

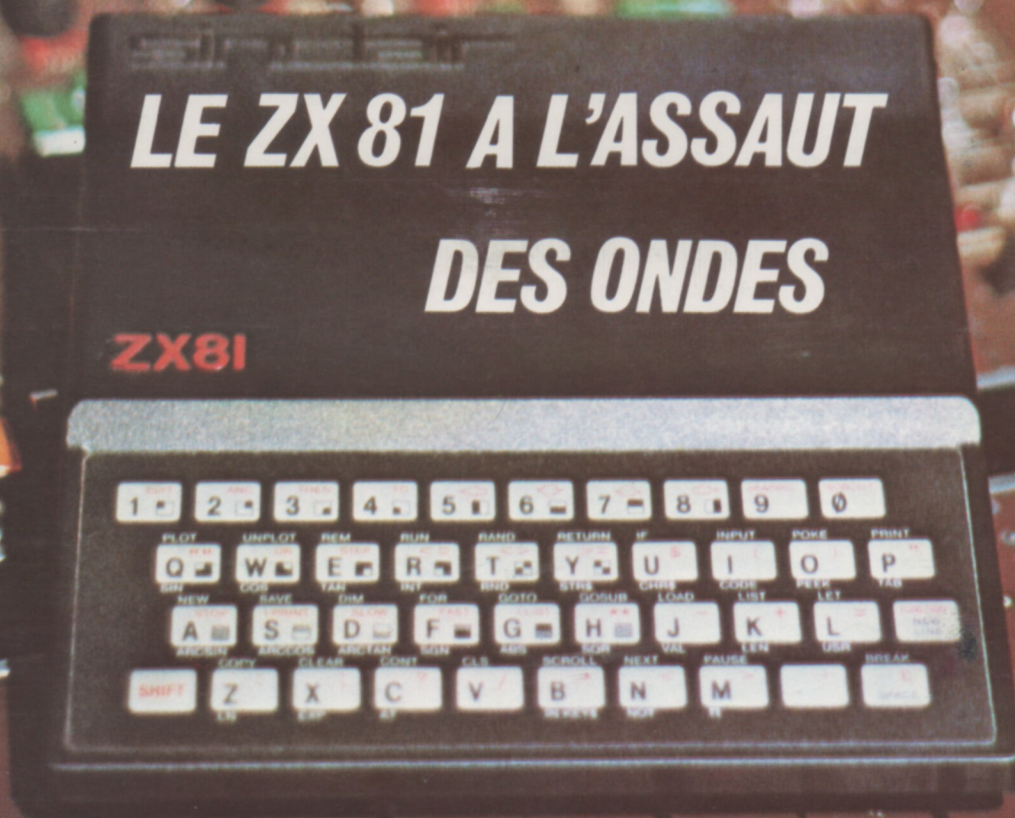
Echos sinclair

NO 4

SPECIAL BANCS D'ESSAIS UTILITAIRES
UNE MANETTE DE JEUX POUR VOTRE ZX 81

**LE ZX 81 A L'ASSAUT
DES ONDES**

ZX81



Editoria/

La télématique : l'association des télécommunications et de l'informatique est depuis plusieurs années une expérience courante et une réalité quotidienne dans le domaine de la grosse informatique.

Aujourd'hui, des micro-ordinateurs professionnels peuvent être équipés pour la transmission et le traitement d'informations à distance par les ondes hertziennes ou par câbles téléphoniques.

Les Echos Sinclair ont voulu tenter, en coopération avec la station radio « Ici et Maintenant », une expérience nouvelle : transmettre des programmes par les ondes à des utilisateurs du ZX81.

Les résultats plus qu'encourageants de cet essai ont démontré qu'une transmission de ce type est non seulement possible, mais qu'elle fonctionne bien, ce qui ouvre des horizons nouveaux aux utilisateurs du Sinclair.

Un courrier « magnétique » pourra être envoyé à la station de radio qui en diffusera ultérieurement le contenu lors d'émissions prévues à cet effet. Utilisateurs du ZX, la télématique entre chez vous.

J.M Cohen

SOMMAIRE

N°4

INFORMATIONS

Vie des Clubs.	4
Courrier des lecteurs.	6
Produits nouveaux.	8

LOGICIELS

Nos lecteurs ont du génie.	10
Nos lecteurs s'amuse.	14
RAMTOP.	24
La recherche en table.	26

BANCS D'ESSAI

ZX assembleur-éditeur/moniteur-désassembleur.	21
ZXTRI.	28

MATERIELS

ZX 81 : la manette de jeu.	32
Petites annonces.	31

Photo de couverture : prise de vue réalisée aux studios Johanna 64 rue Sadi Carnot
93170 Bagnolet

Echos
Sinclair

Revue bimestrielle éditée par
Joker International Editions
12 Villa St-Michel 75018 Paris

Directeur de la publication : Jean-Louis Karl

RÉDACTION :

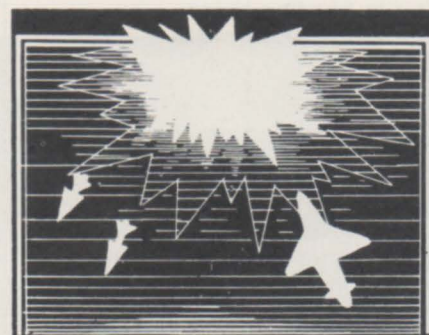
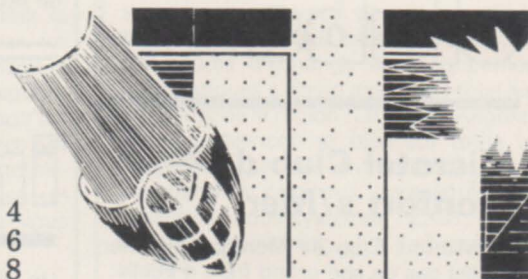
Rédacteur en chef : Jean-Michel Cohen

Secrétaire de rédaction : Marie-Ange Noguer

Photographe : Patrick Garrouste

Publicité : Michel Morise

Photocomposition : Kass - 75018 Paris



ABONNEZ-VOUS
PAGE 13

la vie des clubs

Microtel Club de Monfort s/Mer

Le Microtel Club de Montfort sur Mer sur les pas de son grand frère d'Ecully. Après le Microtel club de Lyon-Ecully (Échos n° 3) qui offre une formule séduisante pour venir à bout du montage du Kit ZX81 et introduire les néophytes dans le monde de l'octet, le Microtel de Montfort sur Mer ouvre ses portes à tous les particuliers amateurs de ZX81.

Renseignements :
Pays de Montfort-Mairie
35/60 Montfort sur mer
(99) 36.51.54

Quelques adresses

BOULOGNE INFORMATIQUE CLUB

22, rue de la Belle Feuille
92100 Boulogne
Séance Portes Ouvertes + cours d'initiation à la micro-informatique.

CLUB SINCLAIR

38, Chemin du Moulin
1328 Ohain Belgique
Échanges — Bulletins de liaison — Achats groupés.

CLUB GIZMO ZX80/81

9, rue Auguste Gal
06300 Nice
61, rue Pierre Brossolette
95200 Sarcelles
Échanges logiciels — Échanges Hard etc...

PARIS MICRO

19, rue Tilly
92700 Colombes
Tél : 242.94.71
Nombreux montages Hards Grande bibliothèque de programmes — Échanges, bulletins de liaison — Ouverture prochainement d'un local à Paris.

ENHANCED COMPUTER

21C rue Faidherbe
94140 Nogent

Échanges de logiciels, astuces de programmation, idées d'extensions, information des nouveautés, initiation au Basic.

Microtel Ecully

Très nombreux projets en cours. Cours de programmation sous forme de sémi-

naires de deux jours ou modules de trois séances.

PUCE AZUR ZX81

17, rue des Capucines
06800 Cagnes s/mer
Échanges de programmes, trucs, idées, conseils, musique PIO, astuces, programmes scolaires et familiaux.

information

Un génial possesseur de ZX81, en association avec deux autres utilisateurs de micro-ordinateurs entreprend d'organiser un tournoi de PUISSANCE 4 sur micro-ordinateurs. Ce tournoi est ouvert à tout possesseur d'une machine et d'un programme de son cru. Les ordinateurs de poche (suivant la différenciation établie dans le Guide de l'Ordinateur Individuel 82-83) formeront une catégorie à part.

Ce tournoi doit se dérouler le dimanche 20 mars 1983 dans les locaux du Collège de Munster (Haut-Rhin).
NB : cette activité est strictement extra-scolaire.

Renseignements : M. STURTZER

Collège Munster
9, rue Sébastopol
68140 Munster

formation

Stage initiation à la micro informatique et au langage Basic, dans l'Aude

Le C.E.R.A. (Centre d'Etude et de Recherche Audiovisuel) organise du 14 au 18 février 1983 à Castelnaudary (Aude) un stage d'initiation à la micro informatique et au langage Basic.

Au programme

1. Technologie d'un micro ordinateur
2. Le choix d'un langage
3. Initiation au Basic
4. Réalisation et mise au point de programmes
5. Utilisation des programmes.

Nombre de participants maximum : 8

Matériel de stage : 3 TRS 80 Basic II 16K, 1 TRS 80 Basic II 48K + 2 unités de diskettes, 1 TRS 80 modèle III 48K + 2 unités de diskettes, 4 Sinclair ZX 81 + 1 extension 16K, 1 imprimante TRS 80, 1 imprimante Sinclair.

Logiciels : comptabilité, fichier, jeux, etc...

Renseignements

C.E.R.A.

«La Raque» 11400 Castelnaudary
Tél : (68) 60.21.89

et pour les animateurs...

Le C.N.F.A. (Centre National de Formation à l'Animation) organise un stage destiné aux animateurs. Celui-ci se déroulera du 7 au 13 février et du 13 au 17 juin 1983 à Fublaine en Seine et Marne.

Renseignements

J.P. Gowlvestre

C.N.F.A. / W FCV BP 5
Fublaine
77470 Trilport Tél (6) 434.61.54

ondes

Ici et Maintenant Compte-rendu radio

L'émission « Echos Sinclair - Ici et Maintenant » s'est soldé par un succès. La bonne transmission des programmes dans leur totalité, ainsi que le nombre important d'appels téléphoniques lors de l'émission le mardi 4 janvier a prouvé l'intérêt des auditeurs pour la micro-informatique.

Étaient présents à l'émission, des représentants du club Paris Micro que nos lecteurs connaissent bien, des utilisateurs, ainsi que des représentants de votre revue.

L'émission commença vers 20 h 30, et parmi les sujets abordés, on peut citer l'approche de l'informatique, l'intérêt de l'informatique personnelle.

Les auditeurs ont pu débattre de la question et exposer leurs opinions.

Les demandes de renseignements techniques ont été nombreuses et les invités ont essayé de répondre à celles-ci. Les principales concernaient le Fast Load Monitor dont le concepteur se trouvait parmi nous d'ailleurs, la compatibilité entre certains programmes système, etc...

Mais l'innovation de la soirée était la transmission de programmes par les ondes. Et après un programme d'essai et la confirmation de bonne réception, cette transmission devait durer jusqu'à 1 h du matin. Plus de 10 programmes ont été ainsi « envoyés ». Pour la petite histoire, nous avons même rediffusé un programme que certains d'entre vous avaient mal reçu.

Pour tout renseignement concernant les émissions à venir, téléphonez ou écrivez à Radio Ici et Maintenant

45, rue de Léningrad
75008 Paris

Tél : 293.28.38

en joignant une enveloppe self-adressée.

Radio-Mayenne

Depuis le mois de décembre, Radio-Mayenne (96,6 MHz), première radio départementale de Radio-France, existant depuis le 16 juin 1980, ouvre ses antennes aux adeptes de **la micro informatique**.

Ainsi, chaque jeudi soir, à 23 heures, Yves Derisbourg vous propose de diffuser vos programmes de micro ordinateurs enregistrés sur cassettes.

Un bourse d'échange de programmes informatiques s'ouvre ainsi pour tous les amateurs de la région (Le Mans, Angers, Rennes, Alençon et bien sûr toute la Mayenne) qui peuvent adresser leurs programmes originaux enregistrés sur cassettes, (tous systèmes : ZX81, Commodore, Apple, etc...) à l'adresse suivante :

Yves Derisbourg

Radio-Mayenne

9, rue du Lieutenant

53000 Laval

Pour tous renseignements, appeler dans la semaine Yves Derisbourg au 56.38.88 par le 43

A vos claviers... A vos cassettes...

INFORMATIQUE

Spécialistes en France

SI VOUS DESIREZ APPARAÎTRE A
CET EMPLACEMENT, TELEPHONEZ
AU 201 04 50

PARIS 13

VISMO INFORMATIQUE

68, rue Albert Tél: 586 60 10

Toutes extensions Matériels Sinclair Stock permanent
Conseils et démonstrations
Produits exclusifs : logiciels - claviers
inverseurs vidéo - moniteurs

PARIS 8

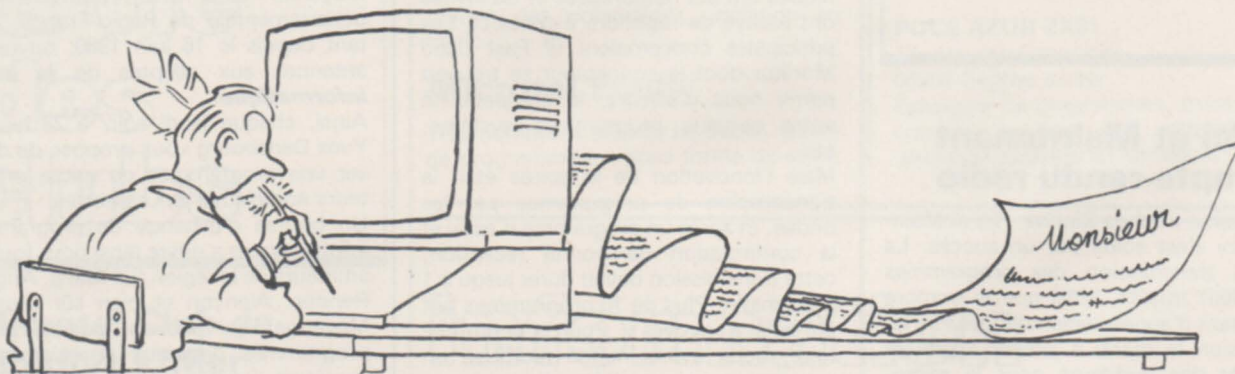
**RAMTOP Services — 5, rue d'Artois
MEMOIRE ECRAN POLYGRAPHIQUES**

Placez 20 pages écrans dans un programme de 16k RAM en utilisant seulement 500 octets de la mémoire haute. Rappel instantané. Possibilité de sauvegarde page par page.

Tracez « à vue » cartes, figures, etc., à l'aide d'une grille translucide aux coordonnées de PRINT et PLOT, dimensions 240 x 170 mm + programme d'avance de PLOT + affichage des coordonnées sur la 23ème ligne.

Accès à tous les caractères, inversion d'écran, 3 dimensions de lettres, « scroll » fugitif ou circulaire dans 4 directions, « scroll » automatique, diagonales, cercles et arcs, etc. Translucides en toutes dimensions sur demande. 85F - Envoi recommandé - Délai: 2 semaines.

courrier des lecteurs



ZXCALC

F. Lehuta - St Jean de Monts

Dans le numéro 1 d'Echos Sinclair page 22, sous le titre ZXCALC, vous avez publié le listing d'un programme « d'aide à la décision ». Malheureusement son impression n'est parfaite à la ligne 250. Beaucoup d'entre vous nous écrivent pour obtenir des renseignements concernant la ligne 250 du listing. Voilà qui vous satisfera.

250 PRINT AT (J-Y)★2+4,(I-X)★10+3; T\$(I,J)

ZXAS + FAST LOAD ?

J.C Pomes - Mulhouse

Pourriez-vous me dire s'il existe un moyen permettant d'utiliser le ZXAS conjointement avec le FAST LOAD MONITOR ?

Non, il n'est pas possible d'utiliser conjointement ZXAS et FLM. Car il faudrait pouvoir déplacer le FLM à un autre endroit dans la mémoire.

D'après les renseignements pris chez le concepteur du programme FAST

LOAD, l'étude de modification est lancée.

Si des lecteurs ont une solution, qu'ils nous la fasse parvenir, nous l'éditerons...

PEEK, POKE

On constate des erreurs de copie d'une ligne lorsqu'on utilise la possibilité d'éditer le numéro de ligne. Ceci ne se produisant semble-t-il qu'avec des lignes assez longues.

Il existe deux cas où l'interrupteur Basic du ZX peut être perturbé dans l'édition d'une ligne.

1. La présence d'un code 7E hex. dans cette ligne, le code 7E lorsqu'il est rencontré par l'éditeur ordonne à celui-ci de ne pas lister les cinq octets suivants. Six octets disparaissent donc lors d'une édition. Ce code correspond à LD A, (HL) en assembleur Z80.

2. La présence d'un code 76 hex. correspondant à NEW-LINE qui signifie à l'éditeur une fin de ligne.

FAST LOAD

A. Sellier - Tervuren

Les copies rapides contiennent-elles, en plus du programme utilisateur (P.U) une version utilisable (non écrasée) du FLM ?

Le FLM réside en RAMTOP sur 2,4K RAM. Il doit être au préalable chargé en mémoire et peut alors permettre de sauvegarder ou d'enregistrer des programmes.

Les programmes devant être sauvegardés à haute vitesse par le FLM n'ont pas besoin d'être réintroduit à la main. Il suffit de les charger par cassette en faisant le LOAD à « nom » normal.

Le FLM est protégé puisqu'il est au-dessus du RAMTOP.

Les programmes 2 et 3 sont les routines de sauvegarde et de chargement de programmes que vous pouvez utiliser à part.

N.B : Le FAST LOAD n'est pas sauvegardé avec le programme principal.

32 et 64K RAM

D^r M. Rousset - Pont Audemer

J'ai une extension 32K et je désire écrire des programmes de plus de 16K en basic.

Comment faire.

Après avoir mis le ZX en route, faire :

POKE 16388, 0
POKE 16389, 192 et NEW

Ceci parce que lors de la mise en route, le ZX « pense » avoir au maximum 16K de RAM, alors il faut lui préciser manuellement la nouvelle capacité mémoire. Pour ce faire, il suffit de mettre à l'endroit de sa mémoire où il stocke le nombre d'octets disponibles (normalement 32768 soit 2 X 8K de ROM Basic + 16K de RAM) la nouvelle valeur.

32768 0 + 256 ★ 128

dans votre cas :

49152 0 + 256 ★ 192

Ne pas oublier le NEW

M. Maigrot - St Brice s/s Forêt

Je comptais, avec une mémoire de 64Ko, pouvoir augmenter la capacité de mes déclarations de matrice jusqu'à 50Ko. Or j'ai eu la surprise de ne pas pouvoir déclarer de matrices de dimensions supérieures à celles acceptées par la mémoire de 16Ko.

Pour vous :

POKE 16388, 255

POKE 16389, 255

NEW

et vous pouvez commencer à programmer.

Produits Nouveaux

SOFT jeux

TRICTRAC (Backgammon)

Le tricarac est un jeu magnifique, facile à apprendre, rapide, très amusant. C'est une combinaison parfaite d'astuces et de chance. Ce très beau programme de tricarac utilise un code-machine rapide et efficace pour choisir les coups. Ce jeu complet comprend un damier graphique, des dés que l'on jette et un dé de double. Il ne suffit pas d'avoir de la chance ! Un programme de jeu de dés figure également sur la face B 16K, 115 F

SIMULATEUR DE VOL

Le Simulateur de Vol est un programme extraordinaire qui utilise en temps réel toutes les possibilités du ZX81. La remarquable représentation graphique montre les nombreux instruments de la cabine de pilotage, ainsi que le monde extérieur vu des hublots de la cabine. Quand vous inclinez l'avion, que vous piquez ou montez, vous voyez monter ou descendre le ciel et la terre par les hublots. Pendant l'atterrissage final, vous avez une perspective mouvante complète de la piste. 16 K, 115 F



PATROUILLE DE L'ESPACE

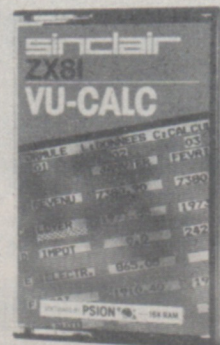
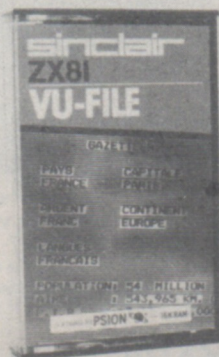
2 jeux de «cafés» très rapide, en temps réel, avec fonctions graphiques entièrement interactives. Avec Patrouille de l'Espace, défendez la Terre contre des armées successives d'affreux extra terrestres.

BOMBARDIER (Face B).
Lancez des bombes et des fusées pour anéantir cité gratte ciel avant que votre avion ne s'écrase dans un immeuble.
16 K, 115 F

gestion

VU-CALC

Construit, réalise et calcule de grands tableaux pour diverses applications telles que analyses financières, budgets et projections. Il transforme le Sinclair ZX81 en un analyseur de données extrêmement puissant, (dimensions des tableaux : 26 lignes/26 colonnes). 16 K 156 F



VU-FILE

Système de classement polyvalent. Un auxiliaire logiciel indispensable à tous ceux qui possèdent le ZX81. Pour les noms et adresses, données de membres de sociétés, dossiers, catalogues, agendas, comptes et listes de toutes sortes. Nombreux

ses caractéristiques dont utilisation de sous-programmes pour stocker le maximum de données, instructions d'entrée, d'impression et listes de recherche. Sur la face B, GAZETTE en est un bon exemple. Il fournit des renseignements sur n'importe quel pays du monde. 16 K 156 F

utilitaires

Tous les programmes sont disponibles avec un manuel de programmation très complet en français.

ASSEMBLEUR ARTIC

Vous permet d'utiliser au maximum les possibilités du microprocesseur Z 80A, en évitant les fastidieux «PEEK-POKE». Cette cassette vous permet donc d'écrire tout programme en «mnémorique» spécifique Z 80 A et d'assembler le programme dans les «adresses basses» de la mémoire. Il comprend un moniteur et un éditeur incorporés qui se chargent de la réservation de place aussi bien pour le programme «source» que pour le programme «objet». Il utilise en outre des étiquettes symboliques en toutes lettres. L'assembleur occupe 5 K octets de mémoire «RAMTOP». Utilisable

avec un ZX81 et son extension 16 K RAM, l'assembleur est indispensable à toutes personnes désireuses de programmer en langage machine. 95 F TTC (cf. Banc d'essai).

MONITEUR DÉSASSEMBLEUR

Complément parfait de l'assembleur le programme vous permet de désassembler vos routines, d'effectuer des recherches d'erreurs de programmation (recherche de chaînes de caractères, d'adresses, positionnement de points d'arrêts, visualisation de registres...). Il permet en outre de programmer en langage machine directement en hexadécimal et de sauvegarder et restituer sur cassette à haute vitesse les programmes ainsi créés. Utilisable avec un ZX81 et son extension 16 K RAM, il occupe les «adresses hautes» de

Produits Nouveaux



la mémoire «RAMTOP» et occupe 4 K octets (cf. Banc d'essai) 95 F TTC

- SAVE : programme d'écriture rapide sur bande vierge du programme Moniteur
- LOAD : programme de l'écriture rapide du Moniteur 86 F TTC.

ZXTRI

Puissant utilitaire de gestion de



tableaux, entièrement en langage machine, très rapide, permettant de trier et d'effectuer des recherches sur des tableaux multidimensionnels. Possibilités d'insertion et de retrait des éléments d'un tableau (cf. Banc d'essai) 95 F TTC

TOOL KIT ARTIC

Le programme Toolkit permet de disposer de huit fonctions nouvelles et performantes pour ZX81 muni d'une extension 16 K RAM. Ces fonctions sont les suivantes :

- RENUM permet d'accéder à la renumérotation totale des programmes (GOTO, GOSUB, RUN).
- DELETE permet la suppression d'un groupe de lignes d'un programme.
- MEM précise la quantité de mémoire disponible.
- DUMP imprime la liste de toutes les valeurs courantes de variables numériques et chaînes.
- FIND recherche d'une chaîne de caractère spécifique.
- REPLACE remplace une chaîne par une autre.
- SAVE permet le transfert d'un programme au dessous de RAMTOP.
- APPEND supprime toutes les instructions REM d'un programme.

Ce programme occupe 2,4 K octets de mémoire et est livré avec une brochure détaillée de toutes ces fonctions. 86 F TTC

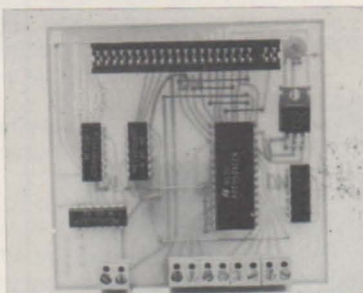
FAST LOAD MONITOR 64 K

Cette cassette développe la version 64 K du célèbre programme qui permet de réaliser ce que beaucoup tenaient pour impossible : générer la lecture et l'écriture de programmes sur K7 à la vitesse d'environ 4000 bauds (au lieu de 250 bauds sur le ZX81 d'origine). Cette cassette comporte 3 programmes différents.

- FAST LOAD MONITOR : programme principal

CARTE 8 ENTRÉES ANALOGIQUES POUR ZX81-ZX80

La carte « 8EA » permet d'entrer dans le ZX81 ou (ZX80) 8 signaux analogiques dont la tension est comprise entre 0 et 5 volts. Ces tensions pourront provenir de potentiomètres, de capteurs de position, de capteurs de température, etc... La



précision de la mesure est meilleure que 7 pour 1000. La carte comporte également une alimentation stabilisée 5 volts, qu'il est possible d'utiliser comme tension de référence pour alimenter les capteurs. Le résultat d'une mesure se trouve sous la forme d'une variable BASIC, qu'il est donc facile d'exploiter par programme. Trois cartes « 8ES » peuvent être simultanément commandées par le ZX81, ce qui permet un total de 24 entrées analogiques. Prix 420 F TTC.



CLAVIER MÉCANIQUE EN « ABS »

Se fixant directement sur le clavier initial du ZX81, ce clavier auto-collant en plastique «ABS», trois couleurs vous permet en quelques secondes de bénéficier d'un véritable clavier mécanique sans changer la configuration de votre ZX81. 160 F TTC

HARD

BIBLIOGRAPHIE

LA PRATIQUE DU ZX 81 (Tome II)

Destiné aux possesseurs de ZX81 ayant acquis une bonne expérience de la programmation Basic approfondie, l'ouvrage étudie le microprocesseur Z80-A en cinq étapes progressives et illustrées d'exemples : les opérations de base, les opérations complexes, les problèmes de l'affichage, les questions d'animation et la manière d'exploiter au mieux le programme moniteur. 72 F TTC

ÉTUDES POUR ZX81 (Tome II)

C'est plus particulièrement au langage assembleur appliqué aux modules d'extension comme l'imprimante ou la carte génératrice de caractères qu'est dédié le 2^e tome d'Études pour ZX81. Ses 20 Programmes vous permettront de créer des mélodies, de dessiner des histogrammes ou tout simplement de jouer au Baccara, aux Piranhas et au Taquin. 82 F TTC



constituez
dès maintenant votre
bibliothèque spécialisée

gratuit



sur simple demande
la liste complète de
tous nos titres.

Plus de 200 livres
classés par catégorie.

NIVEAU:

débutant à l'ingénieur.

Informatique, micro,
les langages, la programmation,
les matériels, technique
générale, électronique, etc.,

Je désire recevoir la liste gratuite de vos livres spécialisés.

ATTENTION : joindre un timbre à 1,80 francs, pour frais d'envois.

ptodis

3, rue du Colonel Moll - 75017 Paris

SIN 4

les expéditions de livres sont
faites en normal ou
recommandé, selon l'affran-
chissement demandé,

NOM _____ Prénom _____

N° _____ Rue _____

Ville _____ Code postal _____

NOS LECTEURS ONT DU GENIE

Delete

Une routine recherche d'adresses permet d'ajouter au ZX81 l'ordre BASIC DELETE : suppression d'une ou plusieurs lignes de programme. Associée à la routine «RECLAIM», elle est utilisée quand on modifie ou supprime une ligne, quand on efface les variables...

Toute la partie mémoire située au-dessus du dernier octet enlevé est décalée vers le bas.

Les pointeurs concernés sont modifiés (DFILE...)

Application

Économie de mémoire = suppression de lignes qui ne sont exécutées qu'une seule fois au début du programme. Si des textes apparaissent sur l'écran dès le début (par PRINT, PLOT...) il peut être intéressant de sauvegarder l'écran initialisé après avoir supprimé les lignes BASIC correspondantes.

Géographie : la France

Voici un programme de contrôle de vos connaissances en géographie. La carte de France se dessine sur l'écran et un texte défile expliquant le jeu. Un plot clignote sur la carte à l'emplacement d'une ville et deux noms s'inscrivent avec chacun un numéro.

L'ordinateur demande la réponse, la contrôle et donne son verdict : «CORRECT» ou «FAUX», avec le calcul et l'affichage du score. En cas de mauvaise

De plus, supprimer quelques lignes d'initialisation rend plus difficile la copie de nos cassettes.

Remarque : La routine «RECLAIM» peut aussi servir à supprimer une variable sans effacer les autres, à n'enlever que quelques octets d'une ligne.

F. Malard

```

ST 1 REM LN SCROLL *5? LN *** FA
2 LN TO *SGN ??TAN XXXX
3 REM * DELETE ASSEMBLEUR *
4 REM * FRANCOIS MALARD *
5 REM *
6 REM * CALL $02E7 CD E7 02
7 REM * LD HL,DEBUT 21 XX XX
8 REM * CALL $09D8 CD D8 09
9 REM * PUSH HL E5
10 REM * LD BC,FIN 01 XX XX
11 REM * INC BC 03
12 REM * CALL $09DF CD DF 09
13 REM * POP DE D1
14 REM * JP $0A5D C3 5D 0A
15 REM *
16 REM * PARTIE BASIC *
17 REM * "LIGNE DEBUT" *
18 PRINT L
19 INPUT L
20 POKE 16518,L-256*INT (L/256)
21
22 POKE 16519,INT (L/256)
23 PRINT "LIGNE,FIN"
24 INPUT L
25 POKE 16525,L-256*INT (L/256)
26
27 POKE 16526,INT (L/256)
28 USR 16514
29 RAND
30 STOP
31 FORST

```

```

5 REM GEOGRAPHIE:LA FRANCE
10 REM (C)RENE LAGACHE
15 REM
20 DIM F$(17,40)
25 LET F$(1)="2743264325432442
234123402339233822372137"
30 LET F$(2)="2037193620351934
183417351635153515361537"
35 LET F$(3)="1437133713361335
143414331432133212321132"
40 LET F$(4)="1032093308320732
063205320432053104300530"
45 LET F$(5)="0529062907290628
092810271126122612251224"
50 LET F$(6)="1223132214221521
152015191518151715161515"
55 LET F$(7)="1416141514141413
141214111410140913081307"
60 LET F$(8)="1306130514051505
180517041604190420032103"
65 LET F$(9)="2203230324022502
260227022602260329022804"
70 LET F$(10)="2805290529063002
6318732073307340735073607"
75 LET F$(11)="370638063905400
6410742064309431043114211"
80 LET F$(12)="411241134114401
5411642174118411941204021"
85 LET F$(13)="392139223823392
4402541264227422842294330"
90 LET F$(14)="433143324333423
4410440353935383537363636"
95 LET F$(15)="353634373336323
8313830393840294028412842"
100 LET F$(16)="491246114711461
0460946064607470648054806"
105 LET F$(17)="490749084909491
0491149114911491149114911"
500 FOR N=1 TO 17
505 FOR M=1 TO LEN F$(N)-3 STEP
4
510 LET X=VAL F$(N,M TO M+1)
515 LET Y=VAL F$(N,M+2 TO M+3)
520 PLOT X,Y
525 NEXT M
530 NEXT N
535 PRINT AT 1,0;"LA FRANCE";TA
B 22;"GEOGRAPHIE";TAB 22;"=====
===="
1000 PRINT AT 21,0;"APPUYER SUR
UNE TOUCHE"
1005 IF INKEY$="" THEN GOTO 1005
1010 LET A$="*****"
*****BONJOUR*****
*****VOICI LA CARTE D
E FRANCE,UN POINT VA CLIGNOTER S
UR L ECRAN ET 2 NOMS DE VILLES V
ONT APPARAÎTRE EN BAS DE L IMAGE
AVEC CHACUN UN NUMERO,REPONDEZ
PAR LE NUMERO CHOISI ET L ORDINA
TEUR CONTROLERA VOTRE REPONSE...
T...ATTENTION,ETES-VOUS PRÊ
T...? (OUI/NON)....."
1015 FOR N=1 TO LEN A$-31
1020 LET B$=A$(N TO N+31)
1035 PRINT AT 21,0;B$
1040 FOR P=1 TO 2
1045 NEXT P
1055 NEXT N
1060 INPUT Z$
1065 IF Z$(1)="O" THEN GOTO 1075
1070 IF Z$(1)="N" THEN GOTO 1010
1075 GOTO 1060

```

réponse, le nom de la ville s'inscrit.

A la fin du jeu, soit après 20 questions, l'ordinateur commente le score et vous invite à rejouer.

Décomposition du programme

Lignes 20 à 105 : Dimensionnement de la variable indicée F\$ (17,40) et calcul de chaque valeur attribuée à celle-ci. Ces valeurs se composent de 10 groupes de 2 Numéros de 2 chiffres, ex. pour F\$(1) = 2743..., 27 est la coordonnée I et 43 est la coordonnée Y du premier PLOT. (NOTA : pour la variable F\$(17) le fait d'avoir plusieurs fois la même valeur 4911 s'explique par la nécessité de terminer cette variable par des nombres, sinon l'ordinateur se plante sur l'erreur C).

Lignes 500 à 530 : Une boucle N sert de compteur pour la lecture des 17 valeurs de F\$, et une deuxième boucle M lit chaque groupe de 4 chiffres et calcule les coordonnées X et Y, puis positionne les PLOTS.

Ligne 535 : Donne le titre du jeu.

Lignes 1000 à 1005 : Invitent à jouer.

Lignes 1010 à 1050 : Initialisent dans la variable A\$ la

présentation du jeu et font défiler, sur la ligne 21, ce texte de droite à gauche en s'arrêtant sur la question «ETES VOUS PRET ?»

(OUI/NON). La boucle P règle la vitesse de défilement.

Lignes 1060 à 1071 : Posent la question et la contrôlent.

Lignes 1075 à 1235 : Redimensionnent la variable F\$ avec cette fois le nom des villes. L'originalité de ces valeurs est la présence dans les 4 premiers caractères du groupe des 2 numéros de 2 chiffres de la position du PLOT de la ville.

Ligne 1245 : Initialise le score S.

Ligne 1250 : Démarre la boucle N pour 20 questions.

Lignes 1255 à 1280 : Calculent le numéro de deux villes, les coordonnées I et Y de la première, et la positionne sur la carte.

Lignes 1282 à 1287 : Tirent l'ordre d'écriture du nom de la ville ainsi que du deuxième.

Lignes 1290 à 1305 : La boucle P fait clignoter le PLOT

Lignes 1310 à 1335 : Demandent la réponse, la contrôlent et affichent «FAUX» ou «CORRECT».

Lignes 1340 et 1345 : Calculent et affichent le score.

Ligne 1360 : Dans le cas d'une mauvaise réponse, donnent la bonne.

Lignes 1372 à 1375 : Effacent le PLOT et retournent à la ligne 1250.

```

1205 LET F$(26) = "1922POITIERS"
1210 LET F$(27) = "3631NANCY"
1215 LET F$(28) = "1726ANGERS"
1220 LET F$(29) = "2331CHARTRES"
1225 LET F$(30) = "2318LIMOGES"
1230 LET F$(31) = "4708AJACCIO"
1235 LET F$(32) = "48105ASTIA"
1245 LET S=0
1250 FOR N=1 TO 20
1255 LET V=INT (RND*32)+1
1260 LET T=INT (RND*32)+1
1265 IF T=V THEN GOTO 1260
1270 LET X=VAL F$(V,1 TO 2)
1275 LET Y=VAL F$(V,3 TO 4)
1280 PLOT X,Y
1281 PRINT AT 21,0; ""
1282 LET H=INT (RND*2)+1
1283 IF H=1 THEN GOTO 1287
1285 PRINT AT 21,0;V;"-";F$(V,5
TO ),T;"-";F$(T,5 TO )
1286 GOTO 1290
1287 PRINT AT 21,0;T;"-";F$(T,5
TO ),V;"-";F$(V,5 TO )
1290 FOR P=1 TO 30
1295 UNPLOT X,Y
1300 PLOT X,Y
1305 NEXT P
1310 PRINT AT 3,24;"REPOSE?"
1315 INPUT Z$
1320 IF CODE Z$(1)<29 OR CODE Z$(
11)>37 THEN GOTO 1315
1325 LET Z=VAL Z$
1330 IF Z<>V THEN PRINT AT 3,24;
"FAUX"
1335 IF Z=V THEN PRINT AT 3,24;
"CORRECT"
1340 IF Z=V THEN LET S=S+1
1345 PRINT AT 6,25;"SCORE:";AT 8
,25;S;" / ";N
1350 FOR P=1 TO 10

```

```

1075 DIM F$(32,15)
1080 LET F$(1) = "2633PARIS"
1085 LET F$(2) = "2235ROUEN"
1090 LET F$(3) = "2739ARRAS"
1095 LET F$(4) = "1330RENNES"
1100 LET F$(5) = "1425NANTES"
1105 LET F$(6) = "1713BORDEAUX"
1110 LET F$(7) = "2307TOULOUSE"
1115 LET F$(8) = "0631BREST"
1120 LET F$(9) = "3417LYON"
1125 LET F$(10) = "3707MARSEILLE"
1130 LET F$(11) = "4109NICE"
1135 LET F$(12) = "3826BESANCON"
1140 LET F$(13) = "4232STRASBOURG"
1145 LET F$(14) = "1906TARBES"
1150 LET F$(15) = "3114LE PUY"
1155 LET F$(16) = "2425BOURGES"
1160 LET F$(17) = "1927TOURS"
1165 LET F$(18) = "2036LE HAVRE"
1170 LET F$(19) = "3717CHAMBERY"
1175 LET F$(20) = "1928LE MANS"
1180 LET F$(21) = "3425DIJON"
1185 LET F$(22) = "3134REIMS"
1190 LET F$(23) = "3734METZ"
1195 LET F$(24) = "1028VANNES"
1200 LET F$(25) = "1619LA ROCHELLE"

```

```

1355 NEXT P
1360 IF Z<>V THEN PRINT AT 21,0;
"REPOSE JUSTE:";V;"-";F$(V,5 TO
15)
1365 FOR P=1 TO 20
1370 NEXT P
1372 UNPLOT X,Y
1373 PRINT AT 3,24; ""
1375 NEXT N
1380 IF S<8 THEN PRINT AT 21,0; "
CE N EST PAS TRES BON,REVISEZ"
1385 IF S>8 OR S<15 THEN PRINT AT
21,0; "C EST UN SCORE MOYEN"
1390 IF S>15 AND S<20 THEN PRINT
AT 21,0; "C EST UN BON SCORE"
1395 FOR P=1 TO 30
1400 NEXT P
1405 PRINT AT 21,0; "APPUYEZ SUR
UNE TOUCHE SUP"
1410 IF INKEY$="" THEN GOTO 1410
1415 CLS
1420 GOTO 20

```

NOS LECTEURS ONT DU GENIE

Lignes 1380 à 1390 : Donnent un commentaire sur le score.

Lignes 1405 à 1420 : Demandent de redémarrer le jeu

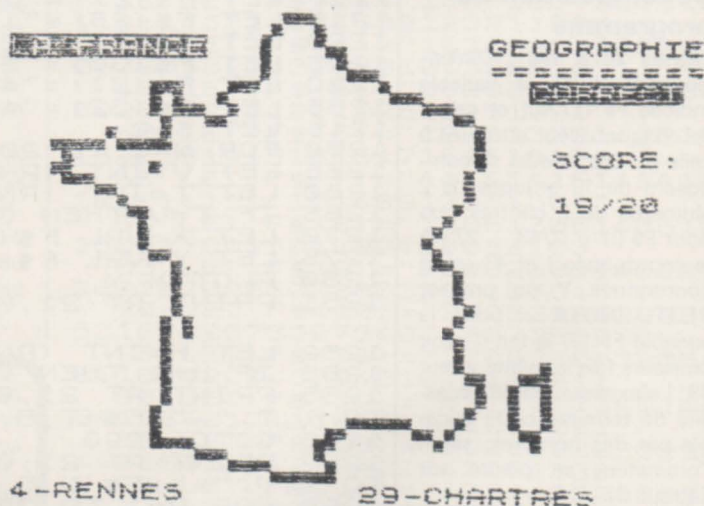
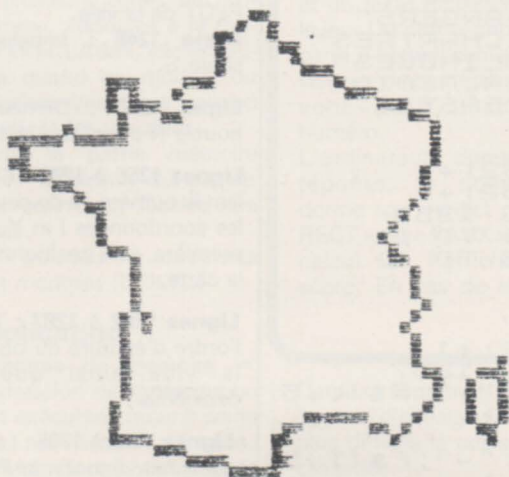
Remarques :

Les boucles P, lignes 1350/1355, 1365/1370 et

1395/1400 déterminent le temps de pause.

Le nombre de villes est ici de 32, mais il est possible d'avoir plus de noms en n'oubliant pas de modifier la valeur 32 des lignes 1255 et 1260 et en déplaçant le reste du programme.

R. Lagache



INFORMATIQUE

Spécialistes en France

Si vous voulez apparaitre dans cette rubrique
TELEPHONEZ AU 201 04 50

Près de 50 000 ZX81
sont utilisés en France,
et ce n'est pas fini !

Aujourd'hui, un nombre
considérable de périphériques
d'extensions et de
programmes sont disponibles.



Pour être tenu au courant
de ces nouvelles possibilités
d'emploi de votre
Sinclair et pour avoir
accès aux « trésors cachés »
de votre micro-ordinateur,

nous avons créé une revue spécialisée pour vous

l'indispensable

Echos sinclair

Ce magazine est un bimestriel (6 numéros par an)
vendu par abonnement au prix de 20 F
le numéro, soit 120 F par an. Mais pour
son lancement, une offre
spéciale d'abonnement au prix de
100 F les 6 numéros vous
est offerte. Profitez de cette
offre de lancement
en retournant dès
aujourd'hui le
bon ci-joint.

**ABONNEZ
VOUS**

Je souhaite m'abonner à « Echos » **sinclair**
au prix exceptionnel de lancement
de 100 F pour 6 numéros.

Bon et chèque, mandat postal
ou CCP à retourner à
Joker Editions,
12, Villa Saint-Michel,
75018 Paris.

Nom

N°

Rue

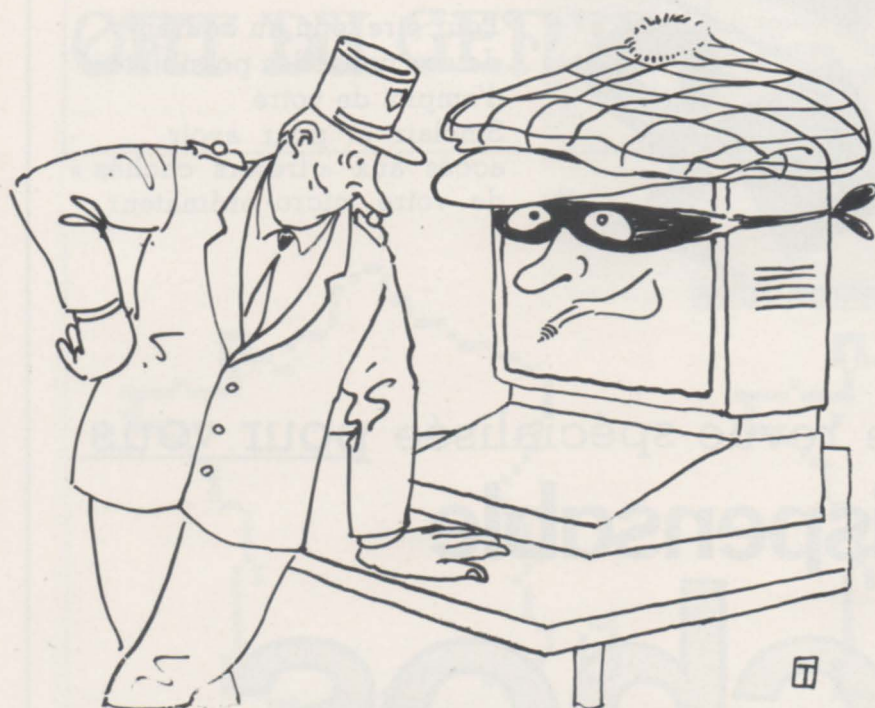
Ville

Prénom

Code Postal

« Echos »
sinclair

LOGICIELS



NOS LECTEURS S'AMUSENT

Gendarmes et voleurs

Ce jeu se joue sur un échiquier 8x8 dont les cases sont numérotées de 11 à 88. Du côté de la machine, 8 pièces : 3 voleurs, 3 innocents, 1 chef voleur et 1 chef innocent.

Du côté du joueur 8 pièces également : 7 gendarmes et 1 chef gendarme.

Le but du jeu

La machine doit faire passer un des voleurs de l'autre côté du jeu.

Les règles du jeu

Pour la machine : les pièces avancent d'une case à la fois en diagonale.

Pour le joueur : les pièces avancent ou reculent d'une case à la fois.

Seul le chef gendarme a le droit d'arrêter un chef. Le joueur ne sait pas qui sont les voleurs ni les innocents, seuls les chefs sont connus de lui.

Pour appréhender un suspect, le gendarme doit venir sur sa case. La machine indique alors si le suspect est un voleur, auquel cas le voleur est arrêté et retiré du jeu.

Si le suspect est innocent, c'est le gendarme qui, révoqué, est retiré du jeu. La machine n'accepte que les coups valides.

Le joueur choisit parmi les 8 gendarmes le chef (en inversion vidéo à l'écran).

Au début du jeu, le règle est donnée si le joueur la demande et le programme mélange ses pièces en mode FAST. Le reste du jeu est en mode SLOW et les réponses sont instantanées... (oui ! quasiment !).

A la fin du jeu, des commentaires indiquent qui a gagné, le programme ou le joueur.

Les déplacements se font en donnant la case de départ et la case d'arrivée (s'il y a prise d'une pièce, le programme la traite automatiquement).

La stratégie de la machine est simple mais redoutablement efficace.

```

1 REM GENDARMES ET VOLEURS
2 PRINT TAB 5: "GENDARMES ET VOLEURS"
3 PRINT TAB 5: "=====
4 PRINT
5 PRINT
6 PRINT "VOULEZ-VOUS LES REGL
ES DU JEU ?"
7 INPUT R$
8 CLS
9 IF R$(1)<>"N" THEN GOTO 30
10 FAST
11 RAND
15 DIM T(100)

```

```

20 DIM Z(8)
25 LET S$="XIXIGG- "
90 FOR I=1 TO 100
100 LET T(I)=7
110 LET I1=I-INT (I/10)*10
120 IF I1=0 OR I1=1 OR I<12 OR
I>89 THEN LET T(I)=8
130 NEXT I
140 FOR I=2 TO 9
150 LET T(10+I)=1
160 LET T(80+I)=5
165 NEXT I
170 FOR I=1 TO 5
180 LET Z(I)=INT (RND*8+12)
190 FOR J=1 TO 5
200 IF J=I THEN GOTO 230
210 IF Z(I)=Z(J) THEN GOTO 180
220 NEXT J
230 NEXT I
240 LET T(Z(1))=2
250 LET T(Z(2))=3
260 LET T(Z(3))=3
270 LET T(Z(4))=3
280 LET T(Z(5))=4
300 FOR I=6 TO 8
301 FOR J=12 TO 19
310 IF Z(1)<>J AND Z(2)<>J AND
Z(3)<>J AND Z(4)<>J AND Z(5)<>J
AND Z(6)<>J AND Z(7)<>J THEN GO
TO 1800
340 NEXT J
360 NEXT I
370 GOSUB 2000
375 SLOW
380 PRINT AT 18,0;"OU PLACEZ-VO
US VOTRE CHEF ?"
390 INPUT I
395 PRINT AT 18,0;"
"
400 IF I<81 OR I>88 THEN GOTO
380
420 LET I=I+1
430 LET T(I)=6
440 PRINT AT VAL (STR$ I)(1)*2,
VAL (STR$ I)(2)*2-1;"G"
450 PRINT AT 18,0;"OU JOUEZ-VOU
S ?"
455 PRINT AT 19,0;"
"
460 INPUT I$
467 PRINT AT 18,0;"
"
468 IF VAL I$<1000 OR VAL I$>88
77 THEN GOTO 450
470 LET D=VAL I$(1 TO 2)+1
480 LET A=VAL I$(3 TO 4)+1
485 IF ABS (A-D)<>11 AND ABS (A
-D)<>9 THEN GOTO 450
490 IF T(D)<>5 AND T(D)<>6 THEN
GOTO 450
500 IF T(A)<>7 THEN GOTO 540
505 LET X=T(A)
510 LET T(A)=T(D)
520 LET T(D)=X
530 GOTO 660
540 IF T(D)=6 AND (T(A)=2 OR T(
A)=1) THEN GOTO 600
550 IF T(D)=5 AND T(A)=1 THEN
GOTO 600
560 IF T(D)=6 AND (T(A)=4 OR T(
A)=3) THEN GOTO 630
570 IF T(D)=5 AND T(A)=3 THEN
GOTO 630
580 PRINT AT 19,0;"ERREUR.....
RECOMMENCEZ"
583 FOR P=1 TO 50
584 NEXT P
590 GOTO 450
600 PRINT AT 21,0;"BRAVO, VOUS
ARRETEZ UN VOLEUR ."
605 LET T(A)=7

```

```

610 GOTO 505
630 PRINT AT 21,0;"DOMMAGE, C E
ST UN INNOCENT ."
640 LET T(D)=7
660 PRINT AT VAL (STR$ D)(1)*2,
VAL (STR$ D)(2)*2-1;"-"
665 PRINT AT VAL (STR$ A)(1)*2,
VAL (STR$ A)(2)*2-1;S$(T(A))
666 PRINT AT 21,0;"
"
670 LET F1=0
680 FOR I=1 TO 8
700 IF T(Z(I))=1 OR T(Z(I))=2 T
HEN GOTO 820
710 NEXT I
720 PRINT "J*AI PERDU.PLUS DE V
OLEURS."
730 PRINT
740 PRINT "APPUYEZ SUR UNE TOUC
HE"
745 IF INKEY$="" THEN GOTO 745
750 CLS
760 PRINT "VOULEZ-VOUS REJOUER
?"
770 INPUT R$
775 IF R$(1)="O" THEN RUN
780 IF R$(1)="N" THEN GOTO 800
790 GOTO 770
800 PRINT AT 10,10;"AU REVOIR"
810 STOP
820 FOR I=82 TO 89
860 IF T(I)=1 OR T(I)=2 THEN G
OTO 1900
870 NEXT I
875 LET D2=0
880 LET P=INT (RND*8+1)
900 LET P1=P
910 IF T(Z(P))=1 OR T(Z(P))=2 O
R T(Z(P))=3 OR T(Z(P))=4 THEN G
OTO 1000
920 LET P=P+1
925 LET D2=D2+1
930 IF P=9 THEN LET P=1
940 IF P<>P1 THEN GOTO 910
950 IF F1=0 THEN GOTO 720
960 LET P=K2
965 LET D1=K1
967 GOTO 1060
1000 LET D=RND
1010 IF D<=.5 THEN LET D1=9
1030 IF D>.5 THEN LET D1=11
1035 IF T(Z(P)+D1)<>7 THEN GOTO
1100
1040 IF T(Z(P)+D1)>89 THEN GOTO
1200
1060 LET T(Z(P)+D1)=T(Z(P))
1070 LET T(Z(P))=7
1080 PRINT AT VAL (STR$ Z(P))(1)
*2,VAL (STR$ Z(P))(2)*2-1;"-"
1082 LET Z(P)=Z(P)+D1
1085 PRINT AT VAL (STR$ Z(P))(1)
*2,VAL (STR$ Z(P))(2)*2-1;S$(T(Z
(P)))
1090 GOTO 450
1100 IF D1=9 THEN LET A=1
1110 IF D1=11 THEN LET A=0
1120 IF A=0 THEN LET D1=9
1130 IF A=1 THEN LET D1=11
1140 LET D2=D2+1
1150 IF D2=2 THEN GOTO 920
1160 GOTO 1035
1200 LET K1=D1
1210 LET K2=P
1220 LET F1=1
1230 GOTO 920
1800 LET Z(I)=J
1810 GOTO 360
1900 LET T(I)=7
1905 PRINT AT 16,VAL (STR$ I)(2)
*2-1;"-"
1910 PRINT AT 21,0;"J*AI GAGNE."

```

```

* ..... "
1920 STOP
2005 PRINT AT 0,0;"      1 2 3 4 5
6 7 8"
2010 PRINT
2020 FOR I=1 TO 8
2030 PRINT " ";I;" ";
2040 FOR J=10*I+2 TO 10*I+9
2050 PRINT S$(T(J));" ";
2060 NEXT J
2070 PRINT
2080 PRINT
2085 NEXT I
2090 RETURN
3000 PRINT
3010 PRINT " ZX A 4 VOLEURS DONT
UN CHEF, ET4 INNOCENTS DONT UN
CHEF (X,1) ."
3014 PRINT " SON BUT EST DE FAIR
E PASSER UN VOLEUR SUR LA LIGNE
SUD DU JEU . "
3016 PRINT
3018 PRINT "VOUS DISPOSEZ POUR L
EN EMPÊCHER 8 GENDARMES DONT
UN CHEF (G,8) "
3022 PRINT " UN GENDARME PEUT AV
ANCER OU RE-CULER D UNE CASE EN
DIAGONALE, "
3024 PRINT "UN VOLEUR NE PEUT QU
AVANCER . "
3026 PRINT
3028 PRINT " POUR BLOQUER UN SUS
PECT, IL FAUT SE PLACER SUR SON CH
EMIN ; POUR L ARRETER, IL FAUT O
CCUPER SA "
3030 PRINT "CASE . SEUL LE CHEF
GENDARME PEUT ARRETER UN CHEF
SUSPECT . "
3034 PRINT " UN GENDARME QUI ARR
ETE UN INNO-CENT EST REVOCUE, UN
VOLEUR AR- RETE EST EMPRISONNE
. "
3040 PRINT
3050 PRINT "APPUYEZ SUR UNE TOUC
HE S.V.P. "
3060 IF INKEY$="" THEN GOTO 306
0
3070 CLS
3080 GOTO 10

```

Thésée dans le labyrinthe

Le joueur est Thésée (T en inversion vidéo) et la machine le Minotaure (M en inversion vidéo). Thésée doit rejoindre la sortie du labyrinthe sans se faire dévorer par le Minotaure. Le tracé du labyrinthe est tiré aléatoirement au début du jeu et comporte toujours une issue. Les déplacements de Thésée se font par les 4 touches de direction du curseur. Le déplacement du monstre varie de 1 à 2 cases dans la proportion 6 fois 1 case pour 1 fois 2 cases, d'une seule case à la fois. Il n'y a pas de déplacements obliques. Tout le programme est réalisé en mode SLOW et l'affichage est fait par « POKES ». Très joli graphisme et temps de réponse inférieur à la seconde. Bel effet quand Thésée se fait dévorer et le labyrinthe se bouche dès que Thésée s'est échappé. La stratégie de la machine est de calculer l'abscisse et

l'ordonnée de la disance entre Thésée et le Minotaure et de retirer 1 ou d'ajouter 1 à l'un de ces paramètres pour que la distance soit la plus courte possible. Simple mais efficace !

```

10 REM PROLOGUE
20 CLS
30 PRINT TAB 4;"THESEE ET LE M
INDTAURE."
40 PRINT
50 PRINT "VOULEZ-VOUS CONNAITR
E LES REGLES DU JEU ? (O/N)"
60 IF INKEY$="" THEN GOTO 60
70 IF INKEY$<>"D" THEN GOTO 5
00
90 GOTO 9600
499 REM LABYRIINTE
500 CLS
510 PRINT "I
"
520 FOR A=1 TO 4
530 PRINT ":";TAB 30;":"
540 FOR B=1 TO 3
550 PRINT "
"
560 NEXT B
570 NEXT A
580 PRINT ":";TAB 30;":"
590 PRINT "
"
600 LET C=1+PEEK 16396+256*PEEK
16397
610 RAND
620 FOR D=0 TO 500 STEP 100
630 GOSUB 1000+D+10*INT (RND*3)
640 FOR E=1 TO 3
650 LET F=INT F/1000
660 LET G=INT (1000*(F-INT F)+.
5)
670 POKE C+G,136
680 POKE C+G+1,136
690 POKE C+G-1,136
700 POKE C+G+33,136
710 POKE C+G-33,136
720 NEXT E
730 NEXT D
735 REM PLACEMENT T ET M
740 LET U=1
750 LET V=1
760 LET T=C+33*V+U
770 POKE T,185
780 POKE C+1,136
790 LET N=INT (RND*14)+15
800 LET O=INT (RND*8)+9
810 LET M=C+33*O+N
820 IF PEEK M<>0 THEN GOTO 790
830 POKE M,178
840 LET Z=0
850 LET Y=0
860 GOTO 2000
1000 LET F=36104172
1005 RETURN
1010 LET F=40168232
1015 RETURN
1020 LET F=100236240
1025 RETURN
1100 LET F=44180248
1105 RETURN
1110 LET F=48176184
1115 RETURN
1120 LET F=52112244
1125 RETURN
1200 LET F=56128252
1205 RETURN
1210 LET F=188192256
1215 RETURN
1220 LET F=60124260
1225 RETURN
1300 LET F=304372564

```

Tous les caractères soulignés
doivent apparaître en vidéo inversée.

NOS LECTEURS S'AMUSENT

```

1305 RETURN
1310 LET F=300432568
1315 RETURN
1320 LET F=364436504
1325 RETURN
1400 LET F=312440580
1405 RETURN
1410 LET F=508512308
1415 RETURN
1420 LET F=316448572
1425 RETURN
1500 LET F=384520524
1505 RETURN
1510 LET F=388516588
1515 RETURN
1520 LET F=392452584
1525 RETURN
2000 REM DEPLACEMENT M
2005 FOR W=1 TO INT (RND*1.2+1)
2045 GOSUB 8000
2050 LET R=SGN H
2060 LET S=SGN I*33
2070 IF ABS I>ABS H THEN GOTO 2
250
2080 IF PEEK (M+R)<>0 OR Y=1 THE
N GOTO 2120
2090 LET K=R
2100 LET Z=0
2110 GOTO 2420
2120 IF PEEK (M+S)<>0 OR Z=1 THE
N GOTO 2160
2130 LET K=S
2140 LET Y=0
2150 GOTO 2420
2160 IF PEEK (M-S)<>0 THEN GOTO
2210
2170 LET K=-S
2180 LET Y=0
2190 LET Z=1
2200 GOTO 2420
2210 LET K=-R
2220 LET Y=1
2230 LET Z=0
2240 GOTO 2420
2250 IF PEEK (M+S)<>0 OR Z=1 THE
N GOTO 2290
2260 LET K=S
2270 LET Y=0
2280 GOTO 2420
2290 IF PEEK (M+R)<>0 OR Y=1 THE
N GOTO 2330
2300 LET K=R
2310 LET Z=0
2320 GOTO 2420
2330 IF PEEK (M-R)<>0 THEN GOTO
2380
2340 LET K=-R
2350 LET Y=1
2360 LET Z=0
2370 GOTO 2420
2380 LET K=-S
2390 LET Y=0
2400 LET Z=1
2420 POKE M,0
2430 LET M=M+K
2440 POKE M,178
2450 LET O=INT ((M-C)/33)
2460 LET N=M-C-33*O
2480 GOSUB 8000
2490 NEXT W
3999 REM DEPLACEMENT T

```

```

4000 LET A$=INKEY$
4005 LET L=0
4010 IF A$="5" THEN LET L=-1
4020 IF A$="6" THEN LET L=33
4030 IF A$="7" THEN LET L=-33
4040 IF A$="8" THEN LET L=1
4050 IF PEEK (T+L)<>0 THEN GOTO
4120
4060 POKE T,0
4070 LET T=T+L
4080 POKE T,185
4090 LET V=INT ((T-C)/33)
4100 LET U=T-C-33*V
4110 IF T=C+656 THEN GOTO 9500
4120 GOTO 2000
8000 LET H=U-N
8010 IF H=0 THEN LET H=-1
8020 LET I=V-O
8030 IF I=0 THEN LET I=-1
8040 IF SQR (H*H+I*I)<2 THEN GO
TO 9000
8050 RETURN
8999 REM MORT DE THESEE
9000 POKE T,130
9005 RAND
9010 POKE T,134
9015 RAND
9020 POKE T,1
9025 RAND
9030 POKE T,0
9040 PRINT AT 20,5;"THESEE A ETE
DEVORE ."
9050 STOP
9499 REM VICTOIRE
9500 POKE C+623,136
9510 PRINT AT 20,5;"THESEE S EST
ECHAPPE ."
9600 REM REGLES DU JEU
9605 PRINT
9610 PRINT "VOUS ETES THESEE (I)
ET VOUS VOUS TROUVE DANS UN LAB
YRYNTHÉ. LE MINOTAURE (M) VEUT VOUS
DEVORER ."
9620 PRINT "VOS DEPLACEMENTS SON
T LES 4 FLECHES (5 A 8) CURSEUR.
"
9630 PRINT
9640 PRINT "APPUYEZ SUR UNE TOUC
HE ET....."
9650 PRINT "BONNE CHANCE....."
....."
9660 IF INKEY$="" THEN GOTO 966
0
9670 GOTO 500

```

Dans la lune

Les programmes d'alunissage sont de grands classiques. Pour la première fois dans notre revue, paraît un exemple de ce type de jeux.

Ce programme nous a été envoyé par un lecteur qui a omis d'indiquer son nom mais qui se reconnaîtra sûrement !

```

40 LET M=75
50 LET T=0
60 LET X=2350
70 LET V=0
80 LET L=300

```

```

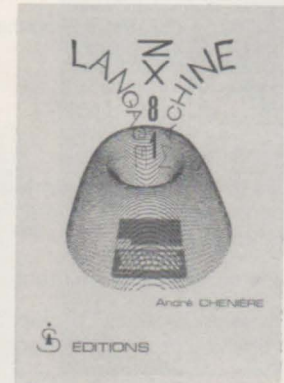
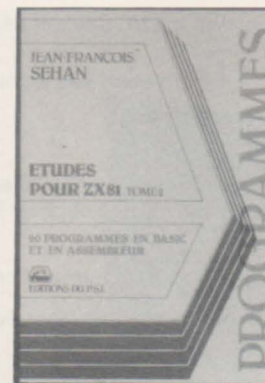
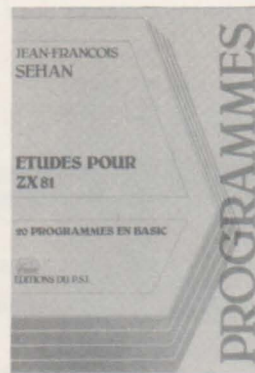
90 CLS
100 PRINT
110 PRINT " ----A.L.U.N.I.S.S.
A.G.E----"
120 PRINT
130 PRINT " TEMPS----";INT T
140 PRINT " ALTITUDE-";INT X
150 PRINT " VITESSE---";INT V;"
-M/S"
160 IF V<0 THEN PRINT " DESCEN
TE-";
170 IF V>=0 THEN PRINT " MONTE
E---";
180 PRINT 3.6*ABS V;"-KM/H"
190 PRINT " CARBURANT:";INT L
200 IF L<=0 THEN GOTO 450
210 PRINT
220 PRINT "FUEL-?";
230 INPUT L1
240 PRINT L1
250 IF ABS L1>M THEN GOSUB 480
260 IF NOT L1 THEN GOSUB 590
270 LET L=L-ABS L1
280 IF L<=0 THEN GOTO 530
290 LET T=T+1
300 LET X=X+V-(5-L1)/2
310 IF INT X<=0 THEN GOTO 340
320 LET V=V-(5-L1)
330 GOTO 90
340 IF INT V<-20 THEN GOTO 570
350 PRINT
360 GOTO 400
370 PRINT "VOUS AVEZ TOUCHE LA
LUNE"
380 PRINT "A LA VITESSE DE-";L;
INT (3.6*V);"-KM/H"
390 PRINT "IL VOUS RESTE-";L;"L
ITRES DE FUEL"
400 PRINT ",,,";"VOULEZ-VOUS RECO
MMENCER-U"
410 INPUT Z$
420 IF CHR$ CODE Z$="O" THEN G
OTO 40
430 PRINT "TERMINE"
440 STOP
450 PRINT "ALARME"
460 LET L1=0
470 GOTO 280
480 LET L1=M
490 PRINT "MAX= ";M;"-LITRES"
500 FOR G=1 TO 50
510 NEXT G
520 RETURN
530 LET L1=ABS (L1+L)
540 LET L=0
550 PRINT "RESERVOIR.VIDE"
560 GOTO 400
570 PRINT "CRRASHHH"
580 GOTO 400
590 PRINT "TEMPS DE DERIVE-?";
600 INPUT DER
610 PRINT DER
620 FOR I=1 TO DER-1
630 LET T=T+1
640 LET X=X+V-(5-L1)/2
650 IF INT X<=0 THEN GOTO 340
660 LET V=V-(5-L1)
670 NEXT I
680 RETURN
9990 SAVE "ALUN"

```

Tous les caractères
soulignés
doivent apparaître
en inversion vidéo.

BIBLIOGRAPHIE

PROGRAMMES, ANALYSES, CONSEILS SONT LES DIFFERENTS THEMES DEVELOPPES PAR DE NOMBREUX OUVRAGES EDITIONES AUTOUR DU ZX81
ILS SE REVELENT D'EXCELLENTS COMPLEMENTS AU MANUEL D'INITIATION FOURNI AVEC LE MICRO-ORDINATEUR.



JE DESIRE RECEVOIR :

- | | | | |
|---|-------|--|-------|
| ☐ LA CONDUITE DU ZX81 | 55.00 | ☐ ETUDE POUR ZX81 (tome 1) | 82.00 |
| ☐ LE PETIT LIVRE DU ZX81 | 72.00 | ☐ (tome 2) | 82.00 |
| ☐ LA PRATIQUE DU ZX81 (tome 1) | 72.00 | ☐ TRUCS ET ASTUCES EN LANGAGE MACHINE | 75.00 |
| ☐ (tome 2) | 82.00 | ☐ LE ZX81 A LA CONQUETE DES JEUX | 65.00 |
| ☐ PILOTEZ VOTRE ZX81 | 63.00 | ☐ ET SA CASSETTE | 65.00 |
| ☐ PROGRAMMATION EN LANGAGE MACHINE | 96.00 | ☐ 70 PROGRAMMES POUR ZX81 ET SPECTRUM | 60.00 |

BON DE COMMANDE à envoyer à DIRECO International, 30 avenue de Messine - 75008 PARIS

Nom : _____ Prénom : _____

Adresse : _____

(Code postal complet) _____

A _____ le _____

Signature

Paiement

- ☐ par C.C.P. ou Chèque Bancaire à l'ordre de DIRECO international
- ☐ par Contre remboursement moyennant une taxe de 14 F (Montant minimum de commande : 200 F)



LES NOUVEAUTES DU ZX 81

Ordinateur domestique par excellence, le ZX 81 a d'autres cordes à son arc. Outre les possibilités de branchement de l'imprimante et de l'extension de mémoire 16 K RAM, une gamme très complète de périphériques est proposée pour augmenter ponctuellement les propriétés de votre micro-ordinateur.

(Carte : haute définition, son, génératrice de caractères, entrées sorties, extensions de mémoires, etc...).

D'un montage aisé, comme l'imprimante et la 16 K, ces extensions se connectent directement sur votre unité de base par le connecteur arrière et vous permettent ainsi de faire évoluer votre machine en fonction de vos besoins précis.

Le logiciel sur cassette :

Une large gamme de logiciels est disponible. Ceux-ci développent des thèmes aussi différents que :

- Jeux de réflexion.
- Jeux animés.
- Gestion.
- Formation des jeunes.
- Utilitaire.

Ces différents programmes dont les prix varient entre 50 et 150 F, vont permettre aux néophytes comme aux programmeurs chevronnés d'accéder immédiatement à des programmes très performants.

Une recherche permanente dans ce domaine permet d'enrichir constamment cette gamme.

Outils incontestés, la plupart de ces programmes sont écrits en BASIC et vous permettront d'analyser leur élaboration.

D'autres conçus en langage machine rapide et efficace, feront de vous des utilisateurs de puissants logiciels ou vous transporteront dans le monde des jeux électroniques.

- 1 - 64 K
- 2 - Interface Centronics — carte Bus — câble
- 3 - 8 Entrées - Sorties
- 4 - Clavier mécanique
- 5 - Connecteur, Carte mère

- 6 - Carte haute définition graphique
- 7 - Carte sonore
- 8 - Carte génératrice de caractères
- 9 - Z P 82
- 10 - Clavier ABS
- 11 - Imprimante

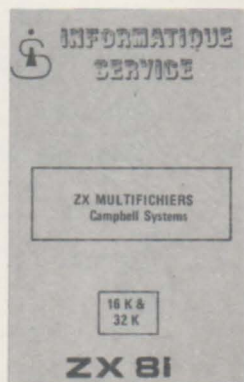
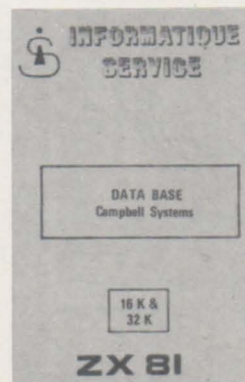
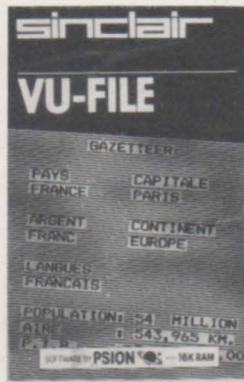
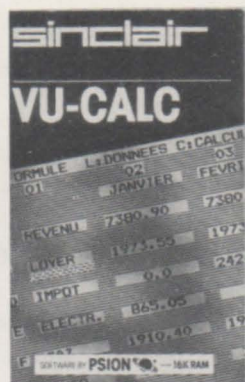
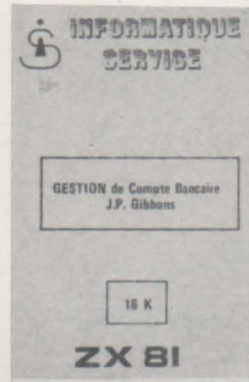
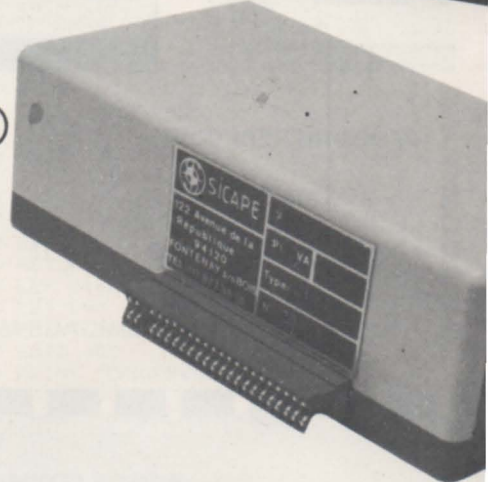
10

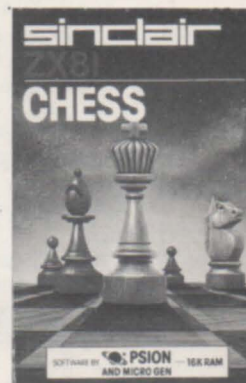
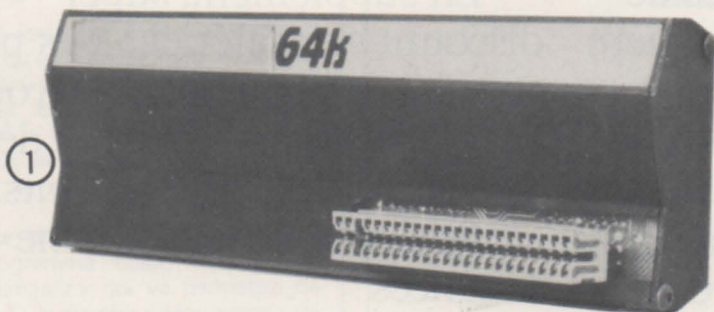
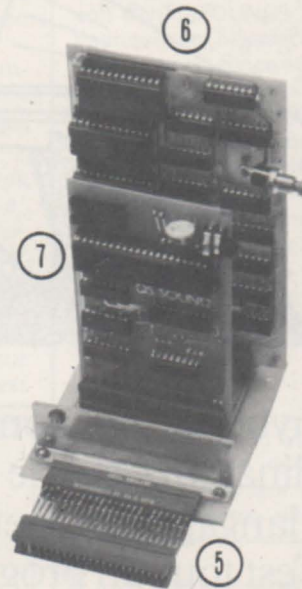
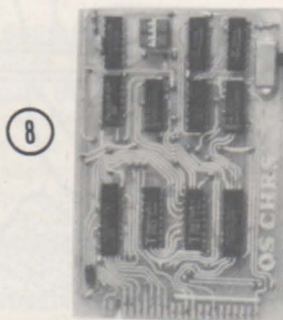
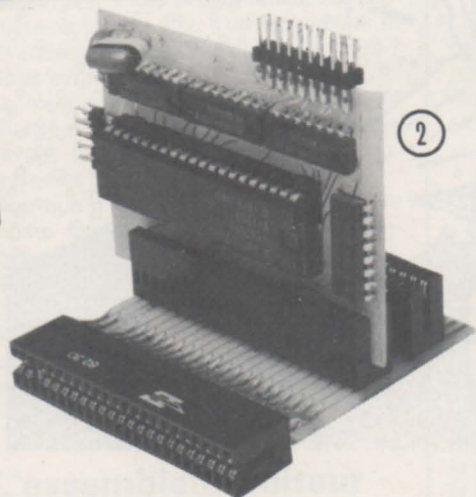
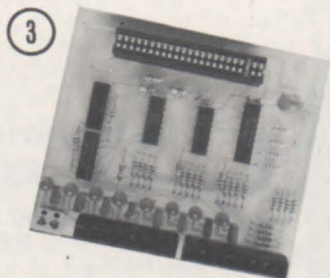
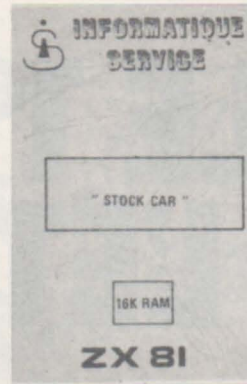
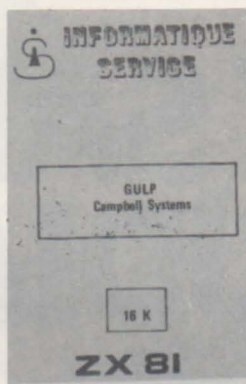
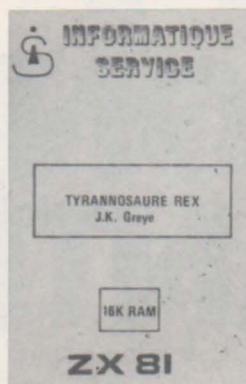
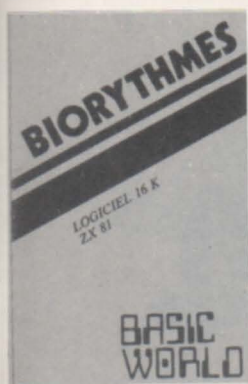


11

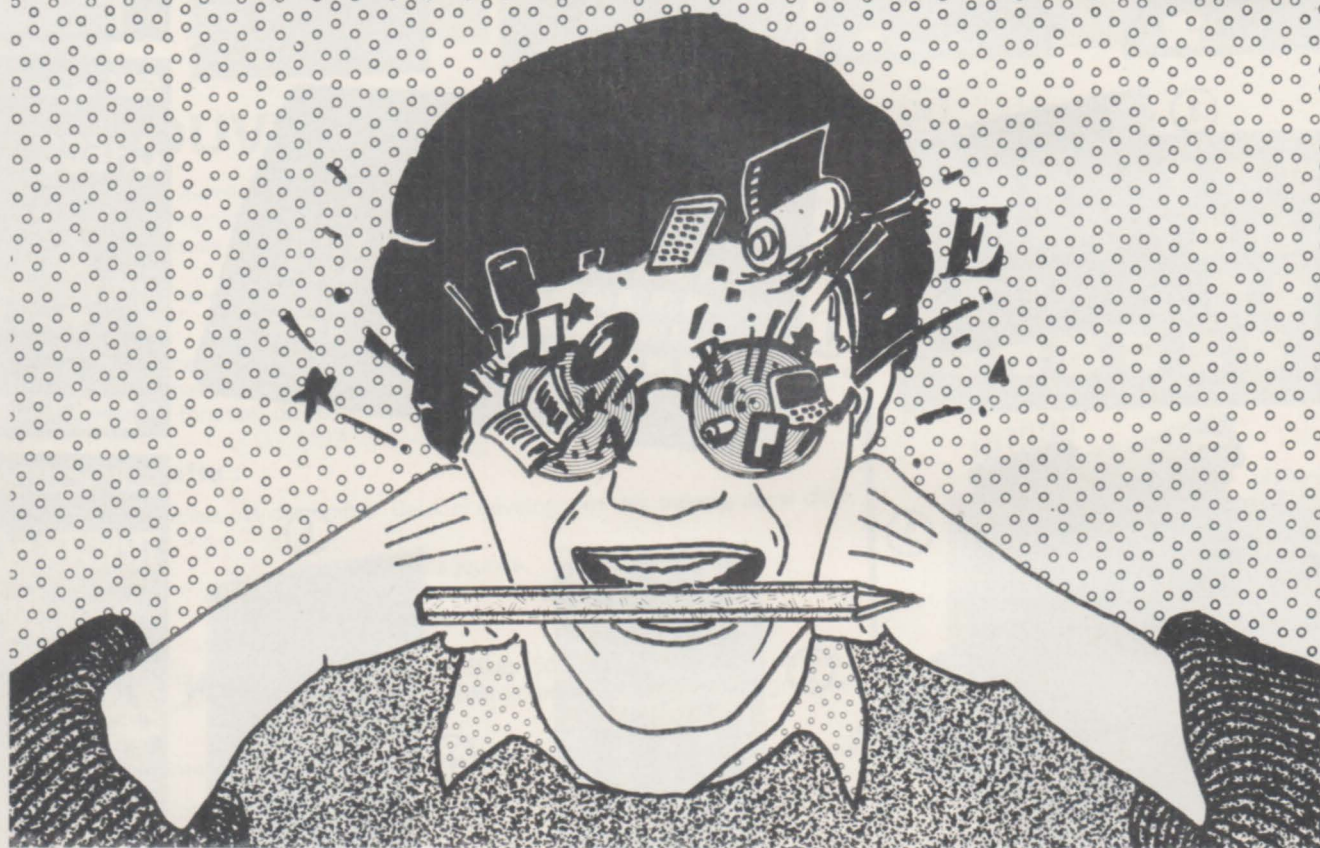


9





LE 1^{er} SALON DE L'ETUDIANT(E).



Néophytes ou passionnés de micro-ordinateurs, c'est le moment de parler langages, jeux et trucs : le Salon, c'est tout un programme !

Au Salon, il y aura tout ce qui vous concerne, pour la première fois en couleurs, en relief et en stéréo : l'avenir, les loisirs, la fête, les affaires. Et des forums sur tout : emploi et formation, XXI^e siècle, micro-informatique, littérature, sport, cinéma, photo, musique, voyages, aventures...

En supplément, super-discount de matériel à des prix scandaleux, concerts pop-rock-classique non-stop, grand débit de débats, films permanents.

Bref, une super-semaine de vacances !

1^{er} Salon de l'Etudiant(e).
Quai d'Austerlitz.
19-27 février. De 11 h 30 à 22 h 30
Sans parler des nocturnes !

Banc d'essai

ZX assembleur-éditeur/ moniteur-déassembleur

Deux programmes pour ZX81 16 Ko

(ou : deux utilitaires professionnels
pour le ZX81)

Quand, pour la première fois, j'ai entre les mains les deux cassettes correspondant à ces programmes, ma première réaction a été : « un assembleur et un déassembleur ? Pour quoi faire ? Nous avons déjà ZXAS et ZXDB ! ».

Ce n'est que lorsque l'assembleur a été chargé en mémoire que j'ai compris que j'avais, là, affaire à de véritables programmes professionnels.

Bien que n'ayant pas été conçus par le même fabricant, ces deux programmes, par les puissantes fonctions et facilités qu'ils offrent, semblent être faits l'un pour l'autre : amateurs de langage machine, voici dorénavant les logiciels qu'il vous faut ! Mais ne faisons pas languir le lecteur, et rentrons dès maintenant dans le vif du sujet.

L'assembleur-éditeur d'Artic

Vous qui lisez ces lignes vous devez savoir ce qu'est et à quoi sert un assembleur, mais pourquoi celui-ci porte-t-il l'extension EDITEUR ? Tout simplement, parce qu'il propose pour commencer de nombreuses facilités d'édition, qui ne nous feront pas regretter ZXAS !

Et comme si cela n'était pas suffisant, notre programme inclut, en plus, un petit « moniteur » qui va permettre de procéder à un premier « débogage » de nos programmes.

C'est d'ailleurs sous ce moniteur que nous nous retrouverons dès qu'aura été exécuté un « RAND USR 3E4 » (3E4, rappelons-le, est la notation scientifique du ZX81 pour 30000). Un message de présentation apparaît alors (voir listage n° 1) : notre assembleur est opérationnel.

A ce niveau, un certain nombre de commandes peuvent être exécutées. Profitons-en pour explorer rapidement le manuel utilisateur, assez fourni, très

Listage 1

```
<<<< ZX - ASSEMBLER >>>>
```

BY D.P. AKNAI M. STREETON
(C) 1982 ARTIC COMPUTING LTD.

>

clair et entièrement en français. Un sommaire en page 2 nous invite alors à nous reporter à la page 8. Là, sont citées et explicitées clairement les commandes utilisables sous moniteur. Citons en vrac :

- **ASSEMBLE**, pour assembler nos programmes

- **COPY BLOCK**, pour recopier un bloc de mémoire d'une adresse à l'autre : il suffira de donner comme paramètres hexadécimaux l'adresse d'origine, l'adresse de fin et l'adresse de destination.

- **MEMORY EDIT**, qui nous permettra de voir et/ou modifier une zone mémoire directement en hexadécimal : la petite flèche que vous pouvez voir sur le listage n° 2 de ce mini « DUMP » est

Listage 2

4084	FF
4085	AE
4086	12
4087	05
4088	15
4089	EE
408A	D3
408B	E5 < 00
408C	13
408B	03
408C	BC
408D	AB
408E	10
408F	13
4090	2E

l'adresse courante que l'on pourra à volonté modifier ou sauter par un simple « NEWLINE ». Ce petit « éditeur de mémoire » — qui n'est pourtant qu'une partie de notre éditeur-assembleur — est en lui même une merveille : outre la possibilité de se déplacer à volonté sur toute

la mémoire, il peut répéter automatiquement l'entrée d'une valeur et calculer les sauts relatifs !

- **I**, pour inspecter et modifier les principaux registres du ZX80 (BC, DE, HL, A et le registre d'indicateurs). Très utile, cette commande permet soit d'initialiser les registres avant d'exécuter une routine, soit de les inspecter après exécution d'une routine.

- **SEARCH FOR**, pour effectuer une recherche de deux octets à partir d'une adresse donnée, et qui nous affichera soit « NOT FOUND » (pas trouvé), soit l'adresse en hexadécimal à laquelle la première occurrence aura été rencontrée.

- **RUN**, enfin, qui exécute une routine en langage machine à partir d'une adresse donnée, sans avoir besoin de quitter notre assembleur-éditeur.

Mais, malgré cette liste déjà impressionnante, j'ai gardé le meilleur pour la fin : appuyons sur la touche E, nous entrons alors sur l'éditeur qui va nous permettre d'écrire notre source en mnémoniques Z80.

L'Éditeur

C'est un éditeur du type « pleine page », c'est-à-dire que nous allons pouvoir déplacer un curseur (ici un carré noir) dans les quatre directions de notre

Listage 3

```
;EXEMPLE DE PROGRAMME
;REALISE AVEC L'ASSEMBLEUR D'ARTIC
;-----
```

```
DEBUT XOR A      ;A=0
      LD B,63    ;B=63
LOOP  RST 10     ;ECRIT A
      INC A      ;A=A+1
      DJNZ LOOP  ;BOUCLE
      CALL PRINT
      "CECI EST UN MESSAGE"FF
      RET        ;RETOUR BASIC
```

PRINT=73FA

```
;TOUT LES TEXTES OU DC-
;TETS SUIVANTS SERONT IN-
;CLUS DANS LE CODE OBJET
```

```
AE 01 FF 45 6A
"VOILA UN TEXTE"
```

écran. Ce sont tout naturellement les quatre flèches (touches 5 à 8 du clavier) qui vont servir notre dessein. L'écran de travail est divisé en deux parties : un « champ étiquettes » (c'est-à-dire une « portion d'écran » qui contiendra uniquement des étiquettes symboliques) de six caractères sur la gauche, et un « champ instruction » qui comprend la partie droite restante de notre page. A ce niveau, nous pouvons alors écrire notre programme en mnémoniques Z80 tout à fait standard : les connaisseurs jugeront d'après le listage n° 3. Mais Artic ne s'est une fois de plus pas laissé aller à la facilité ! Car, de nouveau, plusieurs commandes sont disponibles à ce niveau :

- SHIFT Ø pour effacer le dernier caractère écrit
- SHIFT 9 pour insérer un caractère
- SHIFT D pour supprimer la ligne courante

- SHIFT E pour insérer une ligne
- SHIFT G pour obtenir un listage sur imprimante
- SHIFT T pour revenir au début du programme
- SHIFT S pour effectuer une recherche de chaîne de caractère dans le listing du programme ! (aussi bien pour un texte quelconque que pour une étiquette symbolique).

Ah, j'oubliais un détail. Au bout d'une seconde, n'importe quelle touche se met en « auto-repeat », ce qui m'a permis de tracer sans peine la ligne de tirets que vous voyez sur le listage n° 3 !

Mais parlons maintenant de l'Assembleur lui-même et de ses facilités.

Intelligemment conçu, cet Assembleur ne nous fera pas non plus regretter ZXAS. Voici en quelques lignes ses caractéristiques :

- Etiquettes symboliques en toutes lettres
- Valeurs immédiates hexadécimales ou décimales
- Insertion de « datas » sous forme d'hexadécimal ou directement en caractères à n'importe quel endroit du programme (voir listage)
- Commentaires (précédés d'un point-virgule)
- Accession à une adresse avec un décalage de + ou - 9 par rapport à une étiquette (ex : Ld HL, (zone + 7)).

En bref, il devient alors facile d'écrire des « sources » claires, commentées et explicites.

Notre programme source une fois écrit, SHIFT Q nous fera revenir au niveau moniteur où la commande « Assemble » va nous permettre d'assembler notre programme, exclusivement (hélas !) à l'adresse 16516, c'est-à-dire dans une REM en ligne 1 que le programme aura automatiquement créée, et dont la taille

sera celle exactement de notre code objet.

L'écran de notre ZX81 se brouille alors durant la durée de l'assemblage (très rapide). Celui-ci s'interrompt dès qu'une erreur quelconque est détectée : un message suffisamment explicite apparaît alors ; en appuyant sur la touche « E », nous revenons sous Editeur, exactement sur la ligne à laquelle aura été détectée l'erreur.

Devant une telle « débauche » de facilités, que puis-je encore vous dire d'autre, sinon que l'assembleur d'Artic occupe 7 Ko de MEV, qu'il se loge au-dessus de RAMTOP, et qu'il gère lui-même la place mémoire occupée par les programmes sources et objets ! Exactement comme le programme exécutable, la source est placée dans une instruction REM qui porte le numéro de ligne 2. Il nous suffira, pour sauvegarder l'un ou l'autre ou les deux sur cassette, de revenir au Basic et d'effectuer un « SAVE » tout-à-fait normal, après avoir supprimé l'une ou l'autre ligne ou bien aucune des deux.

En bref...

Le pour :

- grande souplesse et facilités d'éditions
- étiquettes symboliques en toutes lettres
- « nécessaire » de débogage incorporé
- gestion transparente de la place mémoire
- manuel utilisateur clair et complet avec de nombreux exemples de programmes.

Le contre :

- une seule adresse (16516) pour la localisation du code objet
- programme non relogeable.

Mais passons maintenant au complément indispensable de notre assembleur.

Le Moniteur Désassembleur de Crystal

Crystal est un nouveau-né dans le monde des fabricants de logiciels pour Sinclair, mais pour un premier coup, c'est un coup de maître !

En effet, sur Moniteur-Désassembleur, il est fourni avec un manuel utilisateur lui aussi très complet, et est un puissant outil pour tout programmeur en langage machine.

Le programme, après chargement, va lui aussi se loger au-dessus de RAMTOP. Un RAND USR 28500 va nous faire

découvrir ses nombreuses possibilités. Apparaît alors à l'écran une page de présentation, suivie du mot « READY » et d'un curseur clignotant (voir listage n° 5). A partir de ce moment, si l'on appuie sur une des touches indiquées par le mode d'emploi, le mot complet correspondant à l'action à exécuter va s'afficher. Essayons donc la touche A : ASSIGN apparaît alors à l'écran. Cette première commande permet d'affecter une valeur à un des registres du Z80. Cette valeur pourra être rentrée directement en décimal (nombre précédé du symbole « E ») ou en hexadécimal (ce qui est très pratique : oublions là ZXDB et ses entrées à faire exclusivement en hexadécimal ! De la même manière, et pour tous les ordres qui vont suivre, tous les paramètres pourront être écrits,

Listage 5

CRYSTAL COMPUTING
00000000000000000000

MONITOR AND DISASSEMBLER

COPYRIGHT 1982 IAN STAMP

indifféremment en décimal ou en hexadécimal. Nous venons donc de voir « ASSIGN », passons aux autres commandes de notre Désassembleur.

BREAK permet de positionner un point d'arrêt (un seul, malheureusement !) à une adresse en MEV. Après exécution d'une routine contenant un « BREAK », un ordre, que nous verrons plus bas, va nous permettre d'inspecter le contenu des registres.

COPY permet de copier un bloc d'octet d'une adresse à une autre.

EDIT permet de passer en mode « édition de mémoire » et, tout comme l'assembleur, le lister et/ou modifier des octets en hexadécimal (voir listage 6)

JUMP RELATIVE CALCULATOR, extrêmement utile, nous permet de calculer l'octet de déplacement à ajouter après un « JR » ou une instruction « DJNZ » entre deux adresses. Le résultat de ce calcul pourra être rentré directement en mode « EDIT »

CONVERT va nous permettre d'effectuer des conversions hexadécimal-décimal et inversement.

Listage 6

```

752A      00
752B      = 14
752C      80
752D      CD
752E      F3
752F      74
7530      CD < ?
7531      D5
7532      70
7533      C9
7534      CD
7535      CF
7536      70
7537      76
7538      R 37
7539      A 26
753A      N 33

READY
>EXIT

```

LOAD, qui va de pair avec **SAVE** sont deux ordres de sauvegarde/lecture sur cassette à haute vitesse ! Ils vont faciliter la sauvegarde de n'importe quelle zone de MEV, mais ne sont pas compatibles avec les « **SAVE** » et « **LOAD** » classiques du Basic. Les paramètres à fournir à **SAVE** sont le nom du programme à sauvegarder, son adresse de début et son adresse de fin. Il faudra néanmoins utiliser le « **LOAD** » du désassembleur pour lire le programme sur cassette.

MESSAGE permet d'insérer une chaîne de caractère quelconque à un endroit donné de la MEV.

Listage 7

```

>REG
BC DE HL IX IY A
0000 3B3A 4171 02B1 4000 00

BC" DE" HL" SP (SP) (HL
029A 0001 4D3F 6BE2 4A72 3E
BREAKPOINT : -

FLAGS : SZ-H-PNC
        01000100

READY
>EXIT

```

PRINT affiche en caractères le contenu de 64 adresses consécutives de mémoire.

REGISTERS, une des commandes les plus importantes, affiche le contenu de tous les registres du Z80, registres « primes » y compris, le contenu bit par bit du registre d'indicateur, le contenu des adresses pointées par HL et SP et l'adresse d'un éventuel point d'arrêt ! Voilà de quoi réjouir les fouineurs... (voir listage 7).

TABULATE provoque un « **DUMP** » en hexadécimal sur 64 adresses consécutives à partir d'une adresse donnée.

VERIFY compare deux blocs de mémoire : le message « **OK** » apparaît si ces deux blocs sont identiques.

ZERO permet de mettre à zéro (00 hex) une zone de mémoire.

EXIT, enfin, permet de sortir du moniteur et de revenir au Basic.

Cette liste ne serait pas complète sans l'ordre **DESASSEMBLE**, qui permet bien évidemment de désassembler un programme à partir d'une adresse donnée ! Comme on peut le constater sur le listage n° 8, les mnémoniques Z80 sont, contrairement à ceux de ZXDB, parfaitement standard. Seule ombre au tableau, il n'est pas possible, avec le désassembleur tel qu'il est, de sortir ce listing sur imprimante. Mais qu'à cela ne tienne : une routine donnée en annexe de ce banc d'essai permettra d'y remédier !

Listage 8

```

READY
>DISASSEMBLE 000
0000 D3FD OUT (FD),A
0002 01FF7F LD BC,7FFF
0005 C3CB03 JP 03CB
0008 2A1640 LD HL,(4016)
000B 221840 LD (4018),HL
000E 1846 JR 0056
0010 A7 AND A
0011 C2F107 JP NZ,07F1
0014 C3F507 JP 07F5
0017 FF RST 3B
0018 2A1640 LD HL,(4016)
001B 7E LD A,(HL)
001C A7 AND A
001D C0 RET NZ
READY
>EXIT

```

Là aussi, avec un tel nombre de possibilités pour un programme qui n'occupe que 4,3 Ko de MEV, on ne peut qu'être comblé.

Le Désassembleur de Crystal est donc un outil parfaitement autonome, pratique et fiable. Il conviendra aussi bien à celui qui ne désire programmer qu'en hexadécimal qu'à l'heureux possesseur de l'assembleur d'ARTIC.

En bref

Le pour :

- utilisation simple et souple (entrées aussi bien en décimal qu'en hexadécimal)

- désassemblage rapide et en mnémoniques standard Z80.

- autonome

- manuel utilisateur clair et complet

Le contre :

- pas de listage de désassemblage d'origine sur imprimante

- un seul point d'arrêt à la fois.

Conclusion

Nous venons donc d'avoir un aperçu de

ce que sont ces deux programmes. Si vous n'êtes pas convaincus de leurs réelles efficacité et puissance, il ne vous reste plus qu'à les acheter, les essayer et finalement les adopter !

Pour ma part, je pense que ces deux logiciels sont un réel progrès par rapport à ZXAS et ZXDB.

Le seul problème qui apparemment semble se poser est le fait qu'ils sont implantés tous les deux dans les adresses hautes de la mémoire : on ne peut donc pas les changer et les utiliser simultanément. Néanmoins, dans le cas de ZXAS et ZXDB, qui sont eux parfaitement compatibles, il faut avouer que la place restante allouée au malheureux programmeur n'est pas énorme !

C'est pourquoi, ma conclusion sera, et vous vous en doutez, de préférer l'Assembleur d'Artic et le Désassembleur Crystal à leurs concurrents directs ZXAS et ZXDB : ils représentent un réel progrès par rapport à leurs aînés. Aussi bien au débutant qu'au programmeur chevronné, je ne saurais que conseiller l'achat de ces deux logiciels réellement professionnels.

E. Viau

Comment modifier le désassembleur

Obtention du listage de désassemblage sur imprimante :

1. A l'adresse 32170, remplacer les instructions d'origine par les suivantes :

```

7DAA CD576F CALL 6F57
7DAD 00 NOP
7DAE 00 NOP
7DAF 00 NOP

```

2. A l'adresse 28503, qui, d'origine ne contient que des NOP (00 Hex), planter la routine suivante :

```

6F57 ED4B2540 LD BC,(4025)
6F5B 0C INC C
6F5C 28F9 JR Z,6F57
6F5E CD4B0F CALL 0F4B
6F61 7E LD A,(HL)
6F62 FE3F CP 3F
6F64 C0 RET NZ
6F65 160E LD D,0E
6F67 CD6B0B CALL 0B6B
6F6A C9 RET

```

Utilisation de la routine

Lancer le désassemblage à l'adresse désirée pour enrayer les 14 premières lignes de l'écran sur l'imprimante, appuyer sur la touche « **COPY** ». Pour continuer le désassemblage sans imprimer, appuyer sur une autre touche.

RAMTOP

Qui ne s'est jamais trouvé, au milieu d'une construction de programme avec un RAMTOP mal placé ? et en conséquence, l'absence ou l'insuffisance d'une zone de travail ou de stockage à l'abri des NEW et des déplacements intempestifs ? (voir page 168 du manuel Sinclair).

Avec ces 72 octets de code machine dans une REM et quelques lignes de Basic, vous pourrez remédier à vos oublis ou parer à l'inattendu en déplaçant à volonté RAMTOP par programme (sans NEW et sans POKE aux adresses 16388 et 16389).

Le programme correspondant fonctionne aussi bien avec la RAM Sinclair de 1 K qu'avec une extension 16 K ou avec une RAM 64 K, qu'il s'agisse de déplacer RAMTOP vers le haut ou vers le bas de la mémoire. Dans ce dernier cas (vers le bas), RAMTOP ne sera déplacé effectivement que si le changement vous laisse au moins 128 octets de réserve en sus de ce qu'exige le système d'exploitation Sinclair :

Précautions à prendre

Ne pas essayer de loger RAMTOP au-delà du « premier octet non-existant de la RAM », c'est-à-dire au-delà de 17408 pour 1 K ou de 32768 pour 16 K, etc... Avec une RAM de 64 K, vous pouvez utiliser évidemment 65535, mais aussi 0 (zéro représentant 65536).

Vers le bas, ne pas essayer de déplacements de moins de 6 octets (six)... Si vous vouliez déplacer RAMTOP d'un seul octet vers le bas (!), faites-le en deux fois (un premier déplacement vers le bas, par exemple de 101 octets, et un autre vers le haut de 100 octets dans ce cas).

Attention aux accidents si vous donnez une adresse-mémoire très basse et voisine de zéro. Prenez par exemple pour règle de ne jamais donner à ce programme d'adresse inférieure à 9999 —sauf 0 avec la mémoire de 64 K qui correspond à un mouvement vers le haut—.

Possibilités d'utilisation

Fonctionnement sans problème à l'intérieur des sous-programmes Basic, même récursifs, puisque ce programme prend soin de la pile des adresses de retour GOSUB aussi bien que de la pile machine (voir schéma page 171 du manuel).

Mode d'emploi

Tel qu'il est proposé, avec 72 octets de code dans une REM constituant la première des six lignes du programme Basic, il vous suffit de faire RUN ou GOTO 10 et de répondre à la question « VAL RAMTOP ? » par l'adresse désirée suivie de NEXLINE.

Dans tous les cas, l'écran affiche votre demande et la position de RAMTOP après l'exécution du programme : cette position est soit celle de votre demande si elle était raisonnable, soit la position précédente inchangée s'il n'y avait pas assez de place pour le mouvement vers le bas souhaité ; dans ce dernier cas, essayez un déplacement plus réduit si vous pouvez vous en satisfaire.

Corollaire : Si vous ne vous souvenez plus de l'adresse actuelle de RAMTOP, faites RUN (ou GOTO) et répondez systématiquement 9999 puisque c'est toujours un déplacement impossible : le programme affiche naturellement l'adresse de RAMTOP. C'est plus rapide et plus commode que le calcul avec les PEEK 16388 et PEEK 16389.

Pense-bête : Si vous voulez, avec une extension RAM 16 K ou plus, vérifier que RAMTOP peut être placé par programme très au-dessous de 17408 (1K), n'oubliez pas de l'amener dans un premier temps plus bas seulement que 19712 (3,25 K) pour obtenir un affichage compacté, —sinon l'écran prend déjà à lui seul 793 octets de la RAM.

Remplacement en mémoire-programme : bien entendu, en dehors du code machine de la ligne 2 qui doit

obligatoirement être au début de la MP à l'endroit habituel (sauf si vous connaissez un moyen de détecter automatiquement son adresse lorsqu'il est ailleurs et de modifier l'adresse USR en conséquence), les 5 autres lignes de Basic peuvent être absolument n'importe où, y compris à partir de la ligne 9995.

Principe de fonctionnement

Le Basic : L'adresse que vous fournissez à la ligne 15 en réponse à la question de la ligne 10 est placée, aux lignes 20 et 25, dans les deux octets libres aux adresses 16507 et 16508 (voir page 179 du manuel).

A la ligne 30, USR 16514 fait exécuter le code machine de la ligne 2, et le résultat, quel qu'il soit, est affiché par le PRINT.

Le programme en code : Son principe est le déplacement simultané de RAMTOP et de l'ensemble des deux piles de stockage utilisées par le système, l'une pour les adresses retour de GOSUB, l'autre pour le travail entre les registres internes (et entre eux et la mémoire).

BOUGER RAMTOP

Le programme va prendre l'adresse rangée par le Basic et détecte s'il s'agit d'un mouvement vers le haut ou vers le bas. Si le résultat doit être un déplacement nul, il saute à la fin pour une simple lecture de l'adresse RAMTOP actuelle ; sinon le programme emprunte l'une des deux branches OPTION HAUT et OPTION BAS :

OPTION HAUT

Transfert immédiat des deux piles sans vérification.

OPTION BAS

D'abord, test de la réserve (voir manuel, page 171) qui subsisterait après le déplacement. Ensuite transfert des deux piles seulement au cas où la réserve est considérée comme suffisante pour travailler sans problème ; dans le cas contraire, pas de transfert et saut à la fin pour simple lecture du RAMTOP actuel.

M.A.J. SYS

S'il y a eu mouvement et dans ce cas seulement, les variables système concernées sont mises à jour dans ce tronc commun, ainsi que le pointeur de pile machine « sp ».

LECTURE RT

Dans tous les cas, le programme lit le contenu de la variable système (RAMTOP) et renvoie au Basic l'adresse correspondante.

(C) JCF 1982. ~~CE REMPLACEMENT DE RAMTOP~~
~~PAR PROGRAMME~~ (C) JCF 1982

```

2 REM E,RND GOSUB ???TF FOR E
?RND? GOSUB ?S?CI; FOR GOSUB S/
=T FAST EORND);SGN GOSUB ?K,5
T GOSUB ? FOR GOSUB ???T RAND E
RAND;6 ?RND?RND6,RND GOSUB ?RND
TAN
10 PRINT "VAL RAMTOP?",
15 INPUT A
20 POKE 16508,INT (A/256)
25 POKE 16507,A-INT (A/256)*25
6
30 PRINT A;"?VAL ";USR 16514

```

RESULTATS

```

VAL RAMTOP? 9999?VAL 32768
VAL RAMTOP? 19000?VAL 19000
VAL RAMTOP? 17000?VAL 17000
VAL RAMTOP? 28000?VAL 28000
VAL RAMTOP? 65535?VAL 65535
VAL RAMTOP? 50000?VAL 50000
VAL RAMTOP? 0?VAL 0
VAL RAMTOP? 18000?VAL 18000
VAL RAMTOP? 16700?VAL 18000
VAL RAMTOP? 16950?VAL 16950
VAL RAMTOP? 9999?VAL 16950
VAL RAMTOP? 32768?VAL 32768

```

99 REM CI-DESSOUS ~~PROGRAMME AS~~
~~SEMBLEUR~~ COMMENTE:

```

100 REM (*BOUGER RAMTOP
105 REM LD HL.(16388);SBC HL,SP
;LD B,H;LD C,L;* OLD RT-SP=PILES
110 REM ADD HL,SP;DEC HL;EX DE.
HL;* OLD RT-1
115 REM LD HL.(16507);DEC HL;AN
D A;SBC HL,DE;* DIFF=NEW-OLD
120 REM JR C,L1;JR Z,L3;* C:RT
DESCEND,Z:RT RESTE
123 REM (*OPENING
125 REM ADD HL,DE;EX DE,HL;LDDR
;JR L2;* MONTEE;L2:M.A.J. SYS
128 REM (*OPENING
130 REM :L1 ADD HL,SP;PUSH HL;*
SP=ABS (DIFF)
135 REM LD HL.(16412);LD DE,128
;ADD HL,DE;POP DE;* TEST RESERVE
140 REM SBC HL,DE;JR NC,L3;* NC
:RT RESTE
145 REM LD HL,0;ADD HL,SP;LDIR;
* DESCENTE
148 REM (*OPENING
150 REM :L2 EX DE,HL;SBC HL,DE;
LD D,H;LD E,L;ADD HL,SP;LD SP,HL
;* NEW SP
155 REM LD HL.(16386);ADD HL,DE
;LD (16386),HL;* NEW ERR.SP
160 REM LD HL.(16507);LD (16386
),HL;* NEW RT
163 REM (*OPENING
165 REM :L3 LD BC.(16388);RET
170 REM )
175 REM CI-DESSOUS CODE HEXA:
2A0440.ED72.44.4D.39.2B.EB.2A7B4
0.2B.A7.ED52.3808.282E.19.EB.ED8
8.1814.39.E5.2A1C40.118000.19.D1
.ED52.301A.210000.39.ED80.EB.ED5
2.54.5D.39.F9.2A0240.19.220240.2
A7B40.220440.ED480440.C9

```

La liste des instructions en langage assembleur est donnée aux lignes 100 à 170 (dans la forme acceptée par le ZXAS, tout ce qui est précédé d'un astérisque étant considéré comme simple commentaire).

La liste des 72 octets du code machine résultant est donnée à la ligne 175 en code hexadécimal. Si vous le faites précéder directement d'un autre programme en code de votre cru, il faut intercaler un octet d'effet "and a" (A7h).

Obtention de la Ligne 2 : Après avoir réservé 72 octets à la suite de REM dans la ligne 2, utilisez par exemple le chargeur de code hexa du n° 1 d'« ECHOS SINCLAIR » (page 12) en mettant dans A\$ le code hexa de la ligne 175.

Pour les utilisateurs de la RAM de 1 K

Avec ses 78 octets, la ligne 2 seule est essentielle.

Les lignes 10 à 30 proposées occupent 125 octets de mémoire-programme.

Il peut vous arriver d'être dans une situation où chaque octet compte. S'il en est ainsi, il vous reste des masses de ressources :

1. Vous pouvez raccourcir ou même supprimer les lignes 10 et 30 : dans ce dernier cas, faire RUN ou GOTO..., donner l'adresse et, après NEWLINE, exécuter au clavier PRINT USR 16514.

2. Vous pouvez aussi gagner de la place sur les lignes 20 et 25, notamment en utilisant quatre fois la fonction VAL suivie de guillemets.

3. Enfin en désespoir de cause, ne gardez que la ligne 2, entrez l'adresse au clavier par LET A = ..., et faites les deux POKE au clavier avant le print usr... que vous pouvez même faire précéder de CLEAR si vous n'avez pas de variables à protéger.

4. Et lorsque vous n'avez plus besoin de l'espace au-dessus de RAMTOP, donnez-vous de l'air en remplaçant RAMTOP au bout de la mémoire. Quand ce ne serait que pour faciliter l'affichage.

Jean-Claude Fischhof

La recherche en table

N'avez-vous jamais eu à vérifier si une donnée, un texte, un chiffre et autres variables étaient catalogués dans une liste ?

Si c'est le cas, vous avez sûrement essayé plusieurs méthodes, et vous avez sûrement noté que cette recherche d'existence était longue, très longue.

C'est pourquoi, je vous propose d'étudier avec moi une routine non seulement capable de vérifier si un mot existe dans une table, mais aussi de localiser deux mots et de fournir le ou les codes de ces mots pour une éventuelle utilisation. Faisons un plan.

Tout d'abord — les objectifs : après l'impression d'une phrase ou d'un mot sur l'écran et après appel de la routine il sera possible d'avoir les codes des mots s'ils existent.

Ensuite — les contraintes : nous nous contenterons d'une ligne de 32 caractères de long au maximum avec des mots d'une longueur totale de 15 au maximum. On écrira la phrase ligne 21 à partir de la colonne 8.

La table

Pour pouvoir trouver les éventuels codes, il faudra au préalable, créer une table de référence, contenant les mots ou tout au moins leur début, c'est-à-dire suffisamment de caractères pour pouvoir les localiser, plus le code qu'on leur attribuera.

Pour aller plus vite dans la recherche, je vous propose la structure suivante : sur 1 octet longueur du mot (entre 1 et 15)

sur n (entre 1 et 15) le mot (avec son code sur 1 octet).

Ex. : 47X811

Il est à noter que les longueurs et les codes seront notés sous forme de code caractère.

Ex : 4 → 13 → \$
7 → 38 → A voir livre Sinclair

Ceci permettra de faire un dictionnaire d'au moins 255 mots différents ou ayant 255 codes différents.

Pourquoi et comment

A. Remise à zéro

Pourquoi La routine étant destinée à servir plusieurs fois dans le même programme, il sera

souhaitable si des mots recherchés n'existent pas, que les résultats ne soient pas ceux du tour d'avant.

Comment En mettant simplement des zéros aux adresses contenant les variables de la routine.

B. Trouver le début de table

Pourquoi Pour que la routine recherche l'existence depuis le premier mot et non pas là où elle s'était arrêtée précédemment.

Comment En recalculant l'adresse de début de table.

C. Trouver le début du 1^{er} mot

Évident mon cher Watson !

Comment En se positionnant à la première colonne ligne 22.

D. Le 1^{er} mot existe-t-il ?

Pourquoi A question b...

Comment C'est le but du programme.

E. Trouver le second mot

Comment En commençant par la fin de la 22^e ligne.

F. Ramener les bons codes

Plusieurs cas peuvent se présenter :

— code différent de code 2 normal deux mots.

— code 1 égal code 2 normal un mot

— code 1 égal zéro et code 2 différent de zéro, on mettra code 2 dans 1 et zéro dans 2.

— code 1 différent de zéro et code 2 égal zéro : rien ne change.

Tous les caractères soulignés doivent apparaître en vidéo inversée.

```

0 REM /. * ? : ; 5 6 ? RNDLN I RNDLN
COS RNDLN PEEK RNDM ? RNDLN CLEA
R RNDLN PEEK RNDM ? RNDU ? RND RETUR
N C ? RND ? 1 " " / . U ? RNDM ? RND J M ? RND
TAN ED RND ) : 6 ? RND TAN ED ) : 6 ? RND
DTAN E ? RND FAST GOSUB ? * RND 7 < . 1
4 * ( SAVE AT 700 : TAN INT = ED ) STR #
* : 7 4 RUN FOR < GOSUB ? * RND TAN
TAN TAN * RND TAN ZZ
2 REM (
3 REM * RECHERCHE EN TABLE *
4 REM * (C) ECHOS SINCLAIR
ET JM COHEN
5 REM JR L1
6 REM : L20 NOP : NOP * L20 = 16516
8 REM : L21 NOP : NOP * L21 = 16518
9 REM : L1 LD HL 0
10 REM LD (16507) . HL
15 REM CALL L5 * HL DATA
20 REM CALL L7 * DE 1ER
30 REM CALL L6 * ROUTINE
40 REM LD (16507) . A
60 REM CALL L8 * DE 2EME
70 REM CALL L6 * ROUTINE
75 REM LD (16508) . A
76 REM LD A. (16507)
77 REM CP 0
78 REM JR Z . L9
79 REM LD HL. (16507)
80 REM LD A . H
81 REM SBC A . L
82 REM RET NZ
83 REM JR L13
91 REM : L9 LD A. (16508)
92 REM LD (16507) . A
93 REM : L13 XOR A
94 REM LD (16508) . A
95 REM RET

```

```

100 REM * HL DATA
110 REM : L5 LD HL. (16425)
120 REM LD DE . 5
130 REM ADD HL . DE
140 REM LD (L20) . HL
150 REM RET
160 REM * DE 1ER
170 REM : L7 LD HL. (16396)
180 REM LD DE . 693
190 REM ADD HL . DE
200 REM LD (L21) . HL
210 REM RET
220 REM * ROUTINE
230 REM : L6 LD HL. (L20)
240 REM : L2 PUSH HL
250 REM LD DE. (L21)
260 REM LD A. (HL)
270 REM CP 15 * CP LONGMAX
280 REM JR NC . L10
290 REM LD B . A
300 REM : L4 INC HL
310 REM INC DE
330 REM LD A. (DE)
340 REM CP (HL)
350 REM JR NZ . L3
360 REM DJNZ . L4
370 REM POP BC
380 REM INC HL
385 REM LD A. (HL)
430 REM RET
440 REM : L3 POP HL
450 REM LD E. (HL)
460 REM LD D . 0
470 REM INC E
480 REM INC E
490 REM ADD HL . DE
500 REM LD A. (HL)
510 REM CP 15
520 REM JR C . L2
530 REM XOR A
550 REM RET
560 REM : L10 RST 0B
570 REM INC D

```

Choix de l'emplacement de la table

Vous noterez page 179 (en haut de page) qu'il existe (oui, je l'ai rencontré) une variable système nommée NEXTLINE qui contient l'adresse du début de la ligne suivante, cette variable est stockée en 16425-16426. Sachant que le numéro de ligne est sur 2 octets, que la longueur est sur 2 octets que REM prend 1 octet. Nous voyons qu'en additionnant 5 à NEXT LINE nous obtenons l'adresse du 1^{er} caractère du REM de la ligne suivante.

Ex :

```
LET A = PEEK 16425 + 256 * PEEK 16426 + 5
REM LE ZX 81 TRAVAILLE EN BASIC
FOR I = A TO A + 27
PRINT CHR$(PEEK(I))
NEXT I
```

Ce petit programme vous imprimera lettre par lettre le contenu (27 lettres) du REM.

Nous pratiquerons la même méthode pour notre routine Assembleur.

Les avantages sont considérables :

- pas d'effacement si RUN
- on peut mettre la table n'importe où dans le programme.

Choix de l'emplacement de la routine

Cette routine fait 154 octets de longueur, et il sera facile de la loger dans un REM à la ligne 1 du programme.

1 REM suivi de 160 caractères quelconques.

Notez cependant qu'il est conseillé de ne pas faire un « éditer » de la ligne 1 car il pourra survenir quelques problèmes dus à certains des codes utilisés. Pour éviter ceci faite POKE 16510, 0, ce qui renverra la ligne 1 en ligne 0 qu'il n'est pas possible d'éditer.

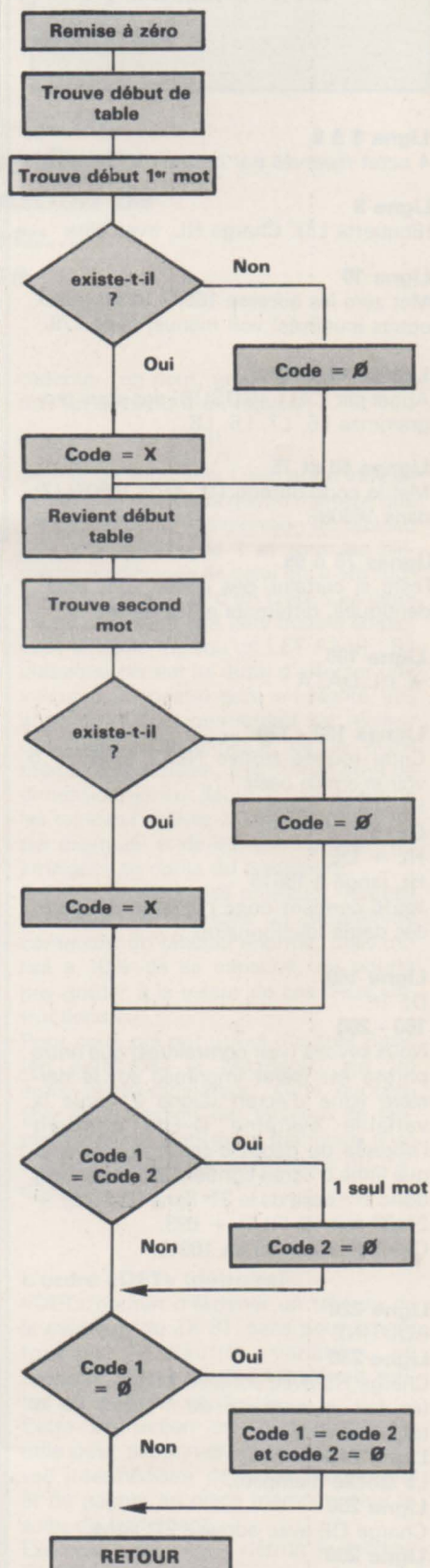
Le programme assembleur ligne par ligne

Ligne 5

Soit à l'étiquette 1 soit début de programme.

Le programme

Organigramme de principe



```

590 REM *DE 2EME
600 REM :LBD HL. (16396)
610 REM LD DE.725
620 REM ADD HL.DE
630 REM :L1LD A. (HL)
635 REM DEC HL
640 REM CP 0
650 REM JR Z.L11
655 REM INC HL
660 REM :L12LD A. (HL)
665 REM CP 118
666 REM RET Z
670 REM DEC HL
680 REM CP 0
690 REM JR NZ.L12
700 REM EX DE.HL
705 REM INC DE
710 REM LD (L21).DE
715 REM RET
720 REM )
1000 INPUT Y$
1005 CLS
1010 PRINT AT 21,0;Y$;
1020 LET L=USR 16514
1030 REM .NORD*EST=SUD*OUEST.
. HAUT*BAS*PREND*POSE*.VASE? *O
R (PORTE)*OUVER* *LAMPE*ALLUME*
REPLI*HUILE*INVENTAIRE*QUITT
E/*DECRIS;
1040 LET A=PEEK 16507
1050 LET B=PEEK 16508
1060 IF A=B THEN GOTO 1100
1070 PRINT " OK " ;A;" " ;B
1080 GOTO 1000
1100 PRINT " PAS TROUVE "
1110 GOTO 1000
1900 REM -----
2000 REM CHARGEUR HEXA
2005 REM -----
2010 LET A$="CODE HEXADECIMAUX"
2020 FOR X=16514 TO 16514+160
2030 POKE X,16*CODE A$+CODE A$(2
)-476

```

```

2040 NEXT X
3000 REM -----
3010 REM CHARGEUR DE TABLE
3020 REM -----
3030 LET A$=""
3035 LET C=0
3040 INPUT Y$
3050 LET C=C+1
3060 LET A$=A$+CHR$(LEN Y$+Y$+CH
R$ C
3070 PRINT AT 0,0;A$
3080 GOTO 3040
3085 REM
3090 REM ENTRER LES MOTS LES
PLUS COURTS D'ABORD.
3100 REM RESERVER DANS UN REM LA
PLACE SUFFISANTE.
PUIS FAITE :
3105 LET A=PEEK 16425+256*PEEK 1
6426+5
3110 REM ...PLACE POUR TABLE....
.....ETC
3115 FOR I=A TO A+LEN A$
3120 POKE I,CODE A$(I-(A-1))
3130 NEXT I
3140 REM
PLACER ENSUITE LE REM QUI
CONTIENT LA TABLE OU VOUS
LE DESIREZ
8000 STOP
9000 FAST
9010 INPUT ZZZ
9020 POKE 32641,INT (ZZZ/256)
9030 POKE 32640,ZZZ-256*INT (ZZZ
/256)
9040 RAND USR 28565
9050 PRINT AT 21,0;"ERROR ";PEEK
32651
9060 STOP
9070 STOP
9080 SAVE "DL"
9090 RUN

```

La recherche en table

Ligne 6 à 8

4 octet réservés par la routine (variable)

Ligne 9

(Étiquette L1). Charge HL, avec zéro

Ligne 10

Met zéro les adresse 16507-16508 (deux octets inutilisés) voir manuel page 179.

Lignes 15 à 70

Appel par CALL (GOSUB) des sous programmes L5, L7, L6, L8

Lignes 40 et 75

Met le code obtenu (1^{er} dans 16507) (2^e dans 16508).

Lignes 76 à 95

Teste si certains des codes sont zéro, identiques, différents et les traite.

Ligne 100

★ HL DATA

Lignes 100 - 120

Cette routine trouve NEXT LINE + 5 voir exemple basic.

HL. (Next Line).

DE. 5

HL + DE

HL rangé à 16516

16516 contient donc l'adresse de début des datas (dictionnaire)

Ligne 160

DE 1^{er}

160 - 200

Nous savons (voir contraintes) que notre phrase est (sera) imprimée sur la dernière ligne d'écran (Ligne 21) que la variable système D-File contient l'adresse du début d'écran et que chaque ligne d'écran contient 33 caractères, donc 1^{ère} case de la 21^e ligne (D-FILE) + 21x33 soit (D-FILE) + 693.

Que l'on stockera en 16518.

Ligne 220

ROUTINE

Ligne 230

Charge HL avec adresse 1 lettre 1^{er} mot (en fait la longueur du 1^{er} mot de la table).

Ligne 240

La stocke (l'empile).

Ligne 250

Charge DE avec adresse 21^e ligne.

Ligne 260

Met dans A le contenu de (HL).

inversion-vidéo.

```
16514 18 04 01 47 97 4E 21 00
16522 00 22 7B 40 CD BD 40 CD
16530 CB 40 CD D3 40 32 7B 40
16538 CD FD 40 CD D3 40 32 7C
16546 40 3A 7B 40 FE 00 28 08
16554 2A 7B 40 7C 9D C0 18 06
16562 3A 7C 40 32 7B 40 AF 32
16570 7C 40 C9 2A 29 40 11 05
16578 00 19 22 84 40 C9 2A 0C
16586 40 11 B5 02 19 22 86 40
16594 C9 2A 84 40 E5 ED 5B 86
16602 40 7E FE 0F 30 1B 47 23
16610 13 1A BE 20 06 10 FB C1
16618 23 7E C9 E1 5E 16 00 1C
16626 1C 19 7E FE 0F 38 DD AF
16634 C9 CF 14 2A 0C 40 11 D5
16642 02 19 7E 2B FE 00 28 FA
16650 23 7E FE 76 C8 2B FE 00
16658 20 F7 EB 13 ED 53 86 40
16666 C9 C9 C9 86 40 C9
```

Ligne 270

Le compare à 15, c'est-à-dire la longueur maximum admise ici (mais vous pourrez aller jusqu'à 32).

Ligne 280

Si cette longueur (volontaire ou involontaire) supérieure à 15 va à erreur «L»

Ligne 290

Met A dans B

Ligne 300

Pointe sur 1^{er} lettre de la table

Ligne 370 - 400

Si pour la longueur précisée le mot correspond,

On met le code qui suit dans A et retour de sous programme.

Étiquette 3

Ligne 440

On récupère adresse mot précédant (voir ligne 240) et la longueur dans E.

Ligne 490

On additionne la longueur

Ligne 510

On teste la fin de fichier le NEWLINE de fin de ligne.

Ligne 520

Si pas fin de fichier on continue en L2.

Ligne 590

DE 2^e

Ligne 590 - 715

Dans tous les cas on passe au deuxième mot, on calcule l'adresse de fin de 21^e ligne puis on remonte tous les blancs, puis toutes les lettres jusqu'à un nouveau blanc ou un NEWLINE (soit début du dernier mot), et on reprend la suite c'est-à-dire la routine de recherche.

Lignes 560 - 570

La routine d'erreur en L10 (Lignes 560 à 570)

RST 08 est utilisé dans la ROM Sinclair pour différentes actions dont la gestion des erreurs. (INC D n'est pas ici, une instruction mais correspond à erreur «L»). Si vous obtenez une erreur «L» ligne xxx, vérifiez votre table. L'erreur montre une erreur de longueur (15) de l'un des mots ; ou vous étant trompé sur l'une d'entre elles, l'addition de longueur pour passer au mot suivant est fausse.

Le programme BASIC

1000 INPUT Y\$ on entre la phrase.

1010 CLS

1020 PRINT AT 21,0;Y\$; le dernier point virgule est important.

1030 LET L =USR 16514 Appel routine recherche

1040 REM ☐ NORD ☐ REM de DATAS

1050 LET A = PEEK 16507 récupère code 1^{er} mot.

1060 LET B = PEEK 16508 récupère code 2^e mot.

1070 IF A = B THEN GOTO 1100 Seul cas A = B = 0 car test dans routine

1080 PRINT "OK";A;" ";B OK et codes

1090 GOTO 1000

1110 PRINT «PAS TROUVE»

1120 GOTO 1000

Ligne 310

Pointe sur 1^{er} lettre de la 21^e ligne.

Ligne 330

Charge A avec lettre ligne.

Ligne 340

Compare avec lettre table

Ligne 350

Si différent va à L3 (étiquette 3 voir listing)

Ligne 360

Sinon passe à seconde lettre

Conclusion

Cette routine pourra être d'un grand intérêt pour des fichiers de type téléphonique, des jeux etc...

Ex : L, NOM, Code, puis avec le(s) code(s) récupéré(s) écrire une variable (tableau).

PRINT A\$(A);A\$(B) avec A et B codes des mots.

A vos claviers.

J.M. Cohen

ZXTRI

**La réponse à tous vos problèmes de tri,
recherche et gestion de tableaux en Basic.
(ou : un index téléphonique, une gestion de
stock, de compte en banque, etc...
à l'aide d'un seul programme
et d'un peu de réflexion !)**

Sous le nom «barbare» de ZXTRI se cache un puissant utilitaire : ce logiciel, qui se veut universel, permet de pallier aux problèmes liés en général à la gestion de tableaux. Il permet, en particulier, de trier et d'effectuer des recherches d'éléments dans des tableaux numériques ou alphanumériques avec une puissance et une rapidité inconnues jusqu'ici du Basic Sinclair.

ZXTRI se présente bien entendu sous la forme d'une cassette pré-enregistrée, accompagnée d'une notice explicative relativement fournie. Sur la cassette sont enregistrés 5 programmes : un programme exemple, complément du manuel utilisateur et quatre versions différentes de ZXTRI. En effet, et contrairement à d'autres programmes commercialisés pour ZX 81, ZXTRI se veut compatible avec toutes les configurations existantes : ces quatre versions correspondant au même logiciel, mais relogé à des adresses différentes. De cette manière, l'acquéreur pourra utiliser sa capacité mémoire au maximum de ses possibilités, qu'il soit possesseur d'une 16 K ou d'une 64 K.

Mais avant de rentrer dans le détail de la description du programme, voyons tout d'abord l'idée principale qui a présidé à sa conception. En effet, si l'on fait un tour d'horizon de tous les utilitaires de «gestion de fichier» existants, on se rend compte rapidement que ces programmes sont chacun dans leur domaine beaucoup trop spécialisés. Celui-ci, par exemple, propose une gestion de compte en banque, cet autre une gestion de stock avec toutes les contraintes que cela suppose : actions, taille des zones du fichier et présentation imposée, etc...

On se rend compte, en général, à l'utilisation de ces logiciels, qui se veulent universels, qu'ils font effectivement toutes les actions possibles, sauf, bien

évidemment, celles que désirerait spécifiquement l'utilisateur !

C'est donc pour répondre à ces besoins spécifiques que ZXTRI existe : il propose une base autour de laquelle l'utilisateur construira son propre programme, entièrement en Basic, et répondant bien évidemment exactement à ses besoins.

Le programme

Entièrement en langage machine, il occupe exactement 1926 octets de RAM. La version que nous utilisons pour le banc d'essai est placée dans une instruction REM à partir de l'adresse 16514. Entièrement transparent pour l'utilisateur, qui n'a pas besoin de connaître le langage machine, ZXTRI s'utilise à partir du Basic à l'aide d'ordres placés dans des instructions REM, immédiatement précédées d'un RAND USR 16516.

Ces ordres sont au nombre de cinq, et fonctionnent aussi bien pour les tableaux numériques qu'alphanumériques.

L'ordre «INS» (insérer)

Cette instruction permet d'insérer un élément dans un tableau déclaré auparavant par un «DIM» normal du Basic. La syntaxe en est des plus simples, exemple :

INS A\$ (1) : B\$.

L'ayant auparavant été initialisé avec la position de l'élément à insérer et B\$ avec la variable à insérer dans le tableau, après exécution, la dimension principale du tableau A\$ aura été incrementée de 1 et tous les éléments en-dessous de celui d'indice 1 auront été «décalés» vers le «bas».

L'ordre «sup» (supprimer)

Cette instruction est l'inverse de la pré-

cedente : on peut, grâce à elle, supprimer un élément d'un tableau,

exemple : sup A\$ (1)

permet de supprimer l'élément d'indice 1 du tableau A\$. De la même manière que précédemment, la dimension du tableau est décrementée de 1 et tous les éléments en dessous de celui qui a été supprimé «remontent» d'un cran. Ces deux ordres, qui semblent faire double emploi avec le Basic (puisque un LET A\$ (1) : B\$ classique permet lui aussi d'affecter une valeur à un poste) sont en réalité très importants : ils permettent de réaliser une gestion dynamique de la place mémoire du Sinclair. Il suffira en effet de dimensionner au début du programme les tableaux utilisés à 1 (DIM A\$ (1, 10) par exemple) et de les faire «grossir» ou «maigrir» au cours du traitement.

Qui n'a jamais passé plusieurs minutes à charger en mémoire un programme contenant un tableau énorme, mais utilisé à 10% de sa capacité, ne pourra pas goûter à la valeur de ces deux instructions !

Pour ceux qui ont utilisé d'autres Basic (en particulier le Microsoft) sur un autre micro-ordinateur, cette conception de la gestion des tableaux se rapproche assez bien de la définition d'un fichier «RANDOM» sur disque ou disquette.

L'ordre «DET» (détruire)

«DET» permet d'éliminer un tableau de la mémoire du ZX 81, sans pour autant toucher à d'autres variables ou tableaux, là où un CLEAR détruit toutes les variables existantes.

Cette instruction est particulièrement utile pour supprimer un tableau de travail intermédiaire dans un programme et de gagner en place mémoire pour la suite du traitement.

Exemple : DET C\$: détruit le tableau C\$ (1).

ZXTRI

Mais passons maintenant aux deux ordres les plus puissants de ZXTRI : RECH et TRI.

RECH permet d'effectuer une recherche d'un élément dans un tableau à n dimensions, numérique ou alphanumérique, et ce, à la vitesse du langage machine !

Pour l'alphanumérique, les paramètres à donner à l'instruction sont les bornes entre lesquelles va s'effectuer la recherche, et les variables numériques dans lesquelles ZXTRI doit placer les indices lorsque l'élément à rechercher aura été trouvé.

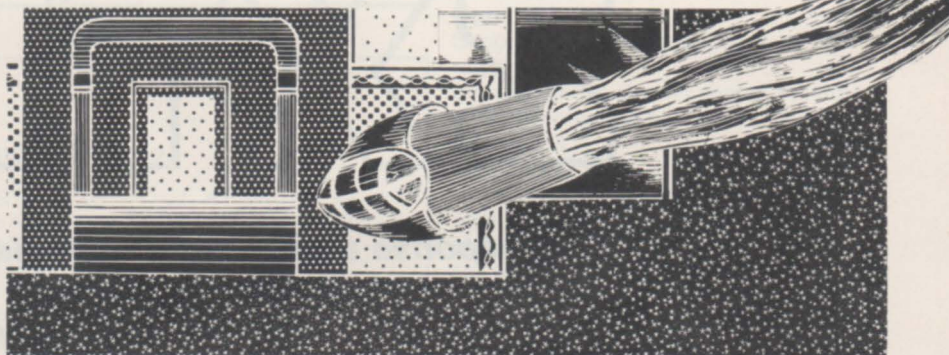
Exemple :

RECH G\$ (10, 3, 10) = C\$: I, J

G\$ est un tableau, qui aura auparavant été déclaré par un DIM G\$ (10, 3, 10) par exemple. Chacune des autres variables utilisées (A, I, J) aura dû être initialisée (LET A = 1 par exemple) et C\$ contiendra l'élément à rechercher. Au terme de cette recherche, si la chaîne contenue dans C\$ est présente dans le tableau G\$, les indices correspondants auront été placés par ZXTRI dans les variables I et J, G\$ (I, J) pointera alors sur l'élément recherché. Comme on peut le constater dans cet exemple, ZXTRI est conçu de manière à se rapprocher le plus possible du Basic : la syntaxe G\$ (A TO 10), qui permet de définir une sous-chaîne à l'intérieur d'une chaîne principale, est tout à fait classique. De plus, si l'on réexécute la recherche de la même manière une seconde fois, ZX SORT ne reverra pas dans I et J les mêmes valeurs si l'élément recherché est présent plusieurs fois dans le tableau : il aura, pour la première fois, donné les indices I et J correspondant à la première occurrence, la seconde fois, ceux correspondant à la seconde occurrence et ainsi de suite... Ceci permet d'extraire très rapidement, en bouclant sur l'instruction, tous les indices correspondant à la valeur ou à la chaîne recherchée (voir listing 1).

Voyons maintenant la dernière et la plus puissante des actions proposées par ZXTRI : le tri. Outre le fait que ce tri soit hyper-rapide (puisque il est en langage machine et basé sur l'algorithme de Shell-Metzner), il permet de trier sur dix critères de tri simultanés et différents, un tableau multidimensionnel !

Mais parlons un peu vocabulaire. La notice de ZXTRI explique à l'aide de nombreux exemples ce qu'est un critère de tri : il ne s'agit pas de trier ici tout simplement un tableau par ordre croissant ou décroissant, mais de «dire» par exemple à ZXTRI : «je désire que mon tableau soit trié en ordre croissant sur la sous-chaîne comprise entre les bornes 3 et 5, et si l'on a égalité, par la suite entre les bornes 1 et 2». Cette très grande souplesse permet par exemple des noms d'articles associés à des numéros de



commandes, et à «l'intérieur» de chaque numéro de commande, d'avoir les articles triés selon leurs noms (voir copie écran).

La syntaxe correspondant à ce tri sera par exemple :

TRI A\$ (11 TO 15) C, (1 TO 10)C

où : A\$ (11 TO 10)

détermine le nom d'article

et : A\$ (11 TO 15)

détermine le n° de facture

La lettre C signifiant que l'on désire trier en ordre croissant.

Conclusion

ZXTRI est donc un peu plus qu'un utilitaire de tri et de gestion de tableaux : il permet de réaliser une simulation de gestion de fichiers sur disque, bien qu'opérant en mémoire centrale. Permettant, de plus, d'agir aussi bien sur des tableaux numériques qu'alphanumériques,

il s'adresse aussi bien aux applications scientifiques que de gestion.

Il permet de réaliser, avec un minimum

de connaissances du Basic, des programmes spécifiques puissants et rapides exactement adaptés aux besoins de l'utilisateur.

ZXTRI, en bref :

Pour :

- Compact (moins de 2 Ko de RAM!)
- Relogeable (4 versions différentes)
- Syntaxe simple et reprenant celle du Basic standard
- Manuel utilisateur clair et complet
- Rapide (tri au tableau de 50 postes en 8 secondes environ)

Contre :

- Un ordre de fusion de deux tableaux serait utile
- Demande un minimum de connaissances en Basic.

E. Viau

```
10 DIM A$(20,20)
20 LET F$="EXEMPLE"
30 LET L=LEN F$
40 LET Z=0
50 RAND USR ZXSORT
60 REM RECH A$(1 TO L)=F$:Z
70 IF Z=0 THEN STOP
80 PRINT A$(Z)
90 GOTO 40
```

```
1 REM **** ESSAI ****
20 DIM F$(22,14)
25 REM INITIALISATION
30 FOR I=1 TO 22
40 INPUT F$(I,1 TO 4)
50 INPUT F$(I,6 TO 10)
60 PRINT F$(I)
70 NEXT I
100 REM TRI DU TABLEAU
110 RAND USR 16516
120 REM TRI F$(1 TO 4)C,(6 TO 14)D
130 REM AFFICHAGE DU TABLEAU TRIE
140 FOR I=1 TO 22
150 PRINT AT (I-1),16;F$(I)
160 NEXT I
```

0401 BANANES	0008 CITRONS
3321 POMMES	0009 COINGS
4444 POIRES	0009 ABRICOTS
0401 FRAISES	0012 MANDARINE
6750 CERISES	0401 PRUNES
0401 PRUNES	0401 FRAISES
4444 ORANGES	0401 BANANES
0012 MANDARINE	3321 POMMES
5678 NOIX	3321 PASTÈQUES
4444 ANANAS	4444 POIRES
4444 AVOCATS	4444 PÈCHES
7864 MELONS	4444 ORANGES
3321 PASTÈQUES	4444 AVOCATS
0009 COINGS	4444 ANANAS
0008 CITRONS	5532 NOISETTES
4444 PÈCHES	5678 NOIX
0009 ABRICOTS	6542
8764 TOMATES	6750 CERISES
5532 NOISETTES	7864 MELONS
9888 AMANDES	8764 TOMATES
9876 0009	9876 0009
6542	9888 AMANDES

SINCLAIR POURQUOI PAS ?

ZX 81 Manette de jeu

Le clavier du ZX 81 étant sensitif, lors des jeux vidéo il est difficile de regarder et l'écran et le clavier. Une manette de jeu permet donc de mieux se concentrer sur l'image (et donc de faire de meilleurs scores).

Les manettes de jeu sont, en général, câblées comme des entrées sorties ce qui nécessite une interface électronique. Celle décrite ci-dessous utilise directement des possibilités du Sinclair puisqu'elle est montée comme un mini-clavier supplémentaire. (Fig 1)

Remarque : si la longueur du fil, entre le circuit imprimé et la manette, est supérieure à 20 cm, la touche SHIFT ne fonctionne plus très bien. Il faut donc pouvoir déconnecter la manette de jeu au niveau du boîtier.

Voyons maintenant comment faire ce branchement : le clavier est connecté par deux nappes de fils. Un premier de 5 fils (F1 à F5) et un deuxième de 8 fils (FA à FH).

Ces deux nappes de fils divisent le clavier de la façon suivante :

Repères Fils															
5	4	3	2	1	1	2	3	4	5						
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
A	1	2	3	4	5	6	7	8	9	0	C				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	0	W	E	R	T	Y	U	I	O	P	E				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D	A	S	D	F	G	H	J	K	L	N	G				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F	S	H	Z	X	C	V	B	N	M	--	--	H			

par exemple un contact entre les fils 5 et A donne le caractère "1" ou encore un contact entre les fils 5 et C donne le caractère "0".

Ces contacts seront testés par la fonction basic INKEY \$.

Voici comment retrouver les fils sur le circuit Sinclair vu du côté soudure :

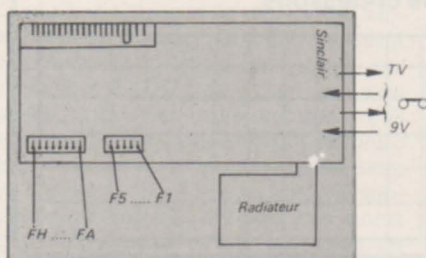


Figure 1 Schéma de principe

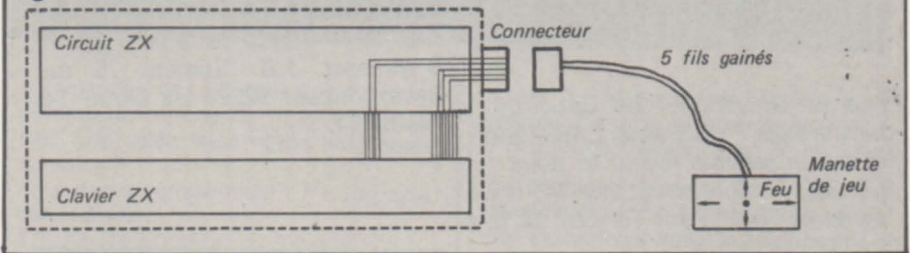
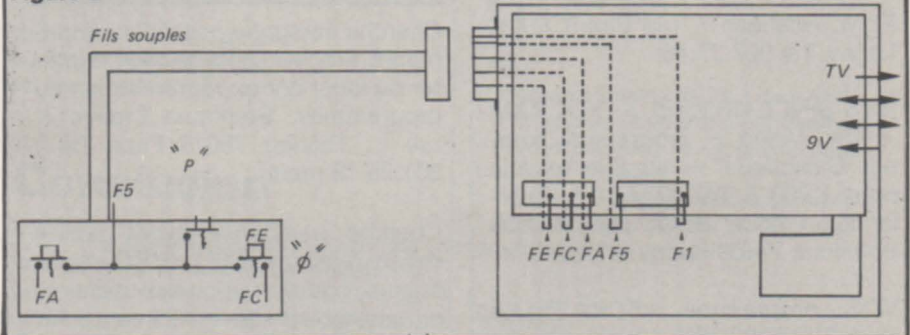


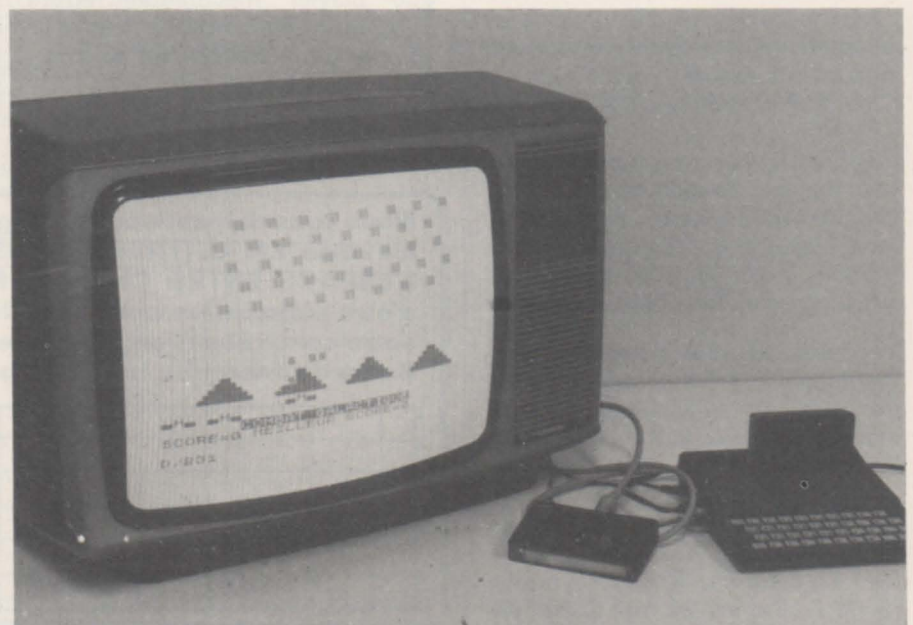
Figure 2



Le problème est le choix des touches à employer. A priori "5", "6", "7", "8" semblent pouvoir être un standard, mais aucun jeu du commerce ne le suit. Il vaut donc mieux opter pour un standard qui demande peu de fils :

"1" ← , "0" ← , "4" ↑ , "7" ↓ , "P" pour un tir.

Voyons maintenant la réalisation d'une manette très simple : ← , → , tir (BREAK OUT, INVADERS,...)(Fig. 2)



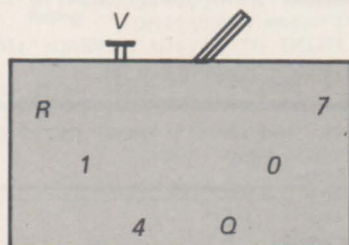
Les fils ((F1), FS, FA, FC, FE) sont à souder directement sous le connecteur du clavier et doivent être amenés à un nouveau connecteur installé sur le boîtier du ZX.

Pour faire une manette 4 directions le plus simple est d'en acheter une et de refaire les bons branchements (Fig. 3).

Remarque : les 5 fils tirés vers la manette permettent de créer un autre contact: touche ?.

La configuration de la manette que j'ai réalisée est la suivante :

V : 1F
R : 2B
1 : 5A
4 : 2A
7 : 2C
0 : 5C
Q : 5B



```

5 FAST
10 LET DFILE=PEEK 16396+256*PE
EK 16397
15 LET R=30000
20 LET M$="2A 20 6E.3A 33 6E.B
C.30 0A.3A 32 6E.BD.30 04.01 01
00.C9."
25 GOSUB 1000
30 LET M$="16 00.1E 00.2A 22 6
E.AF.BE.28 05.16 01.CD 4A 76.2A
24 6E.BE.28 05."
35 GOSUB 1000
40 LET M$="1E 01.CD 4A 76.BA.2
0 11.BB.20 0E.ED 4B 26.6E.09.BE.
28 2D.16 01.5A."
45 GOSUB 1000
50 LET M$="CD 4A 76.BA.28 0F.3
A 26 6E.ED 44.32 26 6E.3A 27 6E.
2F.32 27 6E."
55 GOSUB 1000
60 LET M$="AF.BB.28 0F.3A 28 6
E.ED 44.32 28 6E.3A 29 6E.2F.32
29 6E.18 11."
65 GOSUB 1000
70 LET M$="2A 26 6E.ED 4B 28.6
E.09.ED 4B 20.6E.C5.09.22 20 6E.
"
75 GOSUB 1000
80 LET M$="2A 26 6E.ED 4B 20.6
E.09.22 22 6E.2A 28 6E.09.22 24
6E.7A.83.57.AF.BA.20 07."
85 GOSUB 1000
90 LET M$="E1.36 00.60.69.36 B
0.0E 0A.06 00.04.20 FD.0D.20 FB.
3A 31 6E.47.AF.BB.38 04."
95 GOSUB 1000
100 LET M$="01 00 00.C9.3A 25 4

```

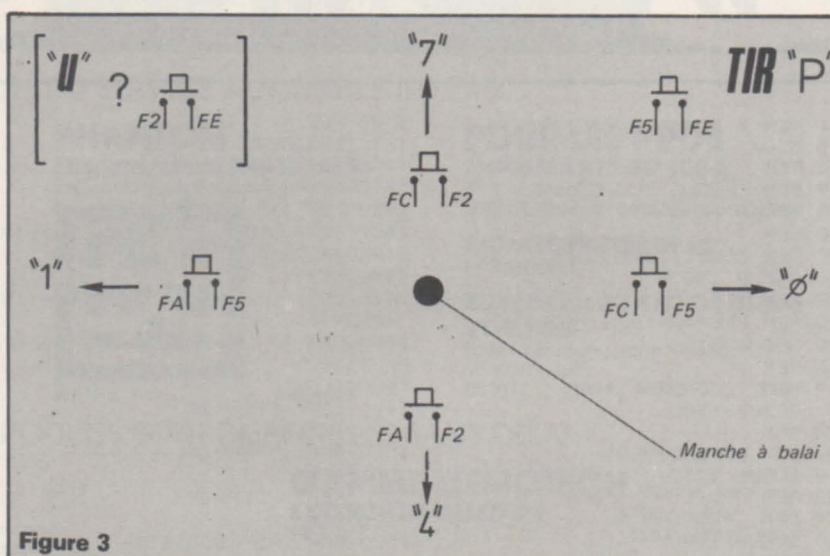


Figure 3

```

0.47.3E F7.BB.20 04.1E FD.18 07.
3E EF.BB.20 4E.1E 03."
105 GOSUB 1000
110 LET M$="3A 2A 6E.F5.83.06 F
E.BB.20 02.3E 01.06 1F.BB.20 02.
3E 1C.21 11 11.22 25 40.C1.48.F5
.06 00.2A 2B 6E."
115 GOSUB 1000
120 LET M$="E5.09.70.23.70.23.7
0.2A 2D 6E.E5.09.70.23.70.23.70.
E1.D1.F1.D5.4F.32 2A 6E."
125 GOSUB 1000
130 LET M$="16 80.09.72.23.72.2
3.72.E1.09.72.23.72.23.72.C3 30
75.3E FD.BB.C2 30 75."
135 GOSUB 1000
140 LET M$="01 01 00.C9.3E 08.B
E.20 06.36 00.21 31 6E.35.AF.C9.
"
145 GOSUB 1000
150 POKE 30153,52
151 POKE 30255,03
170 LET N=0
171 CLS
172 PRINT AT 3,9;"
176 PRINT AT 4,9;"CASSE-BRIQUES";
178 PRINT AT 5,9;"
182 PRINT AT 8,0;"POUR ALLER A
DROITE ""0"";AT 10,0;"POUR ALLE
R A GAUCHE ""1"";
184 PRINT AT 15,0;"QUEL NIVEAU
?","1 TRES BON","2 BON","3 MOY
EN","4 DEBUTANT"
186 PAUSE 40000
188 LET Z=(CODE INKEY$-28)*3+8
190 POKE 30155,Z
200 CLS
210 LET A=28192
220 LET B=INT (RND*30)+2+DFILE+
33*6
230 POKE A,B-256*INT (B/256)
235 POKE A+1,INT (B/256)
240 POKE A+2,B+1-256*INT ((B+1)
/256)
245 POKE A+3,INT ((B+1)/256)
250 POKE A+4,B+33-256*INT ((B+3
3)/256)
255 POKE A+5,INT ((B+33)/256)
260 POKE A+6,1
265 POKE A+7,0
270 POKE A+8,33
275 POKE A+9,0
280 POKE A+10,16
290 LET B=DFILE+33*21+1
295 POKE A+11,B-256*INT (B/256)
300 POKE A+12,INT (B/256)
305 LET B=DFILE+33*14+1
310 POKE A+13,B-256*INT (B/256)
315 POKE A+14,INT (B/256)
320 POKE A+17,89
325 LET B=DFILE+33*14+3
330 POKE A+18,B-256*INT (B/256)
335 POKE A+19,INT (B/256)
340 SLOW
360 PRINT "
380 FOR X=0 TO 21
390 PRINT AT X,0;"";TAB 31;"
400 NEXT X
410 FOR Y=5 TO 7
440 PRINT AT Y,1;"
450 NEXT Y
460 PRINT AT 14,16;"";AT 21,
16;"
500 LET A=USR 30000
510 IF A THEN GOTO 600
520 LET N=N+1
530 GOTO 200
600 LET P=N*90+90-PEEK 28209
610 PRINT AT 10,0;"VOUS AVEZ ";
P;"POINT";
620 IF P>1 THEN PRINT "S"
630 PRINT AT 12,0;"AU NIVEAU ";
(Z-8)/3
640 PRINT AT 20,0;"APPUYEZ POUR
UN AUTRE ESSAI"
650 IF INKEY$="" THEN GOTO 650
655 LET N=0
660 GOTO 200
999 STOP
1000 REM
1005 FOR X=1 TO LEN M$-1 STEP 3
1010 POKE R,(CODE M$(X)-28)*16+C
ODE M$(X+1)-28
1020 LET R=R+1
1030 NEXT X
1040 RETURN
2000 PRINT M$
2010 RETURN
9000 CLEAR
9005 SAVE "BRIQUES"
9010 RUN

```

SINCLAIR POURQUOI PAS ?

```

1 REM * GRUYERE DE POLIZZI **
5 SLOW
10 REM
99 REM REGLE DU JEU
100 GOSUB 1000
108 REM
109 REM CONDITIONS INIT
110 GOSUB 1200
118 REM
119 REM PREPARATION TERRAIN
120 GOSUB 1300
138 REM
139 REM DEPART
140 GOSUB 1500
148 REM
149 REM DEPLACEMENT
150 GOSUB 1730
158 REM
159 REM RESULTATS
160 GOSUB 1900
170 CLS
180 GOTO 120

1000 REM -----
1001 REM
1010 REM REGLE DU JEU
1020 REM
1030 PRINT " **** GRUYERE A TR
DUS ****"
1040 PRINT " -----
-----"
1050 PRINT
1051 PRINT
1060 PRINT "UNE PETITE SOURIS (
) VEUT"
1061 PRINT "GRIGNOTER LE MORCEAU
DE GRUYERE"
1062 PRINT "MAIS ELLE SOUFFRE D
AEROPHAGIE"
1063 PRINT "ET NE DOIT PAS AVALE
R D AIR (※). "
1070 PRINT
1080 PRINT "POUR LES DEPLACEMENT
S,UTILISEZ:"
1090 PRINT ""1",""0",""4",""
7""
1110 PRINT
1120 PRINT "ET TACHEZ D EVITER L
ES TROUS."
1130 RETURN
1200 REM -----
1201 REM
1210 REM CONDITIONS INIT
1220 REM
1230 PRINT
1235 LET HC=0
1240 PRINT "NIVEAU ? : "
1245 PRINT
1250 PRINT "1 DEBUTANT","2 MOYEN
","3 AVANCE","4 CHAMPION"
1260 IF INKEY$="" THEN GOTO 126
0
1265 LET REP=VAL INKEY$
1270 LET V=13-4*REP
1275 LET OB=30+25*REP
1280 RETURN
1300 REM -----
1301 REM
1310 REM PREPARATION TERRAIN
1320 REM
1325 CLS
1330 LET C=0
1335 FAST
1340 FOR X=0 TO 31
1350 PRINT AT 0,X;"※";
1360 PRINT AT 21,X;"※";
1370 NEXT X

```

```

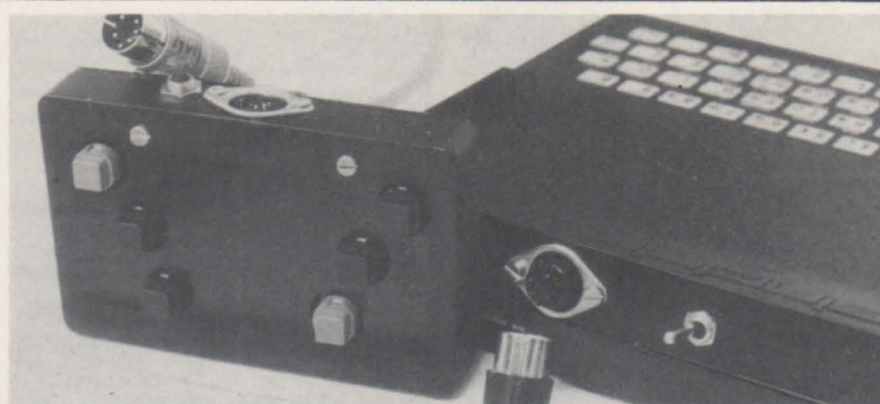
1380 FOR Y=0 TO 51
1390 PRINT AT Y,0;"*";
1400 PRINT AT Y,31;"*";
1410 NEXT Y
1420 FOR I=1 TO 0B
1430 PRINT AT (INT (RND*22)),INT
(RND*32);"*";
1440 NEXT I
1445 LET POS=INT (RND*29)+2+(INT
(RND*3)+1)*33
1450 LET DFILE=PEEK 16396+256*PE
EK 16397
1460 SLOW
1490 LET APOS=POS
1495 RETURN
1500 REM -----
1501 REM
1510 REM DEPART
1515 REM
1520 FOR I=5 TO 0 STEP -1
1530 POKE DFILE+POS,128+I*28
1540 FOR T=1 TO 20
1560 NEXT T
1570 NEXT I
1620 LET B$=INKEY$
1625 IF B$<>" " THEN RETURN
1627 FOR I=1 TO 3
1630 POKE DFILE+POS,0
1640 POKE DFILE+POS,128
1650 NEXT I
1660 LET B$="4"
1670 RETURN
1700 REM -----
1701 REM
1710 REM DEPLACEMENT
1715 REM
1716 REM
1717 REM LE CHOIX DES TOUCHES

```

```

EST EN 1700
1718 REM
1719 REM
1720 LET C$=INKEY$
1725 IF C$<>"1" AND C$<>"4" AND
C$<>"7" AND C$<>"0" THEN GOTO 1
730
1727 LET B$=C$
1730 LET POS=APOS+(B$="0")-(B$="
1")-33*(B$="7")+33*(B$="4")
1770 IF PEEK (DFILE+POS)<>0 THEN
RETURN
1780 POKE (DFILE+POS),CODE "1"
1785 POKE (DFILE+APOS),CODE "8"
1790 LET APOS=POS
1795 LET C=C+1
1800 FOR T=1 TO V
1810 NEXT T
1830 GOTO 1720
1900 REM -----
1901 REM
1910 REM: RESULTATS
1920 REM
1930 IF C>HC THEN LET HC=C
1940 PRINT AT 0,0;"SCORE:";C;"PO
INTS:";"";MAXI:";HC
1945. PRINT AT 21,15;"NIVEAU:";RE
P
1950 IF INKEY$<>" " THEN GOTO 19
50
1960 IF INKEY$<>" " THEN RETURN
1970 GOTO 1960
8997 REM
8998 REM -----
8999 REM
9000 CLEAR
9005 SAVE "GRUYERE"
9010 RUN

```



Cette configuration ne correspond pas avec celle décrite dans l'article pour les raisons suivantes :

— Je voulais cabler la touche SHIFT à la place du V actuel. La longueur du fil étant de plus de 15 cm cela ne fonctionnait pas. (j'ai donc mis V pour FEU [Pdans l'article]).

— Je n'ai jamais utilisé les touches R et Q ce qui permet de passer d'une prise DIN 7 broches (7 fils) à une prise DIN 5 broches qui est plus courante.

— La prise sur mon ZX81 est mal placée car elle interfère avec l'extension mémoire je suis donc obligé de brancher constamment l'Imprimante.

Pour le câblage de la manette de jeu il faut évidemment démonter les circuits du ZX si l'on n'a pas de fer à souder basse tension.

Sur mon ZX la carrosserie inférieure n'est pas maintenue par des vis.

Vous trouverez ci-contre deux programmes fonctionnant avec « ma » manette :

* Brigues (break out)

* Gruyère (met en évidence la difficulté d'utiliser des interrupteurs plutôt qu'une manette).

M. Polizzi
Club Paris Micro

VISMO

Vente Informations Services Micro-Ordinateurs
68, rue Albert - 75013 PARIS
Tél (1) 586.60.10
DEMONSTRATION TOUS LES JOURS DE 10h à 20h

**STOCK
PERMANENT**

LA QUALITÉ DU SERVICE AUX MEILLEURS PRIX

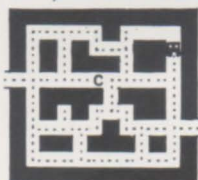
NOUVEAU

CLAVIER SPÉCIAL 160 F
(très esthétique)
se place simplement sur le clavier
d'origine par scotch double face.

INVERSEUR TV ET VIDEO 100 F
montage très facile à la portée de tous
Préréglé (avec schéma de montage).
Dimension : H = 2 cm, l = 7 cm.
Se loge sur le côté du boîtier.

EXT. MÉMOIRE 16 K 380 F
(en stock permanent)

**PHANTOM
LE (pac-man) FRANÇAIS**
sans problème de chargement.



K7 16 K Prix 100 F

**POUR LES PROS
IMPRIMANTE GP 100 A**

80 Col. Graph.
Listings tableaux copy d'écran. 2250 F

MONITEUR ZENITH 12"
ECRAN VERT 980 F

**SUPER CLAVIER MÉCANIQUE
TYPE PRO**

(avec barre d'espace touches
Jean Renaud)

EN KIT 300 F MONTÉ 390 F

TOUS CES PRODUITS SONT DISPONIBLES SANS DÉLAI

INTERFACE ZP 82 900 F

Pas de programme à charger. Permet de faire
du traitement de texte sur 80 col. Minusc. Accent. Livré
avec câble recopié d'écran avec la fonction copy.

PACK VISMO

GP 100 A + ZP 82 avec câble + 1000 pages listing... 3195 F
GP 100 A + INT. MEMOTECH + CABLES + 1000 pages
listing 3150 F

GAMME MEMOTECH [doc. en français]

EXTENSION MÉMOIRE 16 K 32 K 64 K
PRIX 430 F 695 F 995 F

H.R.G. Haute résolution graphique
192 x 248 points 795 F

INTERFACE CENTRONICS
compatible avec module HRG 595 F

CABLE LIAISON GP 100 A 150 F

Documentation gratuite contre 2 timbres à 1,80 F

NOUS EFFECTUONS LES ENVOIS CONTRE REMBOURSEMENT SUR SIMPLE APPEL TÉLÉPHONIQUE

sinclair ZX 81

VOUS NE POUVEZ L'IGNORER

Mais vous ne connaissez peut-être pas toute la
gamme des accessoires et des nombreux logiciels
directement utilisables sur ZX 81.

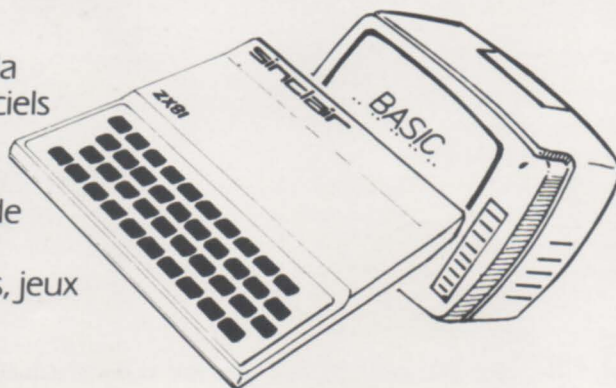
Cartes graphiques, sonores, génératrices de
caractères, extensions mémoire 64K, lecteur de
cassettes, consommables...

programmes de gestion personnelle, utilitaires, jeux
éducation, aide à la programmation.

En vente sur stock à



65/67 Bd St-Germain 75005 Paris
Tél. 325 68 88
Télex ETRAV 220064F/1303 RAC



Nom
Prénom
Adresse
Code Postal Ville

Je désire recevoir la liste des produits disponi-
bles pour le ZX 81.

Bon à retourner à **la Règle à Calcul**,
65/67 Bd St-Germain 75005 PARIS.



ES 2/82

PROMOTION



1

« OTHELLO »

Un jeu de stratégie désormais classique sur les micro-ordinateurs où vous jouez contre votre ZX 81. Deux programmes vous sont proposés sur la même cassette : l'un écrit en langage «basic», l'autre écrit en langage «machine». Chacun de ces programmes vous proposant plusieurs niveaux de difficultés.

135.00 Frs TTC.



CLAVIER MECANIQUE EN « ABS »

Se fixant directement sur le clavier initial du ZX 81, ce clavier autocollant en plastique «ABS» vous permet en quelques secondes de bénéficier d'un véritable clavier mécanique sans changer la configuration de votre ZX 81.

140.00 Frs TTC.

2

BON A ENVOYER A DIRECO INTERNATIONAL

30, avenue de Messine
75008 - PARIS

Je désire recevoir par paquet-poste :

① OTHELLO au prix promotionnel de 135 Frs

② Le clavier mécanique ABS au prix promotionnel de 140 Frs.

Je paie par CCP ou Chèque Bancaire établi à l'ordre de DIRECO

NOM.....PRENOM.....

N°.....RUE.....

CODE POSTAL.....VILLE.....

LE.....SIGNATURE.....