

PROGRAMME

UTILITAIRE

Ceux d'entre vous qui disposent du logiciel Pictor sur TO 7 se plaignent souvent du manque de précision du crayon optique. C'est pour ces puristes ou pour ceux qui ne l'auraient pas acheté que nous proposons, à l'occasion de ce spécial graphisme sur ordinateur, un outil de premier choix.

de M. ROUSSEAU
et M. PETREMAN

Ordinateur :

TO 7 + joystick
et cartouche Forth 79

Langage :

Forth 79

COMPILEUR GRAPHIQUE POUR LE TO7

La création de motifs graphiques de manière « traditionnelle », comme en Basic par exemple, reste une tâche ingrate. Le programme que nous vous proposons permet de créer des motifs graphiques de manière interactive sans avoir besoin d'écrire une seule ligne de code. Les dessins réalisés à l'aide du joystick sont compilés sous forme de mots s'intégrant au vocabulaire de base du langage Forth, et permettent ainsi de redessiner les motifs qu'ils représentent, ceci à n'importe quel endroit de l'écran.

Pour que le dessin soit tracé, il suffit de taper : n°-de-ligne, n°-de-colonne, NOM-DU-DESSIN.

Le compilateur graphique est équipé d'un menu d'aide, avec fenêtrage, et d'une routine de transfert du contenu de l'écran vidéo vers une imprimante graphique Seikosha GP 500A.

Le compilateur graphique

Le compilateur graphique permet la gestion d'un curseur graphique, via le joystick, et la compilation des fonctions de tracé ou de déplacement.

Les mots générés par le compilateur graphique représentent les dessins réalisés à l'aide du joystick et peuvent être retracés en n'importe quel endroit de l'écran en précisant simplement

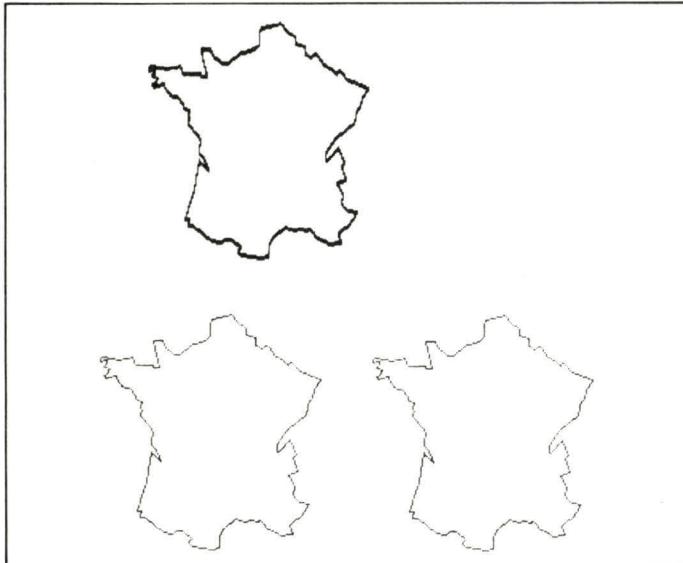


Fig. 1. – Un exemple de dessin pour lequel on peut effectuer le tracé et la génération d'une routine Forth.

les coordonnées du point de départ du dessin.

Par exemple, la carte de France montrée **figure 1** a été compilée grâce au mot DESSIN (voir **listing**). Lors de l'exécution de DESSIN, divers renseignements sont demandés, à commencer par le nom qui sera affecté au dessin à compiler (ici France), puis les coordonnées x et y du point de départ. Le contour de la France a été dessiné en manœuvrant le joystick et en effectuant le tracé par segments successifs. Une fois le contour achevé, on choisit l'option S, réalisant ainsi la compilation du mot France. Pour retracer France, il suffit de préciser les coordonnées x et y où l'on désire voir le tracé s'effectuer.

Les paramètres à gérer

Afin d'éviter des perturbations dues à d'éventuelles homonymies de définitions, les mots du compilateur graphique sont définis dans le vocabulaire JOYSTICK.

Le système de gestion des graphismes, sur le TO 7, ne permettant pas une gestion correcte des fonctions de tracé et d'effacement d'un point, il a été nécessaire de recourir à des fonctions particulières. Chaque point de l'écran vidéo (320 par 200) représente un bit dans un groupe de huit bits. Par ligne graphique, il y a quarante groupes de 8 bits soit 40 octets. Ainsi, pour allumer ou éteindre un point sur l'écran, on utilisera

de préférence les fonctions logiques OR et XOR. Pour réaliser ceci, le mot ADRCALC (**listing** écran 11) calcule l'adresse et la position dans l'octet du point de coordonnées x et y de l'écran. Ce calcul est simple : l'adresse contenant le bit correspondant au point x y est égale à l'adresse du début de l'écran (16384d, 4 000h) à laquelle s'ajoutent le produit y*40 et le quotient entier de x/8. Le reste de x/8, c'est-à-dire x MOD 8 correspond à la position du bit dans l'octet en question. Par exemple, si le reste est 3, c'est le troisième point en partant de la gauche qui est pris en considération. Cette lecture s'effectue de gauche à droite. Le mot DECALAGE (**listing**, écran 11) délivre la valeur du bit correspondant à son poids (2EO = 1, 2E1 = 2, 2E2 = 4, etc.) où n dans 2En est égal à 7 - (x MOD 8). Le mot PLOT trace le point situé en colonne x, ligne y. Si l'octet pointé n'est pas vide, la fonction OR ne fera que modifier le bit pointé en le faisant passer à 1.

Par exemple :

00010110 octet d'origine calculé par ADRCALC
OR 10000000 valeur calculée par DECALAGE

10010110 valeur remise à l'adresse calculée par ADRCALC pixel nouvellement allumé.

Le mot UNPLOT efface le point précédemment tracé par PLOT. Le bit est remis à zéro sans altérer les autres bits de l'octet pointé.

10010110 octet d'origine calculé par ADRCALC
XOR 10000000 valeur calculée par DECALAGE

00010110 valeur remise à l'adresse calculée par ADRCALC

Notons que PLOT et UNPLOT ont été conçus de manière à permettre le passage du curseur au travers de dessins déjà tracés sans que ceux-ci en soient altérés.

PROGRAMME

UTILITAIRE

Les mots OCT@ et OCT! prélèvent et restaurent la valeur initiale de l'octet avant l'exécution de PLOT et UNPLOT.

La position du curseur graphique est gérée par deux paramètres, DX et DY, positions relatives par rapport au point de départ dudit curseur. Ce point de départ dépend des valeurs contenues dans xo et yo (endroit où vous appuyez pour la première fois pour tracer votre dessin). Le mot X1Y1 va délivrer les coordonnées x et y de votre curseur en valeur absolue ; c'est-à-dire l'endroit où se trouve votre curseur (sans faire référence au point de départ cette fois-ci). Afin de limiter la taille de notre code, nous allons enfermer les coordonnées relatives de notre curseur (DX et DY) dans l'intervalle -127 à +127. Ceci est réalisé par le biais du mot LIMITE. Ne vous étonnez donc pas si, votre segment étant trop long, votre curseur refuse d'avancer ! Il vous faudra tracer le segment en plusieurs fois, voilà tout ! A titre indicatif, sachez que la diagonale de l'écran se fait en trois segments et ne consomme que 9 octets !

Les mots INCX, DECX, INCY et DECY autorisent l'incrémentation ou la décrémentation du contenu des variables DX et DY si leur position absolue n'est pas en dehors des limites de l'écran vidéo et si elle est en conformité avec les conditions testées par LIMITE.

Le mot INCREMENTE modifie les valeurs contenues dans DX et/ou dans DY en fonction de la valeur déposée sur la pile. Cette valeur, variant de 1 à 8, correspond à la position du joystick (fig. 2).

La modification du contenu de DX et DY s'accompagne d'un affichage conditionnel de la position en valeur absolue du curseur. L'affichage des coordonnées est réalisé si le contenu de COOR, la variable qui sert de drapeau, est différent de zéro. Puis le contenu de l'octet pointé par le curseur est sauvegardé avant altération par PLOT (listing écran 13).

Le mot DEPLACEMENT gère l'affichage, l'effacement et le déplacement du curseur graphique. Ce déplacement est effectif et immédiat pour toute manœuvre du joystick.

La variable COMPTEUR est alors incrémentée, et le déplacement ne reprend que lorsqu'elle dépasse 1000. En pratique, ceci permet de manœuvrer le curseur point par point, la répétition pour un même sens de déplacement se trouvant ainsi temporisée. Le mot TRACE (listing écran 14) récupère les valeurs de DX et de DY et trace un segment de droite, en partant de la position d'origine du curseur en allant vers sa position actuellement occupée. Afin de redémarrer un nouveau tracé, la nouvelle position du curseur est réinitialisée par incorporation de ses coordonnées dans XO et YO, ceci s'accompagnant de la remise à zéro de DX et de DY. Le mot DEPLACE assure les mêmes fonctions que TRACE, mais, à la différence de celui-ci, aucun segment n'est dessiné.

Le mot MENU (listing écran 15) affiche un menu lors de la définition d'un dessin, permettant d'informer l'utilisateur sur les options à sa disposition : (T)racer, (E)ffacer, (I)mprimer et (S)top.

Afin de ne pas recouvrir un dessin en cours de définition, par l'exécution de MENU, une fonction de fenêtrage a été prévue, réalisée à l'aide des mots SAUVEDESSIN et RAMEDEESSIN. SAUVEDESSIN (listing écran 15) sauvegarde le contenu des quatre premières lignes de l'écran vidéo commençant à l'adresse 16384, ceci donc sur 1 280 octets, dans une zone mémoire située 80 octets au-dessus du dictionnaire

(PAD). Le fait que cette adresse change constamment n'est pas gênant, car l'opération de sauvegarde graphique est faite à un moment où il n'y a pas de compilation.

RAMENEDESSIN réalise l'opération inverse. Ces deux mots sont combinés dans IMA-GESWAP. Dans un premier temps, le contenu des quatre premières lignes est sauvegardé. Dans un second temps, le menu d'aide est affiché, et dans un dernier temps, dès l'appui sur une touche, le dessin est restauré.

La compilation des données

Si l'on cherchait à compiler de manière « traditionnelle » un motif graphique, la mémoire disponible sur notre système diminuerait rapidement. Ainsi, pour une séquence du type xo yo xe LINE, il ne faut pas moins de 18 octets en mémoire, ceci pour tracer un simple segment de droite. Il s'avère nécessaire de trouver un moyen plus économique pour représenter notre motif. Or un dessin, pour un micro-ordinateur, n'est le plus souvent qu'une suite de segments de droite de tailles diverses. Si chaque coordonnée peut tenir sur un octet, nous pourrons concentrer l'information représentant notre dessin sur deux octets. Si chacun de ces octets représente une donnée relative, l'amplitude du tracé, par rapport au point d'origine, sera comprise entre -127 et +127 (ce qui explique la limitation de la longueur de notre segment que nous évoquions auparavant).

Si, de plus, on prévoit la possibilité de se déplacer sans tracer de trait en « levant le crayon », il faudra réservé un octet supplémentaire pour ranger le code de l'opération : 1 pour tracer un trait, 2 pour se déplacer sans tracé (il reste 254 possibilités que nous laissons aux bons soins de votre imagination débordante).

En partant de ce principe, la représentation d'un dessin se réduit à une suite d'octets, rangés

trois par trois en mémoire. Le premier octet représente la nature de l'opération (tracé ou déplacement) et les deux autres les coordonnées dx et dy du tracé. Le tracé du premier segment du dessin a pour origine le point dont les paramètres sont sur la pile au départ, les segments suivants ayant pour origine l'extrémité des segments précédemment tracés, ou le point d'arrivée d'une opération de déplacement sans tracé. On peut donc compiler un motif graphique composé de plusieurs dessins qui ne sont pas visuellement liés.

Le mot COMPLEMENT (listing écran 16) transforme un octet non signé (intervalle 0..255) en une valeur 8 bits signnée (intervalle -128..+127).

Le mot INC-PFA incrémentera le contenu du pfa du mot à définir, permettant par la suite, lors de l'exécution de ce mot, de connaître le nombre de segments ou de déplacements compilés. Le mot !DICT-DXY compile le contenu de DX et de DY dans le dictionnaire.

La compilation du dessin est réalisée par EN-ROUTE (listing écran 17). On sélectionne une option en appuyant sur le bouton situé à côté du joystick (commande BUTTON). Le changement de couleur du bord de l'écran indique que la commande a bien été prise en compte. Choisissez alors une option. Si vous choisissez l'option « A », le menu s'affichera. Si vous avez fait une fausse manœuvre, appuyez sur la barre d'espacement.

Enfin, « last but not least », le mot DESSIN (listing, écran 18). Ce mot a deux actions : une première en mode compilation, une seconde lors de l'exécution. En compilation, DESSIN, étant un mot de définition (jetez un coup d'œil sur les structures CREATE..DOES), après s'être branché sur le vocabulaire JOYSTICK, va vous demander le nom du mot à créer. Ce nom, il le conserve temporairement dans le TIB (opération réalisée via QUERY). Ensuite, DESSIN



Fig. 2. – Les codes des huit directions possibles d'un joystick.

PROGRAMME

UTILITAIRE

va vous demander tour à tour les coordonnées verticales et horizontales de départ du tracé à définir. Puis il vous demandera si vous désirez voir s'afficher en permanence les coordonnées du curseur (**fig. 3**). Votre en-tête

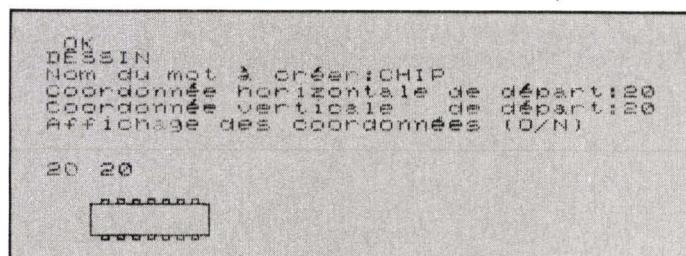


Fig. 3. – La routine DESSIN permet de situer l'origine du dessin ainsi que de choisir la visualisation des coordonnées courantes du curseur.

étant ainsi créé, à vos manettes !

Pour dessiner, choisissez un sens de déplacement en inclinant votre joystick. Une fois votre dessin terminé, choisissez l'option S (Stop) qui arrêtera la compilation. Pour sauvegarder votre dessin, choisissez l'option de sauvegarde en version compilée (n SAVE). Pour rappeler par la suite un dessin sauvegardé, choisissez l'option de récupération (n RUN). Il vous suffit alors, afin de voir votre œuvre apparaître à l'endroit désiré, de taper son nom précédé des coordonnées x et y de son point de départ. En guise d'illustration des possibilités de ce compilateur, reportez-vous aux exemples présentés. Mais ceux-ci ne sont pas exhaustifs, loin de là ! Voyons maintenant comment imprimer nos gribouillis sur l'imprimante.

Copie de l'écran sur l'imprimante

La réalisation d'une routine de copie du contenu de l'écran vidéo sur une imprimante graphique est parfois un travail fastidieux en assembleur. Outre une bonne connaissance des instructions spécifiques du microprocesseur, il est nécessaire de générer un code de préférence relogable et, si possible, de l'implanter dans un espace mémoire neutre. Aussi doit-on connaître parfaitement l'architecte de la mémoire (memory-

mapping et variables système notamment).

La création en Forth d'une routine remplissant la même fonction pose nettement moins de problèmes, même si l'exécution s'avère un peu plus lente.

cielle, la représentation d'un élément graphique est différente de celle d'un élément de l'écran vidéo. Sur l'imprimante Seikosha GP 500 A, un élément graphique est représenté par sept points alignés verticalement.

Les bits b0 à b6 représentent le motif graphique à imprimer. Un bit à zéro correspond à un point vide, et un bit à un à un point plein. Quant au bit b7, il est toujours à un. Les octets reçus par l'imprimante, en mode graphique, sont compris dans l'intervalle 128... 255. Les autres codes sont simplement exécutés (CR, LF, etc.). Le passage en mode graphique à partir du mode texte est réalisé par l'envoi vers l'imprimante du code ASCII de valeur 8. Le retour en mode texte est réalisé par l'envoi des codes ASCII 14 ou 15, selon que l'on est en double ou simple largeur.

La transmission du contenu de l'écran vidéo est donc réalisée par SCRCOPY (**listing écran 10**). L'image vidéo est analysée par tranches de sept pixels. Deux boucles DO...LOOP sont initialisées l'une avec le nombre de lignes graphiques, l'autre avec le nombre de points par ligne. Le nombre de lignes à analyser est donné par une valeur correspondant à un multiple du nombre de lignes par tranche de sept pixels. Ce nombre est immédiatement inférieur au nombre maximum de lignes de l'écran vidéo, d'où la valeur retenue : 196 (n1 = 200 mod 7, n2 = n1 × 7). Une troisième boucle, imbriquée dans les deux précédentes, explore un segment d'image, point par point,

sur une largeur d'un pixel et une hauteur de sept pixels. Au cas où la valeur délivrée par le test du pixel est supérieure ou égale à zéro (point « forme » allumé), la variable OCTET est incrémentée de la valeur du poids correspondant à la position du bit testé (ce qui est en réalité égal au pixel ; cf. **figure 4**). Lorsque la totalité du segment de sept pixels a été testée, la valeur de la variable OCTET est incrémentée de 128 unités (ce qui place le bit b7 à 1) puis est envoyée à l'imprimante. Une fois toute une tranche (soit 320 segments de 7 pixels) ainsi envoyée, un retour chariot est exécuté, ce qui provoque l'impression du contenu du tampon graphique de l'imprimante. L'analyse de l'image vidéo se poursuit sur les sept lignes de pixels suivantes. Arrivée en fin d'image, l'imprimante est remise en mode texte, ceci après impression de la dernière tranche, puis l'interpréteur Forth redonne la main à l'utilisateur. Signalons que lors de l'exécution de la routine SCRCOPY, ce processus ne peut être interrompu qu'au moyen d'un RESET ; mais, dans ce cas, on perd le dessin en cours d'impression.

Adaptation à d'autres imprimantes

Afin de permettre l'utilisation d'autres imprimantes dont les fonctions graphiques diffèrent de celles de la GP 500A, voici la liste des paramètres devant être modifiés (**fig. 5**).

Premier paramètre : Le code 8 sélectionne le mode graphique.

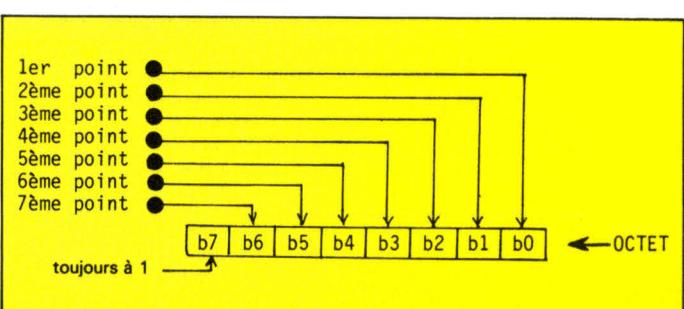


Fig. 4. – L'octet envoyé à l'imprimante utilisée est constitué de 7 bits d'information et d'un huitième toujours à 1.

PROGRAMME

UTILITAIRE

```

: SCRCOPY
PRINTER 8 EMIT
196 0
DO
320 0
DO
0 OCTET !
1 J DUP [7 + SWAP]
DO
J I POINT DUP 0= SWAP 0 > OR
IF
DUP OCTET +!
THEN
2 *
LOOP
OCTET @ 128 +
REGD ! [ HEX ]
E812 CALL
[ DECIMAL ] DROP
LOOP CR
7 +LOOP
15 EMIT
CONSOLE ;

```

Fig. 5. – Liste des paramètres de SCRCOPY devant être adaptés en cas de changement d'imprimante.

CORRESPONDANCE POINT/VALEUR							
1er point	● 1	● 64	● 1	● 128			
2ème point	● 2	● 32	● 2	● 64			
3ème point	● 4	● 16	● 4	● 32			
4ème point	● 8	● 8	● 8	● 16			
5ème point	● 16	● 4	● 16	● 8			
6ème point	● 32	● 2	● 32	● 4			
7ème point	● 64	● 1	● 64	● 2			
8ème point	x x	x x	● 128	● 1			
cas 1			cas 2		cas 3		cas 4

Fig. 6. – Table de correspondance entre le code d'un point et sa position dans l'octet à transmettre à l'imprimante choisie.

Ce paramètre est à modifier en fonction des caractéristiques propres à l'imprimante dont vous disposez. Dans le cas de certaines imprimantes, le passage en mode graphique doit être suivi du nombre d'octets du tampon graphique, le retour en mode texte étant automatique. La séquence de sélection du mode graphique devra donc être implantée après le premier DO de SCRCOPY.

Deuxième paramètre : Ce nombre correspond au poids du bit qui sera envoyé comme premier point du segment graphique. Si ce premier point fait partie d'un segment de sept points et cor-

respond au bit de poids le plus élevé, il faut remplacer cette valeur par 128 (cas 2, fig. 6 : cf. le quatrième paramètre). Si le point fait partie d'un segment de huit points ayant les mêmes caractéristiques que précédemment (cas 4, fig. 6), remplacer 1 par 256.

Troisième paramètre : La valeur 7 dépend du nombre de points à imprimer par unité graphique. Dans les cas 3 et 4 (fig. 6), remplacer cette valeur par 8.

Quatrième paramètre : Pour une impression correspondant aux cas 2 et 4 (fig. 6), remplacer 2 * par 2 /.

Cinquième paramètre : Pour une impression correspondant aux cas 1 et 2 (fig. 6), avec une valeur d'octet comprise entre 128 et 255, ne rien changer ; dans les autres cas, cette routine (128+) peut être supprimée.

Sixième paramètre : Ces commandes sont spécifiques au Forth du TO 7 (ou TO 7-70). La valeur hexadécimale E812 correspond à la routine RSCO\$ du moniteur. Lors de l'emploi de EMIT, Forth filtre les codes compris entre 128 et 255 et empêche l'envoi de ces codes pour certaines valeurs. Par exemple, les valeurs correspondant à des caractères accentués sont trans-

codées en leur équivalent non accentué. On contourne ce filtrage en injectant ces codes dans le registre D et en appelant la routine RSCO\$. Sur d'autres systèmes, cette sous-routine peut être remplacée par EMIT.

Septième paramètre : Pour une impression correspondant aux cas 1 et 2 de la figure 6, ne rien changer. Dans les autres cas, remplacer 7 par 8.

Huitième paramètre : La valeur 15 correspond à la sélection en mode texte en simple largeur pour l'imprimante GP 500A. Ce paramètre est à adapter en fonction des caractéristiques propres de votre imprimante. ■

```

SCR: 10
VARIABLE OCTET
: SCRCOPY ( --- ) 
PRINTER 8 EMIT
196 0
DO

```

```

320 0
DO
0 OCTET ! ( Mise à zéro de OCTET )
1 J DUP 7 + SWAP
DO
J I POINT DUP 0= SWAP 0 > OR
IF
DUP OCTET +! ( Mise à jour )
THEN ( de l'octet à envoyer )
2 *
LOOP
OCTET @ 128 +
REGD ! [ HEX ]
E812 CALL
[ DECIMAL ] DROP
LOOP CR ( Impression du tampon )
7 +LOOP
15 EMIT ( Mode caractères standards )
CONSOLE ; ( Affichage vers écran )

```

```

SCR: 11
VOCABULARY JOYSTICK IMMEDIATE
JOYSTICK DEFINITIONS
VARIABLE OCTET
VARIABLE X0 VARIABLE Y0
VARIABLE DX VARIABLE DY
VARIABLE COOR 0 COOR ! ( Affiche x y )
VARIABLE ADROCTET ( Adresse de l'octet )
VARIABLE COMPTEUR ( Tempo. joystick )

```

```

: EXP2 ( n --- 2En Elévation au carré )
1+ 1 SWAP 0 DO 2* LOOP 2/ ;
: ADRCALC ( x y --- décalage adresse )
16384 SWAP 40 * + SWAP ( Donne adr der- )
8 /MOD ROT + SWAP ; ( -nier point tracé )

```

Listing du programme.

PROGRAMME

UTILITAIRE

```

: DECALAGE ( décalage --- 2Enversedéc.)  

7 SWAP - EXP2 ;  
  

: PLOT ( COL LIG --- met pixel en x y )  

( Trace le point sans passer par PSET )  

ADRCALC DECALAGE OVER C@ OR SWAP C! ;  
  

: UNPLOT ( COL LIG --- efface pixel )  

ADRCALC DECALAGE OVER C@ XOR SWAP C! ;  
  

SCR: 12  

: X1Y1 ( --- x1 y1 )  

( Délivre les coordonnées absolues X Y)  

X0 @ DX @ + Y0 @ DY @ + ;  
  

: LIMITE ( adr --- )  

DUP >R @ DUP 127 >  

IF 127 R@ ! DROP ELSE -127 <  

IF -127 R@ ! THEN THEN R> DROP ;  
  

( Incrémentation conditionnelle des co-)  

( -ordonnées relatives DX et DY )  

: INCX ( --- ) X1Y1 DROP  

319 < IF 1 DX +! THEN DX LIMITE ;  

: DECX ( --- ) X1Y1 DROP  

0 > IF -1 DX +! THEN DX LIMITE ;  

: INCY ( --- ) X1Y1 SWAP DROP  

199 < IF 1 DY +! THEN DY LIMITE ;  

: DECY ( --- ) X1Y1 SWAP DROP  

0 > IF -1 DY +! THEN DY LIMITE ;  
  

: OCT@ ( --- ) X1Y1 ADRCALC  

DROP DUP ADROCTET ! C@ OCTET ! ;  
  

: OCT! ( --- )  

OCTET @ ADROCTET @ C! ;  
  

SCR: 13  

: INCREMENTE ( c --- )  

( incrémente DX et DY et déplace point)  

CASE  

X1Y1 UNPLOT OCT!  

( Restauration du contenu d'origine de)  

( l'octet apres suppression du point)  

1 OF DECY ENDOF  

2 OF INCX DECY ENDOF  

3 OF INCX ENDOF  

4 OF INCX INCY ENDOF  

5 OF INCY ENDOF  

6 OF DECX INCY ENDOF  

7 OF DECX ENDOF  

8 OF DECX DECY ENDOF  

ENDCASE  

COOR @ ( Affichage conditionnel des pa-)  

IF ( -ramètres absolu X1 et Y1 )  

X1Y1  

0 0 LOCATE .  

THEN  

OCT@ ( Lecture et stockage avant )  

X1Y1 PLOT ; ( altération )

```

```

SCR: 14
: DEPLACEMENT ( --- )
INCREMENTE ( déplace le pixel en X1Y1 )
0 COMPTEUR !
BEGIN
COMPTEUR @ 1000 >
IF ( si oui, déplace le pixel )
0 STICK
INCREMENTE
ELSE ( sinon incrémente le compteur )
1 COMPTEUR +!
THEN
0 STICK
0= ( teste si le stick )
UNTIL ; ( est revenu au repos )

```

```

: TRACE ( --- )
X0 @ Y0 @ OVER DX @ + OVER DY @ +
OVER OVER Y0 ! X0 ! LINE
0 DX ! 0 DY ! COLOROFF OCT@ ;

```

```

: DEPLACE ( --- )
( Remise à zéro des vecteurs relatifs)
X1Y1 Y0 ! X0 ! 0 DX ! 0 DY ! ;

```

```

SCR: 15
: MENU ( --- )
HOME 0 1 SIZE
." AIDE MEMOIRE " 0 0 SIZE
." (T)race" 12 SPACES
." (D)éplace" 10 SPACES
." (E)fface l'écran "
." (I)mprime" 10 SPACES
." (S)top" 6 SPACES
." Appuyez sur une touche "
KEY DROP ;

```

```

: SAUVEDESSIN ( --- )
COLOROFF
16384 ( Début de la mémoire écran )
HERE 80 + ( Adresse 1er octet libre )
1280 ( Nombre d'octets à déplacer)
CMOVE ;
```

```

: RAMENEDESSIN ( --- )
COLOROFF HERE 80 + 16384 1280 CMOVE ;

```

```

: IMAGESWAP ( --- )
SAUVEDESSIN MENU RAMENEDESSIN ;

```

```

SCR: 16
: COMPLEMENT ( c --- c' compl. à 2 )
DUP 127 >
IF
256 -
THEN ;

```

PROGRAMME

UTILITAIRE

```
: INC-PFA    ( incrémentation du contenu)
LATEST      ( du PFA de trois unités )
PFA 3 SWAP +! ;
: !DICT-DXY  ( Met DX et DY dans dict.)
DX @ C,   DY @ C, ;
```

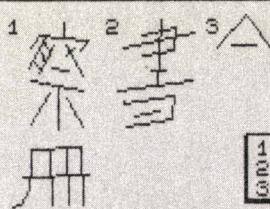
```
SCR: 17
HEX
: EN-ROUTE < --- >
CSROFF
BEGIN
 0 STICK
  IF
    0 STICK COLOROFF DEPLACEMENT
  ELSE
    0 BUTTON
    IF
      03 FRAME KEY
      CASE
        " T" ASC OF 01 C, !DICT-DXY
          INC-PFA TRACE ENDOF
        " D" ASC OF 02 C, !DICT-DXY
          INC-PFA DEPLACE ENDOF
        " E" ASC OF CLS ENDOF
        " I" ASC OF SCRCPY ENDOF
        " A" ASC OF IMAGESWAP ENDOF
        " S" ASC OF 6 FRAME CSRON EXIT ENDOF
        ENDCASE 6 FRAME
      THEN
      THEN 0
    UNTIL ; FORTH DEFINITIONS
DECIMAL
```

```
SCR: 18
: DESSIN  ( --- ) JOYSTICK
CR ." Nom du mot à créer:" QUERY
CR ." Coordonnée horizontale de départ:" INPUT DROP X0 !
CR ." Coordonnée verticale de départ:" INPUT DROP Y0 !
CR ." Affichage des coordonnées (O/N)" KEY " O" ASC =
  IF 1 COOR ! ELSE 0 COOR ! THEN
    CREATE 0, EN-ROUTE
DOES>
ROT ROT Y0 ! X0 !
DUP @ SWAP 2+ SWAP OVER >R + R>
DO
  I C@ CASE
    1 OF I 1+ @ HILO
      COMPLEMENT DX ! COMPLEMENT DY !
      TRACE ENDOF
    2 OF I 1+ @ HILO
      COMPLEMENT DX ! COMPLEMENT DY !
      DEPLACE ENDOF
  ENDCASE
3 +LOOP ;
```

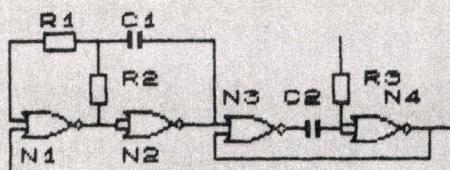
```
SCR: 19
: MENU-CHINOIS
CLS 1 1 LOCATE ." 1      2      3"
20 20 CHA' 20 60 CE"
65 20 SHU'
110 20 JI'
16 7 LOCATE ." 1..Consulter une fiche"
16 8 LOCATE ." 2..Ecrire une fiche"
16 9 LOCATE ." 3..Fusion de fiche"
0 0 ENCADRES ;
```

```
SCR: 20
CLS
CR ."           R1  C1" CR CR CR
."           R2           R3"
CR 13 SPACES ." N3 C2  N4"
CR CR CR ."     N1  N2"
CR CR CR 1 0 SIZE
."           TESTEUR" CR
."           DE SERVOCOMMANDE"
```

```
30 20 RES 60 20 HCAP 55 30 VRES
25 50 NOR 65 50 NOR 105 52 NOR
130 57 HCAP 155 52 NOR
151 30 VRES
20 20 CIRCUIT
```



1..Consulter une fiche
2..Ecrire une fiche
3..Fusion de fiche



TESTEUR
DE SERVOCOMMANDE

PROGRAMME

UTILITAIRE

Ce logiciel, écrit exclusivement en langage machine, permet de rendre le clavier du Canon X 07 conforme au standard français Azerty, lui ouvrant ainsi de nombreux horizons.

d'E. SANDER

Ordinateur :

Canon X 07

Langage :

Machine NSC-800
(compatible Z 80)

UN CLAVIER AZERTY POUR CANON X07

Posséder un clavier au standard Azerty constitue, même pour un micro-ordinateur à vocation familiale, un atout appréciable. C'est pourquoi, beaucoup d'ordinateurs portables sont commercialisés dans l'Hexagone dans des versions conformes au standard national. Tel n'est malheureusement pas le cas du Canon X 07 dont les utilisateurs se voient contraints de se conformer à la norme anglo-saxonne Qwerty. Une caractéristique qui peut se révéler bien gênante pour certaines applications telles que le traitement de texte.

Le programme que nous vous présentons propose une solution logicielle à ce problème apparemment insurmontable en transformant le Canon X 07 en un ordinateur polyvalent, laissant le choix du mode (Qwerty ou Azerty) à l'utilisateur.

Pour la forme Azerty, nous avons essayé (compte tenu de la disposition des touches sur le clavier) de respecter le plus complètement la norme française : outre l'inversion des positions des touches A, Q et Z, W, la place de la touche M a été modifiée ainsi que le statut de la zone numérique. La figure 1 représente les fonctions des différentes touches du Canon X 07 dans ce mode. Pour s'y retrouver aisément, nous ne saurions trop recommander la confection d'un cache qui permettrait ainsi une visualisation aisée du rôle de chaque touche et ce, quel que soit le mode sélectionné.



L'implantation

Celle-ci peut s'effectuer de deux manières. Les possesseurs d'un assembleur pourront se référer au listing source de la figure 2. Ce dernier est issu de l'assembleur à deux passes de J. Outhier, paru dans le numéro 49 de *Micro-Systèmes* et à l'aide duquel ce logiciel a été conçu. L'adresse d'implantation choisie par défaut est &H1C00 mais chacun est libre de la modifier à l'aide de la pseudo-instruction ORG.

Pour ceux ne disposant pas d'un assembleur, la saisie peut se faire en se référant à la liste des codes hexadécimaux de la figure 3 et en utilisant le chargeur hexadécimal de la figure 4. L'implantation s'effectue alors par série de huit octets (soit seize chiffres hexadécimaux non séparés par des espaces) suivant une méthode désormais bien connue des lecteurs de *Micro-Systèmes* dont l'efficacité a été fréquemment vérifiée : après chaque validation par un appui sur la touche RETURN, la somme de contrôle est exigée (il s'agit du nombre décimal inscrit à droite de chaque ligne). Dans le cas de détection d'une erreur (si la somme calculée ne correspond pas à celle introduite), il est nécessaire de réintroduire le groupe de huit octets incriminé. La phase d'implantation se termine avec l'affichage du message « fin de saisie » indiquant que le programme est opérationnel.

L'utilisation

L'usage du logiciel est assez simple. Son lancement se fait par EXEC \$H1C00. Cette opération terminée, le logiciel est alors totalement interactif même après l'extinction du micro-ordinateur. Seule l'utilisation du poussoir RESET lui est fatale. Le changement de mode (Qwerty ou Azerty) se fait suivant un classique système de bascule. Un premier

PROGRAMME

UTILITAIRE

appui sur CTRL-P provoque le passage en mode Azerty, tandis qu'une seconde pression renvoie à la configuration originale (Qwerty). Il est à noter que lors de l'utilisation du logiciel, à l'allumage de l'appareil, le traditionnel message de copyright est remplacé par une indication du mode clavier sous lequel l'ordinateur est placé (ceci, afin d'éviter des fautes de frappe dues à une mauvaise estimation du mode clavier).

La sauvegarde et le chargement

Un inconvénient du Basic du Canon X 07 est son absence d'instructions autorisant le chargement et la sauvegarde sur cassette du contenu d'une zone mémoire. Pour pallier ce défaut, il est possible d'utiliser indifféremment les options « S » et « L » du moniteur-désassemblleur paru dans le numéro 42 de *Micro-Systèmes*; pour ceux qui ne l'auraient pas entré se reporter aux programmes des figures 5a et 5b. Les possesseurs de l'assembleur pourront également utiliser le listing source de la figure 2 comme support de sauvegarde (ce qui nécessitera toutefois la présence de l'assembleur en mémoire lors de chaque chargement du programme).

Le programme

Les passionnés de programmation en assembleur qui désireront se plonger dans le logiciel pourront se référer au listing source. Pour conférer à ce programme une certaine compacité, il a été fait usage de plusieurs routines système. Nous ne nous étendrons pas ici sur leur rôle mais nous avons, pour une compréhension plus aisée, inséré de nombreuses remarques dans le listing source. Il est également important de souligner que ce programme ne constitue en aucun cas une unité fonctionnelle. Il doit, au contraire, être considéré, du fait même de son interactivité, comme intégré au système, c'est-à-dire dépendant du logiciel de base et de la structure d'entrées/sorties du microprocesseur. Il découle de cette exploitation des caractéristiques propres au Canon X 07; ce logiciel n'est pas adaptable à un autre micro-ordinateur, même architecturé autour d'un microprocesseur Z 80 ou compatible.

Remarque

Le passage Azerty/Qwerty modifie la configuration clavier du Canon X 07. Ce faisant, les touches de fonction risquent de voir aussi leur rôle modifié. En

```
10000 REM ****
10010 REM ***** CLAVIER AZERTY ****
10020 REM ***** POUR Canon X07 ****
10030 REM ** (c) EMMANUEL SANDER 1985 **
10040 REM ****
10050 , [ ]
10060 , JP #YY
10070 /*SELECTION DES REGISTRES SECONDAIRES
10080 /*JC EXX
10090 , EX AF.AF'
10100 /*POSITIONNEMENT D'UN NOUVEAU MODE ?
10110 , IN A.($F1)
10120 , CP $10
10130 , JR NZ.#YB
10140 , LD A.($170)
10150 , XOR &1
10160 , LD ($170).A
10170 , JP $C79B
10180 /*MODE AZERTY SELECTIONNE ?
10190 /*YB LD A.($170)
10200 , AND A
10210 /*NON : SAUT A LA ROUTINE PRINCIPALE
10220 , JP Z,$C79B
10230 /* OUI : POURSUITE DU DÉROULEMENT
10240 , IN A.($F1)
10250 /* ETAIT-CE UN 0 ?
10260 , CP $30
10270 , JR NZ.#JO
10280 , LD E.&124
```

Fig. 2. – Listing source du programme d'adaptation du clavier.

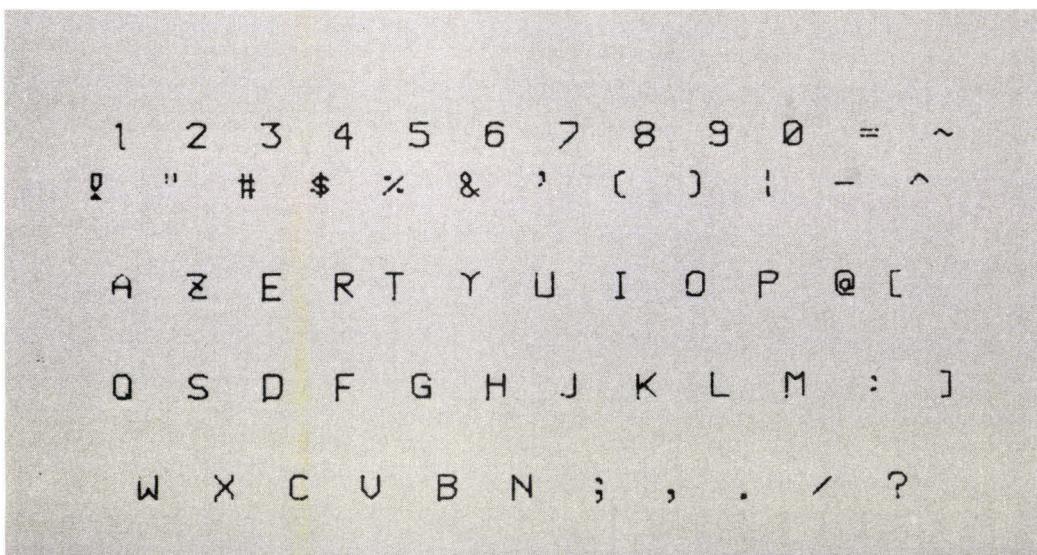


Fig. 1. – Le clavier du Canon X 07 tel que le configurer notre utilitaire.

effet, au niveau interne, l'appui sur une touche de fonction a une conséquence exactement similaire à celle que produirait l'appui successif sur chacune des touches correspondant au message affecté. Ainsi CLOAD programmé en mode Qwerty deviendra CLOAD en mode Azerty. Cette caractéristique n'est en fait nullement gênante; il suffit, pour en prévenir les inconvénients, de programmer toutes les touches de fonction dans un même mode et de revenir à ce mode (par un simple appui sur CTRL-P) lors de l'utilisation de l'une d'entre elles. ■

PROGRAMME

UTILITAIRE

```

10290 , JP #ES
10300 , *TRAITEMENT DU ;
10310 , #JO CP &43
10320 , JR NZ.#IL
10330 , LD E.&109
10340 , JP #ES
10350 , #IL CP &59
10360 , JR NZ.#MJ
10370 , LD E.&77
10380 , JP #ES
10390 , *TRAITEMENT DE LA ZONE NUMERIQUE
10400 , #MJ SUB $31
10410 , JR C.#AR
10420 , CP $9
10430 , JR NC.#FM
10440 , ADD A.$21
10450 , LD E.A
10460 , JP #ES
10470 , #FM IN A.($F1)
10480 , * TEST SI I A ETE PRESSE
10490 , CP &124
10500 , JR NZ.#PP
10510 , LD E.$30
10520 , JP #ES
10530 , #AR IN A.($F1)
10540 , SUB $21
10550 , JP C.$C79B
10560 , CP $9
10570 , JP NC.$C79B
10580 , ADD A.$31
10590 , LD E.A
10600 , JP #ES
10610 , * INVERSION DES TOUCHES DESIREES
10620 , #PP IN A.($F1)
10630 , CP &77
10640 , JR NZ.#NJ
10650 , LD E.&59
10660 , JP #ES
10670 , #NJ CP &109
10680 , JR NZ.#LM
10690 , LD E.&43
10700 , JP #ES
10710 , * MEMORISATION MAJUSCULE-MINUSCULE
10720 , #LM AND $20
10730 , LD C.A
10740 , IN A.($F1)
10750 , *MISE A ZERO DU BIT 5
10760 , AND $DF
10770 , * TOUCHES Q,W,A,Z
10780 , *TRAITEMENT DE LA TOUCHE Q

```

```

10790 , CP $51
10800 , JR NZ.#XL
10810 , SUB $10
10820 , OR C
10830 , LD E.A
10840 , JP #ES
10850 , *TRAITEMENT DE LA TOUCHE A
10860 , #XL CP $41
10870 , JR NZ.#NW
10880 , ADD A.$10
10890 , OR C
10900 , LD E.A
10910 , JP #ES
10920 , *TRAITEMENT DE LA TOUCHE Z
10930 , #NW CP $5A
10940 , JR NZ.#JB
10950 , SUB $3
10960 , OR C
10970 , LD E.A
10980 , JP #ES
10990 , * TRAITEMENT DE LA TOUCHE W
11000 , #JB CP $57
11010 , JP NZ.#LD
11020 , LD A.$5A
11030 , XOR C
11040 , LD E.A
11050 , JP #ES
11060 , #LD OR C
11070 , JP $C79B
11080 , #ES LD A.&1
11090 , OUT ($F5).A
11100 , DEC A
11110 , CALL $C262
11120 , XOR A
11130 , CALL $C2AA
11140 , *RETOUR DE LA ROUTINE DE TRAITEME
NT
11150 , JP $C8C1
11160 , #YY XOR A
11170 , LD ($170).A
11180 , LD HL.#JC
11190 , LD ($3D).HL
11200 , LD HL.#DS
11210 , LD ($6A).HL
11220 , RET
11230 , #DS LD HL.#AE
11240 , LD ($0045).HL
11250 , JR #ML
11260 , #AE DEFW #Z-&8
11270 , #Z DEFW $E428

```

Fig. 2. – (Suite.)

PROGRAMME

UTILITAIRE

```

11280 '#ML RST 38
11290 , DEFB $B8
11300 , RST 38
11310 , DEFB $AB
11320 , XOR A
11330 , OUT ($F4).A
11340 /* EFFACEMENT DE L'ECRAN
11350 , LD A.&12
11360 , CALL $C1BE
11370 /*IMPRESSION DES MESSAGES DE MISE
Sous TENSION
11380 , LD HL.#EJ
11390 , CALL $FEF7
11400 /*TEST DE MODE CLAVIER ACTUEL
11410 , LD A.($170)
11420 , AND A
11430 , JR Z.#CJ
11440 , LD HL.#A2
11450 , CALL $FEF7
11460 , JR #LP
11470 /*CJ LD HL.#QW
11480 , CALL $FEF7
11490 /*LP IN A.($F1)
11500 , SUB $5
11510 , JR NZ,#LP
11520 , IN A.($F0)
11530 , AND $80
11540 , JR Z.#LP
11550 , LD A.$0D
11560 , RST 28
11570 , LD A.$0A
11580 , RST 28
11590 , RST 38
11600 , DEFB $AC
11610 , RST 38
11620 , DEFB $B9
11630 , CALL $C0BD
11640 , XOR A
11650 , LD ($002B).A
11660 , LD HL.$E80C
11670 , LD ($0045).HL
11680 , JP $F23D
11690 /*EJDEFM Copyright(c) 1985 by EM
MANUEL SANDER MODE
11700 , DEFB $0
11710 /*QWDEFM QWERTY
11720 , DEFB $0
11730 /*AZDEFM AZERTY
11740 , DEFB $0
11750 , ]

```

Fig. 2. – (Suite et fin).

1C00	C3	C0	1C	D9	08	DB	F1	FE	:	1354
1C08	10	20	0B	3A	70	01	EE	01	:	469
1C10	32	70	01	C3	9B	C7	3A	70	:	882
1C18	01	A7	CA	9B	C7	DB	F1	FE	:	1438
1C20	30	20	05	1E	7C	C3	B1	1C	:	639
1C28	FE	2B	20	05	1E	6D	C3	B1	:	845
1C30	1C	FE	3B	20	05	1E	4D	C3	:	680
1C38	B1	1C	D6	31	38	15	FE	09	:	808
1C40	30	06	C6	21	5F	C3	B1	1C	:	780
1C48	DB	F1	FE	7C	20	17	1E	30	:	971
1C50	C3	B1	1C	DB	F1	D6	21	DA	:	1325
1C58	9B	C7	FE	09	D2	9B	C7	C6	:	1379
1C60	31	5F	C3	B1	1C	DB	F1	FE	:	1258
1C68	4D	20	05	1E	3B	C3	B1	1C	:	603
1C70	FE	6D	20	05	1E	2B	C3	B1	:	845
1C78	1C	E6	20	4F	DB	F1	E6	DF	:	1282
1C80	FE	51	20	07	D6	10	B1	5F	:	876
1C88	C3	B1	1C	FE	41	20	07	C6	:	956
1C90	10	B1	5F	C3	B1	1C	FE	5A	:	1032
1C98	20	07	D6	03	B1	5F	C3	B1	:	900
1CA0	1C	FE	57	C2	AD	1C	3E	5A	:	916
1CA8	A9	5F	C3	B1	1C	B1	C3	9B	:	1191
1CB0	C7	3E	01	D3	F5	3D	CD	62	:	1082
1CB8	C2	AF	CD	AA	C2	C3	C1	C8	:	1526
1CC0	AF	32	70	01	21	03	1C	22	:	436
1CC8	3D	00	21	D1	1C	22	6A	00	:	471
1CD0	C9	21	D9	1C	22	45	00	18	:	606
1CD8	04	D3	1C	28	E4	FF	B8	FF	:	1205
1CE0	AB	AF	D3	F4	3E	0C	CD	BE	:	1270
1CE8	C1	21	29	1D	CD	F7	FE	3A	:	1060
1CF0	70	01	A7	28	08	21	5F	1D	:	485
1CF8	CD	F7	FE	18	06	21	56	1D	:	884
1D00	CD	F7	FE	DB	F1	D6	05	20	:	1417
1D08	FA	DB	F0	E6	80	28	F4	3E	:	1413
1D10	0D	EF	3E	0A	EF	FF	AC	FF	:	1245
1D18	B9	CD	BD	C0	AF	32	2B	00	:	1039
1D20	21	0C	E8	22	45	00	C3	3D	:	636
1D28	F2	43	6F	70	79	72	69	67	:	975
1D30	68	74	28	63	29	20	31	39	:	538
1D38	38	35	20	62	79	20	20	20	:	456
1D40	45	4D	4D	41	4E	55	45	4C	:	596
1D48	20	53	41	4E	44	45	52	20	:	509
1D50	20	4D	4F	44	45	00	20	20	:	389
1D58	51	57	45	52	54	59	00	20	:	524
1D60	20	41	5A	45	52	54	59	00	:	511

FIGURE 3:LISTE HEXADECIMALE AVEC SOMME D'E CONTROLE

Fig. 3. – Liste hexadécimale du programme.

PROGRAMME

UTILITAIRE

```

20000 REM **** CHARGEUR HEXADECIMAL ****
21000 CLS
22000 X=&H1C00
23000 PRINT HEX$(X); " ";
24000 INPUTA$*
25000 IFLEN(A$)<>16THENCLS:BEEPS,10:GOTO
23000
26000 FORI=0TO7
27000 A=VAL("&H"+MID$(A$,2*I+1,2))
28000 S=S+A
29000 POKEX+I,A
30000 NEXTI
31000 INPUT"SOMME ";R
32000 A$=""
33000 IFR<>STHENS=0:BEEPS,10:CLS:GOTO230
00
34000 X=X+8:S=0:IFX>&H1D67THENPRINT"FIN
DE SAISIE":END
35000 CLS
36000 GOTO23000

```

Fig. 4. – Le programme Basic de chargement de l'utilitaire (lorsqu'on n'utilise pas d'assembleur).

a)

```

40000 REM ***** SAUVEGARDE *****
41000 INIT#1,"CASO:"
42000 PRINT#1,"AZERTY"
43000 FORI=0TO200
44000 NEXT
45000 FORI=&H1C00TO&H1D67
46000 OUT#1,PEEK(I)
47000 NEXT
48000 PRINT"FIN DE SAUVEGARDE"
49000 INPUT"EN DESIREZ-VOUS UNE AUTRE";A
$
50000 IFLEFT$(A$,1)<>"O"THENEND
51000 RUN42000
b)

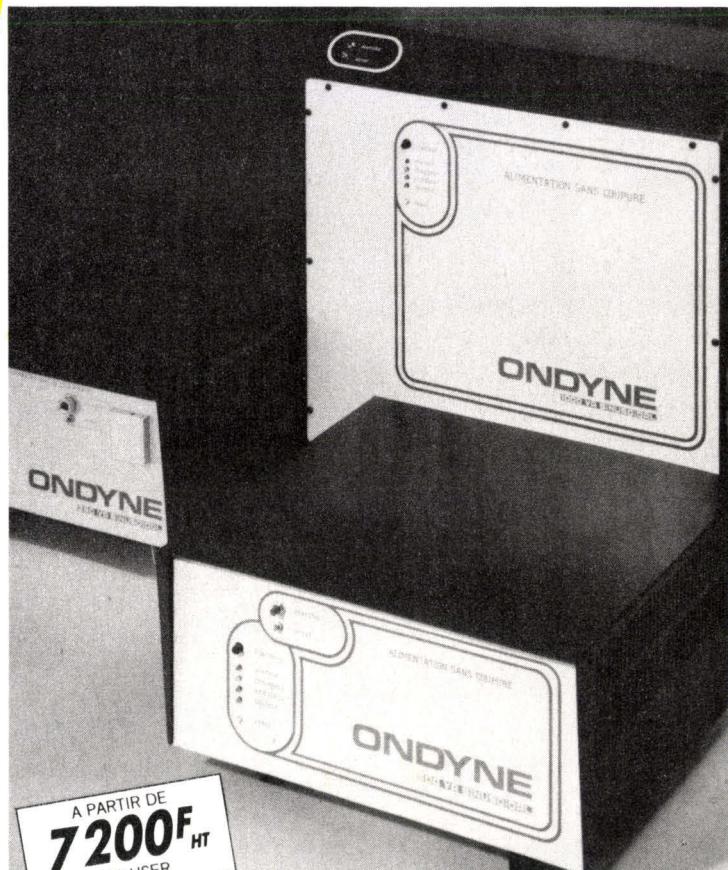
```

```

55000 REM ***** CHARGEMENT *****
56000 INIT#1,"CASI:"
57000 INPUT#1,A$*
58000 IFA$<>"AZERTY"THENGOTOS7000
59000 FORI=&H1BFFTO&H1D67
60000 POKEI,INP(#1)
61000 NEXT
62000 PRINT"FIN DE CHARGEMENT"
63000 END

```

Fig. 5. – a) Le programme de sauvegarde de l'utilitaire compilé. b) Le programme de chargement de l'utilitaire compilé.



AU SECOURS!

**OFFREZ A VOTRE ORDINATEUR
UNE ALIMENTATION SANS COUPURE**

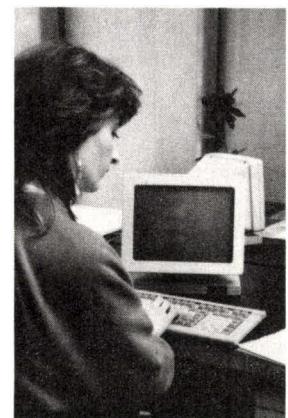
Votre installation informatique est vulnérable.

La moindre défaillance du secteur peut provoquer la destruction de vos fichiers et programmes, voir de votre ordinateur, en cas de surtension.

Les «ONDYNE» sont des alimentations de secours sans commutation, qui produisent leur propre courant, pour protéger votre ordinateur contre les microcoupures et les pannes secteur d'une durée de 30 mn en moyenne.

Les «ONDYNES» sont des unités compactes qui sous un faible volume comprennent : un chargeur régulé, un onduleur à haut rendement, des batteries étanches sans entretien, un module d'alarme. Un simple branchement sur le secteur les rend immédiatement opérationnels.

Gage de sécurité : les tests réalisés auprès d'IBM PC, XT et AT, toute la gamme THOMSON MICROMEGA, APPLE, BULL MICRAL, CANON, RAIR, GOUPIL, OLIVETTI, TANDY, COMMODORE, LEANORD etc.



ONDYNE

LE COMPAGNIE VIGILANT DE VOTRE ORDINATEUR
FRANCE ONDULEURS ONDYNE
8, RUE DE LA MARE 91630
AVRAINVILLE TÉLEX : 690804
TÉL: 082.06.54

NOUVEAUX DISTRIBUTEURS FRANCE ET ÉTRANGER RECHERCHÉS

9900 F TTC !

Le compatible IBM PC-XT
(avec super-bios)

OFFRE LIMITÉE

UNITÉ CENTRALE « DYNAMIT-16XT » COMPATIBLE PC-XT®
CARTE MÈRE (8 SLOTS) AVEC 256 K RESIDENTS
ALIMENTATION 130 W
2 DRIVES JAPONAIS 360 K
CLAVIER AZERTY
CARTE MONOCHROME GRAPHIQUE IMPRIMANTE
(NO-FLICKERING)
CARTE DRIVE

(GARANTIE 1 AN)

Interfaces 16 BITS

Imprimante parallèle	450 F
Monochr. Graph. imprim	1900 F
RS 232C	790 F
Mémoire 512 Ko (OK)	1250 F
Couleur graphique	1800 F
Multi-fonction 384 Ko (OK)	2900 F
Carte transfer Apple-PC	

Light PEN
RESEAUX LOCAUX
LIAISON 3270 IBM

7500 F HT

DISQUE DUR 10 MEG POUR IBM
AVEC CONTRÔLEUR XEBEC !

INTERFACES POUR APPLE

128 K RAM 900 F

80 COL. + 64 K

Z-80

IMPRIM + 64 K

WILCARD

SPEECHCARD

MOQUINBORD

MODEM V 21

CUVRES NUS !

PÉRIPHÉRIQUES

PAVÉ NUM. 2^e

JOYPORT

IMPRIMANTES

MONITEURS

DISQUETTES

DRIVES

JOYSTICKS

TABLETTE GRAPHIQUE

Guerre des prix ?



monte au front !

**CERTAINS DE NOS PRIX
SONT TROP BAS POUR
ÊTRE PUBLIÉS. NOUS
BATTONS N'IMPORTE
QUEL PRIX !**

ÉCRIVEZ-NOUS POUR UNE LISTE COMPLÈTE DE NOS ARTICLES

Pour 16 bits, 8 bits 2^e, MODEM COMMODORE, ATARI

DERNIERE MINUTE :

PROMO RENTRÉE DISQUETTES 5" 1/4 SF/DD 74,50 /10

54, rue de Dunkerque - 75009 PARIS.

Tél. : 282.17.09

**DYNAMIT
COMPUTER**

PC-XT marque déposée d'IBM

PROGRAMME UTILITAIRE

Qui n'a jamais éprouvé le désir de téléphoner à un parent, un ami en composant directement son nom à la place de chiffres anonymes ? Le logiciel que nous vous présentons pour Commodore, associé à un modem Digitelec, rend cette chose désormais possible en composant les numéros de fiches de votre répertoire.

de Olivier DUVERNEUIL
Ordinateur :
Commodore 64 : + Modem Digitelec
Langage :
Basic

Le modem Digitelec, grâce à son type de connexion sur l'ordinateur et sur la ligne téléphonique, dispose d'un potentiel supérieur à celui des modems classiques. Outre la réponse automatique, il autorise l'autocomposition, qui nous donnera la possibilité d'appeler les numéros des fiches sélectionnées. Les programmeurs qui se sont intéressés au problème de téléphonie savent que la composition d'un numéro est une suite d'impulsions, c'est-à-dire une succession de coupures de lignes. Il faudra donc décrocher puis raccrocher le combiné une somme de petites fois pour générer des impulsions de 66 ms toutes les 100 ms.

Dans un premier temps, avant de composer un numéro de téléphone, l'utilisateur devra créer les fiches de ses futurs interlocuteurs, avec la possibilité d'interroger son fichier pour une recherche nominative ou thématique (la recherche d'une sous-chaine de caractères quelconque contenue dans celui-ci) et de le mettre à jour.

De plus, un tel programme ne peut être conçu qu'avec la possibilité de stocker ce fichier sur un disque ou éventuellement une cassette en modifiant les numéros correspondant aux périphériques aux lignes 180 et 870.

Au niveau téléphonique, pour la composition du numéro fichier, le programme commence par fermer la ligne 4 secondes,

UN AUTOCOMPOSEUR TELEPHONIQUE

après s'être assuré que celle-ci a été ouverte au moins les 4 secondes ($240 \times 1/60^{\circ}$) précédentes pour obtenir la tonalité (ligne 2150).

Pour contrôler le temps nécessaire entre deux ruptures de ligne, le sous-programme ligne 2370 vérifie que le temps nécessaire s'est bien écoulé depuis la dernière rupture de ligne (ouverture ou fermeture).

Alors, lorsque le temps nécessaire est écoulé, le programme passe à la numérotation proprement dite (ligne 1070), en lisant chiffre par chiffre le numéro affiché dans la mémoire écran de l'adresse 1784 à 1824 et en composant également chiffre par chiffre avec un délai d'une seconde entre chacun d'eux.

Le sous-programme ligne 2290 génère un nombre d'impulsions correspondant au chiffre à composer, sauf pour le zéro auquel 10 impulsions sont nécessaires.

Selon un principe similaire, la numérotation peut également se faire manuellement, c'est l'objet du sous-programme de la ligne 2010 à 2080.

Mode d'emploi

Après avoir entré le programme représentant environ 6 Ko, on peut passer soit à son application, soit à la correction des erreurs de frappe éventuelles qui, du fait de l'absence de langage machine, ne devraient pas poser de problèmes.

Cette mise au point effectuée, nous pouvons regarder comment utiliser ce programme.

A son lancement, celui-ci commence par lire le fichier sur le disque puis présente un premier menu.



● Ordre du menu fichier

Il est à noter ici que le programme peut très bien fonctionner sans le modem si l'on se contente de l'utiliser en tant que fichier.

F1 ← recherche à partir du nom

Autorise la recherche à partir d'un nom ou des premières lettres le constituant.

Si plusieurs noms conviennent, ils seront affichés à la suite les uns des autres à moins que l'on appuie simultanément sur « SHIFT » et « RETURN ».

F2 ← recherche thématique

Pour rechercher une sous-chaine contenue dans le fichier.

Le programme ne pouvant exécuter à nouveau les ordres du menu qu'après avoir testé tout le fichier, il est à conseiller d'utiliser « SHIFT » et « RETURN » dès que le renseignement recherché est trouvé, sinon n'importe quelle touche permettra de poursuivre.

F3 ← entrée d'un nouveau nom

Pour incorporer un nouveau nom au fichier.

Il est à noter que les noms ainsi entrés dans la mémoire de l'ordinateur ne figureront sur la disquette qu'après avoir sauvegardé le fichier (fonction F7).

F4 ← suppression d'un nom

Pour effacer le nom figurant à l'écran de la mémoire de l'ordinateur, il faut successivement appuyer sur la touche « F4 » puis sur « CLR ».

F5 ← téléphone

Passe du menu relatif du fichier à celui de l'utilisation des possibilités d'auto-numérotation.

F6 ← correction

Cette fonction autorise la correction des renseignements figurant à l'écran. Chacun d'eux étant proposé à l'utilisateur, s'il n'y a pas lieu de le corriger, presser « RETURN » pour passer au renseignement suivant.

F7 ← sauvegarde

Remplace l'ancien fichier du disque par le contenu de la mémoire de l'ordinateur.

F8 ← impression du listing

Copie le contenu du fichier sur l'imprimante.

● Menu téléphone

Avant d'employer les fonctions d'auto composition, il faut savoir que ce programme ne peut fonctionner que si votre téléphone est raccroché pendant que l'ordinateur le numérote.

F1 ← composition du numéro de fiche

Cette fonction compose le numéro fiche apparaissant à l'écran.

F3 ← recomposition du dernier numéro

Rappelle le dernier correspondant.

F5 ← composition manuelle

Composition d'un numéro quelconque puis éventuellement sa recomposition.

F7 ← retour au menu

Libère la ligne et ramène au menu principal.

PROGRAMME

UTILITAIRE

```
100 GOT0120
110 SAVV"0:REP & TEL",B:END
120 REM **** REPERTOIRE C.64 ****
130 REM *** (CD) O DUVERNEUIL ***
140 REM ***** ****
150 REM ***** ****
160 REM
170 CLR:GH=2:DIMH$(500):OPEN1,B,0,"":INPUT#1,C
180 FORQ=1TOC:INPUT#1,I$:INPUT#1,O$:IFQ$=""."THENI$=I$+O$
190 R$(Q)=I$:NEXT:CLOSE1
200 PRINT"0":GOSUB210:GOT0410
210 EC=1:GOSUB220:GOT0340
220 PRINT"0" REPERTOIRE D'ADRESSE ET TEL"
230 PRINT"
240 PRINT" F1 ← RECHERCHE A PARTIR DU NOM "
250 PRINT" F2 ← RECHERCHE THEMATIQUE "
260 PRINT" F3 ← ENTREE D UN NOUVEAU NOM "
270 PRINT" F4 ← SUPPRESSION D'UN NOM "
280 PRINT" F5 ← TELEPHONE "
290 PRINT" F6 ← CORRECTION "
300 PRINT" F7 ← SAUVEGARDE "
310 PRINT" F8 ← IMPRESSION DU LISTING "
320 PRINT"
330 RETURN
340 PRINT" NOM: PRENOMS: "
350 PRINT" ADRESSE: "
360 PRINT" VILLE: CP: "
370 PRINT" TELEPHONE: "
380 PRINT" AUTRES RENSEIGNEMENTS: "
390 PRINT"
400 PRINT" ":RETURN
410 GETR$:
420 IFR$="0" THEN560
430 IFR$="1" THEN950
440 IFR$="2" THEN740
450 IFR$="3" THEN1110
460 IFR$="4" THEN1170
470 IFR$="5" THEN1350
480 IFR$="6" THEN870
490 IFR$="7" THEN1610
500 IFR$="8" THENINPUT"0 NOM DU PROGRAMME A CHARGER "IN$:LOADIN$,8
510 GH=GH:IF4<-2)1GHTHENPRINT"0":GOT0410
520 PRINT"0":GOT0410
530 REM
540 REM RECHERCHE A PARTIR DU NOM
550 REM **** **** **** **** ****
560 GOSUB1590
570 GOSUB1470
580 IFN0$<>MID$(R$(W+1),1,E)THEN:WM=W:GOTO 410
590 GETR$:IFR$="" THEN590
600 IFW=CTHEN:WM=W:GOT0410
```

PROGRAMME

UTILITAIRE

```

610 IF A$="I*" THEN : WM=W : GOT0410
620 IF A$="M*" THEN : WM=W : GOT01110
630 IF A$="B*" THEN : WM=W : GOT01610
640 IF A$="P*" THEN : WM=W : GOT01170
650 IF A$=CHR$(141) THEN : WM=W : GOT0410
660 W=W+1 : GOT0570
670 FOR L=1 TO LEN(A$(W))
680 IF MID$(A$(W),L,1)="." THEN X$=MID$(A$(W),D,L-D) : D=L+1 : GOT0 700
690 NEXT : X$=MID$(A$(W),D)
700 RETURN
710 REM

720 REM ENTREE D UN NOUVEAU NOM
730 REM ****
740 GOSUB210 : GOSUB1440
750 X$=NO$+"." : PRINT "*****RENOMS" : INPUT NO$ : X$=X$+NO$+"."
760 INPUT "?" : NO$ : X$=X$+NO$+"."
770 INPUT "?"
780 PRINT "*****TCP" : INPUT NO$ : X$=X$+NO$+"."
790 INPUT "?" : NO$ : X$=X$+NO$+"." : EC=1
800 PRINT : INPUT NO$ : X$=X$+NO$
810 FOR QE=1 TO C : IF A$(QE)>X$ THEN 830
820 NEXT QE : C=C+1 : A$(C)=X$ : GOT0410
830 FOR WE=C TO QESTEP-1 : A$(WE+1)=A$(WE) : NEXT WE : A$(QE)=X$ : C=C+1 : GOT0410
840 REM

850 REM SAUVEGARDE
860 REM ****
870 OPEN1,8,15 : PRINT#1,"S:." : CLOSE1 : OPEN1,8,1,"." : PRINT#1,C
880 FOR Q=1 TO C : O$=".": IF LEN(A$(Q))<=88 THEN I$=A$(Q) : GOT0900
890 I$=MID$(A$(Q),1,88) : O$=MID$(A$(Q),89)
900 PRINT#1,I$ : PRINT#1,O$ : NEXT
910 CLOSE1 : GOT0410
920 REM
930 REM RECHERCHE THEMATIQUE
940 REM ****
950 TY=0 : GOSUB210 : PRINT "?" : CR$="" : INPUT CR$ : E=LEN(CR$)
960 FOR W=1 TO C
970 FOR Q=1 TO LEN(A$(W))-E+1
980 IF CR$=MID$(A$(W),Q,E) THEN GOSUB1020 : IF 1 THEN 1000
990 NEXT Q
1000 NEXT W
1010 GOT0410
1020 IF TY THEN GOSUB 1540
1030 IF A$=CHR$(141) THEN : GOT0410
1040 IF A$="." THEN : GOT01110
1050 IF A$="B*" THEN : GOT01610
1060 IF A$="P*" THEN : GOT01170
1070 GOSUB1470 : WM=W : TY=1 : RETURN
1080 REM
1090 REM SUPPRESSION D'UN NOM
1100 REM ****
1110 IF E THEN 410

```

PROGRAME

U T I L I T A I R E

Listing du programme (suite).

PROGRAMME UTILITAIRE

Listing du programme (suite).

PROGRAMME

UTILITAIRE

```
2170 POKE57080,176
2180 TEL$=""
2190 RETURN
2200 REM
2210 REM RACROCHE
2220 REM *****
2230 TD=240:T0=TI:RD=0:GOSUB2100
2240 POKE57080,180
2250 RETURN
2260 REM
2270 REM COMPOSE
2280 REM *****
2290 GOSUB2370:TEL$=TEL$+CHR$(TN+48)
2300 FORII=1TOTN-10*(TN=0)
2310 POKE57080,180:FORJ=1TO44:NEXTJ
2320 POKE57080,176:FORJ=1TO22:NEXTJ
2330 NEXTII
2340 PRINTCHR$(48+TN);
2350 TD=60:T0=TI
2360 RETURN
2370 IF TD>TI-T0THEN2370
2380 RETURN
```

READY.

Listing du programme (suite et fin).

NOM:ATARI PRENOMS: .
ADRESSE:9 R. GEORGES-ENESCO
VILLE:CRETEIL CP:94 008
TELEPHONE:377 12 63
AUTRES RENSEIGNEMENTS: .

NOM:CAMPUS PRENOMS:LYNX
ADRESSE:140 B. HAUSSMANN
VILLE:PARIS CP:75 008
TELEPHONE:562 03 08
AUTRES RENSEIGNEMENTS: .

NOM:DEF PRENOMS: .
ADRESSE:Z.I. ST. MITRE
VILLE:AUBAGNE CP:13 400
TELEPHONE:(42) 03 05 45
AUTRES RENSEIGNEMENTS: .

NOM:DOR COMP. PRENOMS:DRAGON
ADRESSE:18 RUE DE ST. QUENTIN
VILLE:PARIS CP:75 010
TELEPHONE:208 67 71
AUTRES RENSEIGNEMENTS: .

NOM:IDEAL LOISIRS PRENOMS:ADAM
ADRESSE:122 AV. DE LA PLAINE DE FRANCE
VILLE:ROISSY C.D.G. CP:95 945
TELEPHONE:665 44 88
AUTRES RENSEIGNEMENTS: .

NOM:LOGI'STICK PRENOMS: .
ADRESSE:LE BONAPARTE
VILLE:LE BLANC MEGNIL CP:93 163
TELEPHONE:867 28 44
AUTRES RENSEIGNEMENTS:G PROBST 865 44 55

NOM:MICRO PRENOMS:SYSTEMES
ADRESSE:2 A 12 RUE DE BELLEVUE
VILLE:PARIS CEDEX 19 CP:75 940
TELEPHONE:200 03 05
AUTRES RENSEIGNEMENTS: .

NOM:PROCEP PRENOMS:COMMODORE
ADRESSE:5-9 R. SENTOU
VILLE:SURENES CP:92 160
TELEPHONE:606 41 41
AUTRES RENSEIGNEMENTS: .

NOM:SBM PRENOMS:SHARP
ADRESSE:151-153 AV. JEAN JAURES
VILLE:AUBERVILLIERS CP:93 307
TELEPHONE:834 93 44
AUTRES RENSEIGNEMENTS: .

NOM:STERCO I. PRENOMS:BBC
ADRESSE:R. DU BASSIN 2 .LOT 6
VILLE:GENNEVILLIERS CP:92 230
TELEPHONE:742 58 20
AUTRES RENSEIGNEMENTS: .

Exemple de fiche sortie sur imprimante.