



Prix : 12 F

MICHEL GANTIER

PLUS LOIN AVEC... LE CANON X07

MICHEL GANTIER

**PLUS LOIN
AVEC ...
LE
CANON X07**

SORACOM
COMPTER
INFORMATIQUE

**PLUS LOIN
AVEC LE
CANON X07**

MICHEL GANTIER

PLUS LOIN AVEC
LE CANON X07

DIFFUSION

SORACOM INFORMATIQUE
10, AV DU GL DE GAULLE
35170 BRUZ

PREFACE

Associé à l'imprimante traceur X-710, le CANON X-07 forme un ensemble compact et complet pour découvrir le monde de la micro-informatique.

Toutefois, point n'est besoin d'avoir l'imprimante X-710 pour débuter dans la programmation (cependant, personnellement je n'envisage pas de longues heures de programmation sans cet accessoire indispensable).

Le Canon X-07 est un micro-ordinateur individuel qui fait partie de cette grande famille des ordinateurs familiaux qui s'agrandit de jour en jour, et bien qu'il puisse se tenir dans la main, les possibilités qu'offre son "BASIC" peuvent faire pâlir de jalouse pas mal de ses compères.

Un "Basic" signé CANON et MICROSOFT, ce qui pour plus d'un amateur éclairé est signe de qualité.

Effectivement, le jeu d'instructions qui nous est offert est très complet et je vous proposerai par la suite de voir les particularités de certaines, peu classiques, au moyen de quelques exemples d'application.

DECOUVERTE DU CANON X07

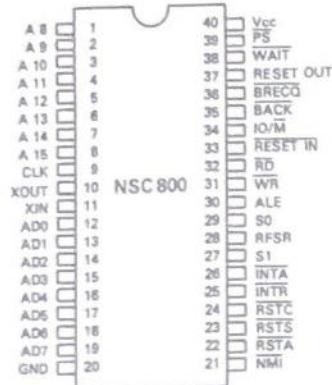
Le micro-ordinateur est architecturé autour d'un microprocesseur en technologie CMOS. Il s'agit du NSC 800 dont vous pouvez voir le brochage si contre.

La technologie CMOS permet à cet appareil d'être alimenté par piles crayon, il n'est donc pas vorace en énergie.

Le NSC 800 semble en fait un enfant du mariage des célèbres microprocesseurs 8085A et Z80 puisqu'il possède le jeu d'instructions du Z80 et la caractéristique d'avoir un bus donnée multiplexé avec le bus adresse comme le 8085. Toutefois son brochage ne ressemble en rien à ses ainés, c'eut été trop simple, bien sûr !

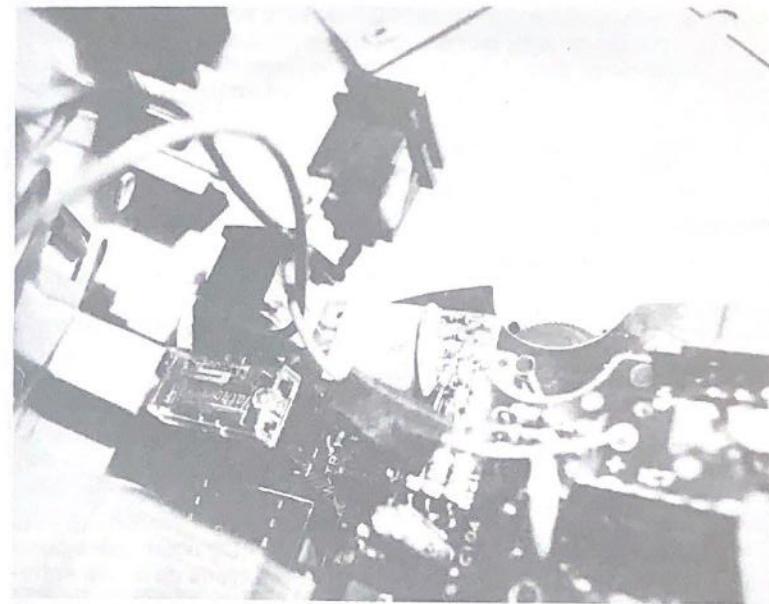
Le NSC 800 possède une entrée de mise en sommeil commandée par la broche 39 (signal PS), dans ce mode, la consommation est encore plus réduite.

N.B. : la sortie horloge du NSC 800 équipant le X-07 offre un signal de période 0,535 µs, soit une fréquence de 1,87 MHz.



AD ₀ -AD ₇	Bus données et adresses poids faibles multiplexées
A&A ₁₅	Bus adresses poids forts
ALE	Validation adresse de poids faibles
S ₀ - S ₁	Code d'état
RD	Signal de lecture
WR	Signal d'écriture
BREQ	Demande de libération des bus (DMA)
BACK	Accusé de réception de BREQ
INTR	Demande d'interruption
INTA	Acquittement du signal INTR
RST A, B,C	Lignes d'interruptions vectorisées
NMI	Interruption Prioritaire non masquable
RESET IN	Initialisation
RESET OUT	Acquittement d'une RAZ
WAIT	Signal d'allongement (mémoires lentes)
XOUT, XIN	Entrées du quartz (ou RC)
CLK	Sortie horloge
I/O/M	Sélection mémoires ou périphériques
RFSR	Signal d'échantillonnage du rafraîchissement
PS	Signal de mise en sommeil (consommation réduites)

La structure interne et les instructions sont celles du Z80. C'est un circuit du type CMOS. La tension d'alimentation peut donc être comprise entre 3 et 12 volts, et la consommation est d'environ 50 mW sous 5 volts.



Relais magnéto situé près du réglage lumière de l'écran LCD

Ce qu'il faut savoir sur votre CANON X-07.

Tout d'abord, l'alimentation se fait par pile (4 piles R6, soit une tension d'alimentation de 6 V).

La consommation, ordinateur coupé par OFF, est de 0,12 mA. Ce n'est pas très élevé et cela sert à maintenir sous tension les circuits de veille et notamment le fonctionnement du circuit d'horloge incorporé.

Vous appuyez sur ON, la consommation s'élève à 6 mA.

Si vous faites RUN d'un programme, vos piles débitent alors 16 mA. Le CANON X-07 permet la télécommande d'un magnétocassette pour l'enregistrement et la lecture d'un programme. Cela consiste à fermer une boucle sèche par un relais interne au X-07 ; ce relais augmente la consommation de 26 mA environ.

Le manuel "guide utilisateur" signale que l'on peut brancher un bloc alimentation secteur AD-4II, toutefois si vous avez l'intention d'acheter l'imprimante X-710, sachez que celle-ci, via son cordon la reliant au CANON X-07, fournit du 6 V à l'ordinateur. Cette même imprimante possède un accumulateur incorporé (6 V) et est elle-même fournie avec sa chargeur.

En laissant branché en permanence l'ordinateur et l'imprimante X-710, même si celle-ci est à l'arrêt, le X-07 recevra son alimentation 6 V par l'imprimante et ainsi on ne court pas le risque de perdre les données en mémoire dans l'ordinateur par la faute de piles usagées. Bien sûr, je ne saurais trop vous recommander de laisser des piles en bon état à l'intérieur du X-07... le cordon peut se débrancher. L'ordinateur X-07 signale une tension d'alimentation trop faible (LOW BATTERY) et si cela arrive, le fait de brancher la X-710 peut vous permettre d'échanger les piles sans risque aucun de perdre votre précieux programme auquel vous venez de consacrer des heures de réflexion...

La X-710 n'est pas la seule imprimante connectable au X-07. Celui-ci dispose d'une prise compatible CENTRONICS et qui permet donc, via l'achat d'un câble adaptateur, de "tirer" vos listings sur la majorité des imprimantes à interface CENTRONICS parallèle (EPSON, MICROLINE, OKI...).

De la même façon, votre X-07 dispose d'une sortie série dite conforme aux spécifications RS232C, et moyennant un adaptateur que je vous proposerai de réaliser — si tant est que vous touchiez un peu au "hard" — vous pourrez là aussi utiliser certaines imprimantes qui disposent d'une entrée série RS232C (MICROLINE 84, SILENT 700, TEXAS...).

Réaliser un adaptateur parallèle CENTRONICS n'est guère compliqué puisqu'il s'agit seulement de relier deux types de prises entre elles. Celle un peu spéciale de CANON (vous aurez sans doute des difficultés pour la trouver chez votre revendeur habituel — CANON fabrique le cordon complet à un prix dépassant nettement celui d'un câble fait soi-même), et l'autre qui est la classique CENTRONICS parallèle 36 points dont la référence est :

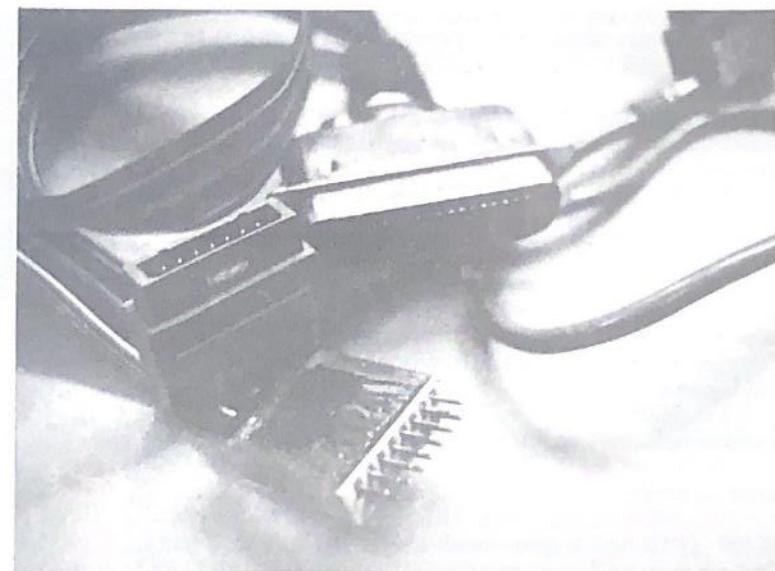
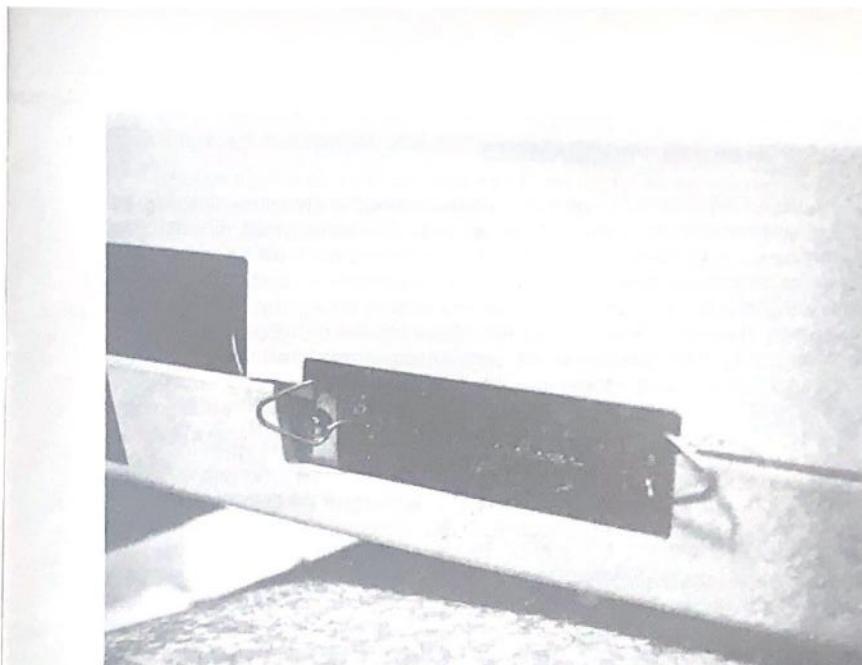
AMPHENOL 57 303 60

En ce qui me concerne, j'ai pris le taureau par les cornes et je me suis fabriqué la prise "CANON" ; si cela vous intéresse, je vous indique ici mon procédé.

Vous connaissez sans doute le wrapping, et on peut constater que ces queues de wrapping s'insèrent parfaitement dans les parties femelles des connecteurs du CANON X-07, aussi, en soudant de façon à respecter la disposition du connecteur X-07, les différentes broches en les coupant à longueur, tout cela sur un support de verre époxy, on se fait un connecteur mâle ; après on soude les fils du câble, on enroule le tout dans la résine époxy (la colle époxy à 2 composants convient parfaitement), et hop, on dispose de son connecteur (pour mouler la résine époxy, j'ai utilisé de la pâte à modeler... mais oui !).

De la même manière j'ai fait un câble pour connecter la prise série et réaliser un adaptateur RS232C.

Cet adaptateur pourra vous permettre de communiquer plus facilement avec le monde extérieur.



QUELQUES PARTICULARITÉS POUR ÉDITER LES PROGRAMMES

Votre CANON X-07 ne peut vous permettre de créer des lignes de programme dont la dimension excède 80 caractères. C'est bien dommage. La limitation est due au fonctionnement de l'écran LCD. De plus, si vous mettez au point un programme, à coup sûr il faudra intervenir par ci par là pour modifier les ordres inclus dans les lignes ; alors pas question dans ce cas de dépasser 60 caractères par ligne (en fait il faut compter aussi les caractères composant le numéro de ligne et l'espace séparateur ajouté derrière, de façon systématique, par le X-07). Si la ligne porte le numéro 1290, il vous reste $60 - 5 = 55$ caractères disponibles seulement. C'est le grand handicap à mon avis de ce petit bijou japonais.

Ce que vous écrivez pour composer une ligne de programme, par exemple :

10 GOTO 20

est mémorisé en RAM dès l'appui de la touche RETURN, et va se placer après la dernière ligne dont le numéro est juste inférieur. Cette mise en mémoire s'accompagne d'une semi-compilation en ce sens que les mots réservés comme GOTO sont traduits en général sous forme d'un octet (8 bits), le numéro de ligne est codé en hexadécimal (2 octets). Exemple : la ligne 1023 sera codée 03FF (2 octets à la place de 4 caractères).

Toutefois cette compilation ne teste pas la validité de ce qui a été écrit ! Taper 10 ANE EST UN CHEVAL puis RETURN est accepté de bonne grâce !

Cependant, dès que vous faites RUN, le X-07 ne "digère" pas ce qui a été écrit et il le manifeste par le messages SN ERROR... erreur de syntaxe.

Certains BASIC (celui de SINCLAIR avec son fameux ZX81) refusent l'entrée d'une ligne de programme incorrecte et indiquent même l'endroit où se trouve l'erreur..., ce qui peut être bien pratique pour le débutant.

Le contrôle de syntaxe d'une ligne de programme par l'interpréteur se fait au moment de l'exécution de cette ligne de programme. Dans certains cas, on peut remarquer que cette analyse syntaxique est incomplète.

Ex.: PRINT "IMPRIME

sans le guillemet de fin de chaîne est accepté sans mot dire.

De même A\$ = "TOTO,
mais... attention,

10 PRINT"C'EST UNE EXEMPLE : GOTO 10

le GOTO 10 risque peu d'être exécuté, il faut dire que dans ce cas l'erreur est visible, puisque GOTO 10 sera affiché à la suite du texte. Si vous faites LISTI, le X-07 refuse, ne comprend pas que c'était LIST que l'on voulait faire (SN ERROR), mais faites DIRI, et hop, l'ordre DIR

est exécuté, puis le I vous donne une SN ERROR. Une erreur identique mais un comportement différent, n'est-ce pas ?

Il faut savoir également que dans une ligne où se trouve un branchement inconditionnel du type GOTO, tout ce qui suit ce même GOTO <numéro de ligne> est absolument indifférent au X-07, donc écrire ceci :

Exemple : 20 A=A+1:GOTO 60 CECI EST EN PLUS et même l'espace qui sépare 60 de CECI peut être absent ; le X-07 va assurer le GOTO 60 sans problème... une autre manière de concevoir les commentaires sans utiliser REM (gain mémoire).

Avec l'écran LCD, certaines erreurs peuvent paraître difficiles à déceler, surtout pour un programmeur débutant. Taper :

10 PRINT"BONJOUR X0"

20 GOTO 10

sans faire RETURN à la fin de la première ligne, condamne à une erreur dès qu'on lance l'exécution car la ligne 10 sera en fait constituée de

10 PRINT"BONJOUR X0"20 GOTO 10

Le listing du programme sur l'écran semblait pourtant correct. Cette erreur n'arriverait pas avec un écran vidéo 80 colonnes, bien sûr.

RÉUNIR DES LIGNES DE PROGRAMME

Votre X-07, acheté nu, ne possède que 8 k RAM au total, moins de 7 pour l'utilisateur en fait. C'est bien mais un peu juste pour de bons programmes, aussi faut-il ménager l'occupation mémoire. Deux petites lignes qui comportent une seule instruction peuvent avantageusement être réunies en une seule (de 60 caractères optimum) ; cela fait gagner quelques octets (ceux de l'adresse et du numéro de ligne, par exemple) et chacun sait que les petits ruisseaux font les grandes rivières

Position 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

exemple

10 INPUT A\$
20 GOTO 60
≥

≥ curseur

Pour réussir ces 2 lignes il suffira, par le déplacement du curseur, de se placer sur le \$ de A\$, puis de rejoindre le 2 du 20 GOTO..., on revient au caractère qui suit le \$ (position 11) et on agit 11 fois sur DEL. 20 GOTO 60 se déplace et finalement on obtient :

10 INPUT AS ≥GOTO 60

≥ curseur

A la place du curseur on tape : et les deux lignes sont réunies. On valide par RETURN.

Dans le programme se trouvent désormais les lignes :

```
10 INPUT A$:GOTO 60  
20 GOTO 60
```

La ligne 20 sera éliminée en tapant 20 puis RETURN : sépare deux instructions différentes dans une même ligne de programme.
Cette particularité peut être utilisée aussi dans une ligne qui comporte l'instruction IF... THEN.

On peut très bien écrire :

```
10 IF A=30 THEN B=20 : C=0 ELSE K=2 : GOTO 60
```

Ce qui suit THEN se compose de B=20 et C=0, deux affectation de valeur de variable (deux instructions).

Et ce qui sera fait si la condition n'est pas réalisée (ELSE), se compose de l'affectation de la variable K et du branchemet inconditionnel à la ligne 60. Autre exemple de ligne embrouillée... mais qui donne satisfaction.... Programmer IF THEN IF THEN...

```
10 FOR I=1 TO 20  
20 IF I MOD3=0 THEN LPRINT I;" /3" ELSE .I  
F I MOD2=0 THEN LPRINT I;" /2" ELSE PRINT  
I  
30 NEXT
```

```
2 /2  
3 /3  
4 /2  
5 /3  
8 /2  
9 /3  
10 /2  
12 /3  
14 /2  
15 /3  
16 /2  
18 /3  
20 /2
```

PRÉCAUTIONS A PRENDRE

Exemple : dans une boucle FOR NEXT

```
30 FOR J=1 TO 10  
40 IF A=10 THEN PRINT "10" ELSE PRINT "NON" : NEXT J
```

Le NEXT sera vu neuf fois mais par la dixième... parce que l'instruction NEXT J n'est pas exécutée si la condition A = 10 est réalisée... En général, éviter de placer un NEXT avec d'autres instructions.

Un NEXT doit toujours être précédé d'un FOR, sinon vous aurez droit à une erreur du type NEXT WITHOUT FOR (NEXT sans FOR), mais on peut très bien sortir d'une boucle FOR sans être passé une seule fois par l'instruction NEXT.

```
20 FOR I=A TO B  
30 IF A>50 GOTO 100
```

```
90 NEXT I  
100 ....
```

Ceci est valide, la ligne 90 ne sera pas exécutée si A est supérieur à 50 avant l'exécution de 20.

De même on peut sortir plus rapidement d'une boucle FOR NEXT en forçant la variable d'incrémentation.

Exemple :

```
10 FOR I=1 TO 10  
20 PRINT I;  
30 I=10  
40 NEXT I  
50 PRINT I
```

Nous verrons seulement s'afficher la valeur 1 sur l'écran LCD, résultat de l'exécution de la ligne 20 ; à la ligne 30, I prend la valeur 10 et va conduire à forcer la variable de boucle au-delà des limites dès le passage de la ligne 40. La ligne 50 va faire afficher 11 à la suite de 1.

Il faut savoir que la variable d'incrémentation d'une boucle FOR NEXT est toujours augmentée du pas d'incrémentation (positif ou négatif d'ailleurs) après l'exécution complète de la boucle.

Les messages d'erreurs que votre CANON signale lors de l'exécution de vos programmes sont en général formatés de façon compacte, par exemple NF ERROR, ce qui, grâce au manuel livré, peut se traduire par NEXT non précédé du FOR associé... ; NEXT WITHOUT FOR en anglais, mais vous aurez sans doute rencontré des messages plus complets, notamment lors de l'utilisation de la fonction INPUT, à savoir :

? REDO FROM START
et EXTRA IGNORED

Le premier de ces messages signifie tout simplement que vous devez recommencer depuis le début. L'interpréteur s'attendait à recevoir, par exemple, une valeur numérique, et on lui donne à manger de l'alphabétique !

Exemple : 50 INPUT A dans le programme et l'opérateur se trompe de touche et tape sur la touche O à la place de la touche 0.

DEUXIÈME MESSAGE : l'interpréteur attend une ou plusieurs données et on lui en donne plus que prévu.

Par exemple :

50 INPUT A

on répond : 1,7 à la place de 1.7.

1,7 pour l'interpréteur, ce sont les chiffres 1 et 7 séparés et non la valeur un et sept dizièmes ! Les anglais et américains n'utilisent pas la virgule pour séparer les décimales comme nous. Votre CANON X-07 suit les mêmes principes.

En conséquence de ce que vous avez tapé (1 et 7), ne sera attribué à la variable A que la valeur 1 et le 7 en supplément (EXTRA) sera éliminé sans autre forme de procès (IGNORED).

A l'inverse, un INPUT double ou triple comme :

50 INPUT A,B,C

dans un programme, fait que l'opérateur qui répond doit introduire les 3 données demandées en entrée, successivement séparées par une virgule.

? — signifie que le CANON attend quelque chose.

1,5, 7, .9 — réponse de l'opérateur.

les variables A, B et C prendront respectivement les valeurs 1,5, 7 et 0,9.

Toutefois, le fait d'omettre une donnée ne va pas gêner le CANON X-07, mais il se contentera de signifier son appétit insatisfait en renouvelant à sa manière qu'il désire encore quelque chose :

? 1,5, 7

?? — Deux points d'interrogation (il manque encore la troisième variable !).

Dans le déroulement d'un programme, vous noterez que dans les deux cas, le programme n'est pas "cassé". Cependant, le formatage que l'on désirait sur l'écran LCD sera sans aucun doute perturbé, et avouez que l'utilisateur du programme qui peut ne pas être habitué, sera perturbé par l'apparition d'un ? REDO FROM START en plein milieu d'un programme élaboré en bon français (le CANON dispose des caractères accentués en plus !).

Si c'est vous l'utilisateur, pas de quoi s'étonner mais un programme doit résister à toutes les fausses manœuvres que l'on peut faire sur le clavier ; cela est surtout vrai pour les jeux...

Dans les exemples de programme qui suivent, vous constaterez que la gestion des entrées est plus élaborée en général qu'un vulgaire INPUT afin de résister aux appuis malencontreux sur les touches nécessaires du clavier.

LE BASIC DU X07

L'interpréteur BASIC du CANON X-07, comme la majorité des interpréteurs, exécute les instructions en suivant les numéros de ligne. Quand il voit GOTO 100, l'interpréteur part de la ligne dont le numéro est le plus faible et "remonte" en sautant les numéros de ligne afin de trouver le numéro 100.

Cette façon de procéder implique que plus la ligne à chercher est loin du début, plus le temps d'exécution sera long. S'il s'agit d'une boucle répétitive ou d'une zone programme qui est souvent utilisée dans le programme principal, on voit de suite qu'il y aura intérêt à placer le sous-programme en début du programme principal ! La rapidité d'exécution ne peut que s'améliorer. De la même façon, les blancs (espaces) entre les instructions ne sont pas utiles la plupart du temps et permettent de gagner en rapidité d'exécution. Cependant la lecture du programme ne sera pas facilitée et les débutants trouveront avantage à séparer clairement les instructions d'une ligne de programme.

C'est pourquoi dans les exemples de programmes qui suivent, j'ai placé des espaces entre les instructions. Il est dommage d'ailleurs que dans les publications multiples qui concernent l'informatique individuelle, l'éditeur tolère l'impression de listings illisibles, ce qui est le contraire de leur vocation ! La rapidité d'exécution a bon dos...

Un bon programmeur doit faire des choses claires.

N'hésitez pas non plus à mettre des commentaires, si vous avez à retoucher un programme quelques mois après la dernière version, vous ne le regretterez pas.

N'oubliez pas quand-même que les commentaires prennent de la place en mémoire.

A ce propos il est intéressant de savoir que sur le CANON X-07, REM et ? signifient la même chose, mais pas en mémoire... ' occupe 3 fois plus de place ! Bizarre, non ?

Par contre ? et PRINT c'est strictement pareil, mais attention en utilisant ? dans une ligne de programme car après un LIST vous verrez PRINT en entier et gare aux 60 caractères fatidiques pour pouvoir réécrire la ligne !

Pour connaître votre consommation de mémoire, vous avez toujours la possibilité de faire ? FRE(X) (?=PRINT). X est un argument factice (DUMMY ARGUMENT pour les anglais). Cela indique l'espace mémoire restant disponible pour l'écriture. Cela tient compte de l'espace pris par les fichiers du type RAM.

Cependant votre programme consomme aussi de l'espace mémoire pour "tourner".

Vous trouverez une différence entre l'espace libre avant RUN et après. Exemple :

```
10 FOR I=1 TO 10
20 A(I)=I
30 NEXT I
? FRE(X)
```

Avant RUN : 6 714, après RUN : 6 607, avant d'entrer le programme : 6 748, soit 107 octets travail.

```
10 FOR I=1 TO 10:A(I)=I:NEXT I
```

Avant RUN : 6 726, après RUN : 6 619 (aussi 107 octets travail !). Notez le gain mémoire en compactant les lignes programmes.

Supprimer les espaces entre mots BASIC augmente la vitesse d'exécution, mais cela n'est pas le seul moyen.

Il faut, dans la mesure du possible, quand un calcul répétitif est à exécuter, utiliser la boucle FOR//...//NEXT, plutôt que I=I+1//...//GOTO début.

Exemple :

```
10 I=I+1          10 FOR I=1 TO 10000
20 IF I=10000 THEN GOTO 40      30 NEXT I
30 GOTO 10          40 END
40 END
58 secondes        18 secondes
```

Il faut utiliser dans la mesure du possible les variables entiers si les calculs impliqués dans le programme peuvent s'en contenter

Exemple :

10 TIME\$ = "00:00:00"	"	"
20 DEFINT I	20 DEFSNGJ	20 DEFDBL
30 FOR I= 1 TO 10000	"	"
40 NEXT I	"	"
50 ? TIME\$	"	"

14 secondes 32 secondes 36 secondes

Du simple au double en ce qui concerne le temps d'exécution ! Assigner une variable avec la valeur d'une constante souvent utilisée...

LES TOUCHES DE FONCTION

Vous connaissez sans doute des ordinateurs où l'on peut entrer tous les mots-clés du BASIC avec une touche réservée (exemple : le ZX81 SINCLAIR). Le CANON X-07 ne dispose pas de cette facilité, mais vous avez la possibilité de mémoriser 12 ordres à l'aide des touches définissables par l'utilisateur : F1 à F12. Ces touches permettent même le déroulement d'un petit programme... Les touches sont programmées par l'intermédiaire d'une variable chaîne spécifique comme KEY\$(12), 12 étant le numéro de la touche.

Exemple :

```
KEY$(12) = "...FOR I=1 TO 12:BEEPI,2:NEXT" + CHR$(13)
```

Dès la sélection de cette touche, l'écran affiche le contenu de la variable qui, s'il respecte la syntaxe correcte d'un ordre BASIC est exécuté en tant que tel (ici cela joue une partie de la gamme). L'absence de CHR\$(13) impliquerait l'appui de la touche <RETURN> pour exécuter l'ordre. En effet 13 est un code non admis dans une chaîne de caractères sans l'utilisation du CHR\$. Même chose pour les guillemets " qui délimitent une chaîne.

Si on désire programmer une touche qui afficherait, par exemple, CSAVE" avec le guillemet à la fin, il faut obligatoirement écrire pour la touche 5 exemple :

```
KEY$(5) = "csa CSAVE" + CHR$(34)
```

34 est le code décimal ASCII de ", csa est obligatoire et ne sert qu'à afficher le contenu de la touche sur l'écran LCD dans certains cas (ordre CONSOLE). Ces 3 caractères sont aussi nécessaires (bien qu'inutiles) pour les touches 6 et 12.

Après une mise en marche à la suite d'une initialisation complète (appui du bouton RESET sous le boîtier) ou un ordre CONSOLE , les touches fonctions ont des valeurs définies par le X-07. Certaines n'ont pas été programmées (exemple touche 6), ce qui, proposé par CANON, ne me semble pas particulièrement pratique ; aussi ai-je été amené à un jeu de touches fonctions qui est celui-ci :

INITIALISATION DES touches FONCTION

```

10 KEY$(1)="lisLIST"
20 KEY$(2)="runRUN"
30 KEY$(3)="gotGOTO "
40 KEY$(4)="forFOR "
50 KEY$(5)="nexNEXT "
60 KEY$(6)="RO "+CHR$(8)
70 KEY$(7)="tim?TIME$"+CHR$(13)
80 KEY$(8)="chrCHR$"
90 KEY$(9)="beeBEEP "
100 KEY$(10)="gosGOSUB "
110 KEY$(11)="retRETURN"
120 KEY$(12)="conCONSOLE0,4,0,1,1"

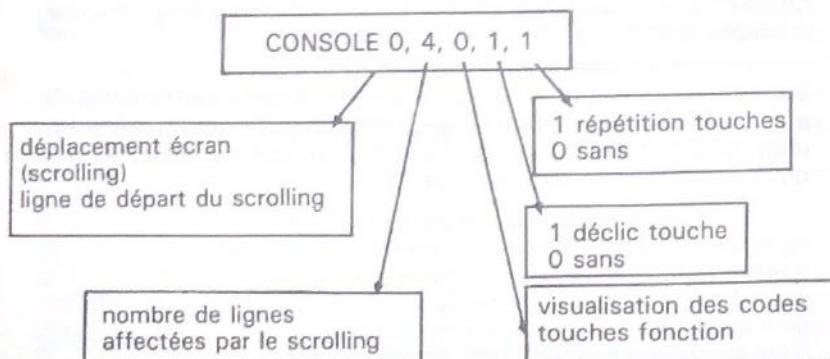
```

Le RUN de la touche F2 n'a pas le CHR\$(13) qui amènerait l'exécution immédiate de la fonction programmé.

Il me semble en effet dangereux de lancer un programme dès l'appui malencontreux de cette touche (l'utilisation des fonctions INIT fichier peut, dans certains cas, détruire des fichiers qu'on avait peut-être besoin de conserver).

La touche F6 est telle maintenant, qu'elle exécute un RUBOUT du caractère qui précède le curseur (pratique pour l'édition).

Le shift de cette touche (F12) permet facilement de modifier les paramètres de l'ordre CONSOLE avant d'appuyer sur RETURN pour valider.



Les autres touches sont affectées à des termes qui reviennent souvent dans l'écriture des programmes ; TIME\$ est là pour palier à l'arrêt de la montre du programme.

Vous constatez qu'une seule de ces touches provoque une action dès son appui, les autres nécessitent l'appui de la touche RETURN afin de terminer l'ordre ou le modifier.

La touche F6, quant à elle, utilise un code de contrôle du curseur (CHR\$(8)) et qui aussi a un effet immédiat.

CHR\$(8) fait partie de ces codes de contrôle qui sont obtenus par l'appui de la touche CNTL et d'une autre simultanément. Le code obtenu est toujours inférieur à 20H.

CNTL A=1	CNTL J=9	CNTL P=16
CNTL B=2	CNTL J=10	CNTL Q=17
CNTL D=4	CNTL K=11	CNTL R=18
CNTL E=5	CNTL L=12	CNTL T=20
CNTL F=6	CNTL M=13	CNTL U=21
CNTL G=7	CNTL N=14	CNTL V=22
CNTL H=8	CNTL O=15	CNTL X=24
		CNTL Z=26

et CNTL[= 27 Code de ESCAPE.

N.B. : noter que CNTLS et CNTLC sont toujours interprétés comme commande interrompant un programme.

CNTLS provoque une PAUSE dans le déroulement, .

et CNTLC un BREAK (arrêt du déroulement du programme).

La fonction INKEY\$ qui permet de connaître la touche qui a été appuyée puisqu'elle permet la scrutation du clavier, ne permet pas le test de CNTLC et CNTLS...

CNTL G dont le code est 7 (BELL) provoque un bip dans le HP du CANON.

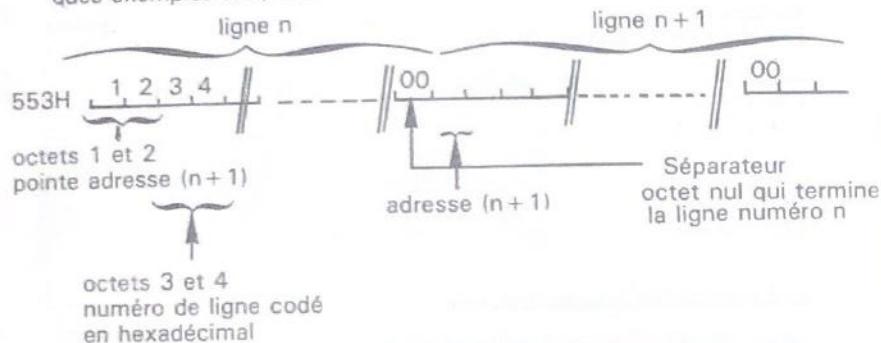
CNTL M est l'équivalent de la touche RETURN.

COMMENT EST PRÉSENTÉE LA LIGNE PROGRAMME EN MÉMOIRE ?

Chaque ligne de programme de la zone texte programme est placée derrière la précédente dans le sens des adresses croissantes de

mémoire, et séparée de la précédente par un octet de valeur nulle (00). La zone texte du CANON commence en 553H (H signifie valeur hexadécimale).

Les deux premiers octets qui composent la ligne représentent la valeur de l'adresse du début de la ligne suivante. Les deux octets qui suivent codent la valeur représentative du numéro de ligne. Ensuite les octets représentent les ordres BASIC contenus dans la ligne de programme ; certains étant, on l'a vu, déjà "compilés" et représentés par un octet spécial dont nous allons voir quelques exemples ci-après.



Exemple :

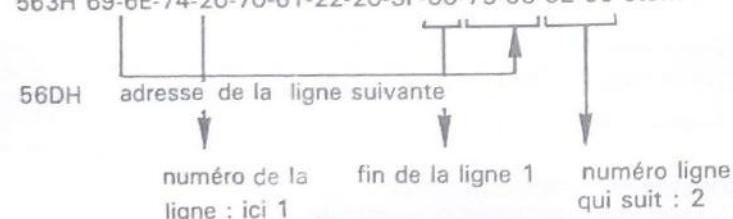
- Ligne n° 1 soit 0100
- Ligne n° 10 soit 0A00
- Ligne n° 256 soit 0001

A la dernière ligne du programme, le pointeur adresse deux octets dont la valeur est zéro, ce qui indique la fin du programme en zone texte.

Exemple :

553H 6D-05-01-00-9F-20-41-38-22-41-22-3A-8E-20-70-72

563H 69-6E-74-20-70-61-22-20-3F-00-75-05-02-00 etc...



Cette structure de la zone "texte" du CANON X-07 se retrouve de la même façon dans un fichier RAM type "P". Un fichier RAM type "P" est celui qu'on obtient par l'ordre SAVE "nom de fichier" (dans la mesure où un programme existe en zone texte). Cette structure est la même, y compris la valeur des adresses des pointeurs de ligne vue ci-dessus !

TRANSCRIPTION D'UNE LIGNE DE PROGRAMME EN MÉMOIRE "TEXTE"

Exemple :

FOR I=1 TO -20 STEP -2.5
81.20.49.DD.31.20.BB.20.D2.32.30.20.D0.20.D2.32.2E.35

On peut constater que les mots FOR, TO, STEP ainsi que les signes =, - sont codés sous forme d'un octet spécial (non représentatif ASCII des signes = et - notamment).

FOR → 81H
= → DDH
TO → BBH
- → D2H
STEP → DOH

Chaque mot-clé de BASIC est ainsi codé et remplacé en mémoire par un octet spécial réservé.

On constate aussi que tout ce qui est chiffre, valeur numérique, est codé sous sa forme représentative ASCII.

9 → 39H
0 → 30H, etc.

On peut aussi voir qu'il n'y a pas transformation en binaire de la valeur, du nombre comme certains autres micro-ordinateurs le font (exemple : ZX81).

Voici quelques exemples de lignes programme transcrisées :

RUN 10	80 20 31 30 (20 est le code de ESPACE)
GOSUB 10	8C 20 31 30
RETURN	8D
ON A GOSUB 30, 256	9B 20 41 20 8C 20 33 30 2C 32 35 36
RESTORE 30	8B 20 33 30
RESUME NEXT	99 20 82
RESUME 45	99 20 34 35
NEXT 1	82 20 49
ON ERROR GOTO 10	9B 20 98 20 88 20 31 30
RESTORE 41	8B 20 3A 31
INIT #1,	B4 23 31 2C
EXEC 8HEE1F	A8 20 26 48 45 45 31 46
INP (#1)	C3 28 23 31 29

```

PRINT A;"A":           9F 20 41 3B 22 41 22 3A
Noter : séparateur de ligne : (3A)

REM
'
?
IF A = T
=
THEN 130
PEEK
A = PEEK (1024)
POKE 65536,&H30

CHR$
+ CHR$(13)           D1 F6 28 31 33 29

```

Ces codes représentatifs des fonctions BASIC sont comme dit plus haut utilisés de la même manière dans un fichier RAM type "P", P comme programme.
En fait, le CANON, par l'ordre SAVE, recopie la zone texte relative au programme dans la zone réservée aux fichiers dont on peut connaître l'adresse et la longueur par l'intermédiaire des variables système RAMSTART et RAMEND (qui peuvent être connues en examinant le contenu de ces adresses, respectivement 210H et 212H).

Supposons un fichier RAM sauvegardé, et par l'ordre DIR on peut voir :

```

RAM
DUMP P  DUMP, nom sous lequel a été sauvegardé le programme
600/52

```

Si l'on examine le contenu des variables RAMSTART et RAMEND, on trouvera, par exemple, 1DACH et 1FF7H, soit une dimension de 587 octets (en effet 1FF7H - 1DACH = 587D). A ces 587 octets il faut ajouter 13 octets nécessairement utilisés par le système, soit un total de 600, valeur utilisée pour l'ordre FSET.

STRUCTURE DES FICHIERS RAM

Fondamentalement il existe deux types de fichiers en RAM : les fichiers dits "P" et les fichiers du type "D" (données qui ont une structure légèrement différente).

- Un fichier type "P" peut être obtenu de deux manières :
- par l'ordre SAVE "nom du fichier" et dans ce cas il aura une structure milieu équivalente à celle de la zone texte dont il est issu (programme BASIC).
- par l'ordre INIT # <numéro>, <nom>, [<largeur>], "type P".
Dans ce cas, sa structure est celle d'un fichier de données. On ne pourra faire un LOAD "nom fichier" puisqu'il ne correspond pas à

- à un équivalent RAM de la zone test.
- Un fichier de données sera obtenu par l'ordre INIT.
INIT # <numéro>, <nom>, [<largeur>], "type" type allant de A à Z sauf P, hormis la réserve citée plus haut...

Il est ainsi possible de créer des fichiers de noms identiques et qui se différencient par le type, mais dont l'utilisation et la structure sont les mêmes.

ORGANISATION D'UN FICHIER RAM TYPE "D"

Dans le sens croissant des adresses RAM.
<nom> <type> <longueur> <réserve> <fichier> <nom>
<type>....

	L octets
<nom>	6 caractères ASCII maximum, des espaces si moins de 6.
<type>	1 caractère ASCII, lettre de A à Z (sauf P...).
<réserve>	5 octets nuls.
<fichier>	le contenu réel du fichier utilisable par le programme et dont la longueur a été spécifiée par l'ordre INIT.
<longueur>	dimension totale du fichier (L octets) donc supérieure à la longueur spécifiée par l'ordre INIT de 14 octets.

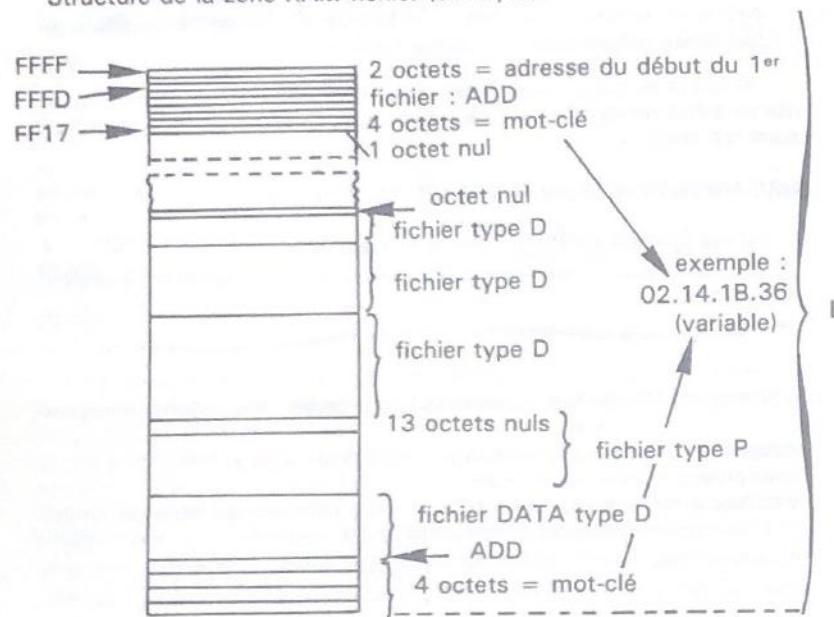
Le fichier qui suit est placé sans séparateur après le précédent (longeur occupe 2 octets).

ORGANISATION D'UN FICHIER RAM TYPE "P"

Il est issu d'un ordre SAVE.

<nom>	<type>	<longueur>	<réserve>	<fichier>
<séparateur>				
<nom>				celui spécifié dans l'ordre SAVE.
<type>				P comme programme (RUN "nom" est possible).
<longueur>				celle du même fichier en zone texte + 27 octets.
<réserve>				5 octets nuls.
<fichier>				celui de la zone texte... copie conforme !
<séparateur>				13 octets nuls.

Structure de la zone RAM fichier (exemple).



L = nombre d'octets réservés par FSET.

Exemple d'organisation d'une zone RAM pour fichier comportant 4 fichiers type D et 1 fichier type P.

Pour créer un fichier de type DATA ("D"), il faut faire l'ordre INIT # ...

L'ordre INIT nécessite dans ce cas de déterminer l'ordre de grandeur du fichier que l'on désire créer en spécifiant dans la commande sa longueur.

Exemple :

INIT #2, "GM001", 50, "K"

5 octets sont réservés dès l'exécution de cet ordre dans la mesure où la zone RAM fichier le permet, sinon l'agrandir par l'ordre FSET. S'il ne restait que 5 octets disponibles, vous devez agrandir la valeur précédente de FSET de $50 + 14 - 5$ (si vous voulez optimiser... bien sûr), soit 59.

Le fait d'exécuter une première fois l'ordre INIT, crée le fichier en RAM en réservant 50 octets (qui sont mis à zéro).
Ensuite vous pouvez faire :

PRINT #1, "MG", A
ou OUT #1, D

etc..., jusqu'à ce que le nombre d'octets transmis au fichier soit égal à 50, au-delà de quoi vous auriez droit à un message d'erreur IO ERROR (input output error). L'accès au fichier se fait donc de façon SÉQUENTIELLE.

En ce qui concerne la lecture de ce fichier, de la même façon, elle sera faite séquentiellement.

Supposons qu'on crée le fichier "TEST".

Par INIT #1, "TEST", 50, "X"
faisons PRINT #1, 25.

Si l'on exprime le contenu du fichier TEST en hexa, on pourra voir en hexadécimal : 20 32 35 20 0D 0A

2 5 CR LF

ce qui aurait été affiché sur l'écran LCD en faisant PRINT 25 puis RETURN (- CR LF).

CR carriage return
LF line feed.

Pour écrire à partir du début du fichier (en "écrasant" ce qui est déjà dedans...), il suffit de faire :

INIT #1, "TEST", "X"

Le pointeur du fichier est ainsi réinitialisé au départ.
Même chose pour lire plusieurs fois de suite le fichier ; notez que la dimension est omise car elle ne sert pas, le fichier existant déjà. Si elle est indiquée, le BASIC ne s'y intéresse pas...

INIT #1, "TEST", 1250, "X"
est accepté !

A vrai dire, la fonction PRINT s'applique plutôt pour mémoriser du texte..., autrement utiliser la fonction OUT# et INP#. Pour lire du texte, INPUT# est très indiqué (ou LINE INPUT#).

Exemple :

10 INIT #1, "TEST", 50, "D"
20 LIST #1

Ce programme crée le fichier RAM dont le nom est TEST et réserve 50 octets pour mémoriser ce que l'on désire. La ligne 20, exécutée, va envoyer dans ce fichier ce même programme de la même manière qu'il serait envoyé sur l'écran LCD en faisant l'ordre LIST.
Chacune des lignes de programme envoyée dans le fichier se termine par CR, LF, comme une ligne normale d'écriture (retour chariot + descente ligne).

Noter que si l'on remplace la ligne 10 par :

10 INIT #1, "LPT:"

on utilise ainsi l'imprimante, et vous observerez que l'on obtient le listing du programme sur l'imprimante comme si l'on avait exécuté l'ordre LLIST (à condition bien entendu d'avoir connecté l'imprimante !). Ce que vous voyez sur le papier de l'imprimante est exactement ce qui est inscrit dans le fichier TEST : une suite de codes ASCII représentative des caractères constituant les lignes du programme proposé...

De la même façon, vous initialisez le périphérique "COM:" avec un ordre comme celui-ci :

10 INIT #1, "COM:", 300, "B"

et hop, vous allez transmettre le listing de votre programme par la sortie SÉRIE de votre X-07, notamment en utilisant l'interface série RS232 qui a été proposée.

Ici la vitesse de transmission a été choisie égale à 300 bds et le mode ACIA est "B", à savoir :

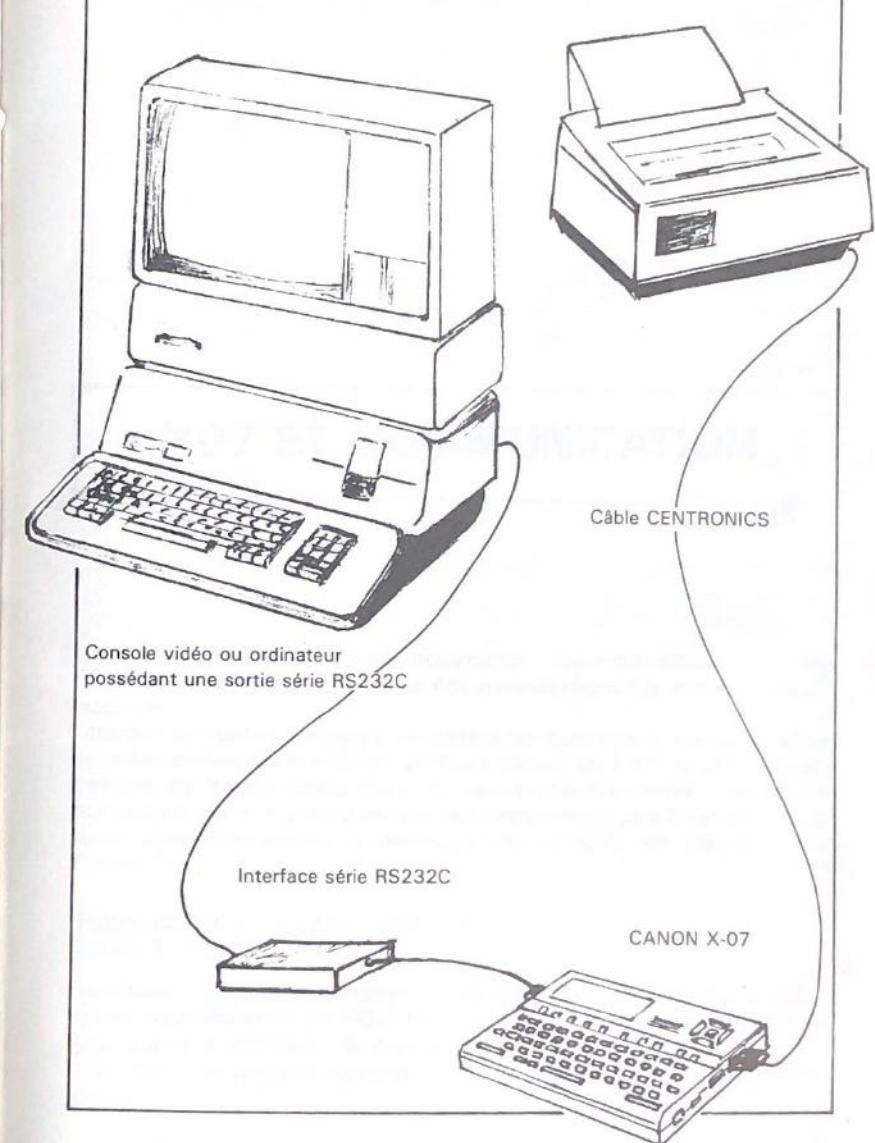
- pas de parité
- caractère de 8 bits.

Tout dispositif acceptant ce type de transmission (exemple : imprimante thermique à entrée série SILENT 700) enregistrera le listing de votre programme.

PARTICULARITÉ CONCERNANT LE FICHIER DONT LE NUMÉRO EST 5

Ce fichier va vous permettre de remplacer le clavier du CANON par tout autre dispositif extérieur accessible par la sortie série, par exemple, ou un fichier du CANON situé dans la zone RAM pour fichier. De cette façon votre CANON recevra ses ordres extérieurement et ne répondra plus aux sollicitations du clavier...

Le X-07 communique avec plus gros que lui...



X07 ET COMMUNICATION

Votre CANON X-07 peut recevoir tout ordre transmis par la prise SÉRIE dans la mesure où ce qui est envoyé répond à la syntaxe qu'il accepte.

Une console vidéo, un autre ordinateur qui possèdent une sortie série peuvent envoyer des lignes d'écriture BASIC au X-07 et celui-ci l'acceptera de bonne grâce dans la mesure où on trouve d'abord un numéro de ligne, l'ordre est exécuté directement et s'il est incorrect, vous verrez apparaître le message SN ERROR, par exemple, sur l'écran LCD.

Faites INIT #5, "COM:", 300, "B"
EXEC &HEE1F

La seconde ligne donne l'ordre d'exécuter un sous-programme en langage machine situé en ROM du X-07 qui autorise le dispositif d'entrée/sortie à remplacer le clavier.

Dès lors vous pouvez appuyer sur les touches du clavier... rien ne se passe.

REMARQUE IMPORTANTE

Le X-07 reçoit ses ordres de l'extérieur, mais ne transmet pas ses réponses à l'extérieur mais uniquement à l'écran LCD... Si un de vos amis possède un ordinateur ayant une interface série, vous aurez la possibilité de transmettre (sous forme de listings) des programmes BASIC de cet ordinateur au vôtre. Évidemment, la syntaxe ne sera sans doute pas toujours au goût de votre X-07, ainsi que certains mots-clés qui lui paraîtront absurdes et donc impossibles à exécuter. Qu'à cela ne tienne, ne faites pas RUN surtout, mais examinez chaque ligne en modifiant ce qu'il y a à modifier ! Pour pouvoir faire cela, après avoir vu défiler toutes les lignes du programme sur l'écran LCD de votre X-07 comme si vous les tapiez, à la fin de la communication, coupez votre X-07 par la touche OFF et appuyez à nouveau sur RUN ou bien faites que votre copain envoie au X-07 l'ordre direct EXEC &H EE33 ; de cette manière le clavier redévient opérant et le dispositif d'entrée/sortie ne sera plus initialisé. Notez bien que plus le BASIC de votre copain présente de similitudes avec le BASIC X-07, moins vous aurez de problème pour transcrire un programme.

Il m'a été possible sans inconvenient, grâce à cette technique, de transférer des logiciels sous forme ASCII (listing) à partir d'un système MDS INTEL (voir photo), (BASIC et traitement de texte CREDIT).

Le fichier 5 permet donc de lire des lignes BASIC autres que celles tapées sur le clavier.

La technique indiquée plus haut ne s'accorde pas de très haute vitesse de transfert (toutefois, c'est plus rapide que de taper à la main !). Cela est surtout vrai si les programmes sont très longs.

Il faut prendre soin de faire NEW afin de vider la zone texte, sinon cela oblige le X-07 à rechercher l'endroit où devra se trouver la ligne entrée et il risque de perdre ce qui suit.

FUSION DE PROGRAMMES

Le procédé est le même mais c'est un fichier RAM qui sera initialisé sous le numéro 5.

N.B. : des lignes qui possèdent le même numéro que celles actuellement en zone texte, prendront la place des originales, donc prudence...

INIT #5, "PROG", , "D"
EXEC &H EE1F

Pour reprendre la main, là pas de problème, le fichier lu, le CANON rend la main au clavier.

N.B. : le fichier est du type "D" est surtout pas "P"... issu d'un LIST # du programme (précédemment vu).

Attention

Si votre X-07 reçoit l'ordre RUN de l'extérieur et que le programme dans le X-07 utilise des fonctions telles que CASI: ou BEEP, l'initialisation première de votre X-07 sera perdue, et le clavier retrouvera sa vocation d'origine au grand désappointement de celui qui télécommandait le X-07.



INTERFACE SÉRIE POUR RÉALISER DES LIAISONS ASYNCHRONES VERS UN AUTRE ORDINATEUR

Cet interface utilise les fameux et classiques circuits dont le suffixe de terminaison est 1488 et 1489.

Ces circuits convertissent respectivement des signaux TTL en signaux respectant la norme RS232 et inversement, des signaux RS232 en signaux TTL.

La norme RS232, en ce qui concerne les niveaux, est relativement simple puisque tout ce qui est entre 3 et 25 volts sera au niveau logique A, et ce qui est compris entre -3 et -25 volts sera au niveau logique B. A et B peuvent être 1 ou 0 comme on l'entend.

La réalisation proposée utilise 2 piles 9 V, type 6F, pour fournir ces tensions ; vue la tolérance dans les niveaux, pas la peine de vérifier ces piles au millivolt près !

Quant à la tension TTL, théoriquement 5 V, elle est issue du 6 V qui provient de la broche 9 de la prise série du X-07. Deux diodes ramènent la valeur à environ 4,8 V pour un fonctionnement correct du circuit 1489.

La prise série V24 est en général réalisée avec la classique CANNON 25 points.

Notez la présence d'inverseur NAND puisque les circuits 1488 et 1489 inversent aussi les signaux.

Les lignes en grisé, DTR et DSR ne sont pas toujours utilisées et le CANON ne s'intéresse pas à DSR.

En fait, dans pas mal de cas, l'ordinateur est relié à un périphérique du type console vidéo à l'aide de 3 fils uniquement. Cela suffit en général car une console vidéo est suffisamment rapide pour prendre à la volée les caractères que lui envoie l'ordinateur. Ces signaux indispensables sont TXD, RXD et SIGNAL GROUND (terre de signalisation).

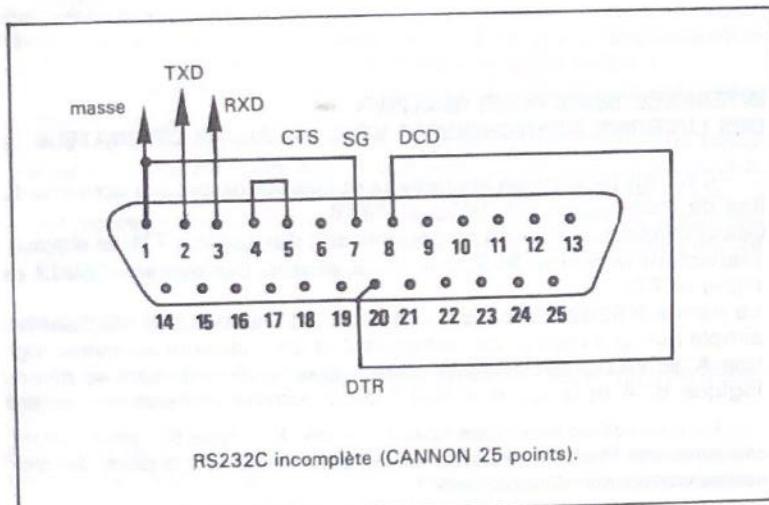
TXD Transmit DATA

RXD Receive DATA

SIGNAL GROUND, masse des signaux.

En plus, par sécurité, on peut relier les masses mécaniques des appareils (souvent confondues avec SIGNAL GROUND). Cependant, les puces qui contrôlent ces entrées/sorties série (en général du type USART : UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER) contrôlent les signaux RTS et CTS (request to send, clear to send) pour indiquer mutuellement si elles sont prêtes à émettre ou recevoir.

Dans la majorité des cas, ces signaux sont bouclés sur eux-mêmes au niveau ordinateur et au niveau périphérique ; la liaison RS232 est dite incomplète et on en a en général le schéma suivant :



DCD est la ligne de détection de porteuse et n'est utilisée que lorsqu'un modem interface la liaison et qu'il place au niveau haut s'il reçoit une fréquence reconnue par lui.

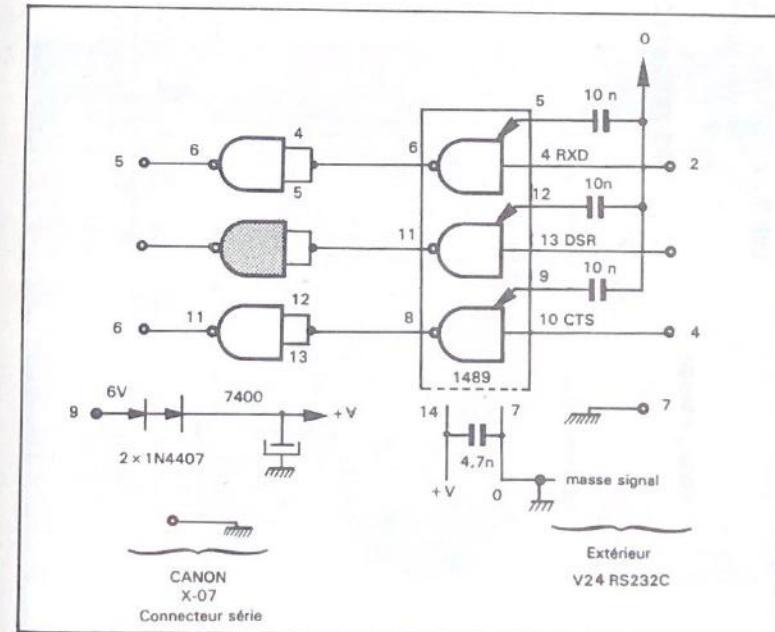
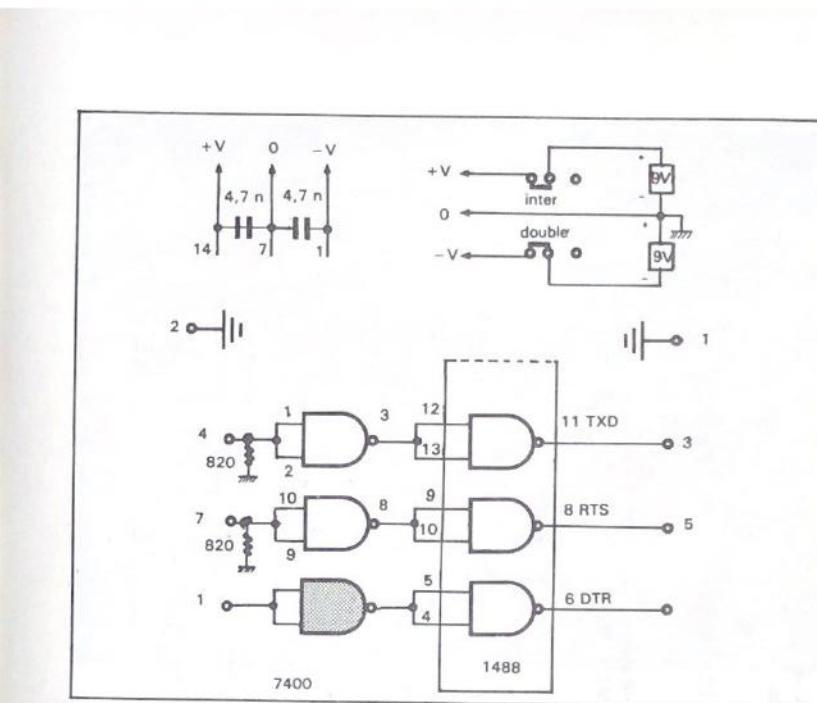
Actuellement beaucoup de transmissions sont assurées de cette manière et utilisent des modems afin de moduler les signaux binaires et de les rendre transportables par une ligne BF classique comme celle du téléphone et de transmettre l'information beaucoup plus loin que la liaison RS232 ne le pourrait.

Avec un modem qui possède l'interface RS232C (dans la majorité des cas) tel l'ANDERSON JACOBSON, pionnier des modems dits acoustiques qui évite ainsi d'aller tripoter la ligne PTT..., vous pouvez avoir accès aux banques de données et à des réseaux spécialisés comme TRANSPAC...





```
*EDITEUR DE TEXTE VIDEO<ARRET>
10 CLS: CLEAR 300T
20 PRINT "           SIMULATION    D'ATERRISSAGE SUR    UNE PLANETE"
20 C=0: E$=STRING$(19, " ")†
40 Z=2: GOSUB 1420: GOTO 140†
50 Q$="": RE=0: FOR I=1 TO CA: INIT#I,"KBD": IT†
60 QS=INKEY$: IF QS="" THEN 60†
70 IF QS=CHR$(13) THEN 130†
80 IF ASC(QS)>57 OR ASC(QS)<48 THEN GOSUB 1410: GOTO 50†
90 PRINT QS: QS=QS+QS†
100 NEXT I†
110 RE=VAL(OS$)†
120 RETURN†
130 IF OS="" THEN RE=0: RETURN ELSE 110†
140 DATA .00,.00,.00,1C,10,1C,00,00,00,00,FC,00,FC,00,78†
150 DATA .00,.00,.00,E0,20,E0,00,00,00,00,78,44,00,88,44,00,30,0C†
150 DATA 84,00,04,48,48,84,00,FC,C0,78,08,44,44,00,88,30,0C†
150 DATA 74,4C,44,2C,00,3C,18,08,00,FC,00,FC,78,FC,78,30†
150 DATA 88,0C,0B,78,00,F0,00,00,00,00,00,04,00,00,IC,00†
150 DATA 09,10,26,C0,00,00,00,00,20,20,20,20,30,30,30,78,00†
150 DATA 48,20,10,0C,00,00,00,00,00,00,00,00,00,40,40,00†
150 DATA .00,.00,.00,.00,.00,.00,.00,.00,.00,.00,.00,.00,.00,.00,.00,.00†
```



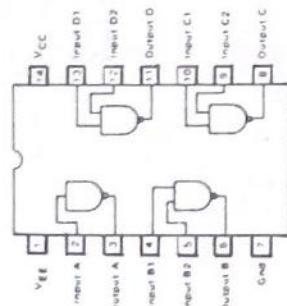
CIRCUITS D'INTERFACE RS232C Modem/terminal

Voltage Mode

RS-232C SPECIFICATION

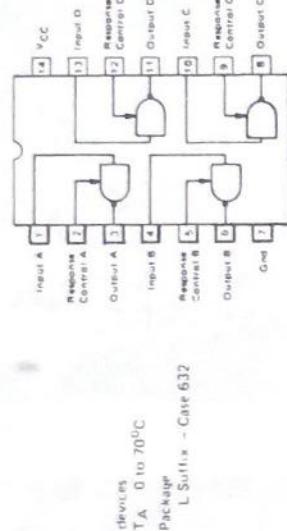
DRIVER

MC 1488 - Quad, output current limiting

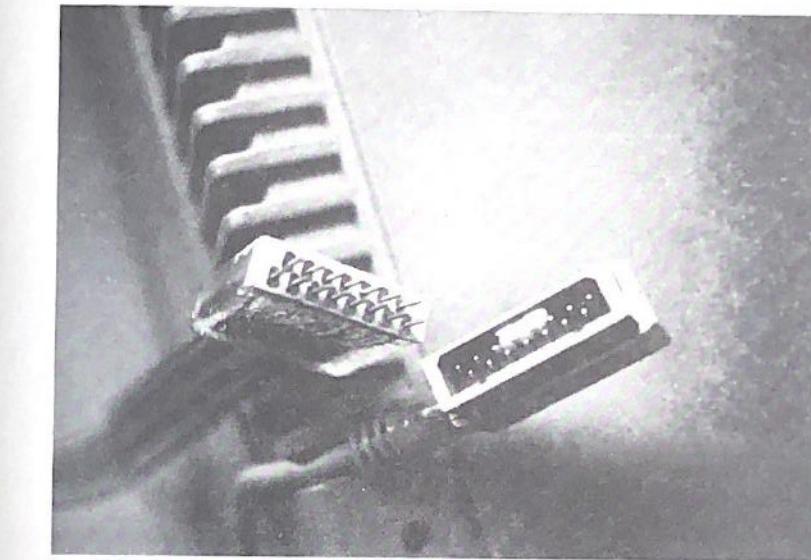
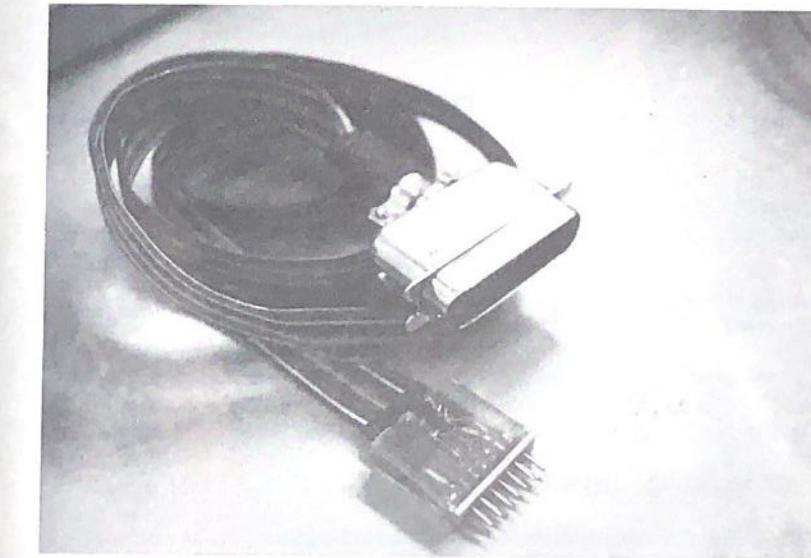


RECEIVERS

MC1489 - Quad, 0.25 V input hysteresis
MC1489A - Quad, 1.1 V input hysteresis



V_{OH} $\text{@ } V_{CC}/V_{EE} = 9.0 \text{ V}$	V_{OL} $\text{@ } V_{CC}/V_{EE} = 9.0 \text{ V}$	I_{OS} mA	t_{PHL} $\text{@ } C_L = 15 \text{ pF}$	t_{PLH} $\text{@ } R_L = 390 \Omega$
Volts Min	Volts Max	mA	ns Max	ns Max
6.0	6.0	6.0 to 12	1.75	5.0
MC1488 MTL MC1489 MTL } Available				MILITARY TEMPERATURE RANGE (-55°C to 125°C)



PROGRAMMES D'APPLICATION

LOTO

Pour vous permettre de gagner de quoi acheter les périphériques complémentaires au X-07, voici un petit programme qui propose des tirages de chiffres pour jouer au LOTO national.

Le programme débute en 100 et commence par initialiser le générateur de séquence pseudo-aléatoire en créant une nouvelle séquence issue de la valeur du temps lue avec la fonction TIME\$.

TIME\$ est de la forme "12:51:45" (12 h 51 min 45 s). De cette façon, chaque fois qu'on fait RUN, la séquence produite risque peu d'être semblable à la précédente, à moins de démarrer toujours à la même heure !

A la ligne 120 : R=RND(-R) initialise le générateur puisque l'argument est négatif.

La fonction MID\$ a extrait la représentation de la valeur des minutes (TIME\$,4,2) et des secondes (TIME\$,7,2) et VAL l'a transformée en expression numérique.

R pourra prendre la valeur 0 à 59 × 59.

A la ligne 130, réservation d'un tableau à 8 dimensions pour mémoriser les 7 chiffres du LOTO.

N(0) à N(7), soit 8 nombres (N(0) n'est pas utilisé). Cette réservation n'était pas strictement nécessaire. Un tableau qui n'excède pas 11 dimensions est accessible sans déclaration préalable (la valeur d'une variable utilisée la première fois après un RUN ou un ordre ERASE est toujours 0).

La boucle 160-210 tire 7 nombres compris entre 1 et 49 (l. 170) et fait appel à la routine de la ligne 30 pour éviter de donner 2 fois le même nombre dans un tirage, ce qui n'est pas possible.

En 220, 230 on affiche le résultat en formatant, grâce à l'utilisation du PRINT USING "# #" et du correspondant LPRINT "USING" "# #", pour l'impression sur imprimante.

A ce propos, on peut écrire :

(PRINT ou) LPRINT " ";USING" # #";A

mais pas

(PRINT ou) LPRINT USING "# #";A;" "

puisque le USING "# #" s'accomode de nombres et non de chaînes de caractères comme " " (espace) dans ce cas précis.

```
10 REM * LOTO *
20 CLS: GOTO 100
30 FOR I=1 TO J-1
40 IF N(I)=N(J) THEN 170
50 NEXT I
60 GOTO 200
100 PRINT "Tirage du LOTO"
110 R=VAL(MID$(TIME$, 4, 2))*VAL(MID$(TIME$,
    7, 2))
120 R=RND(-R)
130 DIM N(7)
140 N=N+1
150 PRINT "Tirage no";N
155 LPRINT "Tirage no";N
160 FOR J=1 TO 7
170 N(J)=INT(RND(1)*49+1)
```

```
180 IF J=1 THEN 200
190 GOTO 30
200 NEXT J
210 FOR J=1 TO 6: PRINT" "; USING"##";N(
    J);
215 LPRINT" "; USING"##";N(J);: NEXT J
220 PRINT: PRINT"complementaire : ";USIN
    G"##";N(7)
225 LPRINT" complementaire : ";USING"##
    ";N(7)
230 A$=INKEY$: IF A$=CHR$(13) THEN 140 E
    LSE 230
```

Exemple d'exécution :

```
Tirage no 1
2 13 41 16 44 48 complémentaire : 17
Tirage no 2
6 19 49 32 4 28 complémentaire : 7
Tirage no 3
4 47 25 45 23 24 complémentaire : 3
```

COURBES

L'écran LCD de votre X-07 est tel qu'il permet le graphisme. Sa définition de 120 × 32 est relativement modeste, mais on peut quand même faire des choses.

Voici un petit programme qui permet le dessin d'une courbe très connue.

PSET (x, y) allume le point d'ordonnée y et l'abscisse x sur l'écran LCD.

```

1 REM*COURBE*
2 CLS
100 H=119:U=31
120 X1=H/2:X2=X1*X1:Y1=U/2:Y2=U/4
130 FOR X=0 TO X1
140 X4=X*X:L=-Y1
150 A=SQR(X2-X4)
160 FOR I=-A TO A STEP U/2
170 R=SQR(X4+I*I)/X1
180 F=(1-R)*COS(8*R)
190 Y=I/5+F*Y2
200 IF Y<L THEN 230
210 L=Y:Y=Y1-Y
220 PSET(X1-X,Y):PSET(X1+X,Y)
230 NEXT I:NEXT X

```

REVEIL MUSICAL

Les opérateurs logiques du CANON X-07 permettent de travailler en binaire et d'effectuer toutes les opérations classiques comme le ET, OU, OU exclusif, etc...

Il ne faut pas perdre de vue que cela nécessite de travailler avec des valeurs du type ENTIER (le X-07 fait le nécessaire si vous l'oubliez).

– 1 équivaut à 32767
 $32767 \text{ AND } -1 = 32767$

de même :

$-1 \text{ AND } 1 = 1$ (1 est égal à 2^0 , soit 0001 et – 1 soit $7FFF_H$

ainsi que

$1 \text{ AND } 0 = 0$ (... 01_B AND... 00_B = 00)
 $1 \text{ AND } 2 = 0$ (... 01_B AND... 10_B = 00)
 $-1 \text{ AND } -1 = -1$ ($7FFF_H$ AND $7FFF_H$ = $7FFF$)

avec l'opérateur OU et OU exclusif

$4 \text{ OR } 1 = 5$ (... 0100_B OR... 0001_B = ... 0101)
 $-1 \text{ XOR } 1 = -2$ ($7FFF_H$ XOR... 0001_H = $7FFE_H$) = – 2
 $-1 \text{ XOR } 2 = -3$
 $-1 \text{ XOR } -1 = 0$ classique, 2 termes identiques pour un opérateur XOR (ou exclusif) donnent un résultat nul.

Opérateur EQU

$1 \text{ EQU } 2 = -4$ (... 0001_B EQU... 0010_B = $7FFD_H$)
 $1 \text{ EQU } 1 = -1$
 $0 \text{ EQU } 1 = -2$ c'est la complémentation du XOR.

Opérateur NOT

$\text{NOT } 1 = -2$ le résultat est la complémentation binaire de l'argument
 $\text{NOT } -1 = 0$ argument de l'opérateur NOT.
 $\text{NOT } 0 = -1$
 $\text{NOT } 13 = -14$

Si on utilise ces opérateurs booléens dans une ligne de programme, classiquement avec l'instruction IF ... THEN dans le genre

IF A = 1 OR B = 2 THEN...

votre X-07 va remplacer, si par exemple A vaut 1 et B vaut 2, cette ligne par la suivante :

IF – 1 OR – 1 THEN ...

La condition A = 1 est vérifiée car A vaut 1, donc elle est équivalente à – 1.

On sait que

$-1 \text{ OR } -1 = -1$

donc la ligne se résume à

IF – 1 THEN...

et ce qui suit THEN sera exécuté ; une condition qui est vraie vaut – 1, inversement une condition fausse est égale à 0.

Si dans la ligne précédente A valait 2, nous aurions

IF 0 OR – 1 THEN...

mais

$0 \text{ OR } -1 = -1$

et la condition d'exécution de ce qui suit THEN est toujours vérifiée. Par contre, dans

IF A = 2 AND B = 1 THEN GOTO 10

le GOTO 10 ne sera pas exécuté si A = 1 et B = 1 car nous aurions

IF 0 AND -1 THEN...

et

0 AND -1 = 0, la condition est fausse, soit

IF 0 THEN...

Le X-07 résume ses pensées à une condition vraie vaut -1, une condition fausse vaut 0.

Je vous propose un petit programme qui tire partie de ce "théorème" pour simplifier quelque peu les lignes de programme. Cela transforme votre X-07 en réveil musical qui vous donnera l'heure tous les 1/4 heure par une amorce de mélodie et sonnera l'heure en exécutant la mélodie complète avec quelques bips supplémentaires pour préciser. En plus, ALM\$ sera utilisé afin de vous réveiller à l'heure prévue puisque vous aimez bien votre lit, n'est-ce pas ?

*** HORLOGE MUSICALE ***

```
10 CONSOLE@0
20 CLS: FONT$(144)="48,72,72,72,132,252,
48,0": X=6: Y=1
30 FONT$(145)="252,0,0,0,252,0,0,0"
40 J$="GANLUNMARMERJEUVENSAMDIM": K$="TI
EMONTUEWEDTHUFRISATSUN"
50 PRINT STRING$(20,CHR$(145)): GOSUB 36
0
60 LOCATE 0,3: PRINT MID$(ALM$,15,5): G
OTO 220
70 LOCATE X,Y: PRINT TIME$: LINE (0,2)-(
2*S+1,2)
80 LOCATE 2,2: IF AL THEN PRINT CHR$(144
); ELSE PRINT " ";
90 S=VAL(MID$(TIME$,7,2))
100 IF S=0 THEN LOCATE 0,0: PRINT STRING$(
20,CHR$(145))
110 RETURN
120 H=VAL(MID$(TIME$,1,2))
130 HM=(H=0)
140 H=H MOD 12: IF HM THEN H=12
```

```
150 FOR I=1 TO H: GOSUB 70: BEEP 0,8: BE
EP 35,6: NEXT I
160 GOSUB 200: RETURN
170 RESTORE 420
180 FOR I=1 TO L: READ N,D
190 GOSUB 70: BEEP N,0*5: NEXT I: RETURN
200 IF NOT(HM AND F) THEN RETURN
210 LOCATE 4,3: GOSUB 360: RETURN
220 M=VAL(MID$(TIME$,4,2))
230 IF F AND M MOD 15=0 THEN L=8: GOSUB
170
240 IF M=0 AND F THEN RESTORE 180: L=18:
GOSUB 180: GOSUB 120
250 IF M MOD 15=0 THEN F=0 ELSE F=-1
260 GOSUB 70: FOR I=1 TO 6
270 IF INKEY$="" THEN NEXT I ELSE Q$=INK
EY$: GOTO 420
280 GOTO 220
290 ON ERROR GOTO 350
300 CLS: PRINT "REVEIL";" ";MID$(ALM$,1
5,5)""
310 LOCATE 0,1:PRINT "entrez l'heure sou
s la forme Hh:Mn sup"
320 INPUT Q$
330 ALM$=",,,,"+Q$
340 GOTO 20
350 BEEP 20,20: RESUME 290
360 FOR J=1 TO 7
370 IF MID$(DATE$,10,3)=MID$(K$,J*3+1,3)
THEN 390
380 NEXT J
390 LOCATE 2,3
400 PRINT MID$(J$,J*3+1,3);";";MID$(DATE
$,2,2);"/";
410 PRINT MID$(DATE$,4,2);"/";MID$(DATE$,
1,2);: RETURN
```

```

420 IF Q$=CHR$(1) THEN A_=-1: CONSOLE@1
430 IF Q$=CHR$(17) THEN A_=0: CONSOLE@0
440 IF Q$=CHR$(18) THEN GOTO 290
450 IF Q$=CHR$(15) THEN SLEEP: GOTO 20
460 INIT#1,"KBD": GOTO 220
470 DATA 25,1,25,1,24,1,20,1,25,1,25,1,2
4,1,20,1
480 DATA 22,1,22,1,24,1,24,1,25,2,25,2,2
5,1,25,1
490 DATA 24,1,20,1,25,1,25,1,24,1,20,1,2
2,1,22,1,24,1,24,1
500 DATA 25,2,25,2,25,1,17,1,17,1,20,1,2
0,1,18,1,15,1,17,1
510 DATA 17,1,17,1,20,1,20,1,18,1,15,2,2
5,1,25,1,24,1,20,1
520 DATA 25,1,25,1,24,1,20,1,22,1,22,1,2
4,1,24,1,25,2,25,2

```

Notez la particularité de la ligne 120

HM = (H=0)

HM vaudra - 1 ou 1 suivant que la valeur de H issue de TIME\$ pour représenter l'heure vaudra 0 ou sera différente de 0.

IF HM THEN H=12 est équivalent à IF HM=0 THEN...

La ligne 190 permet de rafraîchir l'affichage de la date toutes les 12 heures. F est un indicateur valant -1 ou 0, 0 toutes les 1/4 heures, après que le réveil ait sonné le quart de l'heure afin que le programme ne déclenche la mélodie pendant une minute ! (M est représentatif des minutes, et M MOD 15 vaut zéro tous les quarts d'heure pendant une minute).

En 250, 260 et 270 une boucle permet d'afficher l'heure et d'explorer le clavier (INKEY\$) suffisamment pour saisir une touche pressée quelque temps.

CNTL R pour régler un horaire du réveil (alarme)

CNTL A pour amorcer l'alarme

CNTL Q pour annuler l'alarme

CNTL O comme "off" met le X-07 à l'arrêt par la commande SLEEP

SLEEP conserve l'affichage et les variables si bien que dès qu'on appuie sur ON ou que l'alarme remet en route le X-07, le programme redémarre et l'affichage n'a pas bougé, du moins on ne voit pas le COPYRIGHT, etc.

La fonction START\$ aurait permis de faire redémarrer le programme mais on voit le message de COPYRIGHT apparaître ainsi que l'ordre contenu dans la fonction START\$ (par exemple "GOTO 10" + "CHR\$(13)").

Pour que START\$ effectue pleinement sa tâche, il faut arrêter le X-07 avec l'ordre OFF2... (autrement OFF1 sera utilisé).

Dans ce programme des caractères programmables sont utilisés dont CHR\$(144) qui symbolise une cloche et signifie que l'alarme est enclenchée (utilisation de FONT\$).

Un appui sur CNTL S arrêtera le bip de l'alarme.

J\$ et K\$ vont permettre de représenter la date de manière française : jour, mois, année et transformer MONday en LUNDI. Pour régler l'heure de l'alarme, on appuie sur CNTL et R et il faut alors entrer l'heure sous la forme 12:34 pour 12 h 34 min. Un contrôle de syntaxe rigoureux serait nécessaire pour éviter de planter le programme en 320. Au lieu de cela, on utilise le ON ERROR GOTO de la ligne 310 qui aiguille vers le BEEP d'erreur de la ligne 340 et réenclenche le processus par le RESUME 280 qui suit.

N.B. : omettre RESUME 280 ferait qu'une seule erreur serait détectée de cette manière et à la deuxième on aurait droit à un arrêt du programme avec un message tel que SN ERROR IN 320 ou FC ERROR IN 340.

ON ERROR GOTO → RESUME dans la procédure qui a été appellée par l'erreur.

Voici deux autres exemples d'utilisation des fonctions logiques. But des programmes qui suivent : déplacer un point sur l'écran LCD suivant les 4 directions possibles données par les touches curseur. Les touches curseur ont pour code ASCII décimal les valeurs :

30 haut

31 bas

28 droite

29 gauche

Méthode classique : utilisation des fonctions IF ... THEN.

```
5 CLS
```

```
10 INIT#1,"KBD:"
```

```
20 A$=INKEY$: IF A$="" THEN 20
```

```
30 A=ASC(A$)
```

```

40 IF A=29 THEN X=X-1
50 IF A=28 THEN X=X+1
60 IF A=31 THEN Y=Y+1
70 IF A=30 THEN Y=Y-1
80 IF X<0 THEN X=0
90 IF X>119 THEN X=119
100 IF Y>31 THEN Y=31
110 IF Y<0 THEN Y=0
200 PSET(X,Y): GOTO10

```

En utilisant les opérateurs logiques

** FONCTIONS LOGIQUES **

```

5 CLS
10 INIT#1,"KBD:"
20 A$=INKEY$: IF A$="" THEN 20
30 A=ASC(A$)
40 X=X+(1 AND A=28 AND X<119)
50 X=X-(1 AND A=29 AND X>0)
100 Y=Y+(1 AND A=31 AND Y<31)
150 Y=Y-(1 AND A=30 AND Y>0)
200 PSET(X,Y): GOTO10

```

Exemple :

ligne 40 $X = X + (1 \text{ AND } A = 28 \text{ AND } X < 119)$

A est le code de la touche appuyée. Si le point est au milieu de l'écran et la touche droite appuyée, cela équivaut à :

$40 X = X + (1 \text{ AND } -1 \text{ AND } -1)$

soit

$40 X = X + 1$

De la même façon la ligne 50 donnera $X = X - 0$, soit au total l'incrémentation de l'abscisse du point... ce qui était le but recherché. Vous constaterez par vous-même que cela simplifie et raccourcit la programmation tout en donnant un résultat identique, n'est-ce pas ?

PROGRAMMES MUSICAUX

Afin de changer de ton, je m'en vais vous proposer de faire un peu de musique. Il n'est pas besoin d'être musicien expérimenté pour tirer parti des possibilités musicales du X-07. De toute façon, son registre ne vous permettra pas de créer une symphonie ou de reproduire celles de Beethoven...

Le tableau ci-contre vous donne un petit rappel de ce qu'il faut savoir afin de transcrire quelques mélodies connues.

La gamme du X-07 s'étend sur 4 octaves qui pour le CANON correspondent à un nombre de 1 à 48, et le tableau joint donne la relation directe entre le numéro et la note jouée.

Entre le DO et le DO #, la note s'élève d'un demi ton ainsi qu'entre chaque numéro, de 1 à 13 vous avez donc une gamme chromatique. Les deux colonnes du milieu marquées 2^e octet et 1^{er} octet donnent une correspondance avec la notation chiffrée classiquement répandue pour les débutants.

Tout d'abord je me propose de reprendre le petit programme proposé par le manuel CANON et qui transforme le clavier en piano...

Quelques lignes supplémentaires sont ajoutées afin de palier certains défauts du programme.

Ligne 10 : avant l'instruction INKEY\$ qui affecte à I\$ le caractère frappé sur le clavier, a été ajouté CONSOLE@, 0 pour systématiquement remettre le clavier en mode normal (alphanumérique). Ainsi l'appui de la touche GRAPH ou NUM sera sans effet (toujours pour palier aux erreurs de l'utilisateur...).

RAPPEL DE LA FONCTION CONSOLE

console @, x, x, x

clavier en

- 0 mode alphanumérique
- 1 mode japonais
- 2 mode graphique
- 3 mode numérique (10 touches)

1 Alarme autorisée

0 Alarme inhibée

NOTE	2e oct	1e oct	Canon
DO	8	1	1
DO#	8+	1+	2
RE	9	2	3
RE#	9+	2+	4
MI	10	3	5
FA	11	4	6
FA#	11+	4+	7
SOL	12	5	8
SOL#	12+	5+	9
LA	13	6	10
LA#	13+	6+	11
SI	14	7	12

DO	15	8	13
DO#	15+	8+	14
RE	16	9	15
RE#	16+	9+	16
MI	17	10	17
FA	18	11	18
FA#	18+	11+	19
SOL	19	12	20
SOL#	19+	12+	21
LA	20	13	22
LA#	20+	13+	23
SI	21	14	24

DO	22	15	25
DO#	22+	15+	26
RE	23	16	27
RE#	23+	16+	28
MI	24	17	29
FA	25	18	30
FA#	25+	18+	31
SOL	26	19	32
SOL#	26+	19+	33
LA	27	20	34
LA#	27+	20+	35
SI	28	21	36

DO	29	22	37
DO#	29+	22+	38
RE	30	23	39
RE#	30+	23+	40
MI	31	24	41
FA	32	25	42
FA#	32+	25+	43
SOL	33	26	44
SOL#	33+	26+	45
LA	34	27	46
LA#	34+	27+	47
SI	35	28	48

1 Caractères définis par l'utilisateur
0 Effacés et remplacés par ceux du CANON

0 touches de fonction définies par l'utilisateur annulées et remplacées par ce que propose le CANON.

Important

Si un élément de cet ordre est omis, la fonction correspondante n'est pas effectuée.

Exemples :

CONSOLE @1 met l'alarme en service (suivant l'heure programmée par ALM\$) et seulement cela, les touches de fonction gardent les propriétés que vous aviez fixées par KEY\$ ()="..." etc., et le clavier ne change pas de mode.

CONSOLE @ 1, 0 n'affecte que les caractères programmables qui sont initialisés à la mode CANON ; le clavier est placé en mode alphanumérique.

CONSOLE @ 0 les touches de fonction retrouvent les fonctions proposées d'origine.

A la ligne 5, CONSOLE 0, 4, 0, 0, 1 empêche le déclic de touche de se produire en même temps que la note souhaitée, ce qui n'est pas harmonieux...

En 50, chaque touche appuyée et reconnue par l'instruction INSTR de la ligne 40 verra sa valeur affichée à l'écran (et si le graphisme de l'écran était plus élevé, on pourrait ainsi créer la partition musicale de ce qui serait joué à l'instant même, là on se contente d'écrire le nom de la note).

```

1 CLS: PRINT"          MUSIQUE"
4 N$="  Do Do#Re Mi bMi Fa Fa#So lSo#La S
i bSi Do "
5 CONSOLE0,4,0,0,1
10 CONSOLE @,,0: I$=INKEY$
30 IF I$="" THEN 10
40 I=INSTR("ZSXDCUGBHNJM,", I$)
50 BEEP I,4: PRINT MID$(N$,1+I*3,3);":";
60 GOTO 10

```

JEUX INTERDITS

La partition relativement simple nécessite le transcodage de chaque note en sa valeur suivant le tableau proposé. La durée des notes n'est pas relevée, une seule ayant une durée plus importante (3 temps), elle sera détectée par la ligne 1010. Le tempo général de la mélodie pourra être modifié par le deuxième facteur (durée) de l'instruction BEEP aux lignes 1010, 1020 et 1030.

Le programme s'arrête par le STOP de la ligne 1020.

*** JEUX INTERDITS ***

```

10 DATA10,10,10,10,8,6,6,5,3,3,6,10,15,1
5,15
20 DATA15,13,11,11,10,8,8,10,11,10,11,10
,14,11,10
30 DATA10,8,6,6,5,3,5,5,5,5,6,5,3,3,3
40 DATA3,3,3,7,7,7,7,5,3,3,2,2,2,1,2
50 DATA12,12,12,12,14,12,12,10,10,10,12,
14,15;15,15
60 DATA15,14,13,12,12,12,12,10,8,7,7,7,7
,8,5
70 DATA3,3,3,3,3,3,3
1000 FOR I=1 TO 96:READ N
1010 IF I=46 THEN BEEP N,23:BEEP0,1:I=46
:READN:READN:GOTO1040
1020 IF I=94 THEN BEEP N,23:BEEP0,1:STOP
1030 BEEP N,6:BEEP 0,1
1040 NEXT I

```

GIOCHI PROIBITI (JEUX INTERDITS)

57

LE BEAU DANUBE BLEU

La partition est déjà complexe puisque les notes ont une durée non constante et sont maintenues à certains endroits. Les lignes DATA rassembleront la suite des valeurs des notes accompagnées cette fois par leur durée (aux facteurs près) et un séparateur (pas très utile finalement...). La mélodie est jouée d'un seul tenant, sans aucun test.

```

10 DATA1,1,1,1,1,1,5,1,1,8,1,1,8,1,1,-1,
1,1,20,1,1
20 DATA20,1,1,-1,1,1,17,1,1,17,1,1,-1,1,
1,1,1,1
30 DATA1,1,1,5,1,1,8,1,1,8,1,1,-1,1,1,20
,1,1,20,1,1,-1,1,1
40 DATA18,1,1,18,1,1,-1,1,1,0,1,1,0,1,1,
3,1,1,10,1,1
50 DATA10,1,1,-1,1,1,22,1,1,22,1,1,-1,1,
1,18,1,1,18,1,1
55 DATA-1,1,1,0,1,1
60 DATA0,1,1,3,1,1,10,1,1,10,1,1,-1,1,1,
22,1,1
70 DATA22,1,1,-1,1,1,17,1,1,17,1,1,-1,1,
1,1,1,1,1,1,5,1,1
80 DATA8,1,1,13,1,1,-1,1,1,25,1,1,25,1,1
,-1,1,1,20,1,1
90 DATA20,1,1,-1,1,1,1,1,1,1,1,1,5,1,1,8
,1,1,13,1,1,-1,1,1
100 DATA25,1,1,25,1,1,-1,1,1,22,1,1,22,1
,1,-1,1,1,3,1,1
110 DATA3,1,1,6,1,1,10,1,1,10,4,0,7,1,1,
8,1,1,17,4,0,13,1,1
120 DATA5,1,1,5,2,1,3,1,1,10,2,1,8,1,1,1
,1,1,-1,.5,1

```

SULLE RIVE DEL DANUBIO BLU (AN DER SCHÖNEN BLAULEN DONAU)

The musical score consists of seven staves of music, each starting with a treble clef and a key signature of one sharp (F#). The staves are labeled with letters above them: C, G, C, F, D, G, and C. The music includes various note values such as eighth, sixteenth, and thirty-second notes, along with rests. The notes are primarily black, with some white notes appearing in the later staves.

```

130 DATA13,.5,1,13,1,1,13,1,1
1000 READ N,D,P
1010 BEEP(N+1),D*4:BEEP0,0+P
1020 GOTO1000

```

LE BEAU DANUBE BLEU

SERENADE DE SCHUBERT

Les choses se compliquent un peu, dans la mesure (jeu de mot) où il y a répétition, reprise d'une partie du passage musical (signe F). Le compteur I (ligne 1002) va compter le nombre de notes jouées et permettre la répétition musicale en correspondance du signe F . RESTORE 10 permet de relire les mêmes DATA (donc les premières notes de la mélodie) à nouveau.

Il n'y a pas de test pour connaître le nombre total des notes jouées et donc la fin de la sérénade, et à coup sûr on s'achemine vers une OD ERROR (out of data ERROR), mais c'est sans compter avec le ON ERROR GOTO 5000 du début qui évite cela et propose, puisque vous avez apprécié, de rejouer le morceau.

Notez qu'il n'y a pas de RESUME quelque chose après cet aiguillage du ON ERROR GOTO, il n'est pas nécessaire parce que l'on utilise RUN, qui remet toutes les variables du X-07 à zéro comme à l'origine.

```

5 CLS:ON ERROR GOTO 5000:PRINT "SERENADE
DE SCHUBERT"
10 DATA10,1.5,11,.5,10,1,15,2,10,1,8,1.5

20 DATA10,.5,8,1,15,2,8,1,10,3
30 DATA8,1,6,1,5,1,6,6,10,1.5,11,.5
40 DATA10,1,18,2,10,1
50 DATA8,1.5,10,.5,8,1,17,2,15,1,13,2,11
,1
60 DATA11,1,10,1,8,1,10,5,15,1
70 DATA13,1,0,1,11,1,11,1,10,1,8,1,10,6,
10,2,14,1,18,5

```

SÉRÉNADE de SCHUBERT

(arrangement simplifié)





```

80 DATA17,1,15,2,10,1,6,2,3,1,11,1.5,10,
.5,11,1,15,2,11,1
90 DATA10,3,8,1,7,1,8,1,11,2,8,1,6,6,10,
2,14,1,18,5
100 DATA17,1,15,2,10,1,7,2,3,1,12,1.5,11
,.5,12,1
110 DATA15,2,12,1,10,3,17,1.5,16,.5
120 DATA17,1,20,2,14,1,15,6,10,1,10,1,10
,1,10,3
130 DATA2,1,5,1,8,1,7,3,10,1,10,1,10,1,1
0,1,2,1,5,1
140 DATA8,1,10,1,14,1,15,3
1000 READ N,D
1002 I=I+1
1010 BEEP(N),D*6: BEEP0,1
1020 IF I=59 THEN RESTORE 10
1030 GOTO1000
5000 PRINT "Si vous aimez cette SERENADE
, tapez", "<RETURN>"
5010 INPUT A$:RUN

```

SERENADE DE SCHUBERT

ON THE TOP OF THE OLD SMOKY

C'est pour la soif...

BEEP 0, <durée> à BEEP 48, <durée> nous fait parcourir 4 octaves du DO au SI.

Si la valeur de la note dépasse 48, soit de 49 à 4095, le son produit ira en fréquence décroissante de la valeur 3918 Hz à 46,8 Hz.

Si la note vaut 4096, aucun son n'est produit. En vérité, la fréquence produite par le haut-parleur et aussi disponible sur la sortie enregistreur (je pense que vous avez dû le remarquer) est égale à :

$$f = \frac{192\ 000}{\text{note}} \text{ en Hertz}$$

La durée, quant à elle, est toujours réglable de 0 à 255 fois 50 ms,

soit 12,75 secondes maximum d'un seul tenant.

Si on désire un LA (440 Hz), il peut être obtenu de deux manières :

BEEP 10,<durée> ou BEEP 436,<durée>

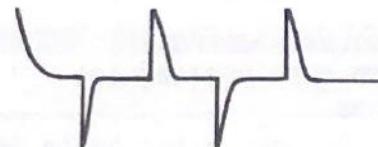
Essayez :

10 BEEP 10,40:BEEP 436,40:GOTO 10

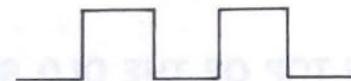
vous percevrez un petit cloc uniquement dû à la boucle ; la note, quant à elle, étant continue.

Un BEEP 0,<durée> ou BEEP 4096,<durée> ne donne aucun son dans le haut-parleur mais la sortie cassette délivre une fréquence égale à 751 Hz dans les deux cas.

Le niveau délivré à la sortie cassette est constant et d'amplitude 25 mV c/c dont l'allure des fréquences basses vers les fréquences croissantes est comme ceci :



fréquences basse

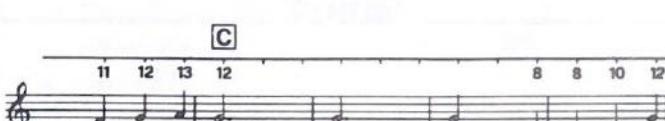
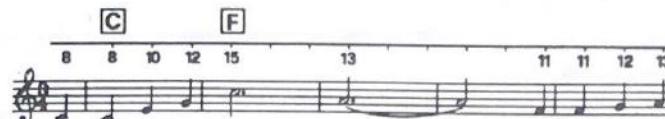


fréquence moyenne



fréquence haute

ON TOP OF OLD SMOKY



```

*** ON TOP OF OLD SMOKY ***
5 CLS: ON ERROR GOTO 5000:PRINT"ON TOP O
F OLD SMOKY"
10 DATA1,1,1,1,5,1,8,1,13,3,10,4,6,1,6,1
,8,1,10,1
20 DATA8,8,1,1,1,1,5,1,8,1,8,3
30 DATA3,5,5,1,6,1,5,1,3,1,1,7
40 DATA0,1,1,1,1,1,5,1,8,1,13,3,10,5,6,1
50 DATA6,1,8,1,10,1,8,8,1,1,1,1,5,1,8,1
60 DATA8,3,3,5,5,1,6,1,5,1,3,1,1,10
1000 READ N,D
1010 BEEP0,D+4 :BEEP0,1:GOTO1000
5000 CLS:PRINT"Tappez <RETURN>":INPUT A$:
RUN

```

MORSE

Voilà, maintenant vous êtes assez familiarisé avec la musique pour rivaliser avec Michel LEGRAND, aussi je vous invite à continuer d'assouvir votre soif de savoir en apprenant à décoder le morse à l'oreille grâce à ce petit programme qui fait appel aussi au charmant gazouillis X-07.

Vous allez pouvoir taper une lettre sur le clavier et entendre son correspondant morse, ainsi que voir sa représentation en traits et points à l'écran...

Deux tableaux de chaînes de caractères sont initialisés et vont contenir d'une part les indications long (1) et court (0), et d'autre part la représentation morse du caractère.

Exemple :

S O S ...— —...

très connu...

Quelques points particuliers du programme

SHIFT E envoie le code morse d'"ERREUR" (e = 101) qui est une succession de 8 points.

Étant donné que le clavier possède un buffer d'entrée, le fait de choisir une vitesse lente permet de taper plus vite que l'émission du message, ainsi le X-07 continue à faire du Morse tout seul...

Pour éviter cela dans un programme (si nécessaire), il suffit de faire, par exemple, INIT #1,"KBD;"

"KBD;" étant le "fichier" représentant le clavier ; cette instruction cause l'effacement de la zone tampon du clavier et vous verrez dans certains programmes son utilité.

```

10 CLS: PRINT" ENTRAINEMENT AU DECOD
AGE DU MORSE"
15 CONSOLE0,4,0,0,1
20 CLEAR100: DIM M$(40):DIM C$(40)
30 FOR I=0TO39: READ M$(I):NEXTI
35 FOR I=0TO39: READ C$(I):NEXTI
40 PRINT:INPUT"Vitesse (1 à 10)":U:U=11-
U
50 D%=38:GOSUB 1000
60 A$=INKEY$: IF A$=CHR$(13)THEN D%=39:
GOSUB1000: GOTO 40
65 IF A$="" THEN 60
70 C%=ASC(A$)
72 IF C%>46 THEN D%=36: GOTO 510
74 IF C%>101 THEN D%=37: A$="erreut": GO
TO 510
75 IF C%<48THEN60
77 IF C%<58THEN D%=C%-48:GOTO510
80 IF C%<65 OR C%>90 THEN 60
90 D%=C%-55
510 PRINTA$;: GOSUB1000
520 GOT060

```

```

1000 A$=M$(D%) :PRINTC$(D%); " ";
1020 FOR A=1 TO LEN(A$)
1030 BEEP34,(VAL(MID$(A$,A,1))+1)*V
1035 BEEP0,1
1040 NEXT A
1050 RETURN
2000 DATA11111,01111,00111
2010 DATA00011,00001,00000,10000,11000,1
1100,11110
2020 DATA01,1000,1010,100,0,0010,110,000
0,00,0111,101,0100
2030 DATA11,10,111,0110,1101,010,000,1,0
01,0001,011
2040 DATA1001,1011,1100,010101,00000000,
10101,01010
4000 DATA-----,-----,-----
4010 DATA....,....,....,-...,----,-...,-
--...,----.
4020 DATA.-,---,---,---,---,---,---,---,---
---,---,---,---,---,---,---,---,---
4030 DATA--,-,---,---,---,---,---,---,---,---,-
---,---,---,---,---,---,---,---,---,---,---
4040 DATA-.-,-.--,-..,.-.-,.-.-
4050 DATA-.-.- debut de transmission,-.-.
-.- fin de transmission

```

ENTRAINEMENT A LA LECTURE DU MORSE

FACTEURS PREMIERS

Un petit programme maintenant qui a tout son intérêt à pouvoir être traité par ordinateur : la décomposition d'un nombre en facteurs premiers.

Je vous rappelle que les nombres premiers sont des nombres qui n'ont pas de division à part eux-mêmes... et qui possèdent des propriétés et des règles de succession pas encore clairement connues. Il n'y a pas de formule pour trouver un nombre premier ; les chinois ont bien proposé

$$\frac{2n - 2}{n}$$

avec n entier il y a de cela beaucoup d'années, mais c'est faux ! Avec un ordinateur qui fait avec une facilité déconcertante des calculs répétitifs, il suffit pour savoir si un nombre quelconque est premier de le diviser par 2 (si c'est possible), de prendre le reste, de diviser encore une fois par 2 jusqu'à ce que l'opération ne soit plus possible, et de recommencer avec 3 et ainsi de suite... Si l'opération n'est pas possible, le nombre est premier, autrement on a trouvé sa décomposition en facteurs premiers.

Avec les ordinateurs puissants qui existent actuellement, on a trouvé des nombres premiers de plus de 6 000 chiffres ($2^{19937} - 1$ je crois en est un).

Votre X-07 n'est pas capable de cela mais essayez donc ce programme.

```

5 REM DECOMPOSITION EN FACTEURS PREMIERS
10 INPUT"NOMBRE" ;A
30 CLS
40 AM=A
50 PRINT A;"=";
60 FOR N=2 TO A
70 X=A/N-A\N
80 IF X=0 THEN A=A/N: PRINT "X";N;; GOTO
60
110 NEXT N
120 LOCATE 0,0: PRINT AM;"=" ;
130 LOCATE 0,3: GOTO 10

```

HARD COPY D'ECRAN

Garder une trace de ce qui se trouve affiché sur l'écran n'a pas été prévu par CANON sur son X-07. Voici donc de quoi y remédier. Un fonction HARD COPY, comme on dit.

Trois façons de lire l'écran et de le transcrire sur l'imprimante et aussi trois résultats différents sur le papier suivant la façon dont le crayon de la X 710 s'est déplacé.

Ces programmes ont pour but de lire chaque point de l'écran LCD et de déterminer s'il est allumé ou non par la fonction POINT qui vaut -1 (allumé) ou 0 (éteint). Dans le premier cas, on abaisse le crayon de l'imprimante sur le papier par l'ordre D en respectant des coordonnées homothétiques de l'écran, et dans l'autre cas, l'ordre M soulève et déplace le crayon.

Le facteur F est un coefficient multiplicateur qui permet de réduire ou d'agrandir la copie.

Trois boucles FOR ... NEXT assurent le balayage de tous les points de l'écran comme le fait le spot électronique d'un téléviseur, toutefois cela va beaucoup moins vite... et l'inertie de l'imprimante y est pour une bonne part.

Vous trouverez certainement des lignes où l'on peut simplifier l'écriture avec ce qui a été vu précédemment sur les fonctions logiques. Notez que les caractères dont on peut modifier la représentation par l'ordre FONT\$ sont représentés sur l'imprimante cette fois.

EiGMi*6677443mabcDF

HARD COPY CANON X-07
*#77443mabcDF

```

1 DEFINT A,Z
5 INPUT "facteur ";F
6 PRINT "HARD COPY CANON X-07&n"
18 LPRINT CHR$(18); "L8"
28 LPRINT "I"
38 FOR Y=0 TO 31: Y1=Y*F
48 FOR X=0 TO 119

```

```

55 IF YM>Y THEN BEEP48,1:GOSUB 300
68 IF POINT (X,Y)=1 THEN GOSUB 200
78 IF POINT (X,Y)=0 THEN GOSUB 300
88 YM=Y
98 NEXT X,Y
228 FOR I=0 TO F-1
218 LPRINT "D";STR$(X*F);",";STR$(-Y1-I)
228 NEXT I: RETURN
308 FOR I=0 TO F-1
318 LPRINT "M";STR$(X*F);",";STR$(-Y1-I)
328 NEXT I: RETURN

```

HARD COPY CANON X-07

```

1 DEFINT A,Z
5 INPUT "facteur ";F
6 PRINT "HARD COPY CANON X-07&n"
18 LPRINT CHR$(18); "L8"
28 LPRINT "I"
38 FOR Y=0 TO 31: Y1=Y*F
48 FOR X=0 TO 119
55 IF YM>Y THEN BEEP48,1:GOSUB 300
68 IF POINT (X,Y)=1 THEN GOSUB 200
78 IF POINT (X,Y)=0 THEN GOSUB 300
88 YM=Y
98 NEXT X,Y
228 LPRINT "D";STR$(X*F);",";STR$(-Y1-I)
228 RETURN
308 LPRINT "M";STR$(X*F);",";STR$(-Y1-I)
328 RETURN

```

:CANON X-07:COPIE écran

```

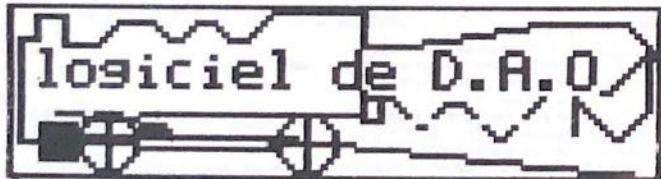
1 DEFINT A,Z
5 INPUT "facteur ";F:CLS
6 PRINT ":CANON X-07:copie écran M&G"
18 LPRINT CHR$(18); "L8"
28 LPRINT "I"
38 FOR Y=0 TO 31: Y1=Y*F
48 FOR X=0 TO 119
55 IF YM>Y OR YM=I THEN BEEP48,1:GOSUB 300
68 IF POINT (X,Y)=1 THEN GOSUB 200
78 IF POINT (X,Y)=0 THEN GOSUB 300
88 YM=Y: IM=I
98 NEXT X,I,Y
228 LPRINT "D";STR$(X*F);",";STR$(-Y1-I)
228 RETURN
308 LPRINT "M";STR$(X*F);",";STR$(-Y1-I)
328 RETURN

```

Programme de recopie d'écran Canon X-07

DAO

Vous n'êtes pas sans avoir vu déjà ces jouets munis de deux manettes et qui permettent, si l'on est adroit, de dessiner sur un écran sur lequel glisse une pointe dirigée par ces mêmes manettes. En améliorant le programme précédent, on obtient un logiciel d'aide au dessin par ordinateur... baptisé pompeusement D.A.O.



```

10 CLS
20 DEFINT A-Y: Z=1: XD=Z: YD=Z: XF=Z: YF
=Z
30 B=Z: YM=0: XM=B: X=B: Y=0: GOTO 270
40 CONSOLE0,,0: INIT#1,"KBD:"

```

```

50 A$=INKEY$: IF A$="" THEN 50
60 A=ASC(A$)
70 IF A=30 THEN Y=Y-1: GOTO 250
80 IF A=29 THEN X=X-1: GOTO 260
90 IF A=28 THEN X=X+1: GOTO 230
100 IF A=31 THEN Y=Y+1: GOTO 240
110 IF A=17 THEN GOSUB 500
120 IF A=84 THEN B=1
130 IF A=15 THEN 320
140 IF A=69 THEN B=2
150 IF A=1 THEN 350
160 IF A=68 THEN XD=X: YD=Y: GOTO 270
170 IF A=70 THEN XF=X: YF=Y: GOTO 270
180 IF A=15 THEN 320
190 IF A=12 THEN LINE (XD,YD)-(XF,YF)
200 IF A=18 THEN 210 ELSE 270
210 LINE (XD,YD)-(XF,YD): LINE (XF,YD)-(XF,YF)
220 LINE (XF,YF)-(XD,YF): LINE (XD,YF)-(XD,YD): GOTO 270
230 IF X>119 THEN X=1
240 IF Y>30 THEN Y=0
250 IF Y<0 THEN Y=30
260 IF X<1 THEN X=119
270 ON B GOTO 280,300
280 PRESET(X,Y): PRESET(0,YM): PRESET(XM
,31): PSET(0,Y)
290 PSET(X,31): PSET(X,Y): XM=X: YM=Y: G
OTO 40
300 PSET(X,Y): PRESET(0,YM): PRESET(XM,3
1): PSET(0,Y)
310 PSET(X,31): XM=X: YM=Y: PRESET(X,Y):
GOTO 40
320 Z=INT(SQR((XF-XD)^2+(YF-YD)^2))
330 CIRCLE (XD,YD),Z
340 GOTO 40

```

```

350 IF X<6 THEN BEEP 20,5: BEEP 10,2: GO
TO 40
360 BEEP 40,2: BEEP 0,1: PO=X^6: LOCATE
PO,Y^8
370 INIT #1,"KBD:"
380 PRESET(XM,31): PRESET(XN,31): XN=PO*
6: PSET(XN,31)
390 A$=INKEY$: IF A$="" THEN 390
400 A=ASC(A$): IF A=1 THEN 450
410 IF A=29 THEN GOTO 460
420 IF A<32 THEN 370
430 GOTO 480
440 PRINT A$;; GOTO 370
450 PRESET(XN,31): PSET(XM,31): BEEP 40,
1: GOTO 40
460 IF PO=1 THEN BEEP 20,5: BEEP 10,2: G
OTO 370
470 PO=PO-1: GOTO 440
480 IF PO=19 THEN BEEP 20,5: BEEP 10,2:
GOTO 370
490 PO=PO+1: GOTO 440
500 F=4: TM=0: IM=0
510 LPRINT CHR$(18); "L0"
520 LPRINT "I"
530 FOR T=0 TO 31: Y1=T*X
540 FOR I=0 TO F-1
550 FOR U=0 TO 119
560 IF TM>T OR IM>I THEN GOSUB 720
570 IF POINT (U,T) THEN GOSUB 620 ELSE G
OSUB 680
580 TM=T: IM=I
600 NEXT U,I,T
610 LPRINT "A": RETURN
620 IF U=119 THEN .660
630 FOR J=U+1 TO 119
640 IF POINT (J,T) THEN NEXT J

```

```

650 U=J-1
660 LPRINT "D"; STR$(U*X+5); ","; STR$(-Y1-
1)
670 RETURN
680 IF U=119 THEN 720
690 FOR J=U+1 TO 119
700 IF NOT(PQINT (J,T)) THEN NEXT J
710 U=J-1
720 LPRINT "M"; STR$(U*X+5); ","; STR$(-Y1-
1)
730 RETURN

```

Grâce aux touches de curseur et quelques codes de contrôle, vous allez pouvoir dessiner facilement sur l'écran de votre X-07. En effet, après avoir fait RUN, l'écran s'efface et un point apparaît en haut à gauche de l'écran. Deux autres points sont des repères assurant la position horizontale ou verticale du premier que vous déplacez par les touches curseur et qui laissera une trace de son passage tant que E ne sera pas appuyé; à partir ce ce moment là, le point qui se déplace efface tout sur son passage jusqu'à un nouvel appui sur la touche T pour tracer à nouveau. En tapant D, vous placez un repère à l'endroit où se trouve le point, même chose en tapant F, deuxième repère.

CNTL O dessine un cercle de rayon DF et de centre D
CNTL L dessine une droite commençant en D et se terminant en F
CNTL R dessine un rectangle dont une diagonale serait la droite DF

En tapant CNTL A vous passez en mode écriture et vous avez ainsi accès aux lettres du clavier et aux caractères graphiques sauf dans la première et dernière colonne...

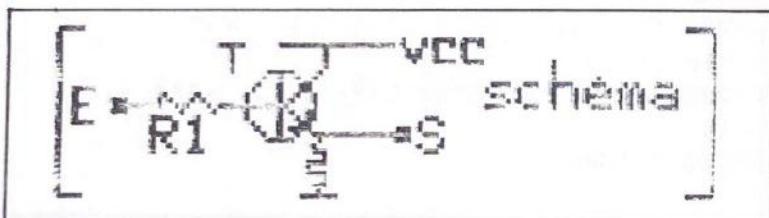
La touche curseur gauche permet de revenir en arrière dans le texte écrit pour être en mesure de le modifier facilement. Pour revenir au mode précédent, il suffit de taper à nouveau CNTL A. CNTL Q vous permet de sauvegarder votre chef d'œuvre sur l'imprimante du X-07 avec une adaptation du programme précédent. Notez l'utilisation de la fonction ¥ représentée sur l'imprimante par le signe \.

A ¥ B et INT(A/B), c'est la même chose sauf que c'est plus rapide dans le premier cas (environ 3 fois).

Afin de repérer l'endroit où l'écriture se fait dans le mode écriture du programme (accès par CNTL A), la position du caractère est déterminée par PO = X¥6 en ligne 360).

Le X-07 possède une fonction POS(0) (0 est un argument factice) qui pourrait remplir cette tâche.

Pour déplacer le point sur l'écran, on utilise les touches du curseur qui, scrutées par la fonction INKEY\$, nous déterminent le sens de déplacement du point dans les quatre directions. Vous pouvez essayer d'adapter le programme en utilisant la fonction STICK(0) qui permettra le déplacement du point en diagonale, soit huit directions au total — réfléchissez avant de regarder la solution...



```
40 CONSOLE0,,0: INIT#1,"KBD:"
42 A=STICK(0): ON A GOTO 70,75,90,95,100
   85,80,65
50 A$=INKEY$: IF A$="" THEN 50
60 A=ASC(A$): GOTO 110
65 Y=Y-1: X=X-1: GOTO 230
70 Y=Y-1: GOTO 230
75 Y=Y-1: X=X+1: GOTO 230
80 X=X-1: GOTO 230
85 Y=Y+1: X=X-1: GOTO 230
90 X=X+1: GOTO 230
95 Y=Y+1: X=X+1: GOTO 230
100 Y=Y+1: GOTO 230
110 IF A=12 THEN GOSUB 500
120 IF A=84 THEN B=1
```

La solution pour se déplacer dans les 8 directions...

Continuons dans le dessin. Voici quelques programmes pas très compliqués pour dessiner des figures géométriques simples sur l'imprimante X-710.

LPRINT CHR\$(18) place l'imprimante en mode graphique. Toutes les instructions envoyées à l'imprimante doivent être données sous forme de chaînes de caractères. En conséquence les coordonnées que devra suivre le stylo sont converties en chaînes de caractères par la fonction STR\$(nombre).

A ce propos, si vous faites (avec N = 15, par exemple) :

```
PRINT "*";(N);"
```

on obtient sur l'écran : * 15 *, un espace de chaque côté du nombre, et si vous faites

```
PRINT "*";STR$(N);"
```

cela donne * 15*, un espace au début du nombre.

STR\$ a fait disparaître l'espace derrière le nombre.

Ce sont des petites choses dont il faut tenir compte quand on présente un résultat sur l'écran LCD.

Parenthèse terminée, dans ces programmes on entre la valeur de M, à savoir 3, 14... parce que ce cher X-07 n'a même pas ça en mémoire... déplorable ignorance.

Si vous aimez les choses précises, vous entrerez M sous la forme :

3.141592653589 #

Ajouter des décimales serait inutile. # indique une variable double précision.

P# = 3.141592653589 #

DEFSNG P	P variable simple précision
P=P#	Affectation de P à la valeur de P# (P et P# sont deux valeurs différentes)

PRINT "P=";P

P=3.14159

DEFDBL D	D variable double précision
----------	-----------------------------

D=P

PRINT "D=";D	Affectation de D à la valeur de P
--------------	-----------------------------------

D=3.14159

Les décimales sont perdues irrémédiablement...

Pour information, sachez qu'on connaît actuellement la valeur de π avec plus de 1 000 000 de décimales* (obtenues sur un ordinateur CONTROL DATA 7600).

Le X-07 travaille toujours en double précision si on ne lui indique pas le contraire, tenez en compte si vous voulez améliorer la rapidité d'exécution d'une boucle...

* J. GUILLOUD et M. BOUYER, « 1 000 000 de décimales de π » Commissariat à l'énergie atomique, Paris, 1974.

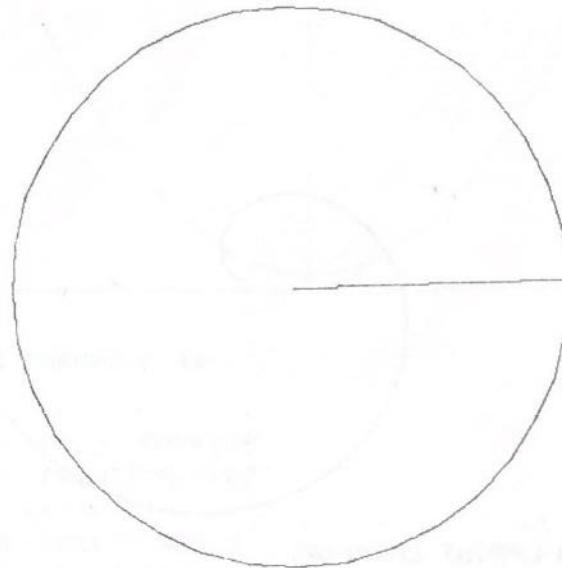
Revenons au graphisme...

LPRINT "L0,D" implique que l'imprimante dessine en trait plein suivant les coordonnées qui suivent D. Dans chaque programme, vous avez une étape de positionnement du stylo (LPRINT "M240,0") : le stylo se place au milieu de la feuille, et ensuite LPRINT "I" déclare que cette position sera considérée comme étant logique (0,0) pour ce qui va suivre. Tout déplacement se fera par rapport à cette origine. LPRINT "M", suivi de coordonnées déplace le stylo aux coordonnées sans laisser de traces.

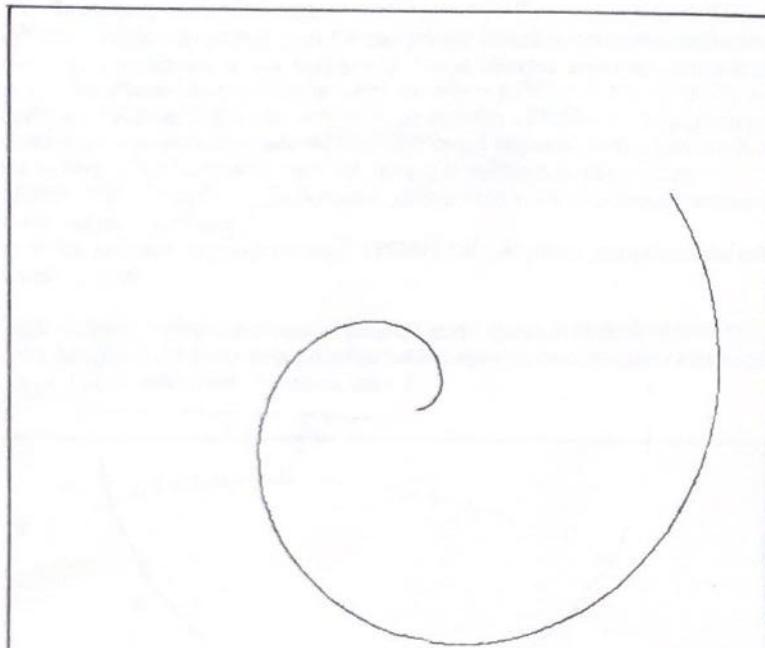
A la fin de chaque programme : LPRINT "A" replace l'imprimante en mode "texte".

N.B. : notez l'influence d'une virgule de suite après le D du LPRINT "D", cela équivaut à forcer une coordonnée à zéro et cela modifie l'aspect de la figure dessinée, n'est-ce pas ?

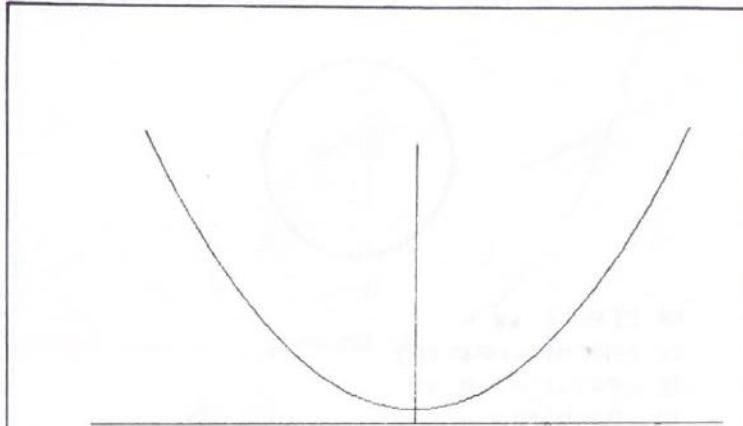
```
10 LPRINT CHR$(18)
16 LPRINT "M0,0"
20 LPRINT "M240,0"
40 PI=3.141592654
50 P=PI/180
55 R=522
60 FOR I=60 TO 110 STEP 1
69 REM ORG cercle : 170,-522
70 X=170+R*COS(I*P)
80 Y=-522+R*SIN(I*P)
82 IF I=60 THEN 85 ELSE 90
85 LPRINT "M";STR$(INT(X));",";STR$(INT(Y))
90 LPRINT "D";STR$(INT(X));",";STR$(INT(Y))
110 NEXT I
120 LPRINT "A"
```



```
10 LPRINT CHR$(18)
20 LPRINT "M240,0"
30 LPRINT "I"
40 P=3.141592654/180
50 FOR I=1 TO 360 STEP 10
60 X=200*COS(P*I)
70 Y=200*SIN(P*I)
80 LPRINT "L0,D";STR$(X);",";STR$(Y)
90 NEXT I
100 LPRINT "A"
```

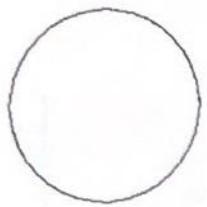


```
10 LPRINT CHR$(18)
20 LPRINT "M240,0"
30 LPRINT "I"
40 P=3 141592654/180
50 FOR I=0 TO 400 STEP 5
60 X=R*COS(P*I)
70 Y=R*SIN(P*I)
80 LPRINT "L0,D";STR$(X);",";STR$(Y)
85 R=R+3
90 NEXT I
100 LPRINT "A"
```



** PARABOLE **

```
10 LPRINT CHR$(18)
20 LPRINT "M0,-200"
30 LPRINT "I"
40 LPRINT "D480,0"
50 LPRINT "M240,200"
60 LPRINT "D240,0"
80 LPRINT "I"
90 FOR X=-200 TO 200 STEP 10
100 Y=(X^2)/200+10
110 IF X=-200 THEN 160
120 LPRINT "D";STR$(X);",";STR$(Y)
130 NEXT X
140 LPRINT "A"
150 END
160 LPRINT "M";STR$(X);",";STR$(Y)
170 GOTO 130
```

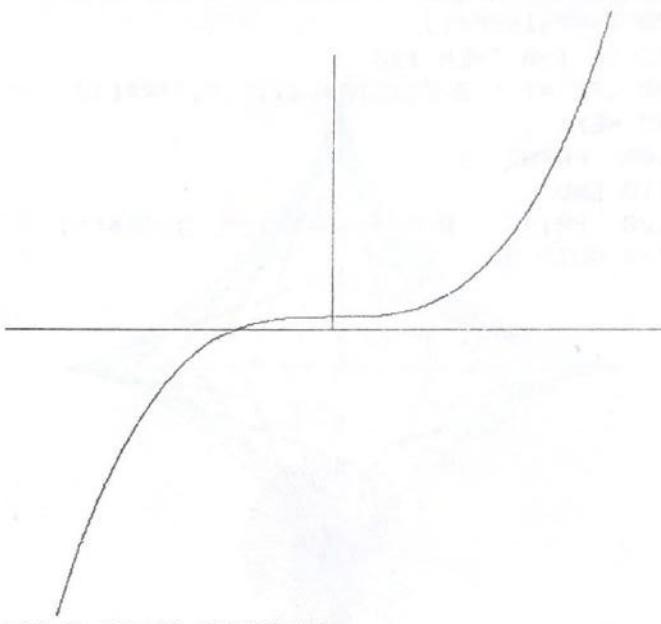


```
** CERCLE **
10 LPRINT CHR$(18)
20 LPRINT "M240,0"
30 LPRINT "I"
35 R=70
40 P=3.141592654/180
50 FOR I=0 TO 360 STEP 5
60 X=R*COS(P*I)
70 Y=R*SIN(P*I)
75 IF I=0 THEN 120
80 LPRINT "L0,D";STR$(X);",";STR$(Y)
90 NEXT I
100 LPRINT "A"
110 END
120 LPRINT "L0,M";STR$(X);",";STR$(Y)
130 GOTO 90
```

** LISSAJOU **



```
60 X=R*SIN(P*I)*COS(P*I)
```



```
10 LPRINT CHR$(18)
20 LPRINT "M240,0"
30 LPRINT "I"
35 R=70
40 P=3.141592654/180
50 FOR I=0 TO 360 STEP 5
55 LPRINT "L0,M0,0"
60 X=R*SIN(P*I)*COS(P*I)
```

```

70 Y=R*SIN(P*I)
75 IF I=0 THEN 120
80 LPRINT "L0,D";STR$(X);",";STR$(Y)
90 NEXT I
100 LPRINT "A"
110 END
120 LPRINT "L0,M";STR$(X);",";STR$(Y)
130 GOTO 90

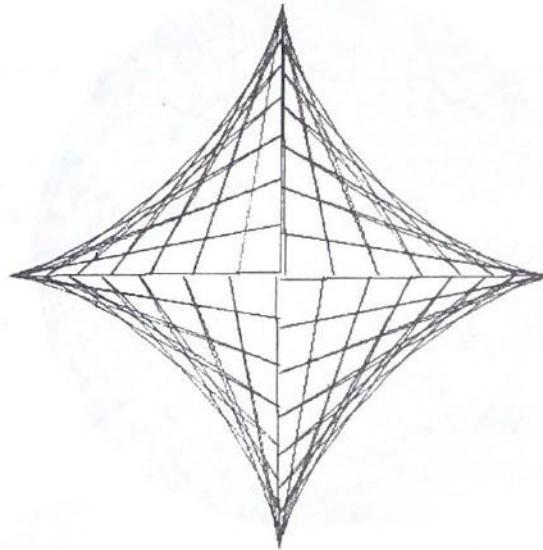
```



```

10 LPRINT CHR$(18)
20 LPRINT "M240,0"
30 LPRINT "I"
35 R=70
40 P=3.141592654/180
50 FOR I=0 TO 360 STEP 5
55 LPRINT "L0,M0,0"
60 X=R*COS(P*I)
70 Y=R*SIN(P*I)
75 'IF I=0 THEN 120
80 LPRINT "L0,D";STR$(X);",";STR$(Y)
90 NEXT I
100 LPRINT "A"
110 END
120 LPRINT "L0,M";STR$(X);",";STR$(Y)
130 GOTO 90

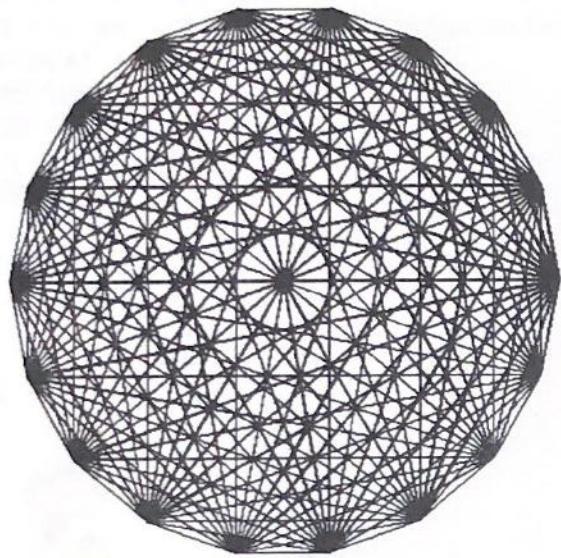
```



```

10 LPRINT CHR$(18)
20 LPRINT "M240,0"
30 LPRINT "I"
40 P=3.141592654/180
50 FOR I=1 TO 360 STEP 10
60 X=200*COS(P*I)
70 Y=200*SIN(P*I)
80 LPRINT "L0,D";STR$(X);",";STR$(Y)
90 NEXT I
100 LPRINT "A"

```



** ROSACE **

```
10 P=3.1415926535898#/180: R=200
20 LPRINT CHR$(18): LPRINT "M240, ;"STR$(R): LPRINT"1"
30 LPRINT" M";STR$(R);",0"
40 FOR I=0 TO 360 STEP 20
50 FOR J=0 TO 360 STEP 20
60 X=R*COS(P*I): Y=R*SIN(P*I)
70 Z=R*COS(P*J): T=R*SIN(P*J)
80 LPRINT" D";STR$(X);", ";STR$(Y);", ";STR$(Z);", ";STR$(T)
90 NEXT J, I
```

Puisque nous sommes dans la partie "artistique", je vous propose ce petit intermède où le X-07 montre ses dons innés pour les arts.

KALEIDOSCOPE

Une fois que l'écran LCD est suffisamment rempli par le programme, celui-ci s'efface pour laisser place à une nouvelle œuvre. Les combinaisons aléatoires sont à chaque fois renouvelées par l'utilisation de la variable TIME\$ pour générer une nouvelle séquence.

*** KALEIDOSCOPE ***

```
10 DEFINT A,Z: CLS: C=0: F=20
20 X=INT(RND(1)*60)
30 Y=INT(RND(1)*16)
40 C=C+4: IF C>2000 THEN X=RND(-VAL(RIGH
T$(TIME$,2))): GOTO 10
50 PSET(X,Y)
60 PSET(119-X,Y)
70 PSET(119-X,31-Y)
80 PSET(X,31-Y): BEEP X+C/F,Y/3
90 GOTO 20
```

SIMULATION D'ATTERRISSAGE

Revenons à des choses un peu plus terre à terre et n'hésitons pas à atterrir avec ce programme de simulation.
Vous vous êtes sans doute émerveillé en 1969 de voir atterrir ARMS-TRONG sur la Lune à bord de son module lunaire LEM. Éprouvez les mêmes sensations grisantes à bord de l'astronef X-07.

Le programme est suffisamment commenté à son démarrage pour éviter d'en dire plus long. Vous noterez cependant les particularités de certaines routines, par exemple :

- à partir de la ligne 40, pour entrer deux caractères maxi qui définiront le débit de l'injecteur (afin d'éviter de bousculer l'affichage en cas de fausse manœuvre du joueur) ;
- la routine d'initialisation des caractères par FONT\$, et comme il y en a plusieurs, une boucle FOR NEXT est utilisée pour lire les DATA.

Les variables chaînes du type FONT\$ n'admettent pas des instructions du genre

FONT\$(I) = " " (chaîne vide)

ou la concaténation

FONT\$(I)=FONT\$(I)+A\$

(si l'on fait DATE\$ = " " (chaîne vide), la date n'est pas modifiée...). Les caractères définis serviront à matérialiser l'astronef et faire des signes inversés blanc sur fond noir (vaisseau monte), v (vaisseau descend, et = (vaisseau stable).

C carburant arrive à sa fin ! attention... vous allez un peu vite pour descendre...

Avec la petitesse de l'écran LCD, il devient vite difficile de loger toutes les informations que l'on souhaite voir apparaître.

Prenez soin de reporter tous les textes et listings en respectant les espaces et les lettres collées l'une contre l'autre afin d'assurer le formatage correct des informations sur l'écran.

Dans ce programme l'altitude, la vitesse et l'accélération du vaisseau sont calculées chaque seconde ce qui amène à quelques simplifications pour les calculs. Les frottements de l'atmosphère sont supposés nuls (c'est vrai sur la Lune et CYRNO\$...). K est un coefficient qui tient compte de la transformation poussée, réacteur et débit injecteur.

Je rappelle que dans un mouvement rectiligne, uniformement varié, le vecteur accélération a est constant.

$$\vec{a} = a \vec{i} \quad \vec{i} \text{ vecteur unitaire}$$

Le vecteur vitesse est $\vec{v} = v \vec{i}$ avec $v = a.t + v_0$.

t = le temps, v_0 vitesse initiale.

Le vecteur position vaut $\vec{OM} = x \vec{i}$ avec $x = 1/2 at^2 + vot + x_0$. x_0 position initiale.

```

10 CLS: CLEAR 300: Z=2: GOSUB 1450
20 PRINT "           SIMULATION D'ATTERRI
SSAGE SUR UNE PLANETE"
30 C=0: E$=STRING$(19, " "): GOTO 130
40 O$="": RE=0: FOR I=1 TO CA: INIT#1,"K
BD":
50 Q$=INKEY$: IF Q$="" THEN 50
60 IF Q$=CHR$(13) THEN 120
70 IF ASC(Q$)>57 OR ASC(Q$)<48 THEN GOSU
B 1440: GOTO 920
80 PRINT Q$;: O$=O$+Q$
90 NEXT I
100 RE=VAL(O$)
110 RETURN
120 IF O$="" THEN RE=0: RETURN ELSE 100
130 DATA 00,00,00,1C,10,1C,00,00,00,00,0
0,FC,00,FC,30,78
140 DATA 00,00,00,E0,20,E0,00,00,0C,78,4
4,88,88,44,30,0C
150 DATA 84,00,84,48,48,84,00,FC,C0,78,8
8,44,44,88,30,00
160 DATA 74,4C,44,7C,00,3C,18,08,00,FC,0
0,FC,78,FC,78,30
170 DATA B8,C8,88,F8,00,F0,60,40,00,00,0
0,04,08,08,1C,00
180 DATA 08,10,20,C0,00,00,00,00,30,30,3
0,30,30,30,78,00
190 DATA 40,20,10,0C,00,00,00,00,00,00,0
0,80,40,40,E0,00
200 DATA F8,F8,F8,70,A8,D8,F8,F8,F8,D
8,A8,70,F8,F8,F8
210 DATA F8,F8,00,F8,F8,00,F8,F8,FC,84,B
4,BC,BC,B4,84,FC
220 DATA FC,CC,CC,CC,CC,FC,CC,FC
230 R=134: FOR I=0 TO 18
240 I$="": REM init chaine

```

```

250 FOR J=1 TO 8: READ O$: BEEP VAL("&H"
+O$),1
260 I$=I$+"&H"+O$+","
270 NEXT J
280 FONT$(R+I)=MID$(I$,1,LEN(I$)-1)
290 NEXT I: CLS
300 LOCATE 0,0: PRINT " ";CHR$(134);CHR$(135);CHR$(136)
310 LOCATE 0,1: PRINT " ";CHR$(137);CHR$(138);CHR$(139)
320 LOCATE 0,2: PRINT " ";CHR$(140);CHR$(141);CHR$(142)
330 LOCATE 0,3: PRINT CHR$(143);CHR$(144);CHR$(145);CHR$(146);CHR$(147);
340 LOCATE 5,0: PRINT "Bienvenu à bord"
350 LOCATE 5,1: PRINT "de l'ASTRONEF"
360 LOCATE 10,3: PRINT "X-07";
370 Z=6: GOSUB 1450: GOSUB 1430: CLS
380 PRINT "Indiquez le nom de la planète
ou vous allez atterrir..."
390 GOSUB 1430
400 CLS: PRINT "TERRE, LUNE, MARS ou l'as
teroïde CYRNOIS"
410 INPUT "svp ";P$: IF P$="TERRE" THEN
1130
420 IF P$="LUNE" THEN 1150
430 IF P$="MARS" THEN 1170
440 IF P$="CYRNOIS" THEN 1190
450 CLS: PRINT "Faites correctement vot
re selection RECOMMENCEZ S.U.P."
460 GOSUB 1440: BEEP 0,1: GOTO 400
470 PRINT : PRINT "Vous allez atterrir s
ur la planète "
480 PRINT : PRINT " ";P$;: GOTO 500
490 C=0 : PRINT "Vous allez atterrir sur
l'asteroïde ": GOTO 480

```

```

500 GOSUB 1430
510 CLS: PRINT " La quantité de carb
urant disponible est de"; R ;"litres"
520 GOSUB 1430
530 CLS: PRINT "ATTENTION... le débit max
i des injecteurs de vos retrofusées"
540 PRINT "est de 80 litres/s";
550 GOSUB 1430
560 CLS: PRINT "Voulez-vous d'autres info
rmations ?"
570 INPUT "svp ";Q$
580 IF Q$="" OR LEFT$(Q$,1) <> "0" THEN 64
0
590 RESTORE 1460: FOR I=1 TO 34
600 READ Q$: LOCATE 0,3
610 IF I <> 27 THEN PRINT Q$; ELSE PRINT "
";H;" m";
620 BEEP 48,1: BEEP 0,5: BEEP 0,1
630 PRINT: NEXT I
640 Z=4: GOSUB 1450: CLS
650 PRINT "Je vous souhaite une bonne des
cente et... prenez soin du X-07"
660 GOSUB 1430
670 CLS: INPUT " Le commandant de bor
d est-il prêt ? svp ";Q$
680 IF Q$="" OR LEFT$(Q$,1) <> "0" THEN BE
EP 40,10: BEEP 20,5: GOTO 670
690 CLS
700 PRINT CHR$(7);" ATERRISSAGE
DU VAISSEAU"
710 GOSUB 1430: CLS
720 LOCATE 0,0
730 PRINT "sec : ";USING"###";T;
740 PRINT "Carb : ";USING"##";P;
750 IF R<25 THEN LOCATE 1,3: PRINT CHR$(151);

```

```

260 LOCATE 0,1
270 PRINT "Alt : "; USING "####"; H;
280 PRINT " inj : "; USING "##"; D;
290 LOCATE 0,2
300 PRINT "km/h : "; USING "##.##"; U3;
310 PRINT " m/s : ";
320 IF ABS(U2)>99 THEN PRINT USING "##.##"; U2;
ELSE PRINT USING "##.##"; U2;
330 LOCATE 2,3: PRINT " ";
340 IF U<-90 THEN LOCATE 2,3: PRINT CHR$ (152);: BEEP 48,3
350 IF U<-.7 THEN 890
360 IF ABS(U)<=.7 THEN 900 ELSE 910
370 LOCATE 0,3: PRINT E$;CHR$(13);"80 iv
s maxi";: GOSUB 1440
380 BEEP 0,1: PRINT CHR$(13);E$;: GOTO 9
20
390 LOCATE 0,3: PRINT CHR$(148);: GOTO 9
20
400 LOCATE 0,3: PRINT CHR$(150);: GOTO 9
20
410 LOCATE 0,3: PRINT CHR$(149);
420 LOCATE 4,3: PRINT STRING$(14, " ");: LOCATE 4,3
430 PRINT "Debit ? ";: CA=2: GOSUB 40: D
=RE
440 IF D>80 THEN 870
450 LOCATE 4,3: PRINT STRING$(14, " ");
460 T=T+1 : REM CALCUL TEMPS PASSE
470 IF P=0 THEN D=0 : REM TEST CARBURANT
480 IF D>R THEN D=R : REM DEBIT TROP FOR
T
490 S=((K*D)-G): REM S ACCELERATION
500 U=U+S : REM VITESSE m/s
510 U1=(U*3.6) : REM VITESSE km/h

```

```

1020 U2=ABS(U) : U3=ABS(U1) : REM VITESSE
ES
1030 H=H+(U+(S/2)): REM ALTITUDE
1040 R=(R-D): REM CARBURANT
1050 IF R<0 THEN R=0
1060 REM CONTROLE ALTITUDE
1070 IF H<0 THEN 1210
1080 IF ABS(H)<.5 AND U>-4 AND U<0 THEN
1280
1090 UM=U: HM=H: GOTO 720: REM DESCENTE
1100 RETURN
1110 T=0 :D=0 :U=0 :K=.5 :CLS
1120 U1=0: U2=0: U3=0: IF C=1 THEN 490 E
LSE 470
1130 REM TERRE
1140 G=9.81: R=650: H=2000: GOTO 1110
1150 REM LUNE
1160 H=1700: G=1.67: R=250: GOTO 1110
1170 REM MARS
1180 G=3.74: R=300: H=2000: GOTO 1110
1190 REM CYRNU
1200 H=1300: G=2: C=1: R=100: GOTO 1110
1210 VA=UM+(S*HM/(HM-H)): CLS
1220 IF ABS(VA)<=1 THEN 1330
1230 IF ABS(VA)<=3 THEN 1280
1240 IF ABS(VA)<=6 THEN 1260
1250 IF ABS(VA)>6 THEN 1300
1260 PRINT "L'astronef X-07 est detruit.
.. mais vous etes indemne";
1270 Z=1: GOTO 1380
1280 PRINT "BRAVO ! Le vaisseau s'est posé... avec quelques secousses.";
1290 Z=3: GOTO 1380
1300 PRINT "J'avertis la familleLe vaisseau n'existe";
1310 PRINT "plus... pauvre X-07! Je vous a

```

```

    mais bien";
1320 Z=1: GOTO 1380
1330 PRINT "BRAVO ! Atterrissage effectué
dans d'ex-"
1340 PRINT "Excellent conditions";: Z=2:
GOSUB 1450: BEEP 0,15: BEEP 0,1
1350 PRINT "La NASA s'intéresse à votre
candidature"
1360 PRINT "pour le prochain vol de la na-
vette."
1370 Z=8
1380 GOSUB 1450: GOSUB 1430: CLS
1390 INIT#1,"KBD": INPUT "On recommande
";Q$
1400 IF Q$="" OR LEFT$(Q$,1) <> "O" THEN 1
420
1410 RUN
1420 CLS : PRINT "AU REVEZ !": END
1430 BEEP 40,5: BEEP 0,35: BEEP 9,1: RET
URN
1440 BEEP 10,2: BEEP 20,4: BEEP 30,10: R
ETURN
1450 FOR I=1 TO 9: BEEP I*2,2: NEXT I: R
ETURN
1460 DATA ,," C'est à vous de", "chois
ir le débit de"
1470 DATA "l'injecteur afin d'", "obtenir
une poussée"
1480 DATA "verticale compensant", "la for
ce de gravité", "planétaire...",
1490 DATA " Votre intervention", " se f
era chaque", " seconde"
1500 DATA " et vous serez tenu", "au cour
ant du carbu"

```

```

1510 DATA "rant disponible et ,," de la
vitesse"
1520 DATA "du X-02 ainsi que"
1530 DATA " de son altitude", "par rappor
t au sol.", ,
1540 DATA "Je vous laisse les", " com
mandes"
1550 DATA " à l'altitude :"
1560 DATA ,," A VOUS D'ATTERIR", " EN
DOUCEUR", ,

```

MASTERMIND

Jouez avec votre X-07 à un jeu de réflexion où vous ne serez pas le seul à réfléchir car le petit CANON va essayer de résoudre la même chose de son côté.

Le but du jeu est de chercher une combinaison de trois chiffres parmi 6 (1, 2, 3, 4, 5 et 6) donc un nombre comme 421 en un minimum de coups. Vous allez chercher la combinaison que le X-07 a caché en mémoire et le X-07 cherchera la combinaison cachée dans vos cellules grises...

Supposons que le X-07 ait en mémoire 542. Vous proposez 152, alors le X-07 vous répondra 1 correct (qui est le 2) et 1 mal placé (le 5). Vous répondrez de la même façon à ses questions "bien placés ?" et "mal placés ?" pour qu'il tente de trouver votre nombre.

Attention, si vous trichez, le X-07 annule la partie...

Ce jeu se déroule d'ordinaire avec des pions colorés, mais tel qu'il est d'origine, le X-07 ne peut vous offrir la couleur, aussi une autre version de ce programme est utilisée :

les symboles



La X-07 n'aura pas plus de difficultés qu'auparavant, il en est autrement de nous pour raisonner avec des symboles.

Le programme initialise plusieurs tableaux dont un à deux dimensions E (30, 20) qui mémorise les essais successifs pour trouver la solution. En toute théorie il est possible de jouer avec plus de 3 chiffres dans le nombre à deviner (initialisation en ligne 10), seulement le temps de réflexion du X-07 s'allonge et son besoin en mémoire de la même façon ; et les 6,8 k octets de RAM seraient vite absorbés (les tableaux prennent beaucoup de place en mémoire).

Les lignes de texte où vous constatez que des lettres sont manquantes sont dues au fait que l'imprimante ne représente pas les caractères accentués qui ont été utilisés ici. Ces lettres seront facilement découvertes, et je n'insiste pas...

** MASTERMIND

```
5 DEFINT A,Z: CONSOLE0,4,0,0,0
10 CLS: C=6: N=3: GOSUB 4000
20 GOSUB 2010
30 E$=STRING$(19," ") +CHR$(13)
40 H=RND(-VAL(RIGHT$(TIME$,2)))
150 FOR L=1 TO 4
160 LOCATE 0,2: PRINT"LE JEU COMMENCE...
";CHR$(13);
170 BEEP 0,5: PRINT E$:: BEEP 15,5: NEXT
200 BEEP 12,12: DIM E(20,10): DIM P(20):
DIM U(20)
220 DIM T(20): DIM M(20): J=0: X=0: GOTO
310
240 GOSUB 4000: LOCATE 7,1
250 PRINT "score": IF J>X THEN 270
260 PRINT "EGALITE,";J;"a";X:: GOTO 300
270 IF J>X THEN 290
280 PRINT X;"a";J;"pour MOI";: GOTO 300
290 PRINT J;"a";X;"pour VOUS";
300 BEEP 0,80: BEEP 0,1
310 E=0: Z=0
320 FOR G=1 TO N: P(G)=INT(RND(1)*C+1):
M(G)=P(G)
360 O(G)=INT(RND(1)*C+1): NEXT G
390 E=E+1: CLS: GOSUB 4000: LOCATE 5,1:
PRINT "ESSAI No ";E
410 PRINT "Je propose : ";: BEEP 15,5
420 FOR G=1 TO N: PRINT CHR$(P(G)+127);"
": NEXT G
450 LOCATE 0,3: PRINT "bien plac, sup ?
";: GOSUB 5000: CO=VAL(O$)
460 LOCATE 0,3: PRINT E$;"mal plac, sup
?";: GOSUB 5000
```

```

470 MP=VAL(O$)
480 IF CO+MP>3 THEN BEEP 20,7: GOTO 450
510 CLS: IF CO=N THEN 1300
520 PRINT"Je t'flchis..."
530 FOR G=1 TO N: E(E,G)=P(G): NEXT G
560 E(E,N+1)=CO: E(E,N+2)=MP
590 G=N
600 P(G)=P(G)+1
610 IF P(G)<C+1 THEN 660
620 P(G)=1: G=G-1: IF G>0 THEN 600
660 K=1: FOR G=1 TO N: IF P(G)<>M(G) THE
N K=0: G=N
690 NEXT G
700 IF K=1 THEN 1360
710 K=K+1: IF K>E THEN 930
730 FOR G=1 TO N:T(G)=E(K,G): U(G)=P(G):
NEXT G
770 CO=0: MP=0
780 FOR G=1 TO N
790 IF U(G)=T(G) THEN CO=CO+1: U(G)=-1:
T(G)=-1
800 NEXT G
810 IF CO<>E(K,N+1) THEN 590
820 FOR G=1 TO N: IF T(G)<0 THEN 890
840 FOR L=1 TO N: IF L=G THEN 880
860 IF U(L)<0 THEN 880
870 IF U(L)=T(G) THEN U(L)=-1: T(G)=-1:
MP=MP+1
880 NEXT L
890 NEXT G
900 IF MP=E(K,N+2) THEN 710
910 GOTO 590
930 CLS: GOSUB 4000: PRINT"Votre proposi
tion sup";
940 GOSUB 6000
960 IF U(1)>6 OR U(2)>6 OR U(3)>6 OR U(1

```

```

)=0 OR U(2)=0 OR U(3)=0 THEN 930
1010 CLS: BEEP15,10: PRINT"Dans votre ré
ponse il y a :"
1020 CO=0: MP=0: FOR G=1 TO N: T(G)=O(G)
1050 IF U(G)=T(G) THEN CO=CO+1: U(G)=-1:
T(G)=-1
1060 NEXT G
1070 FOR G=1 TO N: IF T(G)<0 THEN 1140
1090 FOR L=1 TO N: IF L=G THEN 1130
1110 IF U(L)<0 THEN 1130
1120 IF U(L)=T(G) THEN U(L)=-1: T(G)=-1:
MP=MP+1
1130 NEXT L
1140 NEXT G
1150 PRINT CO;"correct";: IF CO>1 THEN P
RINT"s" ELSE PRINT
1170 PRINT MP;"mal plac";: IF MP>1 THEN
PRINT"s";
1180 BEEP 0,60: BEEP 0,1
1200 IF CO<N THEN 390
1210 CLS:BEEP 30,20 : J=J+1: PRINT "BRAU
0, vous GAGNEZ !"
1220 PRINT"en ";E;"essai";: IF E>1 THEN P
RINT"s";
1240 PRINT: INPUT "On recommence ";O$: C
LS: GOTO 240
1300 IF Z=0 THEN PRINT"    J'AI GAGNE !!"
": X=X+1
1305 PRINT "Il fallait trouver :";: BEEP
40,30: LOCATE 7,2
1310 FOR G=1 TO N: PRINT CHR$(127+O(G));
": NEXT G
1350 BEEP 0,60: GOTO 1240
1360 BEEP 24,24: PRINT "vous avez commis
uneERREUR !!": Z=1

```

```

1370 PRINT "PARTIE TERMINEE": GOTO1300
1999 END
2010 PRINT "vous devez trouver une combi-
naison de 3symboles parmi 6.";
2020 GOSUB 2500
2030 PRINT "Ces 6 symboles sont

2040 GOSUB 2500
2050 PRINT "Je chercherai aussi une combi-
naisons de ces symboles ";
2060 GOSUB 2500
2070 PRINT "combinaison que vous me propos-
erez..."
2080 GOSUB 2500: BEEP 24,24
2090 PRINT "ATTENTION, si vous faites un
e erreur j'annule la partie."
2500 BEEP 0,60: BEEP 0,1: CLS: RETURN
4000 BEEP 15,5: PRINT " MASTERMIND ";
: RETURN
5000 BEEP 15,5
5010 O$=INKEY$: IF O$="" THEN 5010
5020 IF ASC(O$)>57 OR ASC(O$)<48 THEN BE-
EP 10,10: GOTO 5000
5030 PRINT O$;: RETURN
6000 CONSOLE0,,2: INPUT O$
6010 IF LEN(O$)<>3 THEN 6060
6020 FOR G=1 TO 3: L= ASC(MID$(O$,G,1))
6030 IF L>133 OR L<128 THEN 6060
6040 V(G)=L-127: NEXT G
6050 CONSOLE0,,0: RETURN
6060 PRINT "recommencez, sup": BEEP 10,1
0: GOTO 6000

```

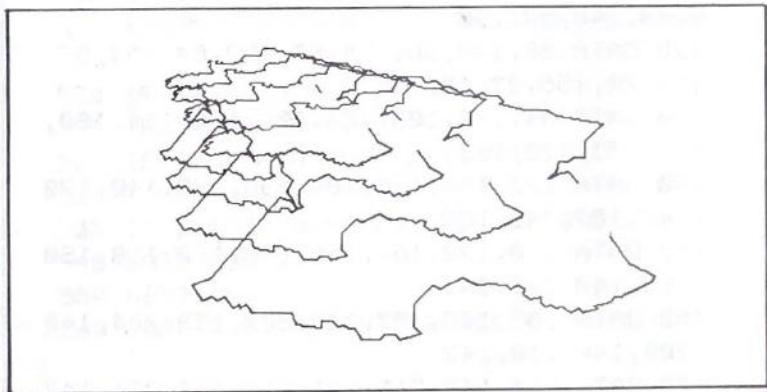
REPRESENTATION DE FIGURES

Dessiner la France

L'opération se résume à transcrire des traits du contour de notre hexagone en une suite de DATAS par rapport à un repère qui va cadrer avec les possibilités de l'imprimante (notamment du point de vue largeur : 480 points possibles).

De la ligne 490 à la ligne 550, une boucle permet de dessiner plusieurs fois la FRANCE sur l'imprimante. On voit aussi que si on applique un facteur correcteur aux coordonnées, il est possible de faire une transformation géométrique du dessin initial (ici c'est seulement une homothétrie...).

LA REPRÉSENTATION DE FIGURES NON GÉOMÉTRIQUES



```

1'*FRANCE*
10 DATA 248,86,241,78,235,80,230,82,234,
78,236,73,241,73
20 DATA 244,71,250,71,250,62,260,52,234,
48,228,45,216,45
30 DATA 203,42,205,41,202,38,196,38,194,
36,190,38,187,35
40 DATA 182,36,176,33,172,33,172,33,168,
30,162,31,160,28

```

```

45 DATA 148,28,144,36,130,37
50 DATA 126,38,124,42,126,43,118,44,116,
46,106,46,104,44
60 DATA 103,46,98,45,99,43,91,43,89,44,9
2,46,91,48,95,54
70 DATA 94,56,90,56,80,56,78,58,74,58,72
,56,68,58,65,57
80 DATA 64,59,60,60,58,58,44,63,43,65,48
,65,44,67,46,68
90 DATA 42,70,47,72,56,70,58,72,67,70,69
,72,78,71,79,72
100 DATA 82,71,84,76,86,75,85,80,91,85,8
8,91,92,96,93,100
110 DATA 91,96,84,94,48,138,38,142,44,15
0,49,148,52,150
120 DATA 56,148,60,154,63,152,64,154,67,
152,70,156,92,156
130 DATA 94,162,108,164,108,160,114,160,
117,163,120,162
140 DATA 123,164,126,164,130,170,140,170
,142,167,148,167
150 DATA 150,170,162,166,158,162,158,150
,168,140,172,142
160 DATA 185,136,192,140,202,139,204,140
,208,140,210,142
170 DATA 214,140,218,141,217,144,221,147
,227,146,227,149
180 DATA 229,151,232,150,235,152,239,151
,243,154,248,152
190 DATA 250,154,258,152,256,150,265,148
,267,145,270,145
200 DATA 287,137,288,134,275,128,264,113
490 FOR M=0.75 TO 1.5 STEP 0.25
500 I=-1:LPRINT CHR$(18)
510 READ X,Y:Y=-Y:I=I+1
515 X=X/M:Y=Y/M

```

```

520 IF I=0 THEN LPRINT"M";STR$(X);",";ST
R$(Y):GOTO 510
530 LPRINT"D";STR$(X);",";STR$(Y)
535 IF I=137 THEN RESTORE 10:GOTO 550
540 GOTO 510
550 NEXT M

```

Conservez ce que vous venez d'écrire et modifiez les lignes 490 à 560 suivant ce qui est donné ci-après et la FRANCE, ô miracle, se dessinera sur l'écran LCD du CANON.

```

490 CLS: FOR M=5 TO 5
500 I=-1
510 READ X,Y:I=I+1
515 X=X/M :Y=Y/M-4: REM 4=OFFSET ECRAN L
CD
520 IF Y>31 THEN Y=31
525 IF X>119 THEN X=119
527 IF I=0 THEN PSET(X,Y):GOTO 535
530 LINE-(X,Y)
535 IF I=137 THEN RESTORE 10:GOTO 550
540 GOTO 510
550 NEXT M
560 LOCATE 6,2:PRINT"FRANCE";CHR$(&H0B);

```

Vous vous êtes sans doute demandé pourquoi l'hexagone était aussi aplati... mais c'est en fait parce qu'il est vu de haut et en perspective... voilà le complément de DATA pour faire le dessin complet... un majestueux 747 d'AIR FRANCE qui est en train de viser ROISSY CHARLES DE GAULLE.

1 * FRANCE *

10 DATA-1, 1, 248, 86, 241, 78, 235, 80, 230, 82,
234, 78, 236, 73, 241, 73
20 DATA 244, 71, 250, 71, 250, 62, 260, 52, 234,
48, 228, 45, 216, 45
30 DATA 203, 42, 205, 41, 202, 38, 196, 38, 194,
36, 190, 38, 187, 35
40 DATA 182, 36, 176, 33, 172, 33, 172, 33, 168,
30, 162, 31, 160, 28
45 DATA 148, 28, 144, 36, 130, 37
50 DATA 126, 38, 124, 42, 126, 43, 118, 44, 116,
46, 106, 46, 104, 45
60 DATA 103, 46, 98, 45, 99, 43, 91, 43, 89, 44, 9
2, 46, 91, 48, 95, 54
70 DATA 94, 56, 90, 56, 80, 56, 78, 58, 74, 58, 72
, 56, 68, 58, 65, 57
80 DATA 64, 59, 60, 60, 58, 58, 44, 63, 43, 65, 48
, 65, 44, 67, 46, 68
90 DATA 42, 70, 47, 72, 56, 70, 58, 72, 67, 70, 69
, 72, 78, 71, 79, 72
100 DATA 82, 71, 84, 76, 86, 75, 85, 80, 91, 85, 8
8, 91, 92, 96, 93, 100
110 DATA 91, 96, 84, 94, 48, 138, 38, 142, 44, 15
0, 49, 148, 52, 150
120 DATA 56, 148, 60, 154, 63, 152, 64, 154, 67,
152, 70, 156, 92, 156
130 DATA 94, 162, 108, 164, 108, 160, 114, 160,
117, 163, 120, 162
140 DATA 123, 164, 126, 164, 130, 170, 140, 170
, 142, 167, 148, 167
150 DATA 150, 170, 162, 166, 158, 162, 158, 150
, 168, 140, 172, 142
160 DATA 185, 136, 192, 140, 202, 139, 204, 140
, 208, 140, 210, 142
170 DATA 214, 140, 218, 141, 217, 144, 221, 147
, 227, 146, 227, 149

180 DATA 229, 151, 232, 150, 235, 152, 239, 151
, 243, 154, 248, 152
190 DATA 250, 154, 258, 152, 256, 150, 265, 148
, 267, 145, 270, 145
200 DATA 287, 137, 288, 134, 275, 128, 264, 113

220 DATA-1, 1, 266, 97, 258, 99, 251, 100, 216, 8
6, 219, 84, 267, 96
230 DATA-1, 1, 275, 89, 285, 34, 276, 37, 253, 83
, 250, 87, 247, 91
240 DATA-1, 1, 281, 95, 301, 92, 280, 97, -1, 1, 2
75, 89, 278, 90, 280, 91
250 DATA 281, 92, 281, 95, 280, 97, 277, 99, 249,
114, 244, 117
260 DATA 239, 118, 188, 111, 186, 106, 213, 110,
214, 115, 213, 112
270 DATA 110, 96, 203, 105, 209, 107, 214, 109, 2
19, 113, -1, 1, 200, 105
280 DATA 201, 101, 204, 96, 206, 94, 210, 92, 213
, 92, 217, 94, -1, 1
290 DATA 214, 94, 232, 94, -1, , 250, 92, 275, 89,
-1, , 265, 106, 328, 100
300 DATA 254, 112, -1, 1, 251, 114, 271, 111, 268
, 108, 259, 109, -1, 1
310 DATA 276, 107, 278, 110, 302, 106, 300, 104,
276, 107, -1, 1
320 DATA 181, 109, 150, 104, 149, 101, 180, 105,
181, 109
330 DATA-1, 1, 205, 95, 208, 95, -1, 1, 202, 104,
205, 104, -1, 1
335 DATA 208, 103, 214, 103, -1, 1, 217, 103, 224
, 103, -1, 1, 228, 102
340 DATA 242, 102, -1, 1, 207, 101, 208, 101, 208
, 105, 206, 105, 207, 101
345 DATA-1, 1, 215, 101, 217, 101, 216, 105, 214
, 105, 215, 101, -1, 1

```

350 DATA226,100,228,100,227,105,225,105,
226,100,-1,1,243,99
355 DATA246,99,245,106,242,106,243,99,-1
,1,287,109,286,111
360 DATA288,114,292,114,297,113,297,110,
292,108,291,112
362 DATA292,114,-1,1
365 DATA297,110,300,106,-1,1,248,115,249
,119,252,120,257,118
370 DATA257,116,252,114,251,116,252,120,
-1,1,257,116,260,113
375 DATA-1,1,184,108,180,112,178,110,179
,108,-1,1,147,103
380 DATA147,107,146,109,146,102
385 DATA 180,116,0,360,1
387 DATA180,115,0,360,2.5
390 DATA179,114,90,360,4
392 DATA147,113,0,360,1
394 DATA147,112,95,440,2.5
396 DATA147,111,95,440,4
398 DATA174,522,70,110,506
490 M=.7
495 FOR I=1 TO 260
500 LPRINT CHR$(18)
510 GOSUB 600
520 IF X>0 THEN 530 ELSE 525
525 GOSUB 600: LPRINT"X";STR$(X);";Y";STR$(Y):GOTO 540
530 LPRINT"D";STR$(X);";Y";STR$(Y)
540 NEXTI
560 FOR P=1 TO 7:'cercles
562 READX,Y,D,F,R
565 X=X/M:Y=Y/M:R=R/M
570 GOSUB 1000
575 NEXT P: END
600 READ X:READ Y:Y=-Y

```

```

610 X=X/M:Y=Y/M:RETURN
1000 'ARC DE CERCLE RAYON R/ COORD X,Y /
1010 PI=3.141592654
1020 FOR I=D TO F STEP 5
1030 Z=X+R*COS(I*PI/180)
1040 T=-Y+R*SIN(I*PI/180)
1050 IF I=D THEN 1060 ELSE 1070
1060 LPRINT"X";STR$(INT(Z));";Y";STR$(INT(T))
1070 LPRINT"D";STR$(INT(Z));";Y";STR$(INT(T))
1080 NEXT I
1090 RETURN

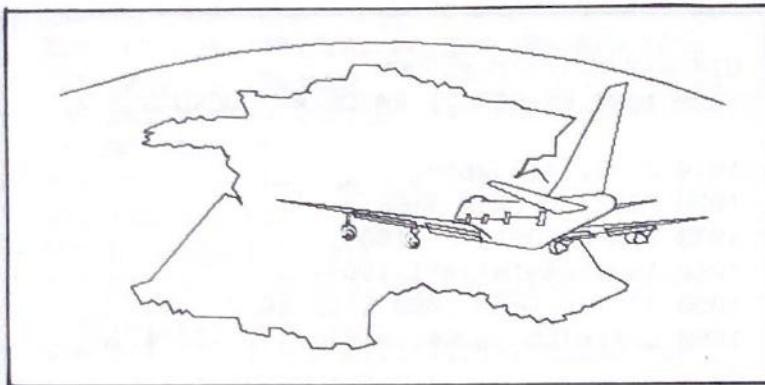
```

Pour palier à l'insuffisance de fonctions mathématiques non prévues à l'origine sur le X-07, voici un programme qui calculera l'arc cosinus d'une valeur comprise évidemment entre -1 et 1. Le X-07 possède tout de même ATN (acrtangente)... heureusement (le ZX81 tant décrié pour ses insuffisances possède les fonction ARCOS, ARCSIN et ARCTAN..., lui !).

Ce qu'il faut éviter dans ce genre de programme est de tomber sur une valeur qui occasionne par exemple une division par zéro (/0 ERROR), ou l'extraction d'une racine carrée d'un nombre négatif puisqu'on utilise SQR.

Quand on sait que $\cos^2 x + \sin^2 x = 1$ et que $\operatorname{tg} x = \frac{\sin x}{\cos x}$, l'obtention de ARCOS x est facile si l'on possède la fonction Arc-tang(x), ce qui est le cas.

Le ON ERROR GOTO évite de tomber dans le piège de la racine carrée d'un nombre négatif.
L'adaptation est aisée pour obtenir l'arcsin x.



```

10 P=3.1415926535898: P1=P/180
20 CLS: ON ERROR GOTO 200
30 GOTO 100
40 T=SQR([(1-U*U)/(U*U)])
50 R=ATN(T): D=R/P1: RETURN
100 BEEP 49,2:INPUT "COS ";U
110 IF U=0 THEN R=P/2: D=90: GOTO 140
120 GOSUB 40
130 IF U<0 THEN R=P-R: D=180-D
140 PRINT TAB(4); "Arccos("; USING "+#.##"
;U;: PRINT ")"
145 PRINT "vaut "; USING "#.#####"; R; :
PRINT " RAD";
150 PRINT "soit ";: PRINT USING "###.###"
###";D;:PRINT " DEG";
160 GOTO 100
200 PRINT "valeur hors gamme",,: BEEP 40
,3: RESUME 100

```

CALCULETTE

Vous êtes en possession d'un X-07 qui de loin ressemble bigrement à une calculette électronique de poche (il cache son jeu remarquez...). Pourquoi ne pas l'utiliser de cette façon ? Évidemment on peut toujours, s'il le faut, calculer la somme de 2 nombres en faisant, par exemple, en mode direct :

PRINT 2.456 + 0.123

ce cher X-07 va répondre correctement 2.579, mais cette façon de procéder est assez lourde, vous en conviendrez.

Aussi, voici un programme qui transforme votre X-07 en calculatrice à notation polonaise inverse.

Elle permettra d'effectuer les classiques opérations +, -, ×, / et l'extraction de racines carrées, possédera une mémoire et quatre registres notés x, y, z et t.

Dans ce qui suit, la touche RETURN sera appelée ENTER.

Il est souvent nécessaire d'avoir deux nombres pour effectuer une opération (opérations +, -, ×, /).

Le calcul portant sur deux nombres va s'effectuer de la même manière que sur un seul nombre, grâce à la touche de fonction ENTER.

Pour effectuer un calcul sur deux nombres, introduire le premier nombre (l'introduction des nombres s'effectue toujours dans le registre x). Appuyer sur ENTER (le contenu de x passe dans y).

Introduire le deuxième nombre.

Appuyer sur le signe +, = * ou / correspondant à l'opération à effectuer.

Vous constatez que x vaut, par exemple, x*y. Si c'est la multiplication que vous avez choisie, les registres x,y,z,t constituent une pile opérationnelle.

Pour obtenir la racine carrée d'un nombre, introduire le nombre, appuyer sur R (comme Racine) et le contenu de x devient égal à la racine du nombre introduit.

Un BEEP et un message d'erreur préviennent les fausses manipulations... (division par zéro, racine négative, dépassement de capacité).

RESUME DES CODES TOUCHES

*	=	+	} obtenues avec shift	les 4 opérations de base notez < = > est le shift de < - >.
---	---	---	-----------------------	--

M mémorise le registre x en mémoire
 CNTL Q rappelle le contenu mémoire dans x
 R extrait la racine carrée de x
 CNTL D établit une rotation des registres, x vaut y, etc... et t vaudra x
 CNTL I permute x et y
 CNTL E efface les 4 registres x, y, z et t.

Ces fonctions sont choisies par un INKEY\$ qui scrute le clavier suivis des classiques IF THEN pour tester le caractère.
 Pour entrer un nombre toutes les touches sont valides sauf celles citées plus haut.
 Tapez <-> pour exécuter un nombre négatif. Si vous faites une erreur, tapez -- qui efface le caractère (RUBOUT).
 Si vous dépassiez le nombre de caractères possibles, un BEEP vous rappelle à l'ordre, de même que l'utilisation du RUBOUT au départ.
 Une opération du genre $4 \times (5 + 4)$ est faite très facilement.

Entrez 4 puis <ENTER>

Entrez 5 puis <ENTER>

Entrez 4 puis <+>

et pour finir <*>

Pas de parenthèse à mettre, pas de signe égal...

Vous allez sans problème pouvoir ajouter des fonctions du type LOG, etc. sur les mêmes principes.

Notez l'usage des fonctions STR\$, VAL() et le formatage écran par le PRINT USING" & ", le test des valeurs avec exposant.

N.B. : ce type de calcul en notation polonaise inverse est classique sur les calculatrices Hewlett Packard, l'avantage ici c'est que l'on voit évoluer les registres.

On peut entrer des nombres sous forme hexadécimale &H... le fait d'appuyer sur ENTER convertit le nombre en décimal (une fonction de plus...).

```

10 CLS: CLEAR 200: E$=STRING$(17," "):ON
      ERROR GOTO 1000
20 GOTO 80
28 IF LEFT$(Z$,1)="-" THEN Z$="-"+Z$
30 PRINT E$;" ";CHR$(13);A$;;IF LEFT$(RIGHT$(Z$,4),1)="E"THEN 50
40 PRINT USING "&           &";Z$;; RETURN
50 PRINT USING "&           &";LEFT$(Z$,LEN(Z$)-4);
60 PRINT " E";RIGHT$(Z$,3);: RETURN
  
```

```

70 RETURN
80 LOCATE 0,0:Z$=STR$(T): A$="t":GOSUB 2
8
90 LOCATE 0,1:Z$=STR$(Z): A$="z":GOSUB 2
8
100 LOCATE 0,2:Z$=STR$(Y): A$="y":GOSUB 28
105 IF E THEN LOCATE 0,3:PRINT "x ERREUR ";
      E=0: BEEP 49,5
110 LOCATE 0,3:Z$=STR$(X): A$="x":GOSUB 28
120 I=1: LOCATE 2,3: INIT#1,"KBD:"
130 CONSOLE@,,0: A$=INKEY$: IF A$="" THE
N 130
132 A=ASC(A$):IF A=29 THEN 170
134 IF A=4 THEN 550
135 IF A=82 THEN 800
136 IF A=9 THEN 600
137 IF A=77 THEN 700
138 IF A=5 THEN 650
139 IF A=17 THEN 750
142 IF A=13 THEN F=-1: GOTO 300
144 IF A=42 THEN 350
146 IF A=47 THEN 400
148 IF A=43 THEN 450
150 IF A=61 THEN 500
152 IF A<33 THEN BEEP 49,5: GOTO 130
154 IF F THEN PRINT E$;; LOCATE 2,3: F=0
      : X$=""
158 I=I+1: IF I=18 THEN I=I-1: BEEP 49,5
      : GOTO 130 ELSE 160
160 LOCATE I,3: PRINT A$;; X$=X$+A$: GOT
O 130
170 IF I=1 THEN BEEP 49,5: GOTO 130
180 I=I-1: LOCATE 1+I,3: PRINT " ";:LOCA
TE 1+I,3
  
```

```

190 X$=LEFT$(X$,LEN(X$)-1): GOTO 130
300 X=VAL(X$): T=Z: Z=Y: Y=X: GOTO 80
350 X=VAL(X$): X=XXY
360 X$=STR$(X): Y=Z: Z=T: F=-1: GOTO 80
400 X=VAL(X$): X=Y/X: GOTO 360
450 X=VAL(X$): X=Y+X:GOTO 360
500 X=VAL(X$): X=Y-X: GOTO 360
550 X=VAL(X$):T1=X:X=Y:Y=Z:Z=T:T=T1:X$=S
TR$(X): F=-1:GOTO 80
600 X=VAL(X$):T1=Y: Y=X: X=T1: X$=STR$(X
1: F=-1: GOTO 80
650 X=0:X$="" :Y=X:Z=X:T=X:F=-1: GOTO 80
700 BEEP 20,1: X=VAL(X$): M=X; F=-1: GOT
D 120
750 BEEP 20,1: X=M: X$=STR$(X): F=-1: GO
TO 80
800 X=VAL(X$):X=SQR(X): X$=STR$(X): F=-1
: GOTO 105
1000 BEEP 49,5: E=-1: F=E: RESUME 80.

```

*** CALCULETTE ***

DUMP MÉMOIRE

Après avoir décrit la façon dont était structurée la "zone texte" du CANON X-07, voici un programme qui va lire la mémoire et afficher le contenu en hexadécimal sur l'imprimante. Le programme demande l'adresse de départ et l'adresse de fin et affiche 16 octets

derrière l'adresse du premier octet, et ensuite s'affiche la correspondance ASCII du contenu de chaque octet mémoire. Vous pourrez ainsi repérer les messages internes que peut vous envoyer ce cher X-07. A la ligne 55, le X-07 envoie l'ordre LPRINT[1,1] pour dire à l'imprimante d'imprimer en caractères réduits, et en couleur bleue. Autrement, le formatage espéré ne serait pas possible, quant à la couleur, question de goût...

A la ligne 60, l'octet lu par la fonction PEEK (adresse) est transcodé en hexadécimal par la fonction HEX\$ et afin d'éviter les décalages suivant la grandeur du nombre, on place un 0 devant le premier chiffre si nécessaire. LPRINT USING aurait pu formater correctement*, sauf placer le 0 et le résultat est ainsi bien meilleur (point de vue esthétique). Pour éviter d'inscrire des codes ASCII non représentables, ou hors gamme, le sous-programme 100 teste la validité du code lu (Q) et transforme ensuite par CHR\$, Q en son caractère correspondant. Le code ASCII s'étend de la valeur 0 à 127, soit 00 à 7FH, le reste (804 à FFH) est la disposition pour faire ce qu'on veut, notamment le constructeur du X-07 les utilise pour faire des caractères spéciaux, et semi-graphiques. A ce propos, les caractères KÄNA sont d'une utilité surprenante pour un européen...

*LPRINT USING "& &" pour réservier deux caractères maximum.

```

1 CLEAR 100:DEFINT A-Z
5 CLS:PRINT"* DUMP MEMOIRE *"
10 INPUT"ADRESSE DE DEPART    ";A
20 LOCATE 0,2:INPUT"ADRESSE DE FIN
";B
40 FOR I=A TO B
45 IF F<>0 THEN 60
50 A$=HEX$(I):A$=MID$("0000",1,4-LEN(A$)
)+A$
55 LPRINT[1,1]A$;" ";
60 Q=PEEK(I):D$=HEX$(Q):IF LEN(D$)=1 THE
N D$="0"+D$
63 GOSUB 100
65 LPRINT D$;" ";
70 F=F+1:IF F=16 THEN GOSUB 130
80 NEXT
82 IF F<16 THEN LPRINT STRING$(48-3*F,"")
:;GOSUB 130
85 LPRINT CHR$(10)

```

```

90 INPUT "TAPEZ <RETURN> POUR REFAIRE UN
DUMP      ";A$
95 RUN
100 IF Q>&H1F THEN 110 ELSE 115
110 IF Q<&H7F THEN M$=M$+CHR$(Q):RETURN
115 M$=M$+"":RETURN
130 F=0:LPRINT"      ";M$:M$=""
140 RETURN

```

Programme d'examen du contenu mémoire du CANON X-07 (en hexadecimal)

Je propose une autre version de ce programme afin de l'utiliser avec une imprimante classique CENTRONICS comme la Microline 82A sur laquelle a été tiré le listing modifié. La variable P compte les lignes affichées et envoie un FORM FEED vers l'imprimante, une fois que la page a été imprimée (~ 60 lignes) pour éliminer l'impression sur le pliage du papier.

```

1 REM VERSION POUR ULINE 82A
2 CLEAR 200:DEFINT A-Z
5 CLS:PRINT "* DUMP MEMOIRE *"
10 INPUT "ADRESSE DE DEPART ";A
20 LOCATE 0,21:INPUT "ADRESSE DE FIN      ";B
40 FOR I=A TO B
45 IF F<0 THEN 60
50 A$=HEX$(I):A$=MID$("0000",1,4-LEN(A$))+A$
55 LPRINT A$;" ";
60 Q=PEEK(I):D$=HEX$(Q):IF LEN(D$)=1 THEN D$="0"+D$
65 GOSUB 100
65 LPRINT D$;" ";
70 F=F+1:IF F>16 THEN 80 ELSE 75
75 GOSUB 130
80 NEXT
82 IF F>16 THEN LPRINT STRING$(48-3*F," ");:GOSUB 130
85 LPRINT CHR$(10)
90 INPUT "TAPEZ <RETURN> POUR REFAIRE UN DUMP      ";A$
95 RUN
100 IF Q>&H1F THEN 110 ELSE 115
110 IF Q<&H80 THEN M$=M$+CHR$(Q):RETURN
115 M$=M$+"":RETURN
130 F=0:LPRINT"      ";M$:M$="" P=P+1
140 IF P=60 THEN LPRINT CHR$(12):P=0
150 RETURN

```

Dans ces programmes, l'utilisation de la fonction de concaténation de chaînes : M\$ = M\$ + CHR\$(Q), nécessite une place plus grande que la zone initialement réservée pour le X-07, à savoir 50 octets, d'où le CLEAR 200 de la ligne 2 (qui initialise à zéro les variables ainsi que les chaînes de caractères (chaîne nulle)).

RENUMÉROTATION

Connaissant la structure interne de la zone texte pour programme, je propose ici un petit programme de renumérotation de ligne écrit en BASIC et qui permettra, moyennant l'intervention manuelle sur les GOTO et GOSUB, etc... internes aux lignes de programme, d'ajouter une ligne de programme entre deux des numéros consécutifs. Avec ce qui a été dit précédemment, le programme est facilement compréhensible, d'autant plus qu'il y a des REMarks.

Le programme, une fois écrit, sera sauvegardé en zone RAM pour fichiers et lancé par un RUN suivi du nom sous lequel il est répertorié. Ex.: RUN "RENUM" et un autre programme situé en zone texte sera renuméroté.

1 PRINT "123"	10 PRINT "123"
2 GOTO 1	20 GOTO 1
3 END	30 END

```

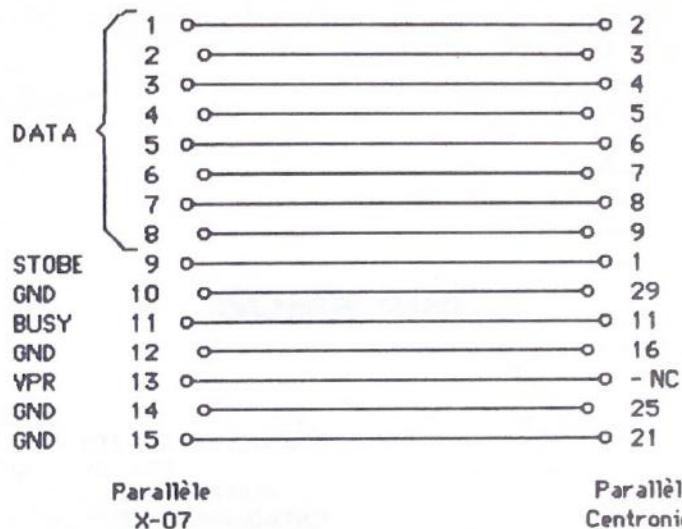
10 REM* RENUM simple *
20 DEFINT I-L:DEFSNG A-F,M-Z
30 A=&H553: REM ADRESSE DEBUT PROGRAMME
40 B=PEEK(A)+256*PEEK(A+1): REM Add DEBU
T proch. ligne
50 IF B=0 THEN PRINT"Ok": END : REM renu
m. termine
60 C=C+10: REM PAS de renumeration
70 O=PEEK(A+2)+PEEK(A+3)*256

```

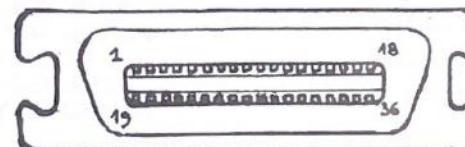
```

80 PRINT 0;">";C
90 POKE A+2,(C-256*INT(C/256)): REM renu
merciation LSB
100 POKE A+3, INT(C/256): REM renumerotat
ion MSB
110 A=B: REM preparation ligne suivante
120 GOTO 40

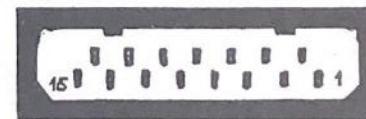
```



Interface pour imprimante avec entrée parallèle Centronics



Prise type Amphénol 57 303 60
Vue côté connecteur mâle



Prise sortie parallèle X07
cordon imprimante

SOMMAIRE

PRÉFACE	7
DÉCOUVERTE DU CANON X07	9
LE BASIC DU X07	19
X07 ET COMMUNICATION	33
PROGRAMMES D'APPLICATION	43
— LOTO	43
— COURBES	45
— RÉVEIL MUSICAL	46
— PROGRAMMES MUSICAUX	53
— JEUX INTERDITS	56
— LE BEAU DANUBE BLEU	58
— SÉRÉNADE DE SCHUBERT	60
— ON THE TOP OF OLD SMOKY	63
— MORSE	66
— FACTEURS PREMIERS	69
— HARD COPY D'ECRAN	70
— D.A.O.	72
— KALEÏDOSCOPE	87
— SIMULATION D'ATERRISSAGE	87
— MASTERMIND	97
— REPRÉSENTATION DE FIGURES	101
— CALCULETTE	109
— DUMP MÉMOIRE	112
— RÉNUMÉROTATION	115

Composition : FIDELTEX
Maquette : SORACOM
Impression : JOUVE MAYENNE
Partitions : BONTEMPI
N° d'éiteur : 41
Dépôt légal : 2^e trimestre 1985

© Editions SORACOM — 1985

ISBN 2-904032-31-2