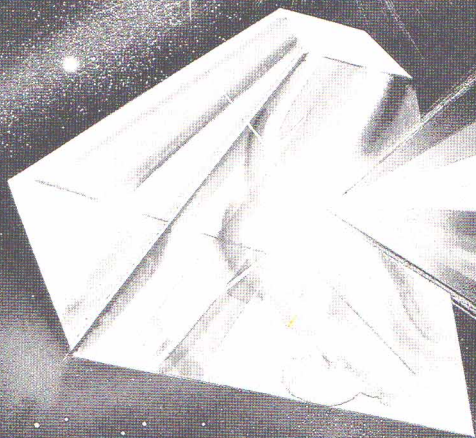


P-Y-R-A-D-E-V

# Manuel d'Utilisation

THE  
COMPLETE  
AMSTRAD  
UTILITY



EDITOR  
ASSEMBLER  
MONITOR  
DISC-NURSE  
UTILITIES  
SELF-TEACH

AMSTRAD CPC 464/664/6128

WHEN YOU HAVE MADE THE

# DISCOVERY

NOTHING ELSE COMES CLOSE

# Manuel d'Utilisation Section 1

## *PYRADEV*

## Table des Matieres

○ Introduction .....	3
○ Préparation .....	4
○ Nouveaux utilisateurs .....	5
○ Disques de travail .....	5
○ Restrictions imposées par le système .....	5

## Section 2

### *Editeur*

○ Sélection des fichiers .....	6
○ Affichage mode édition .....	6
○ Page SOS-ESCAPE .....	6
○ Session ABANDON Edition .....	7
○ File SUSPENSION & SORTIE .....	7
○ Mouvement du curseur .....	8
○ Défilement de l'écran .....	8
○ Insertion/effacement caractères .....	8
○ Insertion/effacement lignes .....	8
○ Copie blocs, effacement .....	9
○ Taquets de tabulation .....	9
○ Annexation de fichiers .....	10
○ Recherche, Arrêt, Remplacement .....	10
○ Mémoire de Frappes .....	11
○ Impression .....	12

○ Affichage répertoire .....	12
○ Renommer fichiers effacés .....	13
○ Mode compression .....	13
○ Changement couleurs encres .....	13
○ Fonctions sauvegarde : chargement .....	13
○ fichiers ZAP .....	13

## Section 3

### *Assembleur*

○ Options Début/Exécution .....	13
○ Commande assemblage .....	14
○ Définition Langage .....	18
○ Expressions .....	19
○ Directives .....	19
○ Assemblage conditionnel .....	23
○ Macro-Instructions .....	24
○ Codes opérations LIBRES .....	25
○ Gestion des mémoires .....	28
○ Statistiques d'assemblage .....	28
○ Récapitulatif des directives .....	29

## Section 4

### *Moniteur*

○ Affichage de l'écran .....	31
○ Translation .....	31
○ Code Chargement test .....	32
○ Changement code, mémoire .....	32
○ Définition pièges .....	33
○ GOTO adresse .....	33
○ Etude pas à pas du code .....	33
○ Code, pas double .....	33
○ Changement registres .....	33
○ Régénération de l'écran .....	34
○ Mode Interrogation/recherche .....	34
○ Mise en route et affichage ROM .....	34
○ Ecriture binaire sur disque .....	34
○ Désassemblage à source .....	34
○ Copie mémoire .....	35
○ Impression écran .....	35
○ Couleurs papier et plume .....	35
○ Affectation zones d'empilage .....	35
○ Accès externe .....	35

## Section 5

### *Disc-Nurse*

○ Prise en charge, disque .....	35
○ Sélection Unité de disques .....	36
○ Sélection piste .....	36
○ Sélection secteur .....	36
○ Lecture secteur .....	36
○ Modification secteur .....	37
○ Ecriture secteur .....	37
○ Non-effacement fichier .....	37
○ Sélection fichier .....	37
○ Mode interrogation/recherche .....	37
○ Impression écran .....	38
○ Couleurs Papier/plumes (fond/graphiques) .....	38

## Section 6

### *Programme Utilitaire*

○ Effacement fichiers .....	38
○ Renommer fichiers .....	38
○ Copie fichiers .....	38
○ Restrictions de copie .....	39

## Section 7

### *Programme d'entraînement*

○ Programme de démonstration .....	40
○ Programme Test Assemblage .....	40
○ Informations générales .....	40

## Section 8

○ Récapitulatif des touches de commande .....	41
---	----

## 1.0 INTRODUCTION

Le PYRADEV est un ensemble de cinq programmes qui, regroupés, fournissent un environnement sûr et confortable pour l'élaboration des logiciels AMSDOS et CPM sur les micro-ordinateurs AMSTRAD 464, 664 et 6128 équipés d'un système de disques DD1. Le système est basé sur l'AMSDOS et n'exige aucune mémoire ROM ou RAM supplémentaire. Il peut prendre en charge d'importants projets de programmation sur systèmes à une ou deux unités, et comporte les programmes suivants:

- Un EDITEUR DE FICHIER SOURCE ultra-rapide, pleine page.
- Un MACRO ASSEMBLEUR/éditeur de liens Z80, multi-fichiers.
- Un puissant MONITEUR et DESASSEMBLEUR de mise au point.
- Un programme DISC-NURSE d'une grande simplicité d'utilisation.
- Un excellent programme utilitaire de GESTION DE FICHIERS.

L'EDITEUR utilise ses propres sous-programmes de lecture et d'écriture de disques spéciaux, ce qui permet d'extraire d'éditer rapidement d'importants fichiers (jusqu'à 32.300 octets). Le code peut être recherché, remplacé, entré, supprimé, déplacé, copié et fusionné (entre les fichiers). En outre, on peut effacer, rebaptiser, supprimer, etc les fichiers. Tous les programmes utilisent la compression des données afin de garantir un emploi optimal de la mémoire AMSTRAD et de la capacité des disques.

L'ASSEMBLEUR permet de sélectionner jusqu'à 32 fichiers dans un seul assemblage. Ces fichiers peuvent se trouver sur des disques différents (si l'on utilise deux unités), ce qui permet de traiter en un seul fichier binaire jusqu'à 992K de code source (5 disques environ). On peut commuter sur une sortie vidéo ou sur imprimante selon les besoins, et produire des tables de correspondance. L'assembleur est extrêmement rapide et peut traiter, sur les systèmes plus grands, la mémoire RAM supplémentaire est utilisée au maximum afin d'augmenter les performances.

Le MONITEUR auto-translatable peut charger, interrompre, piéger, fonctionner en pas à pas, ou en deux pas, modifier et ré-écrire le code sur disque. Par ailleurs, on peut sélectionner et analyser des mémoires ROM. En mode de mise au point, on peut modifier les registres et obtenir à tout moment des impressions sur écran. le désassemblage du code peut être sorti sur imprimante ou mis en fichiers ASCII.

Le programme DISC-NURSE traite les disques système, données et style IBM. On peut rechercher, visualiser, imprimer, modifier et ré-écrire des secteurs de fichiers. La recherche peut également, si besoin est, s'étendre sur la totalité du disque. Chaque fois que le système lit un secteur, l'article principal du fichier est affiché. On peut récupérer les fichiers précédemment effacés si l'affectation ultérieure des secteurs n'a pas dégradé données. Une fonction de répertoire étendu permet de faire imprimer le répertoire de disques avec un message d'en-tête, dans un format détaillé qui convient aux listes de gestion interne des disques.

Le programme UTILITAIRE (gestion de fichiers) offre des fonctions de visualisation, de suppression, de changement de nom et de copie sur disques et bandes et à partir de disques et bandes des répertoires, le tout regroupé en un programme unique et extrêmement simple. Les en-têtes de fichiers d'entrée sont toujours affichés et on peut introduire ou éliminer la protection dans toutes les opérations de copiage. Ceci constitue donc la manière la plus facile de gérer TOUS les fichiers AMSDOS standards.

## 1.1 PREPARATION

Le PYRADEV est fourni sur un DISQUE D'exploitation sur lequel on ne doit jamais écrire. Il ne devrait être utilisé que pour réaliser des copies de TRAVAIL à l'aide du programme utilitaire CPM DISCOPY ou DISCKIT3. Consulter le manuel approprié et utiliser le PYRADEV MASTER DISC comme disque émetteur pour créer un disque de TRAVAIL ou disque récepteur.

Après avoir fini de copier, retirer le PYRADEV MASTER DISC de l'unité et le ranger dans un endroit sûr.

## 1.2 MISE EN ROUTE

Placer le disque de TRAVAIL dans l'unité de disque A et vider la machine en appuyant sur CTRL-SHIFT-ESCAPE. Entrer RUN PYRADEV et appuyer sur la grande touche ENTER. Le MENU DU SYSTEME apparaît. La quantité de mémoire disponible affichée en bas et à droite de l'écran.

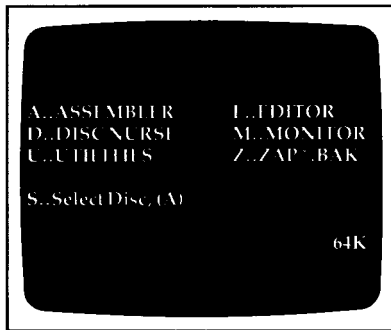


Figure 1.0 menu du système

A. = ASSEMBLEUR	E. = EDITEUR
D. = DISC-NURSE (GESTION)	M. = MONITEUR
U. = PROGRAMMES UTILITAIRES	Z. = ZAP*.BAK
S. = SELECTION DISQUE, (A)	

Les programmes du système se sélectionnent par appui sur la touche unique appropriée «A, E, D, M, U». Ils reviennent tous au menu ci-dessus lorsqu'on sort d'un programme.

L'option «Z» efface tous les fichiers de sauvegarde (\*.BAK) sur le disque implicite en cours d'emploi. N'utiliser cette option qu'avec grand soin.

L'option «S» vous permet de modifier le réglage de l'unité de disques implicite sur (A) ou (B) pour les fichiers données/source. Le disque PYRADEV doit toujours se trouver dans l'unité de disques (A) car il s'attend à y trouver ses programmes, quel que soit le disque implicite. NE PAS utiliser l'option «S» dans les systèmes mono-unité.

### 1.3 NOUVEAUX UTILISATEURS

Lorsque le MENU DU SYSTEME apparaît, appuyer sur «E» pour sélectionner l'Editeur. Entrer le nom PROGRAM.001 comme fichier d'entrée et appuyer deux fois sur la barre d'espace. Le fichier est alors lu en mémoire. Parcourir le fichier à l'aide des touches flèches UP et DOWN (VERS LE HAUT et VERS LE BAS). Le fichier est un petit programme qui contient des quantités de commentaires et d'exemples qui vous expliquent comment utiliser les programmes Editeur et Assembleur.

### 1.4 NOUVEAUX DISQUES

Quand on crée un DISQUE DE TRAVAIL, ce disque peut contenir des fichiers de système superflus. Appuyer sur «U» pour sélectionner le programme Utilitaire, puis appuyer sur «A» ou «B» pour visualiser le répertoire du disque. On peut utiliser la commande «D» (delete) pour éliminer des fichiers. Cette liste décrit les fichiers du système.

PYRADEV.BIN	MENU DU SYSTEME, nécessaire sur -TOUS- les disques de Travail.
PYRAMED.BIN	Editeur page pleine, pour édition du Programme Source.
PYRAMAS.BIN	Macro Assembleur multi-fichiers Z80.
PYRAMON.BIN	Moniteur de Texte et Désassembleur.
PYRADSC.BIN	"Disc-Nurse" pour modification des Secteurs.
PYRAUTL.BIN	Utilitaires fichier pour Copie d'ensemble.
PROGRAM.001	Programme Source de Démonstration pour l'entraînement.
PROGRAM.002	Exemples de Code Assembleur et de Fichier de Test.
PROGRAM.003	Simple programme de copie de fichiers ASCII.

Après avoir supprimé les fichiers non-essentiels, le nouveau disque de Travail est prêt à l'emploi. La suppression de fichiers est également possible dans le programme Editeur.

#### Restrictions Imposées Par Le Système

Le PYRADEV utilise les points d'implantation 40 hexa à 4F hexa pour faire passer des variables dans ses programmes. Veuillez éviter d'utiliser ces zones durant l'élaboration et l'essai d'un programme.

Une fois lancé, le PYRADEV initialise toutes les mémoires ROM de fond en appelant le sous-programme KL.ROM.WALK. Ensuite, le système définit sa propre valeur HIMEM basée sur la valeur HL fournie. Idéalement, seule la DISC.ROM devrait être connectée. Cependant, l'adjonction d'autres mémoires additionnelles (ADD.ONS) ne devrait entraîner de problèmes que si elles utilisent d'importantes zones de la mémoire RAM. La zone de travail de l'Editeur occupe normalement 32.300 octets environ.

Quand vous utilisez le MONITEUR, n'oubliez pas que toutes les ROM de fond raccordées à l'AMSTRAD ont été initialisées. En initialisant deux fois une ROM, on risque d'obtenir des effets indésirables.

## 2.0 EDITEUR DE FICHIER SOURCE

L'Editeur a été conçu pour offrir une élaboration et une édition de programmes extrêmement rapides. On le sélectionne dans le Menu du Système en appuyant sur (E). Une fois chargé, il affiche le répertoire de disques implicite, puis attend que vous entriez les noms de fichiers d'Entrée et de Sortie. L'Editeur revient toujours à l'Ecran Inactif suite à une session d'édition.

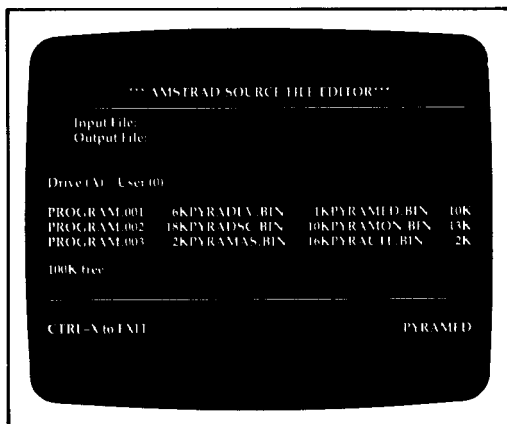


Figure 2.0 Fichier Inactif  
(Voir légende page 1a)

- Pour VISUALISER UN fichier: Entrer simplement un nom de FICHIER D'ENTREE
- Pour CREER un fichier: Entre simplement un nom de FICHIER DE SORTIE
- Pour MODIFIER un fichier: Entrer les noms des fichiers d'ENTREE et de SORTIE
- Pour VISUALISER le répertoire: Appuyer ESPACE ESPACE (c'est-à-dire sans noms). On peut donner au nom du fichier d'entre le préfixe A: ou B: pour déterminer une nouvelle unité de disques implicite.

Figure 2.0

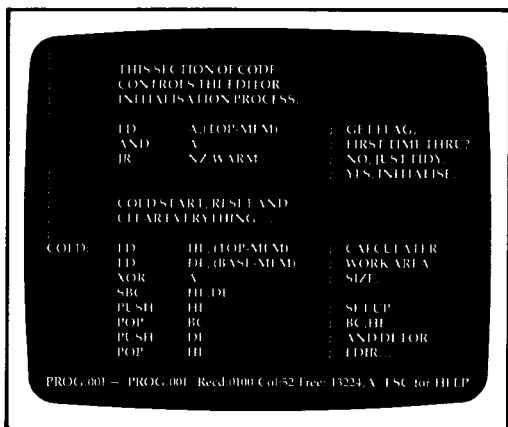
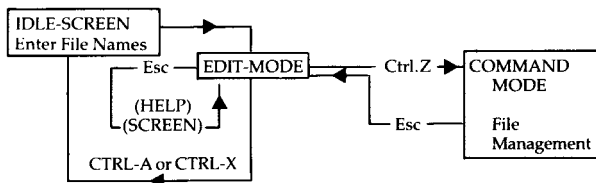
1. EDITEUR DE FICHIER SOURCE
2. Fichier d'entrée
3. Fichier de sortie
4. Unité de disques
5. Utilisateur
6. Libres
7. CTRL-X POUR SORTIR

## 2.1 MODE D'EDITION

Après avoir entré les noms de fichiers, le système lit le fichier d'ENTREE et affiche les 24 premiers enregistrements. Si vous avez sélectionné le mode CREATE (CREER), l'écran demeure vide.

L'emploi de la touche ESCAPE ou de la touche CONTROL-Z fait alterner la Session d'Édition entre le Mode d'Édition, la Page "SOS" (Help-Page) et le Mode de Commande.

1. **ECRAN INACTIF** — IDLE-SCREEN  
Entrer noms fichiers — Enter File Names
2. **Changement** — Esc
3. **MODE EDITION** — EDIT-MODE
4. **MODE COMMANDE**—COMMAND MODE
5. **Gestion de fichiers** — File Management
6. **PAGE "SOS"** — (HELP SCREEN)



La ligne 25 de l'Écran d'Édition affiche toujours les informations d'état:— Les noms des fichiers d'Entrée courants et de Sortie (probables), le numéro d'enregistrement en cours et la colonne curseur, les octets libres et l'Unité de Disques implicite.

### Touche Escape (Changement de Code)

Lorsqu'elle est utilisée en Mode d'Édition, la touche ESCAPE fait afficher l'écran de la Page "SOS". Il s'agit là simplement d'un aide-mémoire qui décrit très brièvement les fonctions qui suivent. En appuyant une seconde fois sur la touche ESCAPE, on fait repasser l'écran au Mode d'Édition.

### Abort & Exit (Suspension & Sortie)

Vous pouvez utiliser la touche (CTRL-A) pour SUSPENDRE la Session d'Édition. En ce cas, tous les fichiers sont abandonnés et aucun changement ne survient au niveau du disque. Il faut entre le caractère (Y) pour confirmer la demande de Suspension. L'Éditeur repasse à l'Écran Inactif et affiche le Répertoire du Disque.



### **Save & Exit (Sauvegarde & Sortie)**

La sortie normale se fait par la touche (CTRL-X). Le nom du fichier de sortie est alors affiché et peut être recouvert selon les besoins. Le fichier sera écrit sur disque et le contrôle repasse alors à l'Ecran Inactif. En cas d'échec d'écriture de fichier, le message **\*\* FILE SAVE FAILED \*\*** apparaît et le message sélectionne le Mode de Commande, ce qui vous permet de sélectionner le disque (c'est-à-dire d'effacer les fichiers superflus) ou d'utiliser un autre disque. Suite à une sauvegarde réussie, l'Editeur repasse à l'Ecran Inactif.

On peut donner au nom du fichier de Sortie le préfixe A: ou B: pour définir une nouvelle unité de disques implicite.

### **Mouvement du Curseur**

Le curseur se contrôle à l'aide des touches de flèches VERS LE HAUT, VERS LE BAS, VERS LA GAUCHE et VERS LA DROITE ou en entrant un texte. le défilement vers l'Avant et l'Arrière est automatique lorsque le système atteint la rangée-1 ou la rangée-24. Le texte est TOUJOURS recouvert par la frappe à l'endroit où se trouve le curseur. Des enregistrements sont ajoutés en tête ou en queue du fichier si l'on déplace le curseur au-delà des premier ou dernier enregistrements à l'aide des flèches vers le haut et vers le bas. La touch ENTER vous envoie toujours à la ligne suivante, colonne un. Les touches TAB et CTRL-TAB déplacement respectivement le curseur jusqu'à la tabulation suivante ou précédente.

### **Vitesse du Curseur**

On peut faire déplacer le curseur alternativement rapidement et lentement, à l'aide de la touche (CTRL-Y). Chaque fois que l'on change de vitesse, la demande (Flash?) apparaît. Répondez (Y) si vous désirez un curseur clignotant ou (N) pour un curseur à éclairage constant.

### **Defilement de l'Ecran**

La flèche de DECALAGE VERS LE BAS et la flèche de DECALAGE VERS LE HAUT font défiler respectivement l'écran d'une ligne vers le haut ou vers le bas, tout en conservant la position due curseur. Les flèches CTRL-VERS-LE-BAS et CTRL-VERS-LE-HAUT font défiler l'écran et vers l'arrière, 24 lignes à la fois. Ces quatre fonctions ne font pas défiler l'écran au-delà du début ou de la fin du fichier.

### **Character-Insert-Delete (Insertion/Effacement de Caractères)**

Les flèches de DECALAGE VERS LA DROITE et de DECALAGE VERS LA GAUCHE (SHIFT-RIGHT/SHIFT-LEFT) assurent ces fonctions. Nous soulignons que la fonction d'insertion place un espace vierge unique dans le texte, espace sur lequel on peut ensuite taper. Les actions d'Insertion et d'Effacement surviennent toutes deux à l'endroit où se trouve le curseur. Les touches (CLR) et (DEL) permettent également d'effacer des caractères ou d'effacer/inverser (comme en BASIC).

### **Line-Insert-Delete (Insertion/Effacement de Lignes)**

Les flèches CTRL-DROITE et CTRL-GAUCHE (CTRL-RIGHT/CTRL-LEFT) assurent ces fonctions; elles agissent toutes deux sur la ligne en cours,, ligne qu'indique le curseur. L'insertion de ligne introduit une ligne vierge dans le texte, ligne sur laquelle on peut taper. Lorsqu'on tape le texte de la fin de la ligne, le système insère une autre ligne vierge. La même chose se produit si vous appuyez sur la touche ENTER. Pour annuler le Mode d'Insertion de Lignes, utilisez la touche CTRL-LEFT-ARROW (pour effacer une ligne vierge nouvellement créée) ou faites défiler au-delà de la rangée 24.

### **Open Space (Ouverture d'un Espace)**

Vous pouvez utiliser la touche (CTRL-O) pour ouvrir ou diviser une ligne à l'endroit où se trouve le curseur, et constituer ainsi deux nouvelles lignes. La première contiendra le texte qui figurait jusqu'à la position du curseur, et la seconde, le texte qui figurait à partir de la position du curseur. Cette touche est utile lorsque l'on veut insérer un nouveau code sur une ligne qui contient déjà un label.

### **Block-Copy (Copie de Blocs)**

Pour copier un bloc de code, amenez le curseur à la première ligne du code et appuyez sur (CTRL-B) pour définir le marqueur BEGIN (DEBUT). Amenez le curseur jusqu'à la dernière ligne du code et appuyez sur (CTRL-E) pour définir le marqueur END (FIN). Le message 'Block Saved' apparaît. Placez le curseur à l'endroit où vous voulez copier le bloc et appuyez sur la touche (COPY). Le bloc sera COPIE-INSERE sur la ligne suivante. Le bloc sauvegardé demeurera en mémoire jusqu'à ce que l'on marque un autre bloc. On peut le copier à plusieurs reprises dans n'importe quel point du fichier, sur un autre fichier ou sur un autre disque.

### **Block-Delete (Effacement de Blocs)**

Marquer le Bloc à l'aide des touches (CTRL-B) et (CTRL-E) comme pour la fonction de Copie de Blocs. Lorsque le message 'Block Saved' apparaît, appuyez sur la touche (CTRL-D). Ceci fait afficher le bloc en ENCREs inverses. Appuyez sur (Y) pour autoriser l'opération d'effacement. Précisons que, suite à un effacement, le bloc demeure sauvegardé et peut être recopié, si nécessaire, au moyen de la touche (COPY).

### **Block-Move (Mouvement de Blocs)**

Constitué par la fonction EFFACEMENT DE BLOCS, suivi de l'appui sur la touche (COPY), comme ci-dessus.

### **Tab-Stops (Taquets de Tabulation)**

Un appui sur (CTRL-T) fait afficher les taquets de tabulation en cours d'utilisation sous forme de petits triangles sur la ligne 24. Les tabulateurs sont posés ou supprimés en modifiant la position du curseur et en appuyant sur la touche (TAB) jusqu'à ce que le symbole triangulaire apparaisse ou disparaisse. Un appui sur «CTRL-TAB» supprime tous les taquets de tabulation. Lorsque vos nouveaux taquets sont prêts, appuyez sur la grande touche ENTER pour les enregistrer. Les tabulations peuvent être sauvegardées pour emploi ultérieur – voir Mode de Commande.

### **Mark-Find-Line (Marquage-Recherche de Lignes)**

On peut marquer une seule LIGNE en utilisant la touche CTRL-L. On peut ensuite la ramener sur l'écran à l'aide de la touche CTRL-F (FIND). Ces deux opérations sont utiles en ce qu'elles permettent de revenir rapidement au code entré après avoir étudié un code qui se trouve dans une autre partie du fichier.

### **View-Begin-End (Debut-Fin-Visualisation)**

Appuyez sur CTRL-V pour sélectionner 'View' puis appuyez sur (B) pour voir le début du fichier (enregistrement-1) ou (E) pour voir la fin du fichier (dernier enregistrement).

### **Go To Record (Passage à un Enregistrement)**

Un appui sur CTRL-G permet d'entre un numéro d'enregistrement. L'Editeur passe immédiatement à cet enregistrement. L'emploi d'un zéro ou d'un chiffre supérieur à celui du dernier enregistrement font respectivement afficher le Début ou la Fin du fichier. La fonction CTRL-G est utile pour retrouver un code programme dans le Listage de l'Assembleur.

### **New-Input (Nouvelle Entree)**

L'emploi de la fonction CTRL-N permet d'ajouter un autre fichier INPUT (ENTREE) au fichier qui se trouve actuellement dans la zone de travail de la mémoire. Il est ainsi possible d'élaborer de nouveaux programmes ou documents à partir d'un code ou d'un texte précédent. Après adjonction du nouveau fichier dans la mémoire, les fonctions Block Delete (Effacement Bloc) et Move (Mouvement) peuvent rapidement amener le code ou le texte à l'endroit nécessaire.

Cette fonction ne peut s'utiliser que si l'on dispose encore au minimum de 1028 octets.

### **Search-System (Systeme de Recherche)**

CTRL-S sélectionne le système de recherche et offre deux types d'opérations.

#### *Search and Stop (Recherche et Arrêt)*

Entrez une chaîne de recherche entre apostrophes ou guillemets, puis appuyez deux fois sur la touche ENTER, en laissant vide la chaîne de remplacement. La recherche débute. Chaque fois que la chaîne est retrouvée, l'enregistrement qui la contient est affiché, accompagné des enregistrements précédent et suivant, dans sa propre 'fenêtre'. Vous pouvez ensuite utiliser les options suivantes:

- Touche Enter... Ignorer cette correspondance et continuer à chercher.
- Touche Escape... Terminer et revenir au texte initial.
- Touche (G)..... Passer à l'enregistrements dans lesquels la concordance a été trouvée.
- Up/Down  
(Vers le haut/bas). Faire défiler les enregistrements dans lesquels la concordance a été trouvée.

La fonction Recherche et Arrêt permet d'obtenir rapidement la position du code programme à l'aide de labels de code source qui sont (généralement) uniques.

#### *Search and Replace (Recherche et Remplacement)*

Entrez une chaîne de recherche entre apostrophes ou guillemets. Entrez ensuite une chaîne de remplacement également entre apostrophes ou guillemets. N'oubliez pas d'utiliser selon les besoins, les majuscules ou les minuscules selon la chaîne de remplacement. Après avoir entré la chaîne de remplacement, répondez à la question concernant la sélection d'un remplacement Automatique ou Conditionnel: la recherche débute.

Si vous avez choisi l'option de Remplacement Automatique, l'opération se poursuit jusqu'à la fin sans aucune autre intervention. Si vous avez utilisé l'option Conditionnelle, chaque fois que la chaîne est trouvée, elle sera affichée, accompagnée des enregistrements précédent et suivant. Appuyez sur (R) pour effectuer le REMPLACEMENT, ou sur (C) pour CONTINUER, sans autre intervention.

Toutes les fonctions de recherche fonctionnent à partir de la ligne en cours jusqu'à la fin de fichier. Utilisez CTRL-V B pour sélectionner l'enregistrement 1 si vous désirez examiner le fichier complet. Une bonne séquence de frappe à utiliser avant l'emploi de la fonction de recherche est CTRL-L CTRL-V B. Ceci repère la ligne en cours et assure le passage au début du fichier. Après l'emploi de la fonction de recherche, la frappe CTRL-F renvoie à la ligne repérée ou marquée.

Vous pouvez annuler n'importe quelle recherche avec la touche ESCAPE.

Note: Il surviendra une troncature si une chaîne de remplacement est plus longue que la chaîne source et si les caractères sont forcés au-delà de la colonne 80.

### **Keystroke-Memories (Mémoires de Frappes)**

L'Éditeur contient neuf mémoires de frappes qui peuvent chacune enregistrer 32 frappes. Vous pouvez les utiliser pour enregistrer des touches de commande ou de texte. Vous pouvez ensuite les "relire" selon les besoins.

#### *Visualisation*

Pour visualiser le contenu des mémoires de frappes, appuyez sur la touche 0 (f0) du bloc numérique. Les mémoires seront affichées. Appuyez sur la touche ESCAPE pour revenir en Mode d'Édition.

#### *Record (Enregistrement)*

Appuyez sur la touche SMALL-ENTER et sélectionnez une mémoire à l'aide des touches 1 à 9 du bloc numérique. Après sélection, le message 'Keystroke Recording ON' sera affiché. Pour terminer l'enregistrement, appuyez à nouveau sur la petite touche Enter. Si un enregistrement excède la limite des 32 touches, la mémoire suivante sera raccordée et utilisée. Si vous atteignez la fin de la mémoire de frappes neuf, l'enregistrement cesse et le message KS-sequence-saved (séquence mémoire de frappes sauvegardée) est affiché.

#### *Replay (Relecture)*

Pour relire une séquence enregistrée, appuyez sur la touche du bloc numérique que vous avez utilisée pour faire l'enregistrement (1 à 9). Les touches sont reprises et utilisées comme si vous veniez de les entrer au clavier. Pour faire cesser la relecture, appuyez sur la touche ESCAPE.

#### *Erase (Effacement)*

Appuyez sur CTRL-SMALL-ENTER et sélectionnez une mémoire à l'aide des touches 1 à 9 du bloc numérique. La mémoire sera vidée et vous allez pouvoir la réutiliser.

Note 1: Les fonctions de mémoires ne peuvent être enregistrées ou initiées pour lecture qu'en Mode d'Édition normal. Les séquences enregistrées peuvent, par exemple, sélectionner le système de Recherche et lancer une recherche de chaîne de caractères, puis revenir en Mode Edition. Il est cependant impossible de lancer un enregistrement ou une relecture de mémoire de frappes si l'on a déjà sélectionné manuellement le système de Recherche.

Note 2: Vous pouvez sauvegarder les mémoires de frappes. Voir Mode de Commande.

## Printing (Impression)

En appuyant sur la touche du point décimal (virgule en Français) du bloc numérique, on fait commencer l'impression à partir de la ligne en cours. En appuyant deux fois sur cette touche, on fait cesser les changements de lignes déterminés par les retours de curseur. L'impression va continuer jusqu'à la fin du fichier. Un appui sur la touche ESCAPE fait cesser l'impression, que vous pouvez ensuite poursuivre avec n'importe quelle touche. Deux appuis sur la touche ESCAPE annulent l'opération d'impression.

On peut inclure des codes d'impression spéciaux dans le texte en cours d'impression, afin de contrôler l'imprimante de la manière suivante...

Pendant l'impression, le symbole flèche vers le haut (qui se trouve sur la touche du signe livre Sterling) signale que le caractère suivant est un symbole de contrôle. La valeur 33 est soustraite à la valeur ASCII du caractère. Si elle est négative, le caractère est ignoré sans quoi il est envoyé à l'imprimante. Un symbole flèche vers le haut à la fin d'une ligne fait sauter le caractère CR invisible, ce qui permet d'obtenir une longue ligne d'impression à partir de deux lignes sur l'écran.

Exemples:

⊠«E représente ESCAPE-E,  
⊠0 représente code 15,

⊠«S représente ESCAPE-S,  
⊠/ représente code 14.

## Command Mode (Mode de Commande)

Vous sélectionnez le Mode de Commande avec la touche CTRL-Z quand vous êtes en Mode d'Édition. La Session d'Édition est provisoirement suspendue, tandis que les sous-programmes de Gestion de Fichiers et d'Options deviennent disponibles. Pour sortir du Mode Commande, utilisez la touche Escape.

Un petit menu est affiché en bas de l'écran; vous sélectionnez des fonctions en appuyant sur les touches appropriées, comme suit:—

A,B: Selection Disque	C: Compression/condensation
D,R: Effacer, Rebaptiser	Z: Effacement fichiers
L,S: Options de charge	sauvegarde Zap
J,K: Enres (papier/plume)	S: Sauvegarde Options

*A,B: Sélection Disque A ou B*

Un appui sur (A) ou sur (B) fait commuter l'unité de disques implicite et affiche le répertoire, les noms de fichiers et l'espace libre du nouveau disque.

*C: Compression (Condensation) Operante/Inoperante*

Si vous êtes en train d'éditer et d'Assembler un code dans le système PYRADEV, vous devez maintenir cette option OPERANTE. L'Éditeur et Assembleur vont en effet utiliser la compression (ou condensation) des données de manière à ce que vos fichiers demeurent aussi courts que possible. Si vous employez l'Éditeur pour éditer des fichiers en BASIC avant d'utiliser CTRL-X pour ré-écrire le fichier sur le disque, vous devez rendre la compression INOPERANTE. Le fichier sera alors ramené à sa longueur normale et peut ensuite être utilisé par le système BASIC.

#### *D: Delete File (Effacement Fichier)*

Cette fonction est analogue à la directive 'ERA' BASIC/CPM. Elle peut s'utiliser pour effacer un seul fichier ou un groupe de fichiers. On peut sélectionner une unité de disques spécifiée en utilisant le préfixe A: ou B:.

Exemples:	TEMPFILE.001	- effacer le fichier
	TEMP.*	- effacer tous les fichiers TEMP.
	TEMP????.*	effacer les noms de fichiers débutant par TEMP.
	B:TEMP.FIL	- effacer TEMP.FIL de l'unité B.

#### *R: Rename File (Rebaptiser Fichier)*

On peut remplacer le nom d'un ANCIEN fichier par celui d'un NOUVEAU fichier si le nouveau fichier n'existe pas encore.

#### *J: Jump Paper Ink (saut de couleur de fond)*

Chaque fois que vous appuyez sur la touche (J), l'Encre à Papier (= le fond) passe à sa valeur suivante. Un appui sur (CTRL-J) exerce l'effet contraire.

#### *K: Jump Pen Ink (saut de couleur écriture/graphismes)*

Chaque fois que vous appuyez sur la touche (K), l'Encre (= couleur écrit.graph.) passe à sa valeur suivante. Un appui sur (CTRL-K) exerce l'effet contraire.

#### *S: Save User Options (Sauvegarde des Options Utilisateur)*

Un appui sur (S) fait écrire les Tabulations, les valeurs Papier et Encre et les Mémoires de Frappes en cours d'utilisation dans un fichier sur disque pour emploi ultérieur. On peut entrer un nom de fichier quelconque. L'extension de fichier (suffixe) suggérée est -OPT. On peut utiliser plusieurs Fichiers d'Options différents pour définir divers 'environnements'.

#### *L: Load User Options (Charger Options Utilisateur)*

L'emploi de la touche (L) permet de créer un fichier d'"Options" avec la commande (S)ave pour remise en oeuvre dans le cadre de la Session d'Edition en cours. Après lecture du fichier, les nouveaux Tabulateurs, ainsi que les nouvelles Encres et Mémoires de Frappes seront opérants.

#### *Z: Zap \*.BAK Files (fichiers Zap \*.BAK)*

Efface tous les fichiers de sauvegarde du répertoire du disque implicite. Cette fonction est identique à l'emploi de la fonction (D)elete mais avec le nom de fichier \*.BAK.

### **3.0 MACRO ASSEMBLER (MACRO ASSEMBLEUR)**

Le Macro Assembleur PYRADEV accepte le langage de programmation ZILOG Z80 (avec quelques modifications et extensions) et peut traiter un ou plusieurs fichiers de programme source pour créer un fichier de chargement binaire, directement sur un disque. On le sélectionne à partir du MENU DU SYSTEME en appuyant sur (A). Après chargement, il affiche le répertoire du disque implicite et attend que les diverses RUN OPTIONS (OPTIONS D'EXECUTION) soient entrées....

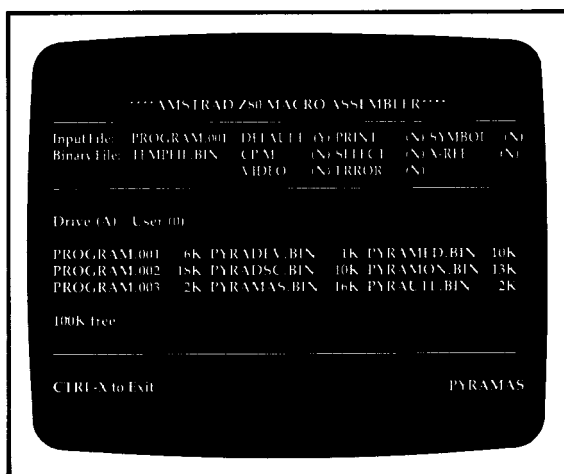


Figure 3.0 Options d'Exécution de l'Assembleur

**INPUT FILE** = FICHIER D'ENTREE  
**BINARY FILE** = FICHIER BINAIRE  
**DEFAULT** = IMPLICITE  
**PRINT** = IMPRESSION

**DRIVE** = UNIT DE DISQUES  
**USER** = UTILISATEUR  
**100 FREE** = 100 LIBRES

### Input-File (Fichier d'Entrée)

Nomme le premier fichier source de la chaîne d'entrée. Il peut s'agir d'un programme complet ou du premier d'un grand nombre de fichiers qui vont être regroupés pour constituer le programme de chargement binaire. On peut utiliser des instructions SELECT intégrées dans les fichiers de code source pour ajouter d'autres fichiers au train d'entrée. L'un des emplois du système SELECT consiste à n'entrer dans le premier fichier que des instructions SELECT. Un bon suffixe à donner à ce fichier est .CTL car ceci représente un fichier d'ORDRES (CONTROL file). Un nom de fichier binaire implicite avec suffixe .BIN est créé à partir du nom de fichier d'entrée. L'inclusion d'une déclaration END (FIN) dans le train d'entrée est optionnelle mais, s'il en existe une, elle termine immédiatement le train d'entrée, même s'il existe d'autres fichiers.

### Binary-File (Fichier Binaire)

Ceci est le nom qui doit être utilisé lorsqu'on catalogue sur disque le fichier à code binaire. Ce fichier est établi à partir du nom du fichier d'entrée (par adjonction d'un suffixe .BIN) et peut être recouvert. Un appui sur la touche ENTRE le laisse tel quel. Si vous n'avez pas besoin d'un fichier binaire, remplissez cette zone de blancs.

### Defaults (Valeurs Implicites)

Si vous appuyez sur la touche Enter ou sur (Y), l'Assembleur va utiliser les valeurs implicites affichées et lancer le processus d'assemblage. Si vous désirez modifier l'une quelconque des options d'exécution, appuyez sur (N) pour cette première option.

### CP/M-Output (Sortie CP/M)

Si cette option est validée (Y), le fichier binaire est écrit sur le disque sans en-tête de fichier et convient à l'emploi comme programme CP/M transitoire. Aucun autre contrôle n'est effectué pour garantir que le programme est conforme aux normes CP/M.

### Video-Display (Affichage Video)

Si validé (Y), l'enregistrement source en cours de traitement sera affiché pendant le passage 1. Durant le passage 2, c'est le code objet et l'enregistrement source qui seront affichés.

### Print-Output (Sortie Impression)

Lorsqu'elle est validée (Y), cette option ordonne à l'Assembleur de produire une liste d'assemblage durant la génération de code objet du passage 2. Cette option fonctionne en conjonction avec l'option SELECT...

### Selective Print (Impression Selective)

On peut répondre (Y) (OUI) ou (N) (NON) à l'option SELECT, qui ordonne à l'Assembleur de respecter (YES) ou d'ignorer (NO) les directives LIST et NOLIST contenues dans le code source.

### List Errors (Erreurs de Listage)

Lorsqu'elle est validée (Y), cette option force les messages d'ERREURS à être listés sur l'imprimante pour emploi ultérieur lors de la correction des erreurs de codage à l'aide de l'EDITEUR. Elle fonctionne indépendamment des options PRINT et SELECT décrites ci-avant, mais les erreurs sont toujours listées si on a répondu (Y) à l'option d'impression (PRINT).

### Recapitulatif des Options d'Impression

PRINT	SELECTIVE	ERRORS	Printed Code	Printed Errors
NO	NO	NO	NONE	NONE
NO	NO	YES	NONE	ALL
NO	YES	NO	NONE	NONE
NO	YES	YES	NONE	ALL
YES	NO	NO	ALL	ALL
YES	NO	YES	ALL	ALL
YES	YES	NO	SELECTIVE	ALL
YES	YES	YES	SELECTIVE	ALL

Figure 3.1 Options d'impression

**PRINT** = IMPRESSION  
**SELECTIVE** = SELECTIVE  
**ERRORS** = ERREURS  
**Printed Code** = Code Imprimé

**Printed Errors** = Erreurs Imprimées  
**NONE** = NEANT  
**ALL** = TOUS/TOUTES



**Avertissement:** L'emploi des options VIDEO ou PRINT ralentit APPRECIABLEMENT le processus d'Assemblage. Pour obtenir les résultats les plus rapides, répondez (Y) aux options implicites. L'exception: faites lister les ERREURS, ce que vous trouverez très utile durant les premiers Assemblages de grands programmes.

### **Symbol-Table (Table des Symboles)**

Si elle est validée (Y), cette option produit une liste des symboles (labels) utilisés dans le ou les programme(s) source(s) accompagnés de leurs valeurs. La sortie se fera sur ECRAN ou IMPRIMANTE en fonction des options sélectionnées ci-avant. On peut faire précéder chaque symbole d'un (U), d'un (D) ou d'un (N), qui signifient, respectivement, non défini, Double Définition ou Sans Référence.

### **Cross-Reference (Correspondances)**

Si cette option est validée (Y), une table de correspondances sera sortie sur Ecran ou Imprimante une fois le processus d'assemblage terminé. Le listage affichera tous les fichiers sources lus, et les numérotera comme fichiers, 1,2,3, etc. La table des symboles est imprimée, avec tous les numéros de fichiers et les numéros de lignes des instructions qui les désignent.

Vous êtes prié de noter que, si elles sont utilisées, les options SYMBOL ou X-REF attendront que les options VIDEO ou PRINT (ECRAN ou IMPRIMANTE) aient été validées avant que le processus d'assemblage puisse sortir les informations demandées.

### **Touches de Durée d'Exécution de l'Assembleur**

Une fois lancé, le processus d'Assemblage peut se contrôler par commande au clavier des OPTIONS D'EXECUTION décrites ci-avant. Pour ce faire, il vous suffit d'appuyer sur des touches uniques. L'affichage des options d'exécution change lorsque vous utilisez les touches.

Vous utiliserez les touches suivantes:

- (P) Inversion de l'option IMPRESSION.
- (S) Inversion de l'option SELECTION.
- (E) Inversion de l'option ERREURS.
- (V) Inversion de l'option VIDEO (ECRAN).

On peut également utiliser la touche Escape pour INTERROMPRE MOMENTANEMENT le programme d'Assemblage, qui est utile pour examiner le code objet affiché sur l'écran. Appuyez sur n'importe quelle touche pour reprendre le processus d'assemblage.

Un appui sur la touche CTRL-A met un terme au processus d'assemblage. Tous les fichiers sont abandonnés et aucun changement de disque ne survient. L'Assembleur vous demande d'appuyer sur n'importe quelle touche afin qu'il puisse repasser la commande au Menu du Système...

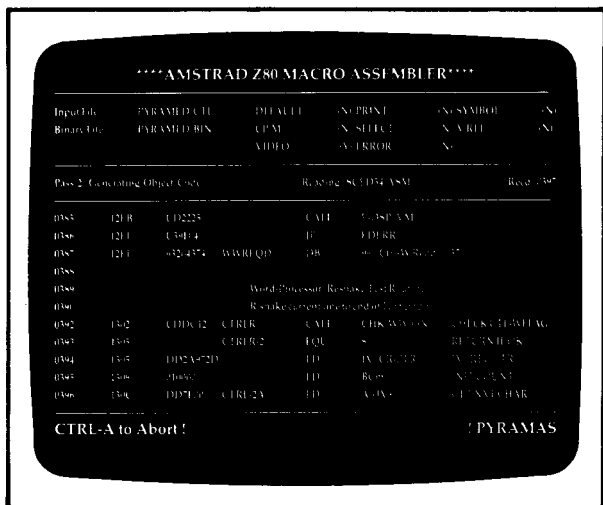


Figure 3.1 Durée d'exécution (VIDEO-ON)

**Input File** = Fichier d'Entrée

**Binary File** = Fichier Binaire

**Pass 2 : Generating Object Code** = Passage 2 : Génération du Code Objet

**Reading** = Lecture

**Recd** = Enreg.

**Word-Processor, Resnake Text Routine** = TdT, Déplacer Sous-Programme Texte

**Resnake current line to end of Paragraph** = Déplacer ligne courante à la fin du Para.

**CTRL-A to Abort!** = CTRL-A : Abandon

### Traitement des Erreurs d'Assemblage

La détection des erreurs par le processus d'assemblage sera traitée de trois manières différentes, en fonction des options de sortie définies plus haut.

a) Si l'on a validé (Y)es l'une quelconque des options PRINT ou ERROR, l'Assembleur suppose que vous avez connecté une imprimante. Le code d'erreur et la description seront imprimés. Le processus d'assemblage ne sera PAS interrompu.

b) Si vous avez validé (Y)es l'option VIDEO, le code d'erreur et la description seront affichés. Le processeur d'assemblage cesse momentanément, mais vous pouvez le reprendre en appuyant sur une touche quelconque.

c) Si vous avez inhibé (N)o les options PRINT, ERROR et VIDEO, le processus d'assemblage fonctionne à sa vitesse maximale. Les erreurs seront envoyées sur l'Ecran Vidéo, mais le processus d'assemblage ne s'interrompra qu'après affichage de dix messages d'erreurs. En ce cas, appuyez sur une touche quelconque pour faire poursuivre l'assemblage.

#### Codes d'erreurs

B - Erreur de Branchement/Saut	N - Expression Numérique
C - Erreur Conditionnelle	OE - Erreur d'Opérande(s)
D - Double Définition	OM - Opérande Manquant
F - Erreur de Fichier	GF - Trop de SELECTIONS
G - Erreur d'Usage Drg	U - Label Non-défini
L - Label (syntaxe)	X - Code Op. supplémentaire invalide
MP - Macro Paramètre	FW - Référence aval invalide

#### Definition du Langage Assembleur

Le Langage Assembleur est le langage Z80 et les programmes sont écrits sous forme d'un ou de plusieurs Fichiers Sources qui sont assemblés en code machine par l'Assembleur.



Chaque fichier source se compose d'instructions de source, une par ligne (rangée sur l'écran). L'EDITEUR PYRADEV est le programme idéal pour la création et l'édition de ces fichiers sources. Chaque instruction de fichier source se compose d'un label optionnel, d'un code opération, d'opérandes optionnels et de commentaires optionnels.

LABEL	CODE OP	OPERANDES	COMMENTAIRES
LABEL-1	LD	HL,(VALUE)	; load HL
	LD	(OLD-VALUE),HL	; save HL
	JP	LABEL-X	; jump ...
;			
VALUE:	DEFW	00	; current value
OLD-VALUE	DEFW	00	; old value

**load** = charger      **save** = sauvegarder      **jump** = sauter  
**current value** = valeur en cours      **old value** = valeur ancienne

**LABELS:** Ce labels doivent débuter dans la colonne un et peuvent avoir n'importe quelle longueur; on les maintient généralement à une longueur inférieure à 10 caractères. L'utilisation des deux points (:) après le label est optionnelle.

**CODES OP :** Ces codes peuvent se trouver à n'importe quel endroit sur une ligne, et doivent être précédés d'un espace au minimum.

**OPERANDES :** Un code opération doit être suivi d'un espace au minimum avant que l'on puisse entrer l'opérande. la zone de l'opérande ne doit pas contenir d'espaces vierges imbriqués.

**COMMENTAIRES :** Les opérandes doivent être suivis d'un espace au minimum avant qu'on puisse entrer un commentaire. L'emploi d'un point virgule (;) avant le commentaire est optionnel, sauf sur les instructions RET. utiliser un point virgule si le commentaire débute à la colonne un.

### Expressions

Une expression est un OPERANDE qui se compose d'une ou de plusieurs variables, de labels et de constantes que l'Assembleur doit évaluer pour produire une valeur entière de 16 bits. Une expression s'évalue de GAUCHE à DROITE et il est interdit d'utiliser les parenthèses. On peut utiliser les opérateurs suivants...

+ - \* / .MOD. .SHR. .SHL. .AND. .OR. .XOR. .EQ. .GT. .LT. .UGT. .ULT.

Ils représentent respectivement : plus, moins, multiplier, diviser, décalage à droite, décalage à gauche, ET, OU, OU-exclusif, Egale, plus grand que, plus petit que, plus grand-sans signe, plus petit-sans signe.,

On peut utiliser le symbole du dollar (\$) pour représenter la valeur du compteur de programme pendant l'assemblage. Par exemple: JP \$+3 générerait un branchement à l'instruction suivante (un JP égale 3 octets).

L'Assembleur accepte la notation numérique pour les expressions binaires, octales, décimales et hexadécimales, dans les formats suivants...

Binaire :	1011100B	ou	%1011100	
Octal :	134O	ou	@134	(ou 134O)
Décimal :	134D	ou	134	
Hexa :	5CH	ou	#5C	(doit débuter numériquement) (c-à-d 0FFH pour 255D)

### Directives de l'Assembleur

Ces directives sont écrites comme des instructions (codes opérations et opérandes) mais sont des commandes adressées à l'Assembleur et sont "exécutées" au moment de l'assemblage. Elles régissent les options de listage d'assemblage, la génération de code, et la construction du fichier binaire (en code machine).

### Définition des Variables et Stockage

On peut définir des Chaînes d'Octets, de Mot et de Caractères, à l'aide des directives précisées dans l'exemple suivant.

	DBXON		; Expansion DB.
LABEL1:	DB	"This is a string"	; Chaînes de 16 octets.
LABEL2:	DEFB	"This is a string"	;
BYTE:	DB	"B"	; Un seul octet
MIXTURE:	DB	1,"A",2,"b"	; Chaîne mixte
SSTRING:	DB	"SPECIAL"+80H	; Le bit 7 de l'est en fonction

```

;
LABEL3: DW 256*2 ; Un mot peut contenir un
LABEL4: DEFW 512 ; nombre constant de 16 bits.
;
LABEL5: DB 255 ; La valeur maximale du multiplét unique
LABEL6: DEFB 0FFH ; unique est 255 (chiffre décimal).
;
VALUE7: EQU 1000 ; La VALEUR7 égale 1000
LABEL8: DW VALUE7 ; Le LABEL8 contient 1000
VALUE7: DEFL 1001 ; La valeur7 a été redéfinie à 1001
LABEL9: DEFW VALUE7 ; Le LABEL9 contient 1001
DBXOFF

```

Les DBXON et DBXOFF font respectivement lister les chaînes DB sous format complet ou abrégé (les quatre premiers multiplés seulement).

Les Fichiers Sources PROGRAM.001, PROGRAM.002 et PROGRAM.003 due DISQUE PRINCIPAL contiennent d'autres exemples des directives de l'assembleur. On peut visualiser et/ou imprimer les fichiers à l'aide de l'Editeur PYRADEV.

### **Eject (Ejection)**

Ceci ordonne au processus d'impression d'assemblage de passer au haut de la page suivante. La page implicite est normalement une page de 80 caractères de largeur, et de 70 lignes de haut, ce qui correspond au format A4 standard. On peut modifier ces paramètres à l'aide de directives PRINTR, expliquées ci-dessous.

### **End (Fin)**

Cette instruction signale la FIN DE L'ENTREE. L'Assembleur traite ceci comme une fin de fichier physique, même si le code de source suit l'instruction. L'emploi d'une instruction END est optionnel.

### **Endbin**

Termine la génération du code binaire. S'utilise généralement à la fin d'un programme, avant les instructions DEFS ou DS pour ne pas trop allonger le fichier, mais peut s'utiliser n'importe où. Son contraire est l'instruction ORG qui fait reprendre la génération du code ou l'instruction LOAD qui "bourre" et fait reprendre la génération du code.

```

;
; Cet exemple de codage montre comment ; Cet exemple de codage
; réserver une zone en tête d'un programme. ; représente comment utiliser
; ; "endbin" à la fin d'un programme.
;
;
;
;
; ORG 5Q/H POP BC
ENDBIN POP DE
; RET
BUFF-1 DEFS 2048 ;
BUFF-2 DEFS 2048 ENDBIN
;
TRUE-START ;
; ORG $ BUFF1: DB 2048
LD (SAVE),SP BUFF2: DB 2048
CALL INIT-PGM ;
JP GO-GAME ;
; END
;
ETC....

```

## Equ

La directive EQUATES (EGALE) : associe un LABEL à une expression, par exemple, MINUS1 EQU-1

## Exec

Cette directive définit l'adresse à utiliser dans l'en-tête du fichier binaire. Lorsque le programme assemblé est EXECUTE, il va être chargé selon l'instruction LOAD (ou ORG). AMSDOS fait ensuite passer la commande à l'adresse EXEC. Si cette directive n'a pas été fournie, l'adresse EXEC est réglée sur la première (vraie) ORG ou LOAD.

## Extra

Permet à l'Assembleur de traiter les instructions Zilog NON-STANDARDS additionnelles. Voir page 14.

## Free (Libre)

Permet à l'Assembleur de traiter des expressions en "Format Libre". Autorise le mélange de différents types d'expressions de stockage. Voir page 15.

## List et Nolist

Met l'Impression Sélective EN FONCTION ou HORS FONCTION.

## Load (Charger)

Cette instruction ordonne à l'Assembleur de générer un code de zéros binaires jusqu'à ce que le Compteur du Programme atteigne l'expression de valeur LOAD, après quoi le système reprend la génération de code normale. Cette instruction a pour but de forcer des sections de code à leurs positions appropriées dans le fichier binaire, afin que, lorsqu'on charge ce fichier, le code se trouve à son adresse correcte.

Si l'adresse n'est pas fournie, l'adresse LOAD choisit implicitement la première adresse ORG vraie. Il faut donc fournir l'une de ces adresse (le plus souvent ORG) avant d'obtenir une génération de code. La première occurrence d'une directive ORG ou LOAD s'utilise dans l'en-tête de fichier binaire (sauf si elle est immédiatement suivie d'une instruction ENDBIN). Les autres ne s'utilisent que pour contrôler la génération du code et la structure du fichier binaire.

## Org

Cette instruction ordonne à l'Assembleur de POSITIONNER le Compteur de Programme sur l'expression opérande. Au contraire de l'instruction Load (Charger), aucun code de remplissage n'est généré. L'emploi de l'instruction ORG permet de resserrer ensemble dans le fichier de disque en cours de génération des section de code qui, dans la mémoire, seront très écartées. Il incombe au programmeur de veiller à ce que ces sections de code soient ramenées à leur position correcte avant leur exécution. Il est NORMAL que l'instruction d'un jeu de programme en course d'assemblage soit une directive ORG.

```
ORG 100H      . Cette section implante
code          . correctement en mémoire,
code          . à partir du point 100 hexa.
LOAD 200H    .
code         .
code         .
code         .

ORG          $+100      . Il faudra ramener ce bloc
code         . à son adresse vraie, étant
code         . donné qu'aucun code de
               . remplissage n'a été généré.

END
```

## Pause

Cette instruction fait "pauser et afficher" le message. Ceci survient lorsque la FIN du fichier d'entrée en course est atteinte. Elle a pour objet de permettre l'emploi de plusieurs disques quand on assemble des fichiers sources multiples. Elle ne doit être utilisée qu'avec les systèmes à DEUX unités étant donné que le fichier BINAIRE doit être écrit continuellement sur un seul disque. Après changement de disque, appuyez sur la barre d'espacement pour pouvoir continuer.

PAUSE 'MOUNT NEXT SOURCE DISC IN DRIVE B' (avec assumption que la sortie se fait sur l'unité A).

## Print

Cette instruction peut s'utiliser pour définir la largeur de Papier, les Lignes par page (hauteur de format), les Pauses de page et la suppression d'Avance de ligne. L'option de Pause de page fait pauser l'impression en haut de chaque page pour permettre de positionner le papier. C'est un besoin très REEL lorsqu'on et rendent difficile la production des Listages d'Assembleur. L'Avance de ligne élimine l'envoi des avances de ligne correspondant à chaque retour de curseur, étant donné que certaines imprimantes le font automatiquement.

PRINTR W80,D70	Largeur 80 car., Hauteur 70 lignes, Pause-P Inopérante, Avance de ligne.
PRINTR W132,D66,P	Largeur 132 car., Hauteur 66 lignes, Pause-P Opérante, Avance de Ligne.
PRINTR N	Utilise les options implicites (W80,D70) et élimine les Avances de ligne.
PRINTR P	Utilise les options implicites (W80,D70) et utilise les Pauses de page.

## Select

Cette directive est très puissante. Quand on l'utilise, le Fichier Source nommé est AJOUTE (et non INCLUS) à la FIN du train d'entrée en cours. On peut CHAINER de cette manière jusqu'à 32 fichiers et les instructions SELECT peuvent apparaître n'importe où. Lorsqu'on assemble de grands programmes, il est possible de lancer le processus d'assemblage avec un petit fichier de commande qui ne fait rien d'autre que SELECTIONNER des fichiers pour le train d'ENTREE....

ORG	100H	; Origine du programme
SELECT	PROG1.ASM	;
SELECT	PROG2.ASM	; Fichiers de l'unité de disques A
SELECT	PROG3.ASM	;
SELECT	B:PROG1.ASM	;;
SELECT	B:PROG1.ASM	;; Fichiers de l'unité de disques B
SELECT	B:PROG1.ASM	;;

Dans les grands projets de programmation, les informations suivantes s'avèreront utiles pourvu que vous utilisiez deux unités de disques.

## UNITE A

Disque 1  
Face A

PYRADEV SYSTEM  
+ Fichiers binaires  
+ Fichiers sources

Disque 1  
Face B

Réserve, suggérons  
emploi PYRA-WORD pour  
documenter le projet.

Le Disque 1 demeure  
fix durant  
la réalisation

## UNITE B

Disque 2  
Face A

CODE SOURCE - 178K,  
accès par Editeur et  
Système d'Assemblage.

Disque 2  
Face B

CODE SOURCE - 178K,  
accès par Editeur et  
Système d'Assemblage.

Le Disque 2 est inversé  
selon les besoins pour éditer/  
assembler des fichiers sources.

### Title (Titre)

Modifie le Message d'En-tête des Listages d'Assemblage, en Haut des Pages.

Exemple: TITLE MegaGame, Section 6.

### Directives d'Assemblage Conditionnelles

L'Assembleur est capable d'inclure ou d'exclure certains blocs de code durant le processus d'assemblage, grâce à l'emploi de drapeaux et de directives conditionnelles. Associé au système SELECT décrit ci-avant, ce système devient extrêmement souple étant donné que les premiers fichiers peuvent définir des drapeaux qui, dans les fichiers ultérieurs, vont contrôler l'assemblage du code. Le mécanisme d'assemblage conditionnel est le classique SI quelque chose ALORS faites ceci OU ALORS faites cela.

Le quelque chose est une expression arithmétique. Si l'expression est VRAIE (c'est-à-dire NON-ZERO), le premier choix (THEN/ALORS) est sélectionné. Sans quoi, le système sélectionnera le second choix (ELSE/OU ALORS). Le second est optionnel et la directive final doit être une instruction ENDIF.

Les exemples suivants montrent comment on peut utiliser ce système. Le DRAPEAU du code de gauche est positionnée sur 1 (VRAI) et le THEN-CODE est assemblé. Dans la code de droite, le DRAPEAU est positionné sur 0 (FAUX) : c'est le ELSE-CODE qui est assemblé.

<pre> ; FLAG:      EQ      1 ;             IF      FLAG THEN-CODE: LD      A,(VALUE-1)             ELSE ELSE-CODE:  LD      A,(VALUE-2)             ENDIF ;             END         </pre>	<pre> ; FLAG:      EQU     0 ;             IF      FLAG THEN-CODE: LD      A,(VALUE-1)             ELSE ELSE-CODE:  LD      A,(VALUE-2)             ENDIF ;             END         </pre>
--	--

Vous noterez que l'emploi est toujours l'ordre IF... ELSE... ENDIF. L'instruction THEN est en effet implicite puisqu'elle constitue le premier branchement suivant l'instruction IF. La section ELSE est optionnelle. Une directive IF doit toujours être accompagnée d'une déclaration ENDIF correspondante.

Au lieu d'utiliser les directives IF..ELSE..ENDIF on peut utiliser les directives COND..ELSE..ENDC. Toutes deux sont valides.



## Definitions et Utilisation des Macro-Instructions

Une MACRO-INSTRUCTION est un petit élément de code, défini dans un fichier au début du processus d'assemblage. Lorsque son nom est utilisé, l'élément de code précédemment défini est à nouveau généré.

Une MACRO-INSTRUCTION détermine le début de la définition et doit avoir un label qui sert de Macro-Nom. Ce nom doit être ENTIEREMENT alphabétique et peut posséder jusqu'à SIX caractères. Imaginez la MACRO-INSTRUCTION comme constituant une nouvelle instruction pour l'Assembleur. Les deux exemples qui suivent représentent une MACRO-INSTRUCTION sans Utilisation de Paramètre à gauche, et avec une Utilisation de Paramètre à droite.

```
SBCX:  MACRO                                SWAP;  MACRO#P1,#P2
        XOR  A                                PUSH  AF
        SBC  HL,DE                            LD    A,#P2      (LD A,C)
        ENDM                                  LD    #P2,#P1   (LD C,D)
;                                             LD    #P1,A     (LD D,A)
        LD   HL,(VALUE1)                    PDP   AF
        LD   DE,(VALUE2)                    ENDM
        SBCX                                 ;
        LD   (VALUE3),HL                    SWAP  D,C
```

L'exemple de gauche explique comment on peut utiliser une petite macro-instruction comme instruction additionnelle; En ce cas, la "macro" SBCX est assemblée sous forme de deux instructions distinctes. La première supprime le drapeau de report avant exécution de la seconde SOUSTRACTION avec instruction de REPORT.'

L'exemple de droite représente une substitution de paramètre. Durant l'Assemblage, l'utilisation de la macro SWAP fait générer un sous-programme de cinq multiplats qui va faire interchanger le contenu des registres C et D. La macro SWAP peut s'utiliser pour interchanger deux registres quelconques, sauf le registre A.

### Utilisation des Macro Paramètres

Comme explicité dans l'exemple ci-avant, l'emploi des paramètres est positionnel et fonctionne par substitution. Si les paramètres contiennent des virgules ou des guillemets, ils doivent être renfermés entre des apostrophes ou des guillemets, comme représenté ci-dessous:

```
STRING:  MACRO      #P1,#P2
          DB         #P1          (DB 5)
          DB         #P2          (DB'21,22,23,24','X')
          ENDM
;
          STRING      5,"'21,22,23,24','X'"
;
```

### Générateur de Symboles Macro

Si une macro définition contient des labels, il surviendra des erreurs de DUPLICATION DE LABELS durant l'assemblage si l'on utilise la Macro-instruction plus d'une fois. En ces cas, il faut ajouter le suffixe #SYM au label. Chaque fois que l'on utilise la Macro, un suffixe à 4 chiffres est incrément et rattaché au label.

L'exemple suivant est une macro-instruction qui éprouve HL et substitue la constante hex-décimale 0FFFFH si HL est égal à zéro. Chaque fois qu'elle est utilisée, l'instruction JR TEST-#SYM et les labels TEST-#SYM sont portés à la valeur suivante. Au premier passage, le JR sera au label TEST-0001 puis il passera à TEST-0002 etc...

```
;
;          MACRO tests HL, if ZERO replace with 0FFFFH.
;          (MACRO teste HL, si ZERO, remplacer par 0FFFFH.)
;
TEST:      MACRO
            XDR          A          ; Clear Carry.          1.
            LD           DE,0       ; Set DE to zero.      2.
            SBC          HL,DE      ; Subtract / TEST      3.
            JR           NZ,TEST-#SYM ; if NZ JMP to TEST 000n  4.
            LD           HL,0FFFFH  ; else set HL to 0FFFFH  5.
TEST-#SYM: EQ          $           ; continue ... TEST-000n  6.
;
```

1. Supprimer report.
2. Set DE à zéro
3. Soustraire / TEST
4. si NZ SAUT à TEST-000n
5. sans quoi pos. HL sur 0FFFFH
6. poursuivre ... TEST-000n

### Listages Macro

En règle générale, seules les Macro Définitions sont listées. Pour visualiser le code complet, vous devez utiliser la directive MLIST. Pour SORTIR de cette facilité, utilisez la directive MNLIST.

### Instructions Supplémentaires

Il existe plusieurs instructions Z80 qui ne figurent pas normalement dans les manuels de programmation Z80 parce qu'elles ne fonctionnent pas toujours ! Si vous écrivez des logiciels pour d'autres utilisateurs de Z80, NE LES UTILISEZ PAS. S'il vous est essentiel de les employer, l'Assembleur PYRADEV les accepte mais vous devez tout d'abord donner la Directive 'EXTRA'.

Le premier groupe de codes opérations permet d'utiliser les registres universels à 16 bits IX et IY comme quatre registres à 8 bits en les catégorisant comme registres de POIDS FORT ou FAIBLE. Nous utilisons les opérandes LX, HX, LY et HY pour représenter respectivement les octets de poids Faible et de poids Fort de IX et IY. On peut également utiliser la variante XH, XL, YH et HL.

LD	HX,A	LD	HX,B	LD	HX,C	LD	HX,D	LD	HX,E	LD	HX,n
LD	LX,A	LD	LX,B	LD	LX,C	LD	LX,D	LD	LX,E	LD	LX,n
LD	HY,A	LD	HY,B	LD	HY,C	LD	HY,D	LD	HY,E	LD	HY,n
LD	LY,A	LD	LY,B	LD	LY,C	LD	LY,D	LD	LY,E	LD	LY,n
LD	A,HX	LD	B,HX	LD	C,HX	LD	D,HX	LD	E,HX		
LD	A,LX	LD	B,LX	LD	C,LX	LD	D,LX	LD	E,LX		
LD	A,HY	LD	B,HY	LD	C,HY	LD	D,HY	LD	E,HY		
LD	A,LY	LD	B,LY	LD	C,LY	LD	D,LY	LD	E,LY		
LD	HX,LX	LD	LX,HX	LD	HY,LY	LD	LY,HY				
INC	HX	INC	LX	INC	HY	INC	LY				
DEC	HX	DEC	LX	DEC	HY	DEC	LY				
ADD	A,HX	ADD	A,LX	ADD	A,HY	ADD	A,LY				
ADC	A,HX	ADC	A,LX	ADC	A,HY	ADC	A,LY				
SBC	A,HX	SBC	A,LX	SBC	A,HY	SBC	A,LY				
SUB	HX	SUB	LX	SUB	HY	SUB	LY				
CP	HX	CP	LX	CP	HY	CP	LY				
AND	HX	AND	LX	AND	HY	AND	LY				
OR	HX	OR	LX	OR	HY	OR	LY				
XOR	HX	XOR	LX	XOR	HY	XOR	LY				

Un second groupe de codes fournit d'autres codes opérations SHIFT-LEFT-LOGICAL (DECALAGE CIRCULAIRE A G.) qui complètent l'instruction SRL existante. Ce sont:

SLL A SLL B SLL C SLL D SLL E SLL H SLL L SLL (HL)

Les opérations sont les mêmes que pour l'instruction SLA, mais un (1) bit est placé dans la position de bit 0, où il remplace un (0).

Note: Les codes opérations qui précèdent ne sont pas standards (puisque'ils ne possèdent aucune définition formelle); néanmoins, ils correspondent bien aux codes mnémoniques utilisés dans plusieurs publications relatives à la programmation du Z80. Chaque fois que possible, nous avons utilisé les définitions communes.

### Format Libre

L'emploi de la directive FREE (LIBRE) permet à l'Assembleur de traiter un type d'expression Z80 qui est très utile à ceux qui écrivent des jeux. C'est un système d'expressions mélangées données et valeurs, qui convient à la définition de tables. La génération des octets est toujours implicite sauf lorsqu'un seul point d'exclamation (!) précède une expression, auquel cas le système génère une valeur de mot. L'emploi de DBXON est idéal lorsque vous commencez à travailler en format libre, car elle vous permet de vérifier si les tables sont correctement définies. Les exemples suivants expliquent comment utiliser le format LIBRE....

```

;
FREE                                ; FREE FORMAT.
DBXON                               ; DB EXPANSION.
;
FLAG1:    EQU            0
FLAG2:    EQU            1
VALUE1:   EQU            2
;
ROUTINE1: CALL          GET-CURSOR
          INC            H
          CALL          SET-CURSOR
          RET
;
CONTROL   TABLE
;
1,2,3,!100,!200,!300,FLAG0,FLAG1,"STRINGA","ABC1","111",!ROUTINE1
4,5,6,!200,!400,!600,FLAG1,FLAG2,"STRINGB","ABC2","222",!ROUTINE2
7,8,9,!300,!600,!900,FLAG2,FLAG2,"STRINGC","ABC3","333",!ROUTINE3
;
END OF CONTROL TABLE
;
DBXOFF                                ; DB EXPANSION OFF
;

```

FREE FORMAT = FORMAT LIBRE    DB EXPANSION    EXPANSION DB  
 CONTROL TABLE = TABLE DE COMMANDE  
 END OF CONTROL TABLE = FIN TABLE DE COMMANDE  
 DB EXPANSION OFF = SORTIE EXPANSION DB

Chacune des trois lignes de la table de commande ci-avant sera assemblée de manière analogue. Nous ne décrivons ici que la PREMIERE ligne, et le code binaire généré...

Trois octets contenant les valeurs 1, 2, et 3.

Trois mots contenant les valeurs 100, 200, et 300. Remarquez le !.

Un octet contenant la valeur drapeau-0.

Un octet contenant la valeur drapeau-1.

Une chaîne d'octets contenant 'STRINGA'.

Une chaîne d'octets contenant 'ABC1'.

Une chaîne d'octets contenant '111'.

Un mot unique avec l'adresse 'ROUTINE1'. Notez bien le !.

## Gestion des Mémoires

Durant le processus d'assemblage, l'affectation réserve en mémoire à la table des symboles, à la mémorisation du code source, à la table des correspondances, et aux macro-définitions, varie en fonction de la mémoire disponible.

Dans les Amstrad 464 et 664, qui possèdent tous deux 64K mémoire, la table des symboles est mémorisée à partir du bas de la mémoire vers le haut, tandis que la table des correspondances (optionnelles) est mémorisée du haut vers le bas. En outre, le système utilise une mémoire-tampon de 2K pour la lecture du code source. Si vous assemblez de très grands programmes, vous parviendrez à un point où il sera impossible d'obtenir une liste des correspondances, étant donné que le système 64K aura été utilisé au maximum.

Dans l'Amstrad 6128 (ou avec un 464/664 avec mémoire RAM supplémentaire\*), les espaces mémoires sont affectés comme suit:

- Le code source est mémorisé dans les blocs supplémentaires durant le premier passage ; ainsi, durant le second passage, la source est lue à partir de la mémoire, ce qui accélère le processus d'assemblage.
  
- La table des correspondances optionnelle est conservée dans 32K du second bloc de RAM si sélectionné, ce qui autorise au maximum 6.500 entrées. Ceci permet d'exécuter très importants assemblages.

\*Le système PYRADEV a fait l'objet d'essais très poussés avec un CPC464, une unité de disques DD1 et un CPC6128. On a pu utiliser des mémoires de 64K jusqu'à 384K au total, en travaillant avec le Kit d'Expansion de RAM 256K de Dk'TRONICS.

En poussant la capacité RAM au-dessus de 128K, on n'obtient qu'une amélioration marginale (10 à 15% environ) de la durée d'assemblage globale.

## Statistiques de l'Assembleur

A la fin du deuxième passage (génération du code objet), les statistiques suivantes sont affichées afin que vous puissiez voir où vous vous trouvez par rapport aux limites de traitement de l'Assembleur. La Mémoire à Symboles Libres est le paramètre critique, et doit toujours demeurer bien AU-DESSUS du zéro !

Nombre d'Erreurs ..... nnnn

Nombre de Symboles ..... nnnn

Table des Symboles de ..... nnnn à nnnn  
 Liste des Macros de ..... nnnn à nnnn  
 Nombre de Correspondances ..... nnnn  
 Table des Correspondances de .... nnnn  
 Mémoire des Symboles Libres .. nnnn  
 Fichier débute : nnnn Finit : nnnn Longueur : nnnn

### Recapitulatif des Directives de l'Assembleur

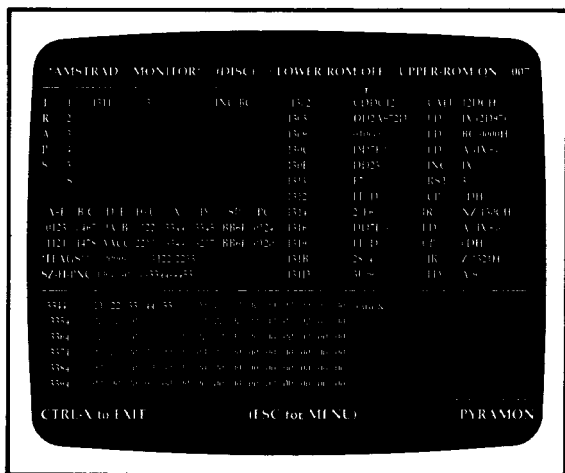
COND	«exp»	Assemblage conditionnel.
DBXOFF		Expansion DB inopérante.
DBXON		Expansion DB opérante.
DEFB	«exp»,«exp»	Définition d'octets.
DEFL	«exp»	Redéfinition d'un label.
DEFM	«exp»	Définition mémoire.
DEFS	«exp»	Définition stockage.
DEFW	«exp»	Définition valeur mot unique.
EJECT		Avance papier.
ELSE		Partie de l'assemblage conditionnel.
END		Fin du train d'entrée source.
ENDBIN		Fin du génération du code binaire.
ENDC		Fin d'assemblage conditionnel.
ENDIF		Fin d'assemblage conditionnel (IF).
ENDM		Fin de MACRO.
EQU	«exp»	Egale un label.
EXEC	«exp»	Définition adresse EXEC.
EXTRA		Activer codes op. supplémentaires.
FREE		Activer format LIBRE.
IF	«exp»	Assemblage conditionnel (IF).
LIST		Activer impression sélective.
LOAD	«exp»	Générer octets zéro jusqu'à «exp».
MACRO	«parm list»	Lancer macro définition.
MLIST		Activer macro listage.
MNLIST		Désactiver macro listage.
NOLIST		Désactiver impression sélective.
ORG	«exp»	Définir adresse code objet.
PRINTR	«parms»	Définir options imprimante.
PAUSE	message	Suspendre assemblage.
SELECT	«filename»	Ajouter un autre fichier.
TITLE	message	Changer l'en-tête de listage.
DB	comme DEFB	
DL	comme DEFL	
DM	comme DEFM	
DS	comme DEFS	
DW	comme DEFW	

## 4.0 MONITEUR

Le Moniteur PYRADEV est un moniteur de mise au point puissant, qui possède toutes les caractéristiques nécessaires pour produire des programmes et routines absolument parfaits, avant de les ajouter à un grand programme de système. On le sélectionne, à partir du MENU DU SYSTEM, en appuyant sur (M).

Il vous permet de CHARGER des programmes de test, de programmer des interruptions, de lancer l'exécution de codes, de travailler en pas à pas, de changer de code, d'écrire un code sur disque, etc. En outre, vous pouvez désassembler le contenu des mémoires et des ROM, en écrivant le code source désassemblé dans un fichier ASCII si besoin est.

**Avvertissement :** Si vous n'avez jamais auparavant utilisé un Moniteur, veuillez vous exercer en exécutant de très simples routines (PROGRAM.001 par exemple). Ne commencez pas immédiatement à utiliser votre moniteur sur un code qui modifie, par exemple, la zone de l'écran, ceci étant difficile à comprendre.



1. ROM INFERIEURE INACTIVE
2. ROM SUPERIEURE ACTIVE
3. CTRL-X POUR SORTIR
4. (ESC) POUR REVENIR AU MENU)

Figure 4.0 Moniteur de mise au point

NOTE: Votre code de test ne doit pas initialiser de mémoires ROM de fond. Celles-ci sont déjà actives (via KL.ROM.WALK), que PYRADEV appelle au moment de la mise en route.

## Visualisation

L'Écran du Moniteur est divisé en cinq sections différentes. La section supérieure gauche contient l'information d'intersection/piégeage.

La section central gauche contient le contenu des registres (actuels et précédents) et les réglages de drapeaux, ainsi que les quatre octets de mémoire correspondant à chacun des quatre registres de paires d'adresses DE, HL, IX et IY.

La section inférieure est une visualisation de mémoires. Elle alterne avec l'affichage d'un menu.

Tout en haut et à droite de l'écran, vous voyez l'état des sélections de mémoires ROM inférieure et supérieure.

La droite de l'écran affiche le désassemblage principal.

## Translation

Le Moniteur est parfaitement translatable et, dans un premier temps, se charge dans une adresse élevée de mémoire (qu'il affiche). Il demande ensuite si la translation est nécessaire. Si vous répondez Oui (Y), vous pouvez ensuite sélectionner le Moniteur (S)andard ou (M)ini.

Le Mini-Moniteur est un sous-ensemble du Moniteur-Standard, que vous pouvez utiliser lorsque l'espace disponible en mémoire est limité. Les fonctions qui suivent sont exclues du Mini-Moniteur : Ecriture-Fichier-à-Disque, Copie-Mémoire, Menu-Help, Changements-d'Encres, Sérigraphie Affichage-ROM et Désassemblage sur Disque ou Imprimante.

Le Moniteur demande ensuite si vous voulez spécifier une borne inférieure (adresse de départ) ou une borne supérieure (adresse de fin).

Si vous spécifiez une borne Inférieure, vous demandez au Moniteur de se traduire de manière à ce que son adresse la plus basse ne descende pas au-delà de la borne Inférieure.

Si vous spécifiez une borne Supérieure, vous ordonnez au Moniteur de se traduire de manière à ce que son adresse la plus élevée n'exède pas la borne Supérieure.

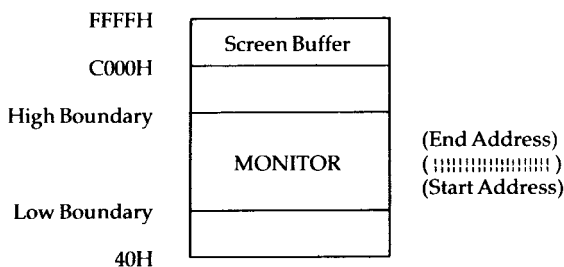


Figure 4.1

1. Tampon d'écran
2. Borne supérieure
3. Borne inférieure
4. Adresse de fin
5. Adresse de lancement



## Affichage du Menu

Une fois prêt, vous pouvez faire afficher le Menu du Moniteur, en appuyant sur la touche ESC. Le menu alterne avec la section d'affichage de la mémoire en bas de l'écran.

T....TRAP	G..GOTO-ADDR	L...LOAD	Q...QUERY	J...PAPER	C...CAT
Z.CLR-TRAP	G...STEP-IST	W..WRITE	N.NEW-SCN	KK...PEN	T...TAPE
M.A...MEMORY	S..SSTEP(f1)	O..OTHER	X...ROMS	D...DIS-ASM	...PRINT
R.....REGS	S...DSTEP(f2)	U.UPDATE	Y....COPY	X...EXIT	

Figure 4.2 Menu du Moniteur

## Selection de Périphérique

Appuyez sur (CTRL-T) pour commuter sur BANDE ou DISQUE. Le périphérique sélectionné est indiqué en haut de l'écran, et sera utilisé pour LIRE et ECRIRE des fichiers. Appuyez sur (CTRL-C) pour consulter la visualisation du CATALOGUE périphérique.

## Code de Test de Chargement

Appuyez sur (L) pour charger un fichier de programme, et entrez le nom du fichier. L'en-tête de fichier sera examiné par le système afin qu'il puisse afficher le point de chargement et l'adresse d'exécution. Vous pouvez modifier le point de chargement avant d'appuyer sur la touche ENTER pour lire le fichier en mémoire si vous le désirez.

## Affichage de Désassemblage

Un symbole Tête de Flèche vers la Droite vous signale où le curseur d'adressage se trouve. Appuyez sur la touche (O) jusqu'à ce que la Tête de Flèche vers la Droite se trouve en haut et à droite de l'écran, puis entrez une adresse de désassemblage. Vous pouvez faire défiler l'affichage à l'aide de la flèche et des touches de déplacement de flèche.

## Pour Changer Code et Mémoire

Utilisez la touche ESC jusqu'à ce que l'affichage de mémoire apparaisse en bas de l'écran. Frappez la touche (O) jusqu'à ce que la Tête de Flèche vers la Droite apparaisse sur la ligne d'adresse de l'affichage de mémoire. Entrez l'adresse à modifier. Appuyez sur (M) pour sélectionner MODIFICATION-MEMOIRE. Déplacez le curseur et recouvrez la mémoire selon les besoins, en utilisant des codes hexa à deux chiffres par octet. Au lieu de (M), vous pouvez utiliser (CTRL-A) pour modifier la mémoire en code ASCII. Appuyez sur la touche ESC lorsque vous aurez fini toutes les modifications. Si vous avez modifié des codes, appuyez sur (O) pour ramener le curseur en haut et à droite de l'écran, entrez l'adresse où les modifications ont été faites : le nouveau code sera désassemblé sur l'écran.

Vous pouvez faire défiler l'affichage vers le haut et le bas d'un octet, de 16 octets ou de 128 octets, en utilisant les touches avec flèche vers le bas et le haut, dans les états normal, décalé et de commande, selon les besoins.

### **Positionnement des Pièges (Traps)**

Utilisez la partie supérieure droite de l'écran pour étudier le code, puis appuyez sur (T) pour positionner un piège. Entrez la première adresse d'octet de l'instruction que vous voulez piéger. L'une des cinq lignes de piégeage sera utilisée pour signaler l'instruction qui a été remplacée par le piège. Utilisez la fonction (G) pour Goto) pour lancer l'exécution du code. Lorsque le piège survient, la ligne de piégeage appropriée sera affichée avec couleurs inversées. A partir d'un piège, vous pouvez continuer de deux manières.

S – Lancer l'exécution de code pas-à-pas à partir de l'adresse du piège.

G – Goto (allez à) l'adresse : l'adresse implicite poursuit l'exécution.

Lorsqu'un piège survient, il s'efface automatiquement. Si vous voulez qu'il survienne à nouveau, vous devez le repositionner avant de poursuivre.

### **Intruction "Goto Address"**

Après avoir positionné les pièges, ou s'ils sont survenus, appuyez sur (G) pour lancer ou relancer l'exécution du code. Une adresse de relancement sera affichée et vous pouvez l'accepter en appuyant sur la touche Enter. Pour passer à un point d'exécution spécifique, recouvrez l'adresse affichée avec une adresse de votre choix.

### **Etude Pas à Pas du Code**

Après un piège ou en définissant le premier point de piégeage avec (CTRL-G), vous pouvez étudier un code pas à pas.

Appuyez sur (S) ou sur (f1) pour faire une étude PAS A PAS. Cette forme de progression suit un parcours d'adressage logique (débit de commande) au fur et à mesure de l'exécution du code. S'il survient un code "CALL to ROM" (APPEL MEMOIRE ROM), l'opération pas à pas est suspendue étant donné que le mécanisme d'avance pas à pas ne peut pas recouvrir la mémoire ROM pour établir les pièges nécessaires.

Appuyez sur (CTRL-S) ou (f2) pour fonctionner en DOUBLE PAS. Cette forme d'avance va définir des pièges dans des emplacements de mémoire, en ordre croissant. Ainsi, lorsque l'on travaille pas à pas et qu'il survient un "CALL to ROM" (APPEL DE MEMOIRE ROM), on utilise un double pas. Ceci définit un piège sur l'instruction de mémoire suivant, et on peut ainsi reprendre l'avance pas à pas (pas unique ou double) après achèvement de l'opération CALL et retour de la commande à partir du sous-programme ROM. Les programmes AMSTRAD utilisent en règle générale de nombreux appels ROM pour l'exploitation du système.

Le MODE (0,1,2) qu'utilise votre programme peut être modifié à l'aide de shift 0,1, ou 2. Lorsqu'il est sur 2, le mode est identique au MONITEUR et, par conséquent, la visualisation ne sera PAS effacée chaque fois qu'on exécute un piège ou un pas.

### **Changement du Contenu d'un Registre**

Appuyez sur (R) pour changer le contenu d'un registre, puis recouvrez soigneusement les valeurs existantes par les nouvelles valeurs. Appuyez sur la touche ESCAPE pour sortir de la fonction de changement. Les nouvelles valeurs seront alors utilisées quand on reprend l'exécution du code, soit par emploi de la touche (G)oto soit en fonction (S)tep (pas unique).

### **Nouveau et Actualisation**

Vous pouvez actualiser l'affichage de mémoire ou de désassemblage en cours, en appuyant sur (U). Vous devez faire cette opération lorsque vous surveillez une mémoire en cours de changement. Lorsque l'exécution du code recouvre l'écran, (ce qui le rend difficile à lire), vous pouvez utiliser la commande (N)ew (nouveau) pour actualiser l'écran complet.

### **Interrogation, Recherche**

Un appui sur (Q) sélectionne le mode interrogation (query) et permet de faire rechercher un HEXA ou un ASCII de 30 caractères au maximum. On peut utiliser des caractères de remplacement (?) en format ASCII aussi bien que HEXA. Chaque fois que le système trouve une correspondance, il l'affiche sur la ligne deux de la limitée entre une adresse de mémoire inférieure et supérieure, et peut être suspendue ou poursuivie après chaque correspondance.

### **Examen des Mémoires ROM**

Les affichages de désassemblage de mémoire lisent tous deux la mémoire en fonction des sélections de LOWER et UPPER ROM (ROM INFÉRIEURE ET SUPÉRIEURE). En appuyant sur la touche (X), vous pouvez modifier l'état des sélections ROM.

Les zones d'affichage seront modifiées si elles affichent des zones de mémoire qui se trouvent au-dessous de 16K ou au-dessus de 48K. lorsqu'on commute les ROM inférieure et supérieure.

### **Désassemblage sur Disque, Ecran ou Imprimante**

Appuyez sur CTRL-D et entrez une adresse START et END pour identifier la zone de mémoire à désassembler. Entrez une adresse de Zone de Travail dans laquelle pourra être définie la table des symboles. L'adresse implicite fournie définit une zone de 1000 octets et peut s'utiliser dans la plupart des cas. Si l'espace est entièrement occupé par le processus de désassemblage (500 labels environ), un message d'erreur sera affiché et il faudra reprendre le désassemblage avec une zone de travail différente plus largement dimensionnée.

On peut définir des zones de données par des paires d'adresses Début-Fin. Utilisez une paire de zéros pour terminer vos sélections. On peut sélectionner la sortie du processus de désassemblage à l'aide de (V), (D) ou (P) qui dénotent respectivement l'Ecran, le Disque ou l'Imprimante. Note: Il peut survenir des références à des labels inexistant si les zones de données imbriquées ne sont pas définies avant le désassemblage.

Si vous avez sélectionné (D), entrez le nom de fichier à utiliser. Le désassemblage sur Ecran ou Imprimante peut être suspendu à l'aide de la touche Escape et repris avec n'importe quelle autre touche. Si vous avez choisi l'option (D), un seul appui sur la touche Escape fait abandonner le processus.

Le désassemblage sur fichier(s) de disque pause après génération de chaque groupe de 30K de code source, ce qui permet d'entre les noms de fichiers suivants avant la reprise du processus.

### **Ecriture**

Pour écrire une section de mémoire sous forme d'un fichier binaire, appuyez sur (W). Le point de chargement, l'adresse d'exécution et la longueur (sauvegardée au moment du chargement du fichier) sont réaffichés et peuvent être recouverts le cas échéant. Après avoir entré un nom de fichier, la section de mémoire appropriée est écrite sur disque sous forme d'un fichier binaire.

### **Copie de Mémoire**

On peut faire copier une section de mémoire en appuyant sur (Y). Définissez le bloc à déplacer avec une adresse DEBUT et FIN (BEGIN et END). Spécifiez ensuite l'adresse de destination. Le bloc va être copié "intelligemment". Ceci signifie que la destination peut se trouver n'importe où et peut chevaucher la zone du bloc initial.

### **Sélection de Bloc-Zone Mémoire**

Appuyez sur (CTRL-B) pour sélectionner un autre bloc de mémoire à implanter dans la gamme d'adresses 4000H à 8000H. (Systèmes 128K et plus grands uniquement).

### **Impression de l'Ecran**

Appuyez sur la touche (.) du bloc numérique (fonctions). Le contenu de l'écran sera décodé et imprimé. Appuyez deux fois sur la touche ESC pour annuler l'impression.

### **Couleurs, Papier et Plume**

L'emploi des touches (J) et (K) fait passer les couleurs de papier et de plume à leurs valeurs d'encre suivantes. En expérimentant avec les encres, vous améliorez la lisibilité de l'écran sur les moniteurs monochrome aussi bien que couleurs, en fonction des conditions d'éclairage de fond. Vous pouvez utiliser CTRL-J et CTRL-K pour faire défiler les couleurs d'encre en sens inverse.

### **Affectation de Zones d'Empilage**

Si le Moniteur est chargé au-dessus de 4100H, il affecte une zone d'empilage (stack area) à l'aide de la valeur HIMEN courante. Si le Moniteur est chargé au-dessus de 4100H, la zone d'empilage est affectée just au-dessous du point de chargement utilisé.

La fonction catalogue (CTRL-C) utilise un tampon de 2K qui se trouve à 2048 + 200 octets au-dessous du "stack".

### **Acces Externe**

La première instruction du moniteur est JP nnnn. Vous pouvez l'utiliser pour "sauter" dans le moniteur à partir de votre programme si besoin est.

## **5.0 LE "DISC-NURSE"**

On sélectionne DISC-NURSE dans le menu du système en appuyant sur (D). Ce sous-programme attend alors que vous ayez chargé un disque. Après le chargement du disque, appuyez à nouveau sur la touche (D) sans quoi vous ne pourrez pas utiliser le DISC-NURSE. Il vous offre plusieurs fonctions regroupées dans un menu très facile à utiliser. Ces fonctions vous permettent d'explorer votre disque ou vos disques, et si nécessaire, d'y apporter des modifications.

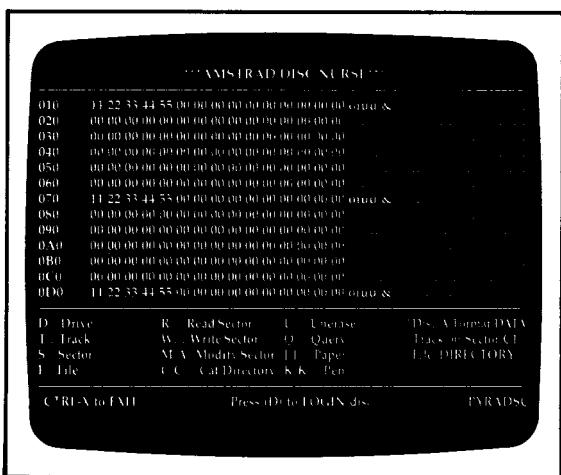


Figure 5.0 Disc-Nurse

- |                                    |                                      |                                 |
|------------------------------------|--------------------------------------|---------------------------------|
| <b>D</b> = Unité de disques        | <b>T</b> = Piste                     | <b>S</b> = Secteur              |
| <b>F</b> = Fichier                 | <b>R</b> = Secteur lecture           | <b>W</b> = Secteur écriture     |
| <b>M.A.</b> = Secteur modification | <b>C.C.◇.</b> = Catalogue/répertoire | <b>U</b> = Annuler l'effacement |
| <b>Q</b> = Interrogation           | <b>J.J.◇.</b> = Papier               | <b>K.K.◇.</b> = Plume           |

CTRL-X to EXIT = Appui sur CTRL pour SORTIER

Press (D) to LOGIN disc. = Appui sur (D) pour faire prendre le disque en charge.

**Avertissement :** Vous ne devez pas modifier directement des secteurs de disque sans avoir au préalable effectué une version de sauvegarde du disque, sans quoi vous pouvez aisément perdre votre jeu favori, des semaines de développement de code source ou même le contenu total du disque. Vous devez être conscient des structures de disque AMSTRAD et des constructions d'en-tête de fichier avant de changer quoi que ce soit. Prenez bien note de cet avertissement !

### Unité de Disques

Appuyez sur (D). Le système affiche la demande 'Drive A or B'. Appuyez sur (A) ou (B) pour sélectionner l'unité de disques requise : le répertoire approprié est affiché.

Il est IMPERATIF d'effectuer cette opération si vous changez le(s) disque(s) en cours d'examen !

### Piste et Secteur

Appuyez sur (T) pour entrer une adresse de piste, et sur (S) pour entrer une adresse de secteur.

### Lecture

Appuyez sur (R) pour lire le secteur sélectionné. Après lecture d'un secteur, vous pouvez utiliser les touches suivantes. Elles vous permettent de suivre des chaînes de fichiers ou de lire progressivement et régressivement au niveau du secteur.

TOUCHE	Normal	Décalage	Commande
GAUCHE DROITE	Recul chaîne 1/2 sec Avance chaîne 1/2 sec	Recul chaîne 1 sec Avance chaîne 1 sec	Trouver début Trouver fin
VERS LE HAUT VERS LE BAS	1/2 secteur précédent 1/2 secteur suivant	Secteur précédent Secteur suivant	Inutilisée Inutilisée

Figure 5.1

### Modification d'un Secteur

Appuyez sur (M) pour modifier, en utilisant HEX ou (A) si vous modifiez en ASCII. Utilisez les touches de curseur (flèches) pour amener le curseur sur les octets que vous voulez, et surfrappez les valeurs selon les besoins. Si vous modifiez les octets d'en-tête de fichiers, appuyez sur (CTRL-H) après la modification pour recalculer l'octet du total de contrôle. Une fois les changements de secteur achevés, appuyez sur la touche ESC. Le secteur modifié peut être écrit sur disque à l'aide de la commande (W).

### Écriture

Appuyez sur (W) pour ré-écrire le secteur affiché/modifié sur disque. La demande d'écriture doit être confirmée avec la touche (Y). Toute autre touche fera abandonner la demande d'écriture.

**Avertissement :** Le secteur sera écrit sur la Piste et le Secteur indiqué en bas et à droite de l'écran.

### Annulation d'Effacement

Vous pouvez utiliser cette fonction pour reprendre un fichier ayant été accidentellement effacé. Appuyez sur (U) pour sélectionner la fonction d'annulation d'effacement, et entrez un nom de fichier. Le DISC-NURSE va vérifier les tables d'affectation de secteurs. Si les secteurs de fichier n'ont pas été utilisés, le fichier sera rétabli alors à l'emploi normal et réapparaîtra sur les affichages de répertoire ultérieurs.

### Accès aux Fichiers

Pour accéder aux secteurs appartenant à un fichier spécifique, appuyez sur (F) et entrez un nom de fichier. Le premier secteur du fichier sera lu. Voyez la fonction (R)ead (Lecture) ci-avant qui contient une définition des fonctions de la touche défilement/parcours.

### Interrogation

En appuyant sur (Q), on sélectionne la fonction interrogation/recherche. La recherche peut se limiter à un (S) pour secteur, (F) pour fichier, ou (D) pour disque, et peut s'appliquer à une chaîne ASCII ou à une chaîne HEXA. Les caractères de remplacement (7) sont autorisés.

Vous pouvez entrer la chaîne hexa sous forme d'une chaîne continue ou interrompue de chiffres hexa, c'est-à-dire NNNNNN ou NN NN NN.

La chaîne débute à partir de la position courante de piste/secteur, et se poursuit jusqu'au dernier secteur. La recherche d'une chaîne ASCII peu probable constitue une bonne manière de vérifier un disque.

Lorsque vous trouvez une correspondance, l'adresse du secteur, le décalage du mot et le contenu du secteur sont affichés sur l'écran. Pour poursuivre la recherche, appuyez sur la BARR D'ESPACEMENT (SPACE) ou terminez avec la touche ESC.

### **Catalogue et Répertoire Entendus**

Appuyez (C) pour faire afficher le catalogue du disque en format de visualisation standard. Pour obtenir des informations de répertoire plus complètes, ainsi que des fonctions de reprographie, appuyez sur (CTRL-C). Si la sortie doit être imprimée, répondez (Y) à la question sur l'impression. On peut entrer une ligne de titre qui apparaîtra en haut de la liste.

### **Impression de l'Écran**

Appuyez sur la touche (.) du bloc numérique (fonctions). Le contenu de l'écran sera décodé et imprimé. Appuyez deux fois sur la touche ESC pour annuler l'impression.

### **Couleurs, Paper et Plume**

L'emploi des touches (J) et (K) fait passer les couleurs de papier et de plume à leurs valeurs d'encre suivantes. En expérimentant avec les encres, vous améliorez la lisibilité de l'écran sur les moniteurs monochrome aussi bien que couleurs, en fonction des conditions d'éclairage de fond. Vous pouvez utiliser CTRL-J et CTRL-K pour faire défiler les couleurs d'encre en sens inverse.

## **6.0 PROGRAMME UTILITAIRE**

Le Programme Utilitaire offre des fonctions générales de gestion de fichiers et de copie regroupées dans un logiciel unique et d'emploi aisé. Les fonctions sont les suivantes:

- Visualisation des répertoires d'unités de disques (A) et (B).
- Changement de noms de fichiers, avec liaison à la commande !REN.
- Effacement de fichiers, avec à la commande !ERA.
- Transfert de fichiers ; n'importe quel des fichiers AMSDOS, bande et disque.

### **Message Initial**

L'écran affiche le message suivant ....

Drive (A), (B), (D)elete, (R)ename, (C)opy, ou (X) to Exit.

En sélectionnant (A) ou (B), vous faites afficher le répertoire approprié. La sélection de (D) ou (R) vous donne les fonctions de suppression et de changement de nom de fichier, et s'explique d'elle-même. (Vous pouvez utiliser des caractères de remplacement).

### **Copie de Fichiers**

La fonction de Copie est un programme de copie universel, qui peut copier N'IMPORTE QUEL fichier AMSDOS standard sur DISQUE ou BANDE ou à partir d'un DISQUE ou d'une BANDE. Si vous la sélectionnez, le système affiche:

1: Disc-Disc    2: Disc-Tape    3: Tape-Disc

Vous devez donner une réponse valide sans quoi la commande repasse au message initial. Après avoir sélectionné 1, 2 ou 3 vous devez entrer un nom de fichier d'ENTREE. L'option 3 ne requiert pas de nom de fichier.

Après ouverture du fichier d'ENTREE, le système affiche l'information d'en-tête. Vous pouvez ensuite poursuivre la copie en répondant (Y) à la question COPY-?.

Selon l'option choisie et le type de fichier détecté, l'opération de copie produit plusieurs messages. Les options et réponses valides sont décrites sur la page suivante.

NOTE: Vous pouvez modifier la Bande ou le Disque Destinataire AVANT d'entrer la réponse à 'PROTECTION?'.

#### Restrictions de l'Option de Copie

	Disque/Disque	Disque/Bande	Bande/Disque
ASCII	Compression? Répondez Y ou N Limite 36K ***		
BINAIRE	Limite 36K ***	Note-1	Note-1
BASIC	Limite 36K ***		
CPM	Compression? Répondez SPACE-BAR Limite 36K ***		

Vous pouvez INTRODUIRE ou RETIRER la protection dans tous les cas.

*Figure 6.0 Options de Copie*

Note-1: Un fichier BINAIRE écrit sur bande par l'intermédiaire du système Utilitaire PYRADEV DOIT être ré-enregistré sur disque avec le même programme Utilitaire de manière à obtenir l'adresse de point de CHARGEMENT correcte dans l'en-tête de fichier. Ces fichiers NE DOIVANT PAS être EXECUTES avant cette opération. Vous pouvez cependant les CHARGER dans un emplacement de mémoire spécifique, à partir d'une bande et après avoir utilisé la touche CALL. (Si vous écrivez des jeux : ATTENTION !).



## 7.0 ENTRAINEMENT

Comme vous l'aurez réalisé, le système PYRADEV est petit mais puissant. Il possède de nombreuses caractéristiques et fonctions qu'il vous faudra découvrir et de nouvelles commandes que vous devrez apprendre. C'est pourquoi le fichier 'PROGRAM.001' existe sur le DISQUE SYSTEME (MASTER DISQUE) afin de vous permettre d'apprendre à Assembler, à Editer et en fait à jouer.

Le programme contient un seul sous-programme dénommé 'DEBUG'. Il peut être CHARGE (LOAD) et APPELE (CALL) à partir du BASIC et affichera le contenu des registres sur la rangée 25 de l'écran. Ce programme est accompagné de nombreux commentaires en deux types de texte.

Le (TEXTE EN MAJUSCULES) qui est entre parenthèses concerne entièrement le code du programme. Nous espérons que vous apprendrez rapidement comment le programme fonctionne, et que vous pourrez peut-être l'adapter comme sous-programme de mise au point additionnel au code que vous allez écrire.

Le texte normal (;\*Normal Text\*) renfermé entre des points virgules, contient des informations sur l'emploi de l'Editeur de Fichier Source. Vous devrez lire ces commentaires et vous exercer aux fonctions. Lorsque vous aurez fini de "jouer" avec l'Editeur, appuyez sur CTRL-A pour l'abandonner, puis (Y) pour confirmer et sortir.

Le fichier PROGRAM.001 peut être Assemblé tel quel. Faites-le dès le début avec le Macro Assembleur, et utilisez les diverses options OUTPUT (SORTIE) pour voir ce qui arrive. Nous vous recommandons d'utiliser tout d'abord les options implicites.

Le fichier PROGRAM.002 est aussi fourni sur disque. Il constitue un exemple complet de TOUTES les instructions Z80. Si vous avez des programmes de code, vérifiez si vous avez utilisé la syntaxe correcte, et les mnémoniques de code en examinant ce programme. Il contient également des exemples des directives d'Assemblage décrites dans la section 3. Il constitue un élément de code de référence utile et peut être Assemblé et Listé comme tableau de référence.

Le fichier PROGRAM.003 est un simple programme FILE-COPY (FICHIER/COPY). Vous pouvez le modifier et l'utiliser pour transférer d'autres formats de fichiers ASCII dans le système PYRADEV, en éditant un code spécialisé qui effectuera des modifications spéciales durant l'opération de transfert/copie.

Soyez sûr que vous comprenez à fond le fonctionnement du système avant de l'utiliser pour un code de projet réel, et veillez à toujours conserver des copies de sauvegarde de votre code source d'élaboration.

Merci d'avoir acheté PYRADEV. Nous espérons qu'il vous plaira.

## 8.0 RECAPITULATIF DES TOUCHES DE COMMANDE

### Editeur (mode d'édition)

CTRL-A: Abandon et Sortie

CTRL-B: Début Pointeur

CTRL-D: Effacement Bloc

CTRL-E: Fin Pointeur

CTRL-F: Trouver Marqueur

Bloc num.(.): Impression

ESCAPE: SOS-Affichage

CTRL-G: Goto Ligne nnn

CTRL-L: Posit. Marqueur

CTRL-O: Ouvrir Text

CTRL-S: Recherche Système

CTRL-T: Poser les Tabs

### Editeur (mode commande)

A: Unité AS Opérante

B: Unité B Opérante

C: Compression

D: Effacement Fichier

R: Changement Nom Fichier

Z: Sauvegardes ZAP

### Assembleur (touches de commutation de temps d'exécution)

V: Sortie Vidéo

P: Listage Impression

S: Impression Sélective

E: Impression Erreurs

### Moniteur (menu principal)

T: Définition valeur piste

Z: Définition valeur secteur

R: Changement Registres

M: Modif. avec HEXA

CTRL-A: Modif. ASCII

G: Goto une Adresse

CTRL-G: Défini. adresse de pas

CTRL-T: BANDE/DISQUE

SHIFT 0,1,2: Déf. Mode

S: Exéc. Pas Unique

CTRL-S: Double Pas

L: Charge. Fichier Test

W: Ecrire Fichier Disque

### Disc-Nurse (menu principal)

D: Sélection Unité Disques

T: Définition valeur piste

S: Définition valeur secteur

F: Sélect. Fichier

U: Annul. effacement fichier

Bloc num.(.): Impression écran

R: Lecture Secteur

W: Ecriture Secteur

### Programme Utilitaire (menu principal)

A: Affichage Unité A

B: Affichage Unité B

D: Effacement Fichier

### Programme Utilitaire (menu copie)

1: Disque/Disque

(Compression facultative ASCII

)

2: Disque/Bande

(Introduction ou Extraction de Protection

)

3: Bande/Disque

(Affiche toujours l'Information d'entête

)

Bloc num. (0): Mémoires KS

Petit-Enter: Commute KS

CTRL-U: Annuler Changements

CTRL-V: Visual. Début/Fin

CTRL-X: Sauvegarde et Sortie

CTRL-Y: Varier Curseur

CTRL-Z: Mode Commande

Bloc num(1-9): Relecture KS

Marche/Arrêt enregistrement

L: Options Chargement

S: Options Sauvegarde

J,CTRL-L: Papier

K,CTRL-K: Encre Plume

O: Autre Affichage

U: Actualisation Ecran

N: Régénération écran

CTRL-C: Catalogue

CTRL-B: Sélection Bloc

Q: Interrogation / Recherche

X: Examiner toutes ROM

Y: Copie Zone Mémoire

CTRL-D: Désassemblage

CTRL-X: Sortie Moniteur

J,CTRL-J: Encre Papier

K,CTRL-K: Encre Plume

Bloc num.(.): Impression Ecran

Q: Interrogation/Recherche

J,CTRL-J: Papier

K,CTRL-K: Encre Plume

C: Affichage Catalogue

CTRL-X: Répertoire Etendu

M: Modifier avec HEXA

A: Modifier avec ASCII

(CTRL-H: Total contrôle En-têtes)

R: Changement nom Fichier

X: Sortie Programme

C: Copier Fichier AMSDOS