

**Standard  
AMSTRAD**

# CPC

REVUE DU STANDARD AMSTRAD ET SCHNEIDER

## JUMPING RUNNER

Les problèmes  
d'imprimantes

CONSTRUISEZ  
VOS RSX

**cadeau:  
VOCA 100**

de TECHNIMUSIQUE

Un logiciel gratuit  
de synthèse vocale  
sur les cassettes  
et disquettes de CPC



# loriciels

UNE RENTREE FRACASSANTE!!

# EDITORIAL

Vous êtes nombreux, très nombreux, à avoir envoyé des logiciels pour le concours. Au moment où nous mettons sous presse, le dépouillement est à peine commencé. Nous pensons pouvoir donner les premiers résultats dès octobre. Afin d'améliorer votre revue, nous avons placé, dans ce numéro, un sondage. Il doit nous permettre de tenir compte de vos désirs, de vos commentaires. Merci d'y répondre. Pour beaucoup d'entre vous, les vacances sont terminées. Alors, bon courage avec votre AMSTRAD.

La Rédaction

## SOMMAIRE

n°14

|                                                     |    |
|-----------------------------------------------------|----|
| Actualité                                           | 7  |
| Banc d'essai : Xrom, Ramdisc, Mirage Imager, Utopia | 8  |
| Vu en Angleterre                                    | 11 |
| Banc d'essai des logiciels de jeu                   | 12 |
| Logiciels d'Outre-Manche                            | 14 |
| Un éditeur se présente : D3M                        | 16 |
| La vitrine du libraire                              | 18 |
| Jumping Runner                                      | 19 |

|                             |    |
|-----------------------------|----|
| Cadeau : VOCA 100           | 22 |
| Colditz                     | 24 |
| La Musica                   | 26 |
| Catalogue amélioré          | 28 |
| Mémocarte                   | 32 |
| Les problèmes d'imprimantes | 38 |
| Trucs et astuces            | 44 |
| DésaCPC                     | 46 |
| Au cœur du 6128             | 51 |
| Au cœur du PCW              | 54 |
| AMSTRAD et Minitel          | 61 |
| Initiation à CP/M           | 66 |
| Comment construire une RSX  | 68 |
| Enquête lecteurs            | 72 |
| Initiation à l'assembleur   | 74 |
| Petites annonces            | 80 |

### CPC

Revue des utilisateurs  
du standard AMSTRAD  
Editions SORACOM  
La Haie de Pan  
35170 BRUZ  
RCS Rennes B319 816 302  
Tél.: 99.52.98.11  
Télécopieur : 99.57.90.37.  
Télex : SORMHZ 741 042 F  
CCP RENNES 794.17 V

Directeur de publication  
Sylvio FAUREZ

Rédacteurs en chef

Marcel LE JEUNE

Denis BONOMO

Photocomposition

FIDELTEX

Impression

Imprimerie de La Haye-Mureaux

Photogravure Noir et Blanc  
SORACOM

Photogravure Couleur

Bretagne Photogravure

Maquette

Patricia MANGIN

Jean-Luc AULNETTE

Secrétaire de rédaction et  
abonnements-ventes réassort

Catherine FAUREZ

Distribution NMPP

Dépôt légal à parution

Copyright 1986

Publicité

Patrick SIONNEAU

Fabienne JAVELAUD

IZARD CREATION

66, rue Saint Hélier,

35100 RENNES

Tél.: 99.31.64.73

### Vente au numéro

Rassort NMPP

Gérard PELLAN

Code APE : 5120

Distributeur en Suisse par :  
SEMAPHORE  
C.P. 32  
CH-1238 LA PLAINE  
Tél.: 022.54.11.95

et en Belgique par :  
COMPUTER MARKET  
150, rue Antoine Dorsaint  
1050 - BRUXELLES  
Tél.: 513.53.58

CPC est une revue mensuelle totalement indépendante d'AMSTRAD (GB) et d'AMSTRAD FRANCE

Commission paritaire en cours

Les articles et programmes que nous publions dans ce numéro bénéficient pour une grande part du droit d'auteur. De ce fait, ils ne peuvent être reproduits, imités, contrefaits, même partiellement sans l'autorisation écrite de la Société SORACOM et de l'auteur concerné. Les différents montages présentés ne pouvant être réalisés que dans un but privé ou scientifique mais non commercial. Ces réserves concernent les logiciels publiés dans la revue.

## CLUB AMSTRAD A AUXERRE...

Albert BAGOT à l'immense plaisir de vous faire part de la création d'un Club Amstrad à Auxerre. On se réunit tous les samedi matin à partir de 8 h. dans la salle bleue de la Maison de Quartier Sainte Geneviève et en plus, il y a une grande manifestation le 20 septembre avec exposition et un grand bal.

## ...ET A BOUGUENAI

Le Club Amstrad de Bouguenais quant à lui reprendra ses réunions chaque samedi à partir du 6 Septembre. Tél. 40 65 31 34.

## BRUAY EN ARTOIS

"Exposition Amstrad à Bruay en Artois le 13 Septembre de 10 à 18 heures à la Salle des Fêtes Municipale, place Marmottan, entrée gratuite. Nous ne recevons les inscriptions des exposants que par courrier adressé au Club Informatique Bruaysien, rue latérale, 62470 Calonne Ricourt.

## BOOMERANG !

Vous connaissez Boomerang ? Je n'ai pas dit le boomerang mais Boomerang ! La différence entre les deux réside dans le fait que l'un est un objet que l'on lance et qui vous revient dans la main, tandis que l'autre est un centre d'échange de logiciels. L'analogie entre les deux est simple : Vous envoyez deux logiciels au centre et il vous en revient deux de votre choix. Cet avantage ne s'adresse pour l'instant qu'aux amateurs de jeux. Mais si la cotisation est modique, il ne peut y avoir d'échange qu'entre originaux. Le catalogue est fourni et des nouveautés l'enrichissent régulièrement. Les délais pour l'échange sont eux aussi une référence. Pour plus de renseignements adressez-vous à Boomerang B.P. 585 - 74014 ANNECY cedex

## COOPÉRATION

Depuis le 1<sup>er</sup> Septembre, les produits ERE Informatique sont distribués en France par France Image Logiciels à la suite d'un accord de coopération commerciale entre les deux sociétés.

## LES NOUVEAUTES DE L'ETE

### CHEZ ERE INFORMATIQUE

**L'Ere du Verseau** : pour les amateurs d'astrologie les plus exigeants.

**SRAM** : jeu d'aventure se déroulant sur une planète au charme étrange, doté d'un analyseur syntaxique sophistiqué.

**Tobrouk** : dernier fleuron des wargames de PSS : l'affrontement en Libye entre les forces de l'Axe et les Britanniques.

**Sai Combat** : les arts martiaux vous passionnent ! 48 adversaires vous attendent.

Une nouvelle collection : **Le Livre d'Images** destinée aux tout jeunes.

**Quatre Saisons** : en est le premier volet : 4 jeux mariant le puzzle, le jeu de société et l'album à colorier.

**L'Animalier** : seconde page du Livre d'Images aborde l'apprentissage de la lecture.

### CHEZ LORICIELS

**Billy la Banlieu** : comment déjouer les pièges de loubards quand on est un banlieusard passionné de jeux vidéo. Un jeu d'arcade et d'aventure.

**Maracaibo** : un nom qui fait rêver. L'aventure au fond d'une mer hostile.

VOUS AVEZ VU MES DOIGTS ? !!!  
LA PROCHAINE FOIS, JE COMMANDE  
LA DISQUETTE CPC !



## La disquette du CPC Hors-Série n° 1 est DISPONIBLE...

Plus de 160 kilo-octets du meilleur cru, voilà ce que vous offre la disquette CPC Hors-Série n° 1 aux conditions habituelles. Vous trouverez le bon de commande en dernière page de ce numéro. Tous les programmes y sont, avec, en prime, une superbe adaptation de Elite Synco-pation de Scott JOPLIN par notre maestro Henri BITTNER. De quoi ajouter un look professionnel à vos propres programmes.

Nous sommes désolés pour les utilisateurs de 464, mais le volume des programmes était trop important pour être dupliqué sur cassette.

## CHERRY PAINT

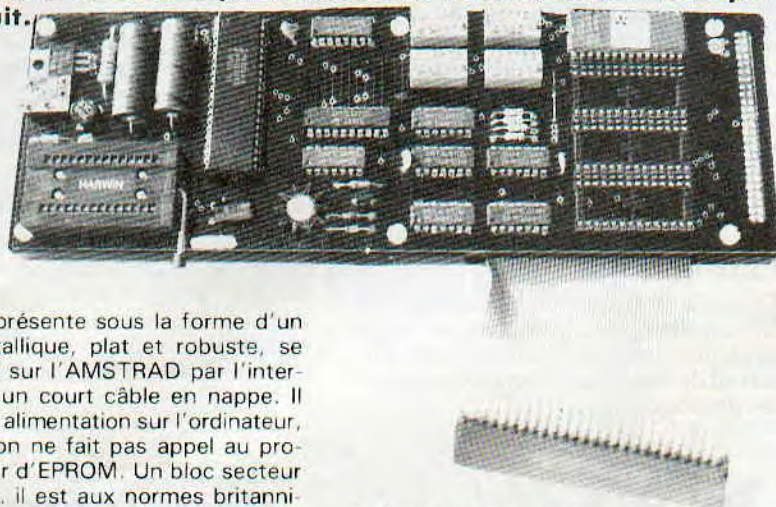
Un incident, indépendant de notre volonté, pour lequel nous vous prions de bien vouloir nous excuser, nous a contraint de différer au prochain numéro la suite du programme Cherry-Paint. Mais, rassurez-vous, il sera bien là, et en plus, ce sera le dernier article de la série.

## A NOS "ABONNES DISQUETTES"

ATTENTION ! La disquette Hors Série n° 1 n'est pas comprise dans votre abonnement. Il faut donc la commander séparément si vous désirez la recevoir. Rappelons que cette disquette HS1 contient tous les programmes publiés dans notre numéro Hors Série, sauf, bien sûr, les deux qui étaient sur la cassette d'accompagnement.

# XROM : ROM D'EXTENSION ET PROGRAMMATEUR

**T**IMATIC Systems met la programmation d'EPROM à la portée de tout utilisateur d'AMSTRAD, avec son extension XROM. Inutile de le cacher, nous avons été véritablement séduits par ce produit.



XROM se présente sous la forme d'un boîtier métallique, plat et robuste, se connectant sur l'AMSTRAD par l'intermédiaire d'un court câble en nappe. Il prélève son alimentation sur l'ordinateur, tant que l'on ne fait pas appel au programmeur d'EPROM. Un bloc secteur (attention... il est aux normes britanniques) est fourni avec ; c'est lui qui délivre le 5 V et le 25 V de programmation. Redressement, filtrage et régulation ont lieu à l'intérieur du boîtier. Un support à force d'insertion nulle est directement accessible de l'extérieur ; c'est là que vous mettez l'EPROM à lire, copier ou programmer. Un PIO 8255 assure l'interface avec l'AMSTRAD.

Le logiciel gérant le fonctionnement du programmeur est contenu dans une EPROM de 16 k-octets. A côté de cette EPROM, trois supports restent vides ; c'est là que vous pourrez installer vos ROM d'extension. XROM autorise l'implantation de 3 ROM d'extension, dont l'adresse est paramétrable au moyen de DIP switches. C'est donc très simple ;

vous sélectionnez vos meilleures RSX, les regroupez et programmez une EPROM avec. Il ne vous reste plus qu'à la mettre en place sur un des supports disponibles et voilà vos RSX directement accessibles.

Moniteur, désassembleur, fonctions "circle" ou "box"... tout est permis !

A la mise sous tension de XROM, plusieurs options sont disponibles, regroupées en menus auxquels on accède par l'intermédiaire des touches de fonction. Le programmeur n'accepte que les 2764 (8K) et 27128 (16 K). Il est vrai que le prix des EPROM a bien baissé, mais on aurait quand même aimé pouvoir programmer des 2532 (ou 2516). Le logiciel utilise de la RAM, bien sûr, donc attention aux programmes que vous pourriez avoir en mémoire....

Que peut-on faire ? Presque tout ! Bien entendu, on peut charger à partir d'un disque ou d'une cassette le logiciel à mettre sur EPROM. Si celui-ci est déjà sur EPROM, c'est encore plus simple ! Test de virginité, vérification après programmation, programmation lente ou rapide... rien ne manque.

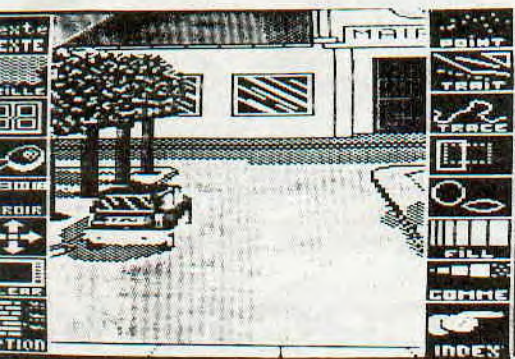
S'il apparaît évident que l'on puisse mettre sur EPROM des programmes en langage machine, il nous a paru très intéressant de pouvoir également y mettre des programmes Basic. Le logiciel de XROM se charge de tout...

Examen des ROM d'extension disponibles, dump ou édition de la mémoire, sortie sur imprimante sont autant de possibilités qui vous sont offertes. Un dernier utilitaire permet de travailler avec la disquette et informe sur les fichiers et RSX disponibles.

Nous nous sommes amusés à programmer une EPROM pendant le test ; le temps à vitesse lente, est de 6 minutes pour une 2764. Il faut 14 minutes pour une 27128. Ces temps sont raccourcis jusqu'à 2 et 5 minutes en mode rapide. XROM est une extension qui n'est pas encore importée en France et c'est dommage. Elle est vendue à moins de 800 F en Angleterre. Compte tenu de la qualité de sa réalisation (circuit imprimé à trous métallisés, boîtier robuste) et de ses possibilités, elle est fort séduisante. Pour tout renseignement complémentaire s'adresser à : TIMATIC SYSTEMS, FAREHAM MARKET, FAREHAM, HANTS PO 16 0LB. Tél.: 19.44..3.292.399.53.

## DESSINEZ AVEC GRAPHISCOP

Les logiciels d'aide au dessin ne sont pas toujours faciles à utiliser. Dans certains cas, il est bon de pouvoir reproduire facilement un dessin d'un papier sur l'écran.



GRAPHISCOP est une tablette graphique de conception originale, qui fonctionne un peu comme un joystick à l'envers. Un crayon, solidaire d'une potence, commande deux potentiomètres dont les positions donnent les valeurs en X et Y de la pointe. Une interface assure la liaison avec l'ordinateur. Un logiciel permet l'exploitation de la tablette. Nous avons eu l'occasion de tester l'ancienne version de GRAPHISCOP et n'avions pas été convaincus par la précision du positionnement. Ce défaut aurait, d'après les concepteurs, disparu sur la nouvelle version de ce matériel. Nous attendons d'en avoir un en prêt pour en faire un banc d'essai complet. Le nouveau prix a été fixé à 990 F. GRAPHISCOP, distribuée par MMC, tél. 16.1.42.56.12.82.

**Denis BONOMO**

# ESSAI DU LOGICIEL RAMDISC



**S**i vous hésitez à casser votre tirelire pour acheter un second lecteur de disquettes, ou si vous trouvez celui que vous possédez trop lent ou trop bruyant, le banc d'essai de RAMDISC va vous intéresser.

RAMDISC est un disque virtuel, c'est-à-dire qu'il permet d'utiliser la deuxième banque de 64 k de votre 6128 (ou une éventuelle extension 64 k sur un 464, 664) comme un lecteur de disque supplémentaire. Les 64 K endormis vont enfin se réveiller !

Pour démarrer le programme, il suffit de faire placer la disquette du programme dans le lecteur (l'aurez-vous deviné ?) et de taper RUN''RD. Au bout de quelques secondes apparaît le texte suivant : Commandes supplémentaires : IM, ICOPY, IFORMAT, ITYPE, le disque virtuel est alors installé.

Le formatage, lui non plus, ne traîne pas : 15 secondes par face ! Avis aux formateurs (jeu de mots !).

ICOPY transfère des fichiers d'un lecteur vers un autre. Il est donc possible de copier un fichier de A sur B, de A sur M, de M sur B, etc. Cet utilitaire autorise les jokers dans les noms de fichiers. On peut, par conséquent charger tous les fichiers d'une disquette vers le disque virtuel en début de travail avec ICOPY, ''A: \*.\*'', ''M'' puis de travailler entièrement sur le disque M. A la fin du travail, il suffit alors de sauvegarder le résultat sur

ITYPE affiche à l'écran le contenu d'un fichier.

L'utilisation de ce logiciel nous a paru particulièrement agréable en conjonction avec un autre programme : nous avons essayé l'assembleur DEVPAC et le traitement de texte AMSWORD. Les utilisateurs assidus de DEVPAC savent qu'il est pénible d'assembler à partir du disque (on se prend à guetter la petite lumière rouge !), avec RAMDISC, il assemble à la même vitesse que pour un texte en RAM.

Une autre surprise que nous vous avons gardée pour la fin : le logiciel n'est absolument pas protégé, vous pouvez donc le placer sur chacune de vos disquettes pour l'avoir sous la main à tout moment, vous pouvez aussi l'incorporer dans vos propres programmes et, par conséquent, formater ou copier des fichiers à partir de vos propres programmes en Basic. Impressionnant, non ?

Le seul reproche que l'on puisse faire à ce logiciel est qu'il faille taper RUN''RD à chaque mise sous tension de l'appareil (il fallait bien trouver quelque chose !).

Pour tout renseignement, téléphoner au 99.51.76.41.

| FICHIER TRANSFERE  | LECTEUR A    | LECTEUR M     |
|--------------------|--------------|---------------|
| 1 K Basic          | 2,4 secondes | non mesurable |
| 11 K Basic         | 5,0 secondes | 0,4 seconde   |
| 20 K Basic         | 6,6 secondes | 0,7 seconde   |
| Image d'écran 16 K | 5,9 secondes | 0,6 seconde   |

Pour vérifier la présence du lecteur M, nous tapons : IM puis CAT. Sur l'écran apparaît :

Drive M : User 0  
63 K Free

Nous essayons alors de sauver un petit programme Basic de 1 K sur M en tapant : SAVE''M:TEST, le programme est sauvegardé avant d'avoir eu le temps de relâcher la touche RETURN ! En fait, toutes les instructions du Basic Locomotive fonctionnent parfaitement sur le lecteur M (IERA, IREN, OPENOUT, CLOSEOUT, enfin toutes, quoi) avec une vitesse surprenante.

En plus, RAMDISC contient 3 utilitaires très pratiques, qui vous évitent d'avoir à recourir à CP/M pour la moindre copie de fichier ou le plus insignifiant des formatages.

Ces utilitaires font partie intégrante de RAMDISC, ils sont donc à tout moment prêts à servir.

IFORMAT permet de formater (qui l'eut cru), d'une part le disque virtuel, mais aussi des disquettes sur les lecteurs A ou B, et ceci au format Vendor ou DATA.

une disquette "physique".

## IBM PC POUR AMSTRAD

Des logiciels pour IBM PC tournant sur votre AMSTRAD... fabuleux, non ? Vous pensez tout de suite au fameux compatible que tout le monde attend... et bien, non ! PC-EMULATOR, tel est le nom de cette nouvelle extension, produite en Allemagne par KERSTEN et PARTNER. Utilisant MS-DOS, le PC-EMULATOR autorise l'accès à la bibliothèque de logiciels professionnels développés pour IBM.

Du côté hard, le PC EMULATOR est équipé d'un 8088, microprocesseur 16 bits, tournant sur 5 MHz. Il est doté de 512 Ko de mémoire. Deux versions sont disponibles : PCE-I avec un drive double face 360 Ko ; PCE-II avec 2 drives..., soit 720 Ko. Notons qu'il existe une version sans drive...

Disponible en Allemagne depuis la mi-août, PC-EMULATOR sera prochaine-



ment distribué en France. Il ne reste plus qu'à attendre...

Les prix pratiqués Outre-Rhin sont de 1295 DM pour le PCE sans drive, 1645 DM avec un drive, 1995 DM avec deux drives. Compte tenu du change, avec un DM à 3,25 F, des taxes de douane et marges d'importation, les prix, selon la configuration, s'étaleront entre 4500 et 7000 FF.

# MIRAGE IMAGER : DES TRANSFERTS FACILES

**Hé ! Ça y est... on nous a sorti un produit miracle, capable de transférer sur disquette pratiquement tous nos logiciels cassette protégés ! Son nom ? MIRAGE IMAGER...**

Petite boîte miracle, de la taille d'une classique et banale extension, MIRAGE IMAGER se branche sur le bus d'extension, en moins de temps qu'il n'en faut pour l'écrire. Sur 464, l'interface disque se connectera derrière. Pour les 664 ou 6128, il faudra y adjoindre une rallonge de bus de quelques centimètres. Par cette greffe, votre AMSTRAD se trouve tout transformé. Ces logiciels sur cassette, que vous n'utilisiez que rarement, à cause du temps de chargement, et que vous aimeriez bien pouvoir transférer sur vos disquettes, malgré leurs protections, vont enfin reprendre du service. Comment ? C'est magique ! Il suffit d'appuyer sur un petit bouton rouge, situé sur le boîtier de MIRAGE IMAGER, pour déclencher le processus... Un logiciel intégré dans la boîboîte fait le reste... Il est important de souligner que ce transfert peut s'effectuer n'importe quand ! Vous chargez donc votre jeu favori, laissez passer les écrans de présentation qui ne servent qu'à faire joli. Au début du jeu, une petite pression sur le bouton rouge, et c'est parti ! Un message affiche le copyright en haut de l'écran et sera suivi d'un menu dès que vous aurez appuyé sur ENTER. A partir de là, on peut sauvegarder le logiciel, dans l'état où il est (marrant, si vous avez atteint un score fantastique, vous pouvez le retrouver), recharger un programme préalablement sauvegardé par MIRAGE IMAGER, cataloguer la disquette, réinitialiser le système ou tout simplement lancer le programme en mémoire. C'est vraiment simple à utiliser !

Le programme se trouve transféré sur disque après avoir subi un traitement spécial (compactage). Les opérations de

transfert peuvent durer plusieurs dizaines de secondes.

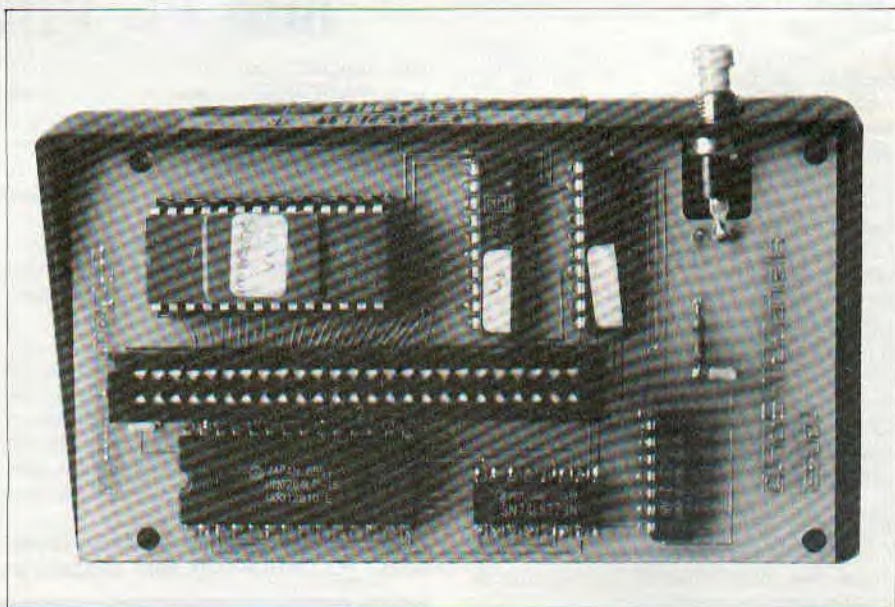
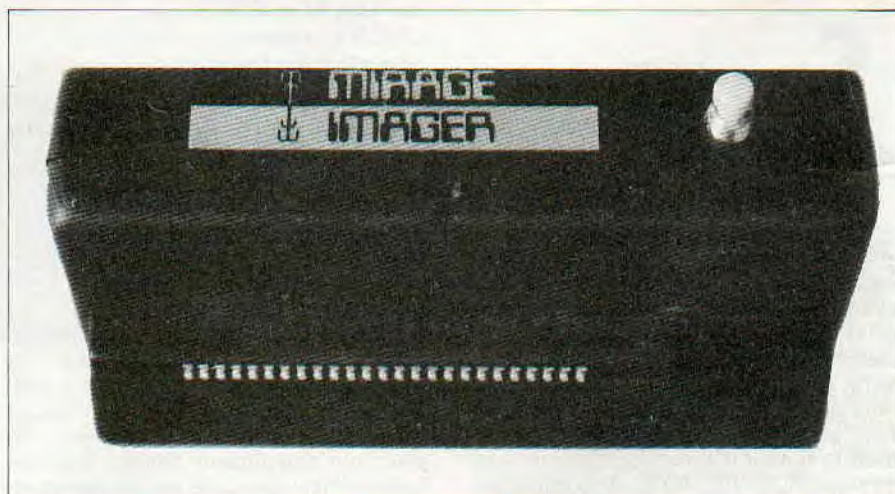
Je vois venir d'ici les petits malins, qui pensent se constituer ainsi une logithèque à bon prix... Pour fonctionner, le logiciel aura besoin de l'interface. La disquette seule ne suffit pas...

Dans la boîte, un circuit imprimé, tout propre et tout joli, supporte les deux connecteurs et les composants : de la mémoire et une ROM contenant le logiciel de transfert. Une notice de 4 pages, en français, explique comment utiliser MIRAGE IMAGER et donne la raison des différents messages d'erreur que vous

risquez d'obtenir en cours d'utilisation. A notre avis, MIRAGE IMAGER est un excellent investissement pour qui possède une importante logithèque sur cassette et veut en assurer le transfert sur disquette.

MIRAGE IMAGER n'est, à ce jour, vendu en France que par DUCHET Computers, société implantée en Angleterre. Rassurez-vous, si vous voulez téléphoner, ils parlent très bien le français ! DUCHET, tél. 19.44.291.257.80.

**Denis BONOMO**



# utopia

## au banc d'essai

La société britannique ARNOR est déjà bien connue des lecteurs de CPC par la présentation que nous avons effectuée de l'éditeur-assembleur MAXAM, unanimement reconnu comme étant l'un des outils les plus performants pour Amstrad. Au catalogue de cette société ne figurent que peu de produits, mais chacun d'eux fait référence dans la spécialité. Citons par exemple le traitement de textes Protext et UTOPIA qui fait l'objet de la présente étude.

Les logiciels Arnor présentent la particularité d'être livrés en Rom, ce qui procure une disponibilité immédiate lors de la mise sous tension de l'ordinateur. Utopia utilise les 16 kilo-octets d'une 27128. On y trouve une collection de fonctions utilitaires, près d'une cinquantaine au total, destinées à faciliter la tâche du programmeur et réparties en cinq catégories :

- Commandes utiles à la programmation en Basic permettant par exemple de déplacer des lignes de programme, de lister

des fichiers Ascii, de concaténer des programmes, de modifier des chaînes de caractères etc...

- Commandes de traitement de fichiers sur cassettes ou disquettes telles que Type, Dump, Vérify ou Copy.

- Utilitaires pour disquettes. Toutes les fonctions utiles à l'utilisateur de disquettes sont contenues dans la Rom et il n'est par conséquent plus nécessaire d'employer le disque CP/M pour formater ou copier un disque.

- Gestion de Roms extérieures permettant d'effectuer une sélection de l'application désirée lorsque plusieurs Roms sont connectées à l'Amstrad.

- Fonction Echo qui envoie à votre console toutes les informations apparaissant à l'écran vers un périphérique extérieur, l'imprimante par exemple.

Toutes ces commandes fonctionnent en Rsx et leur nom doit donc être précédé du bâton vertical traditionnel que l'on trou-

ve, rappelons-le sur la touche du clavier comportant le a commercial. Certaines d'entre elles peuvent être utilisées dans un programme.

À la mise sous tension, Utopia détecte la présence ou l'absence du ou des lecteurs de disquettes. Si un lecteur est présent, la séquence de touches CTRL et ENTER qui démarre habituellement le lecteur de cassette se transforme en RUN "DISC" ce qui permet un lancement quasi automatique de vos programmes. De plus, les touches de fonction du clavier sont redéfinies et des commandes usuelles deviennent directement accessibles, comme par exemple CAT ou MODE 2 ou encore LIST. La fonction HELP permet à tout moment d'obtenir à l'écran la signification et la syntaxe des commandes, ce qui permet de s'affranchir du manuel après une première lecture.

Utopia fonctionne sur les 3 modèles de CPC avec cependant une petite limitation sur le 464 pour ce qui concerne les paramètres liés aux commandes mais constitue malgré tout, à notre avis l'un des programmes utilitaires les plus intéressants pour Amstrad. Il suffit pour s'en convaincre de devoir s'en séparer après une longue période d'utilisation.

**Marcel LE JEUNE**



### LE COMPATIBLE PC ARRIVE

Par l'intermédiaire de notre source en Angleterre, généralement bien informée, nous avons appris qu'Amstrad présenterait son compatible IBM Pc à Londres le 1<sup>er</sup> Septembre. Le grand public pourra le voir au Personal Computer Show qui se déroulera également à Londres à partir du 3 Septembre jusqu'au 7 Septembre. Tout le monde espère que son prix de vente sera inférieur à 400 Livres, ce qui devrait nous le ramener à Paris autour de 5 000 Francs.

### AMSTRAD SPEAKS ENGLISH

Superior Software vient de commercialiser SPEECH qui est un logiciel de synthèse de la parole ne nécessitant pas d'électronique. Occupant 9 kilo-octets en ram, ce programme dispose de 49 phonèmes avec 8 intonations différentes. Tph : 19 44 15 32 45 94 53.

### DES OUTILS POUR LES PROS

The Software Toolshop est une boutique spécialisée dans la distribution d'outils de développement destinés au programmeurs confirmés. On y trouve des éditeurs-assembleurs, des langages tels que Pascal, Pilot, C, Fortran, Cobol et des compilateurs parmi lesquels, nous avons retenu ZBasic à 70 Livres. TSL Ltd, The Old School, Greenfield, Bedford MK45 5DE Tph : 19 44 525 71 82 71.



### AMSTRAD COMPUTER SHOW

Me croiriez-vous si je vous disais que Derek Meakin s'ennuie lorsqu'il n'organise pas de salon informatique ? Et pourtant ce doit être le cas puisque la 5<sup>e</sup> édition de son désormais célèbre Amstrad Computer Show aura lieu du 3 au 5 Octobre, toujours dans les salons de l'hôtel Novotel de Londres. Pour y aller, c'est facile : à partir de l'aéroport de Heathrow, prendre Picadilly line et descendre à Hammersmith. Le droit d'entrée est fixé à 2 Livres.



- 1- Get dexter
- 2- Spin Dizzi
- 3- Elite
- 4- Batman
- 5- Bombjack
- 6- The covenant
- 7- Tau ceti
- 8- Sorcery plus
- 9- Commando
- 10- Highway encounter
- 11- Rambo
- 12- Green beret
- 13- Who dares win 2
- 14- Fairlight
- 15- Marsport
- 16- Starion
- 17- 3D Grand Prix
- 18- Spellbound
- 19- Sheewo's world
- 20- Yie arg Kung fu

**Marcel LE JEUNE**

# BANC D'ESSAI DES LOGICIELS

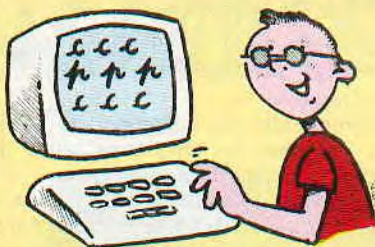


**"BILLY-LA-BANLIEUE"**  
(LORICELS)  
Arcade/Aventure

Voici le premier jeu d'arcade-aventure sentimental ; fans des sixties, sortez vos mouchoirs ! Billy-la-Banlieue, est un brave rocker sympathique. La banane tout juste un peu plus développée que la moyenne, le look discret et anodin, Billy n'en est pas moins un fan des bons vieux jeux d'arcade. Que de périples, que d'aventures pour arriver à sa boîte de jeu préférée : sa belle lui demande son cœur, Loulou et sa bande de loubards n'aiment pas Billy ; les flics non plus d'ailleurs... Un bon jeu dont le niveau de difficulté vous séduira. Le graphisme est soigné et l'intérêt du jeu est unique. Il vous prendra tout de suite, et après impossible de s'en défaire : plus de temps mort avec Billy-la-Banlieue. Nous ne possédions qu'une pré-série, sans notice, et nous ne connaissons pas toutes les finalités du jeu. Cependant, il nous a tout de suite enchanté. Pour conclure, n'oubliez pas que Billy-la-Banlieue est le sujet d'un concours dans AMSTAR, alors un conseil : courez vite vous le procurer, il est super. Décidemment, avec AMSTRAD, les auteurs et les éditeurs français on le vent en poupe.

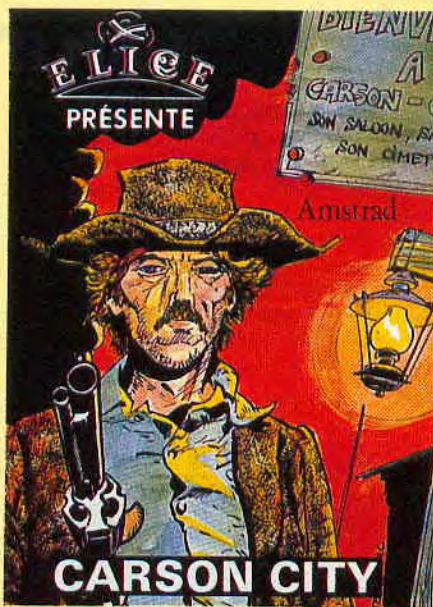
**CARSON CITY**  
Elice  
Aventure

Il était une fois (c'est toujours comme cela que commencent les contes de fée) un cow-boy solitaire et éloigné de chez lui... Après une longue poursuite dans les plaines du Texas, il se retrouva à Carson city où se cachaient six bandits qui avaient volé son bétail (et sa bétailière). Une longue enquête commençait alors... Ils étaient tous suspects, et Carson City n'était pas réputée pour son hospitalité. Dieu, que la vie de citadin est dure pour un gardien de vaches ; le pauvre ! Il accumulait dépenses inutiles sur dépenses inutiles. Manque de chance, il perdit ses gains du poker de la veille dans une malencontreuse erreur judiciaire. Il avait arrêté un honnête



citoyen et les portes, les bouches et les oreilles qui pouvaient s'ouvrir pour l'aider devinrent très hermétiques. On lui tira même dessus d'un toit, sans l'atteindre heureusement, mais son manque d'expérience lui fit perdre la trace de son agresseur.

Le graphisme n'est pas excellent mais le scénario est bien conçu, malgré quelques lourdeurs. Les amateurs du genre seront ravis par cette production d'une nouvelle société.



**"SRAM"**  
(ERE INFORMATIQUE)  
Aventure

Exceptionnel ! gigantesque ! formidable ! les mots manquent pour qualifier cette réalisation qui nous a laissé une très forte impression, tant le souci du détail a été poussé par les auteurs. SRAM est une planète comme bien d'autres, qui, inlassablement, poursuit sa course silencieuse. A sa surface, tout est calme et rien ne laisse imaginer le triste sort de ses habitants, en particulier celui de la famille royale, emprisonnée dans des conditions

innommables. Désigné par les opposants au régime, vous voilà transformé en libérateur : votre corps, dématérialisé, est transféré sur SRAM. L'aventure commence alors, mais que faut-il chercher, que faut-il faire, à qui se fier ? Seule l'expérience vous permettra de répondre à ces questions...

Des graphismes superbes, à la hauteur des possibilités de l'AMSTRAD, un analyseur syntaxique exceptionnel, qui comprend pratiquement toutes vos phrases, sont autant d'atouts qui feront de ce jeu un des best-sellers du moment et... un responsable de nuits blanches. Seul regret : il n'existe que sur disquette, mais c'est normal devant la richesse des situations proposées !

**SAPIENS**  
Loricel  
Arcade/Aventure

Imaginez-vous un instant vêtu d'une peau de bête, la barbe développée et criant des honga-honga incompréhensibles pour le commun des mortels. Un fou me direz-vous, non point du tout, vous êtes un Homo-Sapiens de la meilleure des espèces, un Homo-Sapiens Néanderthalensis. Pour ceux qui n'auraient pas compris, vous êtes un homme des cavernes, sans caverne...

Votre tribu vient de subir d'importantes pertes lors d'une guerre tribale ; la nourriture manque et il ne vous reste plus qu'à partir chasser sur des territoires qui vous sont inconnus. De cette dure expérience sortira un héros ou un cadavre. La gestion de votre personne s'effectue par l'intermédiaire d'un GEM (fenêtres) qui facilite grandement les opérations ! Le graphisme est soigné même s'il n'est pas extraordinaire. Cependant, la richesse du jeu pourrait combler tous les défauts sachant qu'il y a près de quatre millions de lieux différents. La partie arcade du jeu est très agréable puisque des dialogues, des échanges et des combats sont possibles avec les autres protagonistes que sont les hommes ou les animaux. Serez-vous à la hauteur de vos problèmes préhistoriques ?

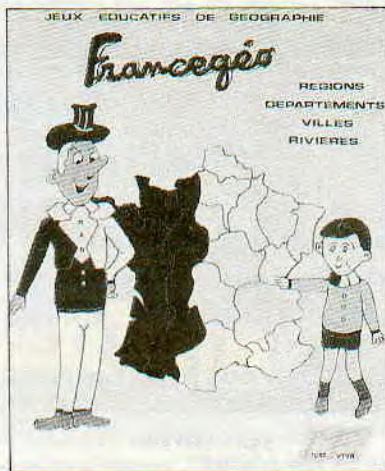


**" ROBBOT "**  
**(ERE INFORMATIQUE)**  
**Arcade**

Tombé en panne d'essence sur une planète nommée 10, vos ordinateurs ont détecté une présence d'énergie. Pour récupérer ce potentiel, votre fusée est équipée de trois robots. Ils ont tous une mission pré-définie, il ne vous reste qu'à les guider judicieusement vers la source d'énergie. XOR est un robot porteur, il peut supporter des hautes températures et transportera l'énergie jusqu'au vaisseau. RHO est un robot de liaison. Il sert de relai entre les robots et la fusée, et par conséquent il ne peut être éloigné de ses congénaires. SAM, enfin est un robot dépanneur car la haute technologie a, elle aussi, ses faiblesses. A vous de gérer votre expédition de manière victorieuse. ... Le graphisme est à la hauteur du scénario : Du béton, les couleurs sont superbes, mais le son est un peu énervant ; ce n'est pas gênant car ce jeu original allie adresse et coordination. Décidément Ere Informatique devient synonyme de qualité !

**" FRANCE GÉO "**  
**(MICRO FUTUR)**  
**Educatif**

Voilà un éducatif bien fait ! FRANCE GÉO se propose de rafraîchir vos connaissances en géographie de notre pays. Rafraîchir ou... accroître ! En cette période de rentrée nous allons pouvoir recenser les éducatifs dignes d'intérêt. Les éditeurs de logiciels, petits ou gros, doivent tenir compte de la percée d'AMSTRAD dans les écoles. ... au moins privées !



FRANCE GÉO aborde 4 thèmes : les régions, les départements, les villes, les rivières. Si je vous dis Mauriac, vous connaissez ? Non ! et bien vous avez tort parce que c'est une charmante (je le sais, j'y suis allé) petite ville du Cantal. Ne croyez pas que j'ai eu 20/20 en jouant avec FRANCE GÉO : il y a toujours un moyen pour se faire coller ! Essayez les questions d'hydrographie, pour voir ! Même un prof de géo se ferait avoir ! FRANCE GÉO est structuré en 2 parties : la première est une banque de données, capable de vous renseigner sur un sujet ; la seconde est un jeu qui mettra vos connaissances à rude épreuve ! On peut jouer avec ou sans la carte de France, ce qui complique encore les choses. Après MICRO GÉO, MICRO FUTUR récidive avec FRANCE GÉO, deux éducatifs de qualité.

**"ONE"**  
**(D3M)**  
**Arcade**

Original ! C'est le premier qualificatif qui s'applique à ce jeu où l'on envisage le conflit des générations avec le sourire. Un garmement affreux et un père à la main trop leste sont les acteurs de cette scène de la vie courante sur ordinateur.

Votre but, sale petit bonhomme, consiste à glisser des peaux de banane, au propre comme au figuré, au père en question. Tartes à la crème, rateaux, balles, tabourets sont autant d'objets anodins qui, entre vos mains, deviennent de dangereux pièges ! Ces objets, dissimulés dans le décor, doivent être ramassés et rangés dans vos dix poches pour être utilisés ultérieurement.

Néanmoins, faites attention aux représentations : en haut de l'écran une tête souriante, fâchée ou... amochée témoigne des coups que vous avez reçus pour vos exactions. Après 5 ou 6 bastonnades, le jeu revient en mode démo : c'est assez ! Avec un nombre de pièges variés et insolites, ce jeu s'avère être à la fois passionnant et original. De plus, animation et graphisme sont fort réussis. Un titre qui figurera en bonne place dans votre collection... sauf si, père de famille, vous craignez que vos enfants ne s'inspirent par trop de certaines situations !



**"CAULDRON II"**  
(PALACE SOFTWARE)  
Arcade

Vous vous souvenez de Cauldron ? Un jeu marquant par son animation... CAULDRON II est du même acabit : haut en couleurs, des animations et graphismes réussis, des illustrations sonores correctes ; tous les atouts pour en faire un bon jeu ! Le thème ? La sorcière et son balai d'or partent à la chasse aux citrouilles. Un véritable massacre. Une seule en réchappe : elle va se venger. Pour vaincre la sorcière, elle devra lui arracher un cheveu, trouver le chaudron et jeter un sort. Tout un programme (sans jeu de mots) surtout si l'on vous dit que monstres et spectres en tous genres cherchent à entraver sa vengeance. Une bonne récréation...

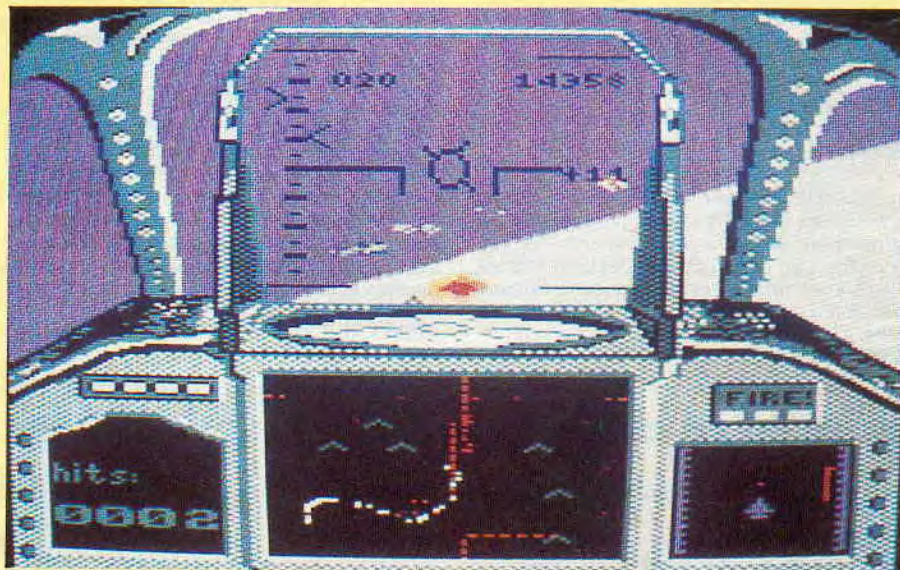


**"SHOGUN"**  
(VIRGIN GAMES)  
Arcade/Jeu de rôle

Le Japon et ses traditions... Honneur, religion, esthétique : 3 valeurs qui faisaient la raison de vivre des seigneurs du 17<sup>e</sup> siècle. Administrateurs et guerriers, ils rivalisaient pour devenir le chef suprême : le Shogun. Votre but dans ce jeu, consiste justement à atteindre ce rang. Vous pouvez vous placer dans la peau de différents personnages : paysan, noble, samouraï, dame(!) ou naufragé. Aptitudes et personnalité diffèrent en fonction du personnage choisi. Livré avec une notice en français, le jeu est doté d'une animation très réussie. Les situations sont assez variées et font appel à la force et à la stratégie. Un bon mélange de rôle et d'arcade !

# LES LOGICIELS D'OUTRE-MANCHE

**N**ous vous proposerons, dans cette rubrique, une sélection mensuelle de logiciels (jeux ou utilitaires) venus d'Angleterre. Ne croyez pas qu'il vous faudra traverser le CHANNEL pour vous les procurer : ils sont disponibles en France et pour la plupart importés par GUILLEMOT INTERNATIONAL SOFTWARE.



**"STRIKE FORCE HARRIER"**  
(MIRRORSOFT)  
Simulation de vol

Encore un simulateur de vol... et de combat aérien. Strike Force Harrier est d'ailleurs plus réaliste en combat qu'en simulation de vol proprement dite. L'avion que vous pilotez est un Harrier (vous savez, cet appareil extraordinaire, à décollage vertical). Assis dans le cockpit, vous avez, face à vous, le dispositif HUD (Head Up Display) : là sont regroupées les informations essentielles pour le vol : vitesse, altitude de l'avion, cap, paramètres de tir. Plus bas, l'écran radar et différents indicateurs : volets, train, freins, poussée moteur et fuel restant, l'écran de poursuite air-air et le tableau de pannes. Manette des gaz presque à fond, volets en position, freins lâchés, une légère action sur le manche : le décollage vertical a lieu. Si vous préférez un décollage plus conventionnel, c'est également possible. Guidé par votre radar, vous vous lancez à l'attaque des forces ennemis. Au sol, c'est des chars qu'il faudra détruire au canon (bonjour !) ou à la bombe. En l'air, c'est des avions ennemis (MIG 23, comme par hasard !) qu'il

faudra descendre.

La simulation est assez réaliste : oublier le train sorti à grande vitesse, et l'avion se met à vibrer. Un virage trop serré à faible vitesse, et l'altitude tombe. La phase d'atterrissage demande un peu de dextérité, bien qu'à la verticale, ce soit assez facile.

En combat, il faudra apprendre à se servir des missiles, des leurres et des contre-mesures. Descendre les ennemis n'est pas tout : encore faut-il pouvoir rentrer intact à la base ! Poursuivi par un missile, il faut faire preuve de sang froid et d'astuce...

La réalisation est très correcte. Chez le même éditeur, nous avons déjà testé SPITFIRE 40. Le graphisme interne du cockpit était meilleur sur ce dernier ; par contre, STRIKE FORCE HARRIER, donne une vue externe du sol (arbres, chars) et du ciel (appareils ennemis) assez réussie. Des bruitages accompagnent les tirs et les différentes phases de vol : réacteurs, alarme, missiles, explosions, etc. Assez technique pour repousser le débutant (bon nombre de touches du clavier sont utilisées), mais servi par une bonne notice, STRIKE FORCE HARRIER devrait ravir les passionnés de ce genre de simulation.

# A LA VITRINE DU LIBRAIRE



## "DICTIONNAIRE DE L'INFORMATIQUE" (LA VILLEGUERIN) 185 F

Les termes utilisés en informatique vous effrayent ? Soit, vous n'aurez plus d'excuse maintenant, car bien des définitions sont données par ce dictionnaire. En 180 pages, vous ne pourrez plus rien ignorer des définitions aussi différentes que celles d'ADA, d'IMPLEMENTER ou de TELECHARGEMENT. Beaucoup de termes nous viennent de l'anglais. L'origine de ceux-ci est proposée. L'expression anglaise est citée. Pour chaque terme important, nécessitant des développements, la définition est suivie d'explications concernant la conception et les applications. Une signification des sigles et acronymes est donnée en début d'ouvrage. Un lexique anglais-français et français-anglais le termine.

Pour que l'informatique ne soit plus une langue obscure et pour ne plus parler des choses sans les connaître, tout en recevant conseils et mise en garde, nous vous suggérons de vous procurer cet ouvrage. Bien sûr, son prix est un peu rébarbatif, mais qui veut la fin...

## "LE GRAND LIVRE DU BASIC (6128) (MICRO APPLICATION) Bernard KOWAL 149 F

Encore un livre dans la collection Micro-Application. Celui-ci est dédié au 6128. En 260 pages, il se propose de vous faire exploiter à fond les possibilités de votre machine. Débutant par une introduction à la programmation, le livre décrit ensuite le Basic de l'AMSTRAD, puis aborde la programmation avancée. Musique et graphisme sont traités dans des chapitres différents. Le lecteur de disquettes a inspiré l'auteur : une gestion de fichiers livres, utilisant le BANKMANAGER, est décrite. Quelques programmes sont proposés à l'utilisateur en guise de conclusion. En règle générale, leur description est effectuée, accompagnée d'une liste de variables.

Dans l'importante collection de Micro Application, où la valeur des ouvrages est assez inégale, ce livre nous paraît être un bon guide pour le programmeur hésitant.



## "L'UNIVERS DES PCW de Patrick LÉON" 224 pages 119 F. Editions SORACOM

La documentation concernant les ordinateurs PCW d'AMSTRAD est encore assez restreinte. L'auteur, Patrick LÉON, a entrepris un travail de foumi et propose au lecteur, intéressé par les richesses cachées de son ordinateur, d'apprendre à bien l'utiliser. Saviez-vous que l'on pouvait obtenir d'excellents graphismes avec un PCW ?

Neuf chapitres composent cet ouvrage. Le premier décrit l'environnement matériel et explique le rôle des différents composants. Passer de 256 K à 512 K est chose facile, mais le saviez-vous ?

Cartographie de la mémoire, commandes propres au CP/M 3.0, organisation du B DOS et du BIOS avec un passage en revue de toutes les fonctions et adresses stratégiques.

Vous désirez gérer des fichiers binaires : le BASIC Mallard ne le permet pas et bien, soit ! L'auteur propose les utilitaires de sauvegarde et de chargement. ... Tous les listings

assembleur fournis sont abondamment commentés.

Travailler sur disquettes demande parfois de pouvoir disposer d'utilitaires particuliers : écriture-lecture de secteurs, récupération de fichiers détruits par erreur etc. ... De même, un bon désassembleur est nécessaire pour travailler sérieusement. Patrick LÉON vous les offre !

Quant aux utilitaires graphiques, ils faisaient défaut jusqu'à présent : l'auteur exploite ici de nouvelles possibilités.

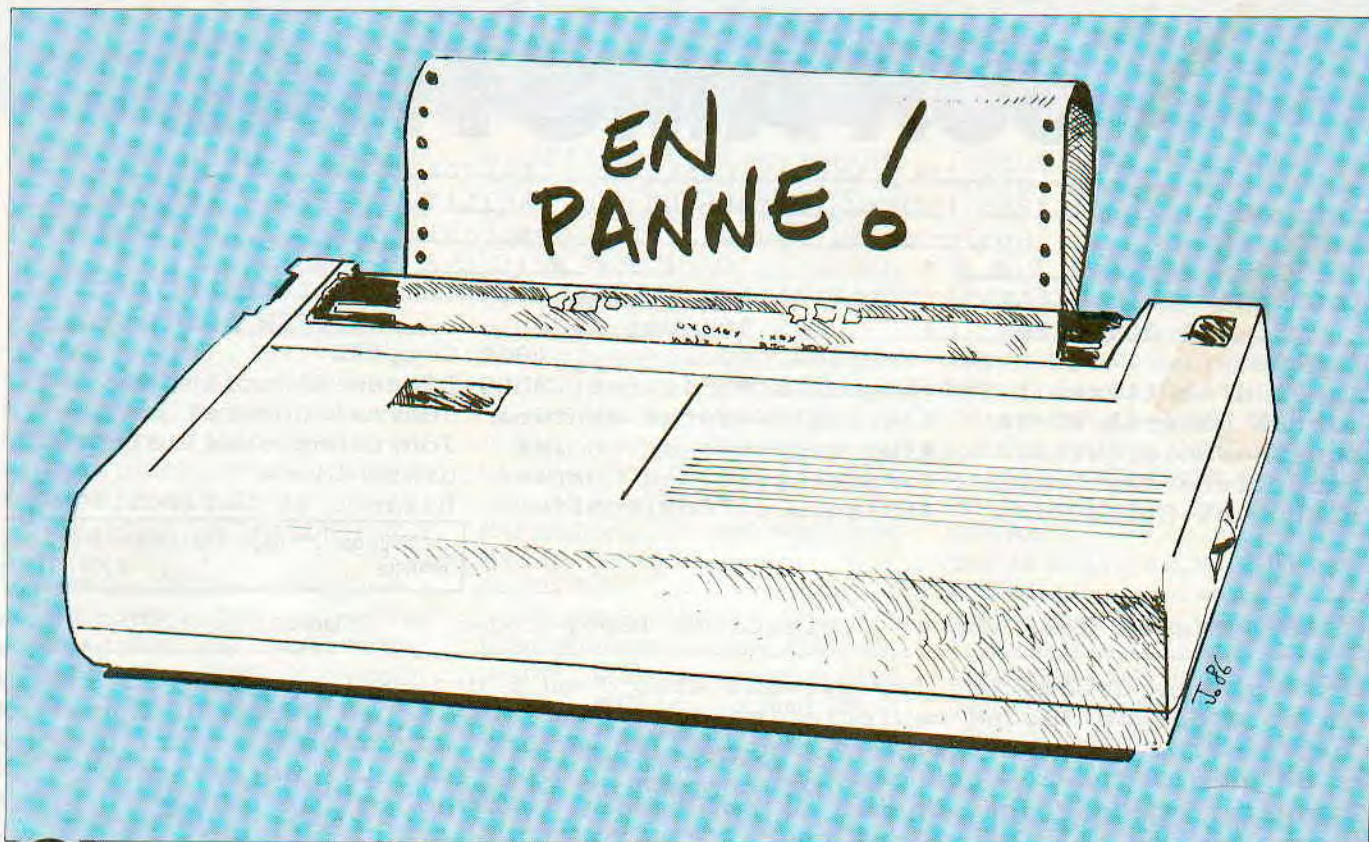
Un livre à posséder impérativement si vous voulez créer et travailler sérieusement sur votre PCW ou... tout simplement si vous êtes curieux !

## "PROGRAMMES POUR AMSTRAD" (CEDIC/NATHAN) X. CADOR et S. HOUEL 89 F

Vous en voulez ? en voilà ! quoi ? des programmes ! en tous genres : jeux, création, utilitaires... Un de plus, me direz-vous. C'est vrai, chaque éditeur y va de son recueil de programmes. Dans celui-ci, les listings sont clairement reproduits, accompagnés d'un mode d'emploi. Ce qui fait défaut, c'est un descriptif détaillé sur les différentes utilisations. Ceci aurait permis au lecteur de tirer un bien meilleur enseignement de cet ouvrage. Certains programmes sont, à la fois, originaux et instructifs, tel celui permettant d'avoir une vue du globe terrestre en 3D. D'autres sont totalement inutiles, telle cette horloge...

A feuilleter avant l'achat !





**C**et exposé n'est pas du tout destiné à "parfaire notre culture technique" pour "le plaisir", mais je constate que les utilisateurs de micro-informatique doivent absolument posséder des notions sur le fonctionnement de leur imprimante. Pourquoi ? Parce qu'ils ont des ennuis depuis le début ou qu'ils en auront un jour (c'est obligé...).

Si tout marche bien chez eux, c'est dans la vie professionnelle où ils seront confrontés à ce genre de "problèmes tout bêtes". Et ne comptez pas trop sur ceux qui ont réussi à vous vendre leur matériel... surtout quand ce ne sont pas les mêmes pour le micro et l'imprimante ! Deux minutes dans la notice plus deux minutes avec un tournevis valent mieux que deux semaines d'immobilisation. Je sais que vous allez dire "Ouais ! mais la notice est en chinois ! Qu'est-ce que c'est que ces LF, SLCT IN, BUSY, etc ?". Je m'en vais vous le dire, et vous verrez que c'est plus facile à comprendre que les commandes couleurs de l'AMSTRAD...

#### QU'Y A-T-IL DANS UNE IMPRIMANTE ?

C'est un petit micro-ordinateur. Il a une mémoire morte ROM assez importante et une mémoire vive RAM (petite). Il reçoit ses ordres par le câble de liaison, en guise de clavier ; sa sortie n'est pas un écran vidéo mais des moteurs électriques et des électro-aimants (les aiguilles de la tête d'impression). Une excellente nouvelle : les fabricants d'imprimantes respectent des normes, des standards. C'est le cas de la prise d'entrée (standard "parallèle CENTRO-

NIC") et de la ROM, à savoir les codes de commande du "Standard EPSON", la taille des caractères (80/ligne en PICA), six lignes par pouce, etc... C'est leur intérêt, car ils ne fabriquent pas de micro-ordinateurs, à de très rares exceptions près (EPSON, CANON, OLIVETTI,...). De plus, ils savent que les fabricants de micro-ordinateurs prennent un malin plaisir à faire des "variantes" sur les sorties pour imprimante. Alors ils prévoient des "SWITCHES" (petits commutateurs) à l'intérieur de leurs machines, pour qu'elles puissent desservir aussi bien un ORIC-ATMOS qu'un IBM-AT. Conclusion logique : l'utilisateur doit assurer l'adaptation micro à imprimante, à savoir trouver le câble de liaison, équipé de la prise farfelue côté micro, et modifier la position de certains switches dans l'imprimante. Dans le pire des cas (rare), modifier le câble de liaison côté prise micro.

#### LA TRANSMISSION DES DONNÉES PAR LE CÂBLE

La prise CENTRONIC a 36 broches, mais, rassurez-vous, elles ne sont pas toutes câblées ! Pour un AMSTRAD CPC, onze fils suffisent. Une imprimante ne reçoit que des nombres (des codes ASCII) ; exemple, si elle reçoit 65, elle imprime un A majuscule. Elle peut recevoir des

nombres de zéro à 255, c'est-à-dire un octet (8 bits) à la fois. L'AMSTRAD ne sait envoyer que sept bits, de ce fait notre imprimante recevra les codes ASCII de 0 à 127. Dans le câble, il y a un fil par bit, d'où le nom de transmission "parallèle". Ils sont numérotés de 1 à 8 ou de 0 à 7 selon les notices.

Sur ces fils, on a soit 5 volts, soit zéro volt, c'est ce qu'on appelle les niveaux haut et bas, ou encore 1 et 0.

Le micro peut "débit" ses codes ASCII bien plus vite que l'imprimante ne peut les écrire. D'accord ? Il faut donc qu'il y ait une entente entre eux deux, du genre "j'envoie ?", "oui", "arrête, j'en ai plein le buffer à imprimer", "ça y est, envoie la suite". C'est ce que les Anglais appellent le "handshake" (serrement de mains). Plusieurs fils sont consacrés à cette conversation (n° borne).

#### (1) STROBE

Le micro prévient qu'il envoie un octet.

#### (10) ACKNLG

L'imprimante répond qu'elle a bien reçu l'octet et qu'elle est prête à en recevoir un autre.

#### (11) BUSY

L'imprimante signale qu'elle ne peut rien recevoir parce qu'elle est en train d'imprimer, parce que vous avez appuyé sur le bouton "OFF-LINE" ou parce qu'elle est en état d'erreur.

#### (12) PE

(PAPER END), arrêt parce qu'en panne de papier.

#### (32) ERROR

Comme BUSY, mais dû à une erreur d'impression, OFF-LINE ou panne de papier.

# LES PROBLEMES D'IMPRIMANTES

Michel ARCHAMBAULT

## (36) SLCT IN

Généralement bloqué à zéro volt *soit* par le micro *ou* par un switch de l'imprimante (un niveau 5 volts bloque l'impression). C'est une commande venant du micro (rarement utilisée).

Voyons d'autres fils de commande :

## (14) AUTO FEED XT

Si à zéro volt (masse), saut de ligne automatique après chaque ligne (un piège AMSTRAD, on en reparlera).

## (31) INIT

Une réinitialisation du micro provoque une impulsion à zéro volt qui réinitialise l'imprimante. Equivaut à une extinction-allumage.

## (2 à 9) LES DATA

Les arrivées des bits des octets transmis.

## (17, 19 à 30, 33)

Retours masse.

**NOTE :** Les bornes masses sont nombreuses car chaque fil de commande a son retour masse : c'est une possibilité sécurisante pour les câbles très longs (plus de 3 mètres). Dans la pratique, on "ponte" ces bornes sur la prise 36 broches du câble, et retour masse par un seul conducteur dans le câble.

Une borne laissée "en l'air" doit être considérée comme étant au niveau *haut* (5 volts). Ainsi, il n'est pas obligatoire de câbler la 31 (INIT), elle sera au niveau 1 et l'impression fonctionne. En revanche, si SLCT IN reste en l'air, l'imprimante restera bloquée, sauf si un switch est prévu pour la mettre à zéro.

## LES SWITCHES INTERNES

Première précaution : notez sur la page de garde du manuel les positions établies (par défaut) par le constructeur. ETEINDRE.

Les rôles de ces switches diffèrent beaucoup d'un modèle à l'autre. Ils fixent certains paramètres d'impression, ceux que vous aurez dès la mise sous tension pour faire un simple LIST # 8. On trouve souvent :

- la forme du zéro, barré ou non,
- caractères "normaux" PICA ou condensés,
- buzzer actif ou non,
- longueur des feuilles (11 ou 12 pouces),
- détecteur de fin de papier (OUT pour les feuilles séparées),
- SLCT IN à zéro ou non,
- LF (automatic LINE FEED) saut de ligne automatique. A inverser si le papier n'avance pas ou au contraire si avance de deux lignes au lieu d'une,
- SKIP-OVER, saut de 1 pouce en bas de chaque page,
- choix de nationalité du jeu de caractères. Pour AMSTRAD CPC, mettez-vous en "ENGLAND" ou à la rigueur en "US". Il faut toujours avoir la même nationalité que le clavier (pour les listings).

Ne manœuvrez jamais les switches avant d'avoir coupé l'alimentation. Non seulement c'est une prudence, mais le nouvel effet n'apparaîtra souvent *qu'après* une remise sous tension.

Inscrivez votre choix des positions de switches sur une feuille qui sera *scotchée* à l'intérieur du manuel.

## LA HIERARCHIE DES COMMANDES

Vous remarquerez que certains ordres ou paramètres peuvent provenir à la fois des switches, des codes EPSON dans le programme Basic ou par des signaux sur certains fils du câble. Qui prime ?

- 1 — la commande par fil spécial,
- 2 — les codes EPSON transmis,
- 3 — les switches de l'imprimante.

Des exemples :

— Elle est switchée en caractères anglais, votre programme Basic envoie :

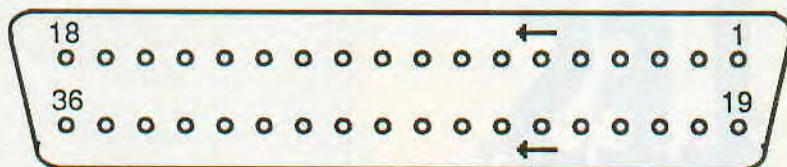
PRINT #8,CHR\$(27);"R";CHR\$(1);

Elle imprimera les caractères français (é, è, à, etc.).

— C'est l'AMSTRAD qui provoque un saut de ligne après une ligne, donc le switch LF est OFF. Mais si la borne n° 14 de la prise CENTRONIC est reliée à la cosse 14 du connecteur AMSTRAD (voir page A5.2 du manuel CPC 464), qui, elle, est à la masse, vous aurez un saut de ligne supplémentaire. Le switch LF est sans effet. Il faut donc éviter cette liaison électrique.

Rappelez-vous que les commandes reçues par codes EPSON restent actives dans la RAM de l'imprimante, sauf si vous envoyez un contre-ordre ou "vidage buffer" (ESC "a commercial"), ou si vous coupez l'alimentation ; même si vous éteignez l'AMSTRAD CPC...

A ce propos, pour faire un listing en 40 caractères de large, vous avez le choix entre le code EPSON ESC "Q" ou la commande WIDTH 40 qui, elle, va rester active uniquement dans la RAM du CPC ! OK ?



Brochage du socle femelle  
de l'imprimante vu de l'extérieur.



Sortie imprimante du CPC vue de l'extérieur.  
Les cosses 18 et 36 n'existent pas.

## LE CÂBLE D'IMPRIMANTE

Comparez le plan de brochage de la prise CENTRONIC dans la notice de votre imprimante (il y a tout !) et le brochage du connecteur arrière de votre micro. Alors là, il n'y a pas grand chose ! On a les 8 fils de DATA (sept sont vraiment actifs), le STROBE et le BUSY. Toutes les autres sont soit non connectées ("NC") ou à la masse. Il suffit donc de 10 + 1 fil de masse = 11 conducteurs. Les numéros du connecteur AMSTRAD correspondent aux numéros de la prise CENTRONIC 36 broches.

Donc, grande conclusion : un câble d'imprimante est *spécifique à un micro* (il devrait être livré avec), il convient alors à toutes les imprimantes. Sur ces dernières, seuls les switches seront peut-être à modifier. Les exceptions sont rares : il s'agit généralement de certains modèles "dédiés" à certains micro américains

(Commodore, Tandy, Hewlett-Packard, etc.).

Dernier petit conseil : à l'aide d'un marqueur noir à l'alcool, inscrivez "HAUT" sur le dessus de la prise côté CPC...

## PANNES, ENNUIS ET REMEDES

Cette petite liste de cas authentiques contient certainement le "problème" (actuel ou futur...).

- 1) Rien ne marche :
  - la prise côté CPC est à l'envers,
  - le switch SLCT IN est sur OFF,
  - présence d'une "interface série" optionnelle ; enlevez-la,
  - la mécanique d'entraînement du ruban (dans la cartouche) s'est bloquée. Changez.
- 2) Seule une partie des caractères est imprimée, le haut ou le bas :
  - le ruban sort de la cartouche plié dans

le sens de la longueur,

- la cartouche ou le ruban sont mal positionnés dans leurs guides,
- le plastique souple transparent à dix conducteurs, reliant la tête d'impression au circuit imprimé, est détérioré ou mal engagé à une extrémité,
- débris de papier coincé devant la tête.

- 3) Tout s'imprime sur la même ligne :
  - switch LF.

- 4) Saut de deux lignes au lieu d'une :
  - switch LF,
  - empêchez que la borne 14 de l'imprimante soit en liaison avec la cosse n° 14 côté PC.

- 5) Elle n'imprime qu'en italiques :
  - switch (rare),
  - la borne n° 36, côté imprimante, est "en l'air". Mettez-la à la masse, soit par la prise du câble, soit par l'intérieur.

- 6) Bourrage du papier perforé, vers la deuxième ou troisième page...

- l'entraînement par friction est en service. Il ne faut pas le combiner avec celui par traction ! (sauf des fois pour amorcer le papier).

- 7) En listing, des chaînes longues présentent des coupures avec retour à la ligne :
  - Sur l'AMSTRAD, programmez un WIDTH plus petit ou plus grand que celui du papier (80 en général).

- 8) Interligne plus petit entre la première et la deuxième ligne :

- Le positionnement du papier à la main n'est pas "tombé en face". Pressez le bouton OFF LINE puis avancez ligne par ligne par le bouton LF (Line Feed) ; bouton ON LINE et imprimez.

- 9) Saut de page de douze pouces au lieu de onze :

- switch,
- cas des EPSON FX85 et FX105 (pas de switch !) : sur le circuit imprimé EPSON, au milieu, vers l'avant, se trouvent deux pastilles étamées marquées "J5". Mettez-les en contact en soudant un petit fil (c'est délicat). Vous serez alors en onze pouces.

- 10) Certains caractères sont différents qu'à l'écran (#, @, ...):

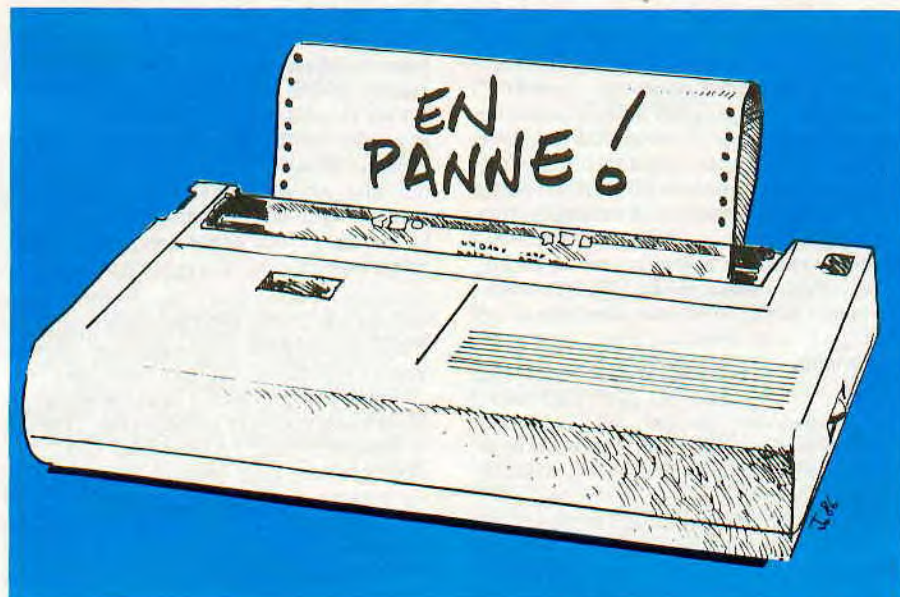
- l'imprimante n'est pas switchée en clavier anglais. Attention : il y a quelques menues différences entre les QWERTY US et anglais.

- 11) Enfin, la plus classique ! Impression super bizarre :

- des codes de commande venant du programme précédent sont restés actifs dans la RAM de l'imprimante ; il suffit d'éteindre et de rallumer l'imprimante.

## CONCLUSION

Toutes ces notions suffisent pour résoudre 98 % des problèmes. Pour les 2 % qui restent, la clé du mystère est presque toujours dans la notice... ■



# TRUCS & ASTUCES

**Christophe MALLET**  
**33720 PODENSAC**

M. Guy PICORE, 46 La Chappelle Marival demandait dans CPC n° 6 de décembre dernier :

"Comment récupérer un programme dont le chargement est incomplet, du fait de l'apparition d'une erreur de type 'Read Error a' (ou b)?"

Je pense avoir trouvé la solution qu'il attendait, ainsi que beaucoup d'Amstradistes.

Mais je tiens à signaler que cette solution ne s'applique qu'à des fichiers Basic, protégés ou non, et qu'elle n'a été testée que sur CPC 464 avec un lecteur de cassettes.

Pour récupérer des fichiers Basic protégés :

1) Faire :

CALL 0 (RESET)  
POKE &AC03,&AE  
POKE &AC02,&45  
POKE &AC01,&32

2) Ecrire : LOAD "nom du programme"  
3) Dès que le "Read Error" apparaît, laisser finir le bloc puis appuyer sur ESC (deux fois valent mieux qu'une).

4) Ensuite, il faut s'armer de patience en pensant que l'on va récupérer tout le fichier jusqu'au moment du "Read Error".

Faire : POKE &170,X puis LIST. X est un nombre qui va varier de 0 à 255.

Comme quelquefois la liste du programme (la liste correcte d'origine, car vous allez vous apercevoir que la valeur de X fait changer la présentation et le contenu de la liste) tarde à venir (il ne faut pas oublier qu'il y a 255 solutions), je vous conseille de faire :

KEY 138,"POKE&170,"  
KEY 139,CHR\$(13)+"LIST"+  
CHR\$(13)

Comme ça, vous appuyez sur le point du pavé numérique, vous rentrez votre chiffre X et vous appuyez sur la touche ENTER du pavé numérique pour voir le résultat.

5) Dès que vous avez la liste correcte de votre programme, faire :

POKE &B8D1,0                   accélération  
POKE &B8D2,29               du speed write

suivi de : SAVE "nom du programme",A  
Il ne faut surtout pas oublier de faire suivre le SAVE du A car si l'on fait un enregistrement normal du programme récupéré, l'ordinateur ne veut pas enregistrer et sauvegarde juste le titre du programme.

6) Le programme fini de sauvegarder, faire :

CALL 0  
LOAD "nom du programme récupéré par  
SAVE...A"  
POKE &B8D1,0  
POKE &B8D2,29  
SAVE "nom du programme"

Vous vous demandez peut-être pourquoi la deuxième sauvegarde (simple) est nécessaire. Et bien, c'est pour se débarrasser du bloc en trop que l'Amstrad traîne pendant la récupération. En effet, vous allez vous apercevoir que le fichier final fait un bloc de moins que le fichier sauvegardé après récupération.

7) Le tour est joué, vous n'avez plus qu'à éteindre votre ordinateur, à le rallumer, à charger votre fichier et à taper la fin de celui-ci pour pouvoir à nouveau le réutiliser au plus vite.

Pour les fichiers Basic non-protégés, il faut suivre le même chemin en laissant de côté la première partie.

ATTENTION : Ma solution ne permet pas de récupérer la totalité du fichier, elle sert juste à repiquer la partie avant le Read Error, de manière à ne pas avoir à retaper tout le programme si le Read Error apparaît dans les derniers blocs de celui-ci.

Bonne récupération.



**Louis GENTY**

## **COPYCHAR\$ SUR CPC 464 UNE ROUTINE POUR FORMER LES CARACTERES...**

La fonction COPYCHAR\$, permettant d'extraire le code du caractère de l'écran à l'emplacement du curseur, n'existe que sur les CPC 664 et CPC 6128.

Si un programme contient cette instruction, elle ne sera pas acceptée par l'interpréteur du CPC 464. Pour la remplacer, voici une petite routine en langage machine :

A200 CALL &BB60  
A203 LD (&A207), A  
A 206 RET

Pour la charger la première fois :

10 MEMORY &A1FF  
20 FOR I=&A200 TO A206  
30 READ B\$:U=VAL("&" + B\$):  
POKE I,U:NEXT I  
40 DATA CD, 60, BB, 32, 07, A2, C9  
RUN

Pour la sauvegarder en binaire :

SAVE "COPYCHAR, B, &A200, &8

Pour la charger par la suite :

LOAD "COPYCHAR"

Pour l'utiliser :

Rempacer (par exemple)  
A\$=COPYCHAR\$ (#0) par  
CALL &A200:A\$=CHR\$(PEEK(&A207))  
Cette routine peut également être placée dans une fonction RSX pour plus de souplesse d'emploi.

## **DU CAPS LOCK AU SHIFT LOCK**

Pour forcer les états CAPS et SHIFT (accessibles à partir du clavier par des touches spécialisées), il est possible d'utiliser, par programme, les deux variables système qui contrôlent ces états. ATTENTION : Si vous faites un POKE en mode direct pour forcer le SHIFT, vous ne pourrez plus revenir en mode non shifté, la virgule n'étant plus accessible. Ceci est donc à utiliser en mode programme.

Pour forcer l'état CAPS ou SHIFT, il suffit d'écrire &FF à l'adresse correspondante. Un 0 fera revenir à l'état initial.

|            | 464  | 664  | 6128 |
|------------|------|------|------|
| CAPS LOCK  | BE48 | B632 | B632 |
| SHIFT LOCK | BE47 | B631 | B631 |

Exemple pour un CPC 664 : POKE &B632,&FF force le CAPS LOCK et POKE &B632,0 assure le retour en minuscules.

## **CALAMITES**

### **PROGRAMME MONITEUR CPC HORS SERIE**

Deux petits problèmes ont été rencontrés sur le programme "MONITEUR", livré sur notre cassette accompagnant le Hors-Série n° 1. Voici le moyen pour les résoudre.

- 1 — Utilisation option P (point d'arrêt)
- 2 — Positionnement Himem sur CPC 464.

### **MODIFICATIONS A APPORTER**

A — Version CPC 664 et 6128.

— Tapez LOAD "monit" puis sauvegarder sur une autre cassette par save "monit".

— Remettre la cassette CPC puis taper RUN et ENTER.

— Choisir option modifiée (touche M).

— Entrer adresse 42DF puis ENTER.

— Entrer les codes suivants : B7, CA, 79, 40, 2A, 0C, 40, E9, 21, 32, 40. ENTER.

— Sauvegarde par option S (indiquer support K ou D) puis entrer 4000,4D36 ENTER.

— Indiquer le nom du fichier :

MONITHEX puis ENTER.

— Mettre en place la cassette comportant MONIT.bas, mettre en position enregistrement, puis répondre (O)ui et ENTER.

Le programme sera ainsi sauvegardé dans sa bonne version.

B — Version CPC 464

Avant de sauvegarder, ajouter les modifications suivantes :

Prendre option M puis ENTER.

— taper 4600 et ENTER,

— puis 7B, / 463F et ENTER,

— 7B., et ENTER.

Lancer la sauvegarde comme précédemment. ■

# 6128

## AU CŒUR DU

Jean-Pierre MANDON

**N**ous allons, ce mois-ci, étudier quelques routines des ROM des CPC qui seront d'une utilité certaine pour les exploitants. Nous utiliserons pour cela les résultats du précédent article concernant les connexions de ROM externes.

### SORTIE DES CHAINES DE CARACTERES

#### Routine de la ROM Basic

Une première routine applicable au traitement de chaînes de caractères est disponible dans la ROM Basic à l'adresse C37D (C337 sur le 464). Cette routine permet de sortir une chaîne de caractères sur la position courante du curseur. Voici la marche à suivre pour l'utiliser :

|                       |                                                  |
|-----------------------|--------------------------------------------------|
| 10 Ent \$             |                                                  |
| 20 Call #BA5F         | Connexion de la ROM Basic (ROM haute).           |
| 30 Ld HL,CAR          | HL pointe l'adresse de début de la chaîne.       |
| 40 CALL #C37D         | Appel de la routine d'affichage de la ROM Basic. |
| 50 CALL #BA66         | Déconnexion de la ROM Basic.                     |
| 60 Ret                | Retour                                           |
| 70 Car:Defm 'AMSTRAD' | Chaîne de caractères à sortir sur l'écran        |
| 80 DefB 00            | Fin de la chaîne.                                |

#### COMMENTAIRES

Le registre HL contient l'adresse du premier caractère (ou octet) de la chaîne, et la routine d'affichage est interrompue lorsqu'elle trouve un octet égal à 00. Il convient donc de placer cette valeur en fin de chaîne pour revenir au programme.

Pour placer à l'endroit voulu le texte, il sera indispensable de positionner le curseur en plaçant les paramètres lignes et colonnes dans la mémoire. Les vecteurs "position du curseur" sont aux adresses B726 et B727.

B726 : numéro de ligne

B727 : numéro de colonne.

Ils pourront être facilement modifiés à partir du programme (sur le 464 : B285, B286).

#### APPLICATION

Nous donnons un exemple commenté permettant d'écrire "AMSTRAD" sur la dixième colonne et la quatrième ligne.

|                          |                                                       |
|--------------------------|-------------------------------------------------------|
| 10 Ent \$                | Programme d'affichage.                                |
| 20 Ld A,#04              | La valeur ligne est placée dans l'accumulateur.       |
| 30 Ld (#B726),A          | Positionnement du curseur sur la 4 <sup>e</sup> ligne |
| 40 Ld A,#01              | Valeur colonne dans l'accumulateur.                   |
| 45 Ld (#B727),A          | Positionnement 10 <sup>e</sup> colonne.               |
| 50 Call #BA5F            | Connexion de la ROM Basic (ROM haute).                |
| 60 Ld HL,Car             | HL pointe le début de zone "chaîne de caractères".    |
| 70 Call #C37D            | Appel de la routine d'affichage.                      |
| 80 Call #BA66            | Déconnexion de la ROM.                                |
| 90 Ret                   | Retour                                                |
| 100 Car : defm 'AMSTRAD' | Chaîne de caractères.                                 |
| 110 DefB 00              | Fin de chaîne.                                        |

#### ROUTINES EN RAM FAISANT APPEL A LA ROM SYSTEME

En regardant de plus près la routine d'affichage de la ROM Basic, on s'aperçoit qu'elle fait appel à la ROM système à travers l'adresse BB5D de la RAM (idem sur CPC 464 et 664).

La principale différence est que la gestion de la chaîne n'est pas

prise en compte et que la valeur à afficher doit être placée dans l'accumulateur. Dans un premier temps, regardons comment l'utiliser, ensuite nous verrons comment l'associer avec d'autres routines.

|               |                                              |
|---------------|----------------------------------------------|
| 10 Ent \$     |                                              |
| 20 Ld A, #42  | Charger l'accumulateur avec le caractère "B" |
| 30 Call #BB5D | Afficher "B"                                 |
| 40 Ret        | Retour                                       |

Toutes les remarques que nous avons faites précédemment concernant la position du curseur restent valables ici.

## EFFACEMENT D'ECRAN

Cette routine est l'équivalent du CLS en Basic. Elle est accessible à l'adresse BC14 (idem 464 et 664), la seule précaution à prendre concerne la gestion du curseur. En effet, après appel de cette routine, l'écran est effacé, mais le curseur reste dans la position courante. Il convient donc d'assurer en permanence la gestion du curseur.

## SAISIE DE CARACTERES

Les deux derniers accès que nous vous proposons pourront être associés aux précédentes routines. Nous savons désormais écrire une chaîne de caractères, un caractère, vider l'écran, reste à lire clavier.

Pour cela, nous avons à notre disposition deux accès, l'un en BB18 et l'autre en BB1B.

**BB18 :** Cette routine fait une boucle tant qu'une touche du clavier n'a pas été enfoncée.

**BB1B :** Saisit le numéro de touche frappée s'il y en a une.

Le numéro de touche est placé dans l'accumulateur, ce qui permet d'aligner les routines étudiées jusqu'ici :

|               |                                                |
|---------------|------------------------------------------------|
| 10 Ent \$     |                                                |
| 20 Call #BC14 | Vide l'écran (position curseur conservée).     |
| 30 Call #BB18 | Attend qu'une touche soit enfoncée.            |
| 40 Call #BB1B | Place le numéro de touche dans l'accumulateur. |
| 50 Call #BB5D | Affiche la valeur placée dans l'accumulateur.  |
| 60 Ret        | Retour.                                        |

## COMMENTAIRES

Nous avons maintenant toutes les routines nécessaires pour réaliser un éditeur de texte performant. Vous aurez remarqué que les routines étudiées ce mois-ci n'ont pas été désassemblées, la plupart faisant appel soit à la ROM Basic, soit à la ROM système, il aurait été fastidieux de s'attaquer au désassemblage des ROM, et notre revue n'aurait pas été suffisamment épaisse pour publier les résultats !

Sachez néanmoins que ces routines font appel à une instruction restart du Z80, en l'occurrence RST 8 (saut à l'adresse 8) et que ceci permet de connecter la ROM système entre 0 et 3FFF. L'adresse à appeler est placée dans les deux octets suivant cette instruction. Pour calculer l'adresse de saut, ne pas prendre en compte les bits A14, A15 car ils servent pour la connexion de la ROM uniquement.

## CONCLUSION

Au fur et à mesure de cette description, nous vous fournirons des programmes d'utilisation de ces routines. Nous les écrivons et attendons les vôtres !



# CP/M INITIATION A

Francis VERSCHURE

**L**e mois dernier, nous avons terminé notre étude des commandes externes de CP/M et CP/M Plus se rapportant à la gestion des fichiers en général.

Nous voyons ce mois-ci une possibilité qui n'est offerte que par CP/M Plus et qui est désignée par le terme "redirection des entrées/sorties".

Les entrées désignent les organes d'entrée d'un ordinateur et, dans le cas de notre CPC 6128, ils se résument au clavier.

Les sorties désignent les deux organes de sortie possibles, c'est-à-dire l'écran et l'imprimante.

Lorsqu'un programme s'exécute, il reçoit en entrée des données au clavier et donne un résultat sur l'écran ou sur l'imprimante. La redirection des entrées/sorties permet donc à un programme de recevoir ses données en entrée d'un fichier comme si celles-ci étaient tapées au clavier et de stocker ses résultats dans un fichier au lieu ou en plus de l'écran.

Deux commandes externes de CP/M Plus permettent cette redirection.

## GET

Cette commande permet la redirection des entrées. En voici la syntaxe :  
GET [CONSOLE INPUT FROM FILE Nom de Fichier] [ECHO | NO ECHO | SYSTEM]

GET [CONSOLE INPUT FROM CONSOLE]  
Le CONSOLE INPUT FROM est en général omis.

La commande GET FILE signale à CP/M Plus que les entrées de la commande système ou du programme, qui sera tapé juste après à la console, seront prises dans le fichier spécifié.

Les entrées sont prises dans le fichier tant que le programme ou la commande s'exécute ; après cette exécution, les entrées sont de nouveau prises au clavier. Si on est arrivé à la fin du fichier d'entrées avant la fin du programme ou de la commande, même chose, on revient au clavier.

L'option ECHO est implicite et signifie que les entrées ne sont plus affichées à l'écran, mais cela ne change en rien les affichages de la commande ou du programme.

L'option SYSTEM signifie que l'entrée clavier est prise immédiatement dans le fichier.

La commande GET CONSOLE permet, dans un fichier destiné à GET FILE, de rendre la main au clavier avant la fin du fichier.

Remarquons au passage que GET permet d'une autre façon de réaliser la même chose que SUBMIT, mais sans le passage de paramètres. En fait, le but à atteindre est la possibilité d'exécuter de manière automatique toute une série d'opérations.

Voici quelques exemples :

— Supposons que le fichier liste.get contienne les deux lignes suivantes :  
dir [full]\*.txt  
dir [full] b:\*.txt

La commande **A>get file liste.get [system]** va provoquer l'exécution des deux lignes contenues dans liste.get.

— Imaginons un programme d'impression qui demande un nom de fichier à imprimer puis demande si on veut imprimer un autre fichier, O pour Oui, N pour Non ; ce programme s'appelant IMPTXT. D'autre part, un fichier livres.get contient les lignes suivantes :

VOLUME1.TXT  
O  
VOLUME2.TXT  
N

Les commandes **A>get file livres.get** et **A>imptxt** vont provoquer l'appel du programme IMPTXT et le passage des lignes de livres.get comme réponse aux saisies de IMPTXT.

Si vous modifiez livres.get en supprimant la dernière ligne "N", alors, après l'impression de VOLUME2.TXT, vous avez le choix de taper au clavier O pour imprimer un troisième fichier de votre choix ou de taper N pour terminer le programme IMPTXT.

## PUT

Cette commande permet la redirection des sorties. En voici la syntaxe :

PUT CONSOLE [OUTPUT TO FILE Nom de Fichier options] | CONSOLE  
PUT PRINTER [OUTPUT TO FILE Nom de fichier options] | PRINTER

Options possibles :

[ECHO | NO ECHO] [FILTER | NO FILTER] [SYSTEM]

**ECHO** permet de gérer l'écho à l'écran de ce qui est envoyé dans le fichier ou l'impression simultanée.

**FILTER** permet de filtrer ou non les caractères de contrôle qui apparaissent alors sous forme ASCII symbolique.

**SYSTEM** signifie que tout ce qui est envoyé par le système CP/M Plus est également stocké dans le fichier.

Remarquons que PUT CONSOLE FILE avec l'option SYSTEM n'est totalement désactivée que par un PUT CONSOLE CONSOLE et idem pour PUT PRINTER PRINTER.

Les deux usages de PUT sont :

— mémoriser tout ce qui est affiché pendant un travail afin de conserver une trace de ce qui a été fait, en vue d'analyse ultérieure,  
— stocker des impressions papier dans des fichiers en vue d'une impression ultérieure ou analyse.

Voici quelques exemples :

**A>PUT CONSOLE FILE liste.put**  
**A>dir [full]**

Le fichier liste.put contient le résultat de ce qui a été affiché à l'écran par la commande dir [FULL].

**A>PUT CONSOLE FILE log.put [SYSTEM]**

A partir de cette commande, tout est mémorisé dans le fichier log.put.

**A>PUT CONSOLE CONSOLE** termine la redirection, ferme le fichier log.put.

Vous voici familiarisés maintenant avec la plupart des commandes de CP/M et CP/M Plus. Le mois prochain, nous continuerons notre série par l'étude des utilitaires destinés au développement de programmes en assembleur. ■

# COMMENT CONSTRUIRE UNE RSX?

**L**a puissance du Basic de l'AMSTRAD ne se limite pas aux seules instructions de son interpréteur ; les concepteurs sont allés beaucoup plus loin et permettent à l'utilisateur d'étendre, à souhait, le jeu d'instructions reconnues par la machine. Ce vocabulaire complémentaire, compris par l'interpréteur est défini au moyen des RSX.

RSX... Derrière ces trois lettres mystérieuses se cache l'abréviation de Resident System eXtension. Ces routines que le programmeur expérimenté écrira en langage machine, auraient pu être simplement appelées par un CALL suivi de leur adresse... Plus confortable, le Basic de l'AMSTRAD permet de les invoquer directement par leur nom. Seule condition : ce nom doit être précédé du symbole !, obtenu en pressant simultanément les touches SHIFT et Arobas. Le but de cet article est de vous apprendre la procédure qui permet d'écrire une instruction RSX.

Pour découvrir la "recette", il suffit de lire le manuel du "Firmware" connu chez AMSTRAD sous la référence "Soft 158". Pour tous ceux qui ne possèdent pas ce document quasi indispensable, CPC explique succinctement le mécanisme qui permet d'écrire une RSX. La RSX n'est pas obligatoirement une instruction simple... Elle peut être suivie de paramètres comme, par exemple, celle décrite par Eddy DUTERTRE dans CPC n° 3, permettant de tracer un cercle dont on définit les coordonnées du centre et le rayon. La forme était CIRCLE x.y.r. Nous allons voir comment passer

ces paramètres.

L'exemple retenu en illustration est une instruction qui fait cruellement défaut sur l'AMSTRAD : celle qui permettrait d'éditer le catalogue d'une disquette sur imprimante. Nous l'avons baptisée PRTCAT (PRinT CATalogue).

Le résultat a été obtenu très simplement, en modifiant deux des indirections du système, détournant la sortie écran vers imprimante. Pour figurer, et surtout pour donner un exemple du passage de paramètres, nous avons ajouté à notre RSX la possibilité d'éditer ce catalogue en 1, 2 ou 4 colonnes sur l'imprimante. L'astuce retenue fait simplement appel à une sélection automatique du mode d'écran... Ainsi :

PRTCAT,0 édite le catalogue en 1 col.  
PRTCAT,1 édite le catalogue en 2 col.  
PRTCAT,2 édite le catalogue en 4 col.  
Pour reconnaître une RSX, le système a  
besoin des conditions suivantes :

- Réserve d'une table contenant les codes ASCII formant l'instruction. Le dernier caractère du "mot" devra nécessairement avoir son bit 7 à 1.

— Comme plusieurs instructions peuvent être présentes dans la table, la fin de celle-ci est signalée par un 0.

- Une zone mémoire de 4 octets doit être réservée pour le système (Kernal).

— Cette instruction devra être présentée au système pour qu'il la reconnaisse par la suite ; c'est le rôle de la routine KL LOG EXT, accessible par le JUMP BLOCK.

Lors de l'appel de l'instruction, on pourra vérifier que le nombre de paramètres transmis est correct ; ici 1. Si ce nombre est incorrect, on n'exécutera pas la RSX ou on sortira un message d'erreur pour informer l'utilisateur... Ici, cette solution n'a pas été retenue.

L'accumulateur (A) contient ce nombre de paramètres alors que le registre IX pointe sur le dernier. Ici, il n'y a qu'un seul paramètre. Pour décoder ce paramètre, on utilise donc le pointeur IX. Ici, IX + 0 donnera la valeur 0, 1 ou 2 qui doit suivre PRTCAT. Ici, il est inutile de tester la validité du paramètre car ceci est fait automatiquement par la routine système SETMODE établissant le mode écran. Par contre, cela pourrait parfois être indispensable. Après avoir recueilli ce ou ces paramètre(s), on exécute la routine d'extension à proprement parler. Notre routine est simple : elle force le mode écran selon la valeur passée par le

## LISTING ASSEMBLEUR

```

|PRTCAT  RSX sortant le catalogue disquette sur imprimante

```

Denis BONOMO pour CPC

```

inixt equ &bcd1      :integration de l'extension
catal equ &bc9b      :etablit le catalogue disquette
restore equ &bb51     :reset indirections TXT
buffcat equ &8000     :buffer 2Ko pour catalogue
printer equ &bdf2     :envoi caractere imprimante
screen equ &bdda      :envoi caractere ecran
setmode equ &bc0e     :force le mode ecran
tstkey equ &bb1e      :test du clavier
busu equ &bd2e        :test imprimante prete

```

```

org &a000          :adresse implantation
ld bc,ext         :adresse de la table
ld hl,mem         :4 octets reserves pour Kernal
jp inext          :loge l'extension creee

```

```
.ext
    defw mot                :adresse du nom de l'extension
    jp routine              :saut a la routine d'extension
```

```

    .mot
    defm "PRTCA"           :nom de l'extension
    defb "T"+%80           :dernier caractere bit 7 a 1

```

paramètre suivant PRTCAT. Elle teste la touche "ESC" pour pouvoir "Breaker" au cas où l'imprimante ne serait pas sous tension. Elle détourne ensuite la partie écran vers l'imprimante. Elle appelle la routine système CAT après avoir indiqué une zone mémoire buffer provisoire de 2 Ko où sera rangé le catalogue (ici, en &8000 mais cette valeur pourrait être modifiée). Pour finir, elle rétablit les indications initiales pour que, par la suite, l'écran soit à nouveau validé.

Le lecteur qui ne possède pas d'assembleur pourra taper le programme Basic, contenant dans ses lignes DATA le code machine correspondant à la RSX. Après avoir fait RUN, il suffira de faire un CALL &A000, afin d'initialiser une fois pour toutes la RSX qui sera, par la suite, appelée par I PRTCAT, n où n prendra les valeurs 0, 1 ou 2. Il est possible de sauvegarder le forme binaire de l'extension par SAVE "PRTCAT",b,&A000,&42. Après l'avoir chargée par LOAD, il faudra la lancer par CALL &A000 pour l'initialiser...

Au moyen de cet exemple simple, nous espérons avoir prouvé au lecteur qu'il est relativement simple de rendre son AMS-TRAD encore plus performant...

Denis BONOMO

```

defb 0 :fin table extension indiquée par 0
.mem
defb 4 :memoire pour Kernal
.routine
co 1 :verifie nombre de parametres RSX
ret nz :pas bon...
ld a,(ix+0) :recueille le parametre 'Mode'
call setmode :force mode ecran correspondant
.no
ld a,66 :code de la touche ESC (break)
call tstkeu :detecte si elle est actionnee
ret nz :sortie si oui.
call busu :si imprimante non prete.
jr c,no :attente en testant ESC.
ld bc,2 :intervertit 2 indirections
ld hl,printer :detournant vers l'imprimante
ld de,screen :ce qui allait vers l'ecran
ldir
ld de,buffcat :pointe sur buffer catalogue
call catal :acces au catalogue disquette
call restore :retablit les indirections initiales
ret
end

```

## I PRCAT - LISTING IMPLANTATION DU LM

```

10 A=&A000:F=&A042:L=100:WHILE A<=F:FOR A=A TO A+15:READ C$:K=VAL("&"+C$):S=S+K+65536*(S+K>32767):IF A<=F THEN POKE A,K
20 NEXT:READ D$:T=VAL("&"+D$):IF T<>S THEN PRINT CHR$(7):"Erreur ligne":L=L+5:WEND
100 DATA 01.09.A0.21.15.A0.C3.D1.BC.DE.A0.C3.19.A0.50.52.069C
105 DATA 54.43.41.04.00.00.00.00.00.FE.01.00.0D.7E.00.0D.0C2F
110 DATA 0E.BC.3E.42.0D.1E.BB.C0.CD.2E.BD.3E.F5.01.02.00.12C7
115 DATA 21.F2.BD.11.DA.BD.ED.B0.11.00.80.CD.9B.BC.CD.51.1BAF
120 DATA BB.C9.00.00.00.00.00.00.00.00.00.00.00.00.00.1D33

```

## I PRCAT,1

```

DEMOPINT.BAS 10K PRTCAT .BIN 1K
ESSAI .FIC 1K PRTCAT .DAT 1K
GAMMUSIC.BAS 8K PRTCAT .SRC 3K
GESFICH .BAS 17K PRTDET .BAK 1K
MEDOR .BAS 1K PRTDET .BAS 1K
MEDORA .BAS 1K

```

99K free

## I PRCAT,2

Drive A: user 0

```

CATDISC .BAS 5K GAMMUSIC.BAS 8K MULTIF .BAS 1K PRTCAT .SRC 3K
CERPAI .BIN 2K GESFICH .BAS 17K MULTIF1 .BAS 17K PRTDET .BAK 1K
DEMOPINT.BAS 10K MEDOR .BAS 1K PRTCAT .BIN 1K PRTDET .BAS 1K
ESSAI .FIC 1K MEDORA .BAS 1K PRTCAT .DAT 1K

```

99K free

## I PRTCAT,0

Drive A: user 0

```

CATDISC .BAS 5K
CERPAI .BIN 2K
DEMOPINT.BAS 10K
ESSAI .FIC 1K
GAMMUSIC.BAS 8K
GESFICH .BAS 17K
MEDOR .BAS 1K
MEDORA .BAS 1K
MULTIF .BAS 1K
MULTIF1 .BAS 17K
PRTCAT .BIN 1K
PRTCAT .DAT 1K
PRTCAT .SRC 3K
PRTDET .BAK 1K
PRTDET .BAS 1K

```

99K free

# INITIATION A L'ASSEMBLEUR

Denis BOURQUIN

**N**ous continuons cette série d'articles par un autre exemple de programme écrit en assembleur. Nous avons choisi ce mois-ci un jeu classique, le tennis, qui fut le premier des jeux vidéo. Il aurait pu être écrit différemment, mais le choix s'est porté sur ce type de programmation afin de faciliter la compréhension du programme pour le débutant.

Pour cela, nous avons fait abondamment appel aux sous-programmes de la ROM de l'AMSTRAD en utilisant toujours les points d'entrée en RAM.

Dans ce jeu, l'écran présente un cadre à trois côtés sur lesquels la balle va rebondir ; il faudra rattraper cette balle avec une raquette qui se déplace en bas de l'écran horizontalement et qui est actionnée par les flèches du clavier. Toute autre touche pourra être choisie. Tous les paramètres peuvent être modifiés aisément, que ce soit la vitesse de déplacement de la raquette ou la vitesse de déplacement de la balle, ces paramètres étant définis dans la zone de variables. Pour une partie vous disposez de 10 balles, chaque frappe de la balle avec la raquette rapporte un point.

Analysons la structure de ce programme. Nous l'avons décomposé en quatre blocs :

- un bloc de déclaration d'équivalence,
- un bloc de déclaration de variable,
- un bloc de sous-programme,
- le programme principal.

Tout d'abord, nous trouvons la directive ORG qui fixe l'adresse de départ du programme. Pour une utilisation de ZEN, on trouve une autre directive LOAD indiquant au programme d'assemblage où il devra loger le code en mémoire.

Le premier bloc de notre programme est un bloc d'équivalences qui attribue un nom aux différents sous-programmes de la ROM que nous allons appeler. Nous choisirons des étiquettes à chacune de ces adresses afin de ne pas être obligés de rechercher à chaque fois les adresses dans la documentation et aussi afin de faciliter la relecture du programme. Il est à noter que la plupart de ces routines détruisent tout ou une partie des registres du Z80 aussi, dans ce programme, nous travaillerons assez peu avec les registres sauf, bien sûr, pour les appels aux sous-programmes pour lesquels ils ont une fonction bien précise.

Nous avons mis ensuite les variables utilisées en début de mémoire. Si nous voulons que l'adresse de lancement reste en

200H, qui est la valeur fixée par le ORG., nous mettrons un saut par dessus cette zone de variable.

Les variables que nous utiliserons sont : XR qui contient la position de la raquette à l'affichage.

XRP qui est la coordonnée de la position précédente de la raquette après un déplacement. La mémorisation de cette coordonnée raquette nous permet d'effacer le dessin de la raquette après un déplacement.

YB et YP sont les coordonnées de la balle visualisée sur l'écran.

XBP et YBP sont les coordonnées de la balle précédant un déplacement.

Ces variables sont de type word, c'est-à-dire sur 16 bits, car elles peuvent prendre des valeurs jusqu'à 640 et un octet, rappelons-le, prend comme valeur maximale 255.

SENS est une variable de type octet qui permet à tout moment, dans le programme, de connaître le sens de déplacement de la balle ; elle peut prendre les valeurs 1, 2, 3, 4 qui correspondent aux quatre sens possibles suivant les deux diagonales, c'est-à-dire le sens 1 correspond à une balle descendant à droite, le sens 2 à une balle montant à droite, le sens 3 à une balle montant à gauche, le sens 4 à une balle descendant à gauche. CPT est une variable de type octet servant de compteur pour le score ; son contenu est particulier puisque ce score est codé en BCD, nous expliquerons cela plus loin.

VITB est une variable sur 16 bits qui contient l'incrément du déplacement de la balle et par conséquent correspond à la vitesse de son déplacement.

VITR reste une variable identique à VITB mais affectée au déplacement de la raquette.

NBB est une variable initialisant le nombre de balles.

Ensuite, vient une série de sous-programmes. Ces sous-programmes sont construits autour d'une fonction particulière à l'intérieur du jeu. Ils peu-

vent utiliser des variables ou des paramètres qui pourront leur être envoyés au travers de registres ou au travers de variables communes.

Le premier, ECRAN, comme son nom l'indique, va initialiser l'écran ; pour cela, il appelle le sous-programme de la ROM qui positionne en mode d'affichage et qui met l'écran dans le mode contenu dans le registre A, ici 1. Il initialise ensuite les encres ; on utilise ici le numéro d'encre 0 qu'on initialise à la couleur 0, soit le noir, et le numéro d'encre 1 qu'on initialise à la couleur 13. Ces encres sont ensuite affectées au fond et au stylo et le bord est mis dans la couleur 0, tout ceci comme en Basic. Les registres utilisés l'ont été en fonction des spécifications des sous-programmes appelés. Ce sous-programme se termine par une réinitialisation de la vitesse de répétition du clavier et du temps d'attente avant cette répétition. Le registre H contient le nombre de fois 1/50<sup>e</sup> de seconde d'attente et le registre L contient le nombre de scrutations effectuées par seconde, 1 correspond à 50, 2 à 25, etc.

Le sous-programme CADRE trace le cadre ou les murs entourant notre jeu ; il n'y a pas de mur derrière la raquette. Il appelle les routines système correspondant au MOVE BASIC et au DRAW, ces routines recevant les coordonnées en abscisse à travers la paire de registres DE et les coordonnées en ordonnée à travers la paire de registres HL.

Le sous-programme RAQUETTE trace la raquette ; ce tracé est effectué en utilisant les routines graphiques de la ROM équivalentes à MOVER et DRAWR du Basic. Les tracés sont donc effectués en relatif par rapport au dernier point, les coordonnées comme pour les sous-programmes DRAW et MOVE se trouvent dans les paires de registres DE et HL. Dans notre jeu, la raquette est constituée par deux lignes horizontales de longueur 35, la valeur hexadécimale OFFDDH correspond à la valeur signée -35.

Le sous-programme TBALL effectue le dessin de la balle ; dans ce sous-programme, nous avons choisi un autre mode de passage des paramètres : la balle est tracée en coordonnées absolues passées dans les registres DE et HL. Ces registres sont sauvegardés dans la pile par des PUSH car le sous-programme PLOT les détruit. Le dessin de la balle est

fait à partir de points placés en croix. Le sous-programme SCORE sert à l'affichage du score. Il incrémente à chaque appel la variable CPT et l'affiche à l'écran sur la ligne 2 en colonne 19. Dans ce sous-programme, nous voyons apparaître une nouvelle instruction DAA que nous allons expliquer.

## LA REPRESENTATION DCB ET L'INSTRUCTION DAA

Jusqu'à présent, nous avons vu qu'un nombre pouvait être représenté en binaire, soit signé, soit non signé. Il existe une autre méthode de représentation des nombres qui est plus proche de notre système décimal, c'est la représentation **décimale codée binaire**. Dans cette représentation, chaque chiffre décimal est codé séparément en binaire, mais voyons tout de suite un exemple.

Le nombre décimal 25 se représente en binaire pur sous la forme : 00011001 (16H). Il se représentera en DCB : 00100101. Nous avons dissocié notre octet en deux sous-ensembles de quatre bits que nous pouvons appeler quartets. Comme vous le savez, pour représenter un nombre de 0 à 9, il est nécessaire de disposer de 4 bits et donc avec un octet, nous allons pouvoir représenter deux chiffres décimaux, mais attention, les règles de l'addition binaire telles que nous les avons vues, ne pourront plus s'appliquer directement à cette nouvelle représentation. Lorsque nous ferons une addition ou une soustraction entre deux nombres en DCB, il va être nécessaire d'effectuer ensuite un ajustement du résultat et ce sera le rôle de l'instruction DAA qui est l'instruction d'ajustement décimal de l'accumulateur.

Nous avons une application de cette représentation dans notre programme : le score est compté en DCB et il n'est donc plus nécessaire de faire une conversion pour l'afficher, il suffit de récupérer chaque chiffre en prenant chaque quartet séparément.

Ceci nous ramène à notre sous-programme, l'affichage se fait en prenant le chiffre de poids fort, pour cela, nous effectuons quatre décalages logiques à droite, ce qui ramène ce chiffre dans les poids faibles de l'accumulateur, on y ajoute 30H pour le coder en ASCII et on l'envoie au sous-programme d'affichage. Le chiffre de poids faible est récupéré en masquant par un ET logique le chiffre de poids fort et affiché par la même méthode.

## LES OPERATIONS DE DECALAGE

Dans une opération de décalage, le contenu du registre est décalé vers la gauche ou vers la droite d'une position binaire. Le bit qui entre est mis à 0 et le bit qui sort est mis dans le drapeau de carry C. Il y a une exception pour le décalage arithmétique à droite : lorsque le bit le plus à gauche est à 1, après une opération de décalage à droite, ce bit est recopié dans la position la plus à gauche. Nous avons deux types de décalages :

des décalages logiques et des décalages arithmétiques ; le décalage logique à gauche correspond à la multiplication par deux en arithmétique non signée et le décalage à droite à l'opération de division par deux, toujours en arithmétique non signée. Les décalages arithmétiques correspondent aux mêmes opérations, mais en arithmétique signée d'où la nécessité, dans le décalage arithmétique à droite, de maintenir le signe dans le bit le plus à gauche.

Nous arrivons maintenant au programme principal.

Il commence par une initialisation du registre pointeur de pile, le registre SP. On initialise ensuite l'écran par l'appel à ECRAN et à CADRE vus plus haut.

Ce programme principal se compose d'une boucle correspondant à une balle, cette boucle sera donc exécutée 10 fois, elle débute à l'étiquette LANCE. La balle arrive dans le jeu par le haut de l'écran avec une abscisse aléatoire. Le nombre aléatoire est obtenu en venant lire le registre de rafraîchissement du Z80 que l'on appelle R. Les coordonnées de la nouvelle balle sont alors initialisées, le score et le nombre de balles restantes sont affichés. La raquette est dessinée aux coordonnées (XR,15) après avoir été effacée des coordonnées précédentes. Le clavier est scruté sans attendre qu'une touche soit frappée et la routine KEY positionne le drapeau de CARRY lorsqu'une touche a été actionnée. Si aucune touche n'a été actionnée, le programme passe au mouvement de la balle, sinon il teste quelle touche a été appuyée. Soit c'est la flèche de gauche, et alors les coordonnées de la raquette sont décrémenteées de la valeur du pas de mouvement de la raquette VITR, soit c'est la flèche de droite, et ces mêmes

coordonnées sont incrémentées de VITR et ce après avoir sauvegardé les anciennes coordonnées dans XRP.

On arrive ensuite au mouvement de la balle. Le sens de ce mouvement est donné dans la variable ENS ; suivant sa valeur, les coordonnées XB et YB seront incrémentées ou décrémenteées de VITB.

Il faut, après ces opérations, vérifier que la balle n'est pas en butée contre un mur par les tests de bornes de ses coordonnées et vérifier qu'elle n'est pas contre la raquette. Dans chacun de ces cas, elle rebondit avec un angle de réflexion égal à l'angle d'incidence. Le test du choc sur la raquette se fait par l'appel au sous-programme, test qui retourne la couleur du point de coordonnées contenues dans les paires de registres DE et HL. Les étiquettes Mi correspondent aux tests de chacun des côtés avec pour chaque côté deux angles d'arrivée possibles. Le bloc de programme "mouvement" se termine par le tracé de la balle après l'avoir effacée de sa position précédente. A l'étiquette LANCE2 se trouvent les lignes de programme nécessaires à l'effacement de la balle lorsqu'elle sort à côté de la raquette.

Ce programme se termine par la réinitialisation du clavier et le retour au Basic. Nous vous souhaitons de passer un agréable moment avec ce jeu, mais il a surtout été écrit dans le but d'être aisément modifiable afin de permettre à chacun de s'initier au langage assembleur. Il peut être transformé en casse-briques, la balle et la raquette peuvent être dessinées différemment. De même, l'emplACEMENT de l'affichage de la raquette pourra être modifié dans le programme, ce qui aura pour action de ralentir ou d'accélérer le jeu.

Bon courage !

# Libtong

ORG 200H  
LOAD 200H

; Initialisation des sous-programmes de  
; de la ROM utilise dans ce jeu

```

MODE: EQU OBC0EH ; A=mode
DRAW: EQU OBEF6H ; DE=abscisse, HL=ordonnee
DRAW: EQU OBBF9H ; DE=abscisse relative, y=ordonnee relative
PLOT: EQU OBBEAH ; DE=abscisse, HL=ordonnee
PLOT: EQU OBBEDH ; DE=abscisse relative, HL=ordonnee relative
INK: EQU OBC32H ; A=numero d'encore, B=1ere couleur, C=2eme couleur
PEN: EQU OBBDEH ; A=numero d'encore
BOPDER: EQU OBC38H ; B=1ere couleur, C=2eme couleur
PAPER: EQU OBBE4H ; A=numero d'encore
MOVE: EQU OBBCOH ; DE=abscisse, HL=ordonnee
MOVE: EQU OBB03H ; DE=abscisse relative, HL=ordonnee relative
KEY: EQU OBB1BH ; Si une touche CY=1 et touche dans A
SPEED: EQU OBB3FH ; H=temps d'attente, L=vitesse repetition
TEST: EQU OBBF0H ; DE=abscisse, HL=ordonnee en retour A=couleur
COLONNE: EQU OBB6FH ; A=colonne du curseur
LIGNE: EQU OBB72H ; A=ligne du curseur
TXTOUT: EQU OBB5AH ; A=caractere

```

JP DEBUT ; brachement au debut du programme

; Initialisation des variables utilisees

; XR :coordonnees de la raquette  
; XRP :coordonnees de la position  
; precedente de la raquette.

; XB et YB :coordonnees de la posi-  
; tion de la balle  
; XBP et YBP :coordonnees de la posi-  
; tion precedente de la balle.

; SENS variable memorisant le sens  
; de deplacement de la balle, elle peut  
; prendre 4 valeurs representant des  
; deplacements suivant les diagonales.

; CPT compteur de score, le nombre est  
; en BCD.

; VIIB et VITR sont les variables  
; d'incrementation des deplacements de

; la balle et de la raquette, donc  
; elles definissent les vitesses de  
; deplacement de la balle et de la  
; raquette.

```

XR: DW 320 ; abscisse de la raquette
XRP: DW 320 ; memorise l'abscisse de la position precedente
XB: DW 0 ; abscisse de la balle
XBP: DW 0 ; abscisse de la position precedente de la balle
YB: DW 0 ; ordonnee de la balle
YBP: DW 0 ; ordonnee precedente de la balle
SENS: DB 0 ; variable contenant sens deplacement balle
CPT: DB 0 ; octet contenant le score en DCB
VITB: DW 4 ; mot contenant l'increment de deplacement balle
VITR: DW 7 ; mot contenant l'increment de deplacement raquette
NBB: DB 10 ; nombre de balles disponible

```

\*\*\* ecritures des differents sous  
\*\*\* programmes utilises dans le  
\*\*\* jeu.

; Initialisation de l'ecran ici en mode  
; 1 avec comme encrre 1 et 0 de couleur  
; 13 et 0  
; initialisation aussi des vitesses de  
; repetition du clavier.

```

ECRAN: LD A,1 ; mode 1=320x200 et 40 caracteres
CALL MODE
CALL OBBBAH
LD A,0 ; numero d'encrre 0

```

```

LD BC,0 ; 1ere et 2eme couleur noire
CALL INK
LD A,1 ; numero d'encre 1
LD BC,0D0DH ; 1ere et 2eme couleur blanc
CALL INK
LD A,0 ; encr 0 soit noir
CALL PAPER ; fond
LD A,1 ; encr 1 soit blanc
CALL PEN ; stylo
LD BC,0 ; couleur noir
CALL BORDER ; bord
LD H,2 ; attente avant repetition 2x20ms
LD L,1 ; repetition 1 touche toutes les 20ms
CALL SPEED
BC7

```

```

;
; *** sous programme de trace des murs.
;

```

```

CADRE: LD DE,630 ; deplacement en 630,10
LD HL,10
CALL MOVE
LD DE,630
LD HL,350 ; trace du mur droit
CALL DRAW
LD DE,10
LD HL,350 ; trace du mur haut
CALL DRAW
LD DE,10
LD HL,10 ; trace du mur gauche
CALL DRAW
RET

```

```

;
; *** sous programme de trace de la raquette
;

```

```

RAQUETTE: LD HL,1
LD DE,35
CALL MOVER
LD HL,1
LD DE,0FFDDH ; trace d'une ligne de 35 pixels vers la gauche
CALL DRAW
LD HL,1
LD DE,35
CALL DRAW
LD HL,1
LD DE,0FFDDH ; trace d'une ligne de 35 pixels en relatif
CALL DRAW
LD HL,1 ; trace de 35 en relatif
LD DE,35
CALL DRAW
LD HL,1
LD DE,0FFDDH ; trace de -35 en relatif
CALL DRAW
RET

```

```

;
; *** sous programme de trace de la balle
;

```

```

TBALL: DE PUSH ; sauvegarde des coordonnees DE=X et HL=Y
HL PUSH
PLOT CALL ; marque le point de coordonnees dans DE et HL
HL POP ; rappel de la coordonnee en Y
DE POP ; rappel de la coordonnee en X
DE PUSH ; sauvegarde a nouveau dans la pile de DE
HL PUSH ; puis de HL
DE DEC ; X=X-2
DE DEC
CALL PLOT ; marque le point de coordonnee X-2
HL POP ; recupere X et Y dans la pile
DE POP ; sauvegarde X et Y dans la pile
HL PUSH
DE DEC ; Y=Y-2
HL DEC
CALL PLOT ; trace le point de coordonnees X,Y-2
HL POP ; recupere X et Y dans la pile
DE POP ; sauvegarde X et Y dans la pile
HL PUSH
DE INC ; X=X+2
DE INC
CALL PLOT
HL POP
DE POP
HL POP
DE INC ; Y=Y+2
HL INC
CALL PLOT
RET

```

```

;
; *** sous programme de calcul et d'affichage du score.
;

```

```

SCORE: LD A,(CPT) ; A=CPT, CPT est en DCB
ADD A,1 ; ajoute a A la valeur en point d'un coup
DAA ; ajustement decimal de l'accumulateur
LD (CPT),A ; stockage de la nouvelle valeur du score
LD A,19
CALL COLONNE ; positionne le curseur en colonne 19
LD A,2
CALL LIGNE ; positionne le curseur en ligne 2
LD A,(CPT) ; A=CPT
AF PUSH ; sauvegarde A dans la pile
SRL A ; effectue quatre decalage a droite de
SRL A ; l'accumulateur pour ramener le chiffre
SRL A ; DCB se trouvant dans les 4 bits de poids
SRL A ; forts dans les 4 bits de poids faibles
ADD A,30H ; ajoute 48 a A pour convertir ce chiffre
CALL TXTOUT ; en ASCII et envoie a l'ecran
POP AF ; recupere A dans la pile
AND OFH ; masque le chiffre de poids fort et donc conserve
ADD A,30H ; le chiffre de poids faible convertit en ASCII et
CALL TXTOUT ; envoie a l'ecran
RET

```

```

;
;*** programme principal.
;
DEBUT: LD SP,09800H ; initialisation du pointeur de pile en 9800H
CALL ECRAN ; initialisation ecran
CALL CADRE ; trace des murs
LD A,(INBB) ; A=NB, nombre de balles
LD B,A ; mets registre A dans B

; boucle principal correspondant a une
; balle
;
LANCE: LD A,R ; mets dans le registre A un nombre aleatoire issu
LD L,A ; de la lecture du registre de rafraichissement
LD H,0 ; ce nombre est mis dans HL
LD (XB),HL ; qui est stocke dans XB et XBP qui sont les
LD (XBP),HL ; absisses de la balle
LD HL,345 ; l'ordonnee est choisit en haut du mur
LD (YB),HL ;
LD (YBP),HL ;
LD A,2
CALL LIGNE ; curseur en ligne 2
LD A,2
CALL COLONNE ; curseur en colonne 2
LD A,B ; recupere le nombre de balles restantes
DEC A
CALL AFFI ; affichage du nombre de balles restantes
CALL SCORE ; affichage du score
LD A,1 ; initialisation du sens au depart
LD (SENS),A ; vers le bas et vers la droite
PUSH BC ; sauvegarde de BC car routines graphiques
; detruisent BC
; le trace de la raquette s'effectue a
; chaque passage pour ralentir le
; deplacement de la balle
;
TRACE: LD A,0 ; couleur de fond pour le stylo
CALL PEN
LD DE,(XRP) ; coordonnees de la position avant deplacement
LD HL,15
CALL MOVE ; place curseur graphique a DE,HL
CALL RAQUETTE ; efface la raquette
LD A,1 ; stylo dans la couleur 1
CALL PEN
LD DE,(XR) ; nouvelles coordonnees de la raquette
LD HL,15
CALL MOVE
CALL RAQUETTE ; trace de la raquette

; lecture du clavier.
ATTENTE: CALL KEY ; une touche a-t-elle ete appuye
JR NC,MVT ; non alors deplacement de la balle
CP OF3H ; oui est ce la fleche a droite
JR NZ,S1 ; non alors on va en S1

; fleche a droite.
LD HL,(XR) ; dans ce cas on recupere l'abscisse de la
LD (XRP),HL ; raquette, on la met dans XRP
LD BC,(VITR) ; et on lui ajoute l'increment de deplacement

```

```

ADD HL,BC
LD (XR),HL ; on met la nouvelle valeur dans XR
LD BC,595 ; comparaison avec le bord droit du mur
XOR A ; en tenant compte de la longueur de
SBC HL,BC ; la raquette
JR C,MVT ; la raquette n'est pas au mur alors MVT
LD HL,594 ; met la valeur max de l'abscisse de la raquette
LD (XR),HL
JR MVT ; va au mouvement de balle
CP OF2H ; la touche appuye est la fleche a gauche
JP NZ,MVT ; non on va a mouvement

; fleche a gauche.
LD HL,(XR) ; on stocke l'abscisse de la raquette dans
LD (XRP),HL ; XRP
XOR A
LD BC,(VITR) ; on lui ajoute l'increment de deplacement
SBC HL,BC
LD (XR),HL
XOR A ; la raquette a-t-elle atteint le bord
LD BC,11 ; gauche
SBC HL,BC ; non alors MVT
JR NC,MVT ; oui on met 11 dans XR
LD HL,11
LD (XR),HL

; mouvement de la balle
MVT: LD A,(SENS) ; on recupere le sens de deplacement de la balle
CP 1 ; est-ce le sens 1
JR NZ,M2 ; non on va en M2
LD HL,(XB) ; qui sens vers le bas vers la droite
LD BC,(VITB) ; HL=XB et BC=VITB
ADD HL,BC ; on ajoute a XB l'increment de deplacement de
LD (XB),HL ; la balle, XB=HL
LD (YB) ; HL=YB
LD BC,(VITB) ; BC=VITB
XOR A ; on soustrait a YB l'increment de deplacement
SBC HL,BC ; YB=HL
LD (YB),HL ;
LD HL,(XB) ; on compare XB avec le bord droit en tenant
LD DE,625 ; compte de la taille de la balle
XOR A
LD HL,DE
SBC C,M1 ; la balle n'a pas atteint le bord alors M1
LD A,4 ; la balle a atteint le bord gauche, elle
LD (SENS),A ; rebondit vers le bas et vers la gauche
JR BALL ; va au trace de balle
LD HL,(YB) ; la balle a-t-elle atteint le bas du cadre
LD DE,22 ; comparaison avec l'ordonne du haut de la
XOR A ; raquette
LD HL,DE ; non alors on va au trace
SBC NC,BALL ; oui alors il faut verifier si la balle
LD DE,(XB) ; a rencontre la raquette pour cela on
LD HL,16 ; test la couleur du point sous la balle
CALL TEST ; est ce la couleur du stylo
CP 1 ; non alors balle sortie et relance une nouvelle
JR NZ,LANCE2 ; oui appel a score qui incremente le compteur
CALL SCORE ; la balle rebondit sur la raquette vers la
LD A,2 ; gauche et en montant
LD (SENS),A

```

