

TELESTRAT

MANUEL  
D'UTILISATION

VERSION 1.2  
AVRIL 1986

©



INTERNATIONAL

*Eureka Informatique*

ATTENTION !

- Si certaines pages annoncées dans le sommaire ne se trouvent pas dans le livre,
- si quelques fautes d'orthographe ou de frappe trainent encore entre les lignes,
- si vous ne trouvez pas tous les éclaircissements que vous pourriez souhaiter sur toutes les possibilités de ce merveilleux appareil qu'est le TELESTRAT,

pardonnez-nous !

Le TELESTRAT est un produit en constant perfectionnement, et nous améliorons chaque jour les documents qui l'accompagnent, celui-ci étant une version provisoire.

Dès que la version définitive sera imprimée, vous pourrez l'obtenir en échange de ce manuel et sur simple présentation de votre carte de garantie.

Bon courage.. et bonne télématique !

# TABLE DES MATIERES

## **I - MISE EN MARCHIE**

- 1 - DEBALLAGE
- 2 - DESCRIPTION DU MATERIEL
- 3 - BRANCIEMENTS
- 4 - DEMARRAGE
- 5 - UN PEU DE PRATIQUE
  - a - Créer un programme
  - b - Execution du programme
  - c - Sauvegarde
  - d - Effacement de la mémoire centrale
  - e - Chargement du programme

## **II - LE CLAVIER**

- 1 - DESCRIPTION
  - a - Touches de caractères
  - b - Touches de fonction
  - c - Touches numériques
- 2 - REFERENCE
  - a - Les caractères imprimables
  - b - Les caractères de contrôle
- 3 - CONFIGURATIONS DE CLAVIER
- 4 - PROGRAMMATION DES TOUCHES DE FONCTION

## **III - LES CARTOUCHES ENFICHABLES**

- 1 - INSTALLATION
- 2 - LE PORT LANGAGE
- 3 - LE PORT UTILISATEUR

## **IV - PRINCIPE DE FONCTIONNEMENT**

- 1 - LE MATERIEL
  - a - La mémoire vive
    - Définition d'un octet

- L'espace mémoire
- b - Le microprocesseur 6502 A
  - Les registres
- c - La mémoire morte
- d - Les interfaces

## **V - LE LANGAGE DE BASE : L'HYPER-BASIC**

### **1 - CARACTERISTIQUES GENERALES**

- a - Quelques définitions

### **2 - LA TRADUCTION D'UN LANGAGE**

### **3 - GENERALITES DE PROGRAMMATION**

- a - Données numériques et chaînes de caractères
- b - Les noms de variables
- c - Les opérateurs arithmétiques
- d - les opérateurs logiques

### **4 - LES BOUCLES ET BRANCHEMENTS DE PROGRAMME**

### **5 - LES LABELS ET LES PROCEDURES**

- a - Definition
- b - Les different types

### **6 - L'EDITEUR DE PROGRAMME**

## **VI - GRAPHISMES ET SONS**

### **1 - GENERALITES . L'AFFICHAGE .**

### **2 - LE MODE TEXT**

### **3 - LE MODES LORES Ø**

### **4 - LE MODE LORES 1**

### **5 - LE MODE HIRES**

### **6 - LES EFFETS SONORES**

### **7 - LA MUSIQUE**

## **VII - LE LECTEUR DE DISQUETTES ET LE STRATSEED**

### **1 - POURQUOI UNE UNITE DE DISQUETTE 3" ?**

### **2 - PRESENTATION**

### **3 - MANIPULATION**

### **4 - LE REPERTOIRE DE LA DISQUETTE**

### **5 - LE SYSTEME D'EXPLOITATION : LE STRATSEED**

- a - généralités
- b - Les noms de fichiers
- c - Les instructions de travail sur disque

charger un fichier  
sauvegarder un fichier  
d - Gestion de fichiers

## **VIII - COMPATIBILITÉ AVEC LES ORDINATEURS ORIC**

## **IX - PÉRIPHÉRIQUES**

- 1 - LECTEURS DE DISQUETTES AUXILIAIRES
- 2 - LES IMPRIMANTES
- 3 - LES MONITEURS / ECRANS
- 4 - MAGNETOPHONE
- 5 - LEVIERS DE JEU ET SOURIS
- 6 - MINITEL
- 7 - NOYAU MIDI
- 8 - MODEM
- 9 - EXTENSION DE L'ORDINATEUR

## **10 - ANNEXES**

- 1 - SPECIFICATIONS
- 2 - TABLE DES TOUCHES DE FONCTIONS
- 3 - CONFIGURATIONS DE CLAVIER
- 4 - TABLE DES MOTS-CLES ET DES CODES DE FONCTIONS
- 5 - TABLE DES CODES ASCII
- 6 - TABLEAU DES FONCTIONS CTRL ET ESC
- 7 - STRUCTURE DE LA MEMOIRE
- 8 - LES MOTS-CLES RESERVES DU BASIC

## PREAMBULE

Ce manuel d'utilisation va vous permettre d'apprendre à manipuler le Telestrat. Les sujets sont classés par thèmes d'utilisation et un index détaillé se trouve à la fin du volume.

Le mode d'emploi détaillé du BASIC et du STRATSED se trouve dans le **MANUEL DE REFERENCE**, consultez-le chaque fois que vous voulez comprendre une instruction dans le détail.

Le mode d'emploi des applications Minitel et de la cartouche **TELEMATIC** se trouve sur le volume **LES APPLICATIONS TELEMATIQUES**.

Chacun de ces manuels possède un index permettant de trouver des informations précises sur un domaine, nous vous invitons à l'utiliser dès qu'un point vous semble obscur, au cours de votre lecture.

Le Telestrat apporte un concept nouveau à l'informatique : il permet de s'adapter à la communication informatique sans difficultés techniques. Malgré sa simplicité d'utilisation, il donne accès aux applications les plus évoluées de la télématique et de la programmation. Le Telestrat possède des caractéristiques rares sur un micro - ordinateur : **BASIC compilé en ligne, définition de procédures, éditeur de programme pleine page, serveur videotex monovoie intégré...** Nous n'aborderons ici que les principes d'utilisation concernant l'utilisation habituelle d'un micro-ordinateur normal (c'est-à-dire noir et rouge bien sûr!), la télématique donne lieu à un manuel à part. La programmation du Telestrat est conçu pour être facile et rapide. Avec l'éditeur pleine page, vous ne perdez plus de temps pour corriger les erreurs. D'autre part, la compilation rend l'exécution des programmes beaucoup plus rapide qu'avec un langage interprété.

L'**HYPER-BASIC**, développé par ORIC, est un des plus puissants BASIC disponibles sur micro-ordinateur, toutes catégories confondues:

- Exécution de certaines boucles 200 fois plus rapide qu'avec un BASIC interprété, exécution globale d'un programme de 2 à 100 fois plus rapide.
- Plus de 200 commandes évoluées.
- Contrôle des Entrées/Sorties (Canaux, instruction **RS232...**).
- Pas d'encombrement du répertoire avec des fichiers multiples "sources" et "objets" comme dans une application compilée classique.

Quand au système d'exploitation, le **STRATSED** (SED pour Système d'Exploitation de Disques et STRAT parce que tel est STRAT !!), il offre les commandes des systèmes professionnels les plus réputés : **RENUMBER**, une étoile en guise de "joker", **PROTECT**, **MERGE**, **AUTO**, **OLD**, **RESET**... Il est optimisé pour ne pas utiliser la mémoire disponible pour les programmes de l'utilisateur et être aussi rapide qu'un système tournant sur des machines 16 bit - le microprocesseur du Telestrat est un **M6502A** 8 bit : 21 Ko chargés par seconde en lecture et en écriture de fichiers, pas mal non?

# I - MISE EN MARCHIE

## I - DEBALLAGE

Après avoir ouvert la boîte de polystyrène, vous devez trouver :

- 1 ordinateur Telestrat (unité centrale avec clavier).
- 1 transformateur d'alimentation 220 V/ 50 Hz.
- 2 cartouches enfichables (HYPER-BASIC et TELEMATIC).
- 3 manuels ( UTILISATION, APPLICATIONS TELEMATIQUE, MANUEL DE REFERENCE).
- 1 jeu d'autocollants pour les touches du clavier.
- 1 carte de garantie.
- 1 câble PERITEL pour le moniteur.
- 1 connecteur en nappe pour le lecteur de disques.
- 1 câble de liaison Minitel.

Dans le 2° emballage, vous trouverez

- 1 lecteur de disquettes MICRODISC.
- 1 disquette formatée pour vos applications.

Si il manque un de ces éléments, veuillez contacter votre revendeur **ORIC**, il veillera à remplacer les articles manquants.

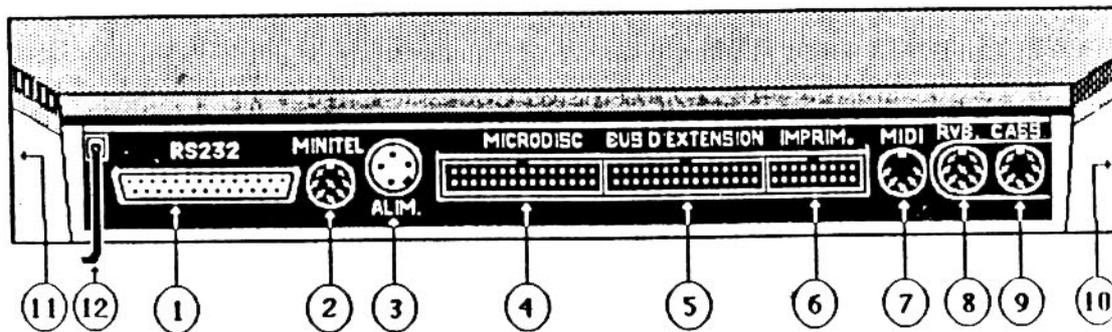
Nous vous conseillons de remplir immédiatement le bon de garantie et de nous le renvoyer, il sera la preuve de votre achat et nous permettra d'assurer notre garantie. **ORIC INTERNATIONAL** fabrique ses produits en France et les garantit un an pièces et main d'œuvre.

## 2 - DESCRIPTION DU MATERIEL

Tourner le Telestrat pour avoir le panneau arrière sous les yeux, vous remarquez :

- La prise de sortie vidéo (RVB) [8].
- La prise d'entrée / sortie magnétophone [9].
- L'interface RS232 [1].
- La sortie CENTRONICS parallèle pour imprimante [6].
- La prise du noyau MIDI (applications de musique électronique) [7].
- L'interface Minitel [2].
- L'extension de BUS complet [5].

- La prise du lecteur de disquettes [4].
- La prise d'alimentation [3].



Certaines prises sont couvertes par des caches en plastique. **Il est important de remettre ces caches quand les prises sont inocuppées.** Des pannes graves peuvent se produire si les connecteurs entrent en contact avec des sources de courant. En particulier, la fiche d'alimentation sortant du transformateur ne doit jamais toucher d'autres prises que celle qui lui est réservée.

- Les prises pour les manettes de jeu ou la souris se trouvent sur les cotés de l'appareil [10 et 11].

• Les connecteurs de cartouches sont protégés par un volet rabattable qui doit être fermé pendant le fonctionnement, cela évite une manipulation des cartouches par inadvertance. Le volet tient fermé par simple pression. Les connecteurs ont des détrompeurs qui ne permettent l'enfichage de la cartouche que dans un seul sens. Les cartouches ne peuvent donc se mettre que dans un seul sens (le logo ORIC vers soi, le bord incliné de la cartouche portant son nom face à l'utilisateur). Les détrompeurs empêchent également de mettre une cartouche "langage" dans le port "application" et vice-versa.

La grille de trous située sur le coté gauche du volet n'a pas de fonction mais rassurez-vous, le Telestrat envoie tous les sons émis par son générateur dans le haut-parleur du moniteur ou du téléviseur qui lui est relié (prise RVB [2]).

- A coté de la prise de la souris, sur le coté droit se trouve le bouton de reset. Une pression sur le reset pendant le fonctionnement interrompt la tâche courante du Telestrat sans vider sa mémoire (reset "à chaud"). Une pression simultanée sur le bouton de reset et sur la touche DEL relance tout le système, charge le DOS, fait les tests internes de mémoire et affiche le menu d'accès au BASIC ou à la cartouche d'application présente (reset "à froid").

### 3 - BRANCHEMENTS

- a - Raccordez l'alimentation à l'unité centrale et au lecteur de disquettes mais ne la

branchez sur le secteur que quand toutes les extensions sont connectées.

b - Branchez un écran sur la sortie RVB :

- Si il s'agit d'un Minitel, reportez-vous au manuel APPLICATIONS TELEMATIQUES et au mode d'emploi du Minitel.

- Si il s'agit d'un téléviseur ne possédant pas de prise PERITEL, vous devez insérer un modulateur haute fréquence (UHF) entre le Telestrat et la prise d'antenne du poste de télévision.

c - Placez la cartouche dans son port : HYPER-BASIC se met dans l'emplacement gauche (LANGAGE). Vous pouvez laisser libre le port droit (UTILISATEUR). La cartouche TELEMATIC, fournie avec l'appareil, ne sert qu'aux applications utilisant le Minitel.

d - Connectez le lecteur de disquettes avec le cable en nappe.

e - Branchez finalement les appareils sur le secteur.

f - Insérez la disquette dans le lecteur.

g - Allumez le moniteur puis l'alimentation de l'ordinateur.

#### **4 - DEMARRAGE**

A la mise sous tension, la disquette contenant une copie du système d'exploitation va être lue pour charger le STRATSED en mémoire centrale. Une fois cette opération terminée, des messages indiquant la version du système et la mémoire disponible se succèdent en haut de l'écran. En supposant que la cartouche TELEMATIC soit présente, voilà ce que cela donne :

```
TELESTRAT
(c) 1986 ORIC International
64 Ko RAM, 24 Ko ROM
```

```
DRIVE : A
```

```
TELEMON V1.0a
(c) 1986 ORIC International
STRATSED V1.0
(c) 1986 ORIC International
TELEMATIC V1.0
(c) 1986 ORIC International
```

```
1 - TELEMATIC
2 - HYPER BASIC
```

Le choix se fait en utilisant les flèches de direction (↓ et ↑), vous déplacez la "fenêtre" de sélection, c'est à dire que vous allumez en vidéo inverse l'option que vous avez choisie. Quand celle-ci est "allumée", vous validez le choix en appuyant sur la barre d'espace.

L'initialisation a suivi une procédure systématique :

- Effacement de l'écran du moniteur, chargement du SED, tests divers de la mémoire, détection des cartouches enfichées.
- Si aucune cartouche UTILISATEUR n'est présente (port droit), exécution du programme **BONJOUR.COM** (programme que vous avez écrit auparavant pour faire exécuter certaines commandes automatiquement lors de l'initialisation), mise en attente en BASIC.
- Si une cartouche UTILISATEUR est présente, lire la disquette.
  - si celle-ci a été enregistrée comme "prioritaire", pour exécuter le programme **BONJOUR.COM** si il en existe un puis mettre en attente en BASIC.
  - si celle-ci n'est pas désignée comme "prioritaire", afficher un menu pour aller dans le BASIC ou dans les applications de la cartouche. Selon les cartouches enfichées, si le lecteur de disquettes n'est pas branché ou si plusieurs le sont, les messages d'initialisation seront différents.

## **5 - UN PEU DE PRATIQUE**

Voici un exemple simple et stimulant ! Il vous permet de comprendre les bases de programmation du Telestrat. Nous vous proposons de créer un programme, cela ne prendra que quelques minutes. Il n'est pas important que vous le compreniez tout de suite, bientôt vous pourrez faire beaucoup mieux :

### **a - CREER UN PROGRAMME**

Ce programme illustre la commande **FILL** ("remplir" en français) pour colorer des zones d'écran.

Comme pour tout programme en BASIC, vous indiquez avant chaque ligne d'instruction un numéro qui permet au Telestrat d'exécuter le programme dans l'ordre des commandes : la ligne 5 est une commentaire, le Telestrat ignore ce qui suit une instruction **REM** (l'utilité des commentaires vous apparaîtra quand vous serez un bon programmeur, utilisant des listings de plusieurs centaines d'instructions), il saute ensuite à la ligne 10, exécute l'instruction puis va en 20 puis de 20 à 30, 40...

Tapez le texte suivant comme il se présente, en appuyant sur la touche RETURN avant de commencer une ligne portant un nouveau numéro. Le curseur se place alors sur la ligne vide suivante :

```
5 REM COLORIAGE
1Ø PAPER Ø
2Ø HIRES : PRINT CHR$ (17)
3Ø REPEAT
4Ø XØ = RND (1) * 231
5Ø YØ = RND (1) * 175 + 7
6Ø AØ = RND (1) * (23Ø-XØ) / 6 + 1
7Ø DØ = RND (1) * (182-YØ) + 8
8Ø CURSET XØ + 6, YØ - 7, Ø : FILL DØ + 7, AØ, 16
9Ø CURSET XØ, YØ, Ø : FILL DØ, AØ, (17 + RND (1) * 7)
10Ø UNTIL FALSE
11Ø END
```

Si vous vous rendez compte que vous avez fait une faute de frappe avant de valider la ligne de commande par RETURN, appuyez sur la touche DEL qui efface les caractères qui précèdent le curseur.

Si vous constatez l'erreur après la validation, placez-vous après la faute avec les flèches de déplacement du curseur, appuyez sur DEL autant de fois qu'il y a de caractères à effacer, tapez les lettres exactes et confirmez par la touche RETURN :

Si vous aviez tapé : **1Ø PAPER 4**

et que vous constatez l'erreur après avoir appuyé sur RETURN, allez derrière le 4, tapez sur DEL, tapez Ø puis RETURN, la nouvelle ligne 1Ø remplacera l'ancienne en mémoire.

**Remarque** : pour changer le texte du programme , vous pouvez aussi retaper la ligne complète : **1Ø PAPER Ø**

puis RETURN, vous aurez ainsi remplacé l'ancienne ligne 1Ø.

## b - EXECUTION DU PROGRAMME

Pour lancer le programme, tapez : RUN puis la touche RETURN pour valider la commande.

Pour arrêter l'exécution, tapez sur les deux touches CTRL et C  
Le message BREAK IN ... apparaît, indiquant à quelle ligne l'interruption s'est produite.

Ce programme réside en mémoire centrale du Telestrat. Si vous éteignez l'appareil, celui-ci perdra les instructions.

Revoyons le programme en tapant **LIST** puis la touche **RETURN**. Pour le conserver, nous allons le transférer sur disquette.

### c - SAUVEGARDE D'UN PROGRAMME

Tapez : **SAVE "COULEUR"** puis la touche **RETURN** pour valider l'ordre.

Vous demandez ainsi le stockage sur disquette du programme sous le nom de : **COULEUR**

Vous n'avez pas besoin d'indiquer ce qu'il faut sauvegarder puisque le Telestrat ne travail qu'avec un programme à la fois dans sa mémoire centrale.

Le programme est enregistré sur la disquette mais il reste aussi en mémoire vive. En tapant à nouveau **RUN**, vous relancerez l'exécution.

### d - EFFACEMENT DE LA MEMOIRE CENTRALE

Pour effacer le programme, vous n'avez pas besoin de couper le courant ! Il suffit de taper : **NEW** ("nouveau" en français), cela libère la mémoire pour commencer un nouveau programme. Vérifiez que la mémoire vive est libre en tapant **LIST** puis la touche **RETURN** : aucun programme ne défile à l'écran (une astuce du Telestrat est de pouvoir rappeler un programme effacé par un **NEW** : en cas d'erreur, il suffirait de taper **OLD**).

Cette parenthèse fermée, nous allons rappeler le programme déjà stocké.

### e - CHARGEMENT DU PROGRAMME

Tapez : **LOAD "COULEUR"** puis **RETURN**

Le programme se charge presque instantanément en mémoire centrale, vous pouvez le vérifier en tapant **LIST** pour l'afficher ou **RUN** pour l'exécuter. N'oubliez pas d'appuyer sur la touche **RETURN** pour valider chaque instruction.

Vous voila arrivé à la fin de ces premiers pas, nous allons maintenant étudier les caractéristiques détaillées de la "machine à communiquer".

## II - LE CLAVIER

Le clavier du TELESTRAT est QWERTY à l'origine, pour faciliter les applications de programmation. De nombreux programmeurs préfèrent en effet utiliser un clavier de type anglais. N'oublions pas que les premiers micro-ordinateurs étaient anglo-saxons!

Une particularité du TELESTRAT, qui est conçu et fabriqué en France, est de pouvoir obtenir par une seule commande (AZERTY, FRENCH...) un clavier français AZERTY accentué.

Pour des applications fréquentes, propres à chaque utilisateur, les touches peuvent aussi être configurées une à une (voir KEYDEF, KEYUSE dans la manuel de référence).

### I - DESCRIPTION

Le clavier comporte 58 touches dont 11 touches rouges de fonctions. Les touches sont à répétition automatique. Si vous gardez le doigt sur une touche plus d'une seconde, le caractère qu'elle représente est répété. Elle sont configurables à partir du BASIC ou d'un logiciel d'application comme un jeu.

#### a - TOUCHES DE CARACTERES

A chaque mise en service, ces touches sont en mode majuscule (CAPS est affiché en haut et à droite de l'écran).

La bascule qui verrouille et déverrouille les majuscules est actionnée en appuyant sur les touches CTRL-T. Dans le mode minuscule, la touche SHIFT sert à obtenir les majuscules, comme sur une machine à écrire.

**Remarque :** les commandes sont toujours traduites en majuscules par l'ordinateur. Même si on les tape en minuscules, elles sont comprises par le TELESTRAT, et au premier LIST, on les retrouve affichées en majuscules !

#### b - TOUCHES DE FONCTIONS

Voir en annexe B les fonctions accessibles par les touches rouges..

**CTRL :** pressée en même temps qu'une autre touche, permet d'accéder à certaines commandes. Nous venons de voir la bascule de déverrouillage majuscule (CTRL-T), on peut aussi citer l'interruption de l'exécution d'un programme BASIC

(CTRL-C).

**ESC** : ~~pressée et relachée avant une autre touche~~, donne accès aux attributs d'écriture et à la synchronisation de l'écran (ESC R rend le fond vert, ESC X synchronisé l'écran à 60 Hz, qui est la fréquence du courant alternatif aux USA...).

**SHIFT** (2 touches) : pressée **en même temps qu'une autre touche**, permet d'accéder aux majuscules pour les touches de caractères et aux signes en indices sur les touches numériques. Elle n'a pas d'autre effet dans le mode majuscule (CAPS affiché en haut de l'écran).

**FUNCT** : ~~pressée en même temps qu'une autre touche~~, permet d'afficher un mot-clef du BASIC ou du système d'exploitation (FUNCT-G génère GOSUB qui est une instruction BASIC). Cette fonction évite la frappe d'instructions fréquemment utilisées.

**REMARQUE** : L'instruction **PRINT**, une des plus employée en programmation BASIC, peut être remplacée par le signe [?] : ? A affiche le contenu de la variable A comme **PRINT A**.

**DEL** : efface le caractère situé derrière le curseur, jusqu'au début de la ligne si l'on maintient la touche enfoncée.

**RETURN** : valide une commande ou une ligne de commande dans un programme et passe à la ligne suivante. Impose un retour chariot comme sur une machine à écrire, la touche sert donc aussi au déplacement du curseur.

[←], [↓], [↑], [→] : Ces touches déplacent le curseur dans les 2 dimensions. Elles servent à accéder à une ligne pour la corriger.

## c - TOUCHES NUMERIQUES

Ces 10 touches sont accessibles en mode minuscule ou majuscule (CAPS), la touche SHIFT permettant d'accéder aux signes inscrits sur la partie supérieure des touches.

## 2 - REFERENCE

Les caractères accessibles par le clavier peuvent donc être divisés en deux groupes :

a - LES CARACTERES IMPRIMABLES :

26 lettres majuscules (Q,W,E,R...)

26 lettres minuscules (q,w,e,r...)

10 chiffres (1,2,3,4...)

30 caractères spéciaux (#, @, ponctuation...)

## b - LES CARACTERES DE CONTROLE

Ils génèrent une fonction spéciale et sont accessibles par les touches **CTRL** (32 fonctions), **ESC** (31 fonctions) et **FUNCT** (fonctions de base prédéfinies et fonctions définissables par l'utilisateur). Les tables en annexes donnent toutes les applications possibles à partir du clavier..

## 3 - CONFIGURATIONS

L'**HYPER-BASIC** du **TELESTRAT** permet de "configurer le clavier à volonté en fonction des goûts de l'utilisateur, ou des applications. Cela est rendu possible par 3 instructions **BASIC**, utilisées en mode direct ou en mode programme, et qui sont **ACCENT**, **AZERTY** et **FRENCH**.

Un coup d'oeil rapide à l'annexe 2 vous donnera déjà une idée de ce que sont les différentes configurations :

**ACCENT** va permettre d'obtenir avec l'**ORIC** les caractères accentués français (e accentués, a accentué, u accentué, c cédille etc..). Ces caractères possèdent des codes **ASCII** qui les rendent compatibles avec les imprimantes possédant une option "caractères français". La plupart de ces "nouveaux" caractères sont obtenus avec la touche **SHIFT**, le reste des touches n'est pas affecté.

**AZERTY**, comme son nom l'indique, va configurer le clavier "en **AZERTY**", c'est à dire que, en plus des touches affectées par la fonction **ACCENT**, les touches **A**, **Z**, **Q**, **W** et **M** changeront de place pour prendre celle qui leur est habituelle sur un clavier français. Par contre, nous avons maintenu les chiffres "en bas", c'est à dire sans **SHIFT**. En effet, les chiffres sont très utilisés en programmation, et le fait d'avoir à appuyer sur 2 touches à la fois (**SHIFT** + chiffre) chaque fois que l'on voudrait obtenir un chiffre, ralentirait considérablement la saisie. Le mode **AZERTY** est compatible avec les imprimantes.

Pour les inconditionnels du clavier français, c'est à dire clavier **AZERTY** accentué avec signes de ponctuation au mêmes endroits que sur une machine à écrire, chiffres et majuscules "en haut", l'instruction **FRENCH** leur donnera le clavier de leurs rêves. Une restriction cependant : les caractères affichés ne seront pas obtenus tels quels à l'imprimante, pour une raison d'incompatibilité des codes **ASCII**.

Pour les 3 configurations, l'accent circonflexe n'est pas disponible seul, le générateur d'écran n'étant pas capable de gérer 2 caractères indépendants sur la même "case" d'écran. On disposera donc de la plupart des voyelles déjà accentuées (ê, â etc..)

2 jeux de petits autocollants permettent de modifier le clavier à volonté si l'on utilise une configuration de façon permanente. Les petits caractères dans les cases blanches rappellent le caractère d'origine, avant la modification du clavier.

## 4 - LES TOUCHES DE FONCTIONS

Nous l'avons vu plus haut, les touches de fonctions permettent, en appuyant sur **FUNCT** et une touche du clavier d'obtenir l'exécution d'une commande BASIC sans avoir à taper celle-ci en entier. Ainsi, appuyer sur **FUNCT** et \ équivaut à taper **LIST** puis appuyer sur **RETURN** (faites l'essai !)

Ainsi, lors de l'allumage du TELESTRAT, le clavier possède 2 fois 46 touches pré-programmées : chaque touche utilisée avec **FUNCT** correspondra à une fonction. Utilisée avec **SHIFT** et **FUNCT**, elle correspondra à une 2° fonction. L'affectation de ces touches a été faite selon 2 critères : rapidité d'utilisation, ou facilité de mémorisation. Ainsi, **LIST** qui est une commande très utilisée en BASIC pour la mise au point d'un programme sera très facile à taper avec le pouce et l'index droit. **RUN** s'obtiendra de même avec **FUNCT** et = . Pour s'en souvenir plus facilement, **FUNCT** et **G** donneront **GOSUB**, **FUNCT** et **L** donneront **LEFT\$**, **FUNCT** , **SHIFT** et **L** donneront **RIGHT \$** etc..

Le tableau complet des fonctions pré-programmées est en annexe 2.

Toutefois, ce tableau n'est pas figé, et l'utilisateur peut définir à volonté des touches de fonctions, affectant une ou plusieurs fonctions à une seule touche.

Il dispose pour cela des fonctions **KEYDEF** et **KEYUSE** .

**KEYDEF** permet d'affecter à une touche du clavier un **code de fonction**, compris entre Ø et 255 :

- Codes de Ø à 15 pour les fonctions définissables par l'utilisateur.
- Codes de 16 à 31 pour quelques mots-clés spéciaux (voir en annexe 4)
- Codes de 32 à 246 pour les mots-clés du STRATSED et de l'HYPER-BASIC (voir en annexe 4)
- Code 254 pour **FUNCT** + **DEL** (effacer la mémoire tampon).
- Code 255 pour **FUNCT** + **RETURN** (numérotation automatique des lignes de programme).

Pour affecter à la touche **E** la fonction **END**, taper donc :

**KEYDEF #80** puis la touche **E** (80 est en hexadécimal le code de fonction du mot-clé BASIC **END**) et enfin **RETURN**.

Nous remarquerons en début de la liste 16 codes "vierges" destinés à l'utilisateur, qui peut à sa guise affecter à ce code une **ou plusieurs** instructions BASIC pour que, à chaque pression d'une seule touche, une séquence complète d'instructions s'affiche ou s'exécute (si l'on a inclus un **RETURN** dans la suite d'instructions). Ce sera le rôle de la fonction **KEYUSE**.

Par exemple :

**KEYUSE 1 , CHR\$(12) + "PAPER Ø:INK3 + CHR\$(13)**

**CHR\$(12)** correspond à l'effacement de l'écran, et **CHR\$(13)** à **RETURN**.

Si l'on tape ensuite **KEYUSE 1** puis **C** par exemple, il suffira de taper **FUNCT** + **C** pour que l'écran s'efface puis passe en écriture jaune sur fond noir !

La chaîne de définition de l'instruction peut comporter 128 caractères alphanumériques, pourvu que leur code ASCII soit compris entre 1 et 126.

### III - LES CARTOUCHES ENFICHABLES

Le Telestrat possède 2 ports (prises) réservés aux cartouches développées pour l'appareil.

Le langage de programmation BASIC réside dans une cartouche appelée **HYPER-BASIC**. Cette particularité fait du Telestrat une des machines les plus souples du marché. La ROM (Read Only Memory ou mémoire morte) de l'ordinateur est amovible !

Contrairement à d'autres appareils, il est facile de changer le langage de programmation du Telestrat (le langage C et le FORTH sont en voie de réalisation).

#### 1 - INSTALLATION

Le Telestrat doit être débranché pendant la fixation des cartouches dans leurs logements. Bien qu'elles soient protégées contre les fausses-manoeuvres et les faux-contacts d'alimentation, il est bon de prendre pour principe de **toujours éteindre l'appareil avant de brancher ou débrancher quoi que ce soit sur le TELESTRAT.**

Les boîtiers des cartouches sont assymétriques pour éviter tout risque d'erreur pendant la pose.

#### 2 - LE PORT LANGAGE

L'emplacement de gauche, marqué LANGAGE, est réservé à la cartouche contenant la ROM du Telestrat. Cette cartouche peut contenir jusqu'à 48 Ko.

La cartouche HYPER-BASIC est livrée avec l'appareil. Elle devient utilisable dès la lecture d'une disquette qui contient le système d'exploitation STRATSED.

La cartouche ROM contient :

- Le **moniteur**, logiciel système de 8 Ko assurant l'initialisation et le chargement du STRATSED.
- Le **BASIC** occupant environ 24 Ko.

#### 3 - LE PORT UTILISATEUR

L'emplacement de droite, marqué UTILISATEUR, est conçu pour une cartouche d'application. TELEMATIC est livrée avec l'appareil et apporte les applications "serveur monovoie " et communication via un terminal Minitel (voir le manuel APPLICATIONS MINITEL). Deux sortes de cartouches peuvent être enfichées :

- RAM (Random Access Memory), extension de la mémoire utilisateur de 64Ko.

• ROM (Read Only Memory), applications pré-enregistrées qui peuvent être :

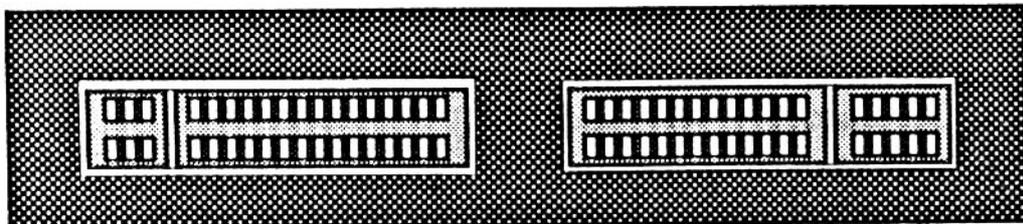
- Logiciels (utilitaires, productivité personnelle, jeux...)
- Interfaces supplémentaires.
- Langages de programmation.
- Gestion des périphériques (cartouche contenant les appli-

cations musicales MIDI).

Voici un schéma des ports pour cartouches, vus de face. L'asymétrie des connecteurs permet de ne pas inverser les cartouches.

PORT LANGAGE  
ROM (HYPER-BASIC)

PORT UTILISATEUR  
ROM ou RAM (TELEMATIC)



## IV - PRINCIPE DE FONCTIONNEMENT

Vous trouverez dans ce chapitre la description du système Telestrat, celui-ci pouvant évoluer au fur et à mesure de vos besoins (lecteurs de disquettes supplémentaires, mémoire additionnelle...).

### 1 - LE MATERIEL

Les composants les plus importants du système sont :

- L'unité centrale composée d'une mémoire vive, d'une mémoire morte et d'un microprocesseur. La mémoire vive est utilisable par l'utilisateur pour stocker ses données et ses programmes. La mémoire morte contient surtout le langage BASIC. Celle-ci est logée en cartouche (port LANGAGE).
- Des circuits d'interfaces (liaison) avec des périphériques.
- Des périphériques qui servent aux **entrées et aux sorties** d'information (lecteur de disquettes, Minitel, écran, imprimante...).
- Des logiciels (suite d'instruction) qui peuvent être résidants (inclus dans la mémoire morte comme ceux qui gèrent la traduction de l'HYPER-BASIC).

#### a - LA MEMOIRE VIVE (RAM ou "Random Access Memory")

Cette zone permet à la fois la lecture et l'écriture de données. On l'appelle aussi "mémoire utilisateur". Elle peut être représentée par des cases vides qui peuvent contenir des nombres jusqu'à une valeur décimale de 255 (dans les calculs arithmétiques, un nombre plus grand est stocké sur plusieurs cases, il fallait y penser!). Ces cases sont identifiées par des adresses notées généralement sous forme de nombres hexadécimaux (#4000, #C000...). Elles servent à stocker des parties de données, des instructions, des mots clé du BASIC ou d'autres adresses. c'est là que sont vos programmes, ce que vous tapez au clavier, ce que l'écran affiche... et tout ce que le microprocesseur est en train de traiter.

Le traducteur implanté en ROM décide de la signification d'un nombre placé dans une case selon son contexte et sa place en mémoire (autre adresse, donnée servant à un calcul arithmétique...).

Le Telestrat à une configuration standard de 64 Ko. Un Kilo-Octets représente 1000 octets et un octet est un mot de 8 chiffres binaires ou bits (0 ou 1). Le Telestrat,

comme tout ordinateur, ne peut enregistrer dans ses circuits que des instructions de base de la forme 0 ou 1, c'est-à-dire absence ou présence de courant. Le microprocesseur M 6502 A, qui est l'unité de calcul de l'appareil, travaille avec des commandes et des données formées de ces mots de huit bits, circulant par paquets dans les circuits de l'unité centrale.

### Définition d'un octet

Les bits qui constituent un octet sont numérotés de 0 à 7, chaque bit représentant une valeur de deux fois celle du bit placé à sa droite. Le bit N° 0 représente 0 ou 1, le bit N°1 représente 0 ou 2, le bit N° 2 représente 0 ou 4...La valeur que prend l'octet est la somme des nombres représentés par ces bits. Cette valeur peut donc varier de 0 à 255 en système décimal.

N° du Bit	7	6	5	4	3	2	1	0
Etat max.	1	1	1	1	1	1	1	1
Val.( $\Sigma = 255$ )	128	64	32	16	8	4	2	1

Le nombre binaire 01101101 est donc équivalent à 109 en système décimal :

N° du Bit	7	6	5	4	3	2	1	0
Etat	0	1	1	0	1	1	0	1
Val.( $\Sigma = 109$ )	0	64	32	0	8	4	0	1

### L'espace mémoire (voir schéma en annexe)

En raison du mode de fonctionnement du 6502, la RAM de base (64 Ko) peut être représentée comme une suite de 4 banques mémoire contenant chacune des centaines de cases vides.

- La première banque (adresse hexadécimale de 0000 à 4000) contient aussi des renseignements sur les tâches en cours : de 0000 à 0800, adresses de début et de fin de programme BASIC, pointeurs indiquant où sont certaines variables...
- La deuxième banque libre va de l'adresse 4000 à 8000.

- La troisième banque mémoire (8000 à C000) contient aussi les données affichées à l'écran (texte ou haute résolution...).
- La quatrième banque est remplie en partie avec le DOS (en un mot le SED!) qui se charge automatiquement à la mise en marche. Le reste de cette zone contient les différents buffers : de C000 à D000, les mémoires tampons servent à libérer le Telestrat pendant qu'il imprime (2 Ko) ou qu'il envoie des informations sur d'autres interfaces. Vous pouvez ainsi continuer à travailler. Si vous voulez déplacer ou agrandir les "buffers" dans la RAM, vous pouvez utiliser les instructions LBUF et SBUF qui sont abordées en référence.

## **b - LE MICROPROCESSEUR 6502A**

C'est l'unité de calcul et de traitement. Son fonctionnement est rythmé par une horloge qui synchronise les cycles des opérations pour éviter les carambolages de données! Cette horloge fonctionne au rythme de 1 Mz (1 million de cycles par seconde). Chaque cycle correspond au déroulement d'une phase de l'opération exécutée. Le nombre de cycles nécessaires à une opération varie de 2 à 7, selon sa complexité.

Le 6502 A contient lui aussi des emplacements mémoire dans lequel il range les données nécessaires à l'exécution de l'instruction en cours.

### **Les registres**

Le 6502 comporte dans sa "puce" 6 registres dont un de 16 bits : le compteur d'instruction. Les registres ont des tâches particulières mais peuvent aussi servir de mémoires temporaires ou de compteurs.

#### **• L'accumulateur (registre A)**

Il sert à accumuler les résultats des opérations arithmétiques, les données sont chargées depuis la mémoire vive utilisateur puis remise une fois le calcul terminé.

#### **• Les registres d'index (X et Y)**

Ils sont utilisés pour la modification des adresses. Certaines instructions modifient leur contenu.

#### **• Le pointeur de pile (S)**

Il sert à repérer le prochain emplacement disponible de la pile. Celle-ci est une zone

de mémoire en liste qui sert aux appels et aux sorties de sous-programmes.

- **Le registre d'état (P)**

Il est composé de huit indicateurs binaires (drapeaux prenant l'état 0 ou 1, "flags" en anglais). Ils indiquent l'état du microprocesseur une fois qu'une instruction a été exécutée (résultat négatif ou positif dans une opération arithmétique...).

- **Le compteur d'instruction des programmes (PC)**

Il contient l'adresse de l'instruction qui suit celle qui est en cours d'exécution. Il peut être chargé dans la pile pour y être examiné. Le microprocesseur sait ainsi où appeler l'instruction qui va devoir être exécutée une fois sa tâche immédiate terminée.

### c - LA MEMOIRE MORTE (ROM)

Elle ne peut pas être transformée mais seulement lue. Cette zone de 48 Ko de mémoire contient des instructions et données en langage machine. Etant en cartouche, elle peut être adaptée aux évolutions de nos produits. Elle contient notre BASIC et des instructions de base qui servent au démarrage de l'appareil chaque fois qu'il est mis en route, c'est ce qu'on appelle le **moniteur**. Ces instructions assurent en particulier le chargement du DOS dans la mémoire vive du Telestrat (RAM).

### d - LES INTERFACES

Elles représentent une des originalités du Telestrat : **presque tout ce qui touche la micro-informatique peut-être relié à l'ORIC.**

- Les deux ports pour cartouches enfichables : permettent de connecter les cartouches ROM ou RAM développées pour le Telestrat (voir Chap. III).
- La prise DIN pour stocker des données sur un magnétophone
- La prise pour moniteur couleur RVB/PERITEL
- Le noyau MIDI (Musical Instrument Digital Interface) : cette prise permet de connecter le boîtier d'adaptation MIDI, vendu avec une cartouche d'applications. Le standard MIDI est utilisé par toutes les machines de musique électronique équipées

d'une entrée/sortie pour adapter des périphériques. Cela va du synthétiseur de jeu au synthétiseur polyphonique de concert, en passant par des boîtes à rythme et autres séquenceurs.

- La sortie CENTRONICS parallèle
- Le connecteur du BUS complet : permet la commande de circuits électriques ou électroniques extérieurs.
- La liaison avec les lecteurs de disques
- Le connecteur Minitel (voir le Manuel des Applications Télématique)
- La prise RS 232
- Les deux prises Joystick : elles fonctionnent selon le standard ATARI® qui est devenu la norme pour les manches à balais (manettes de jeu ou "joysticks" en anglais). Les manettes possèdent des boutons appelés "boutons de tir", du fait de l'utilisation première des manettes dans des jeux d'arcade qui mettaient en scène des vaisseaux spatiaux équipés de canons LASER. Ces boutons peuvent être utilisés comme une touche spéciale, tout en déplaçant le curseur... En parlant de LASER, si vous voulez savoir quel bruit fait un canon LASER, tapez donc ZAP puis la touche RETURN en BASIC... Pour revenir à des outils plus pacifiques, les manches à balai servent surtout à déplacer le curseur de l'écran d'une manière rapide.

La prise de droite est conçue pour recevoir une souris. Ce charmant animal est un des moyens les plus intéressants pour déplacer le curseur : il permet un déplacement rapide et néanmoins précis, bon compromis entre les touches fléchées et un manche à balai. La souris est un boîtier qui repose sur une boule qu'on fait rouler sur une table ou un bureau. Le déplacement de cette boule impose au curseur ses déplacements. Des boutons situés sur la souris permettent d'accéder à des fonctions spéciales.

## V - LE SYSTEME D'EXPLOITATION : STRASED

Le Système d'Exploitation de Disques ("Disk Operating System" ou DOS en anglais) STRASED gère le transfert des données entre les supports de disquettes et l'unité centrale.

Il comprend des instructions de type BASIC qui permettent d'utiliser les lecteurs, sauvegarder un programme (SAVE), le charger en mémoire centrale (LOAD), utiliser des fichiers (OPEN, CLOSE, FIELD...), faire des copies (COPY, BACKUP...).

Le STRASED est stocké sur tous les disques formatés par le Telestrat. Une disquette doit être lue au début de l'utilisation pour transférer les instructions du DOS dans la mémoire de travail de l'appareil. Le transfert se fait en 2 secondes environ. Toutes les disquettes utilisées par le Telestrat contiennent donc STRASED. Cela ne fait pas perdre plus de 10 Ko sur une disquette 3" double face qui peut contenir 400 Ko formatés. Le système est copié sur chaque nouvelle disquette dès l'utilisation de la commande INIT (initialisation). Le système d'exploitation du Telestrat peut gérer jusqu'à un peu plus de 1 Mo par lecteur de disques (1 million d'octets) et 4 lecteurs de disquettes. Vous pouvez d'ailleurs brancher n'importe quel lecteur de disquettes sur le Telestrat. Le STRASED ne tient pas compte du format physique de la disquette, seulement de son "découpage" en pistes et en secteurs magnétiques, tâche qu'il assure lui-même au moment de l'initialisation.

Par défaut, le système formate les disquettes en 42 pistes par face et 17 secteurs de 256 octets par piste. Ces paramètres correspondent à la norme dite "IBM" ® et assure un bon compromis entre la vitesse d'écriture des données, la quantité d'informations stockées et la sécurité de la sauvegarde. En mode standard, une disquette formatée sur MICRODISC contiendra donc un peu plus de 365 Ko utiles, le lecteur étant double face.

L'utilisation des disquettes suppose que celles-ci sont formatées : quand vous insérez dans le lecteur une disquette vierge, elle est illisible pour le Telestrat. Le disque magnétique doit d'abord être divisé en pistes et en secteurs qui formeront les différentes zones où sont stockées les données. C'est ce que fait le Telestrat quand vous tapez la commande INIT(initialisation). Une disquette initialisée par le Telestrat n'est donc pas compatible avec une disquette 3 " formatée par un autre appareil. A l'enregistrement d'un fichier (ensemble de données identifié par un nom

générique : PERSONNEL, TARIF...), le système d'exploitation divise les informations en blocs, pour les adapter à chaque secteur. Certains secteurs sont réservés à la table des matières qui contient les adresses et les caractéristiques de tous les fichiers stockés sur le disque. La commande **DIR** permet de lister à l'écran un répertoire du disque. Des instructions permettent de modifier, à l'initialisation, le nombre des pistes et des secteurs d'une disquette, dans la limite des chiffres donnés ci-dessus. Il faut savoir que la fiabilité et la vitesse des enregistrements diminuent quand on augmente la capacité de stockage. Par contre, la vitesse de lecture ne change pas. Si des valeurs ne sont pas spécifiées, ces paramètres sont donc par défaut :

- 42 pistes par face de disquette,
- 17 secteurs par piste.

Ces valeurs standard correspondent à une capacité de stockage de 365 ko par disquette. Elles permettent de lire et d'écrire dans un fichier à la vitesse de 20 Ko par seconde.

Le lecteur ORIC étant double face (une tête de lecture de chaque côté du disque), vous n'avez pas besoin de retourner la disquette pour lire la face B après avoir lu la face A. Le STRATSED traite les deux faces comme une suite de données, vous n'avez pas à préciser où stocker les informations.

- La disquette contient des pistes concentriques qui ressemblent à un disque micro-sillon de 33 tours (sur un tel disque, en fait, il n'y a qu'une seule piste enroulée du bord vers le centre).
- Une piste est divisée en secteurs.
- Un secteur contient 256 octets, mots de 8 bits constitués par les données compréhensibles par l'ordinateur (0 et 1).

## VI - LE LANGAGE DE BASE : HYPER-BASIC

### 1 - CARACTERISTIQUES GENERALES

Vous trouverez une description détaillée du BASIC du Telestrat dans le manuel de référence.

Le BASIC est un langage évolué : il ressemble à un langage humain, avec son vocabulaire et sa syntaxe. Il a été conçu en 1964 aux Etats-Unis et se rapproche de l'anglais (il existe une version française assez peu répandue). Le BASIC - "Beginners All Purpose Symbolic Instruction Code" - est le langage de programmation le plus utilisé au monde du fait de sa facilité d'apprentissage.

Certains considèrent même qu'il est **trop simple** et donne de mauvaises habitudes de programmation. Ceci est de moins en moins vrai puisque le BASIC continue à évoluer, comme le prouve HYPER-BASIC. Cette nouvelle version allie les qualités d'un langage structuré comme le PASCAL - autre langage répandu - à une souplesse indéniable :

- Le BASIC se **structure** : on peut utiliser des sous-programmes et éviter les renvois inorganisés (dans un programme de plusieurs centaines de lignes, écrit depuis longtemps, il est parfois difficile de comprendre à quoi sert un GOTO 125 - "Aller à la ligne 125" - surtout si à la ligne 125 on trouve un GOTO 1247, et ainsi de suite !).
- On peut créer des procédures récursives (des sous-programmes qui s'appellent eux-mêmes).
- HYPER-BASIC est à la fois compilé et interprété, ce qui accélère le temps d'exécution des programmes de 2 à 200 fois par rapport à un BASIC interprété.
- Les entrées/sorties vers les périphériques peuvent être gérées depuis un programme BASIC, 4 fenêtres d'écran, le clavier, la souris, les lecteurs de disques, l'imprimante... sont autant de périphériques.
- Les entrées/sorties sont "bufférisées" (mot bizarre et franglais venant de "buffer", un mot français équivalent serait "tamponnées", vous comprendrez sans doute le mal que nous avons à l'utiliser, merci !). L'appel à ces procédures n'arrête pas le fonctionnement de l'unité centrale. Le tampon de mémoire attribué à l'imprimante est de 2 Ko, celui de la sortie RS 232 aussi, le clavier possède quant à lui une mémoire tampon de 32 octets...
- Les possibilités graphiques sont étendues par des commandes donnant accès à des

simulations des trois dimensions (TABOX, TDRAW...).

## a - QUELQUES DEFINITIONS

Une **commande** est un ordre donné à la machine et exécuté en mode direct (effet immédiat) :

```
PRINT ESSAI$
```

affiche le contenu de la variable ESSAI\$ à l'écran. Si cette variable n'a pas encore été définie, elle est vide et son contenu sera 0.

Une **instruction** est une commande ou un ensemble de commandes inscrites dans le listing d'un programme (exécution en mode programme), certaines commandes ne sont pas programmables (LIST, DIR...) :

```
20 ESSAI$ = "MAKE MY DAY"  
30 PRINT ESSAI$
```

Ces instructions ne seront exécutées qu'après la commande RUN. Elles font partie d'un programme, comme l'indique les numéros de ligne. En ligne 20, une variable ESSAI\$ est créée et une chaîne de caractères, définie par des guillemets, est placée dedans. En ligne 30, le contenu de ESSAI\$ est affiché à l'écran.

Une **fonction** effectue des manipulations ou des calculs lorsque vous lui fournissez des données. Elle s'utilise généralement avec une commande :

Par exemple SQR ( ) est la fonction mathématique racine carrée ( $\sqrt{x}$ ), INT ( ) renvoie la partie entière d'un nombre, SIN ( ) calcule le sinus d'un angle...on peut écrire :

```
PRINT SQR(25)
```

ou

```
A=SQR(25)  
PRINT A
```

les deux commandes affichant 5. Ces fonctions sont des fonctions **pré-définies** ou intrinsèques. Il existe aussi des fonctions définies par l'utilisateur pour des expressions de calcul fréquemment utilisées.

Pour cette définition, on utilise l'expression DEF FN V(x). Par exemple :

```
10 DEF FN M(R)= 3.1416*R^2 (c-à-d  $\pi.r^2$ )
```

```
20 INPUT A
30 PRINT FN M(A)
40 GOTO 20
```

Ce programme assigne le calcul de la surface d'un cercle à la fonction  $M(R)$ , le rayon étant représenté par  $R$  (ligne 10).

En ligne 20, le déroulement s'arrête dans l'attente d'une donnée tapée par l'utilisateur au clavier, et stockée dans la variable  $A$ .

En 30, le programme affiche le résultat de l'opération définie par la fonction  $M(A)$ .

En 40, le pointeur de programme retourne à la ligne 20 pour demander une autre constante.

## 2 - LA TRADUCTION D'UN LANGAGE

Le BASIC disponible sur le Telestrat est un langage dit évolué. Nous avons vu que le processeur 6502A ne reconnaît que les instructions codées sous forme de chiffres qu'il place dans des cases mémoire. Le Telestrat traduit les langages évolués en assembleur, qui est un langage propre à chaque microprocesseur, intermédiaire entre le code machine (0 et 1) et le langage.

Il y a deux principales sortes de traducteur en informatique:

- Celui qui traduit d'une phrase à une autre, l'interpréteur. Dans ce cas, les commandes du programme sont stockées en mémoire en clair et retraduites à chaque fois qu'il est utilisé. L'exécution d'un ordre nécessite 2 opérations : la recherche et la reconnaissance de l'ordre (interprétation), l'exécution du sous-programme assembleur. Ce procédé entraîne une certaine lenteur d'exécution.
- Celui qui traduit le programme d'un seul coup, une fois qu'il est tapé, le compilateur. Ce programme est généralement appelé quand l'utilisateur a fini d'écrire le programme source, en langage évolué. Le compilateur peut donc être stocké sur une disquette et appelé quand l'utilisateur le demande (ou automatiquement, après la commande RUN, sur le Telestrat). Ce principe impose des accès au texte source, au programme objet, à l'éditeur de texte et au compilateur. Il faut donc disposer d'une mémoire de masse. Le programme traduit est stocké sur disque sous une forme codée utilisable par la machine (programme objet) et sous une forme lisible par l'utilisateur, en langage évolué (programme source). Deux versions du même programme cohabitent en général sur le disque. L'assembleur lui-même est un langage compilé, c'est même le plus puissant puisqu'il reflète directement le programme source.

Le système de traduction du Telestrat est unique puisqu'il utilise les avantages des deux traducteurs : "transparence" de manipulation du programme interprété, codage et vitesse d'exécution du programme compilé.

### 3 - GENERALITES DE PROGRAMMATION

#### **Variables et données**

Les données sont les nombres et les caractères utilisés, les variables sont les endroits où vous les stockez. Ces dernières peuvent donc prendre différentes valeurs selon les données (constantes) qu'elles contiennent.

#### **a - LES DONNEES NUMERIQUES ET CHAINES DE CARACTERES**

- **Les données numériques**

Elles peuvent être entières (135) ou réelles (3,14), elles doivent être comprises entre 2,93874E-39 et 1,7014E+38 en valeur absolue (positif ou négatif). Au fait, la notation en puissance de 10 permet de faire tenir de très grands nombres dans peu de place (notation exponentielle) : 1 000 000 000 peut être noté 1E+9, c'est-à-dire  $1 \cdot 10^9$  (1 multiplié par 10 à la puissance 9).

Le Telestrat affiche automatiquement en puissance de 10 tout nombre qui dépasse l'intervalle [999 999 999 à 0,01].

- **Les données alphabétiques : chaînes de caractères**

Ces données peuvent être constituées de lettres, de mots et de chiffres qui seront alors pris comme des caractères, tous les signes de ponctuation sont admis. Pour marquer une donnée alphabétique, il suffit de la mettre entre guillemets.

"COMMENT ALLEZ-VOUS No 8 ?" peut être stockée dans une variable. Celle-ci doit être signalée par le signe "\$". Mettons notre phrase dans une variable appelée ESSAI\$, tapez :

**ESSAI\$ = "COMMENT ALLEZ-VOUS No 8 ?" (confirmé par RETURN)**

Pour vérifier le contenu de ESSAI\$, tapez : **PRINT ESSAI\$**

L'écran affiche le contenu de la variable :

**COMMENT ALLEZ-VOUS No 8 ?**

#### **b - LES NOMS DE VARIABLE**

Les noms peuvent comporter jusqu'à 16 caractères (lettres ou chiffres). Les espaces

sont pris comme des caractères. Les minuscules sont transformées en majuscules par le Telestrat (**SOMME** est la même variable que **somme**) et les signes de ponctuation ne sont pas valables.

Les nombres entiers de 32767 à - 32768 (c-à-d :  $-2^{15}$ ) peuvent être stockés dans des variables que vous identifiez par le signe "%". Ces variables occupent moins de place en mémoire que les variables réelles (2 octets au lieu de 5). Par exemple : **A3%**, **COMPTE 18%** et **TVA%** sont des noms autorisés de variables entières. Un nombre entier supérieur à  $[2^{15}]$  doit être stocké comme un nombre réel.

**A3**, **SOMME** et **TVA** sont des noms de variables réelles.

**Attention** : Les variables **A3** et **A3%** sont des variables différentes pour l'ordinateur. Elles peuvent contenir des données différentes.

**EXEMPLE** : Placez 6,789 dans la variable **EXEMPLE 2** :

Tapez :           **EXEMPLE 2 = 6,789**  
Tapez :           **PRINT EXEMPLE 2**  
Le Telestrat répond par : **6,789**  
Tout est normal.

Essayez :       **EXEMPLE 2% = 6,789**  
Tapez :           **PRINT EXEMPLE 2%**  
Le Telestrat répond par : **6**

Il n'a pas voulu d'un nombre réel dans une variable entière, le nombre 6,789 a donc été tronqué et seule sa partie entière a été stockée dans la variable.

De même :

Tapez :           **EXEMPLE 2% = 6**  
Puis :            **EXEMPLE 2 = 6**

Cette commande est logique pour le Telestrat car "qui peut le plus peut le moins" ! Une données numérique entière peut être stockée sous forme réelle (6 , c'est aussi 6,0).

#### 4 - LES BOUCLES ET BRANCIEMENTS DE PROGRAMME

Même si l'exécution d'un programme se fait en suivant des numéros de lignes ordonnés, des instructions permettent de "sauter" des lignes, à faire des

branchements. Il existe des **branchements inconditionnels** (GOTO, GOSUB, RETURN) qui placent le pointeur à un autre endroit du programme. Il y a aussi des **branchements conditionnels** qui ne se font que quand une ou plusieurs conditions sont respectées (IF-THEN, IF-THEN-ELSE, IF-GOTO, REPEAT-UNTIL). Les opérateurs logiques permettent d'introduire plusieurs conditions (AND, OR, NOT). Par exemple :

```
5 INPUT "Donnez un chiffre entier entre 1 et 10"; A
7 IF A=2 GOTO 21
10 IF INT(A) <> A OR A < 1 OR A > 10 THEN 17
13 PRINT "Perdu, recommencez SVP"
15 GOTO 5
17 PRINT "Lisez bien les instructions"
19 GOTO 5
21 PRINT "BRAVO, VOUS AVEZ GAGNE"
```

Ce programme diabolique demande un chiffre entier entre 1 et 10, vous félicite si vous tapez 2, vous rappelle à l'ordre sinon. Avouez que c'est passionnant, tant que le joueur ne sait pas ce qu'il faut faire...

Dans un programme, il est souvent utile de faire répéter certaines opérations jusqu'à obtenir un état précis d'une variable, testée à chaque changement. Ceci peut être fait d'une façon simple avec des boucles en utilisant des instructions FOR-NEXT ou REPEAT-UNTIL... Il s'agit en fait de branchements conditionnels. Voici un programme qui change la couleur d'écriture du texte (instruction INK n), émet un son d'explosion (instruction EXPLODE) et s'arrête après avoir affiché toutes les couleurs disponibles (y compris le noir qui correspond au N°0) :

```
10 FOR I=0 TO 7
20 INK I
30 EXPLODE
40 PRINT "EXPLOSION No ";I+1,
50 NEXT I
```

Ici, la variable est I. Après l'explosion, le programme vérifie en ligne 40 que la variable I ne dépasse pas la limite 7. Si la limite n'est pas atteinte, le pointeur de programme retourne à la ligne 10 où I est incrémentée (ajout de 1 unité). Le programme effectue de nouveau les instructions des lignes 20 et 30... Quand la valeur de I dépasse 7, le programme s'arrête.

## 5 - LES LABELS ET LES PROCEDURES

### a - DEFINITION

Un label est un nom qui identifie une procédure ou un sous programme servant à une tâche répétitive. La commande `J` permet de définir ces labels. Ceux-ci remplacent un numéro de ligne auquel on peut accéder avec **GOTO**, **GOSUB**, **THEN** et **ELSE**.

Un label a 16 caractères significatifs, le premier caractère devant être une lettre. Dans un listing, les labels sont repérés par un espace après le numéro de ligne, au lieu de trois espaces pour des instructions "normales". Vous n'avez pas besoin de taper ces espaces, le Telestrat les place automatiquement pour rendre les listings plus lisibles. Voici un exemple :

```
20 REM programme principal
30 JAJFFICHLAGIE
40 OUPS
50 JENDW
60 REM suite du programme principal...
```

Une procédure est définie entre la ligne 30 et la ligne 50. Elle produit le son défini par l'instruction **OUPS** (oui, c'est bien une instruction du BASIC du Telestrat!).

## b - LES 3 TYPES DE SAUT IDENTIFIES PAR DES LABELS

### • Le branchement

Dans ce cas, le label remplace un numéro de ligne dans une instruction de saut :

```
50 GOTO BRANCHE
60 REM
70 REM
.....

200 JBRANCHE
210 PRINT "JOLI SAUT"
220 END
```

Le pointeur de programme saute de la ligne 50 à la ligne 200, il reprend l'exécution à cet endroit.

### • Le sous-programme

Un **S** doit être placé après le label pour indiquer un sous-programme, dans ce cas le **GOSUB** est facultatif pour y accéder. Dans le cours du programme principal, les sous-programmes sont ignorés, ils ne sont accessibles que par un appel spécifique :

```

150 ROUTINE
160 PING
170 EXPLODE
180 JROUTINE, S
190 OUPS
200 RETURN
210 ZAP

```

....

De la ligne 180 à 200, un sous-programme est défini. On y accède à la ligne 150, en appelant son nom (on aurait pu écrire en ligne 150 `GOSUB ROUTINE`). Une fois le sous-programme déroulé, en 200, une instruction `RETURN` renvoie au point de départ, en 160. Le programme principal reprend et quand le pointeur arrive en 180, il saute en 210, pour continuer son cours. Le sous-programme est ignoré dans le déroulement normal du programme, on ne peut y accéder que par un branchement.

### • La procédure

Une procédure ressemble à un sous-programme, mais :

- Sa définition commence par un label suivi de `W` et se termine par `ENDW`.
- La définition est récursive (elle peut s'appeler elle-même et "boucler").
- Il est possible de créer des paramètres (variables locales).

Nous aurions pu écrire une procédure comme sous-programme dans le cas précédent :

```

180 JROUTINE, W
...
200 ENDW

```

Les paramètres, chaînes ou nombres placés dans une variable, suivent l'appel de la procédure. Ils sont rappelés dans celle-ci par le signe `$` suivi d'un numéro d'ordre, plusieurs paramètres pouvant être utilisés. Voici en exemple la procédure servant à calculer une factorielle (en mathématique la factorielle de 5, notée `5!` est 120 car  $5! = 5 \times 4 \times 3 \times 2 \times 1$ ). Cet exemple utilise aussi des branchements simples (en 30, en 40 et en 80) :

```

10 JDEBUT
20 INPUT "Donnez un nombre entier positif"; N
30 IF INT(N) <> N OR N < 1 THEN DEBUT
35 RESULTAT=1
40 IF N=1 THEN FIN
50 FACTORIELLE, N

```

```

60 IFIN
70 PRINT RESULTAT
80 GOTO DEBUT
90 IFACTORIELLE, W
100 RESULTAT=RESULTAT*$1
110 IF $1<>1 THEN FACTORIELLE, $1-1
120 ENDW
...

```

Le programme affiche le résultat du calcul de factorielle.

En 20, la routine DEBUT demande une constante à l'utilisateur.

En 30, si le nombre n'est pas entier positif au moins égal à 1, le pointeur retourne à DEBUT.

En 35, la variable qui contiendra le résultat est mise à 1.

En 40, le cas particulier 1 est traité ( $1!=1$ ).

En 50, la constante est prise pour être utilisé dans FACTORIELLE, en 90.

En 90, la factorielle est calculée par récurrence (la constante N est dans le paramètre \$1, puis N-1, N-2 et ainsi de suite jusqu'à la valeur 1).

En 120, quand le paramètre est à 1, le pointeur retourne d'où il était parti, en 60.

En 70, le résultat est affiché.

En 80, le pointeur retourne à la case départ, et c'est reparti...

La limitation concernant l'usage des paramètres \$n est qu'ils ne peuvent être utilisés que localement, à l'intérieur d'une procédure, ils sont cantonnés dans leur niveau de sous-programme (le \$1 d'une procédure A et le \$1 d'une procédure B faisant appel à A ne sont pas les mêmes, ils contiennent des constantes différentes, stockées dans des cases mémoire différentes) :

```

5 EXEMPLE, 1000
10 IFEXEMPLE, W
20 PRINT $1
30 ROUTINE, 12
40 PRINT, $1
50 ENDW
53 STOP
55 REM un peu d'air...
57 REM
60 IFROUTINE, W
70 PRINT $1
80 ENDW
...

```

Le programme affiche : 1000, 12, 1000

La ligne 5 appelle la procédure EXEMPLE avec la constante 1000.  
La ligne 20 imprime 1000 (stocké dans le paramètre \$1 de EXEMPLE).  
La ligne 30 fait appel à la procédure ROUTINE, en 60, avec la constante 12.  
En 70, 12 est imprimé (stocké dans le paramètre \$1 de ROUTINE).  
En 80, on retourne d'où on était parti : en 40.  
En 40, le programme affiche **1000**, qui est toujours dans le paramètre \$1 de EXEMPLE.

Si, dans une procédure, on fait appel à des paramètres non définis, ceux-ci seront considérés comme étant nuls : Dans le programme précédent,

**40 PRINT \$1, \$2**

fera imprimer au programme : 1000, 12, 0, 1000

## 6 - L'EDITEUR DE PROGRAMME

Nous allons décrire ici une autre caractéristique exceptionnelle de l'ORIC : l'éditeur plein écran. Cette facilité permet de changer une ligne de programme ou une commande erronée en tapant simplement les signes correctes, où que soit la faute. Le Telestrat insère automatiquement les nouveaux caractères entre les anciens. Pour enlever des parties d'instruction, il suffit d'utiliser la touche DEL.

### a - GENERALITES

Le Telestrat doit traduire chacune des commandes BASIC que vous tapez au clavier. Comme vous le savez, un ordinateur n'est pas intelligent : il ne reconnaît que les mots qui ne comportent aucune faute d'orthographe.

Quand vous confirmez la commande par RETURN, l'interpréteur traduit les mots en codes compréhensibles par le microprocesseur 6502. Si ces mots ne lui disent rien, la machine affiche un message d'erreur (voir annexe ?) et se place sur le premier caractère du mot inconnu. Voyons une séquence de correction :

Tapez : **10 PRINR "CECI ET UNE FOTE GRAVE"** puis la touche RETURN.

Vous avez raison : "cette phrase est complètement loufoque", comme la commande PRINR d'ailleurs...

Le message ?SYNTAX ERROR s'est affiché à l'écran.

Le Telestrat ne s'intéresse pas au contenu de votre texte, après tout, nous avons le droit de dire n'importe quoi ! Par contre, il a décelé la faute sur *PRINR*, qui a été tapé au lieu de **PRINT**.

Comme nous sommes des gens sérieux, nous allons corriger les trois fautes d'un coup. Oui, il y en avait trois, ceci dit pour les lecteurs distraits.

Au lieu d'être obligé de retaper toute la ligne, il suffit de déplacer le curseur avec les touches fléchées. Une fois arrivé sur la lettre à changer, tapez le caractère correcte :

Une fois sur le R de *PRINR*, tapez **T**.

Entre le E et le T de "ET", nous devons insérer un S.

Amenez ensuite le curseur sur le O de "FOTE" et tapez **A** puis **U**.

Recopiez enfin le reste de la ligne puis confirmez par RETURN, le Telestrat affiche maintenant : **CECI EST UNE FAUTE GRAVE**. Il a compris l'instruction **PRINT** qui commande l'affichage d'une donnée ou du contenu d'une variable. Les corrections ont bien été enregistrées.

**Remarque :** **PRINT A** et **PRINT "A"** n'ont pas le même sens, l'un affiche le contenu de la variable réelle A, l'autre affiche la lettre "A".

## b - AIDES A LA PROGRAMMATION

Une fois un programme tapé, il est parfois laborieux de changer un caractère ou une instruction fréquemment répétée. Certaines commandes permettent de changer une chaîne de caractères pré-définie, chaque fois qu'elle est trouvée dans le listing du programme édité :

### SEEK PRINT

recherche dans le programme en mémoire la chaîne de caractère "PRINT" qui correspond à une instruction BASIC bien connue. Le Telestrat affichera les numéros de toutes les lignes où se trouvent les *PRINT*.

### CHANGE ZAP TO EXPLODE

remplace toutes les instructions BASIC *ZAP* du programme par des instructions *EXPLODE* (ces instructions envoient des sons dans le haut parleur du moniteur).

Les détails d'utilisation de ces commandes se trouvent dans l'index du BASIC.

## VI - GRAPHISMES ET SONS.

### 1 - GENERALITES

Maintenant que nous sommes familiarisés avec les instructions simples du BASIC, nous pouvons nous tourner vers un des aspects les plus remarquables de la programmation des micro-ordinateurs : la création graphique.

Votre TELESTRAT possède d'excellentes possibilités sur le plan du graphisme, et la programmation de ces fonctions est assez simple. Il est possible de travailler suivant 4 MODES : le mode TEXT, le mode LORES 0, le mode LORES 1 et le mode HIRES. Nous allons les examiner un par un dans ce chapitre.

### 2 - LE MODE TEXT.

C'est le mode dans lequel se trouve votre ORIC quand vous l'allumez (avec la cartouche BASIC bien sur), et, comme son nom l'indique, il est principalement destiné à l'affichage de textes. Nous avons déjà rencontré l'instruction PRINT, qui permet d'afficher à l'écran des textes et des variables. Si vous tapez :

```
10 PRINT "BONJOUR"
```

L'écran affichera **BONJOUR** dès que vous taperez **RUN**.

La syntaxe de cette instruction est plus complète : si vous tapez

```
10 PRINT "BONJOUR"  
20 PRINT "MONSIEUR"
```

quand vous lancerez le programme, l'ordinateur affichera :

```
BONJOUR  
MONSIEUR
```

Après avoir exécuté la première instruction et affiché "BONJOUR", l'ordinateur ira à la ligne et affichera "MONSIEUR".

Une option à cette instruction permet de changer cet état de fait : Si l'on ajoute une virgule à la fin de la ligne 10, on obtient le résultat suivant :

```
10 PRINT "BONJOUR",  
20 PRINT "MONSIEUR"
```

Dès que l'on tape **RUN**, on a à l'écran :

```
BONJOUR MONSIEUR
```

La virgule a bien sur changé la position d'affichage de l'information suivante.

On observe que Le B de BONJOUR et le M de MONSIEUR sont séparés exactement par 8 caractères. Si la première chaîne fait plus de 8 caractères, la deuxième sera affichée en commençant à la position de tabulation suivante, soit 8 caractères plus loin ! Par exemple, avec

```
10 PRINT "EH, BONJOUR",
20 PRINT "MONSIEUR DU CORBEAU"
```

on obtient

```
EH, BONJOUR      MONSIEUR DU CORBEAU !
```

le M de MONSIEUR étant à la 16<sup>e</sup> colonne.

L'écran peut être vu comme ayant 5 positions automatiques de tabulation, espacées chacune de 8 caractères pour chaque ligne. Quand 2 instructions PRINT sont séparées par une virgule suivant le contenu de la première chaîne, la deuxième commence à la position de tabulation suivante. Cela peut être intéressant pour afficher des tableaux (colonnes de chiffres ou de mots).

Un deuxième signe peut servir à séparer des instructions PRINT, il s'agit du point-virgule.

Modifions la ligne 10 de la façon suivante :

```
10 PRINT "EH, BONJOUR";
```

et tapons RUN, nous obtiendrons :

```
EH, BONJOURMONSIEUR DU CORBEAU.
```

Le point-virgule implique que le curseur ne se déplace pas entre le fin de la première chaîne et le début de la deuxième. On peut donc "coller" 2 mots ou chaînes, mais aussi les séparer par un nombre déterminé d'espace et plaçant ces espaces à l'intérieur des guillemets. Il faut remarquer que, chaque fois qu'il affiche un nombre, l'ORIC affiche un espace avant le nombre et un espace après.

```
10 PRINT 1,2;3
20 PRINT "A","B";"C"
1      2 3
A      BC
```

Il y a une autre commande que nous pouvons utiliser pour modifier une instruction PRINT, il s'agit du suffixe TAB. Utilisée avec TAB, l'instruction PRINT se comporte comme la tabulation d'une machine à écrire : PRINT TAB(n) (chaîne ou variable) permet d'afficher une chaîne en positionnant le premier caractère de la chaîne à la n<sup>ième</sup> colonne.

Essayez le programme ci-après, qui devrait produire à l'affichage le résultat qui suit le listing :

```
10 FOR I = 0 TO 5
20 PRINT TAB(I); "BONJOUR"
30 NEXT I
```

BONJOUR  
BONJOUR  
BONJOUR  
  BONJOUR  
    BONJOUR  
      BONJOUR

Si l'on regarde le résultat, il semblerait que TAB(0) , TAB(1) et TAB(2) n'ont pas eu de résultat visible, mais c'est parce que les positions de tabulation 0 et 1 sont réservées à des fonctions spéciales que nous verrons plus bas. La fonction PRINT commence donc déjà à la 3<sup>e</sup> position de tabulation, comme si implicitement on avait un TAB(2) minimum ! Si l'on veut annuler cette "protection" des 2 premières colonnes, il suffit d'appuyer en même temps sur CTRL et J. Cette commande est reconnue par l'ORIC comme un inverseur interne qui bascule un état, en l'occurrence la protection des 2 premières colonnes. Si après cette manoeuvre, nous tapons RUN de nouveau, nous obtenons :

BONJOUR  
  BONJOUR  
    BONJOUR  
      BONJOUR  
      BONJOUR  
      BONJOUR

Pourquoi les 2 premières colonnes étaient-elles réservées ? On commence à en avoir une idée, puisque l'affichage est maintenant en lettres noires sur fond blanc ! En fait, la première colonne assure le contrôle de la couleur du fond (le papier) et la deuxième la couleur des caractères (l'encre). Pour retrouver l'affichage normal, appuyons de nouveau sur CTRL J. Les 2 premières colonnes étant de nouveau protégées, nous allons pouvoir essayer d'agir sur les couleurs d'affichage. L'ORIC code les couleurs du papier et de l'encre au moyen de chiffres entre 0 et 7.

La commande PAPER (n) va permettre de changer la couleur du fond.

La commande INK (n) va permettre de changer la couleur de l'encre.

La couleur est donnée par la variable n suivant la table ci-dessous :

0 = NOIR  
1 = ROUGE  
2 = VERT  
3 = JAUNE  
4 = BLEU  
5 = MAGENTA  
6 = CYAN  
7 = BLANC.

Tapez en mode direct PAPER 1 (puis RETURN) : vous obtiendrez un affichage en lettres noires sur fond rouge. Tapez ensuite INK 7 et vous aurez des lettres blanches toujours sur fond rouge. Le caractère de contrôle est appelé ATTRIBUT et il modifie tout ce qui suit à partir de lui, jusqu'à l'apparition d'un nouvel attribut. Chaque fois que nous modifions une couleur, l'ORIC descend le long des 2 premières colonnes,

attribuant les 2 caractères de contrôle qui vont déterminer la couleur du papier et celle de l'encre pour le reste de la ligne. Tapez le programme suivant et listez-le à l'écran :

```
10 FOR I=1 TO 7
20 INK I
30 FOR J=0 TO 7
40 PAPER J
50 WAIT 50
60 NEXT J
70 NEXT I
80 INK 0
```

Lancez maintenant le programme et vous verrez **toutes les combinaisons** de couleurs de papier et d'encre possibles. Souvenez-vous **que, quand papier et encre** sont de la même couleur, l'affichage devient invisible !

Une nouvelle fonction nous **permet de modifier un PRINT** pour afficher une chaîne ou une variable directement à **n'importe quelle position** de l'écran. Cela se fait en utilisant le symbole **@**, **suivi de 2 nombres** : le premier est la position de la colonne (comme pour un TAB) **comprise entre 0 et 39**, le deuxième est la ligne d'affichage **comprise entre 0 et 26**. **Pour un exemple rapide**, tapez (en commande directe)

```
10 PRINT @,10,10:"BONJOUR"
```

Comme prévu, **BONJOUR** s'affiche à un quart de l'écran en largeur et à un tiers du bas de l'écran. Cette fonction permet, particulièrement dans les jeux, d'afficher un **texte où l'on veut sur l'écran** ! Essayez l'exemple suivant :

```
5 CLS
10 FOR X=2 TO 30
20 PRINT @X,10;"WHIZZ !"
30 NEXT X
```

Cette sorte d'animation est possible dans toutes les directions, en jouant avec des variables **X** et **Y** pour les colonnes et les lignes. Pendant l'impression, on peut voir **clignoter le curseur**. Si l'on veut éliminer le curseur, on a 2 possibilités : en mode direct, taper **CTRL Q** (les 2 touches à la fois) : le curseur disparaît. **CTRL Q** et il réapparaît.

Il existe un moyen d'activer cette bascule à l'intérieur d'un programme au moyen d'un caractère de contrôle par **CHR\$(17)**. Une version modifiée du programme va nous permettre d'éteindre le curseur en début de programme et de le rallumer à la fin.

```
5 CLS
10 PRINT CHR$(17)
20 FOR X=2 TO 30
30 PRINT @X,10;"WHIZZ !"
40 NEXT X
50 PRINT CHR$(17)
```

Le programme suivant est un jeu utilisant les fonctions que nous connaissons déjà. Vous êtes un chasseur de canards et vous tirez vos flèches sur les canards (>) qui volent en haut de l'écran.

```

10 CLS:S=0:PX=20
20 PAPER 6:PRINT CHR$(17);CHR$(6)
30 REPEAT
40 : PRINT @10,0;"SCORE";S
50 : WAIT RND(1)*100
60 : DX=2 : REPEAT
70 : IF KEY$=CHR$(9) AND PX <37 THEN PX=PX+1
80 : IF KEY$=CHR$(8) AND P>2 THEN PX=PX-1
90 : PRINT @ PX,25;" ^ "
100 : DX=DX+1 : PRINT@ 'DX,3;" >"
110 : IF KEY$="S" THEN 230 ELSE IF KEY$<>" " THEN 200
120 : MX=PX+1 : MY=24
130 : REPEAT
140 : PRINT@MX, MY;" ^ "
150 : PRINT@MX, MY+1;" " ;
160 : MY=MY-1
170 : UNTIL MY=3
180 : IF MX=DX+1 THEN S=S+1:DX=38
190 : PRINT@MX,MY+1;" " :PRINT @ MX, MY; " "
200 UNTIL DX=38
210 PRINT@39,3;" ";
220 UNTIL KEY$="S"
230 PRINTCHR$(17);CHR$(6)

```

Les lignes du programme ont été décalées pour mettre en évidence les différentes parties du programme. Les espaces initiaux ne changent rien à l'exécution du programme, les ":" servent à conserver les espaces (si vous ne les mettez pas, les espaces disparaîtront au premier LIST!)

Vous voulez peut-être savoir comment obtenir 2 couleurs à la fois sur l'écran, pour donner par exemple deux couleurs différentes au chasseur et aux canards ! Nous allons utiliser les attributs collectifs, qui modifient le reste de la ligne. Nous pouvons produire des caractères de contrôle en tapant ESC puis @, A, B, C, D, E, F ou G. Ceux-ci donnent les attributs pour l'encre (couleur de caractère) soit INK 0 à 7. De la même façon, ESC P, Q... à W donnent les attributs pour le fond, soit PAPER 0 à 7.

En mode direct, chaque action sur ESC puis une lettre produit une action immédiate sur l'affichage du reste de la ligne. Essayez différentes commandes à partir de ESC et vous verrez tout de suite comment cela fonctionne.

En mode programme, la touche ESC sera remplacée par CHR\$(27) et l'attribut par un CHR\$. Selon la valeur qui suit CHR\$, on agira sur la couleur du papier ou de l'encre. Si n est le code de la couleur désirée, CHR\$(64+n) donnera la couleur à l'encre (équivalent à INKn) et CHR\$(80+n) donnera la couleur au papier (équivalente à PAPER n). Les 2 CHR\$ devront toujours être utilisés ensemble, et séparés par un point-virgule.

Un court programme ci-dessous vous permet de visualiser les effets de cette commande sur la couleur du fond. vous pouvez remplacer (80+n) par (64+n) à la ligne 20 pour voir les effets sur la couleur d'encre !.

```
10 CLS
20 FOR X=0 TO 7
30 PRINT
40 PRINT CHR$(27);CHR$(80+X);"  BONJOUR !"
50 NEXT X
```

Il existe d'autres attributs spéciaux qui peuvent être portés sur l'écran et qui produisent des effets divers sur le reste de la ligne. Le tableau ci dessous montre les caractères qui peuvent succéder à ESC et les attributs ou effets qu'ils produisent.

Mode direct	code	effet
ESCAPE @	0	encre noire
ESCAPE A	1	encre rouge
ESCAPE B	2	encre verte
ESCAPE C	3	encre jaune
ESCAPE D	4	encre bleue
ESCAPE E	5	encre magenta (bleu-violet)
ESCAPE F	6	encre cyan (bleu-vert)
ESCAPE G	7	encre blanche
ESCAPE H	8	texte standard (après une des instructions suivantes)
ESCAPE I	9	texte alterné ( jeu de caractères PRESTEL)
ESCAPE J	10	standard double hauteur
ESCAPE K	11	alterné double hauteur
ESCAPE L	12	standard clignotant
ESCAPE M	13	alterné clignotant
ESCAPE N	14	standard double hauteur clignotant
ESCAPE O	15	alterné clignotant double hauteur
ESCAPE P	16	papier noir
ESCAPE Q	17	papier rouge
ESCAPE R	18	papier vert
ESCAPE S	19	papier jaune
ESCAPE T	20	papier bleu
ESCAPE U	21	papier magenta
ESCAPE V	22	papier cyan
ESCAPE W	23	papier blanc

Le listing suivant contient à la fois l'utilisation des codes ci-dessus, et l'application de nouveaux codes : nous allons donc l' étudier ligne par ligne:

- la première ligne affiche le code ASCII 12 : c'est un caractère de contrôle qui efface l'écran. C'est en fait une autre façon de faire CLS.
- La deuxième correspond à CTRL D, c'est à dire que c'est la bascule qui provoque la répétition d'une ligne (essayez...) en double affichage, ou qui permet, conjointement avec ESC N, d'obtenir l'affichage en double hauteur.
- la troisième provoque l'affichage clignotant en double hauteur . Utilisé tout seul, l'affichage double hauteur ne permet de voir que la moitié du caractère sur une ligne

de hauteur. Pour voir le caractère entier sur 2 lignes, il est nécessaire de taper **CTRL-D** avant.

- La ligne 50 commute à nouveau l'affichage normal en supprimant la double hauteur clignotant.

```
10 PRINT CHR$(12)
20 PRINT CHR$(4);
30 PRINT CHR$(27);"N";
40 PRINT "ET VOILA, ON CLIGNOTE !"
50 PRINT CHR$(4)
```

Essayez de supprimer les lignes 20 ou 30, ou intercalez une ligne 15 avec un **PRINT!** Vous vous apercevrez qu'en double hauteur, l'ordinateur affiche les moitiés supérieures des lettres sur des lignes impaires et les moitiés inférieures sur les lignes paires. Voici un programme court qui permet de voir l'importance de "caler" le haut et le bas des lignes ! Ajoutez éventuellement un **PRINT ""** en ligne 35 !

```
10 CLS
20 REPEAT
30 FOR X=1 TO 7
40 PRINT CHR$(27);"J";CHR$(27);CHR$(80+X) ;
50 PRINT " ORIC TELESTRAT"
55 WAIT 5
60 NEXT
70 UNTIL KEY$<>" "
```

### 3 - LORES Ø

Dans ce mode, l'ORIC affiche un écran noir sur lequel les caractères standard peuvent être tracés d'une manière semblable à celle où ils ont été affichés en mode **TEXT**. En fait, l'affichage **LORES Ø** est similaire à l'affichage **TEXT** mais l'attribut au début de l'écran est celui qui sélectionne le jeu de caractères standard. Si aucun attribut n'est précisé, l'affichage en encre blanche sur fond noir est choisi par défaut.

Comme pour **PRINT@**, la commande **PLOT** a 2 coordonnées. La première, la coordonnée X varie de 0 à 39 dans la largeur de l'écran. La 2<sup>e</sup>, la coordonnée Y, varie de 0 à 26 dans la hauteur de l'écran. L'utilisation de la commande **PLOT** n'est pas aussi souple que celle de **PRINT @**, car elle affiche obligatoirement une chaîne de caractères. Ainsi, si l'on veut afficher un nombre avec la commande **PLOT**, il faudra passer par une astuce du type **PLOT 10,10,STR\$(1)**.

En revanche, nous pourrions afficher avec **PLOT** des caractères spéciaux : on pourra par exemple afficher des caractères inversés en ajoutant 128 au code ASCII du caractère, chose qui ne peut être faite avec **PRINT !**

Lorsqu'un caractère est inversé à l'écran, il n'y a pas simple inversion entre les couleurs d'encre et de papier : il y a inversion logique des couleurs, c'est à dire que fond et caractère changent de couleur pour prendre la couleur dont le numéro est le complément à 7 ! Par exemple avec une encre noire (Ø) sur fond blanc (7), on obtient en inversant encre blanche (7-Ø=7) sur fond noir (7-7=Ø). De même, avec

encre rouge (1) sur fond bleu (4), on obtiendra en inversant une encre cyan (7-1=6) sur fond jaune (7-3=4).

Le listing suivant crée la chaîne de caractères inversés à partir de la chaîne normale, et ensuite utilise PLOT pour les placer sur l'écran. Juste pour prouver que PRINT ne permet pas d'afficher des caractères inversés, ils sont également affichés en haut de l'écran.

```
10 CLS
20 PAPER1:INK4
30 A$="BONJOUR"
40 FOR I=1 TO LEN(A$)
50 B$=B$+CHR$(ASC(MID$(A$,I))+128)
60 NEXT
70 PLOT 10,10,A$
80 PLOT 10,11,B$
90 PRINT A$,B$
```

Une commande associée à PLOT est SCRN, qui renvoie la valeur affectée à l'emplacement de coordonnées X et Y. Cette valeur est habituellement le code ASCII du caractère affiché à cette position, mais peut être aussi l'attribut figurant là si aucun caractère n'est affiché à cette position. Le listing suivant explique les 2 façons d'utiliser SCRN :

```
10 LORES 0
20 PLOT 10,10,"A"
30 PRINT SCRN(10,10)
40 PRINT SCRN(0,0)
```

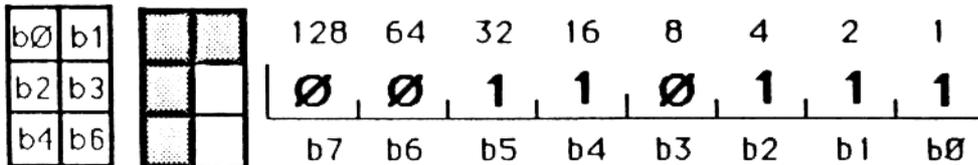
Le 8 produit par le second SCRN représente l'attribut du mode ESC H puisqu'on est en mode texte standard. L'utilisation de ces codes deviendra plus évidente quand nous aborderons le mode LORES 1. Comme on peut s'en douter depuis la syntaxe de SCRN ou de PLOT, l'écran est effectivement représenté en mémoire par un ensemble de 1120 emplacements représentant chacun une position (28 lignes y compris celle du haut x 40 caractères). Ces emplacements mémoire iront de #BB80 à #BFD0 en hexa ou 48000 à 49119 en décimal. Nous pouvons atteindre individuellement ces emplacements en utilisant POKE a,n, qui permettra d'afficher le caractère de code n à la position d'écran correspondant à la case mémoire d'adresse a. Essayez le programme suivant qui remplit tous ces emplacements avec le code ASCII de A:

```
10 FOR I=#BB80 TO #BFE0
20 POKE I,65
30 NEXT I
```



s'arrête au dernier caractère affiché. Une vérification avec **SCRN** devrait éclairer la situation, en ce sens que les autres emplacements ont été fixés à 16, qui est le caractère de contrôle pour **PAPER** noir.

2° cas, le bit b5 est à 1. on a donc affaire à la représentation d'un caractère comme suit :



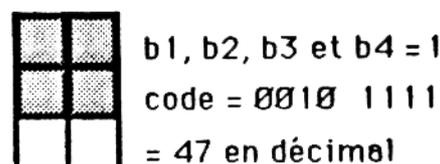
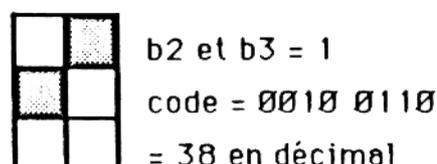
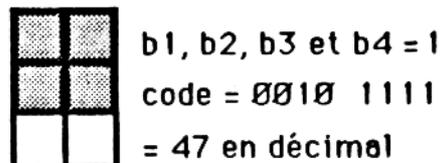
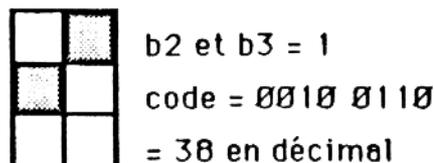
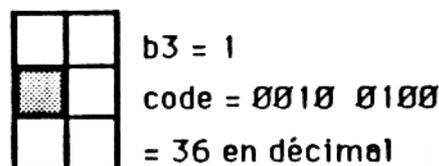
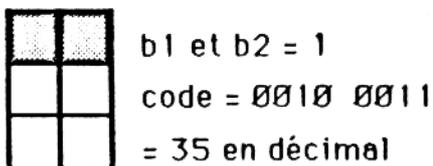
Le caractère ci-dessus est donc représenté par le code binaire **0011 0111**.

Les points allumés sont représentés par des 1 et les points éteints par des 0. Ceci tant qu'on est en vidéo normale, c'est à dire si le bit b7 est à 0.

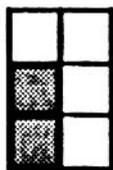
Il est bon de se rappeler que, en mode LORES, on est en affichage blanc sur fond noir à l'origine (c'est à dire si des attributs n'ont pas été spécifiés). Les points dits "allumés" seront donc en blanc et les points "éteints" en noir.

Attention, le bit b5 étant à 1 (sinon, on a un attribut), les codes des caractères commenceront donc à 33 en décimal, et le 6° point sera représenté par le bit b6.

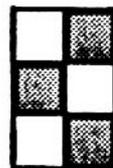
Quelques exemples vous permettront de mieux comprendre :



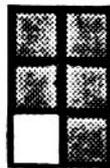
Tant qu'on en est aux "cases" 1 à 5, c'est à dire aux bits b0 à b4, pas de problème. Dès que l'on veut allumer la "case" 6, il faut sauter un bit : le bit 5 devra rester à 1 pour indiquer que l'on a un caractère, et le bit 6 représentera la "case" n°6.



b3 et b5 = 1  
code = 0011 0100  
= 52 en décimal



b2, b3 et b6 = 1  
code = 0110 0110  
= 102 en décimal



à vous de trouver ?

On peut également travailler en vidéo inverse, c'est à dire que les 1 représenteront des "cases éteintes" et les 0 des "cases allumées". Il suffit pour cela de mettre le bit 7 à 1. Pourquoi ce mode, parce qu'il est parfois plus facile de coder 1 bit en vidéo inverse que 5 en vidéo normale ! (voir l'exercice ci dessus , qui donne 1011 0000).

De cette façon, nous pouvons considérer le mode **LORES 1** comme un mode moyenne résolution, l'écran étant composé de cases 0 à 79 en largeur et 0 à 80 en hauteur.

Le programme suivant montre comment tracer un cadre, un titre avec des caractères définis par blocs de 6 cases.

```
10 LORES 1 " "  
20 PLOT 3,2,55- "  
30 PLOT 37,2,180 "  
40 FOR I= 4 TO 36  
50 PLOT I,2,35 "  
60 NEXT I  
70 FOR J=3 TO 18  
80 PLOT 3,J,53 "  
90 PLOT 37,J,181 "  
100 NEXT J  
110 PLOT 3,19,170 "  
120 PLOT 37,19,165 "  
130 FOR I=4 TO 36  
140 PLOT I,19,80 "  
150 NEXT I  
160 FOR N=1 TO 7  
170 READ X,C  
180 PLOT X,10,C  
190 NEXT N  
200 FOR N=1 TO 7  
210 READ X,C  
220 PLOT X,11,C  
230 NEXT  
240 DATA 15,55,16,180,18,55,19,73,21,160,23,55,24,43  
250 DATA 15,170,16,165,18,55,19,182,21,160,23,170,24,167
```

*attention : espace dans chaîne*

*↑ = attention : plus de paramètres*

## 5 - HIRES

C'est le mode Haute résolution. Nous pouvons placer des points n'importe où sur l'écran avec une définition de 240 points en largeur sur 200 en hauteur. Comme dans les autres modes, le point de coordonnées  $\emptyset, \emptyset$  se trouve en haut à gauche. Le mode **HIRES** nous donne accès à de nombreuses commandes puissantes pour les graphismes sur ORIC. La première est **CURSET**, que nous utiliserons pour placer le curseur graphique à un endroit donné de l'écran. Elle est suivie de 3 paramètres X, Y, et p. Les 2 premiers définissent la position du curseur (coordonnées horizontale et verticale), le troisième se retrouvera dans la plupart des commandes **HIRES** : p définit la couleur du point par rapport à celle du fond (définie par **PAPER**) et celle de l'encre, ou du premier plan (définie par **INK**). Ce paramètre p peut avoir n'importe quelle valeur de 0 à 3, qui affecte les commandes de traçage de la façon suivante :

- p =  $\emptyset$  Les points sont tracés dans la couleur du fond (cela peut revenir à ne pas tracer un point)
- p = 1 les points sont tracés dans la couleur du premier plan.
- p = 2 La couleur des points est inversée : si il était dans la couleur du fond, le point passe dans celle du premier plan et vice-versa.
- p = 3 Les points ne sont pas modifiés ni tracés, seule la position du curseur est modifiée.

La commande suivante est **DRAW**, qui présente une syntaxe similaire à celle de **CURSET**, mais qui trace une ligne droite entre la position précédente du curseur (X, Y) et la nouvelle (h,v) définie par les paramètres de **DRAW**. Les paramètres h et v fonctionnent cependant de façon différente de X et Y, en ce sens qu'ils ne représentent pas les coordonnées du point final mais l'incrémentation, c'est à dire que la nouvelle position du curseur sera non pas en h et v, mais en (X+h) et (Y+v). Il découle 2 remarques au sujet de h et v :

- H et V pourront être négatifs (déplacement du curseur vers le haut ou vers la gauche)
- La valeur de h et v devra être calculée en fonction de X et Y pour que les nouvelles coordonnées du curseur restent dans l'écran ( $\emptyset < X+h < 239$ ) et ( $\emptyset < Y < 199$ ).

Le programme suivant nous montre les différences entre **CURSET** et **DRAW**, aussi bien au niveau de la syntaxe que de l'effet. **CURSET** positionne les 4 sommets (lignes 3 $\emptyset$  à 6 $\emptyset$ ) et **DRAW** trace les côtés dès que l'on appuie sur une touche.

```
1 $\emptyset$  PAPER  $\emptyset$ : INK7
2 $\emptyset$  HIRES
3 $\emptyset$  CURSET  $\emptyset, \emptyset, 1$ 
4 $\emptyset$  CURSET  $\emptyset, 199, 1$ 
5 $\emptyset$  CURSET 239, 199, 1
6 $\emptyset$  CURSET 239,  $\emptyset, 1$ 
7 $\emptyset$  GET A$
8 $\emptyset$  CURSET  $\emptyset, \emptyset, 3$ 
9 $\emptyset$  DRAW  $\emptyset, 199, 1$ 
```

```

100 DRAW 239,0,1
110 DRAW 0,-199,1
120 DRAW-239,0,1
130 GET B#:TEXT

```

Il existe une troisième commande de graphiques, qui s'appelle **CURMOV**. Comme **CURSET**, elle déplace la position du curseur sans tracer entre les positions initiales et terminales, et comme **DRAW**, elle utilise des coordonnées relatives h et v. Par exemple, pour tracer 4 points formant les sommets d'un carré de 50 points de côté, on pourra utiliser soit **CURSET**, soit **CURMOV** de la façon suivante :

```

10 PAPER 0 : INK 7
20 HIRES
30 CURSET 10,10,1
40 CURSET 10,60,1
50 CURSET 60,60,1
60 CURSET 60,10,1

```

```

10 PAPER 0:INK 7
20 HIRES
30 CURMOV 10,10,1
40 CURMOV 0,50,1
50 CURMOV 50,0,1
60 CURMOV 0,-50,1

```

La comparaison des paramètres permet de bien comprendre la différence entre ces 2 instructions. Pour mettre en évidence la haute résolution graphique de l'ORIC, essayez le programme suivant . Cela vous donne une figure moirée due à la superposition de lignes très proches.

```

10 PAPER 0 : INK 7
20 HIRES
30 FOR I=1 TO 239 STEP 2
40 CURSET I,0,1
50 DRAW 239-I,199,1
60 CURSET I,199,1
70 DRAW -I,-199,1
80 NEXT I

```

")," To draw 0-I, -199,1

Une caractéristique plus puissante de l'ORIC permet d'agir sur la nature des lignes tracées avec **DRAW** : il s'agit de l'instruction **PATTERN** . Cette commande est suivie d'un paramètre entre 0 et 255 qui définit, par son équivalent binaire, la forme de la ligne sur un intervalle répétitif de 8 points : on convertit le paramètre en binaire, et les chiffres 1 et 0 donnent l'emplacement des points ou des espaces. Par exemple : 0 = pas de point du tout ! 13 devient en binaire 00001101, ce qui donne une ligne composée de segments de 8 points, tous composés de : 4 espaces, 2 points, 1 espace et 1 point . Si le paramètre est 85, soit 01010101, on aura une ligne pointillée avec un point sur deux omis. Avec **PATTERN 255**, la ligne sera pleine. Le programme suivant vous donne toutes les combinaisons possibles de segments de 8 points.

```

10 HIRES
20 FOR I=0 TO 31
30 FOR J=0 TO 7
40 PATTERN J*32+I
50 CURSET J*30,I*6,3
60 DRAW 24,0,1
70 NEXT J
80 NEXT I

```

Le programme suivant permet de combiner l'effet esthétique de l'hyperbole engendrée par ses tangentes, avec les trames créées par la superposition des pointillés.

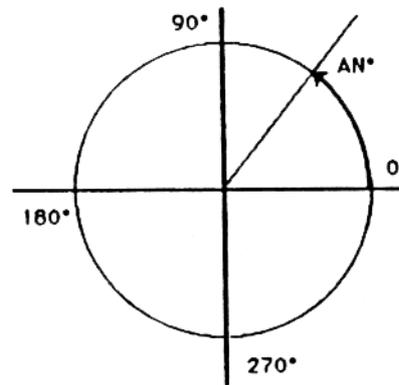
```

10 PAPER1:INK 7
20 HIFES
30 INPUT "PATTERN";P
40 PATTERN P
50 FOR I=0 TO 199 STEP 5
60 CURSET I,0,1
70 DRAW 199-I,I,1
80 CURSET I,199,1
90 DRAW -I, I-199,1
95 GET A#
100 NEXT I

```

Nous arrivons maintenant aux commandes plus sophistiquées que vous offre le BASIC étendu du TELESTRAT.

La première s'appelle **LINE**, et elle fait appel à une variable **AN**, qui définit l'angle courant. Comme en trigonometrie, cet angle a une valeur positive quand le rayon qui l'engendre se déplace dans le sens inverse des aiguilles d'une montre.



Cette variable **AN** va être utilisée avec les instructions **LINE** et **BOX**. Elle est exprimée en degrés.

A partir de la position courante du curseur (ou d'une nouvelle définie par **CURSET**), **LINE n, AN, p** trace une ligne longue de **n** points, de la couleur définie par le paramètre **p** (voir plus haut), dans la direction définie par **AN**. Petit exemple simple pour illustrer cette fonction :

```

10 CLS
20 HIFES
30 FOR I=0 TO 360 STEP 5
40 CURSET 120,100,1
50 LINE 100,I,1
60 NEXT I

```

On observera la principale différence avec **DRAW** qui est que **DRAW 50,50** produit une ligne plus longue que **LINE 50**. La deuxième est, et l'on s'en doutait un peu déjà, que l'affichage de l'ORIC n'est pas "carré", c'est à dire qu'un segment de **n** points tracé horizontalement sera plus court qu'un segment de **n** points tracé verticalement.

Une autre fonction avec une syntaxe similaire sera la fonction **BOX**.

**BOX m,n,p** permet de tracer un rectangle de **m** et **n** points de côté et de mode d'affichage **p**. Ce rectangle aura ses côtés inclinés d'un angle **AN** et son premier sommet situé à la position courante du curseur. La forme la plus simple est:

```

10 HIFES
20 PAPER7:INK0
30 CURSET 40,40,0
40 BOX 50,120,20,1

```

*Box 50, 120, 1*

Pour illustrer les diverses possibilités en jouant sur la taille et l'orientation du rectangle, essayez le programme suivant :

```
10 CLS
20 HIRES
30 FOR I=0 TO 360 STEP 10
35 INKN+1
40 CURSET 120,100,1
50 BOX 80,50,1,1
60 NEXT I
```

**BOX** font appel à des coordonnées relatives. Il existe son équivalent à partir de coordonnées absolues : c'est l'instruction **ABOX**.

On peut également tracer des cercles au moyen de la commande... devinez ? **CIRCLE!** Le cercle verra son centre situé à la position courante du curseur, son rayon précisé en nombre de points et sa couleur dans celle du fond ou du premier plan selon la valeur du paramètre **p**. Sous la forme la plus simple:

```
10 HIRES
20 CURSET 110,100,1
30 CIRCLE 50,1
```

Nous devons nous assurer que, en fonction de la position du centre et de la valeur du rayon, aucun point de la circonférence ne passera en dehors de l'écran. Dans le programme ci-dessous, des cercles successifs sont tracés. Les coordonnées X et Y du centre sont déterminées au hasard (lignes 30 et 40) et les rayons sont alors calculés pour que les cercles soient tangents au bord de l'écran le plus proche.

```
10 PAPER 0:INK 7
20 HIRES:REPEAT
30 X=INT(RND(1)*239)
40 Y=INT(RND(1)*199)
50 IF Y>99 THEN DY =199-Y ELSE DY=Y
60 IF X>119 THEN DX=239-X ELSE DX=X
70 IF DX>DY THEN R=DY ELSE R=DX
80 CURSET X,Y,1
90 IF R>0 THEN CIRCLE R,1
100 UNTIL KEY$="S"
```

Le programme suivant montre une utilisation plus constructive de circle avec des centres non tracés (paramètre **p = 3** pour **CURSET**).

```
10 PAPER 0:INK7
20 HIRES:REPEAT
30 FOR X=3TO50 STEP 3
40 CURSET X+50,100,3
50 CIRCLE X*2,1
60 NEXT
```

## VII - LE LECTEUR DE DISQUETTES

L'utilisation d'un ordinateur puissant impose des moyens de sauvegarde des données rapide et fiable. Le Microdisc 3 pouces ne nécessite aucune manipulation mécanique, son fonctionnement étant géré par le système d'exploitation.

Le lecteur de disquettes double face livré avec l'unité centrale permet de stocker entre 360 et 400 Ko de données. Le Telestrat peut gérer jusqu'à 4 lecteurs de disquettes représentant jusqu'à un peu plus de 4 Mo formatés.

### 2 - PRESENTATION

Le branchement se fait par le câble gris "en nappe" livré avec l'appareil, la prise AMPIENOL 34 broches qui se trouve à son extrémité est munie d'un détrompeur pour placer le connecteur dans le bon sens. L'alimentation du Telestrat sert aussi au Microdisc. La prise DIN située derrière le lecteur permet de relier le transformateur muni d'un interrupteur. Les lecteurs de disquettes supplémentaires doivent avoir leur propre alimentation.

Ces lecteurs auxiliaires se branchent sur le câble du lecteur principal, muni d'une sorte de prise multiple.

### 3 - MANIPULATION

La disquette est contenue dans un boîtier en plastique rigide qui la protège de la poussière et de bon nombre de mauvaises manipulations. Quel habitué des disquettes souples de 13 cm de diamètre (5 " 1/4) n'a pas connu la joie de perdre certains fichiers après avoir manipulé leurs disquettes un peu brutalement.

Le support contenant la disquette est percé de fenêtres permettant la lecture et l'écriture des données. Les trous ronds situés près du bord du boîtier servent à protéger chaque face contre l'écriture (et l'effacement) de nouvelles données. Si le trou est dégagé avec le volet coulissant accessible sur le côté, il devient impossible d'effacer la face correspondante. Malgré ces protections, la disquette est un matériel de précision et doit être manipulée avec soin. En lecture/écriture de fichiers la tête d'enregistrement se déplace à une fraction de millimètres au dessus du disque qui tourne à 300 tours/minute.

### 4 - LE REPERTOIRE DE LA DISQUETTE

La commande DIR affiche le contenu de la disquette livrée avec le Telestrat (LDIR affiche le répertoire sur une imprimante).

Le défilement de la liste peut être interrompu par la frappe de n'importe quelle touche et repris par une pression sur la barre d'espace.

Pour sortir du catalogue, il suffit de taper sur **ESC**.

Chaque nom de fichier (ou de programme) est donné avec le nombre de secteurs qu'il occupe. Si ce nombre est suivi de la lettre **P**, cela signifie que le fichier est protégé en écriture. L'instruction **PROT** (pour "protect" en anglais) permet en effet de verrouiller un fichier contre l'effacement, obtenu par **DEL**.

## **5 - LE SYSTEME D'EXPLOITATION : STRATSED**

Le Système d'Exploitation de Disques ("Disk Operating System" ou DOS en anglais) **STRATSED** gère le transfert des données entre les supports de disquettes et l'unité centrale.

Il comprend des instructions de type **BASIC** qui permettent d'utiliser les lecteurs, sauvegarder un programme (**SAVE**), le charger en mémoire centrale (**LOAD**), utiliser des fichiers (**OPEN, CLOSE, FIELD...**), faire des copies (**COPY, BACKUP...**).

Le **STRATSED** est stocké sur tous les disques formatés par le Telestrat. Une disquette doit être lue au début de l'utilisation pour transférer les instructions du DOS dans la mémoire de travail de l'appareil. Le transfert se fait en 2 secondes environ. Toutes les disquettes utilisées par le Telestrat contiennent donc **STRATSED**. Cela ne fait pas perdre plus de 10 Ko sur une disquette 3" double face qui peut contenir 400 Ko formatés. Le système est copié sur chaque nouvelle disquette dès l'utilisation de la commande **INIT** (initialisation). Le système d'exploitation du Telestrat peut gérer jusqu'à un peu plus de 1 Mo par lecteur de disques (1 million d'octets) et 4 lecteurs de disquettes. Vous pouvez d'ailleurs brancher n'importe quel lecteur de disquettes sur le Telestrat. Le **STRATSED** ne tient pas compte du format physique de la disquette, seulement de son "découpage" en pistes et en secteurs magnétiques, tâche qu'il assure lui-même au moment de l'initialisation.

Par défaut, le système formate les disquettes en 42 pistes par face et 17 secteurs de 256 octets par piste. Ces paramètres correspondent à la norme dite "IBM" ® et assure un bon compromis entre la vitesse d'écriture des données, la quantité d'informations stockées et la sécurité de la sauvegarde. En mode standard, une disquette formatée sur **MICRODISC** contiendra donc un peu plus de 365 Ko utiles, le lecteur étant double face.

L'utilisation des disquettes suppose que celles-ci sont formatées : quand vous insérez dans le lecteur une disquette vierge, elle est illisible pour le Telestrat. Le disque magnétique doit d'abord être divisé en pistes et en secteurs qui formeront les différentes zones où sont stockées les données. C'est ce que fait le Telestrat quand vous tapez la commande **INIT**(initialisation). Une disquette initialisée par le Telestrat n'est donc pas compatible avec une disquette 3 " formatée par un autre appareil. A l'enregistrement d'un fichier (ensemble de données identifié par un nom générique : **PERSONNEL**, **TARIF**...), le système d'exploitation divise les informations en blocs, pour les adapter à chaque secteur. Certains secteurs sont réservés à la table des matières qui contient les adresses et les caractéristiques de tous les fichiers stockés sur le disque. La commande **DIR** permet de lister à l'écran un répertoire du disque. Des instructions permettent de modifier, à l'initialisation, le nombre des pistes et des secteurs d'une disquette, dans la limite des chiffres données ci-dessus. Il faut savoir que la fiabilité et la vitesse des enregistrements diminuent quand on augmente la capacité de stockage. Par contre, la vitesse de lecture ne change pas. Si des valeurs ne sont pas spécifiées, ces paramètres sont donc par défaut :

- 42 pistes par face de disquette,
- 17 secteurs par piste.

Ces valeurs standard correspondent à une capacité de stockage de 365 ko par disquette. Elles permettent de lire et d'écrire dans un fichier à la vitesse de 20 Ko par seconde.

Le lecteur **ORIC** étant double face (une tête de lecture de chaque coté du disque), vous n'avez pas besoin de retourner la disquette pour lire la face **B** après avoir lu la face **A**. Le **STRATSED** traite les deux faces comme une suite de données, vous n'avez pas à préciser où stocker les informations.

- La disquette contient des pistes concentriques qui ressemblent à un disque micro-sillon de 33 tours (sur un tel disque, en fait, il n'y a qu'une seule piste enroulée du bord vers le centre).
- Une piste est divisée en secteurs.
- Un secteur contient 256 octets, mots de 8 bits constitués par les données compréhensibles par l'ordinateur (0 et 1).

Nous avons vu en introduction comment sauvegarder et rappeler un programme (Chapitre I, partie N°5 : **UN PEU DE PRATIQUE**). Vous trouverez dans le **MANUEL DE REFERENCE** la description des commandes qui gèrent les accès disque : **LOAD**, **SAVE**, **SAVEO**, **DIR**, **PROT**, **UNPROT**, **BACKUP**, **COPY**, **DNAME**, **DEL**...

Nous allons définir plus en détail quelques principes concernant les fichiers :

#### a - LES NOMS DE FICHIER

C'est une chaîne de caractères (jusqu'à 16), entourée de guillemets, qui comporte jusqu'à trois parties :

- Le lecteur concerné. STRATSED peut gérer jusqu'à quatre lecteurs indentifiés par les lettres A, B, C et D. Cette partie est facultative.

Si aucun lecteur n'est spécifié, STRATSED utilise le lecteur courant ou le seul lecteur connecté, par défaut. On change la bascule désignant le lecteur courant en inscrivant la lettre l'identifiant suivie d'un tiret : **B-** puis RETURN signifie que le lecteur auxiliaire N°2 doit être utilisé à partir de maintenant. Le prochain **DIR** lira le répertoire du disque logé dans le lecteur B. On peut taper une instruction et indiquer en même temps au Telestrat quel lecteur devient le lecteur courant : **DIR B** affiche le répertoire de la disquette N°2. Après cette instruction, la bascule de sélection étant sur le deuxième lecteur, celui-ci est devenu lecteur par défaut.

- Le nom

C'est la seule partie obligatoire du nom de fichier. Tous les caractères sont acceptés sauf les signes de ponctuation.

- L'extension

Si elle est spécifiée, elle doit être séparée du nom par un point. C'est une abréviation de trois caractères, utilisée par le programmeur pour identifier le type du fichier (les conventions sont généralement : **.BAS** pour indiquer un programme BASIC, **.COM** pour un programme en langage machine, **.DAT** pour un fichier de données, **.TXT** pour une lettre...). La aussi, les minuscules sont automatiquement transformées en majuscules par le Telestrat.

#### "Wild card" (ou joker pour les francophones !)

Une particularité des systèmes évolués est de pouvoir rechercher des fichiers selon certains critères. Le caractère "?" remplace n'importe quelle lettre d'un nom dont on ne connaît pas l'orthographe exacte. Le caractère "\*" remplace un mot entier. Supposons que vous vouliez effacer tous les programmes BASIC d'un disque (non protégés contre l'écriture/effacement), il suffit de taper :

**DEL "\*.BAS"**

Cette commande va détruire tous les fichiers dont l'extension est **.BAS**. Vu l'importance de la commande, le Telestrat vous demandera confirmation mais, de toute façon, attention aux manipulations de ces jokers, gare aux fausses manœuvres !

## b - LES TYPES DE FICHER

Les fichiers peuvent être des programmes en BASIC, des programmes en Assembleur, des fichiers de données et des fichiers d'applications de logiciels. Il est donc utile de spécifier une extension pour les différencier. Une instruction permet au Telestrat de donner une extension par défaut. Si vous écrivez surtout des programmes en BASIC, il est plus facile de laisser au Telestrat le soin de les marquer, dans le répertoire : la commande **EXT "BAS"** indique à l'ordinateur que l'extension à placer après chaque nom est **[.BAS]**.

## c - GESTION DES FICHIERS DE DONNEES

Le Telestrat peut travailler sur 64 fichiers ouverts en même temps. Nous traitons ici des fichiers formés de données brutes : les résultats des calculs faits dans un programme, les noms de vos employés... Les données sont toujours stockées sous forme de texte, les chiffres doivent être reconverties pour être utilisés dans des calculs arithmétiques. Ces fichiers sont souvent identifiés par une extension **[.DAT]** pour "data" ou "données" en français.

Pour utiliser de tels fichiers, il faut les ouvrir en indiquant leurs noms et leur associer un numéro d'ordre pour les identifier (**OPEN S, "TARIFS", 3**). Les opérations de lecture et d'écriture sont alors possibles. Une fois les informations manipulées, les fichiers doivent être refermés (**CLOSE 3** ferme le fichier **TARIF**, ouvert sous le N°3).

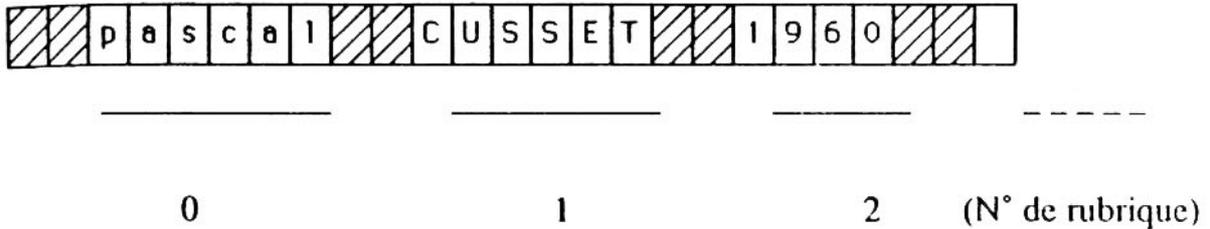
Il existe deux types principaux de fichiers de données, identifiés par le mode d'accès séquentiel ou direct : Le meilleur exemple est celui d'un magnétophone à cassette par rapport à une platine disque. Pour écouter un morceau situé au milieu de la cassette, il faut faire défiler la bande jusqu'à l'endroit voulu. Sur un disque, on soulève le bras pour le poser sur la plage du morceau choisi, l'accès *direct* est donc plus rapide.

### • *Les fichiers à accès séquentiel.*

Les informations sont lues les unes après les autres, dans leur ordre d'écriture. Avant d'arriver aux données recherchées, un pointeur se déplace sur chaque donnée. La lecture et l'écriture se font là où est le pointeur, commandé par les instructions **APPEND, REWIND, JUMP...** Le fichier est composé par une suite

de rubriques, séparées par deux caractères spéciaux placés par le Telestrat.

Un tel fichier est simple à utiliser et doit être choisi quand les informations sont à consulter dans l'ordre de leur écriture ou quand il y a peu de modifications à effectuer. En voici un schéma :

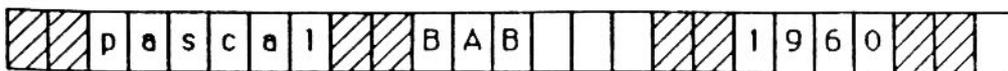


Il y a trois rubriques dans cet exemple. Les caractères se suivent et les rubriques sont séparées par deux octets qui servent au système d'exploitation pour définir les caractéristiques et séparer les champs. Les définitions de champs sont : réel [!], octet [o], chaîne de caractères [\$], entier [%]. Vous verrez dans le manuel de référence qu'il existe aussi un type de fichier dit "à accès disque" qui ne nécessite pas ces espaces et qui permet aux utilisateurs chevronnés de travailler secteur par secteur.

• **Les fichiers à accès direct.**

Ce type reprend les caractéristiques précédentes et permet l'accès à une position précise, dans les données. Lors de la création d'un tel fichier, il faut indiquer la longueur standard des enregistrements (en quelque sorte, la taille de la fiche type), celle-ci est fixe et déterminée. Chaque enregistrement est sub-divisé en rubriques. Si toutes les rubriques concernant l'enregistrement N°n ne sont pas utilisées, une place suffisante est laissée.

Les particularités de cette méthode sont une organisation régulière des informations et la rapidité d'accès dans le cas des gros fichiers, souvent remis à jour. C'est le cas des stocks d'une entreprise quand ceux-ci doivent être tenus en temps réel. Ces fichiers sont lus et écrits d'une façon similaire aux fichiers séquentiels hormis des paramètres supplémentaires permettant l'accès à un champ d'informations. Ceci impose l'utilisation d'une mémoire tampon pour échanger les données entre le Telestrat et la disquette. Cette mémoire joue le rôle d'un masque qui permet de formater les données pour les faire tenir dans les rubriques pré-définies. Les instructions de base d'un fichier à accès direct sont : OPEN, CLOSE, PUT, TAKE, FIELD, LSET, RSET...



0

1

2

0

---

enregistrement N°0

N°1

Dans l'enregistrement 0, qui concerne un individu du fichier, il y a 3 rubriques : la première contient le prénom, la seconde contient le nom et la troisième est un champ numérique contenant la date de naissance. Il y a donc une personne par enregistrement.

## 5 - STOCKAGE DES DONNEES

L'instruction de base pour sauvegarder un fichier est **SAVE**, avec différentes options comme l'exécution automatique après chargement :

**SAVE "TEST2", AUTO...** D'autres commandes complètent l'instruction :

**SAVEU** (remplace une version déjà existante sur le disque, celle-ci prenant une extension **.BAK**), **SAVEO** (remplace une ancienne version sans garder de copie de celle-ci), **SAVEM** (stocke un programme à la suite d'un autre). **ESAVE** permet de sauvegarder une image de l'écran (Haute ou basse résolution) : **ESAVE "ECRAN4"**.

## 6 - LECTURE DES DONNEES

L'instruction principale est **LOAD**, suivie du nom du programme à charger en mémoire centrale entre guillemets (**LOAD "TEST"**). Des options permettent de mettre ce programme à la suite de celui qui se trouve encore en mémoire ("merge" en anglais).

Dans la pratique, vous n'êtes pas obligé d'utiliser cette commande, il suffit de taper le nom du programme pour que celui-ci se charge en mémoire centrale (**TEST** suivi d'une pression sur la touche **RETURN** charge le programme appelé **TEST** en mémoire). La commande complète est nécessaire quand le fichier à charger a un nom de variable (**A\$, B3%...**).

## VIII - COMPATIBILITE AVEC LES ORDINATEURS ORIC

Le Telestrat est le dernier né d'une grande famille. Avec seulement deux produits, l'ORIC-1 et l'ATMOS, la firme aujourd'hui française ORIC a déjà vendu des centaines de milliers d'appareils dans le monde.

Il existe environ 2 000 logiciels écrits pour ces machines. Bien que le Telestrat soit d'une conception différente de ses deux ancêtres, il garde la possibilité de les émuler (simuler leur fonctionnement). Deux logiciels stockés sur la disquette système permettent de simuler l'ORIC 1 ou l'ATMOS. Il suffit de taper **ORIC1** ou **ATMOS** pour charger ces programmes en mémoire centrale, le Telestrat se met automatiquement dans le mode d'émulation.

L'émulation équivaut à charger dans la RAM du Telestrat la ROM des anciennes machines. Le Telestrat tournant avec ces fichiers devient un autre appareil. **Il ne peut donc plus avoir les fonctions de son BASIC étendu et du système d'exploitation STRATSED.** Il ne fonctionne plus qu'avec un lecteur de cassettes pour stocker et lire les données. Pour revenir au mode Telestrat, il suffit de faire un reset général (touches DEL et RESET simultanément).

En émulation, tous les logiciels faits pour les précédentes machines et stockés sur cassettes peuvent être utilisés et chargés dans un Telestrat.

Les programmes stockés sur disquettes ne sont pas utilisables avec ces ROM factices. Elles occupent la place du STRATSED, dans la mémoire du Telestrat. Il n'y a plus de place pour le système de gestion des lecteurs de disques. Certains fichiers non protégés sur disquettes peuvent être récupérés mais certaines adaptations risquent d'être faites.

## IX - LES PERIPHERIQUES

Le Telestrat est compatible avec la plupart des standards utilisés pour les périphériques micro-informatique.

### 1 - LECTEURS DE DISQUETTES AUXILIAIRES

Si vous voulez installer plusieurs lecteurs, il suffit de les raccorder selon la photo ci-après. Le lecteur principal restant branché sur la prise marquée MICRODISC. Les lecteurs auxiliaires se branchent sur le câble du lecteur les précédent. Des alimentations ORIC doivent être ajoutées à partir du deuxième Microdisc, celles-ci étant conçues pour deux Microdisc ou un Telestrat et un Microdisc. Le lecteur principal est appelé maître ("master" en anglais), les autres sont considérés comme des lecteurs esclave. Le lecteur vendu avec le Telestrat est programmé pour être un lecteur maître (DRIVE A s'affiche à l'initialisation du système) : Le Telestrat donnera priorité à ce lecteur après un reset général, il lui faut en effet lire un disque pour recharger le SRATSED en mémoire. En fonctionnement, les autres lecteurs sont repérés par les codes B, C et D.

Une bascule vous permet de signaler au Telestrat quel lecteur utiliser pour ses opérations : Il faut taper la lettre correspondant au lecteur suivie d'un tiret (B- donne la main au deuxième lecteur pour les opérations suivant la commande). En cas de reset, le lecteur maître reprend toujours la main. Certaines commandes permettent d'échanger des fichiers entre les lecteurs :

#### **COPY "A-ESSAI2" TO "B-TEST"**

permet par exemple de recopier le fichier ESSAI2, situé sur le disque A, sur le disque B, en même temps, le fichier est rebaptisé TEST et garde la même extension.

### 2 - LES IMPRIMANTES

Vous pouvez relier la plupart des types d'imprimante au Telestrat, qu'il s'agisse d'une imprimante parallèle ou série.

Une imprimante **parallèle** reçoit les caractères les uns après les autres sur huit fils simultanément (un caractère correspond à un octet ou huit bits). Les caractères de commande de l'imprimante sont transmis sur d'autres fils (longueur des lignes, taille des caractères, saut de page...). Ce type d'imprimante est le plus courant. Les appareils se relient à la prise marquée IMPRIMANTE sur la face arrière du

Telestrat. Cette prise correspond au standard CENTRONICS parallèle. Les commandes pour imprimer sont **LLIST**, **LPRINT**...Un **L** comme "Line" s'ajoute aux instructions courantes du Telestrat. Le mini traceur couleurs MCP-40 d'ORIC se connecte à cette prise.

Une imprimante série ou un traceur de courbes reçoivent les données bit par bit sur un seul fil. Les fils supplémentaires sont utilisés pour les commandes de l'imprimante. Ces appareils se connectent à la sortie **RS232**, à l'arrière gauche du Telestrat. Des instructions permettent d'utiliser cette interface : **SDIR**, **SLIST**, **SPRINT**...Un **S** comme "Serial" précède les instructions courantes.

Il existe plusieurs types d'imprimantes :

- Les imprimante thermiques qui utilisent un papier sensible à la chaleur. La tête d'écriture chauffe certains points de sa matrice pour faire réagir le papier situé sous elle, selon les caractères à dessiner.
- Les imprimantes matricielles ont des têtes d'écriture formées d'aiguilles appuyant sur un ruban encreure. Certains modèles, du fait du grand nombre d'aiguilles de la matrice, ont une qualité qui s'approche de la qualité courrier.
- Les imprimantes de qualité courrier ressemblent à des machines à écrire normales, à la différence qu'elles peuvent être commandées par un ordinateur. Les imprimantes à marguerite sont une catégorie d'imprimantes qualité courrier. Les lettres sont moulées sur une sorte d'étoile qui tourne avant d'être frappée par un marteau pressant la lettre désirée sur le ruban encreur.

### **3 - LES MONITEURS / ECRANS**

Toutes sortes de moniteurs peuvent être reliés au Telestrat, certains nécessitant un modulateur pour rendre compatible les signaux émis par l'ordinateur et reçus par l'écran. A l'origine, le Telestrat est prévu pour se connecter à un appareil équipé d'une prise **PERITEL**, il peut s'agir d'un moniteur ou d'un appareil de télévision. Le moniteur se relie à la prise marquée **RVB** (Rouge/Vert/Bleu), à l'arrière du Telestrat. Si le moniteur fonctionne en vidéo-composite (**PAL**, **SECAM** ou **NTSC** selon les pays), il faut des adaptateurs Haute Fréquence. Ces adaptateurs sont aussi nécessaires pour d'anciens postes de télévision, non équipés de prises **PERITEL**, il faut alors relier le modulateur au Telestrat et à la prise d'antenne **UHF** du poste. Signalons que le Telestrat possède plusieurs modes d'affichage en couleurs : **LORES**, **HIRE**...qui correspondent à la basse et haute résolution selon les applications de textes ou de graphiques. La commande **ESAVE** permet de sauvegarder une image de l'écran en mémoire. Un écran en couleurs est recommandé pour utiliser au mieux les possibilités graphiques du Telestrat.

#### 4 - MAGNETOPHONE

C'est un moyen peu coûteux de stocker ses programmes. Comme le Telestrat est livré avec un lecteur de disquettes, il devient moins intéressant d'utiliser des bandes magnétiques, si ce n'est pour lire les programmes faits pour l'ORIC-1 et l'ORIC ATMOS (voir chapitre VIII : COMPATIBILITE). Un magnétophone est moins fiable qu'un lecteur de disquettes et beaucoup moins rapide quant aux temps d'accès aux programmes. Le magnétophone se branche à la prise marquée CASS., située sur le bord du panneau arrière. Soulignons que n'importe quel magnétophone peut être utilisé, qu'il soit à bande ou à cassette. Les instructions CSAVE et CLOAD servent à sauvegarder et à lire des programmes stockés sur bande magnétique. Vous trouverez dans le Manuel de Référence les options de ces commandes.

- Pour enregistrer un programme sur un magnétophone, il faut faire démarrer le magnétophone en mode enregistrement (RECORD) puis taper CSAVE "TEST" sur le Telestrat (TEST est le nom donné au programme). L'ordinateur envoie les données sur la bande magnétique et signale la fin des opérations. Vous pouvez alors arrêter le magnétophone.
- La lecture d'un programme se fait en tapant CLOAD "TEST" en lançant le magnétophone en mode lecture (PLAY). Si vous tapez CLOAD "", le Telestrat chargera le premier programme rencontré sur la bande.

Pour toutes ces opérations, n'oubliez pas de confirmer les commandes par la touche RETURN.

#### 5 - MANETTES DE JEU ET SOURIS

De chaque côté du Telestrat, il est possible de relier des "joysticks" ou manches à balai. Ces prises 9 broches répondent aux normes ATARI ® qui forment le standard en la matière.

Du côté droit, la prise est conçue pour connecter une souris, dans ce cas, tous ses boutons sont utilisables. Si la souris est connectée à la prise de gauche, seul le bouton gauche de la souris pourra fonctionner.

#### 6 - MINTEL

Cette prise DIN 8 broches, marquée MINDIHEL, permet de brancher un Minitel, celui-ci doit être de type MIR (Retournable) pour les applications de serveur monovoie.

Vous avez accès ainsi à toutes les applications de la télématique. Consultez le Manuel des Applications Télématique pour avoir des détails sur ces fonctions.

## **7 - NOYAU MIDI**

Une prise DIN 5 broches, marquée **EXT. MIDI**, permet de relier le boîtier MIDI (Musical Instrument Digital Interface) qui donne accès aux commandes des instruments de musique électronique : synthétiseurs, boîtes à rythme, séquenceurs entre autre. Une cartouche de logiciels d'applications est prévue pour utiliser cette interface.

## **8 - MODEM**

Un modem peut être connecté sur la prise **RS232**. Cette prise accepte tous les types de modem. Le Telestrat peut utiliser n'importe quel modem pour transmettre et émettre des données via la ligne téléphonique.

## **9 - EXTENSION DE L'ORDINATEUR**

Cette prise, marquée **BUS D'EXTENSION** est une extension complète du BUS de données IDC 34 broches. Elle permet de développer des applications spécifiques comme faire piloter par le Telestrat des instruments de laboratoire ou brancher des interfaces supplémentaires. Le microprocesseur 6502 de l'ORIC comporte 8 lignes de données et 16 lignes d'adresses. Ces lignes permettent l'échange des données. on peut concevoir ses propres périphériques en utilisant des circuits d'interface, sachant que sur une ligne, un état activé (bit à 1) correspond à une tension de +5 V et un état désactivé correspond à une tension de 0 V. On peut ainsi obtenir différents signaux et activer des circuits déterminés.

## Annexe 1 - SPECIFICATIONS

### PROCESSEUR

M 6502A, 8 bits, fréquence d'horloge 1 MHz

### MEMOIRE

ROM 48 K octets

RAM 64 K octets extensibles à 128 Ko

### LECTEUR DE DISQUES

Pour disquette 3 pouces simple et double faces, jusqu'à 400 Ko par disquette (365 Ko en mode standard), vitesse de lecture / écriture de plus de 20 Ko par seconde.

### AFFICHAGE : 8 COULEURS D'ECRITURE SUR 8 COULEURS DE FOND

TEXTE (LORES 0)

40 colonnes/28 lignes

MOSAIQUE (LORES 1, caractères alternés)

40 colonnes/28 lignes

GRAPHIQUE (HIRES, H. résolution)

240 x 200 points et 3  
lignes de 40 colonnes

### ENTREES / SORTIES

RS 232

MINI TEL

LECTEURS DE DISQUES

BUS D'EXTENSION

CENTRONICS PARALLELE

NOYAU MIDI (synthétiseurs)

RVB (écran)

MAGNETOPHONE

Prise DIN 7 broches

2 JOYSTICKS / SOURIS

Prises 9 broches standard.

### CARTOUCHES

ROM / RAM jusqu'à 64 Ko

### GENERATEUR SONORE

3 voix. Transmission au haut-parleur du moniteur.

### CLAVIER DE 58 TOUCHES

Reconfigurable directement ou par logiciel, répétition automatique,  
11 Touches spéciales de fonction.

### ALIMENTATION

Entrée 220 V alternatif / Sortie 5 V - 12 V régulée.  
Valable pour un lecteur de disques et un Telestrat ou 2 lecteurs.

### UNITE CENTRALE

#### DIMENSIONS MAXIMALES :

Hauteur	Largeur	Profondeur
71 mm	346 mm	260 mm

### MICRODISC (sur socle)

H	L	P
116	120	290

### ALIMENTATION

H	L	P
78	94	153

#### POIDS :

1,4 kg

1,2 Kg

1,7 Kg

### TEMPERATURE D'UTILISATION

De 0° C à 35° C

### HUMIDITE RELATIVE

De 5% à 90%

# Annexe

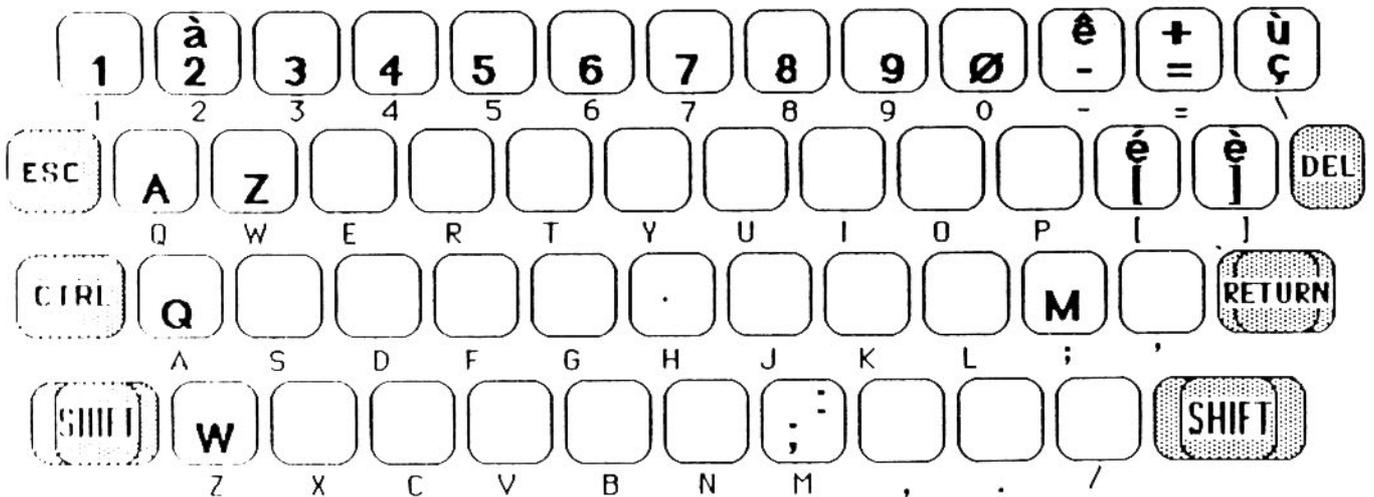
## LES MODES DE CLAVIER DU TELESTRAT

L'**HYPER-BASIC** du **TELESTRAT** permet de sélectionner divers modes de clavier pour l'adapter aux programmes utilisés, ou aux habitudes de l'utilisateur !

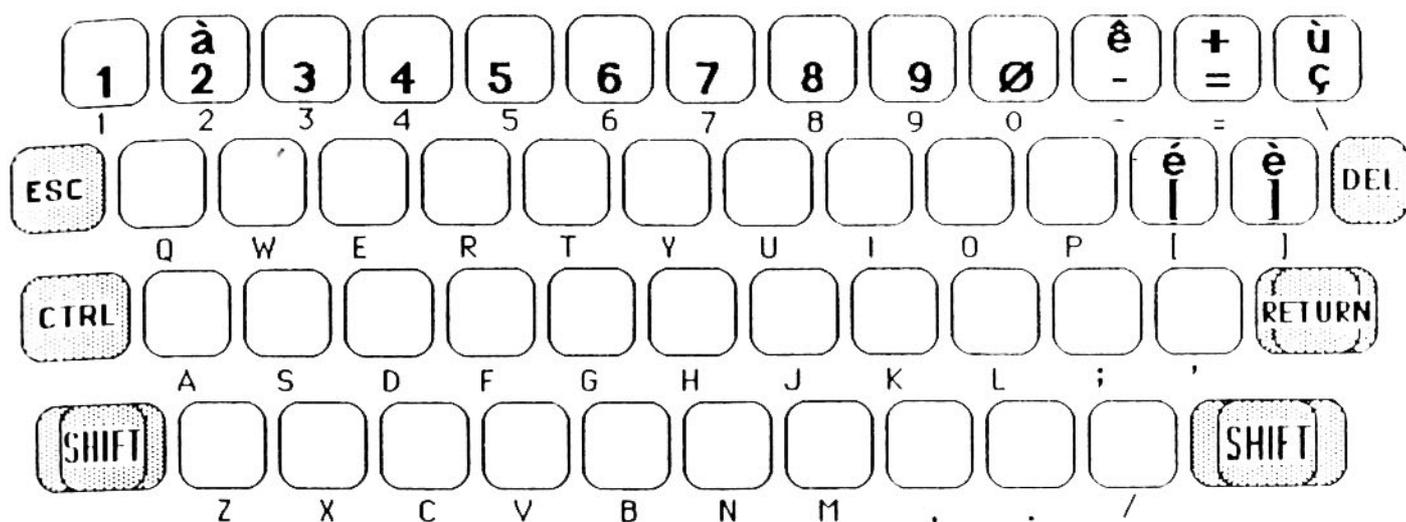
Le **TELESTRAT** est livré avec des petits autocollants à placer sur les touches si l'on utilise une configuration de clavier de façon courante.

Les schémas ci-après montrent les différentes configurations de clavier. La touche originale est indiquée au dessous. Sur la touche elle-même, figurent les caractères redéfinis : au dessous celui obtenu directement, au dessus celui obtenu avec **SHIFT**. Pour plus de clarté, seules les touches qui ont été modifiées ou déplacées sont indiquées.

**AZERTY** permet de redéfinir le clavier pour les gens qui ont l'habitude de taper sur un clavier de machine à écrire au standard français. Cette configuration comprend les caractères accentués, et affecte aux caractères modifiés les codes ASCII qui permettent d'obtenir une impression correcte avec les imprimantes en configuration "caractères français". C'est pour cela qu'il ne sont pas à l'endroit exact où ils se trouveraient sur une machine à écrire. Seuls A, Z, Q, W, M et ; ont été déplacés. Pour une plus grande facilité d'utilisation, particulièrement en mode programmation, les lettres ont été gardées en majuscules, et les chiffres s'obtiennent sans le **SHIFT**.



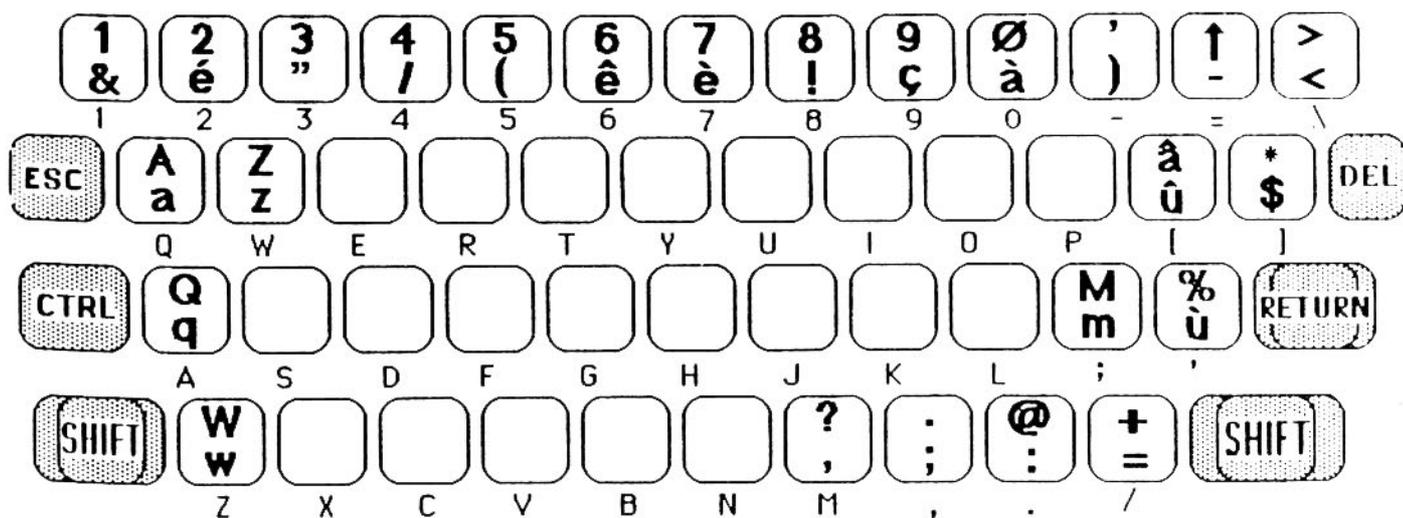
### CLAVIER 'AZERTY'



### CLAVIER 'ACCENT'

**ACCENT** permet d'obtenir les caractères accentués sans modifier le reste du clavier. L'accent circonflexe n'étant pas disponible séparément, les différents **e** sont accentués directement.

**FRENCH** permet de retrouver le clavier d'une machine à écrire : lettres majuscules et minuscules, chiffres avec **SHIFT** etc.. Seule différence, l'accent circonflexe n'est pas disponible seul, pour des raisons techniques. On a donc le jeu de voyelles accentuées.



### CLAVIER 'FRENCH'

## Annexe 6 - LISTE DES CARACTERES DE CONTROLE AVEC LEUR CODE ASCII.

Codes CTRL	Fonction	Code décimal
CTRL A	Tabulation (saut de 8 caractères).	1
CTRL C	BREAK, interruption de programme/accès aux applications de la télématique quand la cartouche spéciale est présente.	3
CTRL D	Bascule de double hauteur.	4
CTRL F	Bascule de l'écho sonore des touches.	6
CTRL G	Emission d'un bip sonore (correspond à OUPS en BASIC)	7
CTRL H	Curseur vers la gauche.	8
CTRL I	Curseur vers la droite.	9
CTRL J	Curseur vers le bas.	10
CTRL K	Curseur vers le haut.	11
CTRL L	Effacement de l'écran.	12
CTRL M	RETURN, retour chariot.	13
CTRL N	Effacement de la ligne courante.	14
CTRL O		15
CTRL P	Curseur fixe/curseur clignotant.	16
CTRL Q	Curseur affiché/non affiché.	17
CTRL R	Bascule de branchement de l'imprimante	18
CTRL S	Copie d'écran sur imprimante (mode texte).	19
CTRL T	Bascule de verrouillage majuscule/minuscule.	20
CTRL U		21
CTRL V	Bascule de vidéo inverse.	22
CTRL W		23
CTRL X	Effacement entre le curseur et la fin de la ligne.	24
^CTRL-DEL	Adressage du curseur (caractère ASCII : US)	31

Ces caractères peuvent être programmés en utilisant la fonction **PRINT CHR\$(n)**, n étant le code ASCII décimal correspondant. *PRINT CHR\$(7)* déclenche un bip sonore.

*CTRL-DEL* permet de placer le curseur n'importe où sur l'écran par la même syntaxe qu'en mode Videotex : *US A,B* (CTRL-DEL A,B) place par exemple le curseur en coordonnées (1, 2) sur l'écran (le code ASCII 1 qui correspond à A sert de point de départ, de coordonnée d'origine. Le caractère C, de code 3, représente donc la coordonnée 3 et ainsi de suite).

## B - LISTE DES CODES ESCAPE

CARACTERE	FONCTION	CODE BASIC
ESCAPE @	Encre noire	0
ESCAPE A	Encre rouge	1
ESCAPE B	Encre verte	2
ESCAPE C	Encre jaune	3
ESCAPE D	Encre bleue	4
ESCAPE E	Encre magenta	5
ESCAPE F	Encre cyan	6
ESCAPE G	Encre blanche	7
ESCAPE H	Texte standard	
ESCAPE I	Caractères "semi-graphiques"	
ESCAPE J	Double hauteur	
ESCAPE K	Double hauteur des caractères graphiques	
ESCAPE L	Clignotement	
ESCAPE M	Clignotement des caractères graphiques	
ESCAPE N	Clignotement en double hauteur	
ESCAPE O	Clignotement des caractères graphiques en double hauteur.	
ESCAPE P	Papier noir	0
ESCAPE Q	Papier rouge	1
ESCAPE R	Papier vert	2
ESCAPE S	Papier jaune	3
ESCAPE T	Papier bleu	4
ESCAPE U	Papier magenta	5
ESCAPE V	Papier cyan	6
ESCAPE W	Papier blanc	7
ESCAPE X	Synchronisation de l'écran : texte 60Hz	
ESCAPE Y	Synchronisation de l'écran : texte 60Hz	
ESCAPE Z	Synchronisation de l'écran : texte 50Hz	
ESCAPE [	Synchronisation de l'écran : texte 50Hz	
ESCAPE }	Synchronisation de l'écran : graphique 60Hz	
ESCAPE ~	Synchronisation de l'écran : graphique 50Hz	
ESCAPE DEL	Synchronisation de l'écran : graphique 50Hz	

Ces codes peuvent être programmés en utilisant la fonction **PRINT CHR\$( 27)** suivie du caractère présenté sous forme de chaîne (entre guillemet ou introduit par CHR\$). 27 est le code ASCII décimal de la touche ESC. *PRINT CHR\$(27) "I"* donne la couleur bleu au texte.

Les codes correspondent aux instructions BASIC : **INK n** et **PAPER n** (n variant donc de 0 à 7). Ces instructions commandent directement la couleur du texte (INK) et du fond de l'écran (PAPER).

Scanné par Andrec