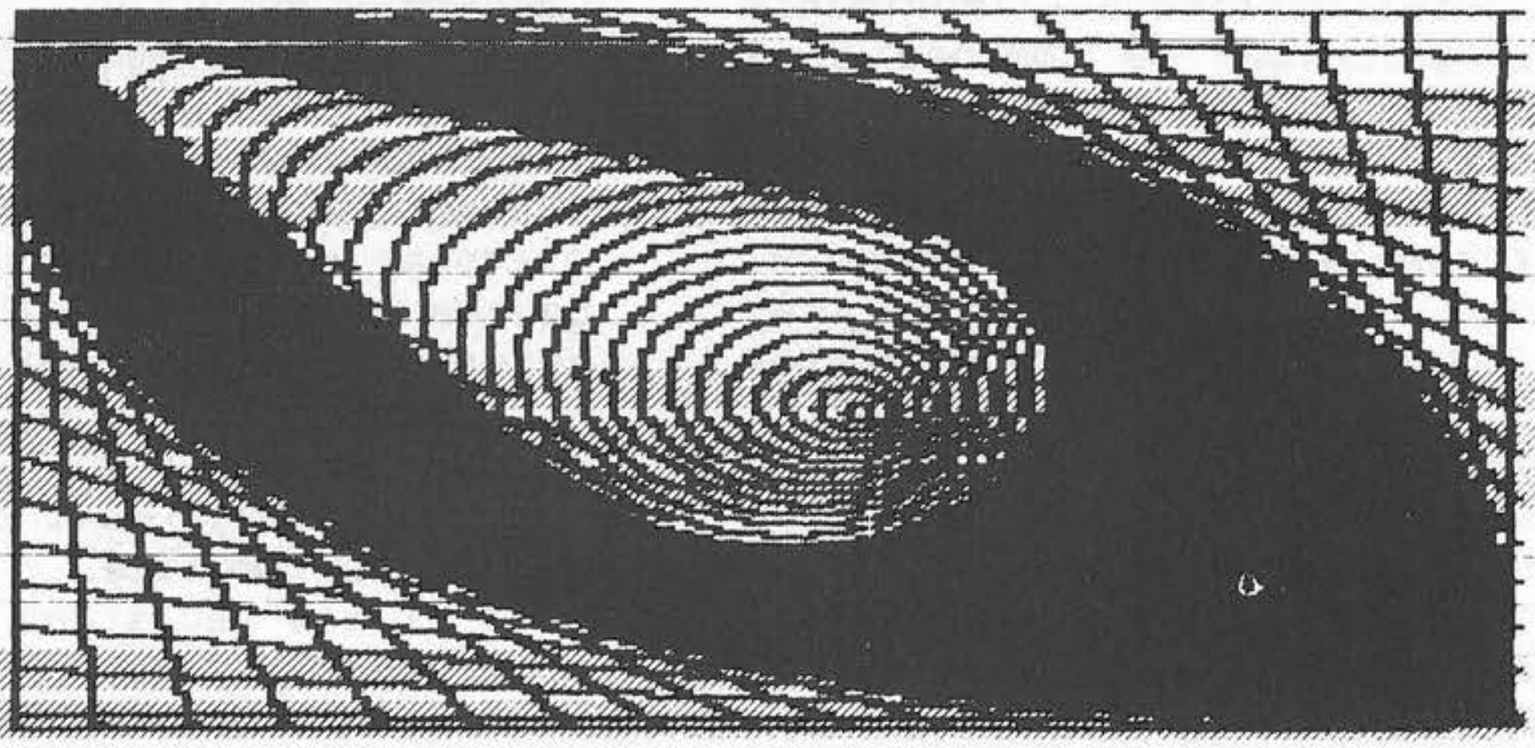
Si les registres 780 et 781 ne sont pas préparés la copie se fera en simple largeur et à gauche.

Programme de démonstration et copie

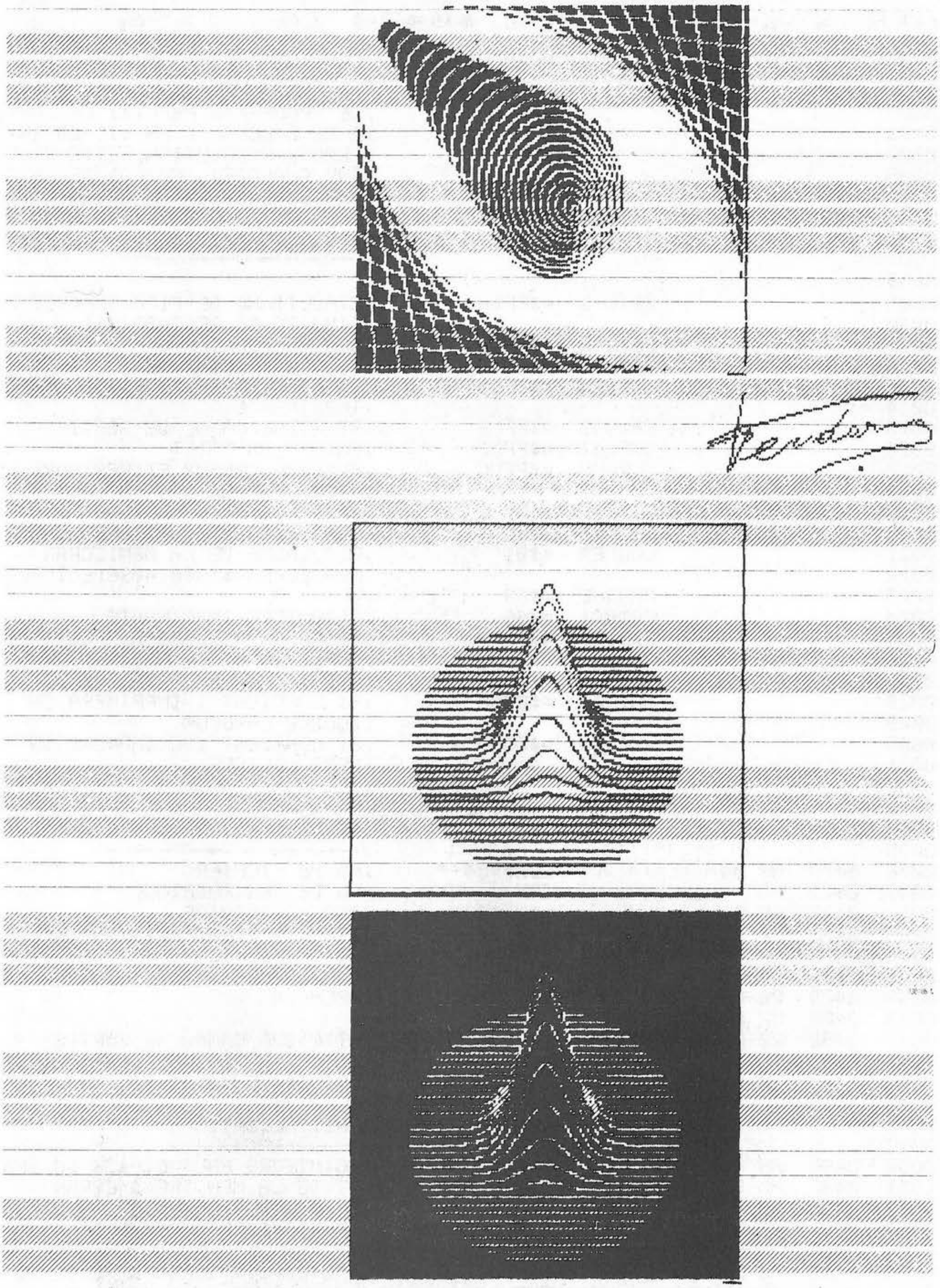
```
10 REM IMP DEMO
20 REM **********
22 ;
25 REM LA ROUTINE EN LANGAGE MACHINE DE COPIE D'ECRAN A ETE CHARGEE
27 :
30 REM LA CARTOUCHE SUPER-EXPANDER EST NECESSAIRE
32 :
33 REM DESSIN QUELCONQUE
34 REM ********************
35 COLOR1,0,0,0
37 GRAPHIC3
40 FORI=0TO1023STEP63:DRAW1,0,ITOI,1023:DRAW1,I,0TO1023,I:NEXT
50 FORI=1TO64STEP5:CIRCLE1,8*I,8*I,3*I,4*I,28,88:NEXT
60 FORI=63TO13STEP-5:CIRCLE1,3*64-I,3*64-I,3*I,4*I:NEXT
70 PAINT1,900,900
100 ;
105 REM COPIE SUR L'IMPRIMANTE
106 REM ***********************
110 POKE780,0:REM COPIE EN SIMPLE LARGEUR
115 POKE781,1:REM COPIE CENTREE SUR LA FEUILLE DE L'IMPRIMANTE
120 SYS11136:REM APPEL DE LA ROUTINE DE COPIE 130END
```



Le même en double largeur soit pour les lignes 110 et 115 : POKE 780,1 : POKE 781,1



Le tout agrémenté d'une routine d'inversion vidéo



HARD-COPY SUR VIC 20

ADRESSE DEBUT HEX. $0470 DEC 1136

```
LIGN  ADR.  CODE HEX  ETIQU.  MNEMONIQUE      COMMENTAIRES
0000                                          ;CE PROGRAMME REALISE LA COPI
0001                                          ;E DE L'ECRAN D'UN VIC 20 EN
0002                                          ; HAUTE RESOLUTION.CETTE ROUT
0003                                          ;INE SUPPOSE L'EMPLOI DE LA C
0004                                          ;ARTOUCHE SUPER-EXPANDER.
0005                                          ;
0006                                          ;TABLE DES SYMBOLES
0007                                          ;-----------------
0008                                          ;
0009                          SETLFS  =$FFBA  ;ETABLIT N0 DE FICH.ADRESSE P
0010                                          ;RIMAIRE ET SECONDAIRE
0011                          SETNAM  =$FFBD  ;NOM DU FICHIER(0 PAR DEFAUT
0012                                          ;POUR L'IMPRIMANTE
0013                           OPEN   =$FFC0  ;OUVRE LE FICHIER EN SORTIE
0014                                          ;IMPRIMANTE
0015                          CHKOUT  =$FFC9  ;ATTRIBUE CANAL DE SORTIE
0016                          CHROUT  =$FFD2  ;ENVOIE UN OCTET
0017                          CLRCHN  =$FFCC  ;REFERME CANAUX ET RESTAURE E
0018                                          ;CRAN/CLAVIER
0019                                          ;
0020                           HIMEM  =$02    ;ADRESSE HAUTE DE LA MEM.EC
0021                          LOWMEM  =$01    ;ADR.BASSE DE LA MEM.ECRAN
0022                                          ;MEM.ECRAN=$1000(4096DECIMAL)
0023                          MASQUE  =$69    ;
0024                          CARACT  =$6A    ;CARACTERE IMPRIMANTE
0025                          ADDOCT  =$6B    ;VALEUR A AJOUTER A L'OCTET A
0026                                          ;IMPRIMER EN FORMATION
0027                          CASSET  =$033C  ;DEBUT TAMPON CASSETTE
0028                           A.REG  =$030C  ;SI CONTIENT 1 IMPRIMERA EN
0029                                          ;DOUBLE LARGEUR
0030                          %.REG   =$030D  ;SI CONTIENT 1 L'IMPRESSION
0031                                          ;SERA CENTREE
0032                                          ;
0033                                          ;
0034                                          ;PREPARE L'IMPRIMANTE
0035                                          ;--------------------
0036  0470  A9 04             LDA#$04        ;N0 DE FICHIER
0037  0472  AA                TAX            ;N0 DE PERIPHERIQUE
0038  0473  A0 00             LDY#$00        ;ADR.SECONDAIRE
0039  0475  20 BA FF          JSR SETLFS     ;ETABLIT FICH.PERIPH.AD.SECON
0040  0478  A9 00             LDA#$00        ;N0 DE FICHIER:0 PAR DEFAUT P
0041  047A  20 BD FF          JSR SETNAM     ;OUR L'IMPRIMANTE
0042  047D  20 C0 FF          JSR OPEN       ;OPEN4,4
0043  0480  A2 04             LDX#$04        ;
0044  0482  20 C9 FF          JSR CHKOUT     ;ATTRIBUE CANAL DE SORTIE
0045  0485  A9 08             LDA#$08        ;MET L'IMP.EN MODE GRAPHIQUE
0046  0487  20 D2 FF          JSR CHROUT     ;PRINT#4,CHR$(8)
0047                                          ;
0048                                          ;INITIALISATIONS
0049                                          ;--------------
0050  048A  A9 10             LDA#$10        ;POINTEURS HAUT ET BAS DU DEB
0051  048C  85 02             STA HIMEM      ;UT DE LA MEMOIRE D'ECRAN
0052  048E  A9 00             LDA#$00        ;EN HAUTE RESOLUTION AVEC SUP
0053  0490  85 01             STA LOWMEM     ;-EXP.($1000OU4096DEC)
0054  0492  D8                CLD            ;
0055  0493  A5 02       LIGSUI LDA HIMEM     ;SAUVE POINTEURS MEM.EC.DANS
0056  0495  48                PHA            ;LA PILE
0057  0496  A5 01             LDA LOWMEM     ;
0058  0498  48                PHA            ;
```

```
0059  0499  A2 A0           LDX#$A0      ;LONGUEUR D'UNE LIGNE(160 DEC
0060  049B  A9 80  RECOMM   LDA#$80      ;MASQUE=80HEX
0061  049D  85 69           STA MASQUE   ;
0062  049F  A0 00  SUITOC   LDY#$00      ;POINTEUR DE BITS
0063  04A1  A9 00           LDA#$00      ;
0064  04A3  85 6A           STA CARACT   ;CARACTERE IMPRIMANTE
0065                                     ;
0066                                     ;TEST FIN DE COPIE
0067                                     ;-----------------
0068  04A5  A9 01           LDA#$01      ;SI CONTENU DE LOWMEM=1 ALORS
0069  04A7  C5 01           CMP LOWMEM   ;FIN DE LA MEM. ECRAN
0070  04A9  D0 06           BNE PROG     ;FIN?
0071  04AB  20 CC FF        JSR CLRCHN   ;OUI.FERME FICHIERS
0072  04AE  68              PLA          ;
0073  04AF  68              PLA          ;
0074  04B0  60              RTS          ;RETOUR DE SYS
0075                                     ;
0076                                     ;CREE LE CARACTERE A IMPRIMER
0077                                     ;-----------------------------
0078  04B1  A9 01     PROG   LDA#$01     ;INITIALISE VALEUR A AJOUTER
0079  04B3  85 6B            STA ADDOCT   ;POUR FORMER LE CARACT. A IMP
0080  04B5  B1 01     OCTSUI LDA(LOMEM),Y ;CHARGE OCTET ECRAN
0081  04B7  25 69           AND MASQUE   ;
0082  04B9  F0 07           BEQ DECALA   ;BIT NUL?SAUT A DECALE MASQUE
0083  04BB  18              CLC          ;
0084  04BC  A5 6B           LDA ADDOCT   ;SI BIT =1 AJOUTE POIDS DE CE
0085  04BE  65 6A           ADC CARACT   ;BIT AU CARACTERE EN FORMATIO
0086  04C0  85 6A           STA CARACT   ;N DE L'IMP.
0087  04C2  06 6B     DECALA ASL ADDOCT   ;DECALE POIDS DU BIT
0088  04C4  C8              INY          ;
0089  04C5  C0 07           CPY#$07      ;DERNIER BIT?
0090  04C7  D0 EC           BNE OCTSUI   ;NON;CONTINUE
0091  04C9  A5 6A           LDA CARACT   ;CARACT.GRAPHIQUE IMPRIMANTE
0092  04CB  09 80           ORA#$80      ;BIT DE POIDS FORT FORCE A1
0093  04CD  9D 3C 03        STA CASSET,X ;RANGE DANS TAMPON CASSETTE
0094  04D0  CA              DEX          ;DERNIER OCTET DE LA LIGN
0095  04D1  F0 14           BEQ IMPRIM   ;OUI;IMPRIME LIGNE
0096  04D3  46 69           LSR MASQUE   ;DECALE MASQUE
0097  04D5  A5 69           LDA MASQUE   ;MASQUE NUL?
0098  04D7  D0 C6           BNE SUITOC   ;NON.CONTINUER CREATION CARAC
0099  04D9  18              CLC          ;
0100  04DA  A5 01           LDA LOWMEM   ;INITIALISE POSITION DES 7
0101  04DC  69 A0           ADC#$A0      ;OCTETS ECRAN SUIVANTS
0102  04DE  90 02           BCC SUITE    ;
0103  04E0  E6 02           INC HIMEM    ;CHANGE PAGE ECRAN
0104  04E2  85 01     SUITE STA LOWMEM   ;
0105  04E4  4C 9B 04        JMP RECOMM   ;CREATION CARACT IMPRIM.SUIV.
0106                                     ;
0107                                     ;IMPRIME LIGNE
0108                                     ;-------------
0109  04E7  AD 0D 03  IMPRIM LDA X.REG    ;SI X.REG=1 CENTRER LA COPIE
0110  04E4  C9 01           CMP#$01      ;
0111  04EC  D0 11           BNE MISPAG   ;
0112  04EE  A2 50           LDX#$50      ;TABULATION DE 80 COL.EL.IM
0113  04F0  AD 0C 03        LDA A.REG    ;SI A.REG=1 COPIE EN DOUBLE L
0114  04F3  D0 02           BNE SAUT     ;A.REG=0;COPIE SIMPLE LARGEUR
0115  04F5  A2 A0           LDX#$A0      ;TAB 160 COLONNES ELEMENTAIRE
0116  04F7  A9 80     SAUT  LDA#$80      ;
0117  04F9  20 D2 FF SUIVAN JSR CHROUT    ;IMPRIME $80 COLONNES EL.BLAN
0118  04FC  CA              DEX          ;
0119  04FD  D0 FA           BNE SUIVAN   ;FIN DE TAB?NON;RECOMMENCE
0120                                     ;IMPRIME
0121  04FF  A2 A0     MISPAG LDX#$A0     ;NOMBRE DE COLONNES ELEMENT.
0122  0501  BD 3C     AUTRE  LDA CASSET,X ;INDEX DANS TAMPON CASSETTE
0123  0504  20 D2 FF        JSR CHROUT   ;IMPRIME OCTET
```

```
0124  0507  A8                    TAY             ;SAUVE A
0125  0508  AD 0C 03              LDA A.REG       ;SI A.REG=1IMPRIME DOUBLE LAR
0126  050B  C9 01                 CMP#$01         ;GEUR.
0127  050D  D0 04                 BNE SIMPLE      ;SINON SIMPLE LARGEUR
0128  050F  98                    TAY             ;RESTAURE A
0129  0510  20 D2 FF              JSR CHROUT      ;IMPRIME DE NOUVEAU A
0130  0513  CA          SIMPLE    DEX             ;DERNIER OCTET?
0131  0514  D0 EB                 BNE AUTRE       ;NON,IMPRIME SUIVANT
0132  0516  18                    CLC             ;
0133  0517  A9 0A                 LDA#$0A         ;
0134  0519  20 D2 FF              JSR CHROUT      ;ALIMENTATION LIGNE
0135                                              ;
0136                                              ;INIT.LIGNE ECRAN SUIVANTE
0137                                              ;
0138  051C  18                    CLC             ;
0139  051D  68                    PLA             ;
0140  051E  69 07                 ADC#$07         ;NOUVELLE LIGNE ECRAN DE 7 OC
0141  0520  85 01                 STA LOWMEM      ;
0142  0522  68                    PLA             ;
0143  0523  85 02                 STA HIMEM       ;RESTAURE POINTEUR MEM ECRAN
0144  0525  4C 93 04              JMP LIGSUI      ;CREATION NOUVELLE LIGNE A IM
0145                                              ;FIN
0146                                              ;
```

Vidage mémoire de $0470 à $0528

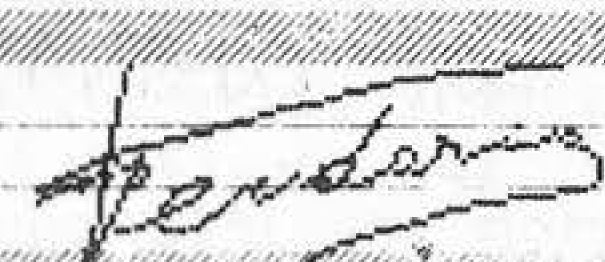
```
B*
    PC   SR AC XR YR SP
.;603E 73 00 63 00 F6
*

.:0470 A9 04 AA A0 00
.:0475 20 BA FF A9 00
.:047A 20 BD FF 20 C0
.:047F FF A2 04 20 C9
.:0484 FF A9 08 20 D2
.:0489 FF A9 10 85 02
.:048E A9 00 85 01 D8
.:0493 A5 02 48 A5 01
.:0498 48 A2 A0 A9 80
.:049D 85 69 A0 00 A9
.:04A2 00 85 6A A9 01
.:04A7 C5 01 D0 06 20
.:04AC CC FF 68 68 60
.:04B1 A9 01 85 6B B1
.:04B6 01 25 69 F0 07
.:04BB 18 A5 6B 65 6A
.:04C0 85 6A 85 6B C8
.:04C5 C0 07 D0 EC A5
.:04CA 6A 09 80 9D 3C
.:04CF 03 CA F0 14 46
.:04D4 69 A5 69 D0 C6
.:04D9 12 A5 01 69 A0
.:04DE 90 02 E6 02 85
.:04E3 01 4C 9B 04 AD
.:04E8 0D 03 C9 01 D0
.:04ED 11 A2 50 AD 0C
.:04F2 03 D0 02 A2 A0
.:04F7 A9 50 20 D2 FF
.:04FC CA D0 FA A2 A0
.:0501 BD 3C 03 20 D2
.:0506 FF A8 AD 0C 03
.:050B C9 01 D0 04 98
.:0510 20 D2 FF CA D0
.:0515 EB 18 A9 0A 20
.:051A D2 FF 18 98 69
.:051F 07 85 01 68 85
.:0524 02 4C 93 04 EA
```

.CHARGEUR BASIC POUR CETTE ROUTINE

```
10000 REM CHARGEUR BASIC POUR COPIE ECRAN GRAPHIQUE DU VIC 20
10010 REM*************************************************
10020 D=1136:REM ADRESSE DU DEBUT DE LA ROUTINE
10022 REM ROUTINE IMPLANTEE DANS LES 3K DE SUPER-EXPANDER "INVISIBLES"POUR
      LE VI C20+16K
10024 S=0:REM SOMME DE CONTROLE
10025 FORI=0TO184
10030 READ A:S=S+A
10035 POKED+I,A:NEXT
10040 IFS<>21071THENPRINT"[illegible]ERREUR DANS LES DATAS[illegible]"
10045 END
10100 DATA 169,4,170,160,0,32,186,255,169,0
10101 DATA 32,189,255,32,192,255,162,4,32,201
10102 DATA 255,169,8,32,210,255,169,16,133,2
10103 DATA 169,0,133,1,216,165,2,72,165,1
10104 DATA 72,162,160,169,128,133,105,160,0,169
10105 DATA 0,133,106,169,1,197,1,208,6,32
10106 DATA 204,255,104,104,96,169,1,133,107,177
10107 DATA 1,37,105,240,7,24,165,107,101,106
10108 DATA 133,106,6,107,200,192,7,208,236,165
10109 DATA 106,9,128,157,60,3,202,240,20,70
10110 DATA 105,165,105,208,198,24,165,1,105,160
10111 DATA 144,2,230,2,133,1,76,155,4,173
10112 DATA 13,3,201,1,208,17,162,80,173,12
10113 DATA 3,208,2,162,160,169,128,32,210,255
10114 DATA 202,208,250,162,160,189,60,3,32,210
10115 DATA 255,168,173,12,3,201,1,208,4,152
10116 DATA 32,210,255,202,208,235,24,169,10,32
10117 DATA 210,255,24,104,105,7,133,1,104,133
10118 DATA 2,76,147,4,

READY.
```



Guy PENDARIES

Display list et modes d'affichage de l'ATARI

Le 6502 de l'Atari est secondé pour les tâches d'affichage par deux boîtiers, le GTIA et ANTIC. Le GTIA gère l'interface télé proprement dit, alors que ANTIC organise l'image télé. C'est lui qui a donc le rôle le plus intelligent et, de fait, c'est un véritable microprocesseur.

La Display List

ANTIC étant un microprocesseur, il passe son temps à exécuter un programme qui réside en mémoire. Ce programme s'appelle la DISPLAY LIST, ce qu'on pourrait traduire par "programme d'affichage". L'adresse du premier octet de la Display List est contenue dans le couple 560(bas),561(haut) soit $230-231. D'où DL=PEEK(560)+256*PEEK(561). On a aussi DL=PEEK(741)+PEEK(742)+1.

La Display List est une suite d'octets qui forment des instructions d'affichage. La plupart des instructions tiennent en 1 octet. Seules quelques unes prennent trois octets car elles contiennent une adresse. Le jeu d'instructions est en fait très élémentaire (bien plus que celui du 6502). Toute instruction spécifie comment sera fait l'affichage sur un certain nombre de lignes élémentaires du balayage télé. L'instruction vaut pour toute la largeur de l'écran.

Il y a trois sortes d'instructions: définition de mode, extinction et saut.

* Les instructions d'extinction sont les plus simples:

code hexa	dec	effet
0	0	laisse en noir 1 ligne élémentaire
10	16	" 2 "
20	32	" 3 "
30	48	" 4 "
40	64	" 5 "
50	80	" 6 "
60	96	" 7 "
70	112	" 8 "

Il faut un certain nombre de telles instructions en début de Display List pour être sûr que l'affichage ne soit pas trop haut sur l'écran en fonction des variations d'une télé à l'autre.

* Les instructions de mode ont leur code de 2 à 15.

hexa	dec	mode Basic	graph. car.	couleurs	lignes élém.	points élem.	colonnes /ligne	octets /ligne	octets /écran	RAM utilisée
2	2	0	C	1,5	8	8	40	40	960	993
3	3	-	C	1,5	10	8	40	40	760	-
4	4	-	C	4	8	8	40	40	960	-
5	5	-	C	4	16	8	40	40	480	-
6	6	1	C	5	8	16	20	20	480	513
7	7	2	C	5	16	16	20	20	240	261
8	8	3	G	4	8	8	40	10	240	273
9	9	4	G	2	4	4	80	10	480	537
A	10	5	G	4	4	4	80	20	960	1017
B	11	6	G	2	2	2	160	20	1920	2025
C	12	-	G	2	1	2	160	20	3840	-
D	13	7	G	4	2	2	160	40	3840	3945
E	14	-	G	4	1	2	160	40	7680	-
F	15	8	G	1,5	1	1	320	40	7680	7891

On voit immédiatement qu'il y a des modes ANTIC qui né correspondent pas à des modes BASIC (obtenus par GRAPHICS G). Le seul moyen de les obtenir sera de manipuler la Display List.

L'instruction de code 2 par exemple signifie: allumer les 8 lignes de balayage où on en est selon le mode Basic 0. On a 40 colonnes. Pour chaque colonne, comme il s'agit d'un mode caractères, on lit le code caractère dans la mémoire d'écran et on cherche le motif binaire à allumer dans le générateur de caractères.

Si le code était 15, on aurait à allumer 1 ligne élémentaire selon le mode Basic graphique 8. Chaque octet de la mémoire d'écran déterminerait ici l'allumage de 8 points élémentaires.

Exemple: Le tableau ci-dessous représente la Display List en mode Basic 0. (On l'obtient par un programme BASIC très simple dont l'essentiel est FOR I=DL TO DL+... :LPRINT PEEK(I): NEXT I. Le programme que nous avons utilisé est un peu perfectionné puisqu'on imprime trois valeurs par ligne). On a imprimé en outre l'adresse de début de display list, l'adresse mémoire d'écran et celle de la fenêtre (texte dans les modes graphiques).

```
MODE      0
DL39368   SC40000   FE40800
0         112       112       112
3         66        64        156
6         2         2         2
9         2         2         2
12        2         2         2
15        2         2         2
18        2         2         2
21        2         2         2
24        2         2         2
27        2         2         65
30        32        156       0
33        0         128       0
36        0         0         0
```

Les trois 112 font laisser 24 lignes élémentaires noires. Les 2 font l'affichage suivant le schéma ci-dessous.

```
112  ------------>  8 lignes noires
112                 8 lignes noires
112                 8 lignes noires
66  64  156         8 lignes en mode 0 + définition
                      adresse mémoire écran
2    ------------>  8 lignes en mode 0
2    ------------>  8 lignes en mode 0
.                     .
.                     .
.                     .
2    ------------>  8 lignes en mode 0
65  32  156         saut au début de la Display List
```

Relation Display List-écran

Le 66 est une instruction 2 modifiée. Lorsqu'à une instruction de mode, on ajoute :

. 16 ($10) permet le scrolling horizontal fin à partir de la position modifiée. Ceci est très difficile à utiliser.

. 32 ($20) permet le déroulement vertical fin du nombre de lignes élémentaires spécifiées par POKE en 54277.

. 64 ($40) transforme l'instruction en définition de l'adresse d'écran. C'est le cas du 66 en 4ème octet de l'exemple. Les deux octets suivants sont l'adresse d'écran dans l'ordre bas,haut. On retrouve que 64+256*156=40000=adresse d'écran.

. 128 ($80) ajoute la possibilité de "Display List interrupt" : lorsqu'on arrive à cet octet de la Display List, il y a une interruption. Si la routine d'interruption change les paramètres d'affichage, on peut avoir un écran mixte (voir plus loin).

* Les instructions de saut sont :

```
  1 ($1)   saut pur et simple
 65 ($41)  saut avec attente du retour du spot en début d'image.
```

Les deux octets suivants forment l'adresse à laquelle on saute.

La dernière instruction de la Display List est un tel saut avec attente. L'adresse où on saute est 32+256*156 = 39968 c'est à dire le début de la Display List. Ce saut signifie tout simplement qu'en fin d'écran, ANTIC boucle pour parcourir la Display List en vue du prochain balayage écran.

Examinons un second exemple, celui du mode Basic 2.

```
MODE      2
DL40536   SC40560   FE40800
0         112       112       112
3         71        112       158
6         7         7         7
9         7         7         7
```

12	7	7	7
15	66	96	159
18	2	2	2
21	65	88	158

On a d'abord les trois 112 habituels. Les octets 7 sont ceux du mode. Le 71 n'est autre que 7+64 c'est la définition d'adresse d'écran. Là l'adresse est 112 + 256 * 158 = 40560. On a ensuite 66 et trois 2. C'est la définition de la fenêtre en mode 0 qu'on a en bas de l'écran. Le début de la mémoire correspondant à cette fenêtre est 96 + 256 * 159 = 40800, c'est bien l'adresse FE. On a enfin le bouclage (65) à l'adresse 88 + 256 * 158 = 40536.

On remarque qu'on a moins de 7 qu'on n'avait de 2 dans le mode GRAPHICS 0. Comptons les lignes élémentaires ainsi formées:

Pour GRAPHICS 0, on a 24 instructions 2 (23 + le 66) qui couvrent chacune 8 lignes élémentaires soit 24 * 2 = 192 lignes élémentaires.

Pour GRAPHICS 2, on a 10 instructions 7 (9 + le 71) qui couvrent chacune 16 lignes élémentaires soit 160 plus 4 instructions 2 (3 + le 66) qui couvrent 32 lignes élémentaires. La somme est encore 192.

De fait, lorsqu'on fait une Display List il faut que le nombre de lignes élementaires ne dépasse pas trop 192.

Applications de la Display List

La Display List permet des affichages spectaculaires, par différents effets. Essayons d'abord trois exemples simples en mode direct.

Faites GRAPHICS 0 et DL = PEEK(560) + 256 * PEEK(561)
puis POKE DL+17,156: POKE DL+16,64: POKE DL+15,66
(On rajoute une instruction de définition de la mémoire d'écran au milieu de la Display List, qui d'ailleurs reprend l'adresse de début d'écran qu'on avait.) Cela produit deux fois le même affichage sur l'écran. Spectaculaire, non ?

Faites RESET, puis POKE DL+17,156: POKE DL+16,32: POKE DL+15,65
(On met le bouclage de la Display List au milieu de l'écran) Cela revient à avoir l'écran éclairé beaucoup plus petit que d'habitude.

Essayons maintenant le scrolling vertical fin. Ecrivez depuis le haut de l'écran:

```
LIGNE 1
LIGNE 2
LIGNE 3
LIGNE 4
LIGNE 5
```

(sans faire RETURN qui créerait des messages d'erreur : allez à la ligne par mouvements de curseur).

Faites alors POKE 54277,5 et POKE DL+7,34. Il ne reste plus que 1/4 de LIGNE 3.

Mélanges de modes

Un gros intérêt des Display Lists est qu'on peut mélanger plusieurs modes sur l'écran alors que, bien sûr, l'instruction GRAPHICS 0 n'en génère qu'un.

En fait il y a un tel mélange dans les modes habituels avec une fenêtre en bas de l'écran.

Il y a deux éléments sur lesquels veiller :

1- assurer la somme des lignes élémentaires voisine de 192

2- savoir comment écrire dans les différentes zones d'écran, car on est en dehors des modes Basic classiques.

Exemple: Nous allons tracer un dessin en mode graphique 8 (ANTIC 15) et au dessus mettre un grós titre d'une ligne (mode 2, ANTIC 7) et un sous-titre sur deux lignes (mode 0, ANTIC 2). Nous avons donc 4 zones horizontales dans le dessin : une zone en mode 2, une en mode 0, la zone graphique en mode 8, et la fenêtre texte. Partons de la Display List du mode 8 (nous notons les adresses sous la forme cccc (bb,hh) avec cccc = bb + 256 * hh.

```
 32822        112        112        112
(54,128)
 32825         79         80        129   -----> mém. ecran = 33104

               15         15         15

                .
                .
                .

               15         15         15
```

Il faut remplacer les 32 lignes élémentaires les plus hautes par les modes 8 et 2. On va donc remplacer le 79 et les 31 premiers 15. Au point de vue mémoire d'écran, on se ramène à 33104 + 32 * 40 = 34384 (80,134). Comme on veut garder intacte la fin de la Display List, le 79 80 134 sera en 32856 (88,128) et suivantes. Notre haut de Display List est donc maintenant :

```
 32822        112        112        112

 32825         79         80        129

               15         15         15

                .
                .
                .

 32856         79         80        134

               15         15         15
```

Mettons juste avant 32856 les octets de définition des deux zones du haut. Nous pouvons prendre 33104 (80,129) pour mémoire d'écran de la première et 33144 (120,129) pour la seconde d'où :

```
 32849         71         80        129
(81,128)

               66        120        129

                2         79         80

              134         15         15
```

La seule chose à faire est maintenant de mettre un saut en 32825 qui envoie le contrôle en 32849. On pourrait aussi faire POKE 560,81:POKE 561,128. Pour faire tous les POKE voulus, on arrangera les données en DATA.

Il reste maintenant à écrire dans les deux zones de titre. Pour cela, on trompe le système : par POKE en 87 on lui fait croire que le mode graphique est celui de la zone considérée. On impose l'adresse de mémoire d'écran par POKE en 88 et 89 et il suffit d'imprimer par PRINT#6. Sinon, on peut toujours faire des POKE en mémoire d'écran. Le programme ci-dessous produit un tel affichage 4 zones.

```
10 GRAPHICS 8:DL=PEEK(560)+256*PEEK(561)
20 COLOR 1:PI=3.14159265
30 PLOT 0,100
40 FOR X=0 TO 313
50 Y=100+50*SIN(PI*X/40)
60 DRAWTO X,Y
70 NEXT X
80 ? "          SINUSOIDE"
100 FOR I=0 TO 2
110 READ X:POKE DL+3+I,X:NEXT I
120 FOR I=0 TO 9
130 READ X:POKE DL+27+I,X:NEXT I
150 POKE 88,80:POKE 89,129:POKE 87,2
160 POSITION 2,0
170 ? #6,"COURBE"
250 POKE 88,120:POKE 89,129:POKE 87,0
260 POSITION 0,0
270 ? #6," TRACEE EN MODES"
280 ? #6,"    MELANGES    "
1000 DATA 73,80,129,71,80,123,66,120,129
,2,73,80,134
```

De 10 à 80 on crée l'affichage de la courbe et de la fenêtre.
De 100 à 130 on modifie la Display List.
De 150 à 170 on écrit le titre en grosses lettres.
De 250 à 280 on écrit le sous-titre.

Une autre possibilité de la Display List est l'animation en BASIC. Il suffit de deux POKE (en DL+4 et DL+5) pour changer l'adresse de la mémoire d'écran utilisée par ANTIC. Il suffit alors de préparer plusieurs images dans des zones mémoires différentes (pensez au POKE 87 ... pour la préparation) puis de mettre leurs adresses en DL+4 et DL+5 successivement.

Une autre possibilité que nous n'étudions pas ici faute de place est celle de créer les modes que BASIC ne peut créer alors qu'ANTIC les possède : on met directement les valeurs dans la Display List. Un tel mode intéressant est le mode ANTIC 3 qui affiche les caractères en grande hauteur ce qui permet des jambages pour les minuscules.

Display List Interrupt

On peut à chaque instruction de la Display List créer une interruption. Il faut :

. ajouter 128 aux instructions de la Display List voulues. Par exemple, en GR.

0, si l'on veut qu'il se passe quelque chose au milieu de l'écran, on affectera l'octet d'adresse DL+16.

. créer en langage machine une routine de traitement. Dans notre exemple, la routine change la couleur du fond de l'écran (registre $D018 en mode 0). On aura donc un fond de deux couleurs, mais on pourrait en avoir bien plus.

. mettre en 512-513 ($200,201) l'adresse de début de la routine qu'on vient de créer.

. faire POKE 54286,192 pour autoriser les interruptions.

Nous implantons en $600 la routine suivante :

code (décimal)			instruction:		
72			PHA		; sauve A
169	70		LDA	#couleur	; rose
141	10	212	STA	WSYNC	; adresse $D40A nécessaire à la synchro
141	24	208	STA	$D018	; registre couleur
104			PLA		; restaure A
64			RTI		; retour

D'où le programme à essayer:

```
10 DL=PEEK(560)+256*PEEK(561)
20 POKE DL+16,130
30 FOR I=0 TO 10:READ A:POKE 1536+I,A:NE
XT I
40 POKE 512,0:POKE 513,6
50 POKE 54286,192
100 DATA 72,169,70,141,10,212,141,24,208
,104,64
```

Ceci offre d'immenses possibilités d'obtenir simultanément sur l'écran plus de couleurs qu'on ne le croyait faisable.

Ce programme va nous permettre aussi de clarifier la notion de registres fantômes. Pourquoi le haut de l'écran reste-t-il bleu ? Parce qu'au retour du spot le contenu du registre fantôme associé à $D018 y est recopié. Remplacez 24,208 (ligne 100) par 198,02 : là le rose devient permanent.

Rita A.

Facteurs premiers

Pour résoudre ce problème, dont l'exécution n'est pas nécessairement instantanée, voici un exemple de programmation en assembleur 6502, et une comparaison des performances d'un programme BASIC équivalent.

"Facteurs Premiers" marche sur tous les micro-ordinateurs équipés d'un 6502. Le listing d'assemblage a été obtenu avec l'assembleur de LA COMMODE.

La décomposition de nombres entiers en facteurs premiers :

Chacun sait en quoi consiste la décomposition d'un nombre entier en facteurs premiers, cependant, il peut être utile de le rappeler, ainsi que la façon d'obtenir cette décomposition.

Un nombre premier ne peut être divisé (division entière ou euclidienne, et reste nul) que par lui-même et par 1. Un nombre non premier est ainsi le produit de plusieurs nombres premiers. Par exemple : 168 = 2x2x2x3x7 .

On simplifie cette écriture en écrivant $168 = 2^3 \times 3 \times 7$ c'est-à-dire en n'écrivant chaque facteur premier qu'une fois dans la décomposition et en faisant figurer son exposant différent de 1. Dans l'exemple,le facteur 2 a l'exposant 3, cela signifie simplement que 168 est divisible 3 fois par 2 soit qu'il est divisible par 2x2x2 dont chacun connaît le résultat 8.., mais 8 n'est pas premier.

La méthode :

La méthode la plus efficace serait bien sûr de passer en revue tous les nombres premiers et seulement les nombres premiers. Malheureusement, il n'existe pas de formule permettant d'énumérer (directement) la suite des nombres premiers. Par contre, il existe des procédés performants permettant de dire si un nombre inférieur à un maximum donné est premier ou non : le lecteur intéressé par ces problèmes pourra consulter le livre "Calcul pour l'informatique" de Marie-José BERTIN, (chez Hermann).

La méthode la plus simple serait de prendre la suite de tous les nombres à partir de 2 et d'essayer toutes les divisions par ces facteurs, en ajoutant chaque facteur ainsi trouvé aux facteurs déjà trouvés et en recommençant avec le quotient obtenu. On s'arrête lorsque le quotient est 1.

Exemple :
168/2 donne le facteur 2 et le
    quotient 84
84/2 donne le facteur 2 et le
    quotient 42
42/2 donne le facteur 2 et le
    quotient 21
21 n'est pas divisible par 2
    on passe à 3
21/3 donne le facteur 3 et le
    quotient 7
7 n'est divisible ni par 3, ni par
    4,5 et 6 qu'il faut essayer
    successivement avec cette
    méthode
7/7 donne le quotient 1, la
    décomposition est terminée.

Certaines améliorations sont possibles en évitant d'essayer certains facteurs non premiers. Ce type de modification est une amélioration de la méthode si le programme perd moins de temps sur cette modification que sur les calculs qu'elle permet d'éviter. Voici les 2 améliorations apportées à la méthode :

1 . Les multiples de 2 et 3 sont faciles à éviter. Les multiples de 2, de façon évidente en partant

d'un nombre impair et en ajoutant 2 à chaque "tour". Les multiples de 3 à l'aide d'un test permettant de "sauter" un nombre impair sur 3 :
5 7 (9) 11 13 (15) 17 19 (21) 23 25 (27) ...

2 . Si le quotient d'une division dont le reste n'est pas nul, est inférieur au facteur essayé (diviseur), on peut affirmer que le nombre que l'on veut décomposer est un nombre premier. Ce n'est bien sûr vrai que parce que, à ce stade du traitement, on sait que ce nombre n'est divisible ni par le facteur essayé, ni par tout nombre inférieur à ce facteur éventuel.

Ainsi, le test d'arrêt de la décomposition sera une des 2 conditions suivantes :

- le quotient obtenu est exact et égal à 1
- le quotient obtenu est inférieur au facteur essayé. Alors le dividende est un facteur premier ayant l'exposant 1.

Le programme BASIC qui suit illustre de façon simple la méthode:

```
1 DIMT(12)
10 INPUT"UN NOMBRE SVP";N
20 FORI=3TO12:T(I)=0:NEXTI
22 N=INT(N):IFN<1THEN10
24 PRINT:PRINTTAB(5);N;"="
30 T(1)=1:T(2)=1:I=0
40 K=2:GOSUB1000
50 IFE=0THEN100
60 T(1)=2:T(2)=E:I=2
100 IFN=1THEN500
110 B9=3:K=3
120 GOSUB1000
130 IFE>0THENI=I+2:T(I-1)=K:T(I)=E
140 IFN=1THEN500
150 IFX<KTHENI=I+2:T(I-1)=N:T(I)=1:GOTO500
160 K=K+2:B9=B9-1
170 IFB9=0THENK=K+2:B9=2
180 GOTO120
500 REM EDITION
510 FORJ=1TOISTEP2
520 PRINTT(J)"↑"T(J+1);
530 IFJ+1<ITHENPRINT"+";
540 NEXTJ:PRINT
550 GOTO10
1000 E=0:X=N
1010 X=X/K:IFN-K*INT(X)>0THENRETURN
1020 E=E+1:N=X:GOTO1010
```

Sous-Programme Assembleur :

Il est appelé par un programme BASIC qui fait toutes les entrées-sorties.

Le sous-programme assembleur est totalement paramétré. Les adresses des zones manipulées sont définies une seule fois en tête du sous-programme, et sont donc aisément modifiables pour adapter le sous-programme à l'une ou l'autre machine.

Le listing donné en exemple a été assemblé sur un VIC 20, équipé de l'extension RAM de 16K, la seule extension de 8K étant d'ailleurs insuffisante.

Les zones B1, B2, ..., B8 ont été mises en fin de page 0 aux adresses 247 à 254.Ces zones étant en page 0, les instructions les utilisant n'occuperont que 2 octets, au lieu de 3 pour des zones situées dans une autre page.

La zone B9 et la table RESUL de 20 octets sont placées au début du tampon cassette.

Le programme est placé en 23552 et occupe 256 octets, soit juste 1 page en fin de mémoire.

Nombres manipulés par le sous-programme :

Le sous-programme manipule des nombres positifs codés sur 2 octets c'est-à-dire sur 16 bits et pouvant valoir de 0 à 65535.

Les opérations effectuées sur ces nombres sont :

- division
- addition de la constante 2
- comparaison

Pour cela, il y a 2 sous-programmes utilisés :

DIVISE qui effectue la division du nombre (B3, B4) par le nombre (B5, B6). Le dividende (B3, B4) est d'abord sauvegardé dans (B7, B8). Le reste de la division est en (B1,B2) et le quotient en (B3, B4) prêt pour la division suivante si la dernière est tombée juste.

Comment opère le sous-programme DIVISE ? Exactement comme vous lorsque vous faites une division à la main. Bien sûr il y a quelques petits aménagements. D'abord, le calcul se fait en base 2, ainsi au moment pénible du "combien y-va-t-il ?" (des adresses $5CD8 à $5CE2 du listing VIC 20 + 16K) la réponse est simple (1 ou 0) ainsi que l'action qui suit puisqu'il n'y a pas besoin de multiplier le diviseur par ce nombre avant de le soustraire...

Techniquement, il faut un quotient de 16 chiffres, le plus simple est donc de les calculer de gauche à droite, en commençant par les chiffres les plus significatifs à gauche, même si ce sont des "0".

Ainsi, le calcul se fait en 16 étapes. A chaque étape, on décale le dividende d'une position (binaire) à gauche; la partie qui déborde (dans (B1, B2)) est comparée au diviseur ("combien y va-t-il ?")

Il y a alors 2 possibilités :

- (B1, B2) > = diviseur (B5, B6) : il y va 1 fois :
  . le chiffre 1 est ajouté à droite des chiffres du quotient déjà calculés.
  . le diviseur (B5, B6) est retranché à (B1, B2) : c'est fait en SUITE 1
- (B1, B2) < diviseur (B5, B6) : il y va 0 fois et le chiffre 0 est ajouté à droite des chiffres du quotient déjà trouvés.

La zone (B3, B4) qui contient le dividence au début est décalée d'1 position à gauche à chaque étape. Cela libère une position de plus à droite de cette zone à chaque étape: ainsi la zone (B3, B4) est utilisable pour ranger le quotient au fur et à mesure de son calcul. Economie : une seule zone (B3, B4) à manipuler.

Les 16 étapes sont contrôlées par le registre X, décrémenté en SUITE 3 .Tant que les 16 étapes ne sont pas effectuées on retourne en DECALE, sinon on ajoute le dernier chiffre de droite au quotient (B3, B4) avant de revenir au sous-programme appelant.

PLUS 2 ajoute 2 au diviseur (B5,B6)

Le programme principal

DEB, MAZ : mise à 0 de la zone RESUL à RESUL 20.

La zone RESUL est organisée en triplets :

(RESUL + 1 + 3n, RESUL + 2 + 3n) contient soit un facteur premier,soit 0

RESUL + 3 + 3n contient l'exposant de ce facteur ou 0.

DIV2 : la recherche du facteur 2 et de son exposant est faite par de simples décalages à droite du dividende, de façon à compter le nombre de 0 situés à droite. Ce nombre est égal à l'exposant du facteur 2, base de numération.

C'est évidemment cette partie qu'il faudrait supprimer, en utilisant le sous-programme DIVISE, s'il fallait réduire le programme assembleur de façon à ce qu'il reste entièrement dans 1 seule page mémoire, si des modifications ou adaptations l'agrandissant étaient nécessaires.

A la fin de cette zone, après BEQ S1, l'exposant non nul est rangé dans RESUL, et si le nouveau dividende est égal à 1, le traitement est terminé, sinon le registre Y est augmenté de 3 pour pointer sur le triplet suivant dans RESUL.

S1 : on initialise la recherche des facteurs supérieurs à 2. On range le facteur éventuel 3 dans (B5, B6) (diviseur) et dans RESUL indexé par (Y-1, Y). De même, on initialise B9 à 3 de façon à "sauter" par la suite les multiples de 3.

CHEXP : c'est le début de la recherche d'un exposant, commençant par l'appel de DIVISE.

Au retour on teste (B1, B2), reste de la division :
- s'il est nul, l'exposant du facteur en cours de test est incrémenté de 1 (RESUL + ,Y),puis si le quotient obtenu en (B3, B4) est different de 1 on retourne en CHEXP

tester à nouveau ce facteur. Si le quotient obtenu est égal à 1, le traitement est terminé.

- s'il n'est pas nul, la recherche de l'exposant du facteur (B5, B6) est terminée : suite en FINEXP

FINEXP : on compare le quotient (B3, B4) au diviseur (B5, B6) s'il est inférieur on va en QINF.

NON : dans (B5, B6) ont met le facteur éventuel suivant.
Un coup sur deux B9 vaut 0 après décrémentation, on ajoute alors 2 de plus à (B5, B6) de façon à éviter les multiples de 3.

Entre S1 et CHEXP, B9 est initialisé un point plus haut de façon à ce que ce traitement en bascule (+2, +4, +2, +4,...) ne commence qu'à partir de 5 et non pas de 3.

S2 : le dividende (B3, B4) est restoré avec (B7, B8), si l'exposant de l'ancien facteur est non nul, ou pointe avec Y sur le triplet suivant dans RESUL.

S3 : le nouveau facteur éventuel (B5, B6) est rangé dans
(RESUL -, Y ; RESUL , Y), puis on retourne en CHEXP.

QINF : si l'exposant du dernier facteur est non nul, on pointe sur le triplet suivant dans RESUL.
Ensuite, on range le dernier dividende (B7, B8) (inférieur au carré du dernier diviseur (B5, B6)) dans RESUL avec l'exposant 1.

Le traitement est alors terminé.

Mise en oeuvre du programme :

Le programme, pour être utilisé, doit être assemblé et rangé à l'adresse indiquée (23552 pour le VIC 20 + 16K). Ensuite, le programme BASIC donné plus loin doit être mis en mémoire puis exécuté.

Il y a cependant un préalable important : le programme occupe une page entière de mémoire qui doit lui être réservée.

Cas du VIC 20 : cette page ou un peu plus doit être réservée en fin de mémoire. Avec l'extension de 16K on peut faire, juste après la mise sous tension du VIC :

```
POKE 51,0 : POKE 52,92 : POKE 53,0
     : 54,92
POKE 56,92 : CLR
```

Ainsi, on réserve 4 pages en fin de mémoire.

Cas de l'ORIC : On peut implanter le programme dans la page 4 ($0400-$04FF). L'appel se fera alors par CALL 400.

Cas de l'ATARI : On peut implanter le programme en page 6 ($0600-$06FF). L'appel se fera alors par SYS 1536.

Cas du CBM 64 : le mieux est d'implanter le programme en $C000. L'appel se fera alors par SYS 49152

Cas des CBM 4000 et CBM 8000 : on réserve la zone à partir de $7C00 par : POKE 49,124 : POKE 53,124 : CLR. L'appel se fera par SYS 31744.

Après ce préalable, l'assembleur peut être chargé et exécuté; le fichier source du sous-programme en assembleur peut être saisi (entré en clavier et sauvegardé ou lu depuis une sauvegarde sur disquette ou sur cassette) puis assemblé.

Ensuite, l'assembleur peut être supprimé par NEW.

Le programme BASIC d'appel et de gestion des E/S peut alors être chargé et immédiatement exécuté.

```
10 B3=249:B4=250
15 R1=830:R2=850
20 INPUTN
30 Q=INT(N/256):R=N-256*Q
40 POKEB3,Q:POKEB4,R
50 SYS23552
60 FORI=R1TOR2STEP3
70 IFPEEK(I)=0ANDPEEK(I+1)=0THENPRINT
                          :GOTO20
80 PRINTPEEK(I)*256+PEEK(I+1);
90 IFPEEK(I+2)<>1THENPRINT"↑";
                          PEEK(I+2);
100 IFPEEK(I+4)<>0THENPRINT"X";
110 NEXT
```

Programme BASIC appelant le programme assembleur

B3 et B4 sont les adresses où le nombre N lu en 20 doit être rangé pour être transmis au sous-programme assembleur qui le décomposera en facteurs premiers.

R1 correspond à RESUL + 1 du sous-programme assembleur
R2 = 830 + 20 = RESUL + 21.

La zone comprise entre les adresses R1 et R2 contient la décomposition en facteurs premiers, sous la forme des triplets décrits précédemment.

En 50, on appelle le sous-programme assembleur, dont le début est en 23552 pour le VIC 20 + 16K, avec SYS 23552.

Au retour, les facteurs premiers et leurs exposants sont lus et édités entre les lignes 60 à 110.

Performances du programme

Elles sont assez significatives de l'efficacité d'un programme en assembleur en comparaison du programme BASIC équivalent. Dans les 2 cas, en BASIC, on initialise TI$ à "000000" après la lecture de N et on édite TI au retour : le résultat est en 1/60ème de seconde, à 1/60ème près. Il faut s'assurer qu'entre l'initialisation de TI$ et la lecture de TI il n'y a aucune opération d'E/S, dans les 2 cas.

Voici les résultats obtenus pour 3 nombres premiers (les plus longs à obtenir) :

Nombre	durée prog. BASIC	durée prog. Assembleur
1423	(29/60) s	(2,30/60) s
12799	(76/60) s	(3,75/60) s
65039	(168/60) s	(6,39/60) s

On constate que les durées mesurées sont dans des rapports variant de 15 à 30. Il se peut que dans cette mesure l'outil BASIC utilisé fausse en grande partie les résultats, par réduction. Bien sûr, un rapport 30 paraîtrait plus spectaculaire s'il était obtenu avec des durées de 1 heure pour le BASIC contre 2 minutes pour l'assembleur: ce serait le cas avec un programme nécessitant davantage d'opérations!

Tableau d'adaptation des adresses

Assembleur	VIC20+ 16 K	CBM 64	ORIC	ATARI
B1	247		1	203
B2	248		2	204
B3	249		3	205
B4	250	id.	4	206
B5	:	VIC	5	207
B6	:		6	208
B7	:		7	209
B8	254		8	210
B9	828		243	568
RESUL	829		244	569
BASIC				
B3	249		3	205
B4	250		4	206
R1	830		245	651
R2	850		246	654
Appel assembleur :				
(1)	(2)		(3)	(4)

(1): SYS23552
(2): CALL#400
(3): SYS49152
(4): USR(1536)

Jean-Louis GARRIVET
Daniel TRECOURT

```
001          B1    =  247
002          B2    =  248
003          B3    =  249
004          B4    =  250
005          B5    =  251
006          B6    =  252
007          B7    =  253
008          B8    =  254
009          B9    =  828
010          RESUL =  829
011          #     =  23552
012 5C00 A9 00     DEB    LDA #0
013 5C02 A2 14            LDX #20
014 5C04 9D 3D 03 MAZ     STA RESUL,X
015 5C07 CA               DEX
016 5C08 D0 FA            BNE MAZ
017 5C0A A9 02            LDA #2
018 5C0C A8               TAY
019 5C0D 8D 3F 03         STA RESUL++'
020 5C10 A5 F9            LDA B3
021 5C12 E8         DIV2  INX
022 5C13 4A               LSR A
```

```
023 5C14 66 FA          ROR B4
024 5C16 90 FA          BCC DIV2
025 5C18 26 FA          ROL B4
026 5C1A 2A             ROL A
027 5C1B 85 F9          STA B3
028 5C1D CA             DEX
029 5C1E F0 10          BEQ S1
030 5C20 8E 40 03       STX RESUL+++
031 5C23 C9 00          CMP #0
032 5C25 D0 06          BNE NVEXP
033 5C27 A5 FA          LDA B4
034 5C29 C9 01          CMP #1
035 5C2B F0 2F          BEQ RETOUR
036 5C2D C8       NVEXP INY
037 5C2E C8             INY
038 5C2F C8             INY
039 5C30 A9 03    S1    LDA #3
040 5C32 99 3D 03       STA RESUL,Y
041 5C35 85 FC          STA B6
042 5C37 8D 3C 03       STA B9
043 5C3A A9 00          LDA #0
044 5C3C 85 FB          STA B5
045 5C3E 20 B2 5C CHEXP JSR DIVISE
046 5C41 A5 F7          LDA B1
047 5C43 D0 1A          BNE FINEXP
048 5C45 A5 F8          LDA B2
049 5C47 D0 16          BNE FINEXP
050 5C49 18             CLC
051 5C4A A9 01          LDA #1
052 5C4C 79 3E 03       ADC RESUL+,Y
053 5C4F 99 3E 03       STA RESUL+,Y
054 5C52 A5 F9          LDA B3
055 5C54 D0 E8          BNE CHEXP
056 5C56 A5 FA          LDA B4
057 5C58 C9 01          CMP #1
058 5C5A D0 E2          BNE CHEXP
059 5C5C 4C B1 5C RETOUR JMP LAFIN
060 5C5F A5 F9    FINEXP LDA B3
061 5C61 C5 FB          CMP B5
062 5C63 90 35          BCC QINF
063 5C65 D0 06          BNE NON
064 5C67 A5 FA          LDA B4
065 5C69 C5 FC          CMP B6
066 5C6B 90 2D          BCC QINF
067 5C6D CE 3C 03 NON   DEC B9
068 5C70 D0 08          BNE S2
069 5C72 20 F4 5C       JSR PLUS2
070 5C75 A9 02          LDA #2
071 5C77 8D 3C 03       STA B9
072 5C7A 20 F4 5C S2    JSR PLUS2
073 5C7D A5 FD          LDA B7
074 5C7F 85 F9          STA B3
075 5C81 A5 FE          LDA B8
076 5C83 85 FA          STA B4
077 5C85 B9 3E 03       LDA RESUL+,Y
078 5C88 F0 03          BEQ S3
079 5C8A C8             INY
080 5C8B C8             INY
081 5C8C C8             INY
082 5C8D A5 FB    S3    LDA B5
083 5C8F 99 3C 03       STA RESUL-,Y
084 5C92 A5 FC          LDA B6
085 5C94 99 3D 03       STA RESUL,Y
086 5C97 4C 3E 5C       JMP CHEXP
087 5C9A B9 3E 03 QINF  LDA RESUL+,Y
088 5C9D F0 03          BEQ S4
089 5C9F C8             INY
090 5CA0 C8             INY
091 5CA1 C8             INY
092 5CA2 A5 FD    S4    LDA B7
093 5CA4 99 3C 03       STA RESUL-,Y
094 5CA7 A5 FE          LDA B8
095 5CA9 99 3D 03       STA RESUL,Y
096 5CAC A9 01          LDA #1
097 5CAE 99 3E 03       STA RESUL+,Y
098 5CB1 60       LAFIN RTS
099 ;;
100 5CB2 A5 F9    DIVISE LDA B3
101 5CB4 85 FD           STA B7
102 5CB6 A5 FA           LDA B4
103 5CB8 85 FE           STA B8
104 5CBA A9 00           LDA #0
105 5CBC 85 F7           STA B1
106 5CBE 85 F8           STA B2
107 5CC0 A2 10           LDX #16
108 5CC2 18              CLC
109 5CC3 26 FA    DECALE ROL B4
110 5CC5 26 F9           ROL B3
111 5CC7 26 F8           ROL B2
112 5CC9 26 F7           ROL B1
113 5CCB A5 FB           LDA B5
114 5CCD C5 F7           CMP B1
115 5CCF 90 0A           BCC SUITE1
116 5CD1 D0 18           BNE SUITE2
117 5CD3 A5 FC           LDA B6
118 5CD5 C5 F8           CMP B2
119 5CD7 90 02           BCC SUITE1
120 5CD9 D0 10           BNE SUITE2
121 5CDB 38       SUITE1 SEC
122 5CDC A5 F8           LDA B2
123 5CDE E5 FC           SBC B6
124 5CE0 85 F8           STA B2
125 5CE2 A5 F7           LDA B1
126 5CE4 E5 FB           SBC B5
127 5CE6 85 F7           STA B1
128 5CE8 38              SEC
129 5CE9 B0 01           BCS SUITE3
130 5CEB 18       SUITE2 CLC
131 5CEC CA       SUITE3 DEX
132 5CED D0 D4           BNE DECALE
133 5CEF 26 FA           ROL B4
134 5CF1 26 F9           ROL B3
135 5CF3 60              RTS
136 ;;;
137 5CF4 18       PLUS2  CLC
138 5CF5 A9 02           LDA #2
139 5CF7 65 FC           ADC B6
140 5CF9 85 FC           STA B6
141 5CFB 90 02           BCC PLUSF
142 5CFD E6 FB           INC B5
143 5CFF 60       PLUSF  RTS
```

Haute résolution purement logicielle sur CBM

Introduction

Pour les lecteurs qui n'ont pas pu réaliser la carte haute-résolution parue dans "LA COMMODE", voici un programme (inédit !) qui permet l'accès aux points élémentaires (pixels) de l'écran.

Il est destiné à tous les CBM 3000 et offre une haute résolution, à raison d'un point sur chacune des 200 lignes élémentaires.

La finesse des courbes obtenues vous surprendra.

Présentation

On ne peut accéder à l'affichage que par MEV écran. Pourtant, connaissant le fonctionnement du balayage, et grâce à la rapidité du langage-machine, la haute résolution devient possible. Voici comment :

Rappels sur le programme vidéo

Le faisceau d'électrons (spot) balaie l'écran ligne après ligne (il ne s'agit pas, bien sûr, des 25 lignes de caractères, mais des 200 lignes élémentaires).

A chaque passage sur un point donné, le système vidéo recherche dans la MEV écran le code-écran du caractère à afficher.

C'est dans le générateur de caractères qu'il va chercher les bits à imprimer (fig. 1).

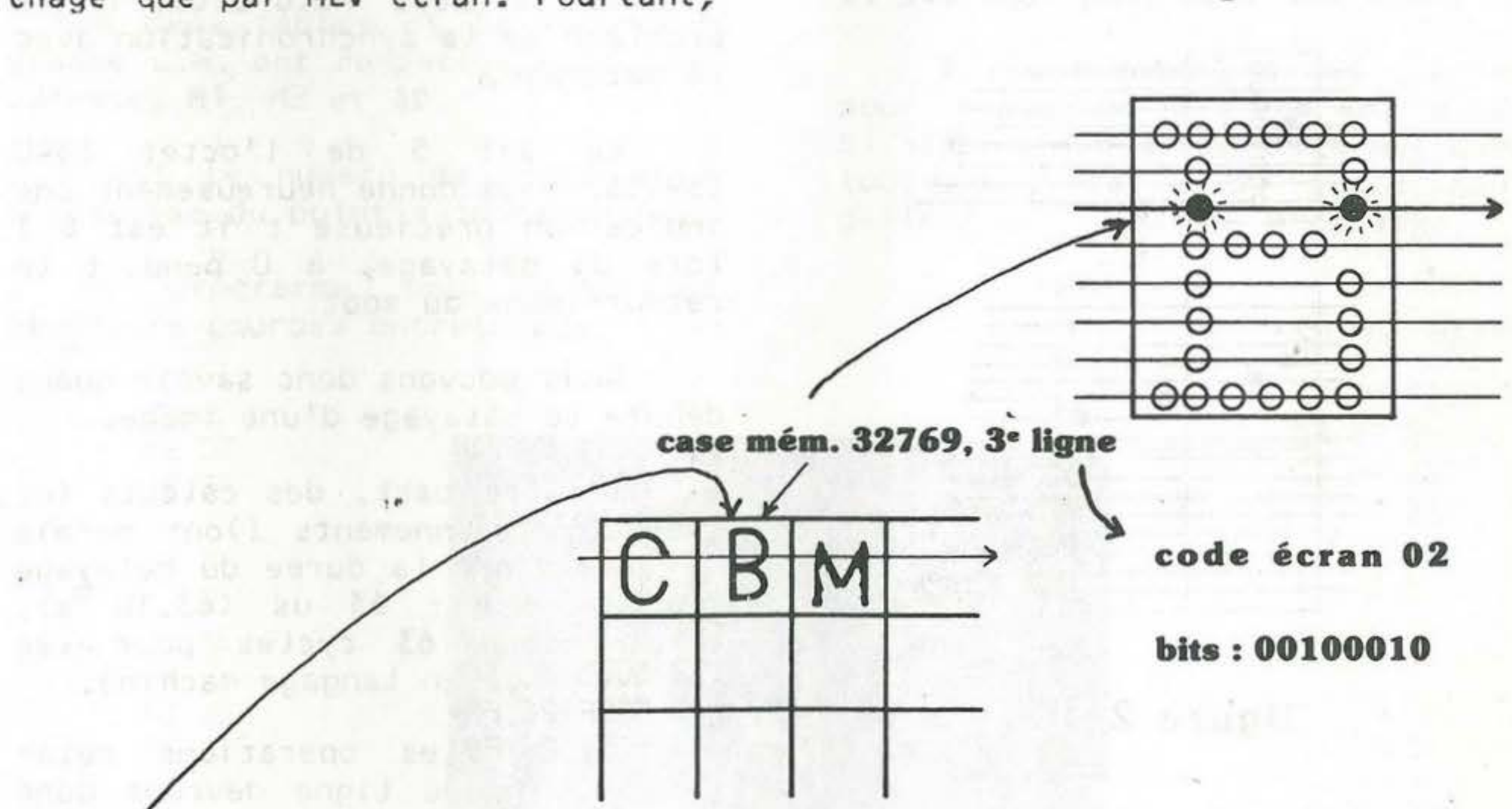


figure 1

Principe du programme

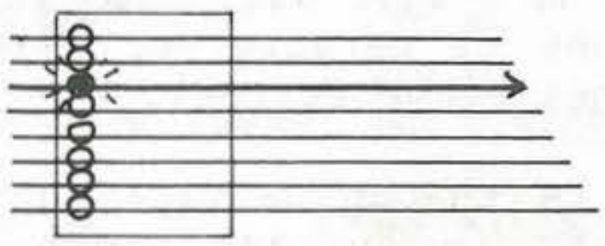
Changeons le code du caractère au cours de son balayage. On verra sur l'écran le haut du premier caractère et le bas du second.

Le jeu de caractères du CBM contient 8 segments verticaux qui permettent d'acceder aux 320 colonnes élémentaires.

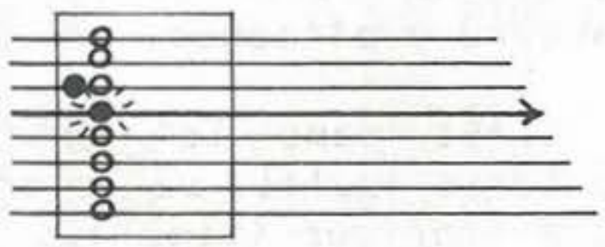
Leurs codes sont : 101, 84, 71, 66, 93, 72, 89, 103.

Si, lors du balayage d'une ligne, la mémoire écran contient un de ces segments, il s'affichera un point (fig. 2 a).

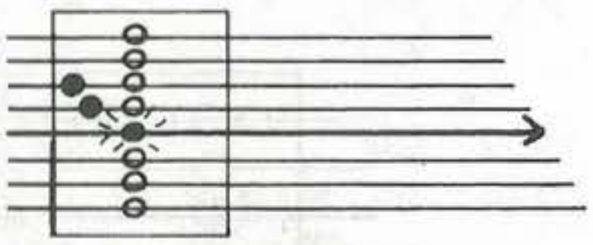
a)



b)



c)



d)

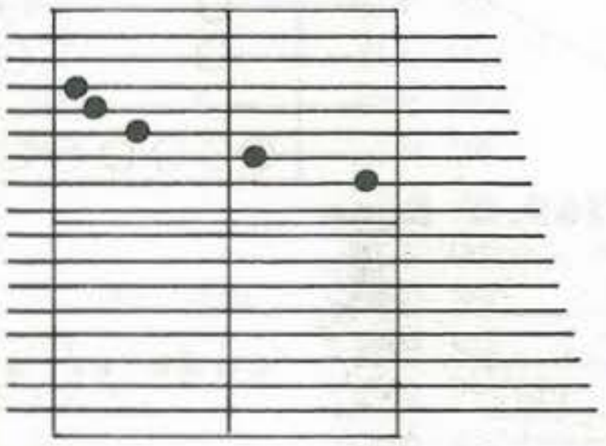


figure 2

Il s'agira donc de placer pour chaque ligne, pendant la période ou le spot la balaie, le segment correspondant à l'abscisse du point voulu sur la ligne.Il est effacé immédiatement après et remplacé par le caractère de la ligne suivante (fig 2 b).

Il faut, bien sûr, renouveler l'opération à chaque balayage de l'écran, c'est-à-dire 60 fois par seconde.

Pour le spectateur, tout se passe comme pour une image normale; le microprocesseur(*), sera occupé presque en permanence à substituer les codes-écran dans la mémoire.

L'image est mise préalablement en memoire dans deux tables de 200 octets chacune : pour chaque point la colonne (entre 0 et 39) et le code-écran du segment (fig 3).

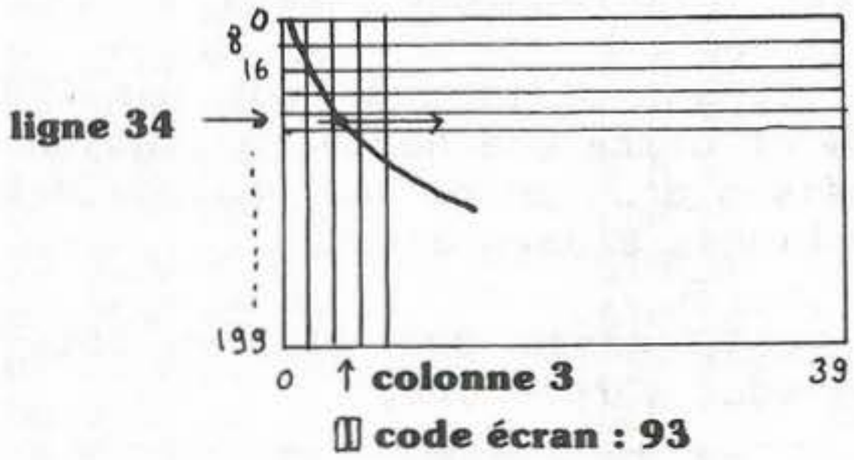


figure 3

Synchronisation

Il se pose naturellement le problème de la synchronisation avec le balayage.

Le bit 5 de l'octet E840 (59456) nous donne heureusement une indication précieuse : il est à 1 lors du balayage, à 0 pendant le retour-image du spot.

Nous pouvons donc savoir quand débute le balayage d'une image.

D'autre part, des calculs (et quelques tatonnements !)ont permis de déterminer la durée du balayage d'une ligne : 63 us (63.10^{-6} s), c'est-à-dire 63 cycles pour les instructions en langage machine.

Toutes les opérations relatives à chaque ligne devront donc être executées en 63 cycles exactement (car on n'a pas de test sur le retour-ligne).

(*) entre chaque balayage, environ 4000 us de répit lui sont laissées (retour...image); elles pourront être utilement exploitées.

Toute desynchronisation est fatale à la stabilité de l'image.

Insistons donc sur ce temps très bref, et qui limitera les possibilités du programme à traiter un seul point par ligne (ce qui n'est déjà pas si mal !).

Remarques sur le programme

. Le programme ci-joint est en BASIC, et contient le programme langage-machine sous forme de DATA.

. Il a l'avantage de s'adapter facilement à la configuration de votre CBM : 8, 16 ou 32 K, pour le rangement des tables et du langage-machine en fin de mémoire, et la réservation mémoire. Les modifications pour le PET sont en REM aux lignes 35, 5010, 5160.

. L'INPUT en ligne 50 permet de ne pas reenregistrer le programme langage-machine si on veut tracer d'autres courbes, en particulier si on change la ligne 140.

. Les deux tables et le sous-programme L.M. ont respectivement pour adresses M1, M2 et AD.

. X est le numero de ligne et Y l'abscisse du point a imprimer.

. Le programme peut mémoriser plusieurs courbes entrelacées.

. L'image apparaît à l'appel du sous-programme machine par un SYS (ligne 160).

Remarque importante : seuls les caractères sur lesquels passe la courbe sont effaces, les autres ne subissant aucune modification, on peut conserver des textes avec la haute définition.

. L'enfoncement de la touche SHIFT permet le retour au BASIC, les courbes s'effacent alors. En effet, le microprocesseur ne peut gérer en même temps la haute-definition et le BASIC.

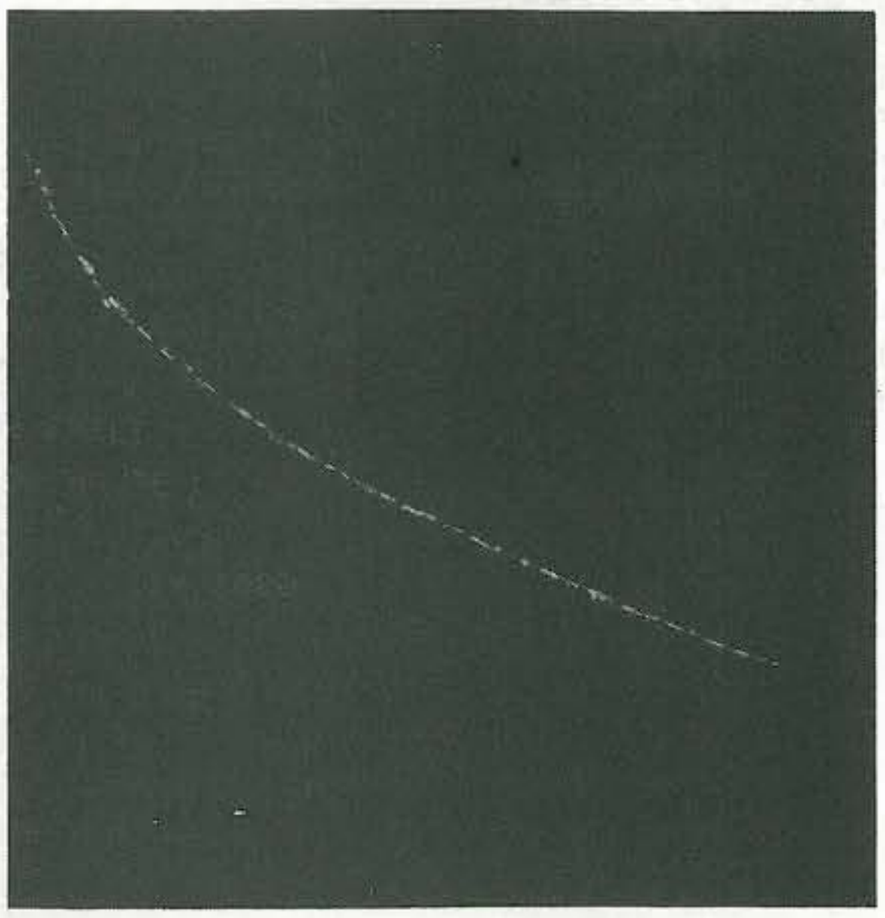
. On retrouve l'image instantanément avec SYS 256 * M(3).

Conclusion

Les habitués du langage-machine devraient pouvoir, sans trop de difficultés, nous l'espérons, adapter ce programme pour les CBM 4000 et pourquoi pas, pour les 8000.

Et maintenant, à vos claviers pour exploiter à fond les possibilités graphiques, mathematiques, ludiques, etc.., de cette haute définition par programme !

Pierre BARRAS



```
10 REM ***HAUTE DEFINITION ***
15 REM ****JUILLET 1983********
20 REM  AUTEURS = P. & C. BARRAS
25 REM
30 REM *** LECTURE DES DATAS ***
35 READ A:POKE 49,A:POKE 53,A:CLR:REM PET= READ A:POKE131,A:POKE135,A:CLR
40 FOR I=1 TO 3:READ M(I):NEXT:M1=M(1)*256:M2=M(2)*256:AD=M(3)*256
45 FOR I=0 TO 8:READ N(I):NEXT
50 INPUT"REMISE EN MEMOIRE DU PROGRAMME";A$:IF A$="NON" THEN 60
55 READ A:IF A<0 THEN A=M(-A):S=S-A
56 IFA<256THEN POKEAD,A:AD=AD+1:S=S+A:GOTO55
57 AD=M(3)*256:IF S<>A THEN PRINT"ERREUR DANS LES DATA!":STOP
60 FOR I=M1 TO AD-1:POKE I,32:NEXT
70 REM
80 REM **POUR PLUSIEURS COURBES**
90 REM
100 INPUT"NB DE COURBES";T
110 FOR G=0 TO (T-1)
130 FOR X=G TO 200 STEP T
140 Y=(G+1)*X*X/100
145 PRINT"■";G,X
150 GOSUB 200
160 NEXT:NEXT:SYS 256*M(3)
170 END
180 REM **MISE EN MEMOIRE ******
190 REM
200 IF Y<0 OR Y>320 THEN RETURN
210 POKE M1+X,Y/8:POKE M2+X,N(Y AND 7)
220 RETURN
4000 REM *** M2,M1,DEBUT PROG ***
4010 DATA 29,30,31    :REM 8K
4015 REM 16K= DATA 61,62,63
4020 REM 32K= DATA 125,126,127
4025 :
4500 REM *** SEGMENTS VERTICAUX ***
4510 DATA 101,84,71,66,93,72,89,103,32
4520 :
5000 REM *** PROGRAMME HIRESCREEN ***
5005 DATA 162,0,188,0,-1,169,0,133,177
5010 DATA 169,128,133,178,173,152,0:REM PET= DATA 169,128,133,178,173,4,2
5020 DATA 240,1,96,169,32,44,64,232
5030 DATA 208,251,44,64,232,240,251
5040 DATA 189,0,-2,145,177,234,234,234
5050 DATA 234,232,224,200,240,52,138:REM 200= NOMBRE DE LIGNES ELEMENT.
5060 DATA 41,7,208,6,32,83,-3,76,72
5070 DATA -3,234,234,234,234,234,234
5080 DATA 234,234,234,234,234,234,169
5090 DATA 32,145,177,188,0,-1,189,0
5100 DATA -2,145,177,76,39,-3,169,40
5110 DATA 24,101,177,133,177,169,0,101
5120 DATA 178,133,178,96,169,0,133,177
5130 DATA 169,128,133,178,162,7,188
5140 DATA 0,-1,169,32,145,177,32,83
5150 DATA -3,138,24,105,8,170,224,200:REM 200=NBR LIGNES
5160 DATA 144,237,76,0,-3,16217:REM SOMME CONTROLE. SUR PET: 16071
```

```
HAUTE-DEF.....PAGE 0001

LOC   CODE           LINE


0000                ;PROGRAMME DE HAUTE DEFINITION
0000                ;A RAISON D'UN PIXEL
0000                ;PAR LIGNE ELEMENTAIRE
0000                ; AUTEUR = P. BARRAS
0000                ;
0000                      * = $1F00
1F00                M1    = $1D00
1F00                M2    = $1E00
1F00                Z     = $B1
1F00                LIBR  = $B4
1F00                OCTET = $E840          ;OCTET INDIQUANT L'ETAT DU SPOT
1F00                SHIFT = $0204          ;SUR C.B.M.=$98
1F00                ECRAN = $8000
1F00                NBCOL = 40
1F00                NBLIGN = 200
1F00                SPC   = 32
1F00                ;
1F00  A2 00         DEBUT LDX #$00         ;POINTEUR N0 LIGNE ELEMENTAIRE
1F02  BC 00 1D            LDY M1,X        ;COORD.CARACT. SUR LIGNE ECRAN
1F05  A9 00               LDA #<ECRAN     ;Z=POINTEUR DE
1F07  85 B1               STA Z           ;DEBUT DE LIGNE
1F09  A9 80               LDA #>ECRAN     ;SUR L'ECRAN
1F0B  85 B2               STA Z+1         ;AU DEBUT Z=32768
1F0D  AD 04 02            LDA SHIFT       ;(SUR C.B.M.= AD 98 00)
1F10  F0 01               BEQ CONTI       ;SI TOUCHE SHIFT ENFONCEE,
1F12  60                  RTS             ;ON S'ARRETE
1F13  A9 20         CONTI LDA #$20        ;TESTE BIT 5 DE $E840
1F15  2C 40 E8      BC1   BIT OCTET       ;ATTEND LA FIN DE
1F18  D0 FB               BNE BC1         ;L'ANCIEN BALAYAGE
1F1A  2C 40 E8      BC2   BIT OCTET       ;ATTEND LE DEBUT
1F1D  F0 FB               BEQ BC2         ;DU NOUVEAU BALAYAGE
1F1F  BD 00 1E            LDA M2,X        ;PREND CODE CARACTERE
1F22  91 B1               STA (Z),Y       ;LE MET A SA POSITION SUR ECRAN
1F24  EA                  NOP             ;TEMPORISATION DE
1F25  EA                  NOP             ;6 CYCLES
1F26  EA                  NOP
1F27  EA            PRINC NOP             ;BOUCLE  PRINCIPALE
1F28  E8                  INX             ;INCREMENTE N0 LIGNE ELEMENT
1F29  E0 C8               CPX #NBLIGN     ;SI FIN ECRAN,ON TERMINE
1F2B  F0 34               BEQ FIN
1F2D  8A                  TXA             ;SI X/8 ENTIER,ON INCREMENTE
1F2E  29 07               AND #$07        ;LE POINTEUR DEBUT DE LIGNE
1F30  D0 06               BNE NONINC      ;DE 40 ET ON IMPRIME LE
1F32  20 53 1F            JSR INCR        ;CARACTERE SUIVANT
1F35  4C 48 1F            JMP NEWCAR
1F38  EA            NONINC NOP            ;SINON,
1F39  EA                  NOP
1F3A  EA                  NOP
1F3B  EA                  NOP             ;TEMPORISATION DE
1F3C  EA                  NOP             ;24 CYCLES
1F3D  EA                  NOP
1F3E  EA                  NOP
1F3F  EA                  NOP
1F40  EA                  NOP
1F41  EA                  NOP
```

```
HAUTE-DEF......PAGE 0002

LOC   CODE            LINE


1F42  EA                      NOP
1F43  EA                      NOP
1F44  A9 20                   LDA #SPC        ;EFFACE CARACTERE
1F46  91 B1                   STA (Z),Y       ;PRECEDENT
1F48  BC 00 1D     NEWCAR     LDY M1,X         ;Y=COORD.SUR LIGNE ECRAN
1F4B  BD 00 1E                LDA M2,X        ;MET CARACTERE A IMPRIMER
1F4E  91 B1                   STA (Z),Y       ;DANS SON ADDRESSE ECRAN
1F50  4C 27 1F                JMP PRINC       ;RETOURNE A LA BOUCLE PRINCIPALE
1F53                ;
1F53  A9 28         INCR      LDA #NBCOL       ;INCREMENTE
1F55  18                      CLC             ;LE POINTEUR DE
1F56  65 B1                   ADC Z           ;DEBUT DE LIGNE ECRAN
1F58  85 B1                   STA Z           ;DE 40
1F5A  A9 00                   LDA #0
1F5C  65 B2                   ADC Z+1
1F5E  85 B2                   STA Z+1
1F60  60                      RTS
1F61                ;
1F61  A9 00         FIN       LDA #<ECRAN      ;REINITIALISE LE
1F63  85 B1                   STA Z           ;POINTEUR ECRAN
1F65  A9 80                   LDA #>ECRAN
1F67  85 B2                   STA Z+1
1F69  A2 07                   LDX #7          ;N0 DE LIGNE ECRAN
1F6B  BC 00 1D     LIGNE      LDY M1,X         ;COORDONEE CORRESPONDANTE
1F6E  A9 20                   LDA #SPC        ;EFFACE LE CARACTERE QUI
1F70  91 B1                   STA (Z),Y       ;N'AVAIT PAS ETE EFFACE
1F72  20 53 1F                JSR INCR        ;INCREMENTE POINTEUR
1F75  8A                      TXA             ;INCREMENTE N0 DE LIGNE
1F76  18                      CLC             ;DE 8
1F77  69 08                   ADC #8          ;
1F79  AA                      TAX             ;SI <200,ON
1F7A  E0 C8                   CPX #NBLIGN     ;CONTINUE.
1F7C  90 ED                   BCC LIGNE       ;SINON ON RETOURNE
1F7E  4C 00 1F                JMP DEBUT       ;AU DEBUT DU PROGRAMME
1F81                          .END


  ERRORS = 0000


  SYMBOL TABLE

  SYMBOL   VALUE
   BC1       1F15     BC2       1F1A     CONTI     1F13     DEBUT     1F00
   ECRAN     8000     FIN       1F61     INCR      1F53     LIBR      00B4
   LIGNE     1F6B     M1        1D00     M2        1E00     NBCOL     0028
   NBLIGN    00C8     NEWCAR    1F48     NONINC    1F38     OCTET     E840
   PRINC     1F27     SHIFT     0204     SPC       0020     Z         00B1

  END OF ASSEMBLY
```

Programme d'inversion vidéo

Voici un programme utilitaire qui permet d'écrire en noir sur fond vert, sur CBM 3032

Le CBM execute 60 fois par seconde une routine d'interruption permettant de prendre en compte une frappe clavier. Cette routine d'interruption est situee à l'adresse $E62E. Le CBM s'y dirige par les adresses $90 et $91 contenant respectivement $2E et $E6.

A ces adresses, imposons l'adresse de notre sous-programme (situe en 826, buffer du magnetophone numero 2 soit $033A)

en $90	$3A	POKE 144,58
$91	$03	POKE 145,3

Avec cette deviation et notre sous-programme tous les 1/60ème de seconde, on inversera les caracteres video qui ne le sont pas (ecriture noire sur fond vert).

Pour revenir à la normale, écriture verte sur fond noir, on retablit les vecteurs d'interruption.

$90	$2E	POKE 144,46
$91	$E6	POKE 145,230

Attention, tant qu'on dévie la routine d'interruption, on ne peut utiliser les magneto-cassettes.

Ci-joint le sous-programme desassemble par EXTRAMON 1000 suffisamment commente et le programme Basic en permettant le chargement.

Ce programme fait suite au programme TREVERSE publié dans le numero 7 de La Commode.

```
.,  033A A5 97     LDA $97      On teste si une touche est enfoncée au
.,  033C C9 FF     CMP #$FF     clavier (Pas de touche: ($97)=$FF )
.,  033E D0 3F     BNE $037F    Si touche enfoncée: saut en $037F
.,  0340 A5 C4     LDA $C4      Sauvegarde du pointeur adresse écran avant
.,  0342 8D A0 03  STA $03A0    exécution du S/P, dans les mémoires
.,  0345 A5 C5     LDA $C5      $03A0-03A1
.,  0347 8D A1 03  STA $03A1
.,  034A A2 00     LDX #$00        0 --> X
.,  034C A9 00     LDA #$00     Initialisation pointeur adresse écran à
.,  034E 85 C4     STA $C4      $8000 soit 32768 qui est le début de la
.,  0350 A9 80     LDA #$80     mémoire écran
.,  0352 85 C5     STA $C5
.,  0354 A0 00     LDY #$00        Y --> 0
.,  0356 B1 C4     LDA ($C4),Y  Contenu case écran pointée par Y --> Acc
.,  0358 C9 80     CMP #$80     Comparaison avec 128 ($80)
.,  035A 10 04     BPL $0360    >128 : branchement en $0360
.,  035C 69 80     ADC #$80     <128 : on ajoute 128 (vidéo inversée)
.,  035E 91 C4     STA ($C4),Y  Réécriture case écran
.,  0360 C8        INY          Y+1 --> Y
.,  0361 C0 28     CPY #$28     A-t-on testé les 40 cases d'une ligne écran
.,  0363 D0 F1     BNE $0356       si non : retour en $0356
```

```
.;  0365 18        CLC
.;  0366 A9 28     LDA #$28     On ajoute 40 au pointeur adresse écran en
.;  0368 65 C4     ADC $C4      tenant compte d'une éventuelle retenue
.;  036A 85 C4     STA $C4
.;  036C 90 02     BCC $0370
.;  036E E6 C5     INC $C5
.;  0370 E8        INX          Les 25 lignes d'écran sont-elles testées ?
.;  0371 E0 19     CPX #$19
.;  0373 D0 DF     BNE $0354       si non : retour en $0354
.;  0375 AD A0 03  LDA $03A0     Restauration du pointeur adresse écran tel
.;  0378 85 C4     STA $C4      qu'il était avant l'appel du S/P
.;  037A AD A1 03  LDA $03A1
.;  037D 85 C5     STA $C5
.;  037F 4C 2E E6  JMP $E62E    Retour à la routine d'interruption système
```

```
 11 REM*****************************
 12 REM*  PROGRAMME FONDINVR  *
 13 REM*                     *
 14 REM*  VIDEO     INVERSEE  *
 15 REM* COPYRIGHT A. HESBOIS *
 16 REM*     JUIN 1982       *
 17 REM*****************************
 18 :
 19 REM CGHT S/P LANGAGE MACHINE
 20 READL
 21 READA$:IFA$="*"THEN26
 22 A=ASC(A$)-48:B=ASC(RIGHT$(A$,1))-48:C=LEN(A$)
 23 N=B+7*(B>9)-(C=2)*(16*(A+7*(A>9))):POKEL,N:L=L+1:GOTO21
 24 :
 25 REM S/P LANGAGE MACHINE
 26 DATA 826
 27 DATA A5,97,C9,FF,D0,3F,A5,C4,8D,A0
 28 DATA 03,A5,C5,8D,A1,03,A2,00,A9,00
 29 DATA 85,C4,A9,80,85,C5,A0,00,B1,C4
 30 DATA C9,80,10,04,69,80,91,C4,C8,C0
 31 DATA 28,D0,F1,18,A9,28,65,C4,85,C4
 32 DATA 90,02,E6,C5,E8,E0,19,D0,DF,AD
 33 DATA A0,03,85,C4,AD,A1,03,85,C5,4C
 34 DATA 2E,E6,*
 35 :
 36 REM MODE EMPLOI
 37 PRINT"■"TAB(15)"■ FONDINVERS "
 38 PRINT"■■POUR ACTIVER CET UTILITAIRE FRAPPEZ
 39 PRINT"■EN MODE DIRECT ■ POKE144,58:POKE145,3 "
 40 PRINT"■■POUR DESACTIVER L'UTILITAIRE FRAPPEZ
 41 PRINT"■EN MODE DIRECT ■ POKE144,46:POKE145,230 ":END
READY.
```

A. HESBOIS

Bibliographie

Nous inaugurons avec ce numéro l'emploi de signes conventionnels:

*** = indispensable . = bof
** = achetez 0 = zéro
* = utile 000 = beurk

EN FRANÇAIS

LE TOUT MICRO (Hachette) ★★★

Ce guide très bien réalisé est indispensable que vous ayez ou non un microordinateur. Il donne un panorama extrèmement bien fait de tout ce qu'il faut savoir en micro-informatique. On trouve successivement une introduction technique à la micro (dont certains points sont détaillés dans une section consacrée aux branchements et interfaces et dans une section spéciale graphiques), un banc d'essai résumé des principaux micros du marché, une liste de clubs (où on ne parle pratiquement que des Microtels), un panorama des logiciels pour chaque micro, une bibliographie complète et bien commentée, une liste de librairies spécialisées, une étude sur la Presse Microinformatique et sur les Radios, une liste de boutiques micros, une étude sur les organismes publics liés à la micro, sur les formations, un calendrier des expos jusqu'à mai 85, un glossaire et enfin la liste des éditeurs en micro. Le tout est très correctement fait et justifie cent fois l'achat.

DEBUTEZ EN FORTH de Léo Brodie (Eyrolles) ★★★

Le meilleur livre sur Forth. Pas étonnant, l'auteur est l'inventeur du langage. Si vous voulez vraîment comprendre FORTH, jetez les autres livres que vous aviez éventuellement achetés (tant pis! il fallait lire cette biblio avant!) et lisez celui-ci. Par exemple un des éléments les plus difficiles à comprendre en FORTH est celui des modes de fonctionnement: est on en mode édition, en mode compilation, en mode adjonction au vocabulaire etc...? Eh bien ceci est parfaitement expliqué dans ce livre. Notons aussi que la compréhension est facilitée par d'excellents petits dessins.

JEUX ET PROGRAMMES POUR L'ORDINATEUR ORIC/ATMOS de Laurent Chemla, Nicolas Stronck (Shift édition) ★

Deux versions du même livre selon que vous avez un ORIC 1 ou un ATMOS. Il s'agit d'un recueil de programmes, surtout de jeux. Les listings sont très bien imprimés. Les programmes en eux-mêmes sont intéressants. On peut regretter l'absence de commentaires, mais alors le livre aurait fait 200 pages grand format. Faites l'effort de comprendre les programmes. Il y en a 39 qui font tous en moyenne 2 pages de listing: c'est donc plus conséquent que les "pour tous" et autres.

ATMOS - ORIC 1 Manuel de Référence d'André Chenière (IS éditions) ★★★

Nous mettons 4 étoiles car ce livre les mérite, ou même 5 ou 6! Si l'ORIC ou l'ATMOS vous passionne, posez votre exemplaire de La Commode et courez l'acheter. C'est ce livre là qui mériterait le titre "tout savoir sur ORIC" et non pas celui de chez Eyrolles qui porte prétentieusement et indûment ce titre. Après la parution d'un tel livre, La Commode n'a plus de raison de parler de l'ORIC (voyez tout de même les articles ORIC de ce numéro!). En effet ce livre offre non seulement la liste de toutes les adresses intéressantes de l'ORIC avec bien entendu les correspondances pour l'ATMOS, mais encore elles sont toutes expliquées de façon détaillée. Il n'y a pas le listing désassemblé de la ROM, mais il y a ce listing pour les parties les plus importantes et elles sont commentées de façon circonstanciee. D'autre part, toutes les notions qui sont expliquées sont accompagnées d'une application utile qui s'y refère. Quoi? Vous n'êtes pas encore parti chercher ce livre?

EN ANGLAIS

ATARI PLAYER MISSILE GRAPHICS IN BASIC de Philip C. Seyer (Reston – Prentice Hall) ★★

Ce livre décrit de façon détaillée et progressive comment programmer les Joueurs et Misiles (les "sprites" de l'Atari). Toutes les notions sont accompagnées de programmes-exemples intéressants et bien imprimés. Le seul défaut de ce livre est la langue anglaise: il suffit d'un éditeur et un traducteur pour supprimer cet obstacle.

A + PROGRAMMING IN ATARI BASIC de John M. Reisinger (Reston – Prentice Hall) ★★

Ce livre est un cours détaillé du langage BASIC ATARI. Il est très fouillé mais l'exposé est progressif. Tous les éléments sont étudiés avec de nombreux programmes-exemples. Les modes graphiques sont, entre autres, bien étudiés. Les listings de programmes sont bien imprimés.

Pierre-Etienne THALBERG

Communiqués de presse

Computer World commercialise une cartouche de conversion pour C64 à l'usage des radio-amateurs (RTTY):

Computer World
P. Box 14
1230 AA LOOSDRECHT
Hollande

Prix : 139 livres ou 189$

* *
*

ETCIB vient de se doter d'un centre serveur expérimental :

ETCIB - ECOLE PIGIER
Centre Serveur Expérimental
7 bis Avenue de la République
93250 VILLEMOMBLE

Les responsables du centre sont :

Messieurs VALDERE et CASTRATARO
tel: (16 * 1) 854 09 80
" 528 60 64

* *
*

GEM organise durant le mois d'août des stages d'initiation à la micro-informatique. Chaque stage dure une semaine et coûte 1000 F, frais de nourriture non compris.

Stage INFORMATIQUE et MONTAGNE
au centre du col de la Charmette en Chartreuse.
6 demi-journées consacrées à l'informatique. 6 demi-journées consacrées à la montagne.

du lundi 6 au samedi 11 août
du lundi 13 au samedi 18 août
du lundi 20 au samedi 25 août
du lundi 27 au samedi 1er septembre

Initiation à la micro-informatique, connaissance du BASIC, programmation en BASIC, applications micro-informatiques, utilisation des programmes d'applications standards.

G.E.M.
GRENOBLE ET MONTAGNE
18 rue Brocherie
38000 Grenoble

tel: (76) 42.53.22

* *
*

EPS (Ecole Professionnelle Supérieure) organise dans les centres EPS de Paris, Guyancourt, Nantes et Lyon les formations aux qualifications professionnelles suivantes, agrées par les Commissions Paritaires de l'Emploi du Commerce, de la Pharmacie et de la Coiffure :

- Analyste-programmeur en micro-informatique (6 mois à temps plein),

- Technico-commercial en micro-informatique (6 mois à temps plein),

- Technicien de maintenance en micro-électronique (6 mois à temps plein),

- Secétaire bureautique (6 mois à temps plein),

- Micro-informatique industrielle et de gestion (3 mois à temps plein).

Les inscriptions pour les prochaines sessions débutant courant septembre et octobre sont en cours (l'admission se fait sur test et entretien).
Pour tous renseignements, appeler EPS (3) 043 57 90 ou (1) 523 35 30.

EPS organise également chaque mois plusieurs sessions de formation aux progiciels : dBase II (3 jours), Wordstar/Mailmerge (2 jours), Multiplan (2 jours), Lotus 1.2.3 (3 jours), etc... notamment sur micro-ordinateurs IBM-PC, AXEL, ZENITH, RAIR, VICTOR, ...

Pour recevoir le calendrier de ces stages ou pour l'organisation de stage sur mesure, écrire ou téléphoner à :

- EPS Paris, 45 rue des Petites Ecuries, 75010 Paris
(tel: (1) 523 35 30)

- EPS Yvelines, 25 rue Ambroise Croizat, 78280 Guyancourt
(tel: (3) 043 57 90)

- EPS Nantes, 8 rue Linné, 44000 Nantes (tel: (40) 73 52 58)

- EPS Lyon, 145 rue Jean Jaurès, 69002 Lyon (tel: (7) 858 90 59)

* *
*

ANNONCES

Création d'une BIBLIOTHEQUE de LOGICIELS pour COMMODORE 64. Par l'intermédiaire de ce club, les abonnés pourront échanger des programmes. Les éventuels intéressés devront écrire ou téléphoner au:

(16*1) 780 10 61

Mr BOURDIN Bruno
12 rue Pasteur
92250 La Garenne Colombes

*
* *

Je mets n'importe quel programme BASIC sur ROMs ou EPROMs, ainsi que tout programme binaire. Ceci sur matériel CBM et tout autre appareil COMMODORE ou non.

Exemples pour EPROMs 2532 (je fournis toutes les ROMs ou EPROMs):

- Copie de ROMs ou d'EPROMs : 200F

- Modification
de ROMs ou d'EPROM : 300F

- Programmation binaire
de ROMs ou d'EPROMS : 300F

- Mise en ROMs ou EPROMs d'un
programme BASIC : 300F

CASTRATARO Pascal
15 rue Saint Roch
Prémont
28500 VERNOUILLET

tel:16*37-82-38-77

La Commode

3e année

Diffusion :
Ed. du P.S.I.
41-51, rue Jacquard
B.P. 86
77400 LAGNY

Rédaction-vente-abonnements :
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75.

Rédacteur en chef :
Daniel TRECOURT

Chefs de rubriques :
Jean DELAVILLE
Jean-Luc DESCHAMPS
Pierre-Etienne THALBERG

Publicité :
FORCE 7
5, place du Colonel Fabien
75010 PARIS
Tél. 240.22.01

Publié par SEDERMI SARL
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

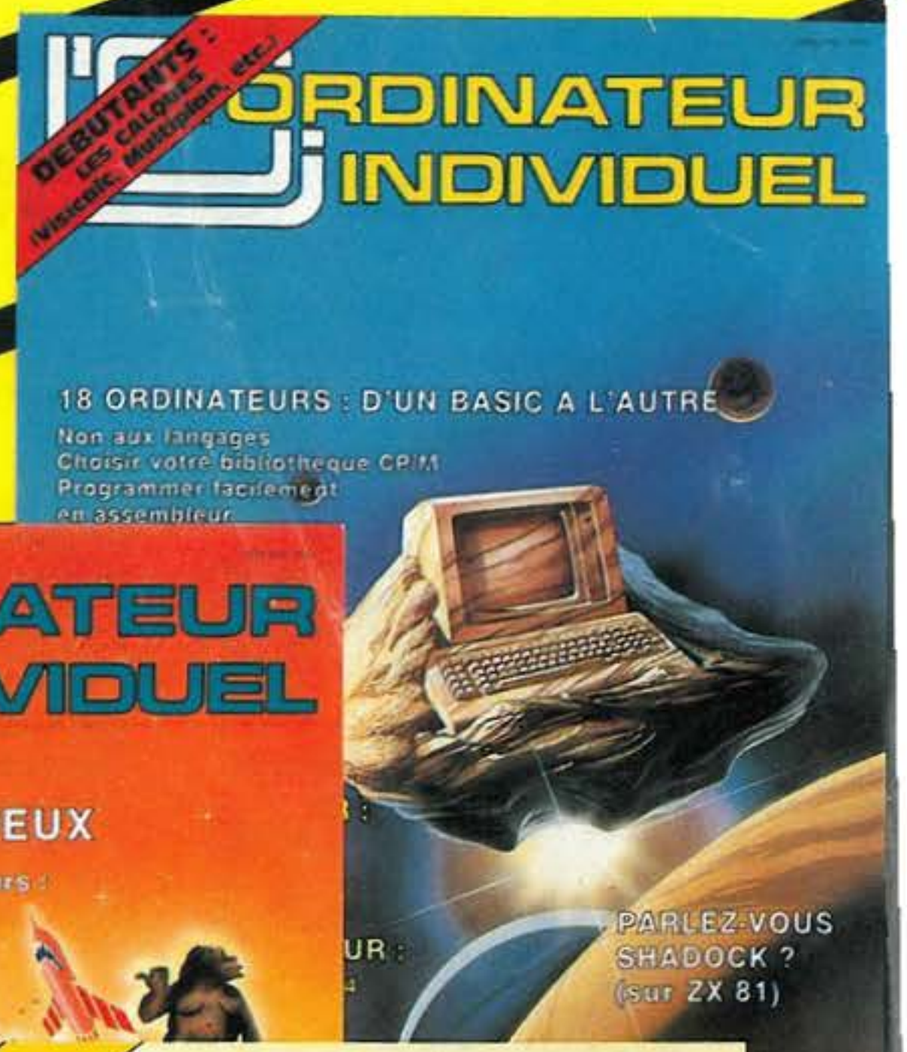
Directeur de la publication :
Daniel-Jean DAVID

Fabrication Scoop Presse Normande - Evreux. Impression ETC - 76190 Yvetot. Directeur de la publication Daniel-Jean David
Dépôt légal : juillet 1984 — Numéro de commission paritaire : en cours. ISSN 0292-5079

POUR MIEUX CHOISIR VOTRE ORDINATEUR ET POUR MIEUX L'UTILISER :

Vous y trouverez :
l'actualité et les tendances de l'informatique individuelle
- les bancs d'essais des principaux matériels
- des panoramas et des tests comparatifs
- le point des grandes manifestations internationales
- des articles d'initiation
- des synthèses
- des programmes
- des interviews "exemplaires"
- des conseils
- des idées
- des astuces

22 FF chez votre marchand de journaux



L'ORDINATEUR INDIVIDUEL

La Référence

Mettez des idées dans votre ordinateur.

La découverte du Commodore 64
par Daniel-Jean David - 176 pages

90,00 FF

Une introduction générale sur l'informatique, suivie d'un apprentissage progressif du langage Basic vous amène pas à pas à construire vos propres programmes.
Des notions nouvelles sont introduites graduellement et, grâces à elles, vous apprendrez vite à maîtriser les points forts du Commodore : les graphiques, les sons, les couleurs, la haute résolution et les " sprites ".

La pratique du Commodore 64
par Daniel-Jean David - 176 pages

90,00 FF

De la cassette au disque souple, de l'imprimante aux poignées de jeux et crayon lumineux, ce livre vous donnera tout ce qu'il faut connaître pour utiliser au mieux les périphériques de votre Commodore 64. Conçu dans l'esprit de " la découverte du Commodore 64 " du même auteur cet ouvrage contient de nombreux programmes écrits tant pour les applications personnelles que professionnelles. Un chapitre est consacré aux notions sur les bases de données et au systèmes d'exploitation disque. La programmation de l'interface RS 232 est décrite.

Commodore 64 pour tous
par Jacques Boisgontier, Sophie Brébion et Gérard Foucault - 176 pages

100,00 FF

Cet ouvrage initie le lecteur au langage Basic du Commodore 64. Les auteurs commencent par présenter les notions fondamentales de la programmation (variables, tests, boucles...). Sont ensuite étudiées les caractéristiques intéressantes du Commodore 64 : les graphiques, les sons, les sprites, avec de nombreux exemples illustrés.

Commodore 64 : méthodes pratiques
par Jacques Boisgontier - 176 pages

100,00 FF

Cet ouvrage contient 50 exercices et programmes de jeux, d'éducation et de gestion qui permettent d'en savoir plus sur les possibilités du Commodore 64, en particulier sur les graphiques haute et basse résolution, les sons, les sprites, les fichiers séquentiels et relatifs, la redéfinition des caractères etc.

Les fichiers séquentiels en Basic sur Commodore 64
par Pierre Fraser - 160 pages

110,00 FF

Le Commodore 64 est un ordinateur individuel qui dispose d'importantes possibilités d'utilisation de fichiers. Pierre Fraser, professeur à l'université de Québec, travaille sur les ordinateurs Commodore depuis plusieurs années. Il nous présente une introduction à la structure et à l'utilisatio du Commodore 64 et des fichiers séquentiels en Basic.

Jeux, trucs et comptes pour Commodore 64
par Michel Benelfoul et Cyril Cambien 192 pages

110,00 FF

Cet ouvrage propose aux novices de la programmation 30 programmes en Basic commentés et décrits à l'aide d'un exemple d'exécution et d'un organigramme (jeux passifs et interactifs, interludes, " trucs ", paie, facturation simple et routines etc.).

102 programmes pour Commodore 64
par Jacques Deconchat - 240 pages

110,00 FF

Au fil de ces 102 programmes de jeux ce livre guidera le lecteur dans l'exploration du Basic du Commodore 64. Les programmes sont classés par niveau, chacun d'eux faisant appel à de nouvelles connaissances et à une plus grande maîtrise du Basic. Les programmes sont décrits abondamment commentés et accompagnés d'un exemple d'exécution.

L'assembleur du Commodore 64
par Daniel-Jean David
prix : nous consulter

Les utilisateurs du Commodore 64 trouveront dans cet ouvrage l'explication et l'utilisation du jeu d'instruction du langage machine de leur ordinateur. L'assembleur symbolique, l'éditeur et le chargeur y sont décrits.

Le livre de bord du Commodore 64
par Mathieu Kokinski - 256 pages

120,00 FF

Ce livre est une introduction à l'ordinateur individuel Commodore 64, à son équipement, à ses périphériques, ainsi qu'à son langage Basic. Sont aussi abordés : la programmation simple, les possibilités graphiques et musicales, l'utilisation des fichiers, les programmes de tri, etc. L'ouvrage est complété de programmes variés.

Le Commodore 64 à l'affiche
par Jean-François Sehan
prix : nous consulter

Une sélection de 30 programmes de jeux utilisant les possibilités graphiques et sonores du Commodore 64. Chaque programme est accompagné d'un organigramme, d'une liste de variables et d'une explication de chaque ligne Basic pour une adaptation éventuelle à d'autres ordinateurs.

Programme interne du Commodore 64
par Milton B. Bathurst - 252 pages

130,00 FF

Ce livre est destiné à ceux qui désirent savoir comment leur Commodore 64 accomplit son travail. Le lecteur y trouvera la liste complète du programme interne de l'ordinateur, détaillé et commenté avec en plus une référence croisée sur l'utilisation des variables et des routines. Cet ouvrage sera tout spécialement indispensable à ceux qui travaillent en langage machine.

La découverte de l'Oric-1
par Daniel-Jean David - 176 pages

90,00 FF

Le but de ce livre est de découvrir l'ordinateur individuel Oric-1 et son langage Basic. Cette exploration est progressive grâce à l'élaboration de programmes de difficulté croissante, au cours de laquelle les notions nouvelles sont introduites. On aborde spécialement les points forts de l'Oric-1 : graphiques, sons, couleurs, horloge.

Oric-1 pour tous
par Jacques Boisgontier et Sophie Brébion - 176 pages

100,00 FF

Consacré à un ordinateur individuel tout nouveau et pourtant déjà très populaire, cet ouvrage d'initiation permet de découvrir les grandes qualités graphiques et sonores de l'Oric-1. Tout d'abord, vous assimilerez les notions fondamentales de la programmation (variables, tests, boucles...), puis vous parviendrez aisément, grâce aux nombreux exemples illustrés et aux programmes commentés, à écrire vos propres programmes.

52 programmes Oric-1 pour tous
par Jacques Boisgontier - 164 pages

100,00 FF

Voici un ouvrage dans lequel sont regroupés des programmes commentés et illustrés, permettant d'approfondir ses connaissances en Basic. Des exercices et des jeux mais aussi des programmes d'éducation et de gestion invitent à utiliser l'ordinateur Oric-1 et l'Oric Atmos avec beaucoup d'originalité et à en découvrir les fonctions particulières.

L'Oric à l'affiche
par Jean-François Sehan

90,00 FF

Une sélection de 20 programmes de jeux d'adresse, de réflexion et de hasard, utilisant les possibilités graphiques et sonores de l'Oric. Chaque programme est accompagné d'un organigramme, d'une liste de variables et d'une explication de chaque ligne Basic.

Boîte à outils pour Oric
par Michel Martin - 128 pages

35,00 FF

Les possesseurs d'Oric-1 ou Atmos trouveront dans ce livre de poche de petits programmes ludiques ou utilitaires (graphiques, dessins, musique etc.) écrits en Basic.

La découverte du Vic
par Daniel-Jean David - 176 pages

90,00 FF

Le Vic à l'affiche
par Jean-François Sehan - 144 pages

90,00 FF

Clefs pour le Vic
de Daniel-Jean David - 120 pages

90,00 FF

La pratique du Vic
de Daniel-Jean David - 176 pages

90,00 FF

Le livre du Vic
par Benoît Michel - 248 pages

110,00 F

Les systèmes à microprocesseurs
Daniel-Jean David - 128 pages

90,00 FF

Ce livre vous initiera aux conditions techniques de la révolution micro-informatique. Les différents circuits intégrés : microprocesseurs, mémoires, boîtiers d'entrées-sorties sont décrits et on montre comment les assembler pour former un système.
Les phases du traitement d'une application et du développement d'un système à microprocesseur sont décrites, notamment du point de vue logiciel (programmation en assembleur) et des choix à effectuer.

P.S.I. DIFFUSION BP 86 - 77402 Lagny-S/Marne Cedex FRANCE
Téléphone (6) 006.44.35

P.S.I. BENELUX
5, avenue de la Ferme Rose
1180 Bruxelles BELGIQUE
Téléphone (2) 345.08.50

En SUISSE
P.S.I. Suisse
Route Neuve 1
1700 Fribourg
Tél. : (037) 23.18.28
CCP 17-56-84

Au CANADA
SCE Inc. 65, avenue Hillside
Montréal (Westmount)
Québec H3ZIW1 Tél. (514) 935.13.14

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg à P.S.I. BENELUX ou pour la Suisse à P.S.I. Suisse.

NOM _____

PRENOM _____

Rue _____

Ville _____

Code postal |_|_|_|_|_|

DESIGNATION	Nbre	PRIX
TOTAL		