

abc

N° 52

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Ordinateurs de poche

Réussir un logiciel

Menus et commandes

Programmes en assembleur

EDITIONS
ATLAS



Dans la poche

Les ordinateurs de poche ont des emplois multiples. Les ingénieurs s'en servent sur les chantiers. Les spéculateurs en font usage à la Bourse. Voici plusieurs machines fabriquées par Casio.

Les calculatrices programmables peuvent être mises en œuvre pour des applications similaires; mais l'avantage des ordinateurs de poche est qu'ils ont recours au BASIC et permettent la manipulation de textes, alors que les calculatrices ne peuvent travailler que sur des chiffres et qu'elles doivent, de surcroît, être programmées dans un langage qui leur est particulier et qui s'apparente au code machine.

Les ordinateurs de poche se révèlent sans égal lorsqu'il s'agit d'effectuer des calculs à l'aide d'une formule donnée sans cesse reprise. De telles opérations sont répétitives et fastidieuses; mais il suffit alors à l'utilisateur de rédiger un programme approprié.

Le FX-720P est le moins cher de la gamme d'ordinateurs de poche Casio. Il ne pèse que 210 g, pour des dimensions de 165 x 85 x 15 mm. Le clavier, qui représente environ un demi-clavier normal, est de disposition QWERTY standard; mais les rangées de touches sont alignées, et non disposées en quinconce, comme c'est le cas généralement.

Chaque touche alphabétique peut produire deux autres caractères (signes de ponctuation, par exemple) ou des mots du vocabulaire BASIC dès lors qu'elle est employée en conjonction avec les touches de fonction ou SHIFT. Toutefois, celles-ci ne sont pas pressées en même temps, mais enfoncées puis relâchées avant usage de la touche désirée.

Il est bien plus facile ainsi de manipuler l'appareil d'une main, avantage extrêmement précieux lorsqu'on travaille loin d'un bureau.

Limiter les dépenses d'énergie

Le FX-720P dispose d'un affichage à cristaux liquides (une ligne de douze caractères); deux touches curseur permettent le déplacement vers la gauche ou vers la droite et, par conséquent, l'affichage d'un texte plus long. Une tête molette permet de modifier le contraste afin de l'adapter à l'éclairage. De plus, l'ordinateur s'éteint de lui-même si, au bout de six minutes, il n'a pas été utilisé; sage précaution pour limiter les dépenses d'énergie.

L'appareil ne dispose que d'une mémoire de 2 K. Mais l'utilisateur se sert de cartouches enfichables, capables de conserver les données lorsqu'on les retire de l'ordinateur. Il peut donc stocker des programmes différents sur chacune d'elles et s'en

servir comme il l'entend. Ces cartouches sont de 2 ou de 4 K. Ce n'est pas beaucoup, et elles sont vite au bout de leurs possibilités, d'autant plus que le FX-720P peut maintenir simultanément en mémoire jusqu'à dix programmes BASIC! Chacun d'eux doit ainsi être entré selon un mode particulier et lancé selon un autre.

Le BASIC de l'appareil est, vu la taille de l'engin, tout à fait complet; il ne lui manque que les fonctions CHR\$ et ASC. La commande BEEP, très simple, permet la production de deux sons différents.

Le FX-720P représente le bas de la gamme Casio; les autres modèles ont donc des performances supérieures. Pourtant, il est le seul à disposer d'une petite base de données, « Data Bank », qui permet de conserver des informations du type adresses, noms, dépenses ou horaires de rendez-vous.

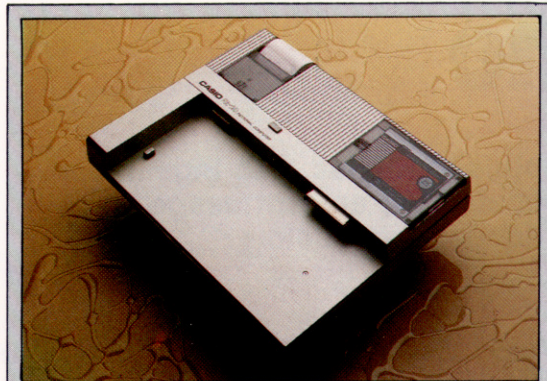
Le programme peut, à la demande, rechercher n'importe quel élément, et ce, à plusieurs niveaux: il lui est possible, par exemple, de rassembler plusieurs données, puis d'entreprendre de nouvelles recherches, parmi celles-ci, en fonction de critères supplémentaires. Le FX-720P est, de ce point de vue, un rival direct de l'Organiser de Psion, qui est pourtant plus cher et moins évolué.

Un modèle bon marché

L'ordinateur de poche FX-720P de Casio est doté d'une mémoire de 2 K, sous forme de cartouches enfichables, d'un affichage à cristaux liquides d'une ligne et d'un ensemble complet de touches alphabétiques et numériques. L'enfichage de cartouches supplémentaires permet d'accroître la capacité mémoire.

(Cl. Chris Stevens.)



**L'interface FA-20**

Chacun des ordinateurs de poche de la gamme Casio peut être fixé sur une interface de ce type. Le modèle présenté ici accueille ainsi le FX-750P. Le dispositif est doté de batteries rechargeables et d'un adaptateur secteur. (Cl. Chris Stevens.)

Le FX-750P et le PB-700

Le FX-720P est avant tout destiné à l'homme d'affaires; le FX-750P, lui, s'adresse plutôt au scientifique et à l'ingénieur. D'un prix légèrement supérieur, il dispose d'un clavier plus performant et d'un affichage de vingt-quatre caractères. Une cartouche de 2 K, très semblable à celle de son prédécesseur, lui fournit la mémoire dont il a besoin; mais il est possible d'en enficher une seconde, et la capacité mémoire peut donc atteindre 8 K en tout. L'appareil pèse 250 g pour une largeur de 185 mm. Cette fois, il faut utiliser les touches SHIFT et de fonction en même temps que la touche alphabétique dont on a besoin.

Dix de ses touches correspondent à des constantes fréquemment utilisées en physique : vitesse de la lumière, accélération gravitationnelle de la Terre, constantes de Planck, de Boltzmann et d'Avogadro, charge élémentaire, masse atomique, masse de l'électron, constante gravitationnelle, volume molaire. Ce sont des nombres constamment employés lors des calculs; ceux-ci en sont donc facilités d'autant. Par ailleurs, l'engin offre des fonctions généralement absentes du BASIC standard : six fonctions hyperboliques et un ensemble de fonctions statistiques. A dire vrai, le BASIC de l'appareil, amélioré en bien des endroits, ne serait pas déplacé sur un micro-ordinateur personnel.

Le PB-700, récemment lancé, représente le haut de la gamme Casio. Il peut afficher quatre lignes de vingt caractères et sa mémoire, de 4 K, peut être quadruplée grâce à des modules-mémoire enfichés sous l'appareil, qui ne pèse que 315 g pour une largeur de 20 cm. Il est le seul à être doté d'une sacoche protectrice.

Le clavier est semblable à celui du FX-750P à ceci près qu'il comporte une touche CAPS; la maintenir enfoncée permet d'obtenir les minuscules. Le BASIC est équipé de quelques commandes graphiques supplémentaires indispensables à la création de graphismes sur l'écran : la résolution de celui-ci est de $32 \times 1,60$.

Par ailleurs, ces graphismes peuvent être imprimés à l'aide de l'interface FA-10, qui se compose d'une petite imprimante-table traçante ainsi que d'une prise cassette. L'ordinateur est simplement fixé sur elle. L'imprimante peut tracer des lignes, des cercles, des graphiques à l'aide de quatre pointes de stylos à bille de couleur, sur un papier de 115 mm de large.

La prise cassette autorise la sauvegarde des programmes et des données, ce que la faible mémoire du PB-700 rend à peu près indispensable. Un minuscule lecteur de cassettes peut être mis en place à côté de l'imprimante, et l'ensemble reste assez petit pour être transporté partout. Une interface Centronics standard rend possible le raccord avec les imprimantes utilisées en informatique.

Le FX-720P et le FX-750P sont pourvus de dispositifs analogues. Le premier fait ainsi usage de l'imprimante FP-12S et d'une interface cassette séparée l'FA-3. Le FX-750P a recours au FA-20, qui combine les deux unités, dispose d'un emplacement destiné à une carte mémoire et se voit doté de batteries rechargeables, d'un adaptateur secteur et d'un boîtier protecteur.

Les trois ordinateurs sont accompagnés de manuels très complets, qui contiennent tous les renseignements nécessaires. Toutefois, la traduction laisse à désirer : elle semble avoir été faite directement à partir du japonais et comporte bien des passages obscurs ou incorrects. Le débutant s'y perdra parfois. Il sera d'autre part handicapé par le manque de logiciels et devra souvent écrire ceux qui lui sont nécessaires ou les recopier dans les manuels.

PRODUIT	PRIX	FORMAT	POIDS	MÉMOIRE	AFFICHAGE	REMARQUES
FX-720P	*	165 x 85 x 15 mm	210 g	2 K	1 x 12 car.	Base de données intégrée. Clavier numérique.
FX-750P	**	185 x 85 x 15 mm	250 g	2 k	1 x 24 car.	Prise pour second lecteur cartouche. Constantes physiques préprogrammées. BASIC étendu.
PB-700	**	200 x 85 x 15 mm	315 g	4 K	4 x 20 car.	Capacité graphique. Possibilité imprimante.



Tortue à jouer

Nos articles sur le LOGO ont été consacrés au mode immédiat. Le LOGO comporte un autre mode qui permet de définir une séquence de commandes en tant que procédure avec un nom spécifique.

Le programmeur de micro-ordinateur peut ne pas être très à l'aise avec la notion de procédure. Dans cet article, nous créons une procédure LOGO appelée «CARRÉ». Contrairement aux commandes LOGO, qui doivent toujours figurer en majuscules, les procédures LOGO peuvent indifféremment être en majuscules ou en minuscules. Nous commençons par taper :

```
ÉDIT CARRÉ
```

L'écran se vide, le message «PROCÉDURE CARRÉ» s'affiche en haut de l'écran, pour rappeler le nom de la procédure, et les messages «ÉDIT CTRL-C POUR DÉFINIR», «CTRL-G POUR SUPPRIMER» s'affichent au bas de l'écran. Ce double message est un aide-mémoire pour l'édition des procédures. ÉDIT signifie que LOGO est en mode édition et qu'il attend la définition d'une procédure. Une fois la définition saisie, « Contrôle-C » permet de valider la nouvelle procédure et « Contrôle-G » de la supprimer pour la recommencer.

En mode édition vous pouvez taper ce que vous voulez, mais aucune des commandes données ne sera exécutée alors.

LOGO prend note de vos instructions et les sauvegarde dans un répertoire sous le nom que vous avez donné à la procédure. Complétez la définition de la procédure en tapant les commandes suivantes :

```
PROCÉDURE CARRÉ
  RÉPÈTE 4 [AV 50 DR 90]
FIN
```

Vous pouvez maintenant demander à LOGO de définir cette procédure (de la répertorier) en tapant « Contrôle-C ». Vous recevrez le message « CARRÉ DÉFINI ». Si vous avez fait une erreur dans votre définition, vous pouvez faire « avorter » la définition par LOGO en tapant « Contrôle-G » (supprimer la définition en cours). Vous devrez alors la recommencer. Vous pouvez aussi, sans supprimer la définition, lui apporter les corrections nécessaires en utilisant le mode éditeur plein écran.

Ce dernier vous permet d'utiliser les touches de déplacement du curseur pour ajouter ou supprimer des caractères en tout point de la procédure. L'éditeur de ligne du mode immédiat, pour sa part, ne permet des corrections que sur la dernière ligne entrée.

Votre procédure est maintenant définie. Voyons comment s'en servir. Tapez TRACER afin d'accéder à l'écran graphique, puis CARRÉ : les commandes de la définition de la procédure sont exécutées et la tortue dessine le carré.

Les commandes d'une procédure sont alors utilisées exactement de la même manière que les commandes élémentaires du mode immédiat. Vous devez seulement taper le nom de la procédure pour que l'ensemble des commandes constitutives de sa définition soient exécutées. Les commandes originelles, présentes dès le chargement de LOGO, s'appellent des « primitives ». Une fois que vous avez appris au langage LOGO une procédure, cette dernière peut être utilisée en tant que primitive.

Si une procédure ne correspond pas exactement à ce que vous attendiez d'elle, vous pouvez en changer la définition. Nous avons défini CARRÉ pour une valeur horizontale de 50. Si vous vouliez un carré plus petit, de 30 unités horizontales par exemple, vous pourriez réintégrer le mode éditeur en tapant :

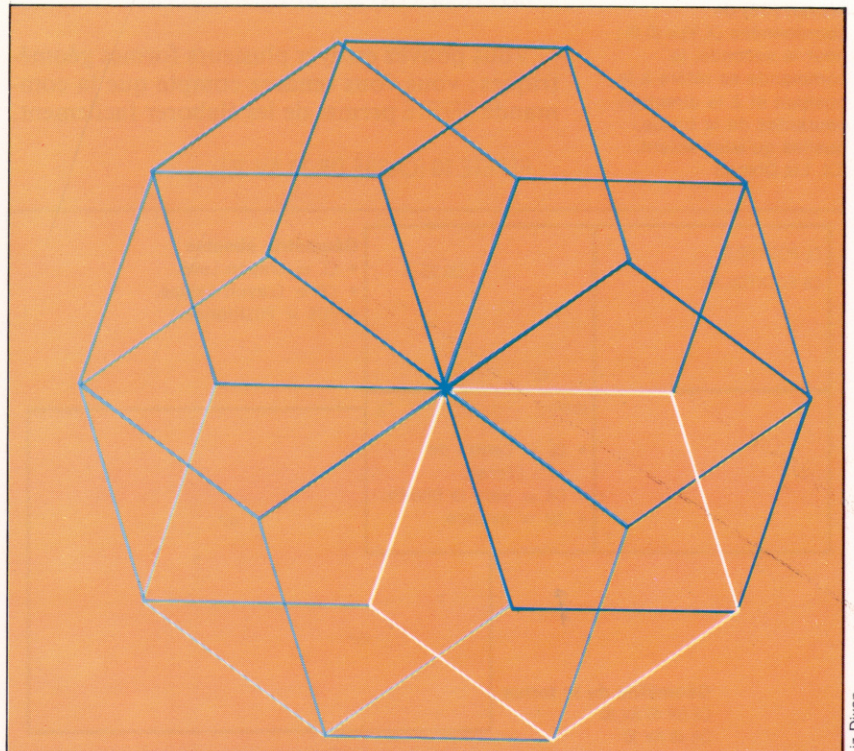
```
ÉDIT CARRÉ
```

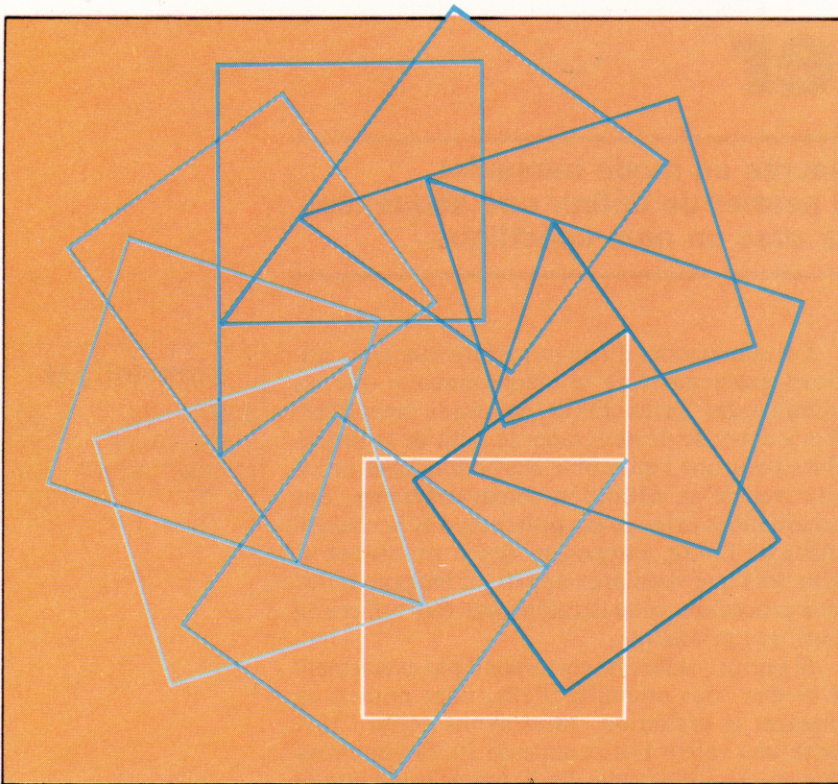
Le texte qui définit la procédure est alors affiché et l'éditeur plein écran peut être utilisé pour changer 50 en 30. Vous redéfinissez alors la nouvelle procédure par « Contrôle-C » et vous vérifiez, en tapant CARRÉ, que le nouveau carré est

Abréviations

ÉDIT	ED
CACHETORTUE	CT
MONTRETORTUE	MT

PENTAGONE





CARRÉ

Je pense, donc LOGO
 La tortue peut être pilotée selon la géométrie absolue ou, selon la géométrie de la tortue, relative. Pour la première, la position de la tortue est donnée par rapport à la position absolue du centre de l'écran (CE) de coordonnées (X,Y) = (0,0); pour la seconde, la commande se réfère à la position et à la direction courantes de la tortue, quelles qu'elles soient. (Cl. Liz Dixon.)

bien celui que vous voulez. Pour vous exercer, essayez donc d'écrire des procédures pour des triangles, des pentagones, des étoiles, etc.

Mettez en œuvre, maintenant, vos procédures avec la commande RÊPÊTE. Définissez par exemple PENT de la manière suivante :

```
RÊPÊTE 5[AV 50 DR 72];  
essayez ensuite :
```

```
RÊPÊTE 10 [PENT DR 36]
```

Vous pouvez essayer plusieurs formes géométriques; vous vous rendrez compte que la commande RÊPÊTE permet de les élaborer facilement.

```
RÊPÊTE 10 [CARRÉ DR 36 AV 25]
```

Les diagrammes ci-contre et de la page précédente montrent les graphiques que l'on peut obtenir avec la tortue. Les exemples d'un article précédent étaient destinés à vous faire expérimenter le LOGO. Vos divers essais ont très bien pu donner des résultats surprenants. Ainsi, pour un triangle, vous aurez pu écrire :

```
RÊPÊTE 3[AV 50 DR 60]...
```

... et découvrir que vous avez créé en fait une moitié d'hexagone! Si vous avez essayé de dessiner l'étoile à six branches, vous vous serez rendu compte que cela est beaucoup plus difficile qu'il n'y paraît. Lorsque vous avez affaire à ce genre de problème, il peut être utile de « jouer à la tortue », c'est-à-dire de vous mettre à sa place et d'apprécier mentalement les déplacements. Pensez à ce que peut être le tracé d'une forme fermée pour la tortue. Elle doit la contourner entièrement et finir à son point de départ, dans sa direction originelle. La tortue doit donc toujours tourner d'un multiple de 360 degrés. Lorsque la forme est convexe (aucun angle interne supérieur à 180 degrés), la tortue aura tourné d'exactly 360 degrés. Dans le cas d'un triangle, trois virages doivent avoir lieu, chacun devant être de $360/3 = 120$ degrés. A partir de cela, vous êtes à même de trouver la commande exacte :

```
RÊPÊTE 3 [AV 50 DR 120]
```

Les mêmes principes s'appliquent à des polygones non convexes, tels que les étoiles. La seule différence est que l'angle total parcouru est un multiple entier de 360 degrés (du fait que vous créez plus d'une forme complète).

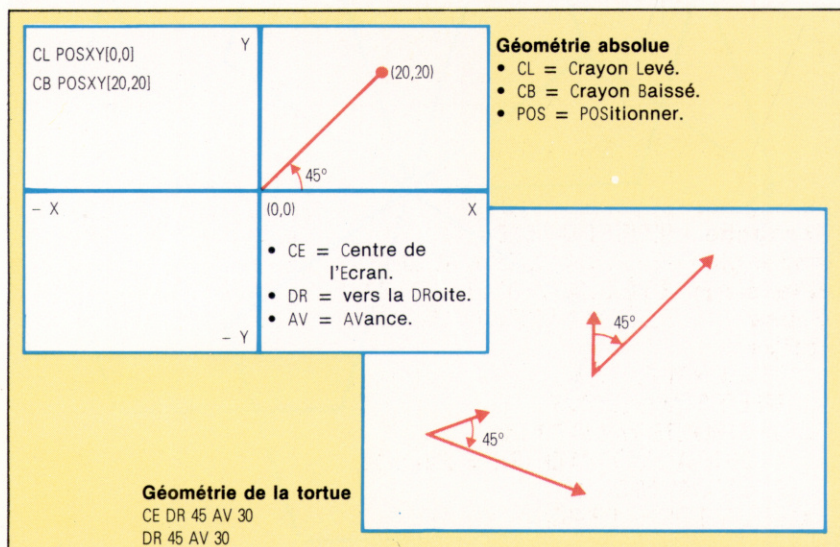
Ces principes sont généraux et reviennent à ce que l'on appelle « le théorème du parcours total de la tortue ».

Si vous examinez attentivement le tracé de l'étoile à six branches du point de vue de la tortue, vous en viendrez à la conclusion qu'elle n'a pas pu aller et venir sur chaque pointe en tournant à chaque fois. Une procédure plus complexe a donc été utilisée.

La géométrie de la Tortue

La géométrie de la tortue est relative, les déplacements se font par rapport au point courant. La géométrie avec coordonnées utilise des valeurs absolues, l'écran étant représenté comme une grille avec un certain nombre d'unités de longueur verticalement et horizontalement. Le point de départ, ou origine, est le centre. Chaque point de la grille a des coordonnées spécifiques, et tout déplacement est défini selon ces références sur la grille. Il est cependant possible d'utiliser des coordonnées absolues pour la tortue.

Par exemple, la commande POSXY 20 30 déplace la tortue de sa position courante sur le point (20,30). Si l'on a spécifié CRAYON-BAISSÉ (CB), la tortue tracera une ligne en se déplaçant. A l'inverse, CRAYON-LEVÉ (CL) fera que la tortue ne laisse aucune trace.



L'origine (0,0) est le centre de l'écran. Les procédures suivantes exécutent la même chose et illustrent bien la différence entre la géométrie de la tortue et la géométrie avec coordonnées.

```
PROCÉDURE CARRÉ1
  RÉPÈTE 4[AV 50 DR 90]
FIN
```

```
PROCÉDURE CARRÉ2
  POSXY 0 50
  POSXY 50 50
  POSXY 50 0
  POSXY 0 0
FIN
```

De retour à TRACER, CARRÉ1 et CARRÉ2 donneront exactement le même résultat. Mais qu'arrive-t-il si l'on tourne les deux carrés de 30° ? DR 30 CARRÉ1 reste exact, mais la deuxième procédure devient fautive et devra être réécrite. Et cela n'est pas facile à faire en coordonnées absolues... Mais il y a des cas où l'utilisation de coordonnées absolues pour LOGO se révèle précieuse.

Une autre caractéristique de la géométrie de la tortue est sa vue à très court terme. La tortue ne s'occupe que d'un seul déplacement à la fois. Elle élabore des formes par courts tronçons. Jouons à la tortue pour voir comment se trace un cercle. Que doit faire une tortue pour parcourir un cercle ? Avancer sur une très courte distance et tourner pour recommencer, cette séquence étant répétée un très grand nombre de fois. En termes de langage LOGO, cela pourrait s'écrire :

```
PROCÉDURE CERCLE
  RÉPÈTE 360 [AV 1 DR 1]
FIN
```

Cette procédure est un peu lente. Le cercle pourrait se tracer plus vite si la tortue n'apparaissait pas à chaque répétition. La commande CACHETORTUE rend la tortue invisible; MONTRETORTUE la fait réapparaître. Notre exemple trace en réalité un polygone de 360 côtés. Les lignes de ce polygone-cercle sont si courtes qu'elles donnent l'impression d'une courbe très douce. 360 traits sont largement suffisants pour donner l'illusion d'un cercle. On peut même dire que 36 côtés seulement suffiraient. Aussi, la procédure suivante peut-elle être retenue dans la mesure où elle est plus rapide :

```
PROCÉDURE CERCLE
  RÉPÈTE 36 [AV 10 DR 10]
FIN
```

Essayez d'écrire des procédures pour des demi-cercles, des quarts de cercle et des sixièmes de cercle. En combinant deux de ces mêmes formes, vous obtiendrez un effet de pétale.

Comment vous y prendriez-vous, si vous étiez une tortue (!) pour tracer une spirale « rentrante » ?

Vous vous déplacerez encore par petits tronçons, tournant à chaque fois d'un angle de plus



en plus grand. La procédure de programmation que nous donnons ici est un peu longue et compliquée du fait que nous n'utilisons que des commandes déjà vues.

```
PROCÉDURE SPIRALE
  AV 10 DR 10 AV 10 DR 20 AV 10 DR 30
  AV 10 DR 40 AV 10 DR 50 AV 10 DR 60
  AV 10 DR 70 AV 10 DR 80 AV 10 DR 90
FIN
```

Nuances entre les Logos

Les éditeurs LOGO sont très semblables entre eux, mais chacun a ses particularités dues au clavier de la machine considérée. Reportez-vous au manuel LOGO pour votre machine. Pour les versions LCS1, la commande d'édition de la procédure CARRÉ est ÉDIT « CARRÉ » (guillemet seulement *avant* le nom).

Réponses aux exercices

Triangle RÉPÈTE 3[AV 50 DR 120]
Pentagone RÉPÈTE 5[AV 50 DR 72]
Hexagone RÉPÈTE 6 [AV 60 DR 60]
Rectangle RÉPÈTE 2[AV 25 DR 90 AV 50 DR 90]
Parallélogr. RÉPÈTE 2[AV 25 DR 70 AV 50 DR 110]
Ellipse RÉPÈTE 2[AV 50 DR 70 AV 50 DR 110]
Étoiles :
 1. RÉPÈTE 5[AV 50 DR 144]
 2. RÉPÈTE 8 [AV 50 DR 135]
 3. DR 30 RÉPÈTE 3[AV 50 DR 120]
 CL GH 30 AV 29 DR 90 CB (GH = Gauche)
 RÉPÈTE 3[AV 50 DR 120]
 4. RÉPÈTE 10[AV 50 DR 108]

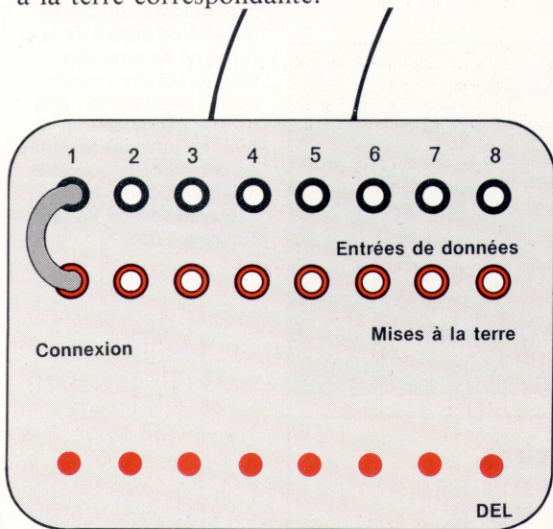




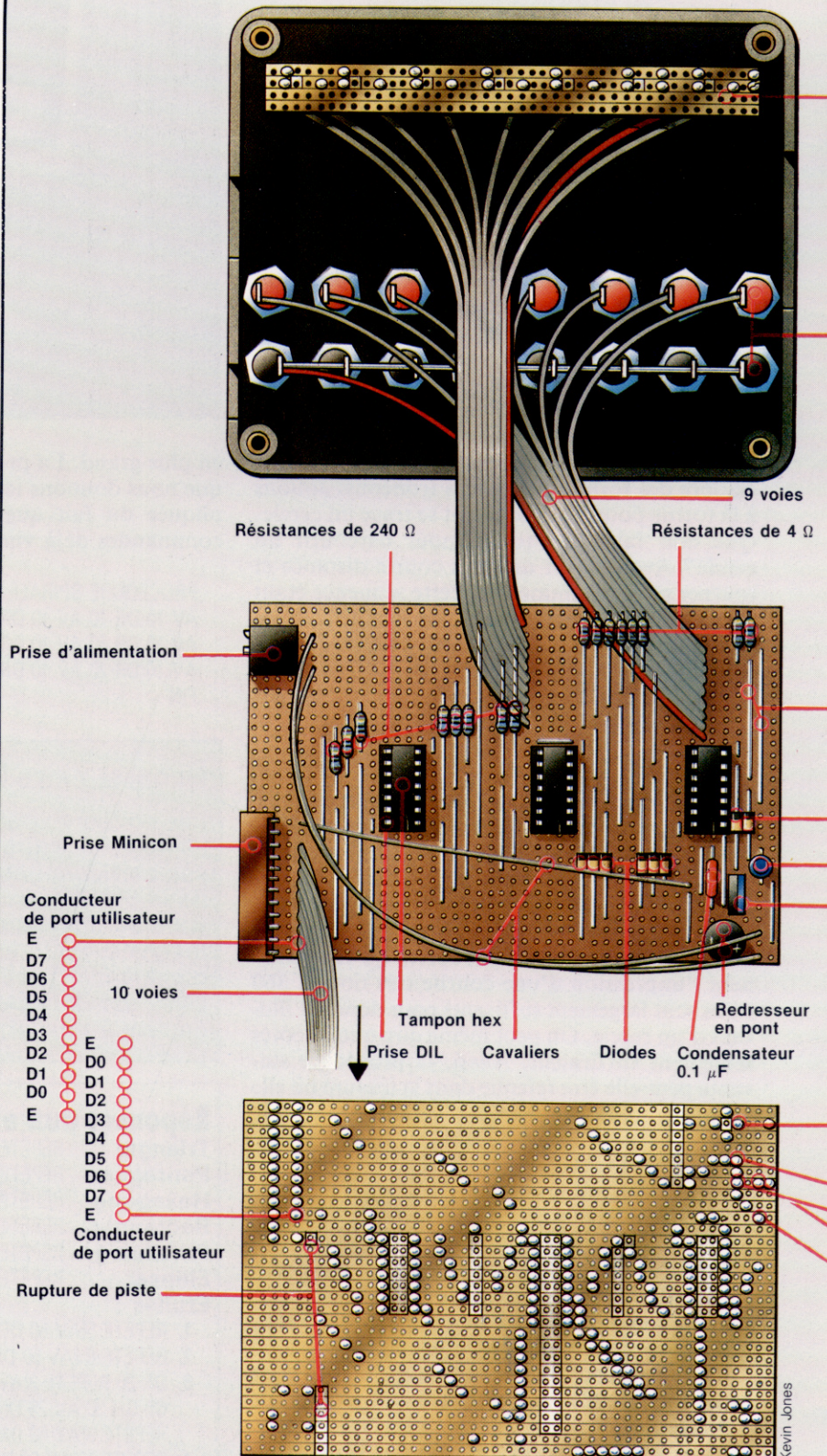
Un boîtier prêt à fonctionner

Dans notre série « Boîte à outils », nous vous avons appris comment construire un boîtier tampon permettant de gérer l'entrée et la sortie. Vérifions le bon fonctionnement de ce boîtier...

Après avoir terminé la construction du boîtier tampon, nous pouvons écrire un programme simple, pour tester son fonctionnement, en l'utilisant comme minuterie de réaction. Nous devons ici utiliser le bit 7 pour l'interrupteur d'entrée et les bits 0 à 6 pour les DEL. L'objectif est de calculer l'intervalle compris entre le moment où un interrupteur est pressé et l'instant précis où la DEL s'allume. Nous devons d'abord établir une connexion entre le terminal 7 du boîtier et la mise à la terre correspondante.



Notre programme définit le registre de direction des données, de telle sorte que le bit 7 soit destiné à l'entrée et que tous les autres bits soient réservés à la sortie. Il met ensuite à zéro le registre de données. Après un laps de temps aléatoire, sept des DEL sont allumées et le chronomètre de l'ordinateur est initialisé. Le chronométrage continue jusqu'à ce que le bit 7 passe à 0 lorsque la connexion est établie. Les LED sont connectées de façon qu'elles s'allument lorsqu'un 0 est présent et qu'elles s'éteignent pour un 1.





Intérieur du boîtier

Introduisez une certaine longueur de fil dénudé dans les contacts des prises. Soudez-le à chaque contact et testez le montage. Prenez 20 cm de câble-ruban et retirez trois fils, laissant un câble-ruban à neuf voies avec une bande latérale colorée. Dénudez les extrémités des fils. Soudez le fil coloré à la prise située à l'extrême gauche. Soudez les huit autres fils aux contacts des autres prises correspondantes. Testez le montage.

DEL

Prises

Liaison

Diodes

Condensateur électrolytique de 1 μ F

Régulateur de tension

Redresseur en pont

Régulateur de tension

Condensateur électrolytique de 1 μ F

Construction de la carte

La carte illustrée a 30 pistes de 45 trous et entre exactement dans le boîtier. Suivez attentivement les illustrations afin de ne pas avoir de problème de construction. Utilisez le minimum de soudure et veillez à ne pas shunter les pistes; vérifiez le montage en respectant la position de chaque composante. Les diodes, le condensateur électrolytique, le redresseur en pont et le régulateur de tension doivent être connectés dans la direction illustrée. Tous les composants sont sensibles à la chaleur, ne les chauffez donc pas trop avec le fer à souder. En positionnant les prises minicon et d'alimentation, veillez à positionner les broches dans les bons trous de carte mais ne pliez pas les broches trop brusquement. Lorsque tout est en place sur la carte, coupez les pistes de cuivre en suivant exactement les illustrations. Vous pouvez utiliser un outil spécial ou couper le cuivre graduellement à l'aide d'une mèche de perceuse tenue entre vos doigts. Ne laissez pas d'éclats de cuivre sur la carte. Soudez les câbles-rubans. L'orientation de la prise et des conducteurs de DEL est illustrée par la bande colorée, mais le conducteur du port utilisateur doit être connecté de façon précise — les deux lignes de mise à la terre doivent être placées dans les trous 1 et 10 et les lignes de signaux doivent être placées dans la séquence appropriée dans les trous 2 à 9, en veillant à ce que la ligne la moins significative se trouve près du bord de la carte. Finalement, en utilisant la carte comme modèle, coupez des ouvertures dans les côtés du boîtier pour recevoir les prises et le conducteur du port utilisateur.

```

10 REM * TEMPS DE REACTION DU BBC **
15 :
20 DDR=&FEG2: DATREG=&FEG0
30 ?DDR=127: REM LINES 0-6 OUTPUT
40 ?DATREG=127: REM LEDS OFF
50 :
60 CLS: PRINT "SOYEZ PRET"
70 DELAY=3000+RND(9000)
80 FOR I=1 TO DELAY:NEXT:REM DELAY LOOP
85 FOR 0=1 TO 200:NEXT D
90 :
97 REPEAT UNTIL ?DATREG AND 128=1
100 ?FEG0=0: REM TURN LEDS ON
110 TIME =0: REM INIT TIMER
120 REPEAT
130 UNTIL ?
140 :
150 PRINT "INTERVALLE=" TIME/100 "SECONDES"
160 END

```

Le programme utilise TIME, une variable réservée qui retourne une valeur correspondant au nombre de centièmes de seconde écoulés depuis la dernière remise à zéro de TIME. L'utilisation de l'opérateur logique AND, dans la ligne 30, isole le bit 7 (valeur 128) afin qu'il puisse être testé séparément. Lorsque la connexion est établie, la valeur du bit 7 passe de 1 à 0.

Un programme similaire peut être écrit pour le Commodore 64 en utilisant son chronomètre. Il ne fonctionne pas comme TIME, il retourne une valeur en soixantièmes de seconde correspondant à la période écoulée depuis la mise sous tension de l'ordinateur. Pour l'utiliser, nous devons lire la valeur de TI au début de l'intervalle et la soustraire de la valeur de TI à la fin de l'intervalle.

```

10 REM CBM 64 ** TEMPS DE REACTION **
20 :
30 DDR=56579: DATREG=56577
40 POKE DDR,127: REM LINES 0-5 OUTPUT
50 POKE DATREG,127: REM LEDS OFF
60 :
70 PRINT CHR$(147): REM EFFACER L'ECRAN
80 PRINT "SOYEZ PRET"
90 DE=3000+INT(9000+RND(1))
100 FOR N=1 TO DE:NEXT: REM BOUCLE DE TEMPORISATION
110 :
120 POKE DATREG,0: REM ALLUMER LES DEL
130 T=TI: REM LIRE LE TEMPS DE DEPART
140 IF PEEK(DATREG) AND 128<>0 THEN 140
150 :
160 TM=(TI-T)/60: REM CALCULER L'INTERVALLE
170 PRINT "INTERVALLE=" TM;"SECONDES"
180 END

```

Expériences à tenter

1. Écrivez un programme qui allumera une DEL à la fois, de gauche à droite.
2. Écrivez un programme qui allume les DEL sur les lignes 0 à 6, comme dans la question 1, mais introduisez un interrupteur sur la ligne 7, pour changer la direction de la séquence. Pouvez-vous changer votre programme afin qu'il fonctionne avec un « train » de trois DEL ?
3. Écrivez un programme qui simule le jeter d'un dé, utilisant six DEL et un interrupteur.
4. Écrivez un programme qui simule un feu de circulation et qui utilise trois DEL.
5. Écrivez un programme qui compte le nombre de « voitures » (impulsions sur un interrupteur) qui arrivent lorsque un feu de circulation est rouge et qui change le feu lorsque le nombre de voitures dépasse dix, ou lorsqu'une minute s'est écoulée depuis le dernier changement.



Réponses aux problèmes « boîtier tampon »

Solutions possibles aux exercices de la page 1027 :

Séquence de DEL (I)

```
10 REM CBM 64 VERSION 2.1
20 DDR=56579: DATREG=56577
30 POKE DDR,255: REM TOUTE SORTIE
40 POKE DATREG,255: REM TOUTES LES DEL ETEINTES
50 GET A$
60 FOR N=1 TO 7
70 POKE DATREG,255-(2*N)
80 NEXT N
90 IF A$="" THEN 50
100 END
```

```
10 REM BBC VERSION 2.1
20 DDR=&FE62:DATREG=&FE60
30 ?DDR=255: REM TOUTE SORTIE
40 ?DATREG=255: REM TOUTES LES DEL ETEINTES
50 REPEAT
60 A$=INKEY$(1)
70 FOR N=0 TO 7
80 ?DATREG=255-(2*N)
85 FOR D=1 TO 200: NEXT D
90 NEXT N
100 UNTIL A$<>""
110 END
```

Séquence de DEL (II)

```
10 REM CBM 64 VERSION 2.2
20 DDR=56579: DATREG=56577
30 POKE DDR,127: REM ENTREE L7
40 POKE DATREG,255: REM TOUTES LES DEL ETEINTES
50 GET A$
60 FOR N=0 TO 7 STEP 5
70 POKE DATREG,255-(2*N)
80 NEXT N
90 IF PEEK(DATREG) AND 128=0 THEN S=-1
100 IF PEEK(DATREG) AND 128=1 THEN S=1
110 IF A$="" THEN 50
120 END
```

```
10 REM BBC VERSION 2.2
20 DDR=&FE62:DATREG=&FE60:S=1
30 ?DDR=127: REM ENTREE L7
40 ?DATREG=255: REM TOUTES LES DEL ETEINTES
50 REPEAT
60 A$=INKEY$(1)
70 FOR N=0 TO 7 STEP 5
80 ?DATREG=255-(2*N)
85 FOR D=1 TO 200: NEXT D
90 IF DATREG AND 128=0 THEN S=-1 ELSE S=1
100 NEXT N
110 UNTIL A$<>""
120 END
```

Séquence de DEL (III)

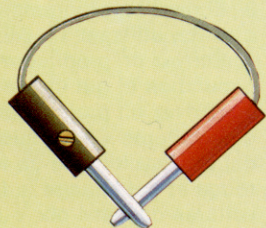
Pour le train de trois DEL, apportez les changements suivants à la réponse (2) :

```
Version CBM
60 FOR N=2 TO 7 STEP 5
70 POKE DATREG,255-((2*N)+2*(N-1)+2*(N-2))
```

```
Version BBC
70 FOR N=2 TO 7 STEP 5
80 ?DATREG=255-((2*N)+2*(N-1)+2*(N-2))
```

Cordon de raccordement

Utilisez 8 cm de fil isolé (une seule ligne du câble-ruban, par exemple) pour connecter une fiche rouge à une fiche noire. Ceci se nomme un cordon de raccordement et sert à relier une prise à une autre. Vous avez besoin de huit cordons comme celui-ci.



Kevin Jones

Jeter de dés

```
10 REM CBM 64 VERSION 2.3
20 DDR=56579: DATREG=56577
30 POKE DDR,127: REM ENTREE DE L7
40 POKE DATREG,255: REM TOUTES LES DEL ETEINTES
50 GET A$
60 IF PEEK(DATREG) AND 128<0 THEN 60
70 NR=INT(RND(1)*6)+1: REM NUMERO DE SELECTION
80 POKE DATREG,255-((2*NR)-1)
90 IF PEEK(DATREG) AND 128<1 THEN 90
100 IF A$="" THEN 50
110 END
```

```
10 REM BBC VERSION 2.3
20 DDR=&FE62:DATREG=&FE60:S=1
30 ?DDR=127: REM ENTREE DE L7
40 ?DATREG=255: REM TOUTES LES DEL ETEINTES
50 REPEAT
60 A$=INKEY$(1)
70 REPEAT UNTIL (?DATREG AND 128)=0
80 NUMBER=RND(6): REM NUMERO DE SELECTION
85 FOR D=1 TO 200: NEXT D
90 ?DATREG=255-((2*NUMBER)-1)
100 REPEAT UNTIL ?DATREG AND 128=1
110 UNTIL A$<>""
120 END
```

Feux de circulation (I)

```
10 REM CBM 64 VERSION 2.4
20 DDR=56579: DATREG=56577
30 POKE DDR,255: REM TOUTE SORTIE
40 POKE DATREG,255: REM TOUTES LES DEL ETEINTES
45 REM BIT2=RED,BIT1=AMBER,BIT0=GREEN
50 RD=255-4: AM=255-2:GN=255-1
60 GET A$
70 POKE DATREG,RD
80 FOR N=1 TO 200:NEXT:REM BOUCLE DE RETARD
90 POKE DATREG,RD+AM
100 FOR N=1 TO 40:NEXT:REM BOUCLE DE RETARD
110 POKE DATREG,GN
120 FOR N=1 TO 200:NEXT: REM BOUCLE DE RETARD
130 IF A$="" THEN 60
140 END
```

```
10 REM BBC VERSION 2.4
20 DDR=&FE62:DATREG=&FE60:S=1
30 ?DDR=255:REM TOUTE SORTIE
40 ?DATREG=255: REM TOUTES LES DEL ETEINTES
50 REM BIT2=RED, BIT1=AMBER,BIT0=GREEN
60 RD=255-4:AMBER=255-2:GREEN=255-1
70 REPEAT
80 A$=INKEY$(100)
85 FOR D=1 TO 200: NEXT D
90 ?DATREG=RD
100 FOR N=1 TO 200: NEXT: REM BOUCLE DE RETARD
110 ?DATREG=RD+AMBER
120 FOR N=1 TO 40: NEXT: REM BOUCLE DE RETARD
130 ?DATREG=GREEN
140 FOR N=1 TO 200: NEXT: REM BOUCLE DE RETARD
150 UNTIL A$<>""
160 END
```

Feux de circulation (II)

Apportez les changements suivants à la réponse (4) :

```
Version CBM
30 POKE DDR,127: REM ENTREE DE L7
65 T=TI: REM INITIALISATION DU CHRONOMETRE
75 IF PEEK(DATREG) AND 128=0 THEN C=C+1
76 IF PEEK(DATREG) AND 128<>1 THEN 76
77 IF C<=10 AND T-TI<3600 THEN 75
```

```
Version BBC
30 ?DDR=127:ENTREE DE L7
75 T=0:C=0
95 REPEAT
96 IF ?DATREG AND 128=0 THEN C=C
97 REPEAT UNTIL ?DATREG AN?D 128=1
98 UNTIL T>6000 OR C>10
```




Carte 6809

Voici reproduit, avec l'aimable autorisation de Motorola Inc., le constructeur du 6809, la première partie de la carte référence du programmeur 6809.

Instruction	Formes	Modes d'adressage												Description	5	3	2	1	0										
		Immédiat			Direct			Indexé			Étendu									Op	-	#							
		Op	-	#	Op	-	#	Op	-	#	Op	-	#																
ABX															3A	3	1						B + X → X (sans signe)	
ADC	ADCA ADCB	89 C9	2 2	2 2	99 D9	4 4	2 2	A9 E9	4+ 4+	2+ 2+	B9 F9	5 5	3 3										A + M + C → A B + M + C → B	†	†	†	†	†	
ADD	ADDA ADDB ADDD	8B CB C3	2 2 4	2 2 3	9B DB D3	4 4 6	2 2 2	AB EB E3	4+ 4+ 6+	2+ 2+ 2+	BB FB F3	5 5 7	3 3 3										A + M → A B + M → B D + M: M + 1 → D	†	†	†	†	†	
AND	ANDA ANDB ANDCC	84 C4 1C	2 2 3	2 2 2	94 D4	4 4	2 2	A4 E4	4+ 4+	2+ 2+	B4 F4	5 5	3 3										A ∧ M → A B ∧ M → B CC ∧ IMM → CC	.	†	†	0	.	
ASL	ASLA ASLB ASL													48 58	2 2	1 1								A } B } M } c ← b7 ← b0	8	†	†	†	†
ASR	ASRB ASR ASR													47 57	2 2	1 1								A } B } M } b7 ← b0 → c	8	†	†	.	†
BIT	BITA BITB	85 C5	2 2	2 2	95 D5	4 4	2 2	A5 E5	4+ 4+	2+ 2+	B5 F5	5 5	3 3										Bit Test A (M ∧ A) Bit Test B (M ∧ B)	.	†	†	0	.	
CLR	CLRA CLRB CLR													4F 5F	2 2	1 1								0 → A 0 → M 0 → M	.	0	†	0	0
CMP	CMPA CMPB CMPD CMPS CMPU CMPX CMPY	81 C1 10 83 11 8C 11 83 8C 10 8C	2 2 5 5 5 4 5 4 4 5 4	2 2 4 4 4 3 4 3 4 3	91 D1 10 93 11 9C 11 93 10 9C	4 4 7 7 7 6 7 7 6 7	2 2 3 3 3 2 3 3 2	A1 E1 10 A3 11 AC 11 A3 AC 10 AC	4+ 4+ 7+ 7+ 7+ 6+ 7+ 7+ 6+ 7+ 7+	2+ 2+ 3+ 3+ 3+ 2+ 3+ 3+ 2+ 3+ 3+	B1 F1 10 B3 11 BC 11 B3 BC 10 BC	5 5 8 8 8 7 8 7 8 4											Compare M à A Compare M à B Compare M: M + 1 à D Compare M: M + 1 à S Compare M: M + 1 à U Compare M: M + 1 à X Compare M: M + 1 à Y	8	†	†	†	†	
COM	COMA COMB COM													43 53	2 2	1 1								A → A B → B M → M	.	†	†	0	1
CWAI		3C	≥20	2																			CC ∧ IMM → CC attend interrup.	.	†	†	0	†	
DAA														19	2	1								Ajustement décimal A	.	†	†	0	†
DEC	DECA DECB DEC													4A 5A	2 2	1 1								A - 1 → A B - 1 → B M - 1 → M	.	†	†	†	.
EOR	EORA EORB	88 C8	2 2	2 2	98 D8	4 4	2 2	A8 E8	4+ 4+	2+ 2+	B8 F8	5 5	3 3										A ⊕ M → A B ⊕ M → B	.	†	†	0	.	
EXG	R1, R2	E1	8	2																			R1 → R2 ²	
INC	INCA INCB INC													4C 5C	2 2	1 1								A + 1 → A B + 1 → B M + 1 → M	.	†	†	†	.
JMP																							EA ³ → PC	
JSR																							Saut à sous-programme	
LD	LDA LDB LDD LDS LDU LDX LDY	86 C6 CC 10 CE CE 8E 10 8E	2 2 3 4 3 3 4	2 2 3 4 3 3 4	96 D6 5 10 DE DE 9E 10 9E	4 4 5 6 5 5 5 6 5	2 2 2 3 2 2 2 3 2	A6 E6 5 10 EE EE AE 10 AE	4+ 4+ 5+ 6+ 5+ 5+ 5+ 6+ 5+	2+ 2+ 2+ 3+ 2+ 2+ 2+ 3+ 2+	B6 F6 6 10 FE FE BE 10 BE BF	5 5 6 7 6 6 6 7 6											M → A M → B M: M + 1 → D M: M + 1 → S M: M + 1 → U M: M + 1 → X M: M + 1 → Y	.	†	†	0	.	
LEA	LEAS LEAU LEAX LEAY																						EA ³ → S EA ³ → U EA ³ → X EA ³ → Y	

- OP : code opération (hexadécimal);
- : nombre de cycles MPU;
- # : nombre d'octets de programme;
- † : transférer dans;
- 0 : non affecté;
- CC : registre de code condition;
- ∧ : concaténation;
- V : ou logique;
- ∨ : ou exclusif logique;
- : multiplier;
- M̄ : complément de M;
- : transférer dans;
- H : demi-retenu (venant du bit 3);
- N : négatif (bit de signe);
- Z : zéro (réinitialiser);
- V : dépassement, complément à 2;
- C : retenue venant de l'ALU;
- † : test; mettre à 1 si vrai, à 0 sinon;

- Cette colonne donne un cycle de base et compte d'octet. Pour obtenir le compte total, additionner les valeurs obtenues à partir de la table de MODE D'ADRESSAGE INDEXÉ en annexe F.
- R1 et R2 peuvent être n'importe quel couple de registres 8 bits ou 16 bits. Les registres 8 bits sont : A, B, CC, DP. Les registres 16 bits sont : X, Y, U, S, D, PC.
- EA est l'adresse effective.
- Les instructions PSH et PUL requièrent 5 cycles plus 1 cycle pour chaque octet entré ou sorti en pile.
- 5(6) signifie : 5 cycles sans branchement, 6 cycles avec branchement.
- SWI met les bits I et F à 1. SW12 et SW13 n'affectent pas I et F.
- Les codes conditions sont mis à 1 comme résultat de l'instruction.
- La valeur du drapeau de demi-retenu est indéfinie.
- Cas particulier — retenue si b7 est mis.

Réussite

L'adolescent qui fait fortune en écrivant des programmes de jeu sur ordinateur fait rêver. Nous racontons comment s'enrichir grâce à une trouvaille originale en programmation.

Au cours de l'été de 1984, Gremlin Graphics a lancé un nouveau jeu sur ordinateur nommé Wanted : Monty Mole. On s'attend à ce que ce jeu remporte beaucoup de succès. Si c'est le cas, le programmeur, Peter Harrap, tirera de l'opération quelques millions de centimes. Le jeune Harrap n'a que dix-neuf ans et Wanted : Monty Mole était son premier jeu commercial. Peter Harrap fait partie d'un groupe de programmeurs amateurs qui, encore adolescents, peuvent gagner d'importantes sommes d'argent en créant un jeu populaire. Un garçon de quatorze ans a gagné près de 50 000 F en écrivant des programmes de jeu pour le BBC Micro. Un autre programmeur, âgé de dix-sept ans, a reçu de la société pour laquelle il travaille une voiture Lotus Esprit.

Comme la plupart des programmeurs de jeu, Harrap est autodidacte. Il posséda d'abord un Sinclair ZX81, et après avoir constaté que peu de programmes étaient disponibles, il décida d'essayer d'écrire ses propres programmes de jeu. Il dut apprendre le code machine du Z80. Harrap s'intéressa rapidement au Sinclair Spectrum et concentra tous ses efforts sur l'amélioration d'un jeu qui l'intéressait particulièrement, Ant Attack de Quicksilva. Harrap entra dans le code machine et modifia le paysage proposé par le programme. Sa version modifiée fut rejetée par Quicksilva, mais un mordu d'informatique, Ian Steward, qui recherchait de jeunes talents pour sa petite société de logiciel, fut impressionné par la bande que Harrap lui envoya. Le jeu d'Harrap violait les droits d'auteur de la société ayant créé Ant Attack, la société de Steward, Gremlin Graphics ne l'accepta donc pas. Cependant, la société avait besoin d'un programmeur hautement qualifié sur Spectrum : ils l'embauchèrent immédiatement.

Quatre mois pour un programme

Toutes les sociétés de logiciel sont à la recherche de thèmes intéressants et inusités qui pourraient éventuellement placer un jeu au premier rang. Gremlin Graphics eut l'idée de créer Wanted : Monty Mole, en espérant que son thème de grève de mineurs plairait beaucoup au public. L'idée fut mise en pratique par Harrap, qui écrivit un programme exécuté sur le Sinclair Spectrum.

Harry mit quatre mois pour écrire le programme. Il commença par écrire un générateur



de plans objets afin de pouvoir concevoir rapidement les diverses formes graphiques, puis il dessina la mine de charbon et programma finalement les règles régissant les mouvements des personnages.

Le jeu reçut un accueil très favorable en raison de son originalité et devint rapidement un succès commercial. Gremlin Graphics paie à Harrap un pourcentage sur chaque vente de cassette. Ce genre d'arrangement varie d'une société à l'autre. La plupart des sociétés versent des pourcentages croissants : cinq pour cent pour les 3 000 premières copies vendues, puis sept pour cent pour les 10 000 suivantes. Le pourcentage peut être calculé sur le prix brut, qui représente le montant que la société de logiciel reçoit du distributeur, ou sur le prix réel, qui est le montant payé par les consommateurs.

La plupart des sociétés versent aussi des avances à leurs programmeurs. Ce montant est soustrait des pourcentages qui seront payés par la suite. Cependant, certaines sociétés optent plutôt pour un montant forfaitaire. Les programmeurs doivent se méfier de tels arrangements qui peuvent se traduire par une perte de revenus si les ventes sont élevées.

Lors d'un arrangement avec pourcentage, le programmeur conserve les droits d'auteur. Ce qui signifie qu'en cas de conflit, le programmeur est entièrement responsable du produit qu'il a créé. Si le programmeur reçoit un montant global pour

Vente au détail

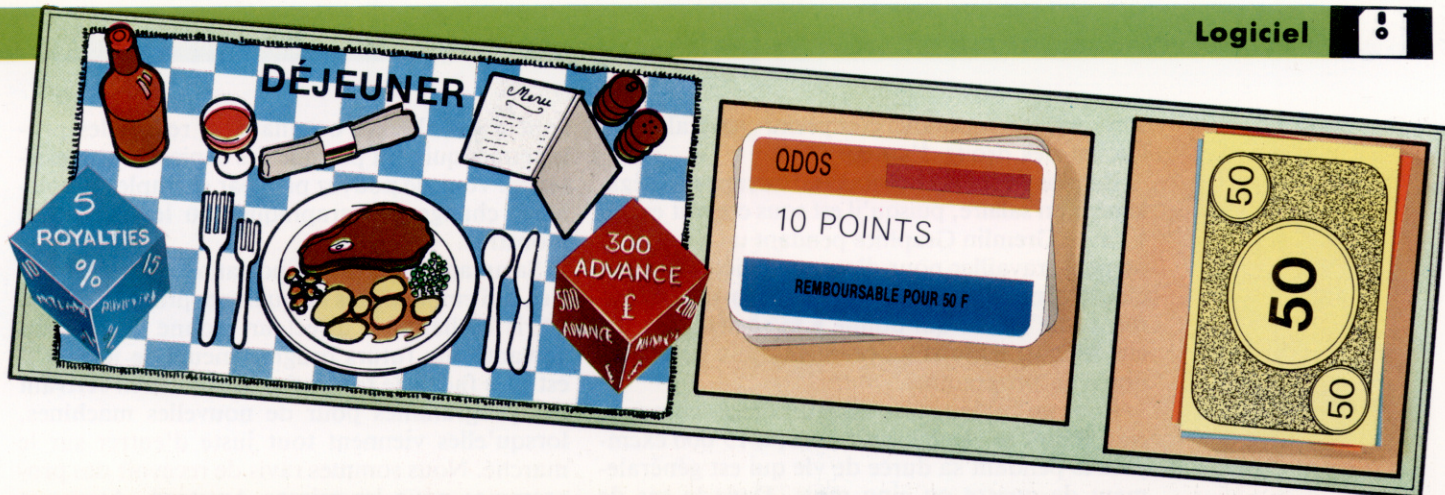
La présentation dans les points de vente représente sans doute le contact le plus important avec l'utilisateur : ici, les effets de la publicité, des critiques, de l'emballage, de la mise en marché, influencent le choix de l'utilisateur. La disponibilité est aussi vitale, il est donc avantageux de distribuer le produit par l'intermédiaire de chaînes de boutiques informatiques bien implantées. (Cl. Ian McKinnell.)

Programmation

Des milliers de propriétaires d'ordinateurs domestiques tentent d'écrire des programmes commerciaux ; certains réussissent, trouvent un éditeur et gagnent un peu d'argent.

A noter

Le jeu présenté ci-contre est une adaptation de l'anglais.



CLAVIER	prendre carte information	GOTO déjeuner	Passer un tour	Prendre carte QDOS	GOTO déjeuner	prendre carte QDOS	COMPILER ICI
RUE DE LA LUNE	Idée libre	ÉDITEUR ESCROC	GOTO déjeuner	ENNEMI	Passer un tour	BUG MART	PUBLICITÉ
ÉDITEUR ESCROC	GOTO déjeuner	ENNEMI	Passer un tour	BUG MART	Prendre carte QDOS	ÉDITEUR TERRIFIANT	« Point — Quelle journée ! »
ENNEMI	Passer un tour	BUG MART	Prendre carte QDOS	ÉDITEUR BRANCHE	GOTO déjeuner	LA PAIX	PRENDRE 45 QDOS
BUG MART	Prendre carte QDOS	ÉDITEUR BRANCHE	GOTO déjeuner	?	Jouer de nouveau	BUG MART	?
?	Jouer de nouveau	?	Jouer de nouveau	AMBI	AMBI	BUG MART	AMBI
ÉDITEUR REQUIN	GOTO déjeuner	AMBI	Jouer de nouveau	NE PAS PASSER PAR RETOUR	GOTO CLAVIER	BUG MART	NE PAS PASSER PAR RETOUR
DÉGAGER	Retour d'une case	NE PAS PASSER PAR RETOUR	GOTO CLAVIER	GOTO CLAVIER	GOTO CLAVIER	BUG MART	GOTO CLAVIER
ÉDITEUR REQUIN	GOTO déjeuner	BUG	BUG MART	ÉDITEUR TIMORE	VOTRE MAGAZINE 65	BUG MART	ÉDITEUR TIMORE
DÉGAGER	Retour d'une case	BUG	BUG MART	VOTRE MAGAZINE 65	BERGSTRÖM	BUG MART	VOTRE MAGAZINE 65
ÉDITEUR REQUIN	GOTO déjeuner	BUG	BUG MART	BERGSTRÖM	BERGSTRÖM	BUG MART	BERGSTRÖM
DÉGAGER	Retour d'une case	BUG	BUG MART	BERGSTRÖM	BERGSTRÖM	BUG MART	BERGSTRÖM
ÉDITEUR REQUIN	GOTO déjeuner	BUG	BUG MART	BERGSTRÖM	BERGSTRÖM	BUG MART	BERGSTRÖM
DÉGAGER	Retour d'une case	BUG	BUG MART	BERGSTRÖM	BERGSTRÖM	BUG MART	BERGSTRÖM

BLUFF

Le jeu des jeux
 Les joueurs sont encouragés à tricher, à escroquer les autres joueurs et à violer le maximum de règles.

- L'objectif est de mettre les autres joueurs hors jeu en accumulant tous les QDOS et tout l'argent. Pour ce faire, vous devez soumettre vos idées à un éditeur et conclure un arrangement avec lui. Votre premier pourcentage ne vous rapporte qu'un sous-centage; les arrangements subséquents vous rapportent un pourcentage et une avance.
- Dès que l'arrangement est conclu, votre cassette est mise sur le marché, chaque joueur vous verse une partie de votre revenu.
- Chaque fois que vous appuyez sur RETURN, vous avez une nouvelle idée et vous récoltez vos gains et selon votre QDOS multiplié par votre pourcentage.
- Lorsque vous allez au CLAVIER vous devez pas appuyer sur RETURN, vous ne devez pas récolter vos gains et vous devez y rester jusqu'à ce que d'un autre. BERGSTRÖM prend votre dernier jeu et l'inclut dans son nouveau jeu et l'inclut dans son revenu provenant de ce jeu.

MAUVAISE DONNÉE
 Depuis l'original
 Les copies piratées de votre logiciel ont empiété le marché
 Perdre 25 QDOS

CHANCE



son jeu, ce montant comprend généralement l'achat des droits d'auteur.

En plus de ses pourcentages, Harrap reçoit également un salaire, puisqu'il est sous contrat exclusif avec Gremlin Graphics pendant un an et qu'il ne peut travailler pour d'autres sociétés de logiciel. De nombreux contrats comportent également une clause qui donne à la société de logiciel le droit exclusif du premier choix pour tout nouveau programme.

Pour qu'un jeu représente un succès commercial, il doit se vendre, au moins, à 15 000 exemplaires pendant sa durée de vie qui est généralement de quatre ou cinq mois. Dans le cas de Wanted : Monty Mole, la société a initialement reproduit 10 000 copies pour chaque machine (Spectrum et Commodore 64), avec la possibilité de faire d'autres copies. Lors du lancement du jeu, Ian avait prédit 20 000 ventes pour chaque machine.

Le succès à dix-sept ans

Christopher Kerry est un autre programmeur dont le succès fut encore plus précoce. A l'âge de dix-sept ans il quitta l'école pour commencer à travailler comme pigiste pour la société de logiciel House of Thor. Kerry écrivit le jeu Jack and the Beanstalk sur l'ordinateur Spectrum qu'il avait chez lui. Sans publicité, en deux mois quelques dizaines de milliers d'exemplaires avaient déjà été vendus, selon la société. House of Thor essaie de protéger son jeune poulain, craignant que d'autres sociétés soient intéressées par ses services. En plus de Christopher Kerry, House of Thor emploie quinze autres programmeurs.

Comme toutes les maisons de logiciel, House of Thor invite les gens à lui proposer des programmes et, s'ils semblent prometteurs, la société propose au programmeur de signer un contrat le jour même. Une réponse rapide est essentielle, puisque il se peut que le même jeu ait été offert à d'autres sociétés. House of Thor n'est intéressée que par les jeux de café et ceux-ci doivent être écrits en code machine. La société recherche éga-

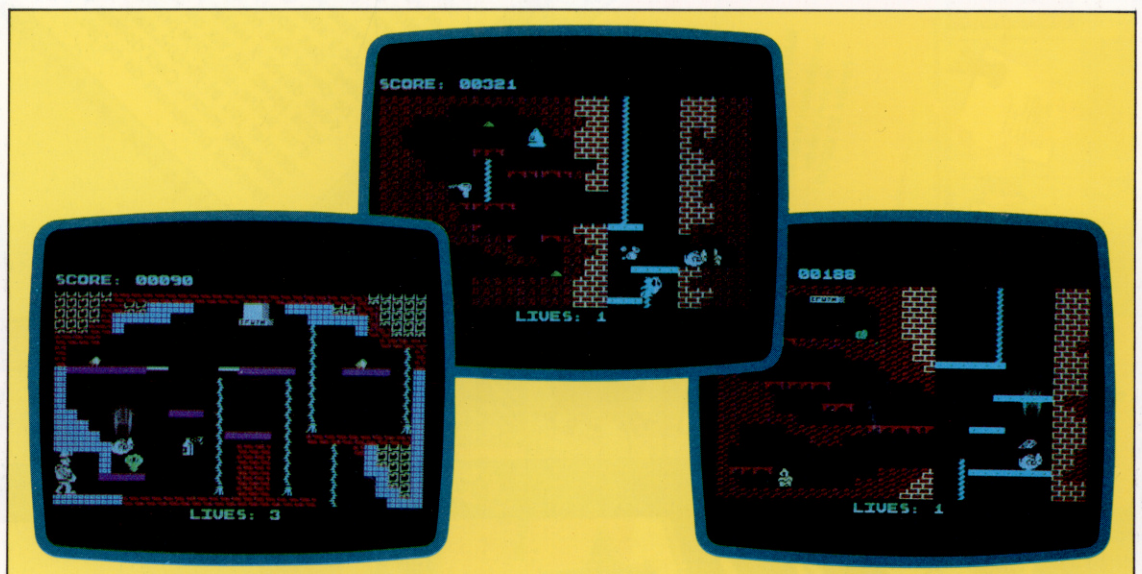
lement des thèmes originaux et rejette les programmes qui sont presque des copies de jeux existants : vous ne pouvez pas, par exemple, simplement changer les graphiques ou les noms des monstres.

Salamander est une importante société de logiciel, qui emploie plus de quarante programmeurs. Chris Holland, le directeur, donne les conseils suivants aux futurs programmeurs de jeu : « Il est plus facile de remporter un succès en écrivant des programmes pour de nouvelles machines, lorsqu'elles viennent tout juste d'entrer sur le marché. Nous sommes ravis de recevoir des programmes pour les micros Amstrad, Atmos et MSX. Pour ce qui est du Spectrum, la tâche est plus compliquée, sauf s'il s'agit d'une idée entièrement nouvelle. » La société propose environ cinquante titres pour la plupart des ordinateurs domestiques populaires — et pas seulement pour le Commodore 64 et le Spectrum.

Plutôt que de vendre des programmes à des sociétés de logiciel, certains programmeurs créent leur propre société. Llamasoft et Bug-Byte font partie des nombreuses sociétés de logiciel qui ont été fondées par de jeunes programmeurs de jeu. Malheureusement, un capital assez important est nécessaire pour démarrer. La société de Ian Stewart, Gremlin Graphics, débuta avec un seul programme, Potty Pigeon, mais espère en proposer cinq autres avant la fin de 1984. Chacun des quatre directeurs a dû investir personnellement pour financer la publicité et la distribution. Même une gamme de jeux populaires ne constitue pas une sécurité financière. Imagine fit faillite en juillet 1984 avec des dettes totalisant plus d'un million de francs, même si à un moment les chiffres de vente mensuels des programmes de jeu dépassaient les trois millions de francs.

Démarrer votre propre société de logiciel est une aventure. Sachez qu'en plus de bons jeux, il est également nécessaire d'avoir un bon conseiller financier. Mais toute personne peut proposer un programme à une maison de logiciel. Qui sait, vous pourriez écrire le prochain numéro un au palmarès des programmes de jeu.

La mine
Le programmeur qui a écrit Wanted : Monty Mole est un exemple parfait de ces adolescents programmeurs, autodidactes, brillants et très bien payés. Le jeu lui-même est très original : il donne une image assez antipathique de la grève des mineurs de 1984, mais chaque copie vendue rapporte cinquante au fond d'aide aux grévistes.





Interface sonore

Nous avons vu comment une interface numérique est utilisée par des musiciens sur scène. Examinons le matériel et le logiciel qui mettent MIDI à la disposition de l'utilisateur d'ordinateurs domestiques.

Le Micon, ou MIDI controller, est produit par XRI Systems. C'est un exemple des programmes MIDI produits par de plus petites sociétés de logiciel. Il est destiné au Sinclair Spectrum et est formé d'une interface MIDI et d'un programme sur cassette. Ses fonctions de séquenceur autorisent huit pistes distinctes, chacune ayant une capacité de 2 951 événements (notes, pauses, etc.). La musique est entrée au clavier du synthétiseur interfacé et la touche espace du Spectrum sert à définir le nombre d'événements à jouer.

Lors de l'utilisation du synthétiseur, la musique défile à l'écran sous la représentation d'une portée de cinq lignes. Cette représentation comporte les clés, les durées et divers indicateurs musicaux. Cette notation indique mal les pauses, et les queues des notes sont toujours dirigées vers le haut au lieu d'être dirigées soit vers le haut, soit vers le bas, selon leur position sur la portée. En plus, les séquences de notes courtes ne sont pas groupées pour refléter le rythme de la musique, ce qui donne un affichage assez difficile à suivre. Mais il peut être imprimé à l'aide de l'imprimante ZX.

Le principal avantage de l'affichage est évident lors d'une séance d'édition. La musique est divisée en barres et, lorsqu'un numéro de barre est entré, des notes individuelles peuvent être effacées, insérées ou modifiées, en spécifiant la nouvelle note. Des barres entières peuvent être effacées et des groupes de barres peuvent être répétés à tout endroit à l'intérieur de la séquence. On peut sauvegarder jusqu'à dix séquences — près de 24 000 notes — sur bande, avec les caractéristiques appropriées, afin de les rejouer. Le tempo du jeu peut être défini à quatre milli-secondes près ou être commandé par une batterie électronique en utilisant la prise SYNC IN de l'interface.

Micon travaille également en temps réel. En fait, il « écoute » une exécution effectuée sur un synthétiseur compatible avec MIDI et entrera toutes les données dans la mémoire du Spectrum. Il n'affichera pas de notation de la performance, mais il sera cependant utile de deux autres manières. Il permettra d'abord aux musiciens de clavier d'entendre un enregistrement de leur propre performance, sans avoir à régler les niveaux d'entrées sur bande; son caractère immédiat sera certainement pratique en éducation musicale. Deuxièmement, pourvu que la musique nécessaire soit à la disposition du musicien, il diminue grandement la tâche de mise en séquence. Le musicien a toujours besoin d'un métronome afin de

maintenir un rythme stable, sinon la séquence serait rejouée avec toutes les erreurs de rythme du musicien. Comme pour tout système MIDI, le résultat obtenu par Micon dépend du niveau de perfectionnement du synthétiseur auquel il est connecté.

L'ensemble MIDI de Jellinghaus Music est destiné à une gamme plus large de machines : l'Apple II, le Commodore 64 et le Sinclair Spectrum. Il est fondamentalement similaire au Micon, puisque l'information musicale est entrée à partir du clavier du synthétiseur connecté. La principale différence réside dans le fait que les séquences sont entrées uniquement en temps réel et que l'affichage de la musique ne se fait pas sur une portée standard de cinq lignes.

L'organisation de la musique en barres est effectuée en spécifiant le nombre de noires ou de croches par groupe. Quatre valeurs sont disponibles : trois, quatre ou cinq noires par barre (3/4, 4/4 ou 5/4), ou sept croches (7/8) et, comme pour le Micon, le rythme de l'exécution doit être coordonné par un métronome ou par une batterie électronique.

Si la performance en temps réel est hors tempo, les données d'entrées deviennent alors une chaîne de notes « non barrées ». Lorsque la séquence sera rejouée, il n'y aura pas de problème, jusqu'à ce que survienne un groupe qui n'est pas formé de 3/4, 4/4, 5/4 ou 7/8. Dès que cela survient,



Commande sonore

En tirant avantage de la puce sonore du Commodore 64, un programme BASIC peut être écrit pour entrer de la musique dans un format visuel, avec un manche à balai, et pour le jouer par la suite. Dans l'exemple illustré ici, un point lumineux apparaît sur un clavier graphique et peut être déplacé vers le haut de la portée, en poussant le manche à balai vers l'avant, ou vers le bas de la portée, en tirant le manche à balai vers l'arrière. Lorsque vous atteignez la note désirée, vous déplacez le manche à balai latéralement pour choisir la durée de la note. La note est « enregistrée » lorsque le bouton de mise à feu est pressé. Vous êtes alors prêt à entrer la prochaine note de la séquence. (Cl. Liz Heaney.)



1. Commande par ordinateur

Le Commodore est responsable de la gestion de la mémoire et du fonctionnement du système musical par l'intermédiaire du logiciel. Le lecteur de disquette peut être utilisé pour effectuer des enregistrements numériques de séquences.

la fonction de mise en séquence est compromise. Ce qui signifie qu'une mesure 12/8 ne peut être entrée, même si le tempo est parfait : c'est dommage, puisque 12/8 est une mesure utilisée fréquemment en jazz, en rock, en funky et en reggae.

Si la séquence a été entrée avec succès, les données sont affichées à l'écran en colonnes, montrant : l'octave dans laquelle la note se trouve; la hauteur de la note dans l'octave; la durée en termes musicaux (noires et croches); un paramètre qui permet de contrôler le phrasé de la ligne musicale et les valeurs des touches 0 à 9 du clavier.

Ce type d'affichage donne en fait plus d'informations au niveau des notes individuelles que la notation standard utilisée par Micon, qui n'a jamais fourni les paramètres de phrasé ou les niveaux de vélocité — du moins sous forme numérique. C'est grâce à l'électronique qu'il est maintenant possible de définir de tels paramètres musicaux.

Cependant, de nombreux musiciens sans réelle formation apprennent rapidement à analyser une section de notation standard et, en se basant sur le mouvement des notes sur la portée, réussissent à imaginer la mélodie résultante ou ce qu'un accord peut apporter à l'harmonie d'un morceau. Avec un peu d'expérience la notation peut être lue et une séquence musicale peut être « entendue » mentalement en temps réel. Les affichages en colonnes sont sans doute excellents pour

vérifier les données, mais peu de personnes peuvent y lire comment se présente une mélodie. Un système idéal comporterait les deux types d'affichage.

Synthétiseurs conceptuels

Le principal problème que doit affronter le propriétaire d'un ordinateur domestique intéressé par la musique est, plutôt que de trouver le bon programme MIDI, d'obtenir un synthétiseur compatible avec MIDI qui soit assez évolué pour utiliser au maximum le potentiel de l'interface, sans que les frais impliqués dans l'achat d'une telle machine deviennent trop lourds. Si nous désirons un synthétiseur en mesure de gérer cinquante pour cent des pistes d'une chanson populaire actuelle, nous découvrons que de tels instruments coûtent entre 9 000 et 18 000 F. En dessous de cette fourchette, la plupart des synthétiseurs n'offrent que des applications limitées.

Le « synthétiseur conceptuel » offre une autre possibilité. Il s'agit d'un ensemble logiciel évolué et d'un clavier périphérique, qui utilisent le potentiel sonore d'un micro-ordinateur, et non une interface vers un synthétiseur. Ce type de système a été conçu avant que MIDI ne soit développé.

Le PDSG (programmable digital sound generator), fabriqué par Clef Products, est un exemple d'un tel système. Il établit une interface avec le micro-ordinateur BBC Modèle B. Il comprend



2. Six Trak à circuits séquentiels

Voici un synthétiseur puissant muni d'une enregistreuse multi-pistes. Il peut enregistrer six pistes différentes. Contrairement à de nombreux synthétiseurs, le Six-Trak permet de définir un son différent pour chacune des voix musicales, ce qui rend possible la création d'ensembles sonores complexes.

3. Model 64 à circuits séquentiels

En branchant le Model 64 dans le port d'extension de mémoire du Commodore 64, il est possible de former un système musical avec un synthétiseur équipé de MIDI, la mémoire de l'ordinateur, un stockage sur cassette ou sur disquette. Le Model 64 mémorise l'information relative à la synchronisation, au registre et à la modulation de quatre mille notes en mode enregistrement. Pour une exécution, l'interface peut soit envoyer le signal numérique exactement comme il fut reçu du clavier, soit le corriger.



Ian McKinnell.

un clavier à 61 notes et deux pédales. Ce clavier est en mesure d'établir une différence entre une frappe vigoureuse et un effleurement des notes, et ces données recueillies lors de l'exécution peuvent être enregistrées et reproduites. Il a 32 unités de génération sonores, plutôt que des oscillateurs, et chacune de celles-ci peut comporter jusqu'à 11 caractéristiques définies. Si nécessaire, les 32 unités peuvent être utilisées pour produire une seule note. Cette seule fonction, aux mains d'un utilisateur qualifié, donne au PDSG richesse et variété sonores.

Si une seule unité de génération est affectée à chaque note, une séquence programmée peut être composée de 32 lignes individuelles. De plus, une portion des générateurs peut servir à produire une partie séquentielle et les autres peuvent jouer en temps réel pour accompagner la séquence. Les caractéristiques des signaux sont affichées à l'écran, ce qui permet d'analyser les sons visuellement — un outil précieux lors d'une composition et un facteur qui rend le PDSG parfaitement adapté à l'enseignement de la musique. L'ensemble de génération sonore — adéquat pour un travail séquentiel non exécuté en temps réel ou pour l'analyse et la création de signaux — coûte à lui seul quelque 2 400 F, le clavier coûte environ le même prix.

Le principal inconvénient du PDSG est sa représentation sonore lors de l'étape de conversion numérique en analogique. L'oreille et le cerveau humains peuvent interpréter des sons allant

de 20 Hz à 20 kHz. Les sons naturels, y compris ceux produits par des instruments musicaux acoustiques, sont actifs sur la totalité de cette plage de fréquences et plus. Le PDSG ne peut représenter des sons que jusqu'à 12 kHz. Sa qualité sonore est donc comparable à un système hi-fi domestique et les fabricants supposent qu'une chaîne hi-fi sera utilisée pour compléter le système. La plupart des joueurs de synthétiseur seraient consternés s'ils étaient obligés d'employer ce type d'amplification. Clef Products prépare des versions du PDSG destinées à d'autres micro-ordinateurs.

MIDI a été perçu comme une innovation, parce qu'il donnait accès à de véritables synthétiseurs musicaux, aux propriétaires de micro-ordinateurs.

Un goût pour la musique

Le développement de l'interface MIDI offre de grandes possibilités musicales aux propriétaires d'ordinateurs domestiques. Mais, en même temps, le système peut être très coûteux — l'interface elle-même, le logiciel additionnel et un synthétiseur.

Il est possible de commencer avec un système musical non coûteux pour se familiariser avec les rudiments de la musique électronique. Un tel système doit évidemment être assez bon pour être satisfaisant sur le plan musical et stimuler un intérêt pour les possibilités plus évoluées de la musique électronique. Le programme sur cassette Ultisynth 64, produit par Quicksilva, est un bon ensemble pour débutant. Il exploite la puce SID (interface sonore) du Commodore 64 et ses trois oscillateurs.

Avec ce programme, chacune des touches du clavier du Commodore devient une commande indépendante de génération et de définition sonores. Les quatre formes d'onde élémentaires — sinus, carrée, triangle, et dents de scie — sont disponibles avec des réglages permettant de définir les caractéristiques d'attaque, de maintien, de chute et de suppression (enveloppe). Les sons peuvent être filtrés (c'est-à-dire qu'une bande de fréquences peut être soustraite de la sortie) pour définir le son de façon additionnelle. La modulation annulaire — un processus qui donne la somme et la différence de deux fréquences — est aussi incluse. Ceci est pratique pour produire des sons de cloche authentiques. En plus, le rythme peut être écrit, des rythmes prédéfinis peuvent être insérés et une séquence de 2 048 notes peut être construite.

Les fonctions de Ultisynth ressemblent beaucoup à celles du VCS 3, un synthétiseur « classique », créé à la fin des années soixante. Aujourd'hui, le prix de l'ensemble Ultisynth a beaucoup baissé, tandis que le VCS 3 est presque devenu une pièce de musée.

Romik's Multisound est un autre système offert sur cassette et destiné au débutant. Ses fonctions de commande sont semblables à celles de Ultisynth, mais il fournit l'affichage graphique d'un clavier musical. Les positions de ce clavier sont sélectionnées à l'aide d'un curseur et les notes sont définies par une information entrée au clavier alphabétique de l'ordinateur. Ceci met ce programme plus à la portée du musicien, en exploitant la familiarité que l'utilisateur peut avoir avec un clavier musical.

Les options du choix

Dans tout programme complexe, il y a des étapes où l'utilisateur doit choisir une option. Nous voyons ici le système logiciel avec menus et le système logiciel avec commandes.

Les menus peuvent être de simples listes d'éléments numérotés ou, à l'opposé, des écrans à fenêtres multiples. Cependant, le principe reste le même.

Le menu intervient lorsque le programme parvient à un embranchement dans le déroulement de son chemin logique. L'écran affiche la liste des choix possibles et le programme demande à l'utilisateur quelle option choisir. Les programmes avec menus sont arborescents.

L'avantage principal de cette approche logicielle est que l'utilisateur ne doit pas nécessairement connaître la structure du programme. Son cheminement est parfaitement balisé par ces panneaux indicateurs que sont les menus.

Cependant, lorsque le programme devient complexe et que le nombre de menus croît, l'expérience prouve que l'utilisateur, même averti, a du mal à s'y retrouver. En outre, avoir à passer par toute une succession de menus pour des tâches répétitives devient vite fastidieux. Les novices aussi peuvent éprouver des difficultés avec la structure de programme fondée sur des menus. En effet, pour revenir sur un mauvais choix, il faut nécessairement remonter jusqu'au menu comportant l'option incriminée et demander une autre option. Le traitement devra reprendre à partir de ce point. Une structure peut être plus ou moins ramifiée, avec un nombre plus ou moins grand de niveaux de décision. Plus ce dernier est important, moins la route à suivre s'accorde facilement d'hésitations et d'erreurs.

La conception des menus d'un programme est assez difficile. Le problème principal est de spécifier la totalité du programme et de ses déroulements possibles avant d'écrire une seule ligne de code. C'est un excellent exercice cependant, même si l'on n'y parvient pas immédiatement. Ajouter tardivement de nouvelles fonctions peut impliquer d'avoir à reprendre plusieurs menus plus haut dans le programme. La structure doit alors être profondément modifiée. Une fois le programme conçu, la logique de l'ensemble des menus doit être consignée dans une seule routine.

La routine de menu peut donc être considérée comme une forme complexe de la routine normale de commande, mais avec les embranchements internes contrôlés par l'utilisateur. Elle permet de mettre de l'ordre dans le programme en séparant la logique de commande des parties fonctionnelles du programme.

Le déroulement du programme se fait selon une organisation déterminée. Pour chaque menu rencontré, la routine de logique de menu trans-

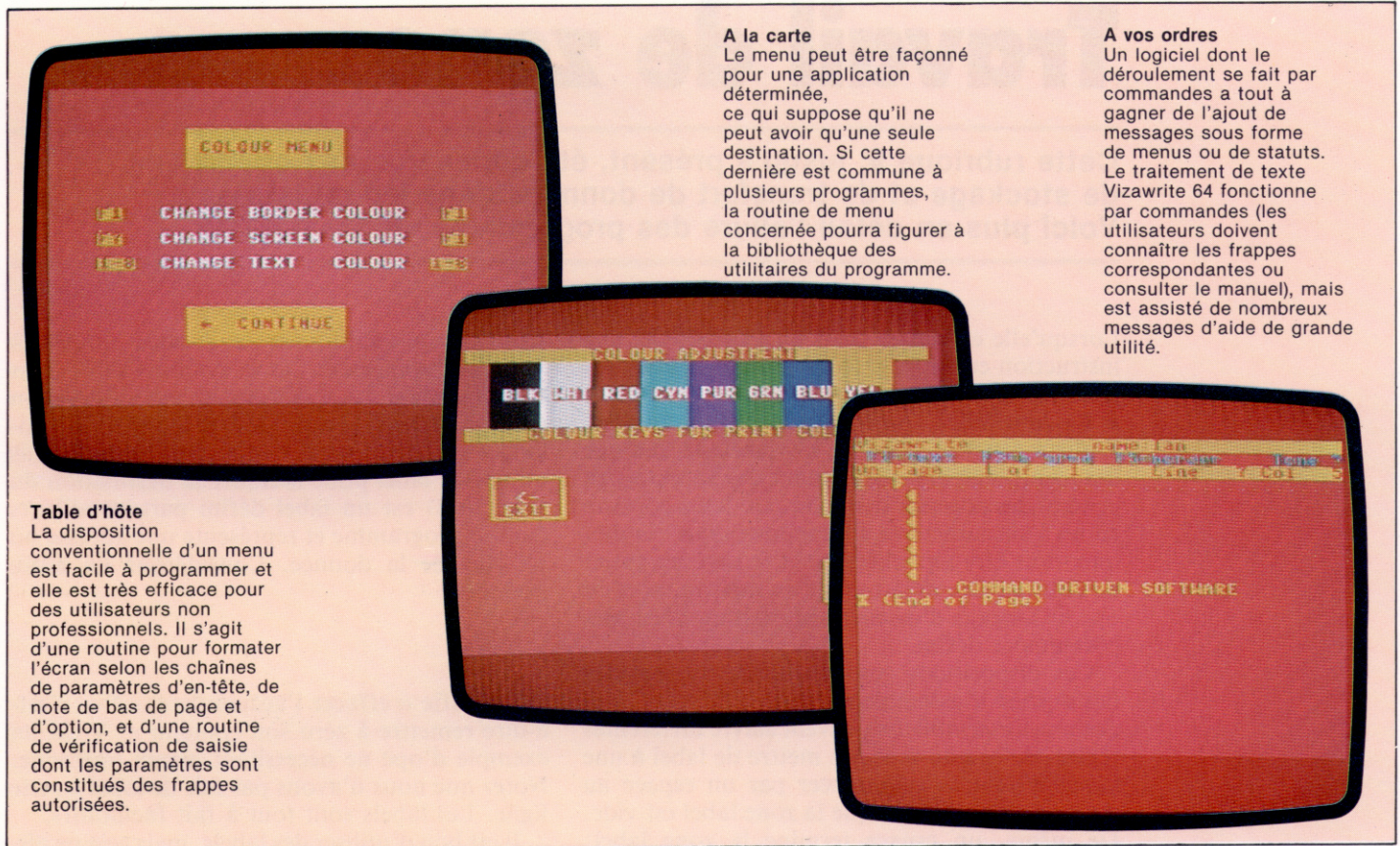
met un ensemble de messages guides-utilisateur à une routine qui les incorpore dans un menu vide. Les cases du menu comportent généralement un message en haut de l'écran, qui affiche le titre du menu et d'autres informations qui lui sont spécifiques, une note en bas de l'écran qui explique comment choisir une option et les options elles-mêmes. Le dessin sur écran le plus efficace pour un menu réunit habituellement jusqu'à huit options sur une colonne, le code de réponse s'inscrivant à leur gauche.

La routine de menu appelle une routine de saisie; elle lui transmet parfois les conditions de validité de la saisie et reçoit en retour la réponse de l'utilisateur. Elle interprète cette dernière. Soit elle donne alors les commandes au prochain menu, soit elle appelle la routine d'application spécifiée par l'option demandée. Ce dernier cas correspond à la rencontre du dernier menu de la chaîne. Les menus nécessitent beaucoup de texte, pour les en-têtes, les notes de bas de page et les messages, mais ce texte est, pour une bonne part, toujours le même. Les explications d'ordre général sur une option (la commande « aide »), sur l'option de sortie du menu en cours et de retour au menu général et sur les autres choix fréquents sont utilisables par de nombreux menus. Il est alors possible de gagner de la place dans le programme en réunissant ces informations constantes, sous la forme d'un tableau-chaîne de caractères. Ces dernières peuvent également figurer en tant que fichier sur un disque à accès direct. Vous pouvez maintenant concevoir la routine d'affichage d'un menu, de sorte qu'elle accepte les références à ce tableau et soit à même d'afficher les en-têtes, notes de bas de page et messages appropriés.

Un système logiciel contrôlé par commandes permet à tout moment l'accès à un certain nombre d'options. Chaque commande appelle directement un sous-programme, qui exécute la tâche voulue. Ce système doit pouvoir vérifier toute saisie au clavier pour distinguer les données des commandes. La différence étant que les commandes supposent la frappe simultanée d'une touche-repère et d'une touche spécifique.

Dans un système avec commandes, les choix sont très nombreux pour chaque rare embranchement. L'arbre-type est alors très ouvert. Une seule routine est utilisée pour contrôler le programme et guider l'utilisateur vers le sous-programme voulu.

Cet interpréteur de commandes a quatre fonctions principales. La première est simplement



A la carte
Le menu peut être façonné pour une application déterminée, ce qui suppose qu'il ne peut avoir qu'une seule destination. Si cette dernière est commune à plusieurs programmes, la routine de menu concernée pourra figurer à la bibliothèque des utilitaires du programme.

A vos ordres
Un logiciel dont le déroulement se fait par commandes a tout à gagner de l'ajout de messages sous forme de menus ou de statuts. Le traitement de texte Vizawrite 64 fonctionne par commandes (les utilisateurs doivent connaître les frappes correspondantes ou consulter le manuel), mais est assisté de nombreux messages d'aide de grande utilité.

Table d'hôte
La disposition conventionnelle d'un menu est facile à programmer et elle est très efficace pour des utilisateurs non professionnels. Il s'agit d'une routine pour formater l'écran selon les chaînes de paramètres d'en-tête, de note de bas de page et d'option, et d'une routine de vérification de saisie dont les paramètres sont constitués des frappes autorisées.

d'attendre la saisie. La seconde fonction de l'interpréteur est de trier la saisie. La troisième est d'interpréter la commande en préparant l'appel des sous-programmes appropriés. Les questions à résoudre sont alors : « Quelle est l'adresse de la routine ? » « Y a-t-il des paramètres à transmettre ? » En dernier lieu, l'interpréteur doit appeler la routine à exécuter.

Le format d'une commande peut varier en complexité. Certains langages de commande s'apparentent à des formes simplifiées du langage humain. Prenons par exemple le langage du système d'exploitation Unix. Le format typique d'une commande pour ce langage est :

Commande + liste des paramètres optionnels
L

ou encore :

L-1

Pour cet exemple, L demande la liste des fichiers au répertoire et L-1 (-1 est un paramètre optionnel) la même liste, mais selon un format détaillé.

L'analyse lexicale, ou tri, doit permettre de reconnaître les divers éléments de la ligne de commande. Unix simplifie les choses en considérant que le premier élément représente la commande et en reconnaissant les paramètres au signe moins qui les précède. Les paramètres d'un langage de commande sont destinés aux sous-programmes qu'il appelle. Les routines du système de commandes doivent exiger un même format pour les

paramètres saisis. L'interpréteur des commandes peut alors les transmettre sous la même forme. Il est manifestement plus facile de créer un interpréteur de commandes que d'écrire un système de menus.

Les utilisateurs expérimentés préfèrent généralement les systèmes avec commandes aux systèmes avec menus, du fait de leur plus grande rapidité et de leur plus grande souplesse d'utilisation. La plupart des systèmes d'exploitation sont régis par des commandes, ce qui ne favorise pas les utilisateurs débutants. En effet, ils ne donnent aucun point de repère et les routines d'aide en cours d'utilisation supposent une certaine connaissance du système. En outre, le nombre de commandes et de paramètres optionnels d'un système de commandes est tel, que même les utilisateurs avertis ont fréquemment besoin des routines d'aide et du manuel d'utilisation.

Les débutants détestent les commandes et les utilisateurs chevronnés ne peuvent pas souffrir les menus. Ce problème d'incompatibilité est pratiquement sans issue bien que des systèmes hybrides aient été expérimentés. Ainsi Wordstar, un logiciel de traitement de texte, est essentiellement à base de commandes, mais il peut se présenter à l'utilisateur sous forme de menus. Les commandes sont des codes de contrôle et l'utilisateur peut piloter le programme exclusivement avec elles. Les menus affichés utilisent ces commandes en tant que mnémoniques pour sélectionner des options. Ainsi, en même temps qu'il utilise les menus pour le déroulement du programme, l'utilisateur novice apprend les commandes.

Deux dans trois
Il y a plusieurs manières d'aborder le problème de la présentation d'un menu. Ces trois photos d'écran mettent en œuvre trois méthodes et illustrent comment l'utilisateur peut modifier les couleurs d'écran, de marges et de texte sur un Commodore 64. (Cl. Ian McKinnel.)

Travail de zone

Cette rubrique a, jusqu'à présent, été consacrée aux opérations de stockage et de transfert de données dans les registres. Voici plus en détail l'allure des programmes en assembleur.

Lorsqu'elle est entrée dans un assembleur, une instruction en langage d'assemblage consiste en trois parties, ou *zones* :

- **La zone label**, qui se trouve dans la colonne le plus à gauche de l'écran. Les labels ne sont que des identificateurs représentant des nombres, en général des adresses mémoire. Un label consiste en un à six caractères alphanumériques : le premier doit être alphabétique et le nom en entier ne doit pas être déjà celui d'un registre, ni d'un opc de langage d'assemblage, ni d'un label précédemment défini.

Ces dispositions constituent les conventions habituelles d'assembleur, mais différents programmes assembleurs peuvent suivre différentes règles. Si l'on ne veut pas mettre de label à une ligne, il faut la commencer par un espace au moins, pour indiquer que la zone label est vide. De même, un espace termine la zone label; LABELDA, par exemple, est un label variable, tandis que LAB LDA est interprété comme le label LAB suivi de LDA, l'opc (code opération).

L'assembleur comporte un *compteur d'emplacement*, qui équivaut au registre compteur de programme du processeur. Il garde l'adresse de l'emplacement mémoire où le prochain octet d'instruction ou de donnée doit être stocké. La première fois que l'assembleur rencontre un label, il sauvegarde l'identificateur dans une zone de mémoire appelée *table des symboles* — analogue à un tableau de programme BASIC. Il stocke également l'adresse contenue dans le compteur d'emplacement au point où le label a été rencontré la première fois. Chaque fois que l'assembleur rencontre un label dans le programme en langage d'assemblage, il cherche ce label dans la table des symboles. S'il s'y trouve, alors l'assembleur le remplace par l'adresse qui lui est attribuée.

- **La zone opérateur instruction ou opc** se trouve à droite de la zone label. C'est un mnémonique, généralement constitué de trois caractères et — si nécessaire — d'un nom de registre. Ainsi, ADDA est formé du mnémonique ADD et du nom de registre A. L'opc représente l'opération de processeur à exécuter.

- **La zone opérande ou adresse**, qui renseigne sur la donnée sur laquelle l'opc doit agir. Normalement cette donnée est sous la forme d'une adresse ou plus souvent d'un label représentant une adresse. Considérons cette instruction en langage d'assemblage :

```
LABEL1 ADDA NUM1
```

signifiant « additionner la donnée stockée à l'adresse représentée par le symbole NUM1 dans l'accumulateur A ». L'adresse de cette instruction est stockée sous le nom de LABEL1; si nous voulons y sauter ou nous y brancher, nous devons utiliser ce label pour indiquer la destination du saut. NUM1 est un label défini partout ailleurs dans le programme et représente une adresse où est stockée la donnée. Considérons un autre exemple :

```
CLRA
```

qui signifie « effacer l'accumulateur A — c'est-à-dire remettre à zéro son contenu ». Voilà un exemple d'opc ne nécessitant aucun opérande. Notez que nous n'avons pas mis de label à cette ligne. Les labels sont tout à fait facultatifs.

Il est bon d'utiliser des labels, mais cela ne dispense pas de fournir des commentaires plus explicites. Ceux-ci peuvent être ajoutés à n'importe quelle ligne, en laissant un espace après le dernier caractère de l'opérande avant d'insérer votre commentaire. Certains assembleurs exigent un caractère spécial pour indiquer le début d'un commentaire et, habituellement, on commence une ligne de commentaire par un astérisque.

Les constantes, sous la forme de nombres ou de chaînes de caractères, peuvent être utilisées dans la zone opérande. Les nombres sont généralement en système décimal, sauf s'ils comportent un préfixe \$ ou un suffixe H indiquant le système hexadécimal (par exemple \$AF08, AF08H); ou octal (base huit) avec le préfixe @ ou le suffixe Q (@6712, 6712Q); ou binaire avec le préfixe % ou le suffixe B (%11010011, 11010011B). Le code ASCII d'un caractère peut être utilisé comme un opérande en le préfixant avec une apostrophe : ainsi, 'A signifie 65 ou \$41.

Une valeur particulièrement utile est le contenu du compteur d'emplacement. On ne le connaît en général pas lorsqu'on entre le programme, mais on peut s'y référer dans la zone opérande par un astérisque. La plupart des assembleurs acceptent cela dans des expressions arithmétiques simples — généralement restreintes à l'arithmétique et à la soustraction. Par exemple :

```
LDA *+5
```

signifie « charger l'accumulateur avec le contenu de l'emplacement mémoire dont l'adresse est de cinq octets plus haut que le contenu actuel du compteur d'emplacement ».



Directives d'assembleur

Un assembleur accepte normalement un certain nombre de *directives* ou *pseudo-op*, qui sont utilisés dans le programme comme des *opc* normaux. Nous en avons déjà utilisé deux :

```
CR          FCB      13
```

FCB (fixe un octet constant) réserve un seul octet à l'adresse de compteur d'emplacement en cours et lui donne la valeur de l'opérande — ici, l'emplacement dont l'adresse est représentée par le symbole CR est initialisé par 13.

```
MEMTOP     FDB      $7FFF
```

FDB (fixe double octet) fait la même chose pour une valeur à deux octets (16 bits). De l'espace mémoire peut être réservé en blocs, sans fixer la valeur de son contenu, en utilisant le pseudo-op RMB, comme dans cet exemple :

```
TABLE1     RMB      7
TABLE2     FCB     $F6
```

qui réserve sept octets pour une table de valeurs dont le premier octet a une adresse représentée par le label TABLE1. En pratique, cela implique que si TABLE1 représente l'adresse \$C104, par exemple, alors TABLE2 représente une adresse de sept octets plus haut, c'est-à-dire \$C10B.

Une chaîne de caractères entière peut être placée en mémoire, comme ceci :

```
ERRMSG     FCC      'ERROR'
```

Cela initialise cinq octets de mémoire avec les codes ASCII pour E,R,R,O et R. Par conséquent, il est bien plus facile d'inclure des messages et suggestions-utilisateur dans les programmes en langage d'assemblage.

ORG (ORiGine) est un important pseudo-op, qui spécifie une valeur pour le compteur d'emplacement et est utilisé au début d'un bloc de programme pour indiquer à l'assembleur à quel endroit de la mémoire il faut commencer à placer le programme lorsqu'il est traduit en langage machine. Il faut toujours commencer les programmes par l'instruction ORG, bien que certains assembleurs puissent fournir une valeur par défaut, sinon il est possible, voire souhaitable, d'avoir plus d'une instruction ORG dans un programme. Une directive ORG ne prend ni label ni opérande.

La seule instruction que vous pourrez être amenés à mettre avant ORG est EQU (égal), puisqu'elle spécifie des symboles variables et ne se réfère pas au compteur d'emplacement. Par exemple :

```
RESET      EQU      $100
```

signifie que le symbole RESET est défini comme représentant la valeur \$100. RESET est ainsi une abréviation pour un nombre, alors qu'un label placé au début d'une ligne d'instruction repré-

sente l'adresse de la donnée ou du langage machine.

Enfin, la dernière directive qu'il nous faut considérer est END, qui est placée à la fin du code source pour l'indiquer à l'assembleur. Comme ORG, il ne prend ni label ni opérande.

Modes d'adressage 6809

Une mesure de la puissance d'un langage d'assemblage est constituée par ses modes d'adressage. Le processeur 6809 en comporte un bon nombre. Les instructions que nous avons vues jusqu'à présent utilisaient toutes soit le *mode direct*, soit le *mode étendu*, impliquant que la valeur ou le label dans la zone opérande est l'adresse de l'emplacement mémoire qui contient la donnée.

L'adressage direct ne spécifie qu'une seule adresse d'octet pour l'opérande d'instruction. Celui-ci est traité par le processus comme l'octet lo d'une adresse d'opérande à deux octets et elle prend comme octet hi de l'adresse le contenu du registre de page direct, registre de l'UC à 8 bits adressable par le programme. L'avantage de ce mode est la flexibilité et la généralité.

L'adressage étendu spécifie une adresse à deux octets comme opérande d'instruction. Cette instruction se référera alors toujours à cet octet particulier de mémoire; c'est donc un élément figé du code. L'assembleur distingue les modes direct et étendu par la nature de l'opérande.

Un autre mode habituel est le mode immédiat, où la donnée en cours est contenue dans la zone opérande elle-même. Ce mode est indiqué en préfixant l'opérande par un #. Par exemple :

ORG \$1000	l'adresse de début du langage machine est \$1000
NUM1 FDB \$FFFF	initialise l'emplacement NUM1 avec \$FFFF
LDD NUM1	charge \$FFFF, contenu de NUM1 dans le registre D
LDD #NUM1	charge \$1000, valeur réelle (ou immédiate) de NUM1, dans le registre D

Notez que le label NUM1 représente l'adresse \$1000. On pourrait penser que l'instruction ORG réside à l'adresse \$1000 et que l'instruction suivante (donc son label, NUM1) a une adresse plus élevée. Mais souvenez-vous que ORG est une instruction au programme assembleur concernant la traduction du programme en langage d'assemblage, et non une partie du programme lui-même. C'est pourquoi elle n'occupe pas d'espace mémoire; NUM1 prend la valeur \$1000 parce que c'est celle du compteur d'emplacement la première fois que NUM1 est rencontré par l'assembleur. Vous remarquerez aussi que LDD NUM1 charge le contenu de l'emplacement NUM1 (\$FFFF comme spécifié par la directive FDB) dans l'accumulateur D, tandis que LDD #NUM1 charge la *valeur* de NUM1 elle-même (l'adresse \$1000 du label).



Petits Japonais

Casio, surtout connue pour ses montres et ses calculatrices, ces dernières années a lancé toute une gamme d'ordinateurs de poche qui devrait lui permettre de s'imposer sur un marché difficile.

Casio déclare détenir près de la moitié du marché mondial de la calculatrice; la compagnie n'emploie pourtant, bizarrement, que 3 300 employés. En 1983, son chiffre d'affaires atteignait 29 millions de dollars, ce qui peut paraître assez peu pour une entreprise de ce type. Les responsables expliquent volontiers : « Nous sommes une petite compagnie très conservatrice, et les grosses campagnes de publicité ne sont pas notre style. »

Casio fut fondée par les cinq frères Kashio peu après la guerre, sous le nom de Kashio Seisakujo. Elle produisait alors des équipements de bureau. Au début des années cinquante, Casio lança la calculatrice électrique 14-A, l'une des premières du genre : elle occupait l'espace d'un bureau entier, et pesait 130 kg ! Mais les recherches en ce domaine se poursuivirent, et en 1957 la firme prit son nom actuel, celui de Casio Computer Company Ltd.

Une filiale britannique fut fondée en 1974 et proposa aux acheteurs de nombreuses calculatrices électroniques bon marché.

Le FX-702P, premier ordinateur de poche de la société, fut lancé en 1982. Il était avant tout destiné à des usages scientifiques et son clavier avait une disposition de type alphabétique (et

non QWERTY ou AZERTY, comme c'est le cas ordinairement).

Peut-être cela explique-t-il l'échec commercial de l'engin; en tout cas son successeur, le FX-700P, se vit équiper d'un clavier standard adapté à chaque marché spécifique.

Il fut suivi du PB-100, ordinateur de poche destiné à l'homme d'affaires. Très semblable au FX-702P, il disposait d'un affichage à cristaux liquides et d'un clavier numérique. Casio fournissait par ailleurs des magnéto-cassettes, des imprimantes et des modules RAM pour l'appareil et pour le FX-702P.

Qui va lentement va sûrement

Le FX-750P a récemment succédé au PB-100. Il est équipé de prises permettant d'accueillir deux cartes RAM de la taille d'une boîte d'allumettes, capables de contenir jusqu'à 4 K de données. Chaque carte est pourvue d'une batterie de 3 V, et la mémoire peut ainsi être sauvegardée lorsque le module est séparé de l'ordinateur. Les programmes sont, de cette façon, enregistrés et rappelés à volonté.

Chaque batterie a une durée de vie de un an et, lorsqu'elle doit être remplacée, les programmes qu'elle protège sont rechargés à partir d'un lecteur de cassettes.

Casio produit aussi le FP-200 : c'est un ordinateur portable doté de 8 K de RAM (extensible à 32 K). Il a un affichage à cristaux liquides de 8 lignes de 20 caractères en mode texte, tandis que la résolution graphique est de 160 x 64. Le SL-800, tout nouveau, est une calculatrice de poche très bon marché, qui a à peu près les dimensions d'une carte de crédit (ses composants sont imprimés sur un film plastique, et non soudés sur un circuit conventionnel).

En Extrême-Orient, la firme distribue des ordinateurs de gestion et des appareils aux standards MSX, destinés à l'usage personnel.

En Europe, en revanche, elle semble se limiter essentiellement à la vente de calculatrices et de petits ordinateurs. Sans doute ses responsables estiment-ils qu'il vaut mieux accroître lentement la part de marché qu'ils détiennent et passer ensuite progressivement à des appareils plus perfectionnés.

Enfin, Casio a mis au point divers engins construits autour de l'interface numérique MIDI, qui devraient être mis en vente très bientôt sur le marché.



Le centre de recherche et de développement de Casio, à Tokyo

Tadao Kashio, Président de Casio, est l'un des cinq frères Kashio qui contrôlent l'ensemble des activités de la compagnie.