memento

CLEFS POUR AMSTRAD

CPC 464-664/6128 et PCW 8256

2. Système disque

Daniel Martin
Philippe Jadoul

Editions du PSI

CLEFS POUR AMSTRAD

2.Système disque

Connaissez-vous toute la collection Amstrad chez P.S.I. ?

Pour les Amstrad CPC 464, 664 ou 6128 :

Initiation :
— La découverte de l'Amstrad — Daniel-Jean David
— Exercices en Basic pour Amstrad — Maurice Charbit

Programmation Basic
— 102 programmes pour Amstrad — Jacques Deconchat
— Amstrad en famille — Jean-François Sehan
— Super jeux Amstrad — Jean-François Sehan
— Super générateur de caractères sur Amstrad — Jean-François Sehan
— Photographie sur Amstrad et Apple II — Pierrick Moigneau et Xavier de la Tullaye

Maîtrise du Basic
— Basic Amstrad, 1. Méthodes pratiques — Jacques Boisgontier et Bruno Césard
— Basic Amstrad, 2. Programmes et fichiers — Jacques Boisgontier
— Basic Plus, 80 routines sur Amstrad — Michel Martin
— Périphériques et fichiers sur Amstrad — Daniel-Jean David

Assembleur et Pascal
— Assembleur de l'Amstrad — Marcel Henrot
— Turbo Pascal sur Amstrad — Pierre Brandeis et Frédéric Blanc

Système
— Clefs pour Amstrad, 1. Système de base — Daniel Martin
— Le livre de CP/M Plus sur Amstrad 6128 et 8256 — Yvon Dargery

A paraître :

— Amstrad en musique — Daniel Lemahieu
— Intelligence artificielle : langage et formes sur Amstrad — Thierry Levy-Abegnoli et Olivier Magnan
— Animation graphique sur Amstrad — Gilles Fouchard et Jean-Yves Corre
— Simulation et intelligence artificielle sur Amstrad — René Descamps
— Clefs pour Amstrad 8256 — Eric Baumarti

Pour tout problème rencontré dans les ouvrages P.S.I.
vous pouvez nous contacter au numéro ci-dessous :

Numéro Vert/Appel Gratuit en France

05 21 22 01

(Composer tous les chiffres, même en région parisienne)

Tous droits de traduction, d'adaptation et de reproduction par tous procédés réservés pour tous pays.

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© Editions du P.S.I., B.P. 86, 77402 Lagny Cedex, 1986
ISBN : 2-86595-256-8

CLEFS POUR AMSTRAD

2.Système disque

par

Daniel Martin

et

Philippe Jadoul

Editions du PSI
1986

Présentation des auteurs :

Daniel Martin : agrégé en mathématique-physique. Après quelques mois au Ministère de l'Education Nationale, attiré par la micro-informatique, il entre comme computer manager chez Tandy Corporation. Un bref passage chez APPLE aux Pays-Bas et, depuis 1981, il est ingénieur système chez Intertechnique, le constructeur français spécialisé en mini-ordinateurs de gestion architecturés sur une base de données PICK. Fin 1984, il publie "Le livre du MSX", début 1985, "Les dessous du SPECTRAVIDEO", en août 1985, "Clefs pour l'AMSTRAD" et, en septembre 1985, "Le livre de l'Amstrad", en collaboration avec Philippe Jadoul.

Philippe Jadoul : ingénieur industriel en électronique. Après des études à l'ISIL, il a suivi une formation complémentaire en technique des microprocesseurs. Ingénieur de maintenance chez Intertechnique pendant un an, il est, depuis 1984, ingénieur au laboratoire de recherche de l'Institut National des Industries Extractives (INIEX).

SOMMAIRE

	Pages
PRESENTATION	**9**
CHAPITRE I - AMSDOS	**11**
Spécifications	12
Instructions et fonctions Basic modifiées par AMSDOS	14
Instructions externes	17
Messages d'erreurs de l'AMSDOS	20
Utilitaires (6128 seulement)	22
CHAPITRE II - LOGICIEL INTERNE DE L'AMSDOS	**25**
Routines cassette modifiées pour le disque	25
Vecteurs propres au disque	29
Principales adresses de la ROM disque	32
Table des principales variables système	36
CHAPITRE III - CP/M 2.2	**39**
Organisation mémoire du CP/M 2.2	40
Carte mémoire	41
Organisation générale et fonctionnement	42
Démarrage à froid ou COLD BOOT	42
Démarrage à chaud ou WARM BOOT	42

SOMMAIRE

Extension résidente en mémoire 43
Octet d'allocation d'entrée/sortie ou I/O byte 43
Caractères de contrôle compris par CP/M 44
Messages d'erreur 44
Secteur de configuration 45
Tables de paramétrage du disque 46
Formats de la disquette 50
Structure du répertoire (DIRECTORY) 51
Octets réservés en page 0 52
Structure du FCB (File Control Bloc) 52
Nom de fichier et carte de sélection 53
Principaux noms d'extensions 54

Table des vecteurs standards du BIOS 55

Table des vecteurs propres au BIOS de l'AMSTRAD 57

Table des vecteurs standards du BDOS 59

Instructions résidentes 67

Utilitaires propres à Digital Research 69

Utilitaires propres à AMSTRAD 77

CHAPITRE IV - CP/M 3.0 ou CPM PLUS 79

Organisation mémoire du CP/M 3.0 80

Organisation générale et fonctionnement du CP/M 3.0 82
Démarrage à froid ou COLD BOOT 82
Périphériques logiques 82
RSXs 83
Valeurs d'initialisation de la page 0 83
Bloc de Contrôle du Système (SCB) 84
Table disque et la DPH 86
DPB (Disk Parameter Bloc) 87
Table BCB 87
FCB 88
Tables XFCB et SFCB 88

Table des vecteurs standards du BIOS du CP/M 3.0 90

Table des vecteurs standards du BDOS du CP/M 3.0 94

Instructions résidentes et utilitaires de DR 109

Utilitaires propres à AMSTRAD 122

Emulateur de Terminal 125

Utilitaires graphiques 127

SOMMAIRE

CHAPITRE V - LE LANGAGE LOGO **129**

Généralités 129
Commandes et mots-clés (primitives) du LOGO 2 130
Commandes et mots-clés propres au LOGO 3 (CPC6128) 148

CHAPITRE VI - CIRCUITS UTILISES **153**

Le Z80-SIO 153
Généralités 153
Interfaçage du SIO avec le CPC 153
Registres et programmation du Z80-SIO 154
Registres d'écriture (WR0 à WR7) 154/163
Registres de lecture (RR0 à RR2) 164/165

Le CTC 8253 166
Généralités 166
Interfaçage avec le CPC 166
Registre et programmation 167
REGISTRE MODE 167
Définition des modes 168
Ecriture dans un compteur 169
Lecture d'un compteur 170

Le FDC PD765A 171
Généralités 171
Programmation et registres du FDC 171
Différentes commandes 174
Registres d'état du FDC 765A 180

CHAPITRE VII - BROCHAGES ET CONNECTEURS **185**

Brochage du Z80-SIO 185
Brochage du CTC 8253 188
Brochage du PD765A 190
Connecteur du disque supplémentaire 193
Connecteur du RS232 195

CHAPITRE VIII - TRUCS ET ASTUCES **197**

AMSDOS : RSXs de lecture et d'écriture secteur 197
AMSDOS : Analyse de l'en-tête de fichier 201

SOMMAIRE

AMSDOS : Chargement des fichiers option P	203
AMSDOS : SUPERZAP	206
AMSDOS : Création d'un fichier fantôme	213
CP/M : Utilitaire de rappel CP/M 2.2 (OUF.COM)	214
CP/M : Conversion en AZERTY	215
CP/M : Mise en route automatique de l'imprimante	216
LOGO : Programmation des notes musicales	217
SCHEMA GUIDE	**218**
INDEX	**219**
CONSEILS DE LECTURE	**229**

PRESENTATION

Le présent ouvrage constitue un véritable répertoire pour les systèmes Amstrad avec disque. De l'architecture interne de l'AMSDOS et du CP/M en passant par le brochage, la structure et la programmation des circuits spécialisés, le logiciel interne et le LOGO, il recense tout et ce en l'agrémentant d'explications succinctes et de quelques exemples.

Du programmeur-analyste au simple curieux de l'informatique, chacun y trouvera les informations nécessaires à la bonne utilisation de son Amstrad.

AMSDOS

L'AMSDOS, acronyme anglo-saxon d'AMStrad Disk Operating System (Système d'exploitation sur disque pour Amstrad) est une extension du BASIC LOCOMOTIVE résidant dans une ROM (mémoire morte) externe. Le mot externe doit être pris au sens large car, dans les systèmes CPC664 et CPC6128, cette ROM est intégrée à l'intérieur du boitier.

L'AMSDOS se compose de deux parties principales. La première est constituée par l'interception et le déroutement des instructions et fonctions Basic de gestion de la cassette. La seconde se présente sous la forme d'extensions résidentes du système (RSX) qui ajoutent une série de nouvelles instructions totalement dédicacées à la gestion du lecteur de disque.

L'AMSDOS n'est pas un système d'exploitation très performant. Il ne possède que très peu de commandes et ne permet pas, en standard, la gestion des fichiers directs (RANDOM ACCESS). L'écriture de programmes de gestion sophistiqués ou nécessitant l'ouverture simultanée de plusieurs fichiers est impossible sans recourir à une série d'astuces peu dignes d'un programmeur d'applications. C'est pourquoi il faut lui préférer le CP/M pour l'élaboration et la maintenance d'applications sérieuses.

Cependant, l'AMSDOS est parfaitement adapté à la mémorisation de programmes de jeux ou d'utilitaires divers (assembleur, moniteur, ...).

En outre, les programmeurs à l'origine de ce système d'exploitation ont réalisé un véritable prodige dont bien des constructeurs devraient s'inspirer. Ce prodige, c'est la CONVIVIALITE TOTALE existant entre AMSDOS et CP/M. Autrement dit, les deux systèmes résidant sur la même disquette utilisent les mêmes formats de fichiers ou de répertoires (DIR) et des commandes de passage de l'un à l'autre sont prévues. Ainsi, le DDT livré avec CP/M peut servir de moniteur de mise au point pour des fichiers écrits sous AMSDOS. Encore une fois, bravo.

SPECIFICATIONS

Possibilités

- Commutation de fonctions de sauvegarde, chargement et gestion de fichiers séquentiels entre la cassette et le disque.
- Affichage du répertoire (DIRECTORY) et de l'espace occupé.
- Effacement et changement de nom de fichier.
- Sélection d'unité par défaut.

Nom de fichier

- Syntaxe générale :

  Numéro d'user numéro d'unité : nom . extension

- Numéro d'user est un nombre compris entre 0 et 15 et qui définit le répertoire utilisateur utilisé. Cette spécification est optionnelle.

- Le numéro d'unité peut être A ou B. Cette spécification est optionnelle. Si elle n'est pas spécifiée, c'est l'unité définie par défaut lors d'une commande ¦DRIVE, ¦A ou ¦B qui est utilisée.

- Le nom de fichier est composé de 1 à 8 caractères alphanumériques. Les blancs ne sont pas autorisés. Les seuls caractères spéciaux utilisables sont :

  ! " # $ & ' + - @ { } ^ ↑

  L'utilisation d'autres caractères (, ; : / ...) produira une erreur (BAD COMMAND).

  Certaines commandes (¦DIR, ¦ERA) peuvent utiliser les cartes de sélection (caractères ? et *). Reportez-vous à la description des cartes de sélection du chapitre réservé au CP/M.

- L'extension ou le type est composé de 1 à 3 caractères optionnels. Si l'extension est omise, une extension par défaut est créée par le système. Cette extension dépend du type de fichier.

Extensions par défaut

rien Extension vide créée par l'ouverture de fichier (OPENOUT) sans spécification d'extension.

$$$ Extension de fichier en cours de création (après l'ouverture et avant la fermeture).

BAS Extension créée par la sauvegarde d'un programme Basic. L'extension créée est identique pour les programmes protégés ou ASCII (options P ou A).

BIN Extension créée par la sauvegarde binaire (option B).

BAK Extension créée automatiquement par le système pour la recopie de l'ancien fichier lors d'une nouvelle sauvegarde. Cette faculté est très intéressante pour "rattraper les erreurs". Cependant, elle double la place occupée sur le disque. N'oubliez pas de procéder de temps à autre à un nettoyage au moyen de la commande ¦ERA,"*.BAK" (664-6128) ou A$="*.BAK" : ¦ERA,@A$ (464).

Les autres extensions sont soit réservées au CP/M (COM, SUB, ...) ou laissées au choix de l'utilisateur.

Formatage et copie

Les opérations de formattage et de copie sont impossibles sous AMSDOS. Il est nécessaire de passer sous CP/M (¦CPM) et d'utiliser les utilitaires correspondants (FORMAT, DISCOPY, COPYDISC et FILECOPY).

Formats des disquettes et du répertoire

L'AMSDOS utilise **trois formats de disquettes** et un format de répertoire complètement compatibles avec le CP/M. Nous vous invitons à vous reporter au chapitre traitant du CP/M 2.2 pour de plus amples informations sur le format des disquettes et du répertoire.

INSTRUCTIONS ET FONCTIONS BASIC MODIFIEES PAR AMSDOS

CAT

CAT

Lit le répertoire de la disquette courante et de l'utilisateur courant (*voir* USER). Les noms et les extensions de chaque fichier sont affichés par ordre alphabétique. Chaque entrée du répertoire est suivie de sa taille. A la fin de la liste, l'espace disponible sur l'unité courante est indiqué.
Attention : si votre espace disponible (*voir tome I*, fonction FRE) est inférieur à 4K (4096 octets), l'instruction CAT ne fonctionnera pas et le système affichera une erreur MEMORY FULL.

CHAIN

CHAIN "nom de fichier"[,N]

Charge un programme en provenance du disque dans la mémoire centrale en supprimant celui qui s'y trouve déjà puis l'exécute en commençant par la ligne N.
Si N n'est pas précisé, l'exécution débute à la ligne portant le plus petit numéro.

CHAIN MERGE

CHAIN MERGE "nom de fichier"[,N] [,DELETE n-m]

Réalise la fusion de deux programmes distincts. Si une ligne du programme existant porte le même numéro qu'une ligne du programme fusionné, c'est cette dernière que l'on retrouvera dans le programme résultant. A la suite de cette opération, le programme formé est exécuté en partant de la ligne N si ce nombre est spécifié ou, à défaut, en partant de la ligne portant le plus petit numéro. Enfin, il est possible de procéder à l'effacement d'une ou de plusieurs lignes du programme résidant initial au moyen de l'option DELETE n-m où n indique le numéro de la première ligne à effacer et m le numéro de la dernière ligne à effacer.
Remarque *:* un programme protégé (P) ne peut pas être fusionné par cette méthode.

CLOSEIN

CLOSEIN

Ferme le fichier ouvert en entrée (lecture) par la commande OPENIN.

CLOSEOUT

CLOSEOUT

Ferme le fichier ouvert en sortie (écriture) par la commande OPENOUT.

EOF

EOF

Cette fonction fournit -1 si la fin du fichier est atteinte en lecture ou s'il n'y a pas de fichier ouvert. Elle fournit 0 sinon.

INPUT

INPUT #9,liste de variables

Lecture de la (des) variable(s) spécifiée(s) dans le fichier ouvert en lecture (OPENIN).
Remarque *:* si le nombre de variables spécifiées dans la liste est supérieur au nombre de champs restant à lire dans le fichier, un message d'erreur EOF MET IN ... se produira. Cependant, si vous essayez de lire un champ numérique dans une variable alphanumérique ou vice-versa, il n'y aura pas de message d'erreur.

LINE INPUT

LINE INPUT #9,variable alphanumérique

Lecture de la variable spécifiée dans le fichier ouvert en lecture. La différence avec l'instruction précédente réside dans le fait que LINE INPUT attend un RETOUR CHARIOT (0DH) ou un marqueur de fin de fichier (1AH) comme séparateur de variable à l'intérieur du fichier. INPUT, par contre, considère le signe virgule ',' comme séparateur de variable.

LIST

LIST #9

Cette commande écrit sur le fichier ouvert en écriture (OPENOUT) le listing du programme courant en mémoire. Cette écriture est réalisée en mode fichier ASCII.

LOAD

LOAD "nom de fichier"[,adresse]

Charge en mémoire un programme Basic en provenance de la disquette, après avoir effacé le programme éventuel déjà présent en mémoire. Cette commande permet également le chargement de fichiers binaires. Dans ce cas, l'adresse optionnelle peut remplacer l'adresse absolue de chargement.

MERGE

MERGE "nom de fichier"

Comme LOAD, cette instruction charge un programme Basic en mémoire, mais le programme résidant n'est pas effacé. Si des numéros de lignes sont identiques, leur contenu devient celui du nouveau programme. Cette commande ne fonctionne pas avec les fichiers protégés.

OPENIN

OPENIN "nom de fichier"

Ouvre un fichier sur disque en lecture. Ce fichier doit exister dans le répertoire (DIRECTORY) de l'utilisateur courant, sinon un message d'erreur apparaît.

INSTRUCTIONS ET FONCTIONS BASIC MODIFIEES PAR AMSDOS

OPENOUT **OPENOUT "nom de fichier"**

Ouvre un fichier en écriture sur la disquette. Eventuellement, si le nom n'existe pas dans le répertoire, il est créé.

PRINT **PRINT #9,liste d'articles**

Ecrit dans un fichier ouvert en sortie (OPENOUT).

RUN **RUN "nom de fichier"**

Charge un programme Basic dans la mémoire (LOAD) et puis lance l'exécution à la ligne dont le numéro est le plus bas. Cette commande efface préalablement le programme résidant en mémoire.

SAVE **SAVE "nom de fichier"[,type][,début,long,PE]**

Sauve, sur disque, un programme Basic ou une zone mémoire sous le nom spécifié.

type : **P** Pour un programme protégé (le programme est encodé sur disque et ne peut plus être chargé).

A Sauvegarde le fichier en ASCII. Cette commande est identique à OPENOUT "nom de fichier" : LIST #9 : CLOSEOUT.

B Pour une sauvegarde binaire (portion de mémoire). Dans ce dernier cas, les paramètres début, long et PE sont nécessaires.

début indique l'adresse de départ de la mémoire.

long indique le nombre d'octets à sauvegarder.

PE indique le point d'entrée éventuel (ce dernier paramètre est optionnel).

WRITE **WRITE #9,liste d'articles**

Comme PRINT, cette instruction écrit dans le fichier ouvert en sortie. Cependant, les signes virgules ou guillemets ne servent pas de séparateurs et sont écrits dans le fichier.

IMPORTANT

La spécification "chaîne de caractères" oblige les possesseurs de CPC 464 à la manipulation suivante :

- définir la chaîne de caractères dans une variable alphanumérique et passer l'argument dans la commande au moyen de la fonction @ (@ est le VARPTR ou pointeur de variable).

Exemple : X$="A" : ¦DRIVE,@X$.

Les possesseurs de CPC 664 ou CPC 6128 peuvent indiquer la chaîne de caractères directement dans leur commande : **¦DRIVE,"A".**

Cette restriction est due à une erreur dans la ROM des CPC 464.

¦A **¦A**

Sélectionne l'unité de disque A par défaut. Cette commande est équivalente à ¦DRIVE avec A comme paramètre. Si le disque présent dans le lecteur A est illisible, la commutation d'unité n'aura pas lieu.

¦B **¦B**

Sélectionne l'unité de disque B par défaut. Les mêmes remarques que pour ¦A s'imposent.

¦BASIC **¦BASIC**

Réinitialise le Basic. Cette instruction est utilisée par l'utilitaire CP/M AMSDOS.COM.

¦CPM **¦CPM**

Charge le CP/M en lieu et place de l'AMSDOS. Cette instruction réalise un démarrage à froid du CP/M (COLD BOOT).

Remarque *:* l'utilitaire AMSDOS.COM réalise la fonction inverse sous CP/M.

¦DIR **¦DIR[,"chaîne de caractères"]**

Réalise l'impression sur écran du catalogue (DIRECTORY) au format CP/M. Cette commande accepte une chaîne de caractères comme paramètre. Pour une description complète des paramètres, reportez-vous à la description de la commande DIR dans le chapitre réservé au CP/M.

INSTRUCTIONS EXTERNES

¦DISC **¦DISC**

Cette commande renvoie les entrées sorties vers le disque en lieu et place de la cassette.

Lors de l'initialisation de la machine, une commande équivalente à ¦DISC est réalisée par défaut.

Tous les vecteurs situés entre BC7A et BC9B sont modifiés.

¦DISC.IN **¦DISC.IN**

Cette commande opère un changement de direction des entrées (lecture) vers le disque, sans modifier l'état des sorties.

Les vecteurs modifiés sont : BC77, BC7A, BC7D, BC80, BC83, BC86, BC89 et BC9B.

¦DISC.OUT **¦DISC.OUT**

Cette commande opère un changement de direction des sorties (écriture) vers le disque, sans modifier l'état des entrées.

Les vecteurs modifiés sont : BC8C, BC8F, BC92, BC95 et BC98.

¦DRIVE **¦DRIVE,"chaîne de caractères"**

Cette commande positionne le lecteur par défaut. La chaîne de caractères peut être "A" ou "B". Les mêmes remarques que pour ¦A et ¦B s'imposent.

¦ERA **¦ERA,"chaîne de caractères"**

Cette commande est identique à la commande ERA du CP/M. Elle permet l'effacement d'un ou de plusieurs fichiers. La chaîne de caractères spécifie le ou les fichiers à effacer. Une description complète de la commande ERA se trouve au chapitre réservé au CP/M.

¦REN **¦REN,"chaîne de car1","chaîne de car2"**

Cette commande, identique à la commande REN du CP/M, rebaptise le fichier nommé "chaîne de car2" en "chaîne de car1". Les noms doivent être indiqués en clair. Les deux fichiers doivent se trouver sur le même disque. Ils peuvent toutefois se trouver chez des USER différents.

¦TAPE **¦TAPE**

Cette commande opère un changement de direction des entrées et des sorties vers la cassette.

La liste des vecteurs modifiés est identique à celle de ¦DISC.

¦TAPE.IN **¦TAPE.IN**

Cette commande opère un changement de direction des entrées (lecture) vers la cassette sans modifier l'état des sorties.

La liste des vecteurs modifiés est identique à celle de ¦DISC.IN.

¦TAPE.OUT **¦TAPE.OUT**

Cette commande opère un changement de direction des sorties (écriture) vers la cassette sans modifier l'état des entrées.

La liste des vecteurs modifiés est identique à celle de ¦DISC.OUT.

¦USER **¦USER,numéro (0 à 15)**

Positionne le numéro d'utilisateur. Cette fonction est identique à la commande USER sous CP/M.

MESSAGES D'ERREURS DE L'AMSDOS

Bad command
Soit il y a une erreur de syntaxe, soit la commande spécifiée n'existe pas.

'nom de fichier' already exist
L'utilisateur a donné, comme nouveau nom de fichier pour l'instruction ¦REN, un nom de fichier qui existe déjà.

'nom de fichier' not found
Le nom de fichier spécifié n'existe pas (OPENIN, ¦ERA, ¦REN ...).

Drive A ou B : directory full
Les 64 entrées possibles du directory sont occupées (on ne peut spécifier plus de 64 noms sur une face de disquette).

Drive A ou B : disc full
Le disque spécifié est plein (il n'y a plus un seul secteur libre).

Drive A ou B : disc changed, closing "nom de fichier"
Le disque a été changé alors qu'un fichier était ouvert.

"nom de fichier" is read only
Essai d'effacement (¦ERA) ou de changement de nom (¦REN) sur un fichier défini comme READ ONLY (lecture seulement).

Drive A ou B : disc missing
Soit il n'y a pas de disquette dans le lecteur, soit la disquette est mal placée.

Drive A ou B : disc is write protected
La disquette est protégée en écriture. Protection matérielle réalisée par le petit ergot sur la disquette (support) elle-même.

Drive A ou B : read fail
La lecture est impossible. Nous vous conseillons de reformater votre disquette après sauvegarde des fichiers importants. Ce message peut annoncer une panne mécanique de votre lecteur de disque ou une destruction du revêtement magnétique de votre disquette (support).

Drive A ou B : write fail
Ecriture impossible. Si votre disquette est correctement formatée, reportez-vous aux remarques faites ci-dessus pour read fail.

Failed to load CP/M
Erreur rencontrée durant le passage au CP/M (¦CPM). Votre disquette est endommagée ou ne contient pas le CP/M (disquettes données seules).

Remarque *:* les cinq derniers messages d'erreur sont suivis d'une question "Retry, Ignore or Cancel" litteralement "Nouvel essai, Ignorer l'erreur, Avorter la commande".

Le système attend une réponse :

R réexécute la commande ou la tentative de lecture ou d'écriture.

I permet à l'ordinateur de continuer son action (résultat non garanti, soyez prudent).

C annule l'opération en cours.

UTILITAIRES (6128 seulement)

Les utilitaires utilisables sous AMSDOS sont peu nombreux, ils tiennent dans un programme Basic appelé BANKMAN.BAS. Ce programme réside sur la face 1 de la disquette système et il doit être lancé à l'initialisation de la machine. Il rajoute des commandes **RSX** au Basic standard. Ces commandes portent sur la gestion de la seconde banque (BANK) de 64 K.

Instructions ajoutées par le lancement de BANKMAN.BAS

Instructions de sauvegarde d'écran

La mémoire écran des CPC comportant 16 K, la mémoire supplémentaire de 64 K dont dispose le CPC 6128 permet la sauvegarde de quatre écrans supplémentaires.

L'écran standard, situé dans le BANK 1 de l'adresse C000H à FFFFH, est appelé ECRAN 1. Les quatre autres sont appelés ECRAN 2 à ECRAN 5.

¦SCREENCOPY[,numéro de bloc],écran destination,écran origine

Cette commande permet la copie d'un écran ou d'un morceau (1/64) d'écran vers un autre.

- *numéro de bloc* est un paramètre optionnel compris entre 0 et 63 qui indique quel bloc de 256 octets il faut copier (l'écran est divisé en 64 blocs de 256 octets). Cette option permet de réaliser la copie en plusieurs passes et de continuer à effectuer d'autres instructions entre la copie de chaque bloc.
- *écran destination* est un nombre compris entre 1 et 5 qui indique le numéro d'écran qui recevra le ou les blocs copiés.
- *écran origine* est un nombre compris entre 1 et 5 qui indique le numéro d'écran SOURCE d'où part la copie.

Ainsi, ¦SCREENCOPY,1,2 est identique à FOR I=0 to 63 : ¦SCREENCOPY,I,1,2 : NEXT I.

¦SCREENSWAP[,numéro de bloc],numéro d'écran,numéro d'écran

Cette commande permet l'échange de deux écrans ou de deux morceaux d'écran.

- *numéro de bloc* est un paramètre optionnel défini comme pour ¦SCREENCOPY.
- *numéro d'écran* est un nombre compris entre 1 et 5 qui représente l'un des numéros d'écran à intervertir.

Remarques :
- Seul l'écran 1 est affiché sur le moniteur. C'est donc lui qui constitue la destination principale finale.
- Le registre pointeur de l'adresse de début d'écran étant continuellement modifié par le logiciel, il convient de s'assurer de sa position lors de l'échange d'écran. En principe, l'utilisation de la commande **MODE n** remet le pointeur sur le début de la mémoire écran (C000H).

Instructions de gestion du disque virtuel

Les 64 K supplémentaires peuvent être considérés comme disque virtuel.

Un disque virtuel est un disque résidant en mémoire à accès très rapide.

Ce disque est composé d'un seul fichier composé d'enregistrements de longueur fixe comprise entre 1 et 255 octets. Les enregistrements sont accessibles en mode direct (accès par le numéro d'enregistrement).

Ce fichier est réservé aux données et n'est pas utilisable pour la sauvegarde des programmes.

¦BANKOPEN,longueur d'enregistrement

Cette instruction permet d'ouvrir le fichier virtuel en fixant la taille de chaque enregistrement. Le paramètre longueur d'enregistrement doit être compris entre 1 et 255. Il peut donc y avoir entre 65536 et 256 enregistrements.

Remarque *:* cette instruction n'efface pas le contenu du fichier virtuel.

¦BANKWRITE,@code,chaîne[,numéro d'enregistrement]

Cette instruction écrit le contenu de la chaîne de caractères spécifiée dans l'enregistrement de numéro spécifié.

- *@code* est le pointeur d'une variable entière (utilisez % ou DEFINT). Cette variable doit être définie et affectée avant l'utilisation de la commande.

  Exemple : A%=0 : ¦BANKWRITE,@A%,...
  Cette variable est utilisée pour transmettre, à l'issue de la commande, un compte-rendu d'erreur. Si l'écriture est correcte, la variable vaut le numéro de l'enregistrement écrit. Si la fin de fichier est atteinte, la variable vaut -1, et si une erreur de commutation entre les blocs mémoires se produit, la variable vaut -2.

UTILITAIRES (6128 seulement)

- *chaîne* représente une variable alphanumérique ou une chaîne de caractères entre guillemets.
- *numéro d'enregistrement* est un paramètre optionnel. Si ce paramètre est omis, c'est le numéro d'enregistrement courant qui est utilisé. Le numéro courant est égal au dernier numéro d'enregistrement utilisé, augmenté de 1.

¦**BANKREAD**,@code,variable de chaîne[,numéro d'enregistrement]

Cette commande lit l'enregistrement spécifié et le pousse dans la variable alphanumérique spécifiée.

- @*code* est défini comme ci-dessus.
- *variable de chaîne* est une variable alphanumérique ($ ou DEFSTR).
- *numéro d'enregistrement* est défini comme ci-dessus.

¦**BANKFIND**,@code,chaîne[,départ][,arrivée]

Cette commande recherche la chaîne spécifiée dans le fichier virtuel.

- @*code* est le pointeur d'une variable recevant le compte-rendu d'erreur. Cette variable doit être entière et définie. Elle contient le numéro d'enregistrement si la chaîne spécifiée est trouvée, -1 si la fin de fichier est atteinte, -2 si une erreur de commutation s'est produite et -3 si l'enregistrement n'est pas trouvé.
- *chaîne* représente une chaîne de caractères entre guillemets ou une variable alphanumérique.
- *départ* est un paramètre optionnel indiquant le numéro d'enregistrement de départ de la recherche.
- *arrivée* est un paramètre optionnel indiquant le numéro d'enregistrement où doit se terminer la recherche.

Remarque *:* la chaîne recherchée peut contenir des JOKERS. Autrement dit, des caractères qui ne sont pas comparés et donc considérés comme égaux à la position spécifiée. Ces JOKERS sont indiqués par le code ASCII 0 : CHR$(0).

Remarques : les lettres **A**, **F**, **BC**, **DE**, **HL** et **IX** identifient les registres du processeur Z80. **CARRY** et **ZERO** sont deux des indicateurs du registre F. **CE** et **CS** sont les abréviations respectives de Condition d'Entrée et Condition de Sortie.

ROUTINES CASSETTE MODIFIEES POUR LE DISQUE

Adresse — *Fonctions*

BC77 Ouvre un fichier en lecture et lit le premier enregistrement.

CE : B contient la longueur du nom du fichier, HL contient l'adresse où l'on trouve le nom du fichier, DE contient l'adresse du tampon de réception (2 K).

CS : si l'ouverture réussit, le CARRY est vrai et le ZERO est faux, A contient le type de fichier, HL contient l'adresse d'un tampon contenant l'en-tête de fichier, DE contient l'adresse de départ du programme (dans le cas d'un programme Basic ou d'un fichier binaire) et BC contient la longueur du fichier (programme et binaire). En outre, IX est modifié.

Si le fichier n'existe pas, le CARRY est faux et le ZERO est vrai, A contient un numéro d'erreur, BC, DE, HL et IX sont altérés.

Si un autre fichier est déjà ouvert, alors le CARRY est faux et le ZERO est faux. A contient OEH et BC, DE, HL et IX sont modifiés.

BC7A Ferme "proprement" le fichier ouvert en lecture.

Pas de **CE**.

CS : si la fermeture est correcte, le CARRY est vrai, le ZERO est faux et A est modifié.

Si le fichier n'est pas ouvert, le CARRY est faux, le ZERO est faux et A contient OEH.

Adresse *Fonctions*

Si la fermeture est impossible, le CARRY est faux, le ZERO est vrai et A contient le numéro de l'erreur.

Dans les trois cas, BC, DE, HL et F sont modifiés.

BC7D Ferme "violemment" le fichier ouvert en lecture.

Pas de **CE.**

CS : AF, BC, DE et HL sont modifiés.

BC80 Lecture d'un caractère hors du fichier ouvert.

Pas de **CE.**

CS : si la lecture a eu lieu correctement, A contient le caractère lu, le CARRY est vrai et le ZERO est faux.
Si le fichier n'est pas ouvert ou si la fin de fichier est atteinte, le CARRY et le ZERO sont faux et A contient un numéro d'erreur (0EH, 0FH ou 1AH).
Dans tous les autres cas, A contient le numéro de l'erreur, le CARRY est faux et le ZERO est vrai.
De toute façon, IX est modifié.

BC83 Lecture complète du fichier en mémoire.

CE : HL contient l'adresse mémoire de stockage du fichier.

CS : si la lecture est correcte, le CARRY est vrai, le ZERO est faux, HL contient le point d'entrée extrait de l'en-tête.
Si le fichier n'est pas ouvert, le CARRY et le ZERO sont faux et A contient 0EH.
Dans les autres cas, le CARRY est faux, le ZERO est vrai et A contient le numéro de l'erreur.
Dans tous les cas, BC, DE, HL, IX et AF sont modifiés.

BC86 Remet le dernier caractère lu dans le tampon de lecture, le caractère sera relu lors de la lecture suivante.

Pas de **CE.**

CS : tous les registres sont préservés.

BC89 Teste si la fin de fichier est atteinte (EOF).

Pas de **CE.**

CS : si la fin de fichier n'est pas trouvée, le CARRY est vrai, le ZERO est faux et A est modifié.
Si la fin de fichier est atteinte, le CARRY et le ZERO sont faux et A contient le code d'erreur (0EH, 0FH, 1AH).

Adresse *Fonctions*

Dans les autres cas, le CARRY est faux, le ZERO est vrai et A contient le code d'erreur.
Dans tous les cas, IX et F sont altérés.

BC8C Ouvre un fichier en sortie (écriture).

CE : B contient la longueur du nom du fichier, HL contient l'adresse du nom du fichier et DE contient l'adresse d'un tampon de 2K.

CS : si l'ouverture réussit, le CARRY est vrai, le ZERO est faux et HL contient l'adresse d'un tampon contenant l'en-tête de fichier.
Si le fichier est déjà ouvert, le CARRY et le ZERO sont faux et A contient OE.
Dans les autres cas, le CARRY est faux, le ZERO est vrai et A contient le code d'erreur.
Dans tous les cas, BC, DE, HL et IX sont modifiés.

BC8F Ferme "proprement" le fichier ouvert en écriture.

Pas de **CE**.

CS : si la fermeture est correcte, le CARRY est vrai, le ZERO est faux et A est modifié.
Si le fichier n'est pas ouvert, le CARRY et le ZERO sont faux et A contient OEH.
Dans les autres cas, le CARRY est faux, le ZERO est vrai et A contient le numéro de l'erreur.
Dans tous les cas, BC, DE, HL et IX sont modifiés.

BC92 Ferme "violemment" le fichier ouvert en écriture.

Pas de **CE**.

CS : AF, BC, DE et HL sont modifiés.

BC95 Ecriture d'un caractère dans un fichier (si le tampon est plein, il est écrit sur disque avant l'insertion du nouveau caractère).

CE : A contient le caractère à écrire.

CS : si l'écriture s'est déroulée correctement, le CARRY est vrai, le ZERO est faux et A est modifié.
Si le fichier n'est pas ouvert, le CARRY et le ZERO sont faux et A contient OEH.
Dans les autres cas, le CARRY est faux, le ZERO est vrai et A contient le numéro de l'erreur.
Dans tous les cas, IX est modifié.

ROUTINES CASSETTE MODIFIEES POUR LE DISQUE

Adresse — *Fonctions*

BC98 Ecriture directe du contenu d'une partie de la mémoire dans le fichier.

CE : HL contient l'adresse mémoire de départ, DE contient la longueur (nombre d'octets à écrire), BC contient le point d'entrée éventuel et A contient le type de fichier (ces données sont écrites dans l'en-tête du fichier).

CS : comme ci-avant (BC95) mais en plus, BC, DE, HL et IX sont modifiés.

BC9B Génère le DIRECTORY (Catalogue) du disque courant et de l'USER courant. Le directory est affiché par ordre alphabétique.

CE : DE contient l'adresse d'un tampon de 2K disponible.

CS : si le directory s'est bien déroulé, le CARRY est vrai, le ZERO est faux et A est modifié.
Sinon, le CARRY est faux, le ZERO est vrai et A contient le numéro d'erreur.
Dans les deux cas, BC, DE, HL et IX sont modifiés.

La ROM contient neuf routines que l'AMSDOS partagé avec le CP/M pour le paramétrage et l'accès direct aux disques.

L'appel de ces commandes est particulier. Il ne peut se réaliser qu'au moyen de l'assembleur et suivant un processus très précis :

- Charger HL avec l'adresse où l'on trouve le numéro de la commande choisie (ce numéro doit être augmenté de 80H pour positionner le bit 7 à 1).
- Appeler la routine KL-FIND-COMMAND à l'adresse BCD4H (vous trouverez une description complète de cette routine à la page 100 du tome 1 du présent ouvrage). La routine vous fournit l'adresse de branchement réel dans HL et le numéro de la ROM dans C.
  Remarque *:* à l'issue de la routine, si le CARRY est faux, la commande choisie n'a pas été trouvée.
- Appeler la routine KL-FAR-PCHL à l'adresse 001BH *(tome 1, page 109)* après avoir rempli les conditions d'entrée.

Exemple : appel de la commande 1.

```
                                ORG     #A000
A000    21 0D A0                LD      HL,COMMAND
A003    CD D4 BC                CALL    #BCD4
A006    D0                      RET     NC      ; pas trouvé
A007    3E FF                   LD      A,#FF   ; message inhibé
A009    CD 1B 00                CALL    #1B
A00C    C9                      RET
A00D    81      COMMAND:        DEFB    #81     ; 1 + 80H
```

Commandes

1 Autorise ou inhibe l'affichage des messages d'erreur concernant le disque.

CE : A=0 autorise l'affichage des messages.
A=FFH (255) inhibe l'affichage des messages.

CS : A contient l'ancienne valeur, F et HL sont modifiés.

2 Initialise les paramètres du disque.

CE : HL pointe sur la table contenant les paramètres.

CS : AF, BC, DE et HL sont modifiés.

VECTEURS PROPRES AU DISQUE

Format de la table :

Octets 0 et 1 : temps d'accès après le démarrage du moteur exprimé en cinquantièmes de seconde (valeur par défaut 50).

Octets 2 et 3 : temps d'arrêt du moteur après le dernier accès exprimé en cinquantièmes de seconde (valeur par défaut 250).

Octet 4 : temps de retombée du courant d'écriture exprimé en dizaines de microsecondes (valeur par défaut 175).

Octet 5 : temps de positionnement de la tête exprimé en millisecondes (valeur par défaut 15).

Octet 6 : temps de déplacement d'une piste à l'autre exprimé en millisecondes (valeur par défaut 12).

Octet 7 : temps de retrait de la tête (facteur utilisé par le FDC 765).

Octet 8 : temps de chargement de la tête et indicateur de DMA (bit 0).

3 Sélectionne un format particulier.

CE : A contient l'indicateur du format.
41H pour le format système.
C1H pour le format données seules.
01H pour le format IBM.
E contient le numéro du lecteur (0 pour le lecteur A et 1 pour le lecteur B).

CS : AF, BC, DE et HL sont modifiés.

4 Lecture d'un secteur en mémoire.

CE : HL contient l'adresse d'un tampon, E contient le numéro du lecteur, D contient le numéro de piste et C contient le numéro du secteur.

CS : si le secteur est bien lu, le CARRY est vrai, A vaut 0 et HL est sauvegardé.
Si la lecture échoue, le CARRY est faux, A contient le numéro d'erreur et HL pointe sur le tampon contenant le statut de l'erreur.

5 Ecriture d'un secteur sur disque.

CE : HL pointe sur le tampon qui contient le secteur, E contient le numéro du lecteur (0 ou 1), D contient le numéro de piste et C contient le numéro de secteur.

CS : si l'écriture est correcte, le CARRY est vrai, A vaut 0 et HL est préservé.

Si l'écriture échoue, le CARRY est faux, A contient le numéro d'erreur et HL contient l'adresse du tampon contenant le statut d'erreur.

6 Formatage d'une piste.

CE : HL pointe sur le tampon qui contient les informations d'en-tête, E contient le numéro du disque (0 ou 1) et D contient le numéro de piste.

CS : si le formatage réussit, le CARRY est vrai, A vaut 0 et HL est préservé.
Si le formatage échoue, le CARRY est faux, A contient le numéro d'erreur et HL contient l'adresse du tampon contenant le statut d'erreur.

7 Déplace la tête du disque sur une piste spécifiée.

CE : E contient le numéro du disque et D contient le numéro de la piste.

CS : si le déplacement s'est effectué correctement, le CARRY est vrai, A contient 0 et HL est préservé.
Sinon, le CARRY est faux, A contient le numéro d'erreur et HL pointe sur le tampon contenant le statut d'erreur.

8 Fournit le statut du disque spécifié.

CE : A contient le numéro du disque.

CS : si le CARRY est vrai, A contient le statut du disque (voir ci-dessous).
Si le CARRY est faux, HL contient l'adresse du tampon contenant le statut d'erreur dont le second octet est défini comme ci-dessous.

STATUT
BIT 7 : inutilisé.
BIT 6 : état de la protection en écriture.
BIT 5 : disque prêt ou non-prêt.
BIT 4 : indicateur de position en piste 0.
BIT 3 : inutilisé.
BIT 2 : 0.
BIT 1 : 0.
BIT 0 : numéro du disque sélectionné.

9 Positionne le nombre d'essais lors d'une opération de lecture, d'écriture ou de formatage.

CE : A contient la valeur du compteur d'essais.

CS : A contient l'ancienne valeur du compteur, HL et F sont modifiés.

PRINCIPALES ADRESSES DE LA ROM DISQUE

Remarque *:* contrairement aux ROMs BASIC et BIOS, la ROM disque est identique sur les systèmes 464, 664 et 6128. Les adresses signalées ici sont donc valables pour tous les systèmes.

C000 Début de la ROM : type et version de la ROM.
C004 Adresse de la table d'instructions.
C006 Bloc de saut des instructions d'extensions de l'AMSDOS.
C033 Bloc de saut des instructions propres au disque.
C04E Bloc de saut du CP/M.
C072 Table des instructions.
C0B6 Table des instructions propres au disque (1 à 9).
C0C0 Sauvegarde contexte interruption.
C0FA Autorisation sauvegarde deuxième jeu de registres.
C132 Inhibition sauvegarde deuxième jeu de registres.
C17F Bloc de saut du BIOS.
C1B2 Entrée de ¦CPM (démarrage à froid du CP/M).
C1BC Entrée commande ¦CPM ROM.
C1DC Démarrage à froid CP/M traitement principal.
C224 Traitement erreur de chargement du BOOT.
C22B Démarrage à chaud CP/M.
C2AC Teste si le secteur lu est vide.
C2BE Traitement démarrage à chaud.
C2C3 Traitement CONIN.
C2C8 Traitement CONOUT.
C2CD Traitement état imprimante.
C2D2 Traitement PRINTER OUTPUT.
C2D7 Traitement PUNCHER.
C2DC Traitement READER.
C2E1 Traitement état console.
C2E9 Recherche de la piste 0 (SEEK).
C2F2 Sélection de l'unité disque.
C2F7 Lire secteur.
C2FC Ecrire secteur.
C313 Tester clavier.
C326 Lire un caractère du clavier.
C348 Attente de frappe de caractère au clavier.
C35B Sortir un caractère sur l'écran.
C379 Teste si imprimante BUSY (occupée).
C37F Sort un caractère sur l'imprimante.
C389 Initialisation de l'interface série.
C3AE Initialisation du 8253 (vitesse).
C3BD Initialisation du SIO.
C3DB Canal A test si tampon rempli (BUFFER FULL).
C3E3 Canal B test si tampon rempli (BUFFER FULL).
C3F7 SIO canal A lire un caractère.
C3FF SIO canal B lire un caractère.
C420 SIO positionne le DTR (réception permise).
C424 SIO positionne le DTR (réception interdite).
C445 SIO envoie un caractère sur le canal A.

PRINCIPALES ADRESSES DE LA ROM DISQUE

C44B SIO envoie un caractère sur le canal B.
C46A Détermination de l'I/O byte.
C47D Table état de la console.
C486 Table entrée console.
C48F Table sortie console.
C498 Table état imprimante.
C4A1 Table sortie imprimante.
C4AA Table PUNCHER.
C4B3 Table état READER.
C4BC Table READER lecture.
C4D3 Test si CONTROL C.
C4F0 Sélection d'unité disque.
C51F Positionnement piste 0.
C529 Envoyer numéro d'enregistrement au FDC.
C52E Ecrire enregistrement.
C54C Lire enregistrement.
C55D Lire secteur identification.
C56C Détermination du format en fonction du secteur ID.
C581 Déterminer formatage du disque (routine 3).
C5C0 Table format IBM.
C5CA Table format données seules.
C5DD Initialisation DPH, DPB ... (voir CP/M).
C603 Fixe le nombre de lectures (routine 9).
C60D Spécifie les caractéristiques du lecteur (routine 2).
C630 Détermine l'état du lecteur (routine 8).
C64E Ecriture secteur (routine 5).
C652 Formatage d'une piste (routine 6).
C666 Lecture d'un secteur (routine 4).
C67C Programmation du FDC.
C6C1 Programme principal écriture/lecture/formatage.
C6FF Positionnement sur le numéro de piste contenu dans le registre D.
C722 Impression du message READ FAIL.
C763 Positionnement piste (routine 7).
C7C7 Routine de temporisation et de lecture du statut du FDC (délai = (A * 12) + 16 ms).
C7E0 Boucle d'attente (délai = A millisecondes).
C8A2 Calcul numéro secteur effectif.
C8B6 Transfert de l'enregistrement dans le tampon.
C8C7 Transfert enregistrement.
C8F9 Lecture du registre état du FDC (DISK READY).
C907 Lecture du registre état du FDC (DISK PROTECT).
C9D6 Routine d'activation du compteur TICK.
C9F4 Organisation des paramètres d'en-tête disque.
CA43 Paramètres standards du DPB.
CA5C Charger dans l'accumulateur une valeur de la DPB située à un OFFSET contenu dans A (A890H+(Disque*40H)+A).
CA72 Message ON/OFF (routine 1).
CA90 BC=BC+IY.

PRINCIPALES ADRESSES DE LA ROM DISQUE

CA98 DE=DE+IY.
CA9F HL=HL+IY.
CAA6 Conversion minuscule → majuscule.
CAAF Remplir BC octets de la mémoire avec 00 depuis l'adresse contenue dans DE.
CAB8 Sortir message d'erreur dont le numéro est contenu dans A et tester la réponse (R,I,C).
CAFE Impression du message.
CB86 Table des messages d'erreurs.
CCA0 Redirection des vecteurs cassettes vers le disque.
CCD1 ¦DISC.
CCD5 ¦DISC.IN.
CCE4 ¦DISC.OUT.
CCFD ¦TAPE.
CD01 ¦TAPE.IN.
CD18 ¦TAPE.OUT.
CD4C Bloc de saut des routines interceptées (CAS ↔ DISC).
CDAF Impression du message "BAD COMMAND".
CDDA ¦A.
CDDD ¦B.
CDE4 ¦DRIVE.
CDFE ¦USER.
CE48 Copie du nom de fichier dans le bloc d'en-tête (OPENIN).
CE57 Copie du nom de fichier dans le bloc d'en-tête (OPENOUT).
CEAF CAS IN OPEN (BC77).
CF37 CAS OUT OPEN (BC8C).
CF64 CAS IN CHAR (BC80).
CFF5 CAS IN DIRECT (BC83).
D065 CAS TEST EOF (BC89).
D069 CAS RETURN (BC86).
D08F CAS OUT CHAR (BC95).
D0D8 CAS OUT DIRECT (BC98).
D1B6 CAS IN CLOSE (BC7A).
D1BC CAS IN ABANDON (BC7D).
D1C2 CAS OUT ABANDON (BC92).
D1D8 CAS OUT CLOSE (BC8F).
D25C Routine de codage/décodage des fichiers protégés par OU EXCLUSIF.
D281 Table de codage.
D299 Table des extensions par défaut ($$$,BAS,BAK,BIN).
D42E ¦DIR.
D48A ¦ERA.
D4C4 ¦REN.
D513 CATALOG.
D676 Recherche d'un fichier dans le répertoire et détermination de sa taille (nombre de blocs).
D6A2 Détermination du nombre de fichiers sur le disque.
D7BB Rechercher une entrée libre dans le répertoire.
D7D8 Rechercher un nom indiqué dans le répertoire.

PRINCIPALES ADRESSES DE LA ROM DISQUE

D9E8 Lire l'enregistrement dont le numéro est dans DE et le pousser dans le tampon d'enregistrement.
D9F3 Ecrire l'enregistrement dont le numéro est dans DE.
DA06 Calculer le numéro de piste et le numéro de secteur en fonction du numéro d'enregistrement.
DA3F Charge le contenu du DPH OFFSET A dans HL.
DBB2 Table des caractères interdits dans les noms de fichiers.
DBDF Transfert de 32 octets (taille d'une entrée du répertoire de HL vers DE).
DBEB Divise le contenu de HL par A exposant 2.
DBF3 Compare HL et DE.
DBF9 Charge le contenu de l'adresse pointée par HL dans HL.
DBFF Fin de la partie gestion disque de la ROM.
DC00 Espace libre jusqu'à DFFF.
E000 Espace LOGO jusqu'à FFFF.

TABLE DES PRINCIPALES VARIABLES SYSTEME

Adresse	*Taille*	*Fonction*
A700	1	Lecteur appelé.
A701	1	USER appelé.
A702	1	Lecteur actif.
A703	2	Pointeur vers le DPH.
A705	1	Sémaphore d'indication de fichier ouvert.
A706	2	Pointeur de pile.
A708	1	Sémaphore OPENIN (numéro d'unité ou FF).
A709	1	Numéro d'USER pour OPENIN.
A70A	11	Nom du fichier + extension pour OPENIN.
A715	1	Numéro de la première entrée ou de l'extension pour OPENIN.
A718	1	Nombre d'enregistrements de l'extension pour OPENIN.
A719	16	Numéros des blocs de l'extension (OPENIN).
A729	2	Nombre d'enregistrements lus (OPENIN).
A72C	1	Comme A708, mais pour OPENOUT.
A72D	1	Comme A709, mais pour OPENOUT.
A72E	11	Comme A70A, mais pour OPENOUT.
A739	1	Comme A715, mais pour OPENOUT.
A73C	1	Comme A718, mais pour OPENOUT.
A73D	16	Comme A719, mais pour OPENOUT.
A74D	2	Comme A729, mais pour OPENOUT.
A750	1	Indicateur CAS IN CHAR (1) ou CAS IN DIRECT (2).
A751	2	Vecteur sur début du tampon lecture de 2 K.
A753	2	Vecteur sur le caractère courant dans le tampon de lecture.
A755	1	Numéro USER pour fichier en lecture.
A756	15	Nom du fichier en lecture + extension + 00.
A765	1	Numéro du bloc courant du fichier en lecture.
A766	1	Numéro du dernier bloc du fichier en lecture.
A767	1	Type de fichier en lecture.
A768	2	Longueur des données en lecture.
A76A	2	Emplacement des données en lecture.
A76C	1	Numéro du premier bloc du fichier en lecture.
A76D	2	Longueur logique du fichier en lecture.
A76F	2	Point d'entrée.
A795	3	Compteur du nombre de caractères lus.
A79A	1	Indicateur CAS OUT CHAR (1) ou CAS OUT DIRECT (2).
A79B	2	Vecteur qui pointe sur le début du tampon du fichier en écriture.
A79D	2	Vecteur qui pointe sur le caractère courant du fichier en écriture.
A79F	1	Numéro d'USER du fichier en écriture.
A7A0	15	Nom du fichier en écriture + extension + octets à 00.
A7AF	1	Numéro de bloc courant du fichier en écriture.

TABLE DES PRINCIPALES VARIABLES SYSTEME

Adresse	*Taille*	*Fonction*
A7B0	1	Numéro du dernier bloc du fichier en écriture.
A7B1	1	Type de fichier en écriture.
A7B2	2	Longueur des données en écriture.
A7B4	2	Emplacement des données en écriture.
A7B6	1	Numéro du premier bloc en écriture.
A7B7	2	Longueur logique du fichier en écriture.
A7BD	2	Longueur du bloc de données pour la routine CAS OUT DIRECT.
A7E4	128	Tampon provisoire d'enregistrement.
A874	26	Tampon pour les vecteurs cassettes.

Pour la signafication des paramètres qui suivent, reportez-vous aux explications sur le *CP/M du chapitre III*.

De **A890** à **A8CE**, on trouve les valeurs de la DPB, de la CSA et de l'ALT du disque A.

De **A8D0** à **A90E**, on trouve les mêmes tables pour le disque B.

A890	2	SPT.
A892	1	BSH.
A893	1	BLM.
A894	1	EXM.
A895	2	DSM.
A897	2	DRM.
A899	2	ALO et AL1.
A89B	2	CKS.
A89D	2	OFF.
A89F	1	FSC.
A8A0	1	PST.
A8A1	1	GPS.
A8A2	1	GPT.
A8A3	1	FLB.
A8A4	1	BPS.
A8A5	1	RPS.
A8A6	1	BCT.
A8A7	1	FTO.
A8A8	1	FLG.
A8A9	16	CSA. 16 octets avec les valeurs de contrôle pour le lecteur A.
A8B9	22	ALT. Table d'affectation des blocs du lecteur A.
A8D0	62	Table du disque B.

De **A910** à **A91F**, on trouve la table DPH du disque A.

De **A920** à **A92F**, on trouve la même table pour le disque B.

A910	2	XLT inutilisé.
A912	2	Numéro de piste courante.

TABLE DES PRINCIPALES VARIABLES SYSTEME

Adresse	*Taille*	*Fonction*
A914	2	Numéro de secteur courant.
A916	2	Numéro de répertoire.
A918	2	Pointeur sur tampon de 128 octets.
A91A	2	Pointeur sur DPB.
A91C	2	Pointeur sur valeur de contrôle.
A91E	2	Pointeur sur table d'affectation.
A920	16	Table pour le lecteur B.
A930	128	Tampon de 128 octets.
A9B0	256	Tampon de 256 octets.
BE40	2	Vecteur sur la DPH du lecteur A.
BE42	2	Vecteur sur la DPB du lecteur A.
BE44	2	Délai après le démarrage du moteur exprimé en cinquantièmes de seconde.
BE46	2	Délai d'arrêt du moteur.
BE48	1	Délai SEEK TRACK.
BE4B	1	Taille du tampon pour l'état du FDC.
BE4C	7	Tampon du FDC.
BE53	1	Lecteur.
BE54	1	Piste.
BE55	1	Secteur.
BE5F	1	Sémaphore moteur ON/OFF.
BE60	2	Vecteur sur tampon d'enregistrement.
BE66	6	Nombre de tentatives de lecture.
BE67 à BE73		Paramètres pour la gestion des interruptions.
BE74	1	Numéro de piste à atteindre.

CP/M 2.2

Le **CP/M** (Control Program of Microcomputer) est un système d'exploitation développé par la société américaine Digital Research en 1976. A l'origine, ce SED (DOS) était prévu pour des systèmes à base de microprocesseur 8080 de chez INTEL. Le Z80 constituant l'évolution naturelle du 8080, bien des constructeurs de machines à base de Z80 ont utilisé le CP/M comme principal système d'exploitation.

Le CP/M est vite devenu le SED le plus répandu dans le monde. A l'heure actuelle, sa suprématie est détrônée par le MS/DOS de Microsoft, cher aux utilisateurs d'IBM PC, et autres compatibles.

Le CP/M, pauvre en fonctions de base, s'est considérablement enrichi par une énorme bibliothèque de logiciels de toutes sortes.

Bien que prévu à l'origine pour fonctionner avec des systèmes équipés de 20 K de mémoire vive, la plupart des logiciels commerciaux demandent une TPA (c'est le nom donné à l'espace mémoire disponible) de plusieurs dizaines de K. Hélas, l'Amstrad 464 ou 664 ne dispose pas de 64 K. La mémoire d'écran occupe 16 K et le SED occupe une douzaine de K. La TPA est donc d'environ 36 K. Cet espace, suffisant pour bien des applications courantes, est malheureusement trop petit pour quelques excellents logiciels du marché dont l'implantation complète est à jamais bannie des CPC 464 et 664. Les heureux possesseurs d'un CPC 6128 ne rencontreront pas ce problème grâce à leur mémoire de 128 K qui leur laisse une TPA de 61 K.

La version actuelle du CP/M est la version 2.2. La version 3.0 ou CP/M+ est uniquement disponible sur des machines possédant plus de 64 K de mémoire vive. C'est le cas du CPC 6128. Les particularités du CPM 3.0 seront analysées au chapitre suivant.

ORGANISATION MEMOIRE DU CP/M 2.2

Le CP/M est composé de trois modules logiciels principaux.

- le **CCP** (Console Command Processor) qui est l'interpréteur de commande ;
- le **BDOS** (Basic Disc Operating System) qui est la partie résidente traitant de la gestion générale des disques en dehors de l'interface direct avec le matériel) ;
- le **BIOS** (Basic Input Output System) qui gère la relation entre le matériel et le logiciel.

Remarque *:* l'ensemble BDOS+BIOS porte le nom de FDOS.

Les deux premiers modules sont identiques d'un CP/M à l'autre, quel que soit le type d'ordinateur utilisé (APPLE, Commodore, Amstrad,...). Le troisième dépend de la configuration matérielle (circuits utilisés, ports utilisés,...). Il est donc propre au système AMSTRAD.

Le **CCP** et le **BDOS** résident sur les deux premières pistes de la disquette système. Quant au BIOS, il est inclus dans la ROM d'extension disque. Il partage donc le même espace que la mémoire écran.

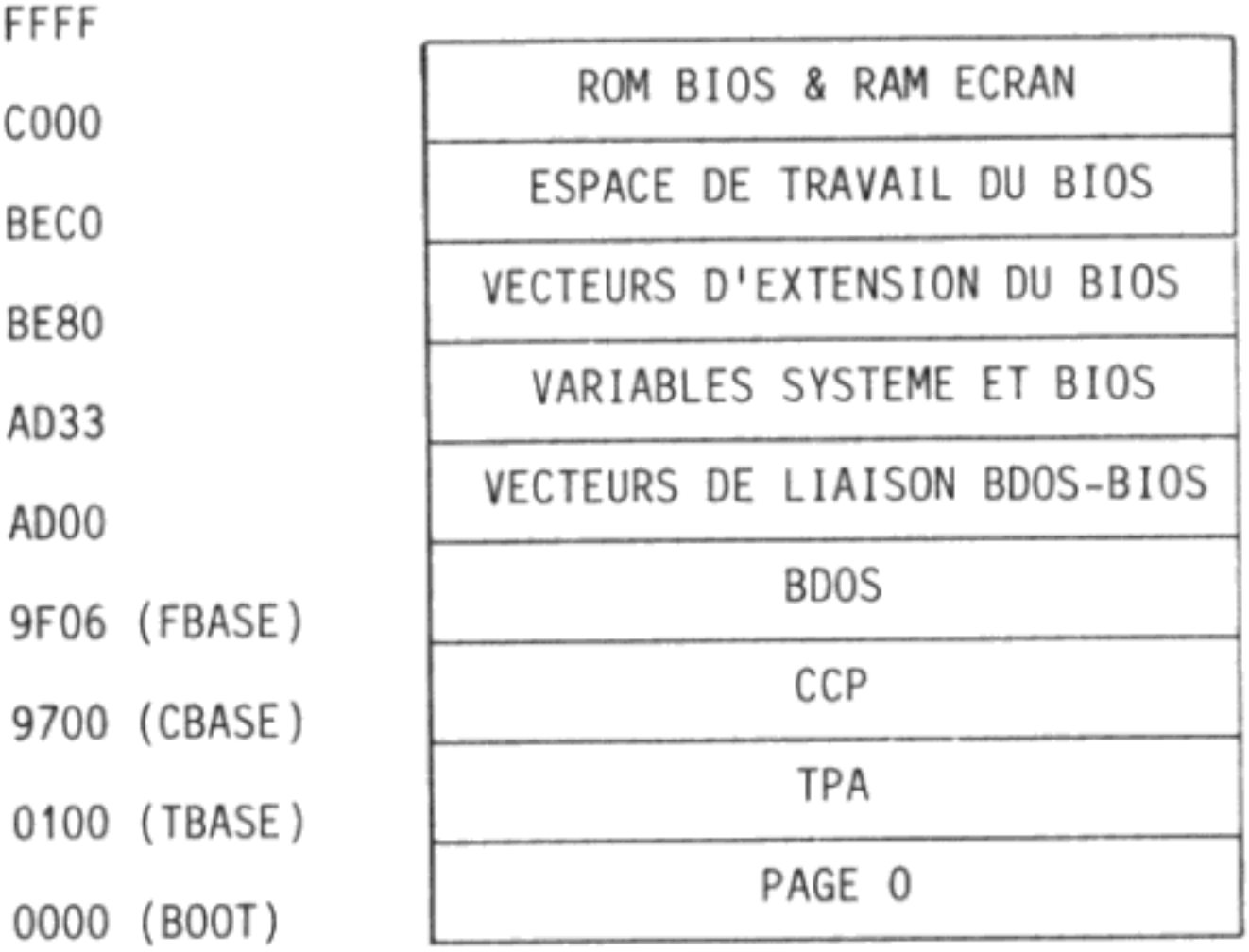
Le **BDOS** communique avec le BIOS au travers d'une série de vecteurs situés juste après le BDOS.

Ces trois modules se chargent en haut de mémoire. En-dessous de ces trois modules se trouve l'espace de travail disponible pour l'utilisateur (TPA pour Transient Program Area). La TPA commence à l'adresse 0100H et les programmes CP/M commenceront donc généralement leur chargement à cette adresse.

Les 256 premières adresses (0 à FFH) contiennent les vecteurs d'appel du BIOS et les RESTART du processeur Z80.

ORGANISATION MEMOIRE DU CP/M 2.2

Carte mémoire



ORGANISATION GENERALE ET FONCTIONNEMENT

Démarrage à froid ou COLD BOOT

Le démarrage à froid est produit lors de la première initialisation du CP/M par la commande AMSDOS ¦CPM. Le démarrage immédiat du CP/M est possible par une modification matérielle.

A l'initialisation, le logiciel appelle la routine d'initialisation située en BD16H. Cette routine lance le programme contenu dans la ROM d'arrière-plan et contenant les routines disque.

La ROM initialise un BIOS minimum et charge à l'adresse 0100H le secteur 41H de la piste 0 appelé SECTEUR BOOT. Ce secteur est exécuté. Il se charge de l'initialisation générale. A l'issue du démarrage à froid, un démarrage à chaud est exécuté.

Pour permettre l'installation éventuelle d'autres systèmes d'exploitation que le CP/M, l'initialisation passe par la lecture du secteur BOOT du disque (piste 0, secteur 41H).

Procédure complète de chargement

- Toutes les routines du BIOS sont accessibles.
- La sauvegarde du second jeu de registres du Z80 et du registre IY est enclenchée.
- Les interruptions sont détournées vers le BIOS.
- Les messages disque sont permis.
- Le tampon de commande est vidé.
- L'I/O byte est positionné à sa valeur par défaut (81H).
- Le disque par défaut est positionné pour l'unité A.
- Le secteur BOOT est chargé à l'adresse 100 et exécuté.
- Le SP est positionné en AD33H.

Démarrage à chaud ou WARM BOOT

C'est l'opération de rechargement du BDOS et du CCP après le passage d'un programme dans la TPA. Le démarrage à chaud est invoqué après le démarrage à froid par l'appel de la routine de RESET du BDOS ou par l'appui simultané sur les touches CTRL et C.

Le démarrage à chaud charge le BDOS et le CCP puis exécute le CCP et attend une commande. La routine initialise les six premiers octets de la RAM (0000 à 0005) et recopie les vecteurs BIOS juste à la fin du BDOS.

Extension résidente en mémoire

Ces extensions (DRIVER spéciaux ou routines assembleurs particulières) sont possibles par l'intermédiaire du programme MOVCPM.COM qui permet de déplacer le BDOS et le CCP pour laisser un espace libre. Il est évident que cette opération se fait au détriment de la TPA.

Octet d'allocation d'entrée/sortie ou I/O BYTE

Le CP/M supporte quatre périphériques logiques d'entrée/sortie appelés **CONSOLE** (CON), **READER** (RDR), **PUNCHER** (PUN) et **LIST** (LST). Ces quatre périphériques peuvent être associés à un périphérique physique. L'affectation courante des périphériques logiques, vis-à-vis des périphériques physiques, est contenue dans un octet appelé I/O BYTE.

La modification de l'I/O BYTE se fait au moyen de la commande STAT.

L'I/O BYTE est situé à l'octet 0003H de la RAM. Sa valeur initiale est 81H (10000001).

Les affectations possibles sont les suivantes :

	00	01	10	11
bit 0 et 1 : CON	TTY	CRT	BAT	UC1
bit 2 et 3 : RDR	TTY	PTR	UR1	UR2
bit 4 et 5 : PUN	TTY	PTP	UP1	UP2
bit 6 et 7 : LST	TTY	CRT	LPT	UL1

L'affectation initiale est donc 10 00 00 01 en allant du bit 7 au bit 0.

CON = CRT : RDR = TTY : PUN = TTY : LST = LPT

TTY : dispositif spécial d'entrée/sortie numéro 0.
CRT : écran et clavier standard de l'Amstrad.
BAT : entrée en provenance de RDR et sortie sur LST.
UC1 : dispositif spécial d'entrée/sortie numéro 1.
UR1 : dispositif spécial d'entrée/sortie numéro 1.
UP1 : dispositif spécial d'entrée/sortie numéro 1.
UL1 : dispositif spécial d'entrée/sortie numéro 1.
UR2 : clavier seul.
UP2 : écran seul.
LPT : port imprimante centronics parallèle.
PTP : ne sort rien sur le PUNCHER.
PTR : génère des EOF (1AH) en permanence

Caractères de contrôle compris par CP/M

CTRL	*Code*	*Effet*
C	03	Arrêt du programme en cours et WARM BOOT.
E	05	Retour chariot à l'écran, mais pas dans le programme.
H	08	Effacement de caractère avec retour arrière.
I	09	Tabulation de huit caractères.
J	0A	Génération d'un LINE FEED (saut de ligne).
M	0D	Génération d'un Retour Chariot (ENTER).
P	10	Commutation de l'imprimante en ECHO.
R	12	Efface et retape la commande.
S	13	Commutation de l'inhibition d'affichage écran.
U	15	Ignore la commande et descend le curseur d'une ligne.
X	18	Efface la commande.
Z	1A	Effet bizarre de réduction de l'écran.

Messages d'erreur

L'AMSDOS utilise une partie des messages du CP/M. Il n'est donc pas étonnant de retrouver, ici, ces messages.

Drive A ou B : disc missing.
Soit il n'y a pas de disquette dans le lecteur, soit la disquette est mal placée.

Drive A ou B : disc is write protected.
La disquette est protégée en écriture. Protection matérielle réalisée par le petit ergot sur la disquette (support) elle-même.

Drive A ou B : read fail.
La lecture est impossible. Nous vous conseillons de reformater votre disquette après sauvegarde des fichiers importants. Ce message peut annoncer une panne mécanique de votre lecteur de disque ou une destruction du revêtement magnétique de votre disquette (support).

Drive A ou B : write fail.
Ecriture impossible. Si votre disquette est correctement formatée, reportez-vous aux remarques faites ci-dessus pour read fail.

Failed to load CP/M.
Erreur rencontrée durant le passage au CP/M (¦CPM). Votre disquette est endommagée ou ne contient pas le CP/M (disquettes données seules).

Failed to load boot sector.
Le secteur BOOT est illisible pendant un démarrage à froid.

Remarque *:* ces messages d'erreur sont suivis d'une question "Retry, Ignore or Cancel" littéralement "Nouvel essai, Ignorer l'erreur, Avorter la commande".

Le système attend une réponse :
R réexécutera la commande ou la tentative de lecture ou d'écriture.
I permet à l'ordinateur de continuer son action (résultat non garanti, soyez prudent).
C annule l'opération en cours.

Le secteur de configuration

Le secteur 42H de la piste 0 constitue le secteur de configuration. Il contient de nombreux paramètres système qui sont utilisés par le BIOS. Ces paramètres peuvent être modifiés au moyen de l'utilitaire SETUP.COM *(voir cette commande)*.

Format du secteur de configuration

OCTETS

0 & 1 35H et 12H signature du disque permettant de reconnaître que la configuration a été chargée.

2 & 3 Délai entre le démarrage du moteur et l'accès aux données exprimé en cinquantièmes de seconde (valeur par défaut 1 seconde).

4 & 5 Délai entre le dernier accès et l'arrêt du moteur exprimé en cinquantièmes de seconde (valeur par défaut 2,5 secondes).

6 Temps de déplacement d'une piste à l'autre (STEP RATE).

7 Valeur initiale de l'I/O BYTE.

8 Autorisation/inhibition d'affichage des messages BIOS.

9 Autorisation/inhibition de sauvegarde du jeu secondaire de registres et du registre IY.

A à 15 Utilisés pour initialiser l'interface série.
- 0A SIO CANAL A registre 4
- 0B SIO CANAL A registre 5
- 0C SIO CANAL A registre 3
- 0D SIO CANAL B registre 4

- OE SIO CANAL B registre 5
- OF SIO CANAL B registre 3
- 10 & 11 8253 TIMER 0
- 12 & 13 8253 TIMER 1
- 14 & 15 8253 TIMER 2

16 Inhibition/autorisation de destruction du tampon de commande initial (00 ou FF).

17 à 63 Réservés (00).

Les octets suivants sont occupés par des champs de longueurs variables :

- Chaîne apparaissant à l'écran lors de l'initialisation et terminée par un $.
- Chaîne de paramétrage de l'imprimante (le premier octet contient la longueur de la chaîne).
- Table de codage du clavier pour les touches sans CONTROL ni SHIFT : cette table est composée d'entrées de deux octets, le premier indiquant le code de la touche pressée et le second le numéro de la touche pressée (pour la table des numéros de touches, reportez-vous au *volume 1 du présent ouvrage)*. Le premier octet de la table indique le nombre d'entrées.
- Table de codage du clavier pour les touches appuyées conjointement à la touche SHIFT : structure identique à la précédente.
- Table de codage du clavier pour les touches appuyées conjointement à la touche CONTROL : structure identique à la précédente.
- Enfin, pour terminer, on trouve une table des codes expansés (fonctions programmables). Le premier octet indique le nombre d'entrées. Chaque entrée est composée d'un octet indiquant le code à expanser, d'un octet indiquant la longueur de la chaîne d'expansion et enfin de la chaîne d'expansion elle-même.

Tables de paramétrage du disque

Pour faciliter l'accès aux disques en provenance d'autres systèmes et donc d'un format différent, tous les paramètres qui concernent le disque sont mémorisés en mémoire vive et donc facilement altérables par l'utilisateur averti.

En général, chaque disque possède deux tables. La première est appelée **DPH** (Disk Parameter Header). Elle contient un pointeur vers la seconde appelée **DPB** (Disk Parameter Bloc).

LA DPH

La DPH est composée de 16 octets (00 à 0FH) divisés en 8 mots de 16 bits.

Octets 0 & 1 : **XLT** : adresse du vecteur de translation des numéros de secteurs logiques et physiques. Si ce vecteur n'existe pas, le mot vaut 0000.

Octets 2 & 3 : 0000 Zone de travail du BDOS.

Octets 4 & 5 : 0000 Zone de travail du BDOS.

Octets 6 & 7 : 0000 Zone de travail du BDOS.

Octets 8 & 9 : **DIRBUF** : adresse d'un tampon de 128 octets pour les opérations du BDOS sur le répertoire.

Octets A & B : **DPB** : adresse de la table DPB.

Octets C & D : **CSV** : adresse d'un tampon pour le test logiciel de changement de disque.

Octets E & F : **ALV** : adresse d'un tampon utilisé par le BDOS pour mémoriser des informations sur l'allocation du disque.

LA DPB

La zone de paramétrage se compose de 25 octets (00-18H) définis comme suit :

Octets 0 & 1 : **SPT** : nombre total de secteurs par piste, codé sur 16 bits.

Octet 2 : **BSH** : facteur de déplacement du bloc d'allocation de données déterminé par la taille du bloc d'allocation de données.

Octet 3 : **BLM** : masque du bloc d'allocation de données ((2 exposant BSH) - 1).

Octet 4 : **EXM** : extension du masque déterminée par la taille du bloc d'allocation et le nombre de blocs sur le disque.

Octets 5 & 6 : **DSM** : capacité totale de stockage du disque exprimée en nombre de blocs - 1 (valeur codée sur 16 bits).

Octets 7 & 8 : **DRM** : nombre d'entrées possibles dans le répertoire - 1 (valeur codée sur 16 bits).

Octets 9 & A : **ALO** & **AL1** : valeur 16 bits qui indique le nombre de blocs réservés pour le répertoire.

Octets B & C : **CKS** : taille du vecteur de contrôle ((DRM+1)/4).

Octets D & E : **OFF** : nombre de pistes réservées pour le système.

Octet F : **FSC** : numéro physique du premier secteur.

Octet 10 : **PST** : nombre de secteurs par piste.

Octet 11 : **GLS** : longueur du GAP (en-tête de synchro) pour les écritures et les lectures.

Octet 12 : **GLT** : longueur du GAP pour le formatage.

Octet 13 : **FLB** : octet de remplissage des secteurs lors du formatage (FILLER).

Octet 14 : **BPS** : cet octet exprime la taille du secteur selon la formule : logarithme en base 2 de la taille -7. Ce nombre est utilisé pour programmer directement le FDC PD765A.

Octet 15 : **RPS** : taille d'un secteur divisée par 128.

Octet 16 : **BCT** : piste courante (réservé au BIOS).

Octet 17 : **FTO** : indicateur d'alignement. 0 = non aligné, FF = aligné (réservé au BIOS).

Octet 18 : **FLG** : indicateur de sélection automatique du format. 00 = sélection automatique, FF = pas de sélection automatique.

Les valeurs de **BSH** et **BLM** déterminent implicitement la taille d'un bloc d'allocation (**BLS**).

BSH	BLM	BLS
3	7	1024
4	15	2048
5	31	4096
6	63	8192
7	127	16384

EXM est déterminé par la valeur de **BLS** et la valeur de **DSM** (SUPERIEURE à 255 ou non).

BLS	EXM si DSM est < 256	EXM si DSM > 255
1024	0	-
2048	1	0
4096	3	1
8192	7	3
16384	15	7

La valeur de **DSM** exprime le nombre maximum de blocs de données de taille BLS supporté par le disque.

Les valeurs de **ALO** et **AL1** sont déterminées selon la formule suivante : nal = DRM * 32 / BLS (une entrée du répertoire occupe 32 octets).

Exemple : dans le cas de l'AMSTRAD, BLS = 1024 et DRM = 64. Il faut donc 32 x 64 / 1024 blocs pour le répertoire, soit 2 blocs.

ALO et **AL1** sont codés de la façon suivante :

ALO								AL1							
B7	B6	B5	B4	B3	B2	B1	B0	B7	B6	B5	B4	B3	B2	B1	B0

Si 1 bloc doit être réservé, alors seul B7 de ALO est à 1.
Si 2 blocs doivent être réservés, alors B7 et B6 de ALO sont à 1.
...
Si 16 blocs doivent être réservés, alors les 8 bits de ALO et les 8 bits de AL1 sont à 1.

Dans le cas de l'AMSTRAD, ALO vaut COH et AL1 vaut 0.

Valeurs standards de la DPB
en fonction du format

Remarque *:* toutes les valeurs sont données en hexadécimal.

Octets	*Système*	*Données seules*	*IBM*
SPT	24	24	20
BSH	3	3	3
BLM	7	7	7
EXM	0	0	0
DSM	AA	B3	9B
DRM	3F	3F	3F
ALO	C0	C0	C0
AL1	0	0	0
CKS	10	10	10
OFF	2	0	1
FSN	41	C1	01
RPT	9	9	8
GLR	2A	2A	2A
GLF	52	52	50
FIL	E9	E9	E9
SS1	2	2	2
SS2	4	4	4
CTK	0	0	0
ALG	0	0	0
AUT	0	0	0

Accès à la DPH et à la DPB

Pour accéder à la **DPH**, il faut appeler la routine BIOS SELDSK. Cette routine retourne l'adresse de la DPH *(voir routines internes)*. Le mot qui se trouve à un OFFSET 0AH (10) de l'adresse fournie est l'adresse de la **DPB.**

Le mot situé à l'adresse BE40H contient lui aussi l'adresse de la DPH.

Une troisième possibilité consiste en l'appel de la fonction 31 du BDOS.

Remarque *:* lors de l'appel de SELDSK, si le bit 0 du registre E vaut 0 et que l'octet 18H (AUT) de la DPB vaut 00, alors le BIOS essaie de déterminer le format de disque et modifie la table DPB en accord avec le format trouvé.

Les formats de la disquette

Les CPC distinguent trois formats différents : le format standard, le format de données et le format compatible IBM. Vous pouvez choisir votre format lors de l'opération de formatage de disquette *(voir commande FORMAT)*.

Points communs aux trois formats :

```
Nombre de pistes                : 40 numérotées de 0 à 39.
Taille d'un secteur             : 512 octets.
Nombre d'entrées du répertoire  : 64.
Taille d'un enregistrement      : 128 octets.
```

Le nombre d'enregistrements par secteur est évidemment de 4.

Caractéristiques du format standard

C'est le format le plus courant.

```
Nombre de secteurs par piste    : 9.
Numérotation des secteurs       : de 41H à 49H.
Nombre de pistes réservées      : 2.
```

Les pistes réservées permettent l'installation du CP/M.

Affectation des pistes réservées :

```
Piste 0 secteur 41H             : secteur BOOT.
Piste 0 secteur 42H             : secteur CONFIGURATION.
Piste 0 secteurs 43H à 47H      : inutilisés.
Piste 0 secteurs 48H & 49H      : CCP et BDOS.
Piste 1 secteurs 41H à 49H      : suite du CCP et du BDOS.
```

Caractéristiques du format de données

Nombre de secteurs par piste : 9.
Numérotation des secteurs : C1H à C9H.
Pistes réservées : 0.

Caractéristiques du format IBM compatible

Nombre de secteurs par piste : 8.
Numérotation des secteurs : 0 à 7.
Pistes réservées : 0.

Structure du répertoire (DIRECTORY)

Remarque *:* le bloc d'affectation minimum est de 1 K, soit 2 secteurs ou 8 enregistrements.

Le répertoire occupe les quatre premiers secteurs du début de la disquette (en format standard : les secteurs 41H à 44H de la piste 2).

Chaque entrée du répertoire occupe 32 octets numérotés de 00 à 1FH.

Octet 0 : numéro d'USER du fichier ou E5H si le fichier est effacé.

Octets 1 à 8 : nom du fichier éventuellement complété par des blancs (20H).

Octets 9 à B : extension du nom éventuellement complétée par des blancs.
Remarque *:* le bit 7 de l'octet 9 indique le statut READ ONLY *(voir STAT)*.
Le bit 7 de l'octet A indique le statut système invisible (n'apparaît pas dans le répertoire).

Octet C : numéro de l'extension d'entrée dans le cas d'un grand fichier.

Octet F : nombre d'enregistrements de 128 octets composant le fichier.
Remarque *:* si cette valeur vaut 80H, le fichier est composé de plusieurs entrées.

Octets 10 à 1F : numéros des blocs occupés par le fichier.

ORGANISATION GENERALE ET FONCTIONNEMENT

Octets réservés en page 0

La mémoire centrale, comprise entre les adresses 0 et FFH, est appelée page 0. Cette portion de mémoire contient de nombreux paramètres indispensables au bon fonctionnement du CP/M.

Octets	*Fonction*
0000H à 0002H	: contient une instruction JUMP vers le point d'entrée du démarrage à chaud (AD03H).
0003H	: I/O Byte : par défaut 81H.
0004H	: numéro de disque courant (0 ou 1).
0005H à 0007H	: JUMP vers le point d'entrée principal du BDOS (8F00H).
0008H à 003FH	: zone pour le traitement des interruptions et des RESTART.
0040H à 004FH	: zone SCRATCH pour CBIOS.
0050H à 005BH	: inutilisés.
005CH à 007CH	: FCB par défaut.
0080H à 00FFH	: tampon de 128 octets pour un enregistrement.

Structure du FCB (File Control Bloc)

Les opérations sur les fichiers faites par le BDOS utilisent le registre **DE** comme pointeur sur une zone appelée **FCB.**

Le FCB se compose de 33 (00 à 20H) ou 36 (00 à 23H) octets suivant le mode d'accès utilisé (séquentiel ou direct).

Octets	*Fonction*
00	: numéro du disque (0 disque courant, 1 disque A, 2 disque B).
1 à 8	: nom du fichier en ASCII majuscule.
9 à B	: extension du nom.
C	: numéro d'extension (mis à zéro au départ).
D & E	: réservé à l'usage interne du système (OPEN).
F	: compteur d'extension (0 à 7F).
10 à 1F	: réservé au système.
20	: numéro de l'enregistrement courant (mis à zéro au départ).
21 à 23	: numéro d'enregistrement pour accès direct.

Nom de fichier et carte de sélection

Les noms de fichiers sous CP/M sont représentés par un nom principal de 1 à 8 caractères alphanumériques suivi d'une extension optionnelle de 1 à 3 caractères. La séparation entre l'extension et le nom principal est réalisée par le caractère '.'. Ce nom peut être précédé d'un indicateur d'unité de lecture (A ou B) suivi de ':'.

Exemple

A:MARTIN.BAS

Les caractères suivants sont interdits dans un nom de fichier ou une extension :

< > . , ; : = ? * [ ] _ % ¦ ( ) / \

Carte de sélection

Les caractères **?** et ***** permettent d'introduire une ambiguïté dans un nom de fichier :

? représente n'importe quel caractère (un seul).
* représente n'importe quelle suite de caractères.

L'utilisation de **?** et ***** forme une carte de sélection (WILD CART).

Certaines commandes CP/M acceptent les noms de fichier ambigus.

Exemples

- XYZ.COM représente le fichier unique XYZ.COM.
- X?Z.COM représente tous les fichiers d'extension COM dont le nom de fichier est composé de trois lettres et dont la première est W et la troisième Z (par exemple : XAZ.COM ou XYZ.COM ou X1Z.COM).
- *.COM représente tous les fichiers d'extension COM.
- L*.COM représente tous les fichiers dont l'extension est COM et dont le nom commence par L (exemple : L.COM, LIT.COM ou encore LOLITA.COM).
- DUMP.* représente tous les fichiers dont le nom est DUMP, quelles que soient leurs extensions.
- *.* représente tous les fichiers.

ORGANISATION GENERALE ET FONCTIONNEMENT

Principaux noms d'extensions

$$$ Extension de fichier en cours de création.
ASM Source assembleur.
BAS Programme Basic.
BIN Programme Binaire.
BAK Copie BACKUP créée automatiquement par une nouvelle sauvegarde ou par l'utilisation de ED.
C Extension courante dans le langage C.
COB Programme source COBOL.
COM Programme "COMMANDE" directement exécutable par CP/M.
DBF Data Base File : fichier de base de données dBASE.
FOR Programme FORTRAN.
HEX Format hexadécimal.
INT Code intermédiaire.
OVR Overlay (morceau de programme).
PLI Source en PL/1.
PRN Format prêt à l'impression.
REL Source "RELOCATABLE" (indépendante de l'adresse).
SUB Programme procéduriel créé pour SUBMIT.
SYM Fichier symbole de SID ou ZSID.
TEX Programme format Texte.
TXT Programme format Texte.

TABLE DES VECTEURS STANDARDS DU BIOS

Le **CP/M 2.2** de l'AMSTRAD contient 17 vecteurs standards d'appel de routines du BIOS. La table de JUMPS se trouve à l'adresse AD00H.

Numéro	*Adresse*	*Saut à*	*Nom*	*Fonction*
1	AD00	C1B2	**BOOT**	Initialisation principale du système et sélection du disque A si le registre C contient 0. En fin d'initialisation, transfert au CCP.
2	AD03	C2BE	**WBOOT**	Démarrage à chaud.
3	AD06	C2E1	**CONST**	Retourne dans A l'état de la CONSOLE.
4	AD09	C2C3	**CONIN**	Lecture de la CONSOLE. Résultat dans A.
5	AD0C	C2C8	**CONOUT**	Sortie du caractère contenu dans A vers la console.
6	AD0F	C2D2	**LIST**	Sortie du caractère contenu dans A vers le périphérique LIST.
7	AD12	C2D7	**PUNCH**	Ecriture du caractère contenu dans A vers le périphérique PUNCH.
8	AD15	C2DC	**READER**	Lecture du périphérique READER. Résultat dans A.
9	AD18	C2E9	**HOME**	Retour du disque courant à la piste 0.
10	AD1B	C2F2	**SELDSK**	Sélection du disque dont le numéro est contenu dans le registre C.
11	AD1E	C524	**SETTRK**	Sélection de la piste dont le numéro est contenu dans le registre C.
12	AD21	C529	**SETSEC**	Sélection du secteur dont le numéro est contenu dans le registre C.
13	AD24	C51A	**SETDMA**	Sélection de l'adresse du tampon d'écriture/lecture de l'enregistrement courant. L'adresse sélectionnée doit être contenue dans BC.

TABLE DES VECTEURS STANDARDS DU BIOS

Numéro	*Adresse*	*Saut à*	*Nom*	*Fonction*
14	AD27	C2F7	**READ**	Lecture de l'enregistrement désigné par SETTRK et SETSEC dans le tampon désigné par SETDMA. A contient 0 si l'opération a réussi et 1 sinon.
15	AD2A	C2FC	**WRITE**	Ecriture de l'enregistrement désigné (*voir* lecture).
16	AD2D	C2CD	**LISTST**	Fournit l'état du périphérique LIST dans le registre A (0 = NOT READY).
17	AD30	C55A	**SECTRAN**	Convertit un numéro de secteur logique en numéro de secteur physique (BC contient le numéro du secteur et DE pointe sur une table de conversion).

TABLE DES VECTEURS PROPRES AU BIOS DE L'AMSTRAD

Le CP/M 2.2 de l'AMSTRAD possède une série de vecteurs qui lui sont propres et ne figurent pas dans les CP/M des autres machines. En voici la description :

Adresse	*Saut à*	*Fonction*
BE80	CA72	Routine équivalente à la routine 1 de l'AMSDOS *(voir chapitre II, vecteurs propres au disque)*.
BE83	C60D	Routine équivalente à la routine 2.
BE86	C581	Routine équivalente à la routine 3.
BE89	C666	Routine équivalente à la routine 4.
BE8C	C64E	Routine équivalente à la routine 5.
BE8F	C652	Routine équivalente à la routine 6.
BE92	C763	Routine équivalente à la routine 7.
BE95	C630	Routine équivalente à la routine 8.
BE98	C603	Routine équivalente à la routine 9.
BE9B	C168	Appel d'une routine FIRMWARE (logiciel interne). Les conditions d'entrée et de sortie dépendent des routines à appeler.
BE9E	C0DB	Sélection/inhibition de la sauvegarde des registres du second jeu (AF', BC', DE' et HL') ainsi que de IY. **CE** : A = 0 pour la sélection. A = FFH (255) pour l'inhibition. **CS** : A contient l'ancien état, F et HL sont modifiés.

Vecteurs du dispositif d'entrée/sortie série (RS232)

BEA1	C389	Réinitialisation de l'interface série (RS232). **CE** : HL pointe sur la table de paramétrage. **CS** : AF, BC, DE et HL sont modifiés.
BEA4	C301	Initialise le tampon d'entrée (tampon de 128 octets pour les commandes DOS). **CE** : A = 0 → Le tampon est nettoyé lors de l'appui sur une touche.

A = 1 → Le tampon est conservé lors de l'appui sur une touche.
HL contient l'adresse du tampon.

CS : AF, BC, DE et HL sont modifiés.

BEA7 C3DB — Teste si le dispositif d'entrée/sortie 0 (canal A) dispose d'un caractère à traiter en entrée.

Pas de **CE**.

CS : si A vaut FF, un caractère est disponible en entrée.
Si A vaut 0, il n'y a pas de caractère disponible.
Dans les deux cas, AF, BC, DE et HL sont modifiés.

BEAA C3F7 — Saisit un caractère en provenance du dispositif d'entrée/sortie 0 (canal A). S'il n'y a pas de caractère disponible, attend l'arrivée d'un caractère.

Pas de **CE**.

CS : A contient le caractère, F, BC, DE et HL sont modifiés.

BEAD C435 — Teste si le dispositif d'entrée/sortie 0 (canal A) est prêt à émettre un caractère.

Pas de **CE**.

CS : si A = FFH, le dispositif est prêt. Si A vaut 0, le dispositif est occupé (BUSY).
Dans les deux cas, F, BC, DE et HL sont modifiés.

BEB0 C445 — Sortie d'un caractère sur le dispositif d'entrée/sortie 0 (canal A). Si le dispositif est occupé, cette routine attend qu'il devienne disponible.

CE : C contient le caractère à sortir.

CS : AF, BC, DE et HL sont modifiés.

BEB3 CEE3 — Comme BEA7 pour le dispositif 1 (canal B).

BEB6 C3FF — Comme BEAA pour le dispositif 1 (canal B).

BEB9 C43A — Comme BEAD pour le dispositif 1 (canal B).

BEBC C44B — Comme BEB0 pour le dispositif 1 (canal B).

TABLE DES VECTEURS STANDARDS DU BDOS

Les vecteurs standards du BDOS sont appelés en disposant le numéro de la fonction dans le registre C et en effectuant un **CALL** à l'adresse 0005H (saut au BDOS).

Le registre **DE** fournit les informations en entrée (positionnement sur le FCB, code de caractère ...). En sortie, les valeurs d'un octet sont transmises dans A et les valeurs des deux octets sont transmises dans HL. Pour des raisons de compatibilité, le registre **A** et le registre **L** sont identiques en sortie. Il en est de même des registres **B** et **H.**

Fonction 0 : Initialisation du système.

Action : réinitialisation totale du système. Cette fonction est identique à un saut au BOOT (0000H).
CE : C = 00.
CS : pas de CS.

Fonction 1 : Lecture CONSOLE.

Action : lecture dans l'accumulateur du caractère entré sur le périphérique physique associé à la console. Les caractères CR, LF, BS et TAB sont traités ainsi que le test du CTRL S et CTRL P.
CE : C = 01.
CS : A contient le caractère ASCII.

Fonction 2 : Ecriture CONSOLE.

Action : écriture dans le périphérique physique associé à la console du caractère contenu dans le registre E. Les caractères TAB, CR, LF, BS, CTRL S et CTRL P sont traités.
CE : C = 02, E contient le code ASCII du caractère à sortir.
CS : pas de CS.

Fonction 3 : Lecture READER.

Action : lecture du caractère en provenance du périphérique physique associé au READER.
CE : C = 03.
CS : A contient le code ASCII du caractère lu.

Fonction 4 : Ecriture PUNCHER.

Action : écriture du caractère contenu dans le registre E dans le périphérique physique associé au PUNCHER.
CE : C = 04, E contient le caractère à écrire.
CS : pas de CS.

Fonction 5 : Ecriture LIST.

Action : écriture du caractère contenu dans le registre E dans le périphérique physique associé au LIST.
CE : C = 05, E contient le caractère à écrire.
CS : pas de CS.

Fonction 6 : Lecture ou écriture directe sur la CONSOLE.

Action : cette fonction ne traite pas les caractères spéciaux (CTRL P, CTRL S, ...) et effectue un traitement direct avec la CONSOLE.
CE : C = 06, E contient le caractère à écrire ou FFH si le caractère doit être lu.
CS : A contient le caractère lu ou un état (STATUS) 00 s'il n'y a pas de caractère lu.

Fonction 7 : Lecture de l'I/O byte.

Action : lecture de l'I/O byte dans le registre A.
CE : C = 07.
CS : A contient la valeur de l'I/O byte.

Fonction 8 : Ecriture de l'I/O byte.

Action : écriture du contenu du registre E dans l'I/O byte.
CE : C = 08, E contient la valeur à écrire.
CS : pas de CS.

Fonction 9 : Impression d'une chaîne de caractères.

Action : imprime la chaîne de caractères pointée par le registre DE vers la CONSOLE. La chaîne doit être terminée par un signe $.
CE : C = 09, DE pointe sur la chaîne de caractères.
CS : pas de CS.

Fonction 10 : Lecture du tampon de la CONSOLE.

Action : lit la CONSOLE et pousse le contenu dans un tampon pointé par le registre DE. L'entrée se termine lorsque le tampon déborde ou lorsqu'un RETOUR CHARIOT (ODH) ou un LINE FEED (OAH) est introduit à la CONSOLE. A l'issue de la fonction, le tampon contient, comme premier octet, le nombre de caractères total que peut contenir le tampon et le second octet contient le nombre de caractères qui suivent dans le tampon.
CE : C = OAH, DE pointe sur le tampon, le premier caractère pointé par DE contient le nombre maximum de caractères accepté par le tampon (1 à 255).
CS : le tampon contient les caractères entrés.

Fonction 11 : Lecture de l'état de la CONSOLE.

Action : cette fonction teste si un caractère a été introduit à la CONSOLE. Si un caractère a été introduit, A contient FFH, sinon A contient 00.
CE : C = 0BH.
CS : A contient l'état de la CONSOLE.

Fonction 12 : Lecture du numéro de version du CP/M.

Action : cette fonction fournit une valeur sur deux octets. Le CP/M 2.2 fournit 0 dans le registre H et 22H dans le registre L.
CE : C = 0CH.
CS : HL contient le numéro de version.

Fonction 13 : Réinitialisation du système disque.

Action : cette fonction réinitialise le disque (disque A par défaut) et positionne le système en lecture et écriture permises. Cette fonction sert principalement à permettre un changement de disque sans faire de BOOT.
CE : C = 0DH.
CS : pas de CS.

Fonction 14 : Sélectionne une unité de disque.

Action : cette fonction sélectionne l'unité de disque dont le numéro est contenu dans le registre E (0=A, 1=B, ...). Le disque est immédiatement considéré comme disponible. Tout essai de changement de disque après le lancement de cette fonction produira un message d'erreur (READ/ONLY STATUS).
CE : C = 0EH, E contient le numéro de l'unité.
CS : pas de CS.

Fonction 15 : Ouverture de fichier.

Action : cette fonction ouvre le fichier dont le nom est contenu dans le FCB (*voir table*) pointé par DE. En retour, cette fonction fournit un état dans le registre A (0, 1, 2 et 3 indiquent une ouverture correcte, FFH indique que le fichier ne peut être trouvé dans le répertoire). ***Remarque*** *:* le nombre 0, 1, 2 ou 3 indique la position du nom du fichier dans l'enregistrement du répertoire (128 octets permettent 4 entrées de 32 octets).
CE : C = 0FH, DE pointe sur le FCB.
CS : A contient le statut d'ouverture.

Fonction 16 : Fermeture de fichier.

Action : cette fonction ferme le fichier dont le FCB est pointé par DE. Un état d'échec ou de succès identique à celui de l'ouverture est fourni en sortie. Une fermeture est indispensable si une ou plusieurs écritures ont eu lieu dans le fichier.
CE : C = 10H, DE pointe sur le FCB du fichier.
CS : A contient le statut de fermeture.

Fonction 17 : Première recherche dans le répertoire.

Action : cette fonction recherche un nom de fichier dans le répertoire en commençant au début et fournit dans A un statut indiquant la position ou l'absence du fichier spécifié.
CE : C = 11H, DE pointe sur le FCB du fichier.
CS : A contient le statut de la recherche.

Fonction 18 : Continuer la recherche.

Action : cette fonction est identique à la précédente, mais la recherche continue à partir de l'endroit où la dernière recherche s'est arrêtée.
CE : C = 12H.
CS : A contient le statut de la recherche.

Fonction 19 : Effacement de fichier.

Action : cette fonction efface du répertoire le fichier dont le FCB est pointé par le registre DE. Si le fichier n'est pas trouvé en sortie, A contient FFH.
CE : C = 13H, DE pointe sur le FCB du fichier.
CS : A contient le statut.

Fonction 20 : Lecture séquentielle d'un fichier.

Action : cette fonction réalise la lecture de l'enregistrement suivant (128 octets) du fichier préalablement ouvert (fonction 15 ou 22).
CE : C = 14H, DE pointe sur le FCB du fichier.
CS : A contient 0 si la lecture est bonne. Toute autre valeur indique qu'il n'y a pas de données dans l'enregistrement suivant.

Fonction 21 : Ecriture séquentielle dans un fichier.

Action : cette fonction lit l'enregistrement suivant (128 octets) d'un fichier préalablement ouvert.
CE : C = 15H, DE pointe sur le FCB du fichier.
CS : A contient 0 si l'écriture est correcte.

Fonction 22 : Création d'un fichier.

Action : cette fonction est similaire à l'ouverture d'un fichier, mais le FCB doit référencer un fichier qui n'existe pas dans le répertoire. L'entrée correspondante est alors créée. En sortie, A contient un compte-rendu identique à celui d'une ouverture.
CE : C = 16H, DE pointe sur le FCB du fichier.
CS : A contient le statut de la création.

Fonction 23 : Changement du nom d'un fichier.

Action : cette fonction utilise le FCB pointé par DE pour changer le nom du fichier contenu dans les 16 premiers octets en nom du fichier contenu dans les 16 octets suivants. En sortie, A contient un compte-rendu identique à celui d'une ouverture.
CE : C = 17H, DE pointe sur le FCB.
CS : A contient le statut du changement de nom.

Fonction 24 : Lecture du vecteur d'état des disques.

Action : cette fonction fournit un mot de 16 bits dans le registre HL qui indique les disques présents dans le système (le bit 0 de L correspond au disque A et le bit 7 de H au disque P). Si le disque est présent, le bit est à 1.
CE : C = 18H.
CS : HL contient le vecteur d'état.

Fonction 25 : Retourne le numéro du disque courant.

Action : cette fonction fournit, dans le registre A, le numéro du disque courant (sélectionné au moment de l'appel de la fonction).
CE : C = 19 H.
CS : A contient le numéro du disque (0 à 15).

Fonction 26 : Positionne l'adresse du DMA.

Action : cette fonction positionne l'adresse du tampon qui contient l'enregistrement à lire ou à écrire. En standard, cette valeur vaut 80H.
CE : C = 1AH.
CS : DE contient l'adresse du DMA.

Fonction 27 : Lecture de l'adresse d'ALLOC.

Action : cette fonction fournit dans HL l'adresse du vecteur d'allocation (ALLOC).
CE : C = 1BH.
CS : HL contient l'adresse du vecteur ALLOC.

Fonction 28 : Positionne le disque en WRITE PROTECT.

Action : cette fonction positionne le disque courant en protection d'écriture jusqu'au prochain démarrage à chaud ou à froid.
CE : C = 1C.
CS : pas de CS.

Fonction 29 : Lecture du vecteur READ/ONLY.

Action : cette fonction fournit dans HL un vecteur de 16 bits qui indique les disques qui possèdent le statut lecture seule (READ ONLY). Le bit 0 du registre L indique le statut du disque A.
CE : C = 1DH.
CS : HL contient le vecteur READ/ONLY.

Fonction 30 : Positionnement des attributs d'un fichier.

Action : cette fonction permet de positionner l'attribut READ/ONLY ou l'attribut SYSTEME (fichier invisible) dans un état ou dans l'autre (ACTIF ou INACTIF).
CE : C = 1EH, DE pointe sur le FCB du fichier spécifié avec les attributs positionnés dans l'état désiré (bit 7 des deux premiers caractères de l'extension).
CS : A contient FF si le fichier n'est pas trouvé. Sinon, A contient un nombre compris entre 0 et 3 qui indique la position de l'entrée dans l'enregistrement du répertoire.

Fonction 31 : Lecture de l'adresse de la DPB.

Action : cette fonction fournit, dans le registre HL, l'adresse du DPB (Disk Parameter Bloc) du disque courant.
CE : C = 1FH.
CS : HL pointe sur la DPB.

Fonction 32 : Positionnement ou lecture du numéro d'USER.

Action : cette fonction fournit le numéro d'USER courant si le registre E contient FFH ou modifie le numéro d'USER si E contient un autre nombre. Le numéro d'USER est pris égal à E modulo 16.
CE : C = 20H, E contient FFH ou le numéro d'USER.
CS : si E contient FFH, alors A contient le numéro d'USER courant.

Fonction 33 : Lecture directe.

Action : cette fonction est presque identique à la fonction de lecture séquentielle (fonction 20). L'enregistrement lu n'est pas le suivant du fichier, son numéro est contenu dans les trois octets qui suivent l'octet 31H

du FCB. L'enregistrement est écrit à l'adresse DMA. En retour, le registre A contient un code d'erreur. 0 indique une lecture correcte, 3 indique que l'enregistrement courant ne peut être fermé, 4 indique que l'enregistrement demandé n'existe pas encore et 6 indique que l'enregistrement demandé dépasse la taille du disque.

CE : C = 21H, DE pointe sur le FCB du fichier.
CS : A contient le compte-rendu d'erreur.

Fonction 34 : Ecriture directe.

Action : cette fonction est presque identique à la fonction d'écriture séquentielle (fonction 21). Comme dans la lecture directe (fonction 33), le numéro d'enregistrement est lu dans les trois derniers octets du FCB (32 à 35). L'enregistrement doit être à l'adresse DMA. En retour, A fournit un code d'erreur identique à celui de la fonction 33 avec, en plus, un code 5 qui indique qu'un nouvel enregistrement ne peut être créé pour des raisons de débordement de répertoire.

CE : C = 22H, DE pointe sur le FCB du fichier.
CS : A contient le compte-rendu d'erreur.

Fonction 35 : Calcul de la taille d'un fichier.

Action : cette fonction permet de déterminer la taille d'un fichier. Le résultat est fourni dans les octets 32 et 33 (ceux qui contiennent le numéro d'enregistrement direct à atteindre).

CE : C = 23H, DE pointe sur le FCB du fichier.
CS : les octets 32 et 33 du FCB contiennent la taille du fichier exprimée en nombre d'enregistrements.

Fonction 36 : Positionne le numéro d'enregistrement.

Action : cette fonction positionne, dans les octets 32 et 33 du FCB, le numéro direct d'enregistrement. Cette routine est intéressante dans le cas où le fichier a été préalablement adressé en mode séquentiel.

CE : C = 24H, DE pointe sur le FCB du fichier.
CS : les octets 32 et 33 du FCB contiennent le numéro d'enregistrement.

Fonction 37 : Réinitialisation du disque.

Action : initialise une configuration disque.
CE : C = 25H, DE contient le vecteur d'activité des disques (16 bits).
CS : A = 0.

Fonction 40 : Ecriture d'un enregistrement avec dés 0.

Action : cette fonction est identique à la fonction 34, mais l'enregistrement est rempli de 0.
CE : C = 28H, DE pointe sur le FCB du fichier.
CS : A contient un code d'erreur *(voir fonction 34)*.

Le BDOS contient six instructions résidentes et une commande de sélection de l'unité de disque par défaut. Ces instructions ne se retrouvent pas dans le répertoire, elles sont interceptées par le CCP et dirigées directement vers le BDOS.

Remarque *:* les abréviations suivantes sont utilisées :

U: : Lettre représentant l'unité sélectionnée (en général A ou B) suivie du caractère ':'.
nfc : Nom de Fichier en Clair (sans carte de sélection).
nfa : Nom de Fichier Ambigu (pouvant comporter une carte de sélection).
n : Nombre entier.
[] : Indique un paramètre optionnel.

Commande sélection de l'unité disque courante

Syntaxe : U:

B: sélectionne le disque B comme disque courant.
A: sélectionne le disque A comme disque courant.

Remarque *:* il faut taper le caractère ":".

Instructions

DIR Syntaxe : DIR [U:][nfa]

Cette instruction produit l'affichage de tous les fichiers qui correspondent au nom de fichier ambigu sur l'unité indiquée. Si l'indicateur d'unité et le nom de fichier ambigu ne sont pas utilisés, tous les fichiers contenus sur l'unité courante sont affichés.

Remarque *:* certains fichiers peuvent ne pas apparaître. C'est le cas des fichiers dit SYSTEM *(voir commande STAT)*.

ERA Syntaxe : ERA [U:]nfa

Cette instruction efface tous les fichiers qui correspondent au nom de fichier ambigu sur l'unité indiquée. Le nom de fichier est obligatoire.

Remarque *:* ERA *.* qui implique l'effacement de TOUS les fichiers demande une confirmation (ALL Y/N). La réponse Y produit l'effacement de tous les fichiers, toute autre réponse annule la commande.

INSTRUCTIONS RESIDENTES

REN Syntaxe : REN [U:]nfc1=[U:]nfc2

Cette instruction change le nom du fichier nfc2 en nfc1. Il est évident qu'avant la commande, nfc2 doit exister et nfc1 ne doit pas exister. De plus, le numéro d'unité optionnel doit être le même pour nfc1 et nfc2.

Si nfc1 existe, un message 'FILE EXISTS' est produit.

Si nfc2 n'existe pas, un message 'NO FILE' est produit.

SAVE Syntaxe : SAVE n [U:]nfc

Cette instruction sauve n blocs de 256 octets sur le disque sous le nom nfc. Ces n blocs sont pris à partir de l'adresse 100H de la mémoire.

TYPE Syntaxe : TYPE [U:]nfc

Cette instruction affiche, sur le périphérique attaché à la CONSOLE, le contenu du fichier nfc.

USER Syntaxe : USER n

Cette instruction positionne le numéro d'USER. n doit être compris entre 0 et 15. L'USER par défaut est l'USER 0.

UTILITAIRES PROPRES A DIGITAL RESEARCH

Ces utilitaires sont présents sur tous les CP/M 2.2, quels que soient la marque ou le type d'ordinateur utilisés. Ces utilitaires se présentent sous la forme de programmes directement exécutables (extension .COM). Ils se lancent en tapant simplement le nom de l'utilitaire sans l'extension.

Le **CCP** charge l'utilitaire dans la **TPA** en commençant à l'adresse 100H.

ASM.COM Syntaxe : ASM nom de fichier[.options]

Cette commande déclenche l'assembleur du système CP/M. Décrire complètement l'assembleur tombe hors du cadre de cet ouvrage. Il faut cependant signaler que l'assembleur ASM.COM est réservé aux mnémoniques du 8080 et que les utilisateurs d'AMSTRAD ont tout intérêt à utiliser un assembleur pour Z80 du genre DEVPAC.

Le nom de fichier à assembler doit comporter l'extension **.ASM.** Ce fichier doit être créé préalablement par un éditeur (traitement de texte ou ED.COM).

L'assembleur fournit deux nouveaux fichiers portant le même nom que l'original, mais l'un avec l'extension **.HEX** et l'autre avec l'extension **.PRN.** Le premier contient le code objet produit sous la forme de nombres hexadécimaux écrits en ASCII, le second contient un listing composé à la fois de la source et de l'objet.

Les options se composent de trois lettres avec la signification suivante :

- la première indique le disque qui contient la source ;
- la seconde indique le disque qui contiendra le fichier .HEX ;
- la troisième indique le disque qui contiendra le .PRN.

La lettre **Z**, en lieu et place de la seconde ou de la troisième lettre empêche la génération du fichier concerné.

La lettre **X** en troisième position produit le listing sur la CONSOLE en lieu et place du fichier .PRN.

DDT.COM Syntaxe : DDT [nom de fichier]

Le **DDT** (Dynamic Debugger Tool) est l'outil de mise au point des programmes en langage machine.

Le DDT peut être appelé seul ou avec un nom de fichier. Ce nom de fichier doit comporter une extension **.COM** ou **.HEX.**

Le DDT permet l'altération de la mémoire centralé, mais il ne permet pas la réécriture du fichier modifié. Il faut donc faire suivre l'utilisation du DDT d'une commande interne SAVE.

Les commandes du DDT

Remarque *:* ne laissez pas d'espace entre votre commande et son premier argument.

A : commande d'entrée directe de mnémoniques en assembleur du 8080. La commande A doit être suivie d'une adresse d'implantation en hexadécimal (*exemple :* A8C00).

D : commande de DUMP (copie du contenu de la mémoire en HEXA et en ASCII sur l'écran). La commande D peut être suivie d'une adresse en hexadécimal (*exemple :* DC000). Si aucune valeur n'est donnée, c'est la valeur courante qui est prise par défaut (au départ 0100H).

F : commande de remplissage mémoire. La commande F doit être suivie de trois paramètres séparés par des virgules. Le premier indique l'adresse mémoire de départ, le second, l'adresse de fin, et le troisième, la valeur hexadécimale à écrire (*exemple :* F1000,17FF,55 remplit la zone mémoire comprise entre 1000H et 17FFH avec la valeur 55H).

G : commande d'exécution. Cette commande peut être suivie d'un, deux ou trois paramètres séparés par des virgules. Utilisée seule, elle lance l'exécution du programme à l'adresse contenue dans le registre PC (Program Counter). Le premier paramètre indique l'adresse de lancement, le second et le troisième peuvent contenir des points d'arrêt (BREAKPOINT) (*exemple :* G,1000 lance le programme à l'adresse contenue dans le PC, arrête le programme et revient au DDT si le PC atteint l'adresse 1000H).

I : commande d'entrée de nom de fichier. La commande I doit être suivie du nom du fichier. Cette commande est utile pour la commande R.

L : commande de désassemblage. La commande L peut être suivie d'une adresse hexadécimale.

M : commande de déplacement de bloc en mémoire. Cette commande nécessite trois paramètres séparés par des virgules. Le premier indique l'adresse de départ du bloc, le second indique l'adresse de fin du bloc et le dernier indique la destination du bloc.

R : commande de lecture du fichier préalablement défini par la commande I. Cette commande admet un paramètre qui indique un OFFSET (décalage) de chargement.

S : commande de modification de la mémoire. Cette commande nécessite un paramètre indiquant l'adresse à modifier. Elle permet la

modification d'adresses successives. Un RETOUR CHARIOT ne modifie pas le contenu d'une mémoire, un point '.' permet de sortir du mode de modification.

T : commande de TRACE. Cette commande permet de suivre un programme en déroulement pas à pas. Un paramètre optionnel permet d'indiquer le nombre de pas à effectuer entre chaque interruption.

U : commande d'arrêt du mode TRACE.

X : commande d'examen et d'altération du contenu des registres. Si la commande X est utilisée seule, le contenu des registres est affiché. Pour modifier un registre, il suffit de faire suivre la commande X d'une lettre avec la convention suivante :

A : accumulateur.
B : paire de registres BC.
D : paire de registres DE.
H : paire de registres HL.
S : pointeur de pile (SP).
P : compteur ordinal (PC).
C : indicateur de CARRY.
Z : indicateur de ZERO.
M : indicateur de NEGATIF.
E : indicateur de PARITE.
I : indicateur de demi-retenue (HALF CARRY).

DUMP.COM Syntaxe : DUMP nom de fichier

Cette commande affiche le contenu du fichier spécifié en hexadécimal sur la console.

ED.COM Syntaxe : ED nom de fichier

Cette commande permet la création directe d'un fichier. Cet éditeur n'est pas très pratique à utiliser.

Fonctions et commandes internes de l'éditeur

CTRL C : démarrage à chaud du système.
CTRL E : CR LF physique.
CTRL H : retour arrière d'un caractère.
CTRL J : tabulation logique.
CTRL L : CR LF logique pour recherche et substitution.
CTRL R : répétition de la ligne.
CTRL U : effacement du mode texte.

UTILITAIRES PROPRES A DIGITAL RESEARCH

CTRL X : passage en vidéo inverse.
CTRL Z : marqueur de fin de chaîne de caractères.

nA : ajouter n lignes.
B ou **+B** : retour au début du tampon d'édition.
-B : aller à la fin du tampon d'édition.
[-]nC : déplacement positif ou négatif de n caractères.
[-]nD : effacement de n caractères en avant ou en arrière.
E : fin d'édition.
nF : recherche d'une chaîne de caractères.
H : fin d'édition, fermeture et réouverture des fichiers.
I : insertion de caractères (sortie par CTRL Z).
nJ : concaténation de chaînes de caractères.
[-]nK : détruit n lignes en avant ou en arrière.
[-]nL : avance le pointeur de n lignes dans un sens ou dans l'autre.
nMxxxx : définition de macrocommande.
nN : recherche de l'occurrence suivante pour la recherche.
O : retour au fichier original.
[-]nP : déplacement et impression de n pages.
Q : quitter l'éditeur sans faire de mise à jour.
R : lecture d'un fichier LIBRAIRIE.
nS : substitution d'une chaîne de caractères.
[-]nT : affichage de n lignes sur la CONSOLE.
[-]U : transformation de MAJUSCULE en minuscule et inversement.
[-]V : vérification des numéros de lignes.
OV : impression de l'espace disponible dans le tampon.
nW : écriture de n lignes.
nZ : sommeil (attente) d'environ n secondes.
[-]n : déplacement avec affichage de n lignes.
n = # : signifie toutes les lignes .

LOAD.COM Syntaxe : LOAD nom de fichier

Cette commande lit le fichier spécifié qui doit posséder une extension .HEX et crée un fichier exécutable avec l'extension .COM.

MOVCPM.COM Syntaxe : MOVCPM

Cette commande permet à l'utilisateur de reconfigurer son CP/M pour un espace mémoire particulier. Deux paramètres optionnels permettent d'indiquer la taille du nouveau système et de garder le nouveau CP/M en mémoire pour une SYSGEN ultérieure.

PIP/COM Syntaxe : PIP 'ligne de commande'

Cette commande est, avec la commande STAT, la plus complexe et la plus puissante des commandes livrées par Digital Research. Ses fonctions sont multiples, PIP est un acronyme pour Peripheral Interchange Program (programme d'échange entre les périphériques), il effectue donc les conversions nécessaires pour l'échange entre la CONSOLE, le LIST, le PUNCHER, le READER et les fichiers sur disques.

PIP peut être lancé seul. Dans ce cas, il affiche un caractère de sollicitation * et attend une ligne de commande.

La forme d'une ligne de commande est :
Destination=source1,source2,source3,...

Destination est de la forme :
[U:][nom de fichier][.extension]

ou bien :
PERIPHERIQUE:

U: représente le nom d'unité disque.
NOM DE FICHIER représente un nom de fichier en clair.
EXTENSION représente une extension en clair.

PERIPHERIQUE représente un périphérique logique.

Les périphériques logiques acceptés sont :
CON : CONSOLE
RDR : READER.
PUN : PUNCHER.
LST : LIST.

Sourcen est de la forme :
[U:][nom de fichier][.extension][options]

ou :
PERIPHERIQUE:[options]

PERIPHERIQUE représente un périphérique logique.

Les périphériques logiques acceptés sont :
CON
RDR
PUN
LST

Mais aussi :

NUL : envoi de 40 caractères 00H au périphérique (pour le PUNCHER).
EOF : envoi d'un marqueur de fin de fichier (1AH).
INP : source d'entrée spéciale pour le PIP.
OUT : destination spéciale de sortie pour le PIP.

PRN : identique à LST, mais les tabulations sont effectuées, les lignes numérotées et un saut de page a lieu toutes les soixante lignes.

Les options sont les suivantes :

B : transfert en mode BLOC.
Dn : effacement des caractères situés après la colonne n.
E : écho de tous les caractères échangés vers la CONSOLE.
F : ôte tous les caractères de saut de page (FORM FEED).
Gn : lecture du fichier depuis l'USER n.
H : transfert de fichier .HEX.
I : ignorer les 00 dans un fichier .HEX (I sélectionne automatiquement l'option H).
L : conversion automatique en minuscules.
N : ajout d'un numéro de ligne à chaque ligne transférée.
O : transfert de fichier objet (les caractères 1AH ne sont pas considérés comme des marqueurs de fin de fichier).
Qs : arrêt de copie lors de la rencontre de la chaîne s.
R : lecture de fichiers système.
Ss : début de la copie à partir de la chaîne s.
Tn : les tabulations se font sur n colonnes.
U : transformation automatique en majuscules.
V : vérification de la copie après écriture.
W : écriture forcée des fichiers même en READ ONLY.
Z : mise à ZERO du bit de parité de chaque caractère ASCII.

Exemples

PIP X.TXT=Y.TXT,Z.TXT,ECT.TXT

Formation du fichier X.TXT par fusion (MERGE) des fichiers Y, Z et ECT, tous trois avec l'extension TXT.

PIP B:TOTO.BAS=A:TITI.BAS

Copie du fichier TITI.BAS de l'unité A vers TOTO.BAS sur l'unité B.

PIP B:=A:*.*

Copie totale du disque A sur le disque B.

PIP LPT:=ESSAI.ASM [nt8u]

Copie vers l'imprimante du fichier ESSAI.ASM avec numérotation des lignes, tabulation de 8 en 8 et conversion en majuscules.

STAT.COM Syntaxe : STAT 'ligne de commande'

La commande **STAT** fournit à l'utilisateur un ensemble de renseignements sur les fichiers ou sur l'assignation des périphériques. En outre, cette commande est capable de modifier l'assignation courante des périphériques.

Rappel : nfa signifie nom de fichier ambigu (pouvant contenir une carte de sélection).

STAT Utilisé seul, fournit l'espace libre sur la disquette courante.

STAT U: Fournit l'espace libre sur l'unité U.

STAT nfa Fournit l'espace occupé par chaque fichier qui cadre avec le nom de fichier ambigu.

STAT U:=R/O Positionne l'unité U en lecture seule.

STAT VAL: Fournit l'état du disque et des possibilités d'assignation de l'I/O byte.

STAT DEV: Fournit l'état courant de l'I/O byte.

STAT DSK: Fournit les caractéristiques du disque.

STAT USER: Fournit la liste des USER qui possèdent des fichiers.

STAT PERLOG:=PERPHY: Permet d'assigner à un périphérique logique (CON, RDR, PUN, LST) un périphérique physique (TTY, CRT, BAT, UC1, PTR, UR1, UR2, PTP, UP1, UP2, LPT, UL1).

STAT nfa $S Fournit un descriptif complet de l'occupation mémoire pour chaque fichier qui cadre avec le nom de fichier ambigu.

STAT nfa $R/O Positionne le ou les fichiers en lecture seule.

STAT nfa $R/W Positionne le ou les fichiers en lecture/écriture.

STAT nfa $SYS Positionne le ou les fichiers en invisible (système).

STAT nfa $DIR Positionne le ou les fichiers en visible.

SUBMIT.COM Syntaxe : SUBMIT nom de fichier par1 par2 ...

La commande **SUBMIT** permet l'exécution de procédures BATCH. Une procédure est un fichier constitué d'une suite d'instructions élémentaires comprises par le SED.

Le nom de fichier doit comporter l'extension SUB.

Les paramètres optionnels permettent de paramétrer la procédure. Les paramètres doivent être précédés du signe $ dans le corps de la procédure.

En général, les procédures sont introduites à l'aide d'un éditeur (ED ou WORDSTAR ou autres).

Exemple

La procédure DEMO.SUB composée des lignes :

```
DIR *.$1
STAT $2.*
```

effectuera la commande DIR *.BAS suivie de la commande STAT ESSAI.* à l'issue de la commande SUBMIT DEMO BAS ESSAI.

SYSGEN.COM Syntaxe : SYSGEN

Cette commande permet la génération d'une disquette système (contenant le CP/M sur les pistes réservées).

Cette commande demande, dans l'ordre, le nom du disque source (en général A) et le nom du disque de destination.

XSUB.COM Syntaxe : XSUB nom de fichier par1 par2 ...

Cette commande est une extension de la commande SUBMIT. L'amélioration se situe au niveau de l'acceptation de commandes propres à un programme en plus des commandes propres au SED.

UTILITAIRES PROPRES A AMSTRAD

En plus des utilitaires standards de Digital Research, AMSTRAD délivre une série de petits utilitaires qui simplifient la vie du programmeur et qui permettent l'utilisation des caractéristiques propres de la machine.

AMSDOS.COM Syntaxe : AMSDOS

Cette commande permet le retour à l'AMSDOS (Basic).

BOOTGEN.COM Syntaxe : BOOTGEN

Cette commande copie le secteur de BOOT (piste 0 secteur 1) et le secteur de configuration.

La commande demande de donner le nom du disque source, ensuite le nom du disque de destination.

CHKDISC.COM Syntaxe : CHKDISC

Cette commande compare la disquette présente dans l'unité A avec la disquette présente dans l'unité B. Elle signale toutes les différences entre les deux disquettes. Cette fonction est intéressante pour vérifier l'état d'une copie physique. Elle est réservée aux possesseurs de deux unités de disque.

CLOAD.COM Syntaxe : CLOAD ["nf cass"] [nf disque]

Cette commande charge le fichier spécifié à partir de la cassette (si le nom est omis, c'est le premier fichier rencontré qui est chargé) et le copie sur disquette sous le nom de fichier indiqué (si ce nom est omis, il sera pris égal au nom sur cassette).

COPYDISC.COM Syntaxe : COPYDISC

Cette commande permet la copie physique d'une disquette sur une autre (copie piste à piste). Elle nécessite deux unités de disque.

CSAVE.COM Syntaxe : CSAVE nf disque [nf cass] n

Cette commande copie le fichier dont le nom est indiqué du disque vers la cassette. Si le nom de fichier cassette est omis, il est pris égal au nom de fichier sur disque. n indique la vitesse d'écriture (0=1000 bauds, 1=2000 bauds).

DISCCHK.COM Syntaxe : DISCCHK

Cette commande permet de comparer deux disquette sur la même unité de disque.

UTILITAIRES PROPRES A AMSTRAD

DISCCOPY.COM Syntaxe : DISCCOPY

Cette commande permet de copier physiquement une disquette sur une autre à l'aide d'une seule unité disque.

DISCKIT2.COM Syntaxe : DISCKIT2

Cette commande affiche un menu qui permet de COPIER, FORMATER ou VERIFIER une disquette. La sélection se fait au moyen des chiffres 1, 4, 7 et 0 du clavier numérique.

DRLKEYS.COM Syntaxe : DRLKEYS

Cette commande réaffecte les touches curseurs et contrôles pour une compatibilité du langage LOGO. Si cette commande n'est pas exécutée avant le chargement du LOGO, l'éditeur de celui-ci est inutilisable.

FILECOPY.COM Syntaxe : FILECOPY nfichier ambigu

Cette commande permet la copie des fichiers correspondant au nom de fichier ambigu sur une seule unité de disque. Pour une copie sur deux unités, il faut utiliser la commande PIP.

FORMAT.COM Syntaxe : FORMAT option

Cette commande permet d'initialiser un disque vierge ou d'effacer totalement le contenu d'une disquette. Quatre formats sont possibles en fonction de l'option :

- pas d'option : format normal ;
- option I : format IBM ;
- option D : format donné ;
- option V : format vendeur (sans CP/M).

FWRESET.COM Syntaxe : FWRESET

Cette commande permet de réinitialiser le système en effaçant l'écran.

SETUP.COM Syntaxe : SETUP

Cette commande permet de configurer le secteur de configuration et donc de définir les principaux paramètres système (tampon, message d'initialisation, imprimante, table de conversion du clavier, I/O byte, délais du moteur du disque, STEP RATE, configuration des SIO,...).

CP/M 3.0 OU CP/M PLUS

ATTENTION CE CHAPITRE EST RESERVE AUX POSSESSEURS DE CPC 6128

En 1982, Digital Research a produit une nouvelle version du CP/M, le **CP/M 3.0** ou **CP/M plus.** Cette version apporte de nombreuses améliorations par rapport au CP/M 2.2.

L'amélioration principale est certainement la possibilité de gérer plus de 64 K de mémoire centrale. Cette gestion est réalisée par une technique dite de BANKING (commutation de partie de mémoire sur les fils d'adresses du processeur).

La diffusion du CP/M 3.0 a été freinée par l'arrivée du MS/DOS qui, au vu du succès remporté par l'IBM PC, s'est imposé comme standard de fait. Cependant, pour les machines 8 bits dont l'AMSTRAD fait partie, le CP/M 3.0 est certainement le meilleur SED disponible actuellement parmi ceux qui assurent une compatibilité entre les machines.

Le CP/M 3.0 est suffisamment différent du CP/M 2.2 pour que nous lui consacrions un chapitre entier. Cependant, nous n'indiquerons ici que les différences avec le CP/M 2.2 Nous ne pouvons donc que conseiller au lecteur une lecture attentive du chapitre précédent.

ORGANISATION MEMOIRE DU CP/M 3.0

Comme le CP/M 2.2, le CP/M 3.0 est composé de trois modules principaux, le **BIOS**, le **BDOS** et le **CCP**.

Contrairement au CP/M 2.2, le système ne réside pas sur les deux premières pistes du disque. Seul le chargeur se trouve sur le premier secteur du disque. Le système se trouve dans un fichier qui porte le nom de C10CPM3.EMS. La taille de ce fichier est d'environ 25 K. En vérité, seule l'extension EMS est importante. Si vous renommez le fichier LOADER.EMS, il s'initialisera de la même façon. Par contre, si l'extension EMS est absente du répertoire, le CP/M 3.0 ne pourra s'initialiser.

Mémoire centrale

Le CPC 6128 est équipé de 128 K de mémoire centrale (RAM). Ces 128 K sont divisés en deux BANQUES (BANKS) de 64 K. Chaque banque est divisée en quatre parties de 16 K.

La **première banque** est appelée **banque 0**. Elle peut être divisée logiquement en deux parties.

La première partie de la banque 0 est composée des 48 K inférieurs (0000H à BFFFH). Elle contient la partie commutable du système d'exploitation (BDOS+BIOS) ainsi que diverses tables de paramètres.

La seconde partie de la banque 0 est composée des 16 K supérieurs (C000 à FFFFH). Elle contient, comme d'habitude, l'image mémoire de l'écran (VIDEORAM). Cette partie n'est jamais en contact direct avec le CP/M. Elle n'est commutée dans le circuit que durant les accès à l'écran.

La **seconde banque** appelée **banque 1** est, elle aussi, divisée en deux parties.

La première partie de la banque 1 est composée des 48 K inférieurs (0000H à BFFFH). Elle contient la page 0 (0000H à 0100H) et la zone TPA (zone disponible pour les programmes de l'utilisateur).

La seconde partie de la banque 1 est composée des 16 K supérieurs (C000H à FFFFH). Elle contient le reste de la TPA ainsi que la partie continuellement résidente du BIOS et du BDOS.

Fonctionnement

D'un point de vue structurel, on peut considérer que la seconde partie de la banque 1 est commune aux deux banques. La commutation des banques ne se fait que sur les 48 K inférieurs.

En vérité, cette partie est commutée lors de l'accès à la mémoire écran.

Le BDOS résident qui est donc toujours présent durant le déroulement normal du CP/M occupe une place de 1,5 K.

Le BIOS résident et les tables occupent une place de 1,5 K.

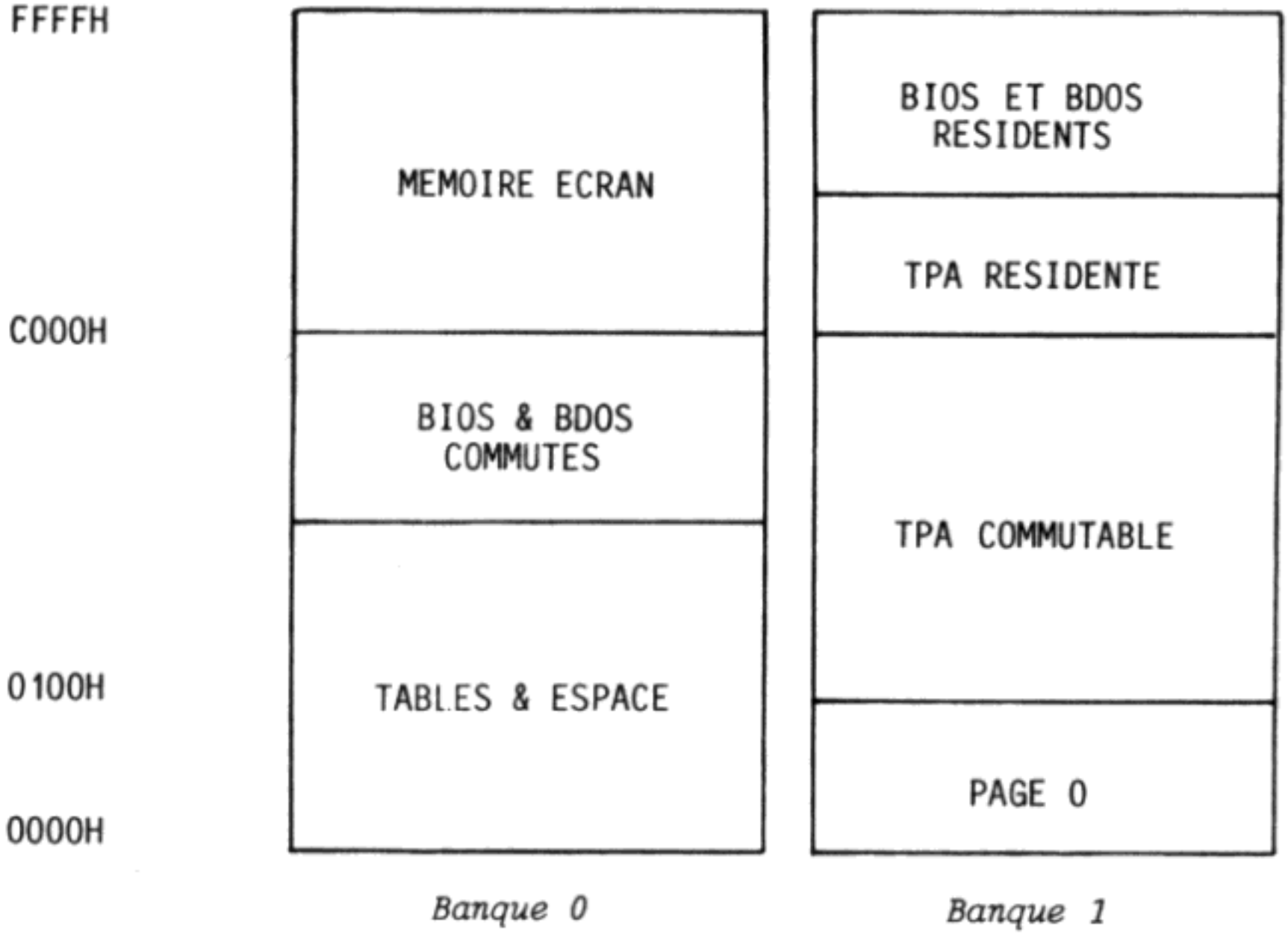
La TPA résidente occupe les 13 K restant de la partie haute de la banque 1, ainsi que les 48 K inférieurs de la même banque. On retrouve bien les 61 K annoncés lors de l'initialisation.

Le BDOS commutable présent dans la banque 0 occupe 8 K.

Le BIOS commutable présent dans la banque 0 occupe environ 3 K.

Ces deux derniers se trouvent dans l'espace 8000H à BFFFH de la banque 0.

Structure



ORGANISATION GENERALE ET FONCTIONNEMENT DU CP/M 3.0

Démarrage à froid ou COLD BOOT

A l'initialisation, le CP/M 3.0 est chargé comme suit :

- chargement en mémoire du chargeur de démarrage (premier secteur de la disquette) ;
- le chargeur charge l'image mémoire du programme chargeur complet CPMLDR (pistes systèmes de la disquette) ;
- CPMLDR lit le fichier SYS qui contient le BDOS et le BIOS. Ensuite, CPMLDR transfère le contrôle à la routine de démarrage à froid du BIOS ;
- le BIOS initialise le matériel et charge le CCP à l'adresse 100H de la TPA. Enfin, le contrôle est passé au CCP.

Après l'initialisation, le système cherche la présence d'un fichier PROFILE.SUB qui est exécuté directement s'il existe. Ce fichier doit être du même format qu'un fichier SUB pour la commande SUBMIT.

Lorsque les routines de démarrage à froid ou à chaud du BIOS prennent le contrôle, elles initialisent deux paramètres système en page 0.

Octets 0, 1 et 2 : installation d'un JUMP au démarrage à chaud (WBOOT) (FC03H).

Octets 5, 6 et 7 : installation d'un JUMP au BDOS (DB00H).

Les périphériques logiques

Dans le CP/M 3.0, les périphériques logiques ont été modifiés par rapport au CP/M 2.2

La CONSOLE et le LIST (imprimante) existent toujours, mais le READER et le PUNCHER ont été modifiés en AUXIN et AUXOUT.

L'I/O byte, tel qu'il existait en version 2.2, n'existe plus et, en CP/M 3.0, un périphérique logique peut se voir attribuer plusieurs périphériques physiques simultanés.

Les périphériques logiques, au nombre de cinq, peuvent être redirigés vers une combinaison quelconque de douze périphériques physiques.

La table de combinaison est contenue dans les octets relatifs 22 à 2B du SCB (System Control Bloc).

Chaque périphérique logique est associé à un mot de 16 bits. Les bits 15 à 4 correspondent aux douze périphériques physiques

(le bit 15 correspond au périphérique physique 0). Les bits 3 à 0 sont réservés au système.

Octets	*Périphérique*
22 & 23	CONIN.
24 & 25	CONOUT.
26 & 27	AUXIN.
28 & 29	AUXOUT.
2A & 2B	LISTOUT.

Les RSXs

Les **RSXs** du CP/M 3.0 ne doivent pas être confondus avec ceux de l'AMSDOS. Si la signification de l'acronyme est identique, l'utilisation est différente.

Le RSX est un module d'extension du système d'exploitation qui peut être ajouté de façon temporaire. Il peut étendre ou modifier une ou plusieurs des fonctions du système. Comme pour le BDOS, les programmes accèdent aux RSXs au travers des appels de fonctions numérotés.

A chaque instant, il peut y avoir aucun, un ou plusieurs RSXs actifs en mémoire. Lorsqu'un programme appelle une des fonctions du BDOS, chaque RSX examine le numéro de la fonction. Si ce numéro est identique au numéro du RSX, il modifie ou complète l'action de la fonction appelée.

C'est l'utilitaire GENCOM qui permet d'attacher le chargement de RSXs particuliers à l'exécution de certains programmes.

Lorsque le CCP charge un programme, il en analyse l'en-tête pour déterminer l'opportunité de charger un RSX ou non. Si le chargement est requis, le LOADER charge le programme dans la TPA puis charge le RSX en haut de la TPA en modifiant le point d'entrée du BDOS (0005H) pour pointer sur le RSX.

Lors de l'existence simultanée de plusieurs RSXs, l'ordre de chargement de ceux-ci affecte l'ordre d'interception des appels au BDOS. C'est le dernier chargé qui interceptera le premier les routines du BDOS.

Les valeurs d'initialisation de la page 0

De	*à*	*Contenu*
0000	0002	JUMP vers le point d'entrée du WARM BOOT.
0003	0004	Réservé.
0005	0007	JUMP vers le BDOS.

De	*à*	*Contenu*
0008	003A	Localisation des routines RESTART.
003B	004F	Non utilisé.
0050		Numéro du disque d'où le programme est chargé.
0051	0052	Adresse du premier champ "mot de passe" dans le tampon DMA par défaut (0080H).
0053		Longueur du "mot de passe".
0054	0055	Adresse du second champ "mot de passe" dans le tampon DMA par défaut.
0056		Longueur du "mot de passe".
0057	005B	Réservé.
005C	007B	FCB par défaut.
007C		Position de l'enregistrement courant dans le FCB par défaut.
007D	007E	Position par défaut de l'enregistrement direct (optionnel).
0080	0100	Tampon DMA par défaut.

Le bloc de contrôle du système (SCB)

Le **SCB** est une zone de 100 octets (64H) qui réside dans le BDOS. Il contient une série de sémaphores et de données propres au BDOS et au CCP, la date et l'heure courantes, ainsi que des informations sur les caractéristiques de la CONSOLE.

Le SCB peut être atteint au moyen de la fonction BDOS numéro 49.

Structure

Remarque *:* RO est utilisé pour indiquer que l'octet peut seulement être lu, RW pour indiquer que l'octet peut être lu et modifié. Les adresses indiquées sont en hexadécimal.

Adresse	*RO/RW*	*Utilisation*
00 - 04	RO	Réservé au système.
05	RO	Numéro de version du BDOS.
06 - 09	RW	Inutilisé. Réservé à l'utilisateur.
0A - 0F	RO	Réservé au système.
10 - 11	RW	Code de retour d'erreur (BDOS fonction 108).
12 - 19	RO	Réservé au système.
1A	RW	Nombre de caractères par ligne de la CONSOLE.
1B	RW	Position courante horizontale de la CONSOLE.
1C	RW	Nombre de lignes par page de la CONSOLE.
1D - 21	RO	Réservé au système.
22 - 2B	RW	Sémaphores de redirection pour chacun des cinq périphériques logiques *(voir page 82, périphériques logiques)*.

Adresse	*RO/RW*	*Utilisation*
2C	RW	Mode page. Si 0, affichage page à page avec arrêt, sinon défilement continu.
2D	RO	Réservé au système.
2E	RW	Si 0, CTRL H (08H) équivaut au BACKSPACE. Si FFH, CTRL H équivaut au RUB/DEL (7FH).
2F	RW	Si 0, RUB/DEL équivaut au RUB/DEL. Si FFH, RUB/DEL équivaut au CTRL H.
30 - 32	RO	Réservé au système.
33 - 34	RW	Mode CONSOLE. Mot de 16 bits qui détermine l'action de certaines fonctions I/O du BDOS.
35 - 36	RO	Réservé au système.
37	RW	Caractère de délimitation de sortie ($). Modifiable par la fonction BDOS 110.
38	RW	Sémaphore de sortie imprimante. 0 = console seule. 1 = console + imprimante.
39 - 3B	RO	Réservé au système.
3C - 3D	RO	Adresse courante du DMA (fonction BDOS 26).
3E	RO	Disque courant (fonction BDOS 25).
3F - 43	RO	Réservé au système.
44	RO	USER courant (fonction BDOS 32).
45 - 49	RO	Réservé au système.
4A	RW	Compteur multisecteur du BDOS (fonction 44).
4B	RW	Mode erreur du BDOS (fonction BDOS 45).
4C - 4F	RW	Ces quatre octets représentent l'ordre de recherche d'un fichier à travers les différentes unités de disque (utilitaire SETDEF).
50	RW	Numéro d'unité disque temporaire.
51	RO	Numéro du disque lors de la dernière erreur.
52 - 56	RO	Réservé au système.
57	RO	Sémaphores du BDOS. Bit 7 = 1 pour message BDOS étendu.
58 - 59	RW	Date en nombre de jours écoulés depuis le 1er Janvier 1978.
5A	RW	Heure en BCD.
5B	RW	Minutes en BCD.
5C	RW	Secondes en BCD.
5D - 5E	RO	Adresse de base de la mémoire commune. Ce renseignement est utilisé par le BANK MANAGER.
5F - 63	RO	Réservé au système.

La table disque et la DPH (Disk Parameter Header)

La table disque est une simple table de 16 fois 2 octets.

Chaque entrée de la table est constituée par l'adresse de la DPH de chacune des unités de disque possibles (A à P).

Si une unité n'existe pas dans le système, l'adresse associée à la DPH vaut 0.

La DPH est une table de 25 octets qui contient les informations principales sur le disque.

Structure de la DPH

XLT	-0-	MF	DPB	CSV	ALV	DIRBCB	DTABCB	HASH	HBANK
16	72	8	16	16	16	16	16	16	8

Remarque *:* les tailles sont indiquées en nombre de bits.

XLT : cette zone contient l'adresse éventuelle d'une table de translation des numéros de secteurs logiques en numéros de secteurs physiques.

-0- : cette zone est une zone de travail de neuf octets réservée au BDOS.

MF : sémaphore d'utilisation de l'unité disque. Permet de détecter l'ouverture de porte et, par là, de considérer que la disquette a été changée.

DPB : adresse de la table DPB *(voir point suivant)*.

CSV : adresse de la zone CSV (tampon pour le test logiciel de changement d'unité disque).

ALV : adresse d'un tampon utilisé par le BDOS pour mémoriser les informations sur l'allocation du disque.

DIRBCB : adresse d'un en-tête du BCB de répertoire (Buffer Control Bloc).

DTABCB : adresse d'un en-tête de BCB de données.

HASH : adresse d'une table optionnelle de HASHING. FFFFH signifie qu'il n'y a pas de table.

HBANK : adresse de la banque mémoire qui contient la table de HASHING.

La DPB (Disk Parameter Bloc)

Cette table est presque identique à celle décrite dans le chapitre réservé au CP/M 2.2.

La table DPB du CP/M est limitée à 17 octets.

Les 15 premiers (00 à 0E) sont identiques à ceux du CP/M 2.2. Reportez-vous au chapitre précédent pour de plus amples informations.

Les deux autres sont :

PSH : facteur de SHIFT des enregistrements physiques.
PHM : masque des enregistrements physiques.

Taille secteur	*PSH*	*PHM*
128	0	0
256	1	1
512	2	3
1024	3	7
2048	4	15
4096	5	31

La table BCB (Buffer Control Bloc)

La table **BCB** se compose de 15 octets et localise les tampons d'enregistrement physique du BDOS.

Le BDOS distingue deux types de BCB : le BCB de répertoire et le BCB de données.

Octet	*Nom*	*Contenu*
1	DRV	Identifie le disque associé à l'enregistrement contenu dans le tampon dont l'adresse est dans BUFFAD.
2 à 4	REC#	Numéro d'enregistrement dudit enregistrement.
5	WFLG	Sémaphore indiquant que la donnée est nouvelle et n'a pas été écrite sur disque.
6	00	Réservé.
7 & 8	TRACK	Numéro de piste de l'enregistrement.
9 & 10	SECTOR	Numéro de secteur de l'enregistrement.
11 & 12	BUFFAD	Adresse du tampon qui contient l'enregistrement.
13	BANK	Adresse de la banque mémoire qui contient l'enregistrement.
14 & 15	LINK	Adresse du prochain BCB.

Le FCB (File Control Bloc)

Le **FCB** est une table initialisée par un programme transitoire et utilisée par les fonctions BDOS d'accès aux fichiers.

Le FCB est donc le lien principal d'échange entre un programme d'application et le BDOS.

Le FCB est composé de 36 octets (0 à 23H).

Structure

Octet	*Utilisation*
0	Numéro du disque.
1 à 8	Nom du fichier en ASCII.
9 à B	Extension du nom.
C	Numéro d'extension courant.
D & E	Réservé.
F	Compteur d'enregistrement.
10 à 1F	Réservé.
20	Numéro d'enregistrement courant pour les accès séquentiels.
21 à 23	Numéro d'enregistrement direct.

Remarque *:* les bits les plus significatifs (B7) des trois octets de l'extension (9 à B) indiquent respectivement :

9 si 1 → fichier READ ONLY.
A si 1 → fichier SYSTEM.
B si 1 → fichier déjà archivé.

Les tables XFCB et SFCB

La table **XFCB** est une table optionnelle associée à un fichier dans le répertoire. Si elle existe, elle contient le "mot de passe".

Structure XFCB

Octet	*Fonction*
0	Numéro du disque.
1 à 8	Nom du fichier.
9 à B	Extension.
C	Mode du mot de passe (B7 = READ, B6 = WRITE, B5 = DELETE).
D à F	Réservé.
10 à 17	Mot de passe encrypté.
18 à 1F	Réservé.

La table SFCB est un complément optionnel à l'entrée du répertoire. Elle contient les informations de date et heure.

La table SFCB réside dans le répertoire, on la retrouve toutes les quatre entrées.

Structure SFCB

La **SFCB** se compose de 32 octets, comme une entrée normale du répertoire.

Le premier vaut 21H pour identifier la SFCB.

Le reste est divisé en trois champs de dix octets (un pour chacune des trois entrées précédentes).

Les dix octets sont utilisés comme suit :

```
Octets 1 à 4 : date et heure de création ou d'accès.
Octets 5 à 8 : date et heure de mise à jour.
Octet  9     : mode du mot de passe.
Octet 10     : réservé.
```

TABLE DES VECTEURS STANDARDS DU BIOS DU CP/M 3.0

Le CP/M 3.0 de l'AMSTRAD contient 33 vecteurs standards d'appel de routines du BIOS. La table de JUMPS se trouve à l'adresse FC00H.

Numéro	*Adresse*	*Saut à*	*Nom*	*Fonction*
0	FC00	3D0C	**BOOT**	Initialisation principale du système et sélection du disque A si le registre C contient 0. En fin d'initialisation, transfert au CCP.
1	FC03	FC7A	**WBOOT**	Démarrage à chaud.
2	FC06	FDE2	**CONST**	Retourne dans A l'état de la CONSOLE.
3	FC09	FDFA	**CONIN**	Lecture de la CONSOLE. Résultat dans A.
4	FC0C	FDC3	**CONOUT**	Sortie du caractère contenu dans A vers la console.
5	FC0F	FDBE	**LIST**	Sortie du caractère contenu dans A vers le périphérique LIST.
6	FC12	FDB9	**AUXOUT**	Ecriture du caractère contenu dans A vers le périphérique AUX.
7	FC15	FDF5	**AUXIN**	Lecture du périphérique AUX. Résultat dans A.
8	FC18	3E74	**HOME**	Retour du disque courant à la piste 0.
9	FC1B	FE06	**SELDSK**	Sélection du disque dont le numéro est contenu dans le registre C.
10	FC1E	3E77	**SETTRK**	Sélection de la piste dont le numéro est contenu dans le registre C.
11	FC21	3E7C	**SETSEC**	Sélection du secteur dont le numéro est contenu dans le registre C.
12	FC24	3E81	**SETDMA**	Sélection de l'adresse du tampon d'écriture/lecture de l'enregistrement courant. L'adresse sélectionnée doit être contenue dans BC.

TABLE DES VECTEURS STANDARDS DU BIOS DU CP/M 3.0

Numéro	*Adresse*	*Saut à*	*Nom*	*Fonctïon*
13	FC27	FE0B	**READ**	Lecture de l'enregistrement désigné par SETTRK et SETSEC dans le tampon désigné par SETDMA. A contient 0 si l'opération a réussi et 1 si non.
14	FC2A	FE10	**WRITE**	Ecriture de l'enregistrement désigné *(voir lecture)*.
15	FC2D	FDD0	**LISTST**	Fournit l'état du périphérique LIST dans le registre A (0 = NOT READY).
16	FC30	3E8C	**SECTRN**	Convertit un numéro de secteur logique en numéro de secteur physique (BC contient le numéro du secteur et DE pointe sur une table de conversion). En sortie, HL contient le numéro physique du secteur.
17	FC33	FDD5	**CONOST**	Lit le statut de la console. En sortie, A contient FFH si la console est prête pour afficher le caractère suivant et 0 si non.
18	FC36	FDDD	**AUXIST**	Lit l'état du port AUX en entrée. En sortie, A contient FFH si le port est prêt et 0 si non.
19	FC39	FDCB	**AUXOST**	Lit l'état du port AUX en sortie. En sortie, A contient FFH si le port est prêt à accepter le caractère suivant et 0 si non.
20	FC3C	FDB5	**DEVTBL**	Cette fonction retourne dans HL l'adresse de la table d'assignation des périphériques (DEVICE). Cette fonction remplace avantageusement les routines de gestion de l'I/O byte du CP/M 2.2.
21	FC3F	FDB0	**DEVINI**	Initialise le périphérique dont le numéro est contenu dans le registre C avec les valeurs contenues dans CHRTBL.

TABLE DES VECTEURS STANDARDS DU BIOS DU CP/M 3.0

Numéro	*Adresse*	*Saut à*	*Nom*	*Fonction*
22	FC42	FE02	**DRVTBL**	Fournit dans HL l'adresse de la table des DPH. La valeur FFFFH signale qu'il n'y a pas de table et que le HASHING est supporté. La valeur FFFEH signale qu'il n'y a pas de table et que le HASHING n'est pas supporté.
23	FC45	3EB4	**MULTIO**	Positionne le nombre de secteurs consécutifs à lire ou à écrire. En entrée, C contient le nombre de secteurs.
24	FC48	3EB9	**FLUSH**	Permet de forcer l'écriture d'un secteur si le BIOS utilise la technique du "SECTOR BLOCKING". En sortie, A contient un compte-rendu d'erreur (0 si l'opération a réussi, 1 si une erreur physique s'est produite et 2 si le disque est READ ONLY).
25	FC4B	FCC0	**MOVE**	Permet d'effectuer une copie d'un bloc mémoire. En entrée, HL contient l'adresse de destination, DE l'adresse de départ et BC le nombre d'octets à transférer. Cette fonction se comporte comme le LDIR du Z80, mais avec une inversion de DE et HL.
26	FC4E	FE15	**TIME**	Si C contient 0, cette fonction indique au BIOS qu'il doit positionner la date et l'heure en fonction des champs du SCB. Si C contient FFH, le BDOS a positionné une nouvelle valeur pour la date et l'heure dans le SCB et le BIOS peut mettre à jour son horloge.
27	FC51	FD04	**SELMEM**	Sélectionne le BANK mémoire dont le numéro est contenu dans A.

TABLE DES VECTEURS STANDARDS DU BIOS DU CP/M 3.0

Numéro	*Adresse*	*Saut à*	*Nom*	*Fonction*
28	FC54	3E88	**SETBNK**	Spécifie le BANK mémoire pour les prochains accès au DMA (READ WRITE). A contient le numéro du BANK en entrée.
29	FC57	FCB6	**XMOVE**	Positionne le BANK pour le prochain déplacement de mémoire (MOVE). En entrée, B contient le numéro du BANK de destination et C le numéro du BANK d'origine.
30	FC5A	FD10	**USERF**	Réservé à AMSTRAD.
31	FC5D	0000	**RESER1**	Non utilisé.
32	FC60	0000	**RESER2**	Non utilisé.

TABLE DES VECTEURS STANDARDS DU BDOS DU CP/M 3.0

Les vecteurs standards du BDOS sont appelés en disposant le numéro de la fonction dans le registre C et en effectuant un CALL à l'adresse 0005H (saut au BDOS).

En sortie, **A** peut contenir un code d'erreur avec les conventions suivantes :

Valeur	*Signification*
0	Opération réussie.
1	Lecture de données inexistantes ou espace insuffisant dans le répertoire.
2	Pas de bloc de données disponible.
3	Extension courante impossible à fermer.
4	Déplacement vers une extension inexistante.
5	Pas d'espace dans le répertoire.
6	Numéro d'enregistrement hors limite.
9	FCB invalide.
0AH	Disquette changée.
FFH	Erreur physique : le code d'erreur se trouve dans H.

Les **fonctions** 15, 16, 17, 18, 19, 22, 23, 30, 35, 99, 100, 102 et 103 fournissent un code de répertoire dans le registre A. Un nombre entre 0 et 3 indique la position de l'entrée du répertoire dans le bloc de 128 octets du répertoire, FFH indique que la fonction a échoué.

Les **codes d'erreurs physiques** fournis dans le registre H ont la signification suivante :

Valeur	*Signification*
00	Pas d'erreur.
01	DISK I/O ERROR.
02	Disque à lecture seule (READ ONLY).
03	Fichier à lecture seule ou fichier protégé par mot de passe.
04	Numéro d'unité disque invalide.
07	Erreur de mot de passe.
08	Fichier déjà existant.
09	Nom de fichier ambigu non utilisable.

Fonction 0 : Initialisation du système

Action : réinitialisation totale du système. Cette fonction est identique à un saut au BOOT (0000H).
CE : C = 00.
CS : pas de CS.

Fonction 1 : Lecture CONSOLE

Action : lecture dans l'accumulateur du caractère entré sur le périphérique physique associé à la console. Les caractères CR, LF, BS et TAB sont traités ainsi que le test du CTRL S et CTRL P.
CE : C = 01.
CS : A contient le caractère ASCII.

Fonction 2 : Ecriture CONSOLE

Action : écriture dans le périphérique physique associé à la console du caractère contenu dans le registre E. Les caractères TAB, CR, LF, BS, CTRL S et CTRL P sont traités.
CE : C = 02, E contient le code ASCII du caractère à sortir.
CS : pas de CS.

Fonction 3 : Lecture du port AUXILIAIRE

Action : lecture du caractère en provenance du périphérique physique associé au AUXIN.
CE : C = 03.
CS : A contient le code ASCII du caractère lu.

Fonction 4 : Ecriture dans le port AUXILIAIRE

Action : écriture du caractère contenu dans le registre E dans le périphérique physique associé au AUXOUT.
CE : C = 04, E contient le caractère à écrire.
CS : pas de CS.

Fonction 5 : Ecriture LIST

Action : écriture du caractère contenu dans le registre E dans le périphérique physique associé au LIST.
CE : C = 05, E contient le caractère à écrire.
CS : pas de CS.

Fonction 6 : Lecture ou écriture directe sur la CONSOLE

Action : cette fonction ne traite pas les caractères spéciaux (CTRL P, CTRL S,...) et effectue un traitement direct avec la console.
CE : C = 06, E contient le caractère à écrire ou un code avec les conventions suivantes :
FFH : lecture de la console. S'il n'y a pas de caractère disponible, en sortie, A contient 0.
FEH : lecture du statut de la console. A contient 0 s'il n'y a pas de caractère disponible et FFH si non.

FDH : lecture avec attente de caractère disponible.
CS : A contient le caractère lu ou un état (STATUS), 00 s'il n'y a pas de caractère lu.

Fonction 7 : Lecture du statut du port AUXILIAIRE en entrée

Action : lecture du statut du port auxiliaire utilisé en entrée (AUXIN) dans le registre A.
CE : C = 07.
CS : A contient le statut du port auxiliaire.

Fonction 8 : Lecture du statut du port AUXILIAIRE en sortie

Action : lecture du statut du port auxiliaire utilisé en sortie (AUXOUT) dans le registre A.
CE : C = 08.
CS : A contient le statut.

Fonction 9 : Impression d'une chaîne de caractères

Action : imprime la chaîne de caractères pointée par le registre DE vers la console. La chaîne doit être terminée par un signe $.
CE : C = 09, DE pointe sur la chaîne de caractères.
CS : pas de CS.

Fonction 10 : Lecture du tampon de la CONSOLE

Action : lit la console et pousse son contenu dans un tampon pointé par le registre DE. L'entrée se termine lorsque le tampon déborde ou lorsqu'un RETOUR CHARIOT (0DH) ou un LINE FEED (0AH) est introduit à la console. A l'issue de la fonction, le tampon contient, comme premier octet, le nombre de caractères total que peut contenir le tampon et le second octet contient le nombre de caractères qui suivent dans le tampon.
CE : C = 0AH, DE pointe sur le tampon, le premier caractère pointé par DE contient le nombre maximum de caractères accepté par le tampon (1 à 255).
CS : le tampon contient les caractères entrés.

Fonction 11 : Lecture de l'état de la CONSOLE

Action : cette fonction teste si un caractère a été introduit à la console. Si un caractère a été introduit, A contient FFH, sinon A contient 00.
CE : C = 0BH.
CS : A contient l'état de la console.

Fonction 12 : Lecture du numéro de version du CP/M

Action : cette fonction fournit une valeur sur deux octets. Le CP/M 3.0 fournit 0 dans le registre H et 31H dans le registre L.
CE : C = 0CH.
CS : HL contient le numéro de version.

Fonction 13 : Réinitialisation du système disque

Action : cette fonction réinitialise le disque (disque A par défaut) et positionne le système en lecture et écriture permises. Cette fonction sert principalement à permettre un changement de disque sans faire de BOOT.
CE : C = 0DH.
CS : pas de CS.

Fonction 14 : Sélectionne une unité de disque

Action : cette fonction sélectionne l'unité de disque dont le numéro est contenu dans le registre E (0 = A, 1 = B,...). Le disque est immédiatement considéré comme disponible. Tout essai de changement de disque, après le lancement de cette fonction, produira un message d'erreur (READ/ONLY STATUS).
CE : C = 0EH, E contient le numéro de l'unité.
CS : A contient un indicateur d'erreur (0 si OK et FFH si erreur).
H contient 01 si l'erreur est du type DISK I/O et 04 si le numéro du disque est invalide.

Fonction 15 : Ouverture de fichier

Action : cette fonction ouvre le fichier dont le nom est contenu dans le FCB *(voir table)* pointée par DE. En retour, cette fonction fournit un état dans le registre A (0, 1, 2 et 3 indiquent une ouverture correcte, FFH indique que le fichier ne peut être trouvé dans le répertoire).
Remarque *:* les nombres 0, 1, 2 ou 3 indiquent la position du nom du fichier dans l'enregistrement du répertoire (128 octets permettent 4 entrées de 32 octets).
CE : C = 0FH, DE pointe sur le FCB.
CS : A contient le statut d'ouverture.
H contient le code d'erreur éventuel.

Fonction 16 : Fermeture de fichier

Action : cette fonction ferme le fichier dont le FCB est pointé par DE. Un état d'échec ou de succès identique à celui de l'ouverture est fourni en sortie. Une fermeture est indispensable si une ou plusieurs écritures ont eu lieu dans le fichier.

CE : C = 10H, DE pointe sur le FCB du fichier.
CS : A contient le statut de fermeture.
H contient le code d'erreur éventuel.

Fonction 17 : Première recherche dans le répertoire

Action : cette fonction recherche un nom de fichier dans le répertoire en commençant au début et fournit dans A un statut indiquant la position ou l'absence du fichier spécifié.
CE : C = 11H, DE pointe sur le FCB du fichier.
CS : A contient le statut de la recherche.
H contient le code d'erreur éventuel.

Fonction 18 : Continuer la recherche

Action : cette fonction est identique à la précédente, mais la recherche continue à partir de l'endroit où la dernière recherche s'est arrêtée.
CE : C = 12H.
CS : A contient le statut de la recherche.
H contient le code d'erreur éventuel.

Fonction 19 : Effacement de fichier

Action : cette fonction efface du répertoire le fichier dont le FCB est pointé par le registre DE. Si le fichier n'est pas trouvé, en sortie A contient FFH.
CE : C = 13H, DE pointe sur le FCB du fichier.
CS : A contient le statut.
H contient le code d'erreur éventuel.

Fonction 20 : Lecture séquentielle d'un fichier

Action : cette fonction réalise la lecture de l'enregistrement suivant (128 octets) du fichier préalablement ouvert (fonction 15 ou 22).
CE : C = 14H, DE pointe sur le FCB du fichier.
CS : A contient 0 si la lecture est bonne, 09 si le FCB est invalide, 0A si la disquette a été changée et FFH si l'erreur est physique (code dans H).
H contient un code d'erreur éventuel.

Fonction 21 : Ecriture séquentielle dans un fichier

Action : cette fonction lit l'enregistrement suivant (128 octets) d'un fichier préalablement ouvert.
CE : C = 15H, DE pointe sur le FCB du fichier.
CS : A contient 0 si l'écriture est correcte, 01 s'il n'y a pas de place dans le répertoire, 02 s'il n'y a pas de bloc de données disponible, 09 si le FCB est invalide, 0A si la disquette a été changée et FFH si l'erreur est physique (code dans H).
H contient un code d'erreur éventuel.

Fonction 22 : Création d'un fichier

Action : cette fonction est similaire à l'ouverture d'un fichier, mais le FCB doit faire référence à un fichier qui n'existe pas dans le répertoire. L'entrée correspondante est alors créée. En sortie, A contient un compte-rendu identique à celui d'une ouverture.
CE : C = 16H, DE pointe sur le FCB du fichier.
CS : A contient le statut de la création.
H contient un code d'erreur éventuel.

Fonction 23 : Modification du nom d'un fichier

Action : cette fonction utilise le FCB pointé par DE pour modifier le nom du fichier contenu dans les 16 premiers octets en celui contenu dans les 16 octets suivants. En sortie, A contient un compte-rendu identique à celui d'une ouverture.
CE : C = 17H, DE pointe sur le FCB.
CS : A contient le statut du changement de nom.
H contient un code d'erreur éventuel.

Fonction 24 : Lecture du vecteur d'état des disques

Action : cette fonction fournit un mot de 16 bits dans le registre HL qui indique les disques présents dans le système (le bit 0 de L correspond au disque A et le bit 7 de H au disque P). Si le disque est présent, le bit est à 1.
CE : C = 18H.
CS : HL contient le vecteur d'état.

Fonction 25 : Retourne le numéro du disque courant

Action : cette fonction fournit, dans le registre A, le numéro du disque courant (sélectionné au moment de l'appel de la fonction).
CE : C = 19H.
CS : A contient le numéro du disque (0 à 15).

Fonction 26 : Positionne l'adresse du DMA

Action : cette fonction positionne l'adresse du tampon qui contient l'enregistrement à lire ou à écrire. En standard, cette valeur vaut 80H.
CE : C = 1AH.
CS : DE contient l'adresse du DMA.

Fonction 27 : Lecture de l'adresse d'ALLOC

Action : cette fonction fournit dans HL l'adresse du vecteur d'allocation (ALLOC).
CE : C = 1BH.
CS : HL contient l'adresse du vecteur ALLOC.

Fonction 28 : Positionne le disque en WRITE PROTECT

Action : cette fonction positionne le disque courant en protection d'écriture jusqu'au prochain démarrage à chaud ou à froid.
CE : C = 1C.
CS : pas de CS

Fonction 29 : Lecture du vecteur READ/ONLY

Action : cette fonction fournit dans HL un vecteur de 16 bits qui indique les disques qui possèdent le statut lecture seule (READ ONLY). Le bit 0 du registre L indique le statut du disque A.
CE : C = 1DH.
CS : HL contient le vecteur READ/ONLY.

Fonction 30 : Positionnement des attributs d'un fichier

Action : cette fonction permet de positionner l'attribut READ/ONLY ou l'attribut SYSTEME (fichier invisible) dans un état ou dans l'autre (ACTIF ou INACTIF).
CE : C = 1EH, DE pointe sur le FCB du fichier spécifié avec les attributs positionnés dans l'état désiré (bit 7 des deux premiers caractères de l'extension).
CS : A contient FF si le fichier n'est pas trouvé. Sinon, A contient un nombre compris entre 0 et 3 qui indique la position de l'entrée dans l'enregistrement du répertoire. H contient le code d'erreur physique éventuel.

Fonction 31 : Lecture de l'adresse de la DPB

Action : cette fonction fournit, dans le registre HL, l'adresse de la DPB (Disk Parameter Bloc) du disque courant.
CE : C = 1FH.
CS : HL pointe sur la DPB.

Fonction 32 : Positionnement ou lecture du numéro d'USER

Action : cette fonction fournit le numéro d'USER courant si le registre E contient FFH ou modifie le numéro d'USER si E contient un autre nombre. Le numéro d'USER est pris égal à E modulo 16.
CE : C = 20H, E contient FFH ou le numéro d'USER.
CS : si E contient FFH, alors A contient le numéro d'USER courant.

Fonction 33 : Lecture directe

Action : cette fonction est presque identique à la fonction de lecture séquentielle (fonction 20). L'enregistrement lu n'est pas le suivant du fichier, son numéro est contenu

dans les trois octets qui suivent l'octet 31H du FCB. L'enregistrement est écrit à l'adresse DMA.
En retour, le registre A contient un code d'erreur. 0 indique une lecture correcte, 3 indique que l'enregistrement courant ne peut être fermé, 4 indique que l'enregistrement demandé n'existe pas encore et 6 indique que l'enregistrement demandé dépasse la taille du disque.

CE : C = 21H, DE pointe sur le FCB du fichier.
CS : A contient le compte-rendu d'erreur.
H contient le code d'erreur physique éventuel.

Fonction 34 : Ecriture directe

Action : cette fonction est presque identique à la fonction d'écriture séquentielle (fonction 21). Comme dans la lecture directe (fonction 33), le numéro d'enregistrement est lu dans trois derniers octets du FCB (32 à 35). L'enregistrement doit être à l'adresse DMA. En retour, A fournit un code d'erreur identique à celui de la fonction 33 avec, en plus, un code 5 qui indique qu'un nouvel enregistrement ne peut être créé pour des raisons de débordement de répertoire.

CE : C = 22H, DE pointe sur le FCB du fichier.
CS : A contient le compte-rendu d'erreur.
H contient le code d'erreur physique éventuel.

Fonction 35 : Calcul de la taille d'un fichier

Action : cette fonction permet de déterminer la taille d'un fichier. Le résultat est fourni dans les octets 32 et 33 (ceux qui contiennent le numéro d'enregistrement direct à atteindre).

CE : C = 23H, DE pointe sur le FCB du fichier.
CS : les octets 32 et 33 du FCB contiennent la taille du fichier exprimée en nombre d'enregistrements.
A contient un compte-rendu d'erreur.
H contient le code d'erreur physique éventuel.

Fonction 36 : Positionne le numéro d'enregistrement

Action : cette fonction positionne, dans les octets 32 et 33 du FCB, le numéro direct d'enregistrement. Cette routine est intéressante dans le cas où le fichier a été préalablement adressé en mode séquentiel.

CE : C = 24H, DE pointe sur le FCB du fichier.
CS : Les octets 32 et 33 du FCB contiennent le numéro d'enregistrement.

Fonction 37 : Réinitialisation du disque

Action : initialise une configuration disque.
CE : C = 25H, DE contient le vecteur d'activité des disques (16 bits).
CS : A = 0.

Fonction 40 : Ecriture d'un enregistrement avec des 0

Action : cette fonction est identique à la fonction 34, mais l'enregistrement est rempli de 0.
CE : C = 28H, DE pointe sur le FCB du fichier.
CS : A contient un code d'erreur *(voir fonction 34)*.
H contient un code d'erreur physique éventuel.

Fonction 42 : Verrouillage d'un enrgistrement

Action : cette fonction est prévue uniquement pour rendre le CP/M 3.0 compatible avec MP/M. En CP/M, cette fonction ne fait rien.
CE : C = 2AH, DE pointe sur le FCB.
CS : A = 0.

Fonction 43 : Déverrouillage d'un enregistrement

Action : cette fonction est prévue uniquement pour rendre le CP/M 3.0 compatible avec MP/M. En CP/M, cette fonction ne fait rien.
CE : C = 2BH, DE pointe sur le FCB.
CS = A = 0.

Fonction 44 : Positionnement du compteur multisecteurs

Action : cette fonction positionne un compteur de secteur. Il permet, à la prochaine opération de lecture ou d'écriture, de lire ou d'écrire de 1 à 128 enregistrements de 128 octets en une opération.
CE : C = 2CH, E contient le nombre de secteurs à traiter.
CS : A contient 0 si le positionnement est bien exécuté et FFH si le nombre de secteurs spécifié dépasse 128. Si une erreur intervient durant l'opération, le nombre de secteurs traités avec succès se trouve dans le registre H.

Fonction 45 : Positionne le MODE ERREUR du BDOS

Action : cette routine permet de déterminer comment les erreurs physiques sont traitées par le BDOS. Trois modes existent : le mode par défaut où le BDOS affiche un message d'erreur à la console et arrête l'exécution du programme en cours, le mode retour où le BDOS positionne le registre A à FFH, place un code d'erreur dans le regis-

tre H et retourne au programme appelant (utilisé par les routines internes), enfin le mode retour et affichage dans lequel le BDOS affiche le message d'erreur à la console et retourne à la routine appelante.
CE : C = 2DH, E contient FFH pour positionner le mode retour, FEH pour positionner le mode retour et affichage, une autre valeur pour positionner le mode par défaut.
CS : pas de CS.

Fonction 46 : Lecture de l'espace libre sur le disque

Action : cette fonction détermine le nombre d'enregistrements de 128 octets libres.
CE : C = 2EH, E contient le numéro du disque.
CS : A contient le sémaphore d'erreur.
H contient le code d'erreur physique éventuel.
Les trois premiers octets du tampon DMA courant contiennent le nombre d'enregistrements libres.

Fonction 47 : Enchaînement de programme

Action : cette fonction permet de lancer un programme sans intervention de l'utilisateur. Le programme appelé doit se trouver dans le tampon DMA et être terminé par un octet à 00H. Si le registre E vaut FFH, le CCP initialise le numéro d'USER et le lecteur par défaut lors du passage du contrôle au programme appelé. La fonction 108 permet de passer un argument sur deux octets au programme appelé.
CE : C = 2FH, E vaut éventuellement FFH.
CS : dépendent du programme appelé.

Fonction 48 : Ecriture forcée des tampons

Action : cette fonction force l'écriture de tous les tampons utilisés par la technique du "BLOCKING/DEBLOCKING". Si le registre E vaut FFH, la fonction nettoie tous les tampons de données actifs.
CE : C = 30H, E peut valoir FFH.
CS : A vaut 0 si l'opération réussit ou FFH si l'opération échoue.
H contient un code d'erreur physique éventuel.

Fonction 49 : Lecture ou modification du SCB

Action : cette fonction permet d'accéder aux paramètres du SCB (System Control Bloc). Le registre DE doit pointer sur un descriptif appelé SCBPB (Parameter Bloc). Le SCBPB

se compose de 2, 3 ou 4 octets qui ont la structure suivante :
SCBPB+0 : offset dans le SCB à atteindre.
SCBPB+1 : type d'opération : FFH = écriture d'un octet.
FEH = écriture d'un mot.
00H = lecture.
SCBPB+2 : octet ou mot (16 bits) à écrire.

CE : C = 31H, DE pointe sur le SCBPB.
CS : A contient l'octet lu éventuel et HL contient le mot lu éventuel.

Fonction 50 : Appel direct du BIOS.

Action : cette fonction permet d'appeler directement le BIOS à travers le BDOS. Le registre DE doit pointer sur un BIOSPB (Parameter Bloc). Le BIOSPB a la structure suivante :
BIOSPB+0 : numéro de la fonction BIOS à appeler.
BIOSPB+1 : contenu du registre A à passer.
BIOSPB+2 : contenu du registre BC à passer (2 octets).
BIOSPB+4 : contenu du registre DE à passer (2 octets).
BIOSPB+6 : contenu du registre HL à passer (2 octets).

CE : C = 32H, DE pointe sur le BIOSPB.
CS : le résultat de la fonction BIOS appelée.

Fonction 59 : Chargement d'un module de recouvrement

Action : cette fonction permet de charger un module de recouvrement (OVERLAY). Cette fonction est réservée aux programmes possédant un en-tête RSX. Les fichiers de recouvrement peuvent être du type PRL (RELOCATABLE).

CE : C = 38H, DE pointe sur l'adresse du FCB du fichier à charger.
CS : A contient 0 si l'opération réussit et FF si elle échoue.
H contient un code d'erreur physique éventuel.

Fonction 60 : Appel d'extension résidente au système

Action : cette fonction permet d'appeler un RSX. L'appel est réalisé par l'intermédiaire d'un bloc de contrôle appelé RSXPB. Il possède la structure suivante :
RSXPB+0 : numéro de la fonction RSX.
RSXPB+1 : nombre de paramètres de 16 bits.
RSXPB+2 : premier paramètre.
..... : paramètres suivants.

CE : C = 3CH, DE pointe sur le RSXPB.
CS : A contient 00 si l'opération réussit, sinon A contient FFH.
H contient un code d'erreur physique éventuel.

Fonction 98 : Récupération des blocs libres

Action : cette fonction fouille les disques déclarés et libère les blocs de données occupés temporairement sur chaque disque. Un bloc occupé temporairement est un bloc qui a été écrit mais non fermé par une fonction CLOSE.
CE : C = 62H.
CS : A contient 0 si l'opération réussit ou FFH si elle échoue.
H contient un code d'erreur physique éventuel.

Fonction 99 : Troncature de fichier

Action : cette fonction positionne le dernier enregistrement d'un fichier au numéro d'enregistrement direct contenu dans les paramètres R0, R1 et R2 du FCB du fichier considéré.
CE : C = 63H, DE pointe sur le FCB du fichier.
CS : A contient 0 si l'opération réussit ou FFH si elle échoue.
H contient un code d'erreur physique éventuel.

Fonction 100 : Positionnement de l'étiquette disque

Action : cette fonction crée une étiquette (un identificateur) pour le disque spécifié. Le programme passe l'adresse d'un FCB qui contient le nom, le type ainsi que les champs à assigner au disque. L'octet relatif 12 du FCB contient les champs de l'étiquette avec la convention suivante :
Bit 7 : mot de passe requis.
Bit 6 : activation de la date et heure d'accès.
Bit 5 : activation de la date et heure de mise à jour.
Bit 4 : activation de la date et heure de création.
Bit 0 : assignation d'un nouveau mot de passe.
CE : C = 64H, DE pointe sur le FCB.
CS : A contient 0 si l'opération réussit ou FFH si l'opération échoue.

Fonction 101 : Lecture de l'étiquette disque

Action : cette fonction retourne dans le registre A les champs d'assignation de l'étiquette tels qu'ils sont définis dans la fonction 100.
CE : C = 65H et E contient le numéro du lecteur de disque.
CS : A contient la valeur lue.
H contient un code d'erreur physique éventuel.

Fonction 102 : Lecture des paramètres d'un fichier

Action : cette fonction fournit, dans le FCB du fichier à l'octet relatif 12, le mode du mot de passe (Bit 7 = mode

lecture, Bit 6 = mode écriture, Bit 5 = mode effacement) et, dans les octets 24 à 31, les paramètres date et heure de création, d'accès et de mise à jour.
CE : C = 66H, DE pointe sur le FCB du fichier.
CS : A contient 0 si l'opération réussit et FFH sinon. H contient un code d'erreur physique éventuel.

Fonction 103 : Ecriture d'un XFCB (FCB étendu)

Action : cette fonction crée un nouveau XFCB ou met à jour un XFCB existant.
CE : C = 67H, DE pointe sur le FCB du fichier.
CS : A contient 0 si l'opération réussit ou FFH sinon.

Fonction 104 : Positionnement de la date et de l'heure

Action : cette fonction positionne la date et l'heure interne du système. DE pointe sur un tampon de 4 octets qui a la structure suivante :
Octets 0 & 1 : champ date.
Octet 2 : champ heure.
Octet 3 : champ minute.
La date est représentée par un entier sur 16 bits. La valeur 1 correspond au 1 janvier 1978. L'heure et les minutes sont mémorisées de façon classique (l'octet contient la valeur).
CE : C = 68H, DE pointe sur le tampon de 4 octets.
CS : pas de CS.

Fonction 105 : Lecture de la date et de l'heure

Action : cette fonction lit la valeur courante de la date et de l'heure et retourne le résultat dans un tampon de 4 octets pointé par DE. Le format est décrit à la fonction 104.
CE : C = 69H, DE pointe sur un tampon de 4 octets.
CS : le tampon contient les valeurs lues et A contient le nombre de secondes.

Fonction 106 : Positionnement du mot de passe par défaut

Action : cette fonction positionne le mot de passe éventuel d'un fichier avant son succès.
CE : C = 6AH, DE pointe sur le mot de passe (8 octets).
CS : pas de CS.

Fonction 107 : Lecture du numéro de série du CP/M

Action : cette fonction fournit le numéro de série de votre disquette dans un tampon pointé par DE.
CE : C = 6BH, DE pointe sur le tampon de réception.
CS : le tampon contient le numéro de série.

Fonction 108 : Positionne ou lit un code de retour

Action : cette fonction arme ou lit le code de retour qui est fourni par certains programmes.
CE : C = 6CH, DE = FFFFH si le code doit être lu. Sinon, DE contient la valeur à écrire.
CS : si DE vaut FFFFH en entrée, en sortie HL contient le code lu.

Fonction 109 : Positionne ou lit le mode CONSOLE

Action : cette fonction positionne ou interroge le mode de la CONSOLE. Le mode CONSOLE est un mot de 16 bits avec les conventions suivantes :
Bit 0 = 1 - Etat du CTRL C pour la fonction 11.
0 - Etat normal pour la fonction 11.
Bit 1 = 1 - Empêche le fonctionnement de CTRL S et CTRL Q.
0 - Autorise le fonctionnement de CTRL S et CTRL Q.
Bit 2 = 1 - Empêche les expansions de TAB, le CTRL P.
0 - Autorise TAB et CTRL P.
Bit 3 = 1 - Empêche l'action du CTRL C.
0 - Autorise l'action du CTRL C.
Bits 8-9 - Déterminent comment les RSX répondent aux demandes de la console.
CE : C = 6DH, DE = FFFFH si le mode doit être lu. Sinon, DE contient le mode à écrire.
CS : si DE contient FFFFH, alors HL contient le mode lu.

Fonction 110 : Positionne ou lit le délimiteur de sortie

Action : cette fonction permet de déterminer ou de lire le caractère qui sert de délimiteur de sortie. Le délimiteur par défaut est le caractère $.
CE : C = 6EH, DE = FFFFH si le caractère doit être lu. Sinon E contient le caractère à positionner.
CS : si DE = FFFFH, alors A contient le caractère lu.

Fonction 111 : Imprime un bloc

Action : cette fonction envoie la chaîne de caractères définie par le CCB (Character Control Bloc) vers la console.
Le format du CCB est le suivant :
Octets 0 et 1 : adresse de la chaîne de caractères.
Octets 2 et 3 : longueur de la chaîne de caractères.
CE : C = 6FH, DE pointe sur le CCB.
CS : pas de CS.

Fonction 112 : Liste d'un bloc

Action : cette fonction est identique à la précédente, mais la chaîne est envoyée à l'imprimante.
CE : C = 70H, DE pointe sur le CCB.
CS : pas de CS.

Fonction 152 : Construction d'un nom de fichier

Action : cette fonction de construire un FCB en partant d'une chaîne de caractères ASCII contenant le nom du fichier.
CE : C = 98H, DE pointe sur une zone dont le premier mot est l'adresse de la chaîne de caractères et le second l'adresse du FCB.
CS : HL contient un code de retour.

INSTRUCTIONS RESIDENTES ET UTILITAIRES DE DIGITAL RESEARCH

Le CP/M 3.0 est beaucoup plus riche en utilitaires officiels de Digital Research que le CP/M 2.2. Les instructions résidentes sont au nombre de six : **DIR**, **DIRSYS**, **ERASE**, **RENAME**, **TYPE** et **USER**. Cependant, certaines options des commandes DIR, ERASE, RENAME et TYPE nécessitent la présence d'un utilitaire avec l'extension .COM. Ces commandes sont donc en partie résidentes et en partie externes (TRANSIENT). La commande **SAVE** qui était résidente dans la version 2.2 est devenue un utilitaire externe.

Pour plus de facilité, les commandes résidentes et externes ont été mélangées. Elles sont présentées par ordre alphabétique.

Conventions

nfc représente un nom de fichier en clair avec ou sans extension.

nfa représente un nom de fichier ambigu (carte de sélection).

<....> représente un élément optionnel.

[...] représente un élément qui doit être introduit entre crochets. En général, les options seront introduites entre crochets.

n représente un nombre entier.

RO représente READ ONLY (écriture seule).

RW représente READ WRITE (écriture et lecture).

S représente une chaîne de caractères.

SYS représente l'attribut d'invisibilité (SYSTEM).

U: représente un nom d'unité (en général sur AMSTRAD A ou B).

COPYSYS.COM Syntaxe : COPYSYS

Cette commande de génération du système a été retirée de la disquette par AMSTRAD. Vous devez utiliser l'utilitaire propre à AMSTRAD DISCKIT3.COM en lieu et place.

DATE.COM Syntaxe : DATE <C>
Syntaxe : DATE SET ou DATE <spécification>

Le *premier format* de la commande (DATE seule) affiche la date et l'heure courante. L'option C permet un affichage continu de la date et de l'heure (pour sortir, taper une touche quelconque).

Le *second format* permet d'introduire la date et l'heure, soit par une méthode interrogative (SET), soit directement sous la forme MM/JJ/AA HH:MM:SS.

DEVICE.COM

Syntaxe : DEVICE <NAMES/VALUES>
Syntaxe : DEVICE PERPHY <option>
Syntaxe : DEVICE PERLOG <=NULL>
Syntaxe : DEVICE PERLOG=PERPHY <option>
Syntaxe : DEVICE CONSOLE [PAGE/COL/LINES]

Le *premier format* affiche tous les renseignements sur les périphériques logiques et physiques. L'option NAMES limite l'affichage aux périphériques physiques. l'option VALUES limite l'affichage aux périphériques logiques courants.

Le *deuxième format* affiche les caractéristiques du périphérique physique indiqué. L'option permet d'assigner des valeurs au périphérique physique indiqué.
Les options possibles sont XON (protocole XON/XOFF), NOXON (pas de protocole XON/XOFF) et la vitesse en bauds éventuelle.

Le *troisième format* affiche les renseignements sur le périphérique logique indiqué. Si l'option =NULL est utilisée, le périphérique logique indiqué est déconnecté de tous les périphériques physiques.

Le *quatrième format* permet d'assigner un ou plusieurs périphériques physiques à un périphérique logique (rappel l'I/O byte est multiple). Les options sont identiques à celles du second format.

Le *cinquième format* permet d'afficher le format de la CONSOLE (PAGE) ou de modifier ce format en indiquant le nombre de colonnes et le nombre de lignes.
Exemple : DEVICE CONSOLE [COLUMNS=40, LINES=24].

DIR & DIR.COM

Syntaxe : DIR <U:><nfa>
Syntaxe : DIR <U:><nfa> [option]

Le *premier format* représente la partie résidente de la commande. C'est le format classique déjà décrit dans le chapitre réservé au CP/M 2.2.

Le *second format* représente la partie externe de la commande. Il permet l'affichage du répertoire sous de nombreuses formes en fonction de l'option utilisée.

Options possibles

ATT	Affiche les attributs (RO, RW, SYS,...).
DATE	Affiche les dates et heures de création si cette option a été initialisée.
DIR	Affiche les fichiers qui ont l'attribut DIR.
DRIVE=ALL	Affiche le répertoire de tous les disques.
DRIVE=(A,B,..)	Affiche le répertoire des disques spécifiés.
DRIVE=U	Affiche le répertoire du disque spécifié.

EXCLUDE	Affiche les fichiers qui ne correspondent pas au nfa spécifié.
FF	Envoi d'un saut de page avant l'impression si CTRL P a été utilisé.
FULL	Donne tous les renseignements du répertoire.
LENGHT=n	Indique le nombre de lignes à afficher entre chaque passage à la page suivante.
MESSAGE	Affiche le nom des disques et des USERS en cours de traitement.
NOPAGE	Défilement continu du répertoire.
NOSORT	Pas de tri alphabétique avant l'affichage.
RO	Affiche uniquement les fichiers READ ONLY.
RW	Affiche uniquement les fichiers READ WRITE.
SIZE	Affiche le nom et la taille des fichiers.
SYS	Affiche uniquement les fichiers SYSTEM.
USER=ALL	Affiche le répertoire de tous les USERS.
USER=n	Affiche le répertoire de l'USER n.
USER=(0,1...)	Affiche le répertoire des USERS spécifiés.

DIRSYS Syntaxe : DIRSYS <U:><nfa> ou DIRS <U:><nfa>

Cette commande se comporte comme la commande DIR du CP/M 2.2, mais n'affiche que les fichiers qui ont l'attribut SYSTEM.

DUMP.COM Syntaxe : DUMP <U:>nfc

Cette commande identique à la commande du CP/M 2.2 affiche le contenu d'un fichier en hexadécimal et en ASCII.

ED.COM Syntaxe : ED <U:>nfc1 <U:><nfc2>

Cette commande active l'éditeur standard du CP/M. Les commandes sont identiques à celles du CP/M 2.2.

En plus, les commandes suivantes existent :

0A	ajoute des lignes jusqu'au demi-remplissage du tampon.
#A	ajoute des lignes jusqu'au remplissage du tampon.
nX	écrit ou ajoute des lignes à l'article X$$$$$$$.LIB.

ERASE.COM Syntaxe : ERASE <U:>nfa <[CONFIRM]>
Syntaxe : ERA <U:>nfa <[CONFIRM]>

Cette commande efface le ou les fichiers qui correspondent au nom de fichier ambigu spécifié. L'option [CONFIRM] permet d'obliger l'utilisateur à valider son choix par une réponse Y ou N (YES ou NO).

GENCOM.COM Syntaxe : GENCOM <nfichier.COM> <nfic-RSX>
<[LOADER/NULL/SCB=(offset,valeur])

Cette commande permet la création de fichier COM attaché à un fichier RSX *(voir première partie de ce chapitre)*. Elle permet aussi de détacher le RSX d'un fichier COM.

L'option **LOADER** positionne un sémaphore qui permet au programme chargeur de rester actif (cette option est valable uniquement s'il n'y a pas de RSX attaché) *(voir fonction 59 du BDOS)*.

L'option **NULL** indique que seuls les fichiers RSX sont spécifiés. GENCOM crée alors un fichier fictif .COM.

L'option **SCB=(offset,valeur)** positionne un octet du BLOC de CONTROLE SYSTEME à la valeur spécifiée en hexadécimal *(voir fonction 49 du BDOS)*.

Exemples

GENCOM TOTO DEM1 DEM2	génère un fichier TOTO.COM avec les fichiers RSX DEM1 et DEM2 attachés.
GENCOM DEM1 DEM2 [NULL]	génère un fichier DEM1.COM avec les fichiers RSX DEM1 et DEM2 attachés.
GENCOM TOTO	détache les RSX qui étaient attachés à TOTO.COM.
GENCOM FICFIC [LOADER]	attache un enregistrement d'en-tête au fichier FICFIC.COM et positionne le sémaphore du chargeur.
GENCOM FICFIC [SCB=(4,2)]	oblige le chargeur à positionner l'octet relatif 4 au BLOC de CONTROLE SYSTEME à la valeur 2.

GET.COM Syntaxe : GET FILE nfc <[ECHO/NO ECHO/SYSTEM]>
GET CONSOLE

Cette commande indique au CP/M 3.0 qu'il doit prendre ses entrées hors du fichier spécifié. Le fichier spécifié peut contenir des commandes CP/M ou des commandes propres à un programme. La deuxième syntaxe (GET CONSOLE) rétablit le fonctionnement normal du CP/M.

L'option **ECHO** est une option par défaut qui indique que toutes les entrées lues dans le fichier doivent être affichées sur la CONSOLE.

L'option **NO ECHO** inhibe l'affichage des entrées.

L'option **SYSTEM** indique le lancement immédiat des lectures dans le fichier spécifié. En son absence, la lecture des entrées commencera à la suite de la commande suivante.

HELP.COM

Syntaxe : HELP motclé sousmotclé ....<[NOPAGE]>
Syntaxe : HELP [EXTRACT]
Syntaxe : HELP [CREATE]

La commande **HELP** fournit des informations sommaires sur toutes les commandes du CP/M 3.0.

HELP, utilisé sans mot-clé, fournit une liste des mots-clés utilisables. Vous pouvez abréger les mots-clés, en général deux à trois lettres suffisent.

L'option **NOPAGE** permet un défilement continu de l'écran.

La commande HELP est fournie avec deux fichiers HELP.HLP et HELP.COM. Le fichier HELP.HLP contient le texte affiché par la commande. Vous pouvez convertir le fichier HELP.HLP en fichier HELP.DAT grâce à la commande HELP [EXTRACT]. Après modification du fichier HELP.DAT, vous pouvez reconstruire le fichier HELP.HLP au moyen de la commande HELP [CREATE]. Cette option vous permet la création de votre propre fichier d'aide.

HEXCOM.COM

Syntaxe : HEXCOM nfc

Cette commande génère un fichier .COM à l'aide d'un fichier .HEX. Elle est identique à la commande LOAD du CP/M 2.2

INITDIR.COM

Syntaxe : INITDIR U:

Cette commande permet d'initialiser un répertoire (DIRECTORY) pour le traitement des dates et des heures. La seule spécification nécessaire est le nom d'unité.

Cette commande ne fonctionne que si l'espace disponible dans le répertoire est suffisant pour la mémorisation des dates et heures.

LIB.COM

Syntaxe : LIB nfc <[I/M/P/D]>
Syntaxe : LIB nfc <[I/M/P]>=nfc, nfc,...

La commande **LIB** permet de créer des librairies ou d'ajouter, remplacer, sélectionner ou effacer des modules d'une librairie existante. Les modules sont produits par l'utilitaire RMAC ou par n'importe quel langage qui produit des modules au format .REL de Microsoft.

L'option I, comme Index, permet de créer une librairie indexée. La librairie indexée est plus rapide qu'une librairie normale.

L'option M produit l'affichage des noms des modules.

L'option P produit l'affichage des noms des modules et des variables publiques.

L'option D affiche le contenu de la librairie en ASCII.

Exemples

LIB ESSAI[P] affiche tous les modules et toutes les variables publiques contenues dans ESSAI.REL.

LIB ESSAI=PROG1,PROG2 crée la librairie ESSAI.REL en composant les deux modules PROG1.REL et PROG2.REL.

LIB **ESSAI=PROG1(MOD1,MOD3),PROG2(MOD1-MOD4,MOD7)** crée la librairie ESSAI.REL qui sera composée des modules MOD1 et MOD3 de la librairie PROG1.REL ainsi que des modules MOD1 à MOD4 et MOD7 de la librairie PROG2.REL.

LIB **ESSAI=PROG1<MOD1=>** crée la librairie ESSAI.REL en utilisant tous les modules de PROG1.REL sauf MOD1.

LINK.COM Syntaxe : LINK U:<nfc,<[opt]>=>nfc<[opt]>.,...>

La commande **LINK** permet de transformer un ou plusieurs modules "RELOCATABLE" (indépendant de l'adresse) en un fichier au format .COM (exécutable).

Les options possibles contenues entre crochets sont les suivantes :

A	Mémoire supplémentaire. Ecriture temporaire sur disque et réduction des tampons de travail.
B	Lien BIOS pour CP/M avec BANKING (128 K).
Dadr	Adresse d'origine pour la zone des données.
Géti	Positionne l'adresse de départ à l'étiquette éti.
Ladr	Positionne l'adresse de chargement à l'adresse adr. Par défaut, cette adresse vaut 0100H.
Mtaill	Définit la taille mémoire.
NL	Pas de listing de la table des symboles.
NR	Pas de génération de fichier contenant la table des symboles.
OC	Génération d'un fichier COM (option par défaut).
OP	Génération d'un fichier PRL (pour MP/M).
OR	Génération d'un fichier RSP (pour MP/M).
OS	Génération d'un fichier SPR (pour MP/M).
Padr	Positionnement de l'origine du programme à l'adresse adr. Par défaut, cette adresse vaut 0100H.
Q	Liste des symboles avec un point d'interrogation.
S	Recherche dans le fichier précédent en le considérant comme une librairie.
$Cd	Destination des messages. d vaut X pour la CONSOLE, Y pour l'IMPRIMANTE et Z pour inhiber la sortie.
$Id	Disque source des fichiers intermédiaires. d vaut le nom du disque (A-P).
$Ld	Disque source des librairies.

$Od Disque de destination du fichier objet. d vaut le nom du disque (A-P) ou Z s'il ne faut pas générer de fichier source.

$Sd Disque de destination du fichier des symboles. d vaut le nom du disque (A-P) ou Z (pas de génération) ou Y (génération sur la CONSOLE).

MAC.COM Syntaxe : MAC nfc <$options>

Cette commande produit le MACRO-ASSEMBLAGE d'un fichier source au format ASM et produit en sortie trois fichiers. Le premier au **format hexadécimal** (.HEX), le deuxième au **format listing** (.PRN) et le troisième au **format table de symbole** (.SYM).

Les options permettent la sélection des unités avec les conventions suivantes : la lettre d peut être : X pour diriger la sortie sur la CONSOLE, P pour diriger la sortie sur l'imprimante, Z pour inhiber la création du fichier et A-O pour indiquer le nom du disque sélectionné.

Options

Ad Unité de disque source du fichier ASM (A-O).
Hd Unité de destination du fichier HEX (A-O ou Z).
Ld Unité qui contient la librairie (A-O).
Pd Unité de destination du fichier PRN (A-O, X, P ou Z).
Sd Unité de destination du fichier SYM (A-O, X, P ou Z).

+L Liste les lignes lues dans la librairie.
-L Inhibe le listing des lignes lues dans la librairie.
+M Liste les MACROS.
-M Inhibe le listing des MACROS.
***M** Liste uniquement le code hexadécimal produit par les MACROS.
+Q Liste tous les symboles locaux.
-Q Supprime l'édition des symboles locaux.
+S Ajoute le fichier des symboles au fichier PRN.
-S Supprime la création du fichier des symboles.
+1 Produit un listing de la première passe pour la mise au point des MACROS.
-1 Inhibe la production du listing défini ci-dessus.

PATCH.COM Syntaxe : PATCH nfc <n>

La commande **PATCH** affiche ou installe le PATCH numéro n. Ce patch peut porter sur une commande ou sur le système (SYS).

n doit être compris entre 1 et 32.

INSTRUCTIONS RESIDENTES ET UTILITAIRES DE DIGITAL RESEARCH

PIP.COM Syntaxe : PIP 'ligne de commande'

La commande **PIP** (Peripheral Interchange Program) permet l'échange entre les différents périphériques. Vous trouverez une description complète de cette commande *au chapitre III, page 73.*

Les seules modifications propres au CP/M 3.0 portent sur les noms des périphériques logiques et sur les options possibles.

Les périphériques de destination acceptés sont :
CON, AUX, PRN et **LST.**

Les périphériques de source acceptés sont :
AUX, CON, NUL et **EOF.**

Options

A Copie uniquement les fichiers modifiés depuis la dernière copie.
C Demande une confirmation avant chaque copie.
Dn Efface les caractères situés après la colonne n.
E Echo des caractères échangés vers la CONSOLE.
F Ote tous les caractères de saut de page.
Gn Lecture du fichier depuis l'USER n.
H Transfert de fichier HEX.
I Ignore les 00 dans un fichier HEX.
L Transformation automatique en minuscule.
N Ajout d'un numéro de ligne.
O Transfert de fichier OBJET (code machine).
Pn Positionne la taille de la page (n lignes).
Qs Arrêt de copie lors de la rencontre de la chaîne s.
R Lecture de fichiers SYS.
Ss Début de la copie à partir de la chaîne s.
Tn Les tabulations se font sur n colonnes.
U Transformation automatique en majuscule.
V Vérification de la copie après écriture.
W Ecriture forcée des fichiers même en READ ONLY.
Z Mise à zéro du bit de parité.

PUT.COM

Syntaxe : PUT CONSOLE <OUTPUT TO> FILE nfc <opt>
Syntaxe : PUT PRINTER <OUTPUT TO> FILE nfc <opt>
Syntaxe : PUT CONSOLE <OUTPUT TO> CONSOLE
Syntaxe : PUT CONSOLE <OUTPUT TO> PRINTER

Les deux premières formes de la commande **PUT** permettent de diriger les sorties de la CONSOLE et de l'imprimante vers un fichier. les deux dernières formes rétablissent le fonctionnement normal du système.

Options

ECHO Echo des caractères vers la CONSOLE (option par défaut).
NO ECHO Pas d'écho.
FILTER Transformation des caractères non imprimables en caractères imprimables (le BELL devient G).
NO FILTER Pas de transformation (option par défaut).
SYSTEM Les messages produits par le système sont aussi dirigés vers le fichier spécifié.

RENAME.COM Syntaxe : RENAME nfa1=nfa2

Cette commande modifie le nom du ou des fichiers correspondant à nfa2 et le transforme en nom correspondant à nfa1.

La principale amélioration apportée par CP/M 3.0 est la possibilité d'indiquer des noms de fichiers ambigus.

RMAC.COM Syntaxe : RMAC nfc <$Rd/$Sd/$Pd>

Cette commande permet d'assembler et de créer des fichiers dits RELOCATABLE (indépendants de l'adresse mémoire) du type REL à partir de fichier ASM.

Elle produit trois fichiers (REL,PRN et SYM).

Pour des compléments d'information, reportez-vous à la commande MAC.COM.

R indique le disque pour le fichier **REL** (A-O ou Z).
S indique le disque pour le fichier **SYM** (A-O, X, P ou Z).
P indique le disque pour le fichier **PRN** (A-O, X, P ou Z).

SAVE.COM Syntaxe : SAVE

Cette commande doit être lancée avant l'utilisation de l'utilitaire de mise au point (SID) et permet, à l'issue de celui-ci ou d'une autre commande, de sauver la mémoire comprise entre les adresses que vous devrez spécifier sous un nom de fichier.

SET.COM Syntaxe : SET [options]
Syntaxe : SET U: [options]
Syntaxe : SET nfc [options]

Cette commande permet de positionner une série de paramètres (date, mot de passe, READ ONLY, SYSTEME,...).

Syntaxe 1 : positionnement de paramètres pour le disque courant.

Options possibles

PASWWORD=mot	Assigne le mot de passe "mot" au disque.
PASSWORD=CR	Efface le mot de passe (CR=RETOUR CHARIOT).
PROTECT=ON	Active la protection par mot de passe.
PROTECT=OFF	Inhibe la protection par mot de passe.
DEFAULT=mot	Positionne un mot de passe par défaut.
CREATE=ON	Active l'inscription de l'heure et de la date à la création d'un fichier.
ACCESS=ON	Inhibe la commande précédente (CREATE).
UPDATE=ON	Active l'inscription de l'heure et de la date à chaque modification d'un fichier.

Syntaxe 2 : positionnement des attributs du disque spécifié.

Options possibles

RO	Positionnement en lecture seule.
RW	Positionnement en lecture/écriture.
NAME=nom	Donne le nom "nom" au disque.

Syntaxe 3 : positionnement des attributs d'un fichier.

Options possibles

DIR	Positionne le fichier en DIR (inverse SYS).
SYS	Positionne le fichier en SYS (invisible).
RO	Positionne le fichier en READ ONLY.
RW	Positionne le fichier en READ WRITE.
ARCHIVE=OFF	Positionne l'attribut d'archivage à OFF *(voir option A de PIP)*.
ARCHIVE=ON	Positionne l'attribut d'archivage à ON.
F1=ON/OFF	Positionne l'indicateur utilisateur 1.
F2=ON/OFF	Positionne l'indicateur utilisateur 2.
F3=ON/OFF	Positionne l'indicateur utilisateur 3.
F4=ON/OFF	Positionne l'indicateur utilisateur 4.
PASSWORD=mot	Positionne le mot de passe mot.
PROTECT=READ	Le mot de passe est requis pour lire, copier, écrire, effacer ou changer le nom du fichier.
PROTECT=WRITE	Le mot de passe est requis pour écrire, effacer ou changer le nom du fichier.
PROTECT=DELETE	Le mot de passe est requis pour effacer ou changer le nom du fichier.
PROTECT=NONE	Le mot de passe n'est pas actif.

SETDEF.COM

Syntaxe : SETDEF [options]
Syntaxe : SETDEF <U:,U:..> <[commandes]>

La commande **SETDEF** permet d'afficher ou de définir l'ordre d'activation des unités disques lors d'une recherche.

La commande SETDEF permet aussi de sélectionner ou d'inhiber le mode DISPLAY ou PAGE.

Syntaxe 1 : sélection du mode DISPLAY et du mode PAGE. Le mode DISPLAY permet l'affichage du disque, du fichier et du numéro d'USER avant le chargement ou l'exécution d'un programme. Le mode PAGE permet l'arrêt après l'affichage d'un écran complet quel que soit l'utilitaire employé.

Options

Pas d'option	Affichage des paramètres d'ordre de recherche.
DISPLAY	Activation du mode DISPLAY.
NO DISPLAY	Inhibition du mode DISPLAY.
PAGE	Activation du mode PAGE.
NO PAGE	Inhibition du mode PAGE.

Syntaxe 2 : elle permet la définition de l'ordre de recherche pour les disques et les types de fichiers.

Options

U1:,U2:,..	Définit l'ordre des disques à scruter lors d'une recherche. Exemple : SETDEF B:,A: commence la recherche sur l'unité B puis continue sur l'unité A.
TEMPORARY=U:	Définit le nom d'unité (U) à utiliser pour les fichiers temporaires.
ORDER=(T,T)	Définit l'ordre de sélection entre les fichiers SUB et COM (T représente COM ou SUB). Exemple : SETDEF [ORDER=(SUB,COM)].

SHOW.COM Syntaxe : SHOW <U:> <[options]>

La commande **SHOW** permet d'afficher divers paramètres concernant l'unité disque spécifiée.

SHOW, utilisé seul, affiche l'espace disponible sur toutes les unités.

Options

SPACE	Affiche le nom d'unité, le type (RO ou RW) et l'espace disponible.
LABEL	Affiche le nom du disque, l'état du mot de passe, l'état de la mémorisation de la date et de l'heure, ainsi que la date et l'heure éventuelles de création et de mise à jour.
USERS	Affiche le numéro de l'USER actif ainsi que le nombre de fichiers existants pour chacun des USERS déclarés.

INSTRUCTIONS RESIDENTES ET UTILITAIRES DE DIGITAL RESEARCH

DIR Affiche le nombre d'entrées libres dans le répertoire.

DRIVE Affiche les paramètres du disque spécifié (identique à la commande STAT DSK: du CP/M 2.2).

SID.COM Syntaxe : SID <nfc><,nfc SYM>

La commande **SID** active le programme de mise au point (Symbolic Instruction Debugger). Il remplace le DDT de la version 2.2.

SID utilise des constantes hexadécimales. Une valeur décimale doit être précédée du signe # .

Malheureusement, ce "DEBUGGER" fonctionne avec les mnémoniques du 8080. Pour le Z80, il existe une version spéciale appelée ZSID qui n'est pas fournie par AMSTRAD.

Commandes

Aadr Permet l'introduction directe de mnémonique assembleur avec écriture directe du code objet généré à l'adresse adr.

Cadr<b,d> Exécute un CALL à l'adresse ADR. b et d sont deux valeurs optionnelles qui permettent de charger respectivement BC et HL avant l'appel.

D<W><adp><,afn> Affiche la mémoire en hexa et en ASCII à partir de l'adresse adp et jusqu'à l'adresse afn. W indique que le format doit être de 16 bits.

Enfc<,nfcsym> Charge le programme nfc spécifié et éventuellement le fichier SYM spécifié.

E* nfcsym Charge le fichier symbolique (SYM) spécifié.

F,adp,afn,vl Remplit la zone mémoire qui va de adp à afn avec la valeur vl.

G<adr><,a<,b>> Lance le programme à l'adresse adr. a et b sont les adresses optionnelles de point d'arrêt.

Hvl Affiche la valeur du symbole vl en hexa, en décimal et en ASCII.

H.vl Affiche la conversion en nombre et en ASCII de la valeur vl.

H.v1,v2 Affiche la valeur de v1+v2 et de v1-v2.

Icommande Entre une ligne de commande CCP.

Ladp,afn Désassemble le contenu de la mémoire comprise entre les adresses adp et afn.

Madp,afn,aar Déplace le bloc mémoire compris entre adp et afn vers l'adresse aar.

P<p<,c>> Positionne, enlève ou affiche le nombre de passes d'un point d'arrêt permanent. p représente l'adresse du point d'arrêt, c représente le nombre de passes avant l'arrêt.

Rnfc<,d> Lecture d'un fichier, d indique un OFFSET de chargement éventuel.

S<W>adr	Modifie la valeur mémoire située à l'adresse adr puis passe à la suivante (CR ne modifie pas la valeur courante). W indique que la saisie se fait sous la forme de mots de 16 bits.
T<W><n<,c>>	Trace l'exécution d'un programme. n est le nombre de pas entre chaque arrêt et c est le point d'entrée éventuel. W permet de tracer l'exécution d'un programme sans exécuter les sous-routines éventuelles.
U<W><n<,c>>	Exécution sans trace.
V	Affiche la valeur du prochain octet disponible, la valeur du PC, l'adresse de fin de mémoire disponible...
Wnfc<,adp,afn>	Ecrit le contenu de la mémoire comprise entre adp et afn dans le fichier nfc.
X<f><r>	Examine et modifie l'état du registre ou du sémaphore spécifié.

SUBMIT.COM Syntaxe : SUBMIT nfc.SUB <arg> <arg> ....

La commande **SUBMIT** permet l'exécution d'un fichier BATCH contenant des commandes. Cette commande regroupe les fonctionnalités des commandes SUBMIT et XSUB du CP/M 2.2. Pour plus de détails, reportez-vous à ces deux commandes *au chapitre III*.

TYPE Syntaxe : TYPE nfc <[PAGE/NO PAGE]>

Cette commande affiche le contenu du fichier spécifié en ASCII à la CONSOLE. l'option PAGE ou NO PAGE arrête ou n'arrête pas l'édition en bas de page.

USER Syntaxe : USER n

Cette commande sélectionne le numéro d'USER. n doit être compris entre 0 et 15.

XREF.COM Syntaxe : XREF U: nfc <$P>

Cette commande génère la table des références croisées du programme spécifié. Elle nécessite la présence des fichiers PRN et SYM du programme. L'option $P dirige l'édition vers l'imprimante.

UTILITAIRES PROPRES A AMSTRAD

En plus des utilitaires de Digital Research, l'AMSTRAD contient des utilitaires qui lui sont propres et ne se retrouvent pas sur d'autres systèmes. Ces utilitaires sont en général construits pour tirer un meilleur profit de la structure matérielle propre au CPC.

AMSDOS.COM Syntaxe : AMSDOS

Cet utilitaire permet le retour au système d'exploitation AMSDOS.

DISCKIT3.COM Syntaxe : DISCKIT3

Cet utilitaire regroupe les fonctions de copie, de formatage et de vérification des disquettes. Il détecte automatiquement le nombre de lecteurs de disque présents dans la configuration et adapte les menus en conséquence. Les réponses doivent être fournies à l'aide du clavier numérique. Cet utilitaire est semblable au DISCKIT2 du CP/M 2.2 ; cependant, il utilise la totalité de la mémoire (128 K) et nécessite moins d'échange de disquettes lors de la copie totale sur un seul disque.

LANGUAGE.COM Syntaxe : LANGUAGE n

Cet utilitaire permet de reprogrammer la fonte de certains caractères pour les adapter à une langue précise. Le paramètre n est un entier compris entre 0 et 7.

La table des caractères s'établit comme suit :

n	*Langue*	*23H*	*40H*	*5BH*	*5CH*	*5DH*	*5EH*	*60H*	*7BH*	*7CH*	*7DH*	*7EH*
0	E-U	#	@	[	\	]	↑	`	{	¦	}	~
1	FRANCE	#	à	°	ç	§	↑	`	é	ù	è	¨
2	ALLEMAGNE	#	§	Ä	Ö	Ü	↑	`	ä	ö	ü	B
3	G-B	£	@	[	\	]	↑	`	{	¦	}	~
4	DANEMARK	#	@	Æ	Ø	Å	↑	`	æ	ø	å	~
5	SUEDE	#	É	Ä	Ö	Å	Ü	é	ä	ö	a	ü
6	ITALIE	#	@	°	\	é	↑	ù	à	ò	è	ì
7	ESPAGNE	Pt	@	¡	Ñ	¿	↑	`	¨	ñ	}	~

PALETTE.COM Syntaxe : PALETTE n,m

Cet utilitaire permet de modifier la couleur du fond et de l'encre d'écriture. Le paramètre n définit la couleur du fond, le paramètre m définit la couleur de l'encre. Ils doivent être compris entre 0 et 63. Plusieurs valeurs produisent une couleur identique. Vous trouverez, dans la *table ci-contre*, les valeurs typiques pour chacune des 27 couleurs.

Couleur	*Valeur*	*Couleur*	*Valeur*
NOIR	0	BLEU	1
BLEU VIF	3	ROUGE	8
MAGENTA	10	MAUVE	11
ROUGE VIF	12	VIOLET	14
MAGENTA VIF	15	VERT	32
TURQUOISE	34	BLEU CIEL	35
JAUNE	40	BLANC	42
BLEU PASTEL	43	ORANGE	44
ROSE	46	MAGENTA PASTEL	47
VERT VIF	48	VERT MARIN	50
TURQUOISE VIF	51	VERT CITRON	56
VERT PASTEL	58	TURQUOISE PASTEL	59
JAUNE VIF	60	JAUNE PASTEL	62
BLANC VIF	63		

SET24X80.COM Syntaxe : SET24X80 ON/OFF

Cet utilitaire permet de sélectionner le mode d'affichage de 24 lignes et 80 colonnes (ON) ou de 25 lignes et 80 colonnes (OFF). Cet utilitaire ne fonctionne que si la ligne d'état (25° ligne) est inhibée par CTRL [0.

SETKEYS.COM Syntaxe : SETKEYS KEYS.XXX

Cet utilitaire permet la redéfinition du code ASCII généré par certaines touches. En standard, trois fichiers de redéfinition sont fournis. En voici le contenu :

KEYS.CCP Définition standard des touches pour l'interpréteur de commande du CP/M 3.0.

KEYS.DRL Définition des touches pour le LOGO de Digital Research.

KEYS.WP Définition des touches pour le traitement de texte.

Vous pouvez construire vous-même des configurations de clavier à l'aide de l'éditeur (ED.COM).

Le format est très simple, il suffit de faire suivre le code de la touche sur le clavier *(voir annexe - tome I)* de la lettre N, S ou C, suivant que la touche est normale, appuyée conjointement à la touche SHIFT ou conjointement à la touche CONTROL (vous pouvez utiliser conjointement deux ou trois lettres). Ensuite, il suffit de donner le code produit entre guillemets ("). Ce code peut être une lettre ou un ensemble de lettres, une constante décimale comprise entre simples apostrophes, une constante hexadécimale comprise entre simples apostrophes et précédée du signe # ou un code de contrôle (précédé du signe ↑). Vous trouverez, au chapitre réservé aux *trucs et astuces*, un exemple de configuration du clavier en AZERTY.

UTILITAIRES PROPRES A AMSTRAD

SETLST.COM Syntaxe : SETLST nfc

Cet utilitaire est à l'imprimante ce que le précédent est au clavier. Il permet de configurer une imprimante en lui envoyant une séquence particulière de caractères contenue dans le fichier spécifié. Ces caractères sont composés de façon identique à ceux de l'utilitaire SETKEYS.

SETSIO.COM Syntaxe : SETSIO ligne de commande

Cet utilitaire permet de configurer le canal A de l'interface RS232.

La commande utilisée seule permet de visualiser l'état courant de la configuration.

La ligne de commande permet de définir les paramètres du SIO. Chaque paramètre doit être séparé par une virgule.

Paramètres

RX vitesse	Sélection de la vitesse de réception. Vitesse = 75, 110, 300, 600, 1200, 2400, 4800, 9600.
TX vitesse	Sélection de la vitesse d'émission.
PARITY type	Sélection de la parité. type = NONE, EVEN ou ODD.
STOP n	Sélection du nombre de STOP BITS (1 ou 2).
BITS n	Sélection du nombre de BITS (7 ou 8).
HANDSHAKE fl	Gestion des états par le matériel (DTR,...). fl = ON ou OFF.
XOFF fl	Gestion des états par logiciel (XON/XOFF).

L'EMULATEUR DE TERMINAL

Pour permettre une compatibilité des logiciels existants sous CP/M et le CPC en réduisant au minimum la configuration du logiciel, le constructeur a permis à l'AMSTRAD de simuler le comportement d'un terminal HEATHKIT ZENITH de type Z19 ou Z29.

Pour vous permettre une configuration personnelle de certains logiciels, nous vous donnons, ci-dessous, la **table des codes compris et interprétés** par le CPC.

CTRL G Signal sonore.
CTRL H Retour en arrière d'un caractère.
CTRL J Saut de ligne (LINE FEED).
CTRL M Retour chariot.
CTRL [ESCAPE.

L'appui sur la touche ESC produit un escape uniquement si le programme SETKEYS KEYS.CCP a été activé.

ESC 0 Désactive la ligne d'état (25° ligne).
ESC 1 Active la ligne d'état.
ESC 2 n Active le jeu de caractères correspondant à la langue n *(voir LANGUAGE.COM)*.
ESC 3 m Sélectionne le MODE écran (m = **@** pour le mode 0, m = A pour le mode 1 et m = B pour le mode 2).
ESC A Curseur haut.
ESC B Curseur bas.
ESC C Curseur à droite.
ESC D Curseur à gauche.
ESC E Efface la page (FORM FEED).
ESC H Curseur en haut d'écran (HOME).
ESC I Curseur haut avec mode rouleau pour le texte (ROLL).
ESC J Effacement de la position courante du curseur jusqu'à la fin de la page (EOS).
ESC K Effacement de la position courante du curseur jusqu'à la fin de la ligne (EOL).
ESC L Insertion d'une ligne.
ESC M Suppression d'une ligne.
ESC N Suppression d'un caractère.
ESC Ylc Positionnement du curseur en coordonnées absolues. Le premier caractère qui suit l'Y indique le numéro de ligne, le second le numéro de colonne. Ces caractères sont formés en ajoutant 32 (20H) au numéro de ligne ou de colonne à atteindre.
ESC b n Sélection de la couleur du texte. n est défini comme dans l'utilitaire PALETTE.COM.
ESC C n Sélection de la couleur de fond d'écran.
ESC d Effacement d'écran de la position courante du curseur jusqu'au début de l'écran.
ESC e Désactive le curseur.
ESC f Active le curseur.

UTILITAIRES PROPRES A AMSTRAD

ESC j	Sauvegarde la position du curseur.
ESC k	Restitue le curseur à la position sauvegardée par ESC j.
ESC l	Effacement total de la ligne courante.
ESC o	Effacement des caractères depuis la position courante du curseur jusqu'au début de la ligne.
ESC p	Inversion vidéo.
ESC q	Arrêt d'inversion vidéo.
ESC v	Curseur en fin de ligne.
ESC x	Sélection du mode 24 X 80.
ESC y	Sélection du mode 25 X 80.

LES UTILITAIRES GRAPHIQUES

Digital Research permet l'utilisation des graphiques à l'aide d'un logiciel appelé **GSX.** AMSTRAD a adapté ce logiciel aux normes du CPC 6128. Cet utilitaire permet d'utiliser les graphiques en plus du mode texte. Cependant, il ne peut être utilisé que conjointement à d'autres logiciels développés par Digital Research.

Ce gestionnaire est prévu pour l'écran dans les trois modes, pour une imprimante EPSON, pour l'imprimante DMP1 d'AMSTRAD ou pour une table traçante de type 7470 d'HEWLET PACKARD.

Le fichier DRIVERS.GSX contient les différents gestionnaires.

Pour l'installation, un utilitaire GENGRAPH.COM est fourni sur la seconde disquette du système de base.

L'utilisation du GSX est bien sûr fonction du logiciel utilisé et nous vous conseillons de vous reporter au mode d'emploi dudit logiciel pour de plus amples informations.

LANGAGE LOGO

GENERALITES

Le langage **LOGO** est distribué à tout acheteur de système disque. Il fait donc un peu partie du logiciel de BASE. C'est pourquoi nous lui consacrons un chapitre complet.

Le chapitre est divisé en deux parties. La première concerne le LOGO 2, autrement dit, le LOGO distribué avec le CP/M 2.2. La seconde contient les nouvelles primitives apportées par le LOGO 3. Le LOGO 3 est disponible uniquement sur le CPC 6128.

Le langage LOGO est considéré à tort comme un langage d'apprentissage de l'informatique destiné aux jeunes enfants. Cette réputation est due à l'existence du système de déplacement de la TORTUE qui permet une visualisation directe et, par là, une matérialisation complète des principes de base. En vérité, cette particularité intéressante n'est que le sommet d'un iceberg informatique. Elle permet d'entrer sans difficulté dans un langage riche et puissant dont les caractéristiques intrinsèques dépassent de loin la plupart des langages Basic.

Les amateurs de Basic qui refusent d'aborder le LOGO risquent de passer à côté de concepts aussi importants que la RECURRENCE, la définition de PROPRIETE ou les notions de FONCTION, VARIABLES LOCALES et VARIABLES GLOBALES.

Enfin, pour terminer, signalons qu'avec un peu de programmation, le LOGO ouvre la voie aux systèmes d'intelligence artificielle et peut être considéré comme le début d'un petit langage PROLOG.

Remarque : le langage LOGO est en partie intégré dans la seconde partie de la ROM disque.

COMMANDES ET MOTS-CLES (PRIMITIVES) DU LOGO 2

*** / + -** permettent d'effectuer des opérations arithmétiques.

```
?2*3
6
?(+ 2 3 4)
9
```

.APV propriété caractéristique des variables globales.

```
?make "a 9
?gprop "a ".APV
9
?to a
>local "b
>make "b 8
>end
a defined
?a
>gprop "b ".APV
[ ]
```

Voir aussi .DEF.

.contents commande qui a pour effet d'afficher à l'écran le contenu de l'espace de travail LOGO.

.DEF propriété caractéristique des procédures.

Voir .APV.

```
?gprop "a ".DEF
[[] [local "b] [make "b 8]]
```

.deposit

.deposit X Y

permet de déposer la valeur Y à l'emplacement mémoire X. Instruction similaire au POKE du Basic. A utiliser de manière judicieuse !

.examine

.examine Y

fonction permettant d'afficher à l'écran le contenu de la mémoire se trouvant à l'adresse Y. Fonction similaire à la fonction PEEK du Basic.

.PRM spécifie que l'on parle d'une primitive (mot-clé du LOGO).

?glist ".PRM : donnera la liste des primitives LOGO.

```
?gprop "glist ".PRM
5417
```

: ce nombre désigne l'adresse mémoire à laquelle commence la routine correspondant à la primitive spécifiée (glist).

< = >

permettent d'effectuer des opérations logiques. La réponse fournie par LOGO sera TRUE pour vrai et FALSE pour faux.

```
?2<3
TRUE
?2=3
FALSE
```

and

permet d'effectuer des ET logiques.
Si on désire plus de deux opérateurs, il faudra mettre toute l'opération entre parenthèses.

```
?and "TRUE "TRUE
TRUE
?(and "TRUE "TRUE "FALSE)
FALSE
```

ascii

```
ascii "mot
```

donne la valeur ascii de la première lettre du mot spécifié.

```
?ascii "abc
97
```

bf

```
bf paramètre
```

ampute le paramètre précisé de son premier élément.

```
?bf "azer
zer
?bf [qs dfg hj]
dfg hj
?bf bf [az er ty]
ty
```

bk

```
bk X
```

instruction ayant pour effet de faire reculer la tortue de X unités.

bl

```
bl paramètre
```

ampute le paramètre précisé de son dernier élément.

```
?bl "antoine
antoin
bl 12345
1234
```

buttonp

```
buttonp X
```

où X désigne une des deux manettes de jeux (0 et 1). Cette instruction vérifie si le bouton de la manette de jeux spécifiée a été enfoncée et fournit la valeur de vérité correspondante (TRUE ou FALSE).

catch

catch variable procédure

lance la procédure spécifiée jusqu'à la rencontre d'un throw suivi du nom de la variable. Exemple :

```
?to test
>catch "bof [suite]
>pr [coucou]
>end
test defined
?to suite
>if keyp [throw "bof]
>suite
>end
suite defined
?test
- enfoncer une touche -
coucou
```

Dans la procédure test, l'instruction catch passe le contrôle à la procédure suite. Le throw, suivi du nom de la variable de l'envoyeur ("bof) repasse le contrôle à la procédure test. Les deux instructions catch et throw peuvent être comparées aux GOSUB et RETURN du Basic.

char

char X

où X est un nombre. Cette fonction fournit le caractère ascii correspondant au nombre X. Elle permet de communiquer des caractères de contrôle au système. En effet, prenons l'exemple de char 7 qui correspond au caractère de contrôle CTRL G. En LOGO, CTRL G n'active pas le générateur de son, mais stoppe l'exécution d'un programme. Pour produire l'effet approprié des caractères de contrôle, il faudra utiliser la fonction char.

clean

permet d'effacer l'écran graphique sans modifier la position courante de la tortue.

co

permet de continuer l'exécution d'un programme interrompu par [CTRL] Z ou par pause.
Après un [CTRL] Z, il est possible d'interroger la machine sur l'état de la procédure en cours d'exécution avant de continuer.

cos

cos A

donne le cosinus de l'angle A exprimé en degrés.

count

```
count [liste]
count "mot
```

fournit le nombre d'éléments composant le mot ou la liste spécifiée.

```
?count [ben heu bof]
3
?count "ehben
5
```

cs efface l'écran graphique et replace la tortue à sa position de départ (centre de l'écran, tête tournée vers le haut).

ct efface l'écran texte et y replace le curseur en haut à gauche.

dir commande fournissant la liste de tous les fichiers LOGO figurant sur le disque. En fait, cette commande recherche les fichiers dont le nom comporte l'extension CP/M.LOG. Cette extension n'apparaît pas à l'utilisateur. Elle est automatiquement ajoutée par le LOGO lors de la commande save, ce qui lui permettra de retrouver ses fichiers sur le disque lors d'une commande dir.

dot

```
dot [x y]
```

allume le point écran de coordonnées (x,y).

ed

```
ed "nom_de_procédure
```

permet de placer une procédure dans le mode éditeur et d'y effectuer différentes modifications. Dans ce mode, le déplacement est permis grâce aux quatre flèches, l'insertion est automatique, la touche DEL efface le caractère précédant le curseur et la touche CLR efface le caractère sous le curseur.
Pour sortir du mode éditeur :
- en gardant les modifications apportées à la procédure éditée : enfoncer la touche [COPY] ;
- sans conserver ces modifications : enfoncer la touche [ESC].

emptyp

```
emptyp[liste]
emptyp "mot
emptyp nombre
```

fournit la valeur système correspondant au paramètre donné. TRUE indiquera un paramètre vide, FALSE un paramètre non vide.

```
?emptyp "coucou
FALSE
```

```
?emptyp []
TRUE
Un nombre n'est jamais vide :
?emptyp 0
FALSE
```

end constitue la dernière ligne d'une procédure. end signale la fin de la procédure et ramène au signal de sollicitation LOGO (?).

ent

```
ent [NE A B C A' B' C'....]
```

permet d'établir la forme d'une enveloppe de ton à laquelle il sera fait référence lors de la commande de sound. NE représente le numéro d'enveloppe (de 0 à 15), A, le nombre de pas, B, la taille du pas en fréquence et C, le temps de pause. Cinq sections A B C peuvent être décrites.

env

```
env [NE A B C ...]
```

permet d'établir la forme d'une enveloppe de volume à laquelle il sera fait référence lors de la commande sound. NE représente le numéro d'enveloppe (de 0 à 15), A, le nombre de pas, B, la taille en volume du pas et C, le temps consacré à chaque pas. Cinq sections A B C peuvent être décrites.

er

```
er "nom_de_procedure
er [liste de noms de procédures]
```

supprime de l'espace de travail LOGO la ou les procédure(s) désignée(s).

ern

```
ern "nom_de_variable
ern [liste de noms de variables]
```

supprime de l'espace de travail LOGO la (les) variable(s) ainsi désignée(s).

ERRACT est utilisé pour modifier ce qui se passe en cas d'erreur. Normalement, la valeur d'ERRACT est FALSE. Si, par un ?make "ERRACT "TRUE, on la rend TRUE, une pause sera effectuée en cas d'erreur, et il sera possible d'interroger la machine sur l'état de la procédure puis d'effectuer un co (continuer). L'utilisation d'ERRACT conjointement à pause et [CTRL]Z permet la mise au point de programmes complexes.

error

le mot error apparaît lors des deux situations suivantes :
1- il constitue le label d'une commande catch dans une procédure permettant d'en corriger une autre ;
2- en tant qu'instruction distincte, il produira une liste dont les éléments décrivent l'erreur examinée.

Exemple

```
?to bonjour
>p
>pr [coucou]
>end
bonjour defined
?make "ERRACT "TRUE
?bonjour
pausing ... in bonjour : p
bonjour ?co
?to corrige
>catch "error [bonjour]
>op error
>end
corrige defined
?corrige
pausing ... in bonjour : p
bonjour ?co
[35 [I don't know how to p] bonjour [p] catch p]
```

-- op error indique le type d'erreur (35) et catch "error [bonjour] va chercher l'erreur dans la procédure désignée.

FALSE

valeur système = faux

```
?2=5
FALSE
```

fd

```
fd X
```

permet de faire avancer la tortue de X unités.

fence

commande empêchant la tortue de sortir de l'écran graphique. La ligne :
?fence fd 500 produira le message suivant :
turtle out of bounds.

first

```
first [liste]
first nombre
first "mot
```

fournit le premier élément du paramètre précisé.

```
?first "abc
a
```

fput

fput élément1 élément2
ajoute l'élément1 à l'élément2.

```
?fput "ne "nni
nenni
?fput [ne] [nni]
[[ne] nni]
?fput "ne [nni]
[ne nni]
?fput [ne] "nni
fput doesn't like nni as input.
```

On ne peut pas ajouter une liste à un mot.

fs

réserve la totalité du moniteur vidéo à l'écran graphique.

glist

glist "propriété
fournit la liste de tous les objets présents dans l'espace de travail LOGO et possédant la propriété précisée. Ainsi :

```
?glist ".DEF
```

fournira la liste de toutes les procédures définies dans l'espace de travail LOGO.

go

go "nom
utilisé au sein d'une procédure pour passer le contrôle à la ligne portant le label "nom désigné.

```
?to essai
>label "ici
>pr [ceci est un bete programme]
>go "ici
>end
essai defined
```

gprop

gprop "nom "espèce
fournit les propriétés de "nom considéré suivant l'espèce désignée (.APV = variable, .DEF = procédure).

```
?pgrop "a ".DEF
```

fournira le contenu de la procédure a.

```
?gprop "a ".APV
```

fournira le contenu de la variable a.

ht

commande rendant la tortue invisible.

if

if proposition [action 1] [action 2]
si la proposition est vraie, l'action 1 sera effectuée. Si elle est fausse, l'action 2 sera effectuée.

```
?if 4=5 [pr "ca alors] [pr "vous rêvez]
vous rêvez
```

int

int X
fournit la valeur entière de X, c'est-à-dire tout ce qui se trouve devant la virgule.
?int -1,2
-1

item

item X yyy
fournit le Xième élément de yyy. yyy est un mot, une liste ou un nombre.
?item 3 [oui non zut flut]
zut
?item 3 1/2
5
Les opérations arithmétiques sont effectuées avant la fonction item. 1/2 valant 0.5, son troisième élément est donc 5.

keyp

commande qui teste le clavier et fournit une valeur de vérité TRUE si une touche a été enfoncée ou FALSE si aucune touche n'a été enfoncée.
>if keyp = "TRUE [action 1] [action 2]
effectuera l'action 1 si une touche a été enfoncée et l'action 2 si non.

label

permet de disposer une étiquette à un endroit déterminé d'une procédure. *Voir l'instruction go* pour l'exemple.

list

list élément1 élément2
place les deux éléments précisés dans une liste.
?list [ab c] "d
[[ab c] d]

load

load "nom_de_fichier
permet de charger un fichier disque dans l'espace de travail LOGO. *Voir save*.

local

local "variable
permet de rendre le contenu de certaines variables valable uniquement à l'intérieur de la procédure comportant cette instruction.
?to essai
>local "c
>make "c "coucou
>pr :c
>end
essai defined
?make "c "bonjour

```
?essai
coucou
? :c
bonjour
```

lt lt X
fait tourner la tortue de X degrés d'angle vers la gauche.

make make "nom_variable "contenu
permet d'assigner une valeur à une variable.

```
?make "d 1/2
?:d
0.5
```

nodes fournit un nombre exprimant la taille de l'espace de travail LOGO non encore utilisé à l'instant t. Il faut tout d'abord savoir que toute action effectuée consomme de l'espace de travail puisqu'elle y est enregistrée.
Essayez : ?repeat 2000 [pr nodes]
Les nombres affichés seront de plus en plus petits car l'espace de travail se réduit du fait des instructions [pr nodes] effectuées. Lorsque cet espace disponible approchera du chiffre 100, vous constaterez un arrêt. Ensuite, le nombre affiché sera de nouveau élevé. En fait, durant cet arrêt, le LOGO a réorganisé son espace de travail afin d'y faire de la place car il considérait en avoir trop peu.
Voir RECYCLE.

not inverse la valeur de vérité.

```
?not "TRUE
FALSE
```

op sort de la procédure et retourne à la procédure appelante en lui passant la valeur désignée.

paddle paddle X
fournit un nombre décrivant la position de la manette de jeux numéro X (X = 0 ou 1). Si la manette de jeux est en position repos, la fonction paddle fournit 255. Dans les autres cas, elle fournit un nombre compris entre 0 et 7 qui, multiplié par 45°, donne la position de la manette de jeux considérée.

pal pal X
fournit une liste de trois nombres représentant la couleur utilisée par le crayon numéro X. Ces trois

nombres sont compris entre 0 et 2 chacun. Ils représentent les intensités des composantes respectives de rouge, vert et bleu. *Voir l'instruction setpal* qui permet de déterminer la couleur d'encre d'un crayon.

pause utilisé dans une procédure pour en stopper momentanément l'exécution. Un co sera nécessaire pour la redémarrer.

pd abaisse le crayon graphique. La tortue tracera des lignes en se déplaçant. *Voir pu.*

pe remplace la couleur du crayon de la tortue par la même couleur que le fond. Cela permet d'effacer des traits en repassant dessus.

plist

```
plist mot
```

fournit la liste des propriétés attribuées au mot désigné.

```
?make "a 2
?plist "a
[.APV 2]
?to add :suj :rel :obj
>pprop :suj :rel :obj
>pprop :obj word :rel "_est :suj
>end
add defined
?add "anne "maman_de "pierre
?plist "anne
[maman_de pierre]
?plist "pierre
[maman_de_est anne]
```

po

```
po "nom d'une_procédure
po [liste de noms de procédures]
```

affiche à l'écran le contenu de la (des) procédure(s) désignée(s).

pots affiche à l'écran la première ligne (celle qui commence par to) de toutes les procédures présentes dans l'espace de travail LOGO.

pprop

```
pprop "objet "propriété
```

permet d'attribuer une propriété nouvelle à un objet.

```
?pprop "toto ".APV 12
```

a le même effet que :

```
?make "toto 12
```

De même :
?pprop "w ".DEF [[][contenu procédure]]
pourra être utilisé au lieu de :
?to w
>contenu procédure
>end

pr

pr "mot
pr [liste]
affiche à l'écran le mot ou la liste suivi d'un retour chariot. *Voir aussi show et type.*
Des parenthèses seront nécessaires si l'on désire faire suivre pr de plusieurs objets.

```
?pr "a pr "z
a
z
?pr [as]
as
?(pr "a "b "c)
a b c
```

pu relève le crayon de la tortue ; ainsi, ses déplacements ne dessineront plus de trait. Pour rabaisser le crayon, il faudra utiliser pd.

px modifie la couleur du crayon en sa couleur logique inverse. Tout ce qui a été dessiné auparavant changera ainsi de couleur.

random

random X
fournit un nombre aléatoire compris entre 0 et X-1.

rc attend un caractère frappé au clavier et l'affiche. Cette fonction est très utile dans les procédures où l'on désire que l'utilisateur fournisse une réponse.

```
?if rc = "o [pr "oui] [pr "non]
```

recycle cette commande ordonne au LOGO de réorganiser son espace de travail afin d'y libérer de la place. Cette opération porte le nom de "garbage collection" (collecte d'ordures). Elle est effectuée automatiquement lorsque l'espace de travail devient trop petit (100 nodes).

REDEFP variable système contenant FALSE en standard et ayant pour rôle d'interdire la redéfinition des primitives en tant que procédure. Si sa valeur est TRUE, il sera alors possible de redéfinir les primitives.

Cette variable est à utiliser prudemmènt. Lorsqu'une primitive est redéfinie, elle ne fait plus ce pourquoi elle a été prévue.
Sauvez votre espace de travail et essayez :

```
?to fd
fd is a primitive
?make "REDEFP "TRUE
?to fd
>end
fd defined
?fd 100
```

Plus rien ne se passera car, ayant été redéfini, fd n'est désormais plus un mot-clé.

release

release X

libère le canal sonore X en attente.

Pour le canal A	X = 1
Pour le canal B	X = 2
Pour le canal A + B	X = 3
Pour le canal C	X = 4
Pour le canal A + C	X = 5
Pour le canal B + C	X = 6
Pour le canal A + B + C	X = 7

Voir sound.

remprop

remprop "nom "nature

supprime les propriétés de nature spécifiée attribuées au nom désigné.

```
?make "x "coucou
?plist "x
[.APV coucou]
?:x
coucou
?remprop "x ".APV
?plist "x
[]
?:x
x has no value
```

repeat

repeat X [action]

effectue X fois l'action spécifiée.

```
?repeat 4 [fd 50 rt 90]
```

dessinera un carré.

rl

attend que l'utilisateur frappe au clavier une série de caractères suivie d'un retour chariot et la retourne sous forme de liste.

```
?rl
bla bla bla [RC]
[bla bla bla]
```

COMMANDES ET MOTS-CLES (PRIMITIVES) DU LOGO 2

rq attend que l'utilisateur frappe au clavier une série de caractères suivie d'un retour chariot et la retourne sous forme de mot.

```
?rq
coucou bla bla [RC]
coucou bla bla
```

rt rt X

fait tourner la tortue de X degrés d'angle vers la droite.

run run instructions

effectue la liste d'instructions désignée. Revient à écrire repeat 1.

save save "nom_de_fichier

permet de sauvegarder l'espace de travail LOGO sur disque dans le but de le réutiliser par la suite. Un load, suivi du même nom de fichier, permettra de le recharger ultérieurement.

se se élément1 élément2

fournit une liste composée des deux éléments spécifiés. Voyez également list.

```
?se [ab c] "d
[ab c d]
?se [[a]] [b]
[[a] b]
```

seth seth X

oriente la tortue dans la direction indiquée par X, en degrés d'angle (*voir aussi* tf).

```
?seth 90
```

set pal setpal X [A B C]

fournit au crayon numéro X une encre définie par les trois paramètres A, B et C. Ces derniers peuvent prendre trois valeurs comprises entre 0 et 2. Ces valeurs représentent les intensités respectives des composantes de rouge, vert et bleu destinées à définir la couleur désirée (*voir aussi* pal).

```
?setpal 0 [2 2 0]
```

setpc setpc X

détermine le numéro de crayon à utiliser. Quatre crayons numérotés 0, 1, 2 et 3 sont à notre disposition. Le crayon 0 est de la même couleur que le fond. Les couleurs des différents crayons sont déterminées par setpal.

setpos

setpos [X Y]

déplace la tortue vers le point de coordonnées (X, Y) tout en conservant sa direction.
?setpos [55 40]

setsplit

setsplit X

permet de modifier la taille de l'écran texte présent en dessous de l'écran graphique. X peut prendre toutes les valeurs comprises entre 1 et 23. Il représente le nombre de lignes que comportera ce texte :
?ss setsplit 2

sf

fournit une liste composée de cinq éléments décrivant l'arrangement courant de l'écran.

1- n° du crayon dont la couleur détermine la couleur de fond. Vaut toujours 0 en CP/M 2.2.

2- état de l'écran : ts = texte
ss = partage texte/graphique
fs = graphique

3- nombre de lignes consacrées au texte dans l'écran graphique, modifiable par setsplit.

4- état des bords de l'écran. Vaut : window, wrap ou fence.

5- rapport d'écran. Toujours égal à 1 en CP/M 2.2. Modifiable par setscrunch.

show

show paramètre

affiche le paramètre tel qu'il se présente en le faisant suivre par un retour chariot. *Voir aussi* pr et type.
?show [a] show [b]
[a]
[b]

sin

sin X

fournit la valeur du sinus de l'angle X exprimé en degrés.

sound

sound [poids_canal période durée volume enveloppe_de_volume enveloppe_de_ton]

poids_canal : nombre permettant de spécifier le numéro de canal désiré ainsi que son état.
le poids 1 correspond au canal A
le poids 2 correspond au canal B
le poids 4 correspond au canal C

Pour sélectionner plusieurs canaux simultanément, il suffit d'additionner leurs poids.
Si l'on désire que le canal sélectionné soit suspendu (*voir* release), il suffit d'ajouter 64 à son poids. Si l'on désire un rendez-vous avec le canal A, il suffit d'ajouter 8 au poids du canal désigné. Pour un rendez-vous avec le canal B, ajouter 16 et, avec le canal C, ajouter 32. Par *exemple :*

```
?sound [10 100 200]   B a rendez-vous avec A
?sound [17 200 300]   A a rendez-vous avec B
```

Les deux sons sortiront ensemble.
Pour éclater un canal, c'est-à-dire stopper sa production, il faut lui ajouter 128. Par *exemple :*

```
?sound [1 30 100000]  sonne longtemps...
?sound [129]          stoppera cette production.
```

période : nombre compris entre 0 et 4095 et permettant de déterminer le ton. Le LA international correspond à la fréquence 440 Hz. La période correspondant à cette fréquence est 284.

```
?sound [1 0 100]      son très élevé, inaudible.
?sound [1 284 100]    donne le LA.
?sound [1 4095 100]   son très grave.
```

Les paramètres suivants sont facultatifs :

durée : nombre compris entre 0 et 32767 exprimant la durée du son en centièmes de secondes.

volume : volume du son compris entre 0 et 15. Vaut 12 par défaut.

enveloppe de volume : nombre compris entre 0 et 15 permettant de sélectionner une enveloppe de volume déterminée par la commande env. Par défaut, on obtient un volume constant.

enveloppe de ton : nombre compris entre 0 et 15 permettant de sélectionner une enveloppe de ton déterminée par la commande ent. Par défaut, on obtient un ton constant.

période de ton : nombre compris entre 0 et 31, valant 0 par défaut. Plus on se rapproche de 31, plus la composante basse est présente dans le son.

ss réserve le bas de l'écran graphique au texte.

st fait apparaître la tortue sur l'écran graphique.

stop utilisé dans une procédure pour en arrêter l'exécution. Dans le cas où stop se trouve dans une procédure appelée par une autre, le contrôle sera rendu

à la procédure appelante. Un throw "TOPLEVEL, par contre, stoppera l'exécution de toute procédure et reviendra au signal de sollicitation LOGO (?).

tf fournit une liste composée de six éléments permettant de définir l'état courant de la tortue :
1 et 2 : coordonnées horizontale et verticale de la position courante de la tortue.
3 : direction de la tortue en degrés d'angle.
4 : vaut soit pd soit pu suivant que le crayon est abaissé ou levé.
5 : désigne le numéro de crayon utilisé par la tortue (1, 2 ou 3).
6 : vaut TRUE si la tortue est visible et FALSE si elle est invisible.

throw suivi d'un nom de variable, renvoie à la procédure contenant le catch correspondant (*voir* catch et TOPLEVEL pour exemples).

to ?to nom_de_procédure
signale qu'on va définir une procédure.

TOPLEVEL throw "TOPLEVEL
stoppe toutes les procédures en cours d'exécution et retourne au signal de sollicitation (?).
Voir aussi stop.
Exemple :

```
?to appel
>hello
>pr [ici appel]
>end
appel defined
?to hello
>if rc = "a [throw "TOPLEVEL] [stop]
>pr [coucou]
>end
hello defined
?appel
-- presser une touche autre que a --
ici appel
?appel
-- presser la touche a --
?
```

Lorsqu'on enfonce une touche autre que a, la procédure hello exécute [stop], ce qui a pour effet de stopper sa propre exécution et de renvoyer le contrôle à la procédure appelante (appel). Lorsqu'on enfonce la touche a, la procédure hello exécute

[throw "TOPLEVEL], ce qui a pour effet de stopper l'exécution de toute procédure en attente et de revenir à ?.

TRUE valeur système : vrai.

```
?3 > 2
TRUE
```

ts met la totalité de l'écran à la disposition du texte.

tt

```
tt "texte
```

permet d'afficher des caractères texte dans l'écran graphique.

```
?repeat 4 [fd 100 tt "XX rt 90]
```

type

```
type "mot
type [liste]
```

affiche à l'écran le mot ou la liste désignés sans effectuer de retour chariot.

```
?type "a type "b
ab
```

Voir aussi pr et show.

wait

```
wait X
```

stoppe le processeur pendant une durée de X fois 0.22 secondes.

window spécifie que le mode écran fonctionne comme une fenêtre. La tortue peut dès lors quitter les bords de l'écran sans que rien ne se passe. Les limites de l'écran graphique sont ainsi étendues de -320 à +319 en horizontal et de -200 à +199 en vertical.

word

```
word "mot1 "mot2
```

fournit un seul mot obtenu par la concaténation des deux mots spécifiés. Si l'on désire concaténer plus de deux mots, il faut entourer le tout de parenthèses.

```
?word "cou "cou
coucou
?(word "valeur " = "1/2)
valeur = 0.5
```

(remarquez que les opérations arithmétiques sont effectuées).

wordp wordp paramètre
fournit TRUE lorsque le paramètre est un mot ou un nombre, FALSE s'il est une liste.

```
?wordp 1/2
TRUE
?wordp "coucou
TRUE
?wordp [bla bla bla]
FALSE
```

wrap après cette commande, lorsque la tortue dépasse les bords de l'écran, elle revient par le côté opposé.

COMMANDES ET MOTS-CLES PROPRES AU LOGO 3

.ENL caractérise la fin d'une ligne de procédure interrompue par un retour chariot ou des espaces.

.EMT caractérise le début d'une ligne de procédure interrompue par un retour chariot ou des espaces.

.REM caractérise une remarque.

.in

.in nombre (0 à 255)
cette fonction lit le contenu du port d'entrée sortie spécifié (comme la fonction Basic INP). Cependant, la structure particulière des PORTS de l'AMSTRAD est inadaptée à cette fonction. La valeur est limitée à 255 (système classique) alors que l'AMSTRAD utilise 65536 PORTS.

.out

.out nport valeur
cette fonction permet d'écrire la valeur spécifiée sur le PORT spécifié. Pour les mêmes raisons que .in, cette instruction est inutilisable sur l'AMSTRAD.

arctan

arctan nombre
fournit l'arctangente en degrés du nombre spécifié.

changef

changef "nfichier1 "nfichier2
change le nom de fichier nfichier2 en nfichier1.

copyon

copyon
active l'écho imprimante.

copyoff

copyoff
inhibe l'écho imprimante.

cursor

cursor
fournit une liste de deux éléments qui indiquent la position courante du curseur (colonne ligne).

defaultd

defaultd
fournit le nom du lecteur de disque courant.

define

define "nom [[] [.....]]
définit une procédure en mode direct.

dirpic

dirpic ":unité
affiche le répertoire de tous les fichiers de sauvegarde de l'écran graphique (extension PIC).

dotc

dotc x y
fournit le numéro de la couleur du point spécifié par x et y. Si le point n'est pas à l'écran, la fonction fournit -1.

edall

edall
charge dans l'éditeur toutes les variables et toutes les procédures existantes dans l'espace de travail.

edf

edf "nom
charge à partir du disque le fichier dont le nom est spécifié. Le chargement se fait dans l'espace EDITEUR et celui-ci est activé.

erall

erall
efface toutes les procédures et toutes les variables de l'espace de travail.

fill

fill
colorie la zone limitée par des traits de la couleur courante du crayon dans la couleur du crayon.

home

home
ramène la tortue à sa position de départ au centre de l'écran.

last

last [liste]
last "mot
last nombre
fournit le dernier élément du paramètre précisé.

lc

lc "mot
convertit le mot spécifié en minuscule.

listp

listp objet
fournit la valeur TRUE si l'objet est une liste, la valeur FALSE dans les autres cas.

loadpic

loadpic "nomfichier
charge le fichier écran graphique spécifié.

lput

lput objet1 objet2
ajoute objet1 à la suite de objet2 pour former un nouvel objet.

memberp

memberp objet1 objet2
fournit la valeur TRUE si objet1 est un élément d'objet2. Sinon, fournit la valeur FALSE.

namep
mamep objet
fournit la valeur TRUE si l'objet est une variable définie, FALSE sinon.

noformat
noformat
supprime le formatage des procédures de l'espace de travail.

notrace
notrace
inhibe la fonction trace (affichage des noms de procédure en cours d'exécution).

nowatch
nowatch
inhibe la fonction watch (affichage des noms d'expression en cours d'exécution).

numberp
numberp objet
fournit TRUE si l'objet spécifié est un nombre et FALSE sinon.

piece
piece debut fin objet
fournit les parties d'objet situées entre le début et la fin spécifiés ; début et fin sont des paramètres numériques. Cette fonction est à rapprocher de la fonction Basic MID$.

poall
poall
affiche toutes les définitions de variables ou de procédures contenues dans l'espace de travail.

pons
pons
affiche tous les noms et toutes les valeurs des variables globales contenues dans l'espace de travail.

pops
pops
affiche toutes les procédures contenues dans l'espace de travail.

pps
pps
affiche toutes les paires de propriétés non standard de tous les objets contenus dans l'espace de travail.

quotient
quotient nombre1 nombre2
fournit le quotient entier de la division de nombre1 par nombre2.

remainder remainder nombre1 nombre2
fournit le reste de la division entière de nombre1 par nombre2. A rapprocher de la fonction BASIC MOD.

rerandom rerandom
réinitialise le générateur de nombre aléatoire (RANDOM) à sa séquence initiale (la première fournie).

round round nombre
arrondit le nombre spécifié. Le passage à l'unité directement supérieure se fait à partir de la demi-unité (.5).

savepic savepic "nomfichier
sauve l'écran graphique sur disque sous le nom spécifié.

setbg setbg nombre (0 à 3)
positionne la couleur du fond dans la teinte correspondant au crayon spécifié.

setcursor setcursor [colonne ligne]
positionne le curseur aux coordonnées spécifiées dans la liste.

setd setd u:
spécifie l'unité disque par défaut (u).

setscrunch setscrunch nombre
donne le nombre spécifié comme rapport de la taille horizontale de l'écran sur la taille verticale.

setx setx nombre
positionne la tortue à la colonne spécifiée.

sety sety nombre
positionne la tortue à la ligne spécifiée.

shuffle shuffle [liste]
fournit une nouvelle liste constituée des éléments de la liste spécifiée mais dans un ordre aléatoire.

text text "nom
fournit le contenu de la procédure "nom spécifiée.

thing thing "nom
fournit la valeur de la variable spécifiée sous forme de nom.

COMMANDES ET MOTS-CLES PROPRES AU LOGO 3

towards

towards [x y]

pointe la tortue en direction des coordonnées spécifiées dans la liste.

trace

trace

active la fonction trace (affichage du nom de la procédure en cours d'exécution).

uc

uc "mot

convertit le mot spécifié en majuscules.

watch

watch

active la fonction watch (affichage du nom de chaque expression en cours d'exécution).

where

where objet1 objet2

fournit l'emplacement de l'objet1 dans l'objet2. Cette fonction est à rapprocher de la fonction BASIC INSTR.

Généralités

Le **Z80-SIO** (pour Serial Input/Output) est un circuit d'interfaçage complexe fabriqué par ZILOG pour satisfaire à un grand nombre d'applications de transmission série.

L'AMSTRAD l'utilise comme composant principal de l'interface RS232 en association avec le CTC 8253.

Le SIO se compose de deux canaux distincts capables de gérer entièrement des transmissions asynchrones ou synchrones suivant des protocoles tels que HDLC ou SDLC. Il peut également être utilisé dans toute application nécessitant une transmission de données série comme un lecteur à cassettes ou encore un lecteur de disques.

Le Z80-SIO peut aussi générer et vérifier des codes CRC (Cyclic Redundancy Check de type CRC - 16 ou CRC - CCITT) dans son application synchrone et gérer tous les signaux modem.

Enfin, il peut travailler en mode POLLING (INTERRUPTION) sur les deux canaux A et B. Chaque source d'interruption peut être activée par programme, le canal A ayant un niveau de priorité supérieur au canal B.

Interfaçage du SIO avec le CPC

Comme tous les circuits périphériques qui composent le CPC, le SIO-Z80 est interfacé par l'intermédiaire des ports d'entrée/sortie du Z80.

Le SIO n'échappe pas à la structure de gestion des ports propre à AMSTRAD. Il est donc situé à des adresses de ports codées sur 16 bits.

Le SIO utilise quatre ports, deux pour le canal A et deux pour le canal B. Ils sont bidirectionnels et peuvent donc être utilisés en entrée comme en sortie.

Table des adresses et des fonctions des ports

Adresse	Fonction
FADC	Données du canal A.
FADD	Contrôle du canal A.
FADE	Données du canal B.
FADF	Contrôle du canal B.

Rappel : pour adresser directement un port en assembleur, vous devez charger le registre B avec l'octet de poids fort (ici FAH) avant toute opération utilisant les mnémoniques IN ou OUT.

Registres et programmation du Z80-SIO

Les registres d'écriture

Pour chaque canal, le Z80-SIO contient **huit registres** d'écriture programmables séparément (WR0-WR7). Ces registres permettent de configurer chaque canal pour une application bien spécifique.

A l'exception du registre WR0, la programmation des registres d'écriture nécessite deux octets. Les bits de poids faible du premier octet (B0-B2) contiennent le numéro du registre à sélectionner. Le deuxième octet contient le mot de commande qui sera écrit dans le registre sélectionné.

Remarque *:* une fois le registre sélectionné, le programmeur en fait ce qu'il désire. Il peut soit le lire pour le tester, soit y écrire une commande. En initialisant le Z80-SIO de manière structurée, le programmeur aura à sa disposition des modules entrée/sortie puissants.

La particularité du registre WR0 réside dans le fait que la programmation de ses commandes de base ne nécessite qu'un seul octet. Sa sélection est obtenue par un RESET interne ou externe.

Les commandes de base (CMD0-CMD2) ainsi que les codes de contrôle CRC (CRC0-CRC1) sont déterminés respectivement par le positionnement des bits (B3-B5) et (D6-D7) du premier octet de chaque registre d'écriture. Ceci permet un maximum de souplesse au système de commande.

Chaque canal possède tous les registres décrits ci-après. Ils sont adressés comme des commandes et non comme des DATAs.

Registre d'écriture WR0

WR0 est le registre de commande. Cependant, il est également utilisé pour l'initialisation des codes CRC et pour la sélection des autres registres.

B7	B6	B5	B4	B3	B2	B1	B0
CRC reset n° 1	CRC reset n° 0	CMD n° 2	CMD n° 1	CMD n° 0	PTR n° 2	PTR n° 1	PTR n° 0

◊ Bits de pointeurs (B2-B0)

Les bits B2 à B0 du registre WR0 déterminent le numéro du registre dans lequel le deuxième octet de commande sera écrit ou lu. Après un RESET exécuté par commande ou par une entrée externe, le premier octet est introduit dans le registre WR0. Une lecture ou une écriture dans n'importe quel registre (excepté le registre WR0) initialisera le pointeur de registre sur WR0.

◊ Bits de commande (B5-B3)

Les différentes combinaisons des bits B5 à B3 permettent la sélection des sept commandes de base du Z80-SIO.

Cde	CMD n° 2	CMD n° 1	CMD n° 0	
0	0	0	0	Commande sans effet.
1	0	0	1	Envoie ABORT en mode SDLC.
2	0	1	0	Initialise les interruptions externes/états.
3	0	1	1	Initialisation du canal.
4	1	0	0	Autorise les interruptions sur le prochain caractère de réception.
5	1	0	1	Initialise les interruptions en transmission.
6	1	1	0	Initialise le LATCH d'erreur.
7	1	1	1	Retour d'interruption (canal A).

Commande 0. Elle est normalement utilisée pour rendre le SIO inactif pendant que les pointeurs sont initialisés en vue de charger l'octet suivant.

Commande 1 (arrêt en mode SDLC). Cette commande est uniquement utilisée en mode SDLC. Elle génère une suite de huit à treize 1.

Commande 2 (initialisation des interruptions externe/état). Après une interruption externe/état, les bits d'état du registre RR0 (registre de lecture) sont mémorisés. La commande 2 les initialise à nouveau et autorise une nouvelle interruption.

Commande 3 (initialisation du canal). Cette commande provoque le même effet qu'une initialisation externe, mais uniquement pour un canal. L'initialisation du canal A initialise également le niveau de priorité des interruptions. Après une initialisation, tous les registres du canal doivent être programmés à nouveau.

Commande 4 (autorisation des interruptions sur le prochain caractère de réception). Si l'interruption est validée sur le premier caractère reçu, cette commande réactivera ce mode après réception de chaque message complet et préparera ainsi le SIO pour le prochain message.

Commande 5 (initialisation des interruptions en transmission). Lorsque les interruptions sont validées sur la transmission, un signal d'interruption est généré quand le tampon de transmission est vide. Dans le cas où il n'y a plus de caractère à transmettre (fin de message), cette commande interdit toute interruption en transmission jusqu'au chargement d'un nouveau caractère dans le tampon de transmission ou jusqu'à l'envoi complet d'un code CRC.

Commande 6 (initialisation de la mémoire d'erreur). Les erreurs de parité ou d'overrun sont mémorisées dans le registre RR1. De cette façon, les erreurs apparues lors d'un transfert de données peuvent être analysées à la fin du bloc. La mémoire d'erreur (latch) est remise à 0 par l'exécution de la commande 6.

Commande 7 (retour d'interruption). Cette commande doit provenir du canal A. Elle est interprétée par le SIO de la même façon qu'une commande RETI pour le Z80. Elle initialise le latch interne d'interruption de la plus haute priorité en service et permet aux circuits de priorité inférieure de générer des interruptions via le DAISY CHAIN (chaînage interne et externe entre circuits de la famille Z80 qui génère les priorités d'interruptions). Cette commande peut également être utilisée pour le DAISY CHAIN interne du SIO, même pour des systèmes sans chaînage externe ou sans commande RETI.

◊ Initialisation des codes CRC (B7-B6)

La combinaison de ces deux bits sélectionne une des quatre commandes d'initialisation suivantes :

CRC n° 1	CRC n° 0	
0	0	Aucun effet.
0	1	Initialisation et vérification du CRC en réception.
1	0	Initialisation du générateur de CRC en transmission.
1	1	Initialisation underrun Tx et fin de message.

Remarque : la commande d'initialisation du générateur de CRC est normalement programmée pour générer des 0. En mode SDLC, elle génèrera des 1. De même, la vérification des codes CRC en réception sera programmée pour des valeurs 1 en mode SDLC.

Registre d'écriture WR1

Le registre d'écriture contient les bits de commandes des différents modes d'interruption aïnsi que les fonctions d'attente (WAIT) et système prêt (READY).

B7	B6	B5	B4
wait/ready validation	$\overline{\text{wait}}$ ou ready fonctions	wait/ready réception transmission	interruption réception mode 1
B3	**B2**	**B1**	**B0**
interruption réception mode 0	vecteur d'affectation d'état	interruption transmission validation	interruption externes validation

B0 : validation des interruptions externes/état

Si le latch UNDERUN/EOM contient 1, alors le positionnement de ce bit permet aux interruptions externes/état d'avoir lieu suite à une transition sur les entrées modem DCD, CTS ou SYNC ou à la détection d'un abandon ou d'un arrêt immédiat ou au début de transmission d'un CRC ou d'un caractère de transmission de synchronisation.

B1 : validation des interruptions en transmission

Validée, l'interruption se produit chaque fois que le tampon de transmission est vide.

B2 : vecteur d'affectation d'état

Ce bit est uniquement programmé dans le canal B. S'il est à l'état 0, le vecteur d'interruption programmé dans le registre WR2 est renvoyé inchangé, suite à un accusé de réception d'interruption. Si ce bit est à l'état 1, le vecteur est renvoyé après avoir été modifié suivant les conditions d'interruption décrites ci-dessous.
Seuls les bits V3 à V1 du vecteur d'interruption sont utilisés.

V3	V2	V1	
0	0	0	Canal B int. sur le tampon de transmission vide.
0	0	1	Canal B int. sur le changement externe/état.
0	1	0	Canal B int. sur le caractère de réception disponible.
0	1	1	Canal B int. sur des conditions spéciales réception.
1	0	0	Canal A int. sur le tampon de transmission vide.
1	0	1	Canal A int. sur le changement externe/état.
1	1	0	Canal A int. sur caractère de réception disponible.
1	1	1	Canal A int. sur des conditions spéciales réception.

Conditions spéciales de réception : erreur de parité, overrun, erreur de trame et fin de trame (SDLC).

B4-B3 : interruption en réception mode 1 et mode 0

La combinaison de ces deux bits spécifie les différentes possibilités d'interruption en réception. En mode d'interruption 1, 2 ou 3, une condition spéciale de réception provoque une modification du vecteur d'interruption.

D4 interruption réception mode 1	D3 interruption réception mode 0	
0	0	Interdiction des interruptions en réception.
0	1	Interruption sur le premier caractère.
1	0	Interruption sur tous les caractères reçus (l'erreur de parité étant une condition spéciale).
1	1	Interruption sur tous les caractères reçus (l'erreur de parité n'étant pas une condition spéciale).

B7-B6 : fonction wait/ready

Les fonctions wait et ready sont sélectionnées par le contrôle des bits B7 et B6. Elles sont validées si le bit B7 du registre WR1 est positionné à 1. La fonction **ready** est sélectionnée en positionnant le bit B6 à 1. La sortie commute d'un niveau haut vers un niveau bas lorsque le Z80-SIO est prêt à transférer des données. La fonction **wait** est sélectionnée par la valeur 0 dans B6 et la sortie à l'état drain ouvert passe à un état bas lorsqu'elle est active. Les fonctions wait et ready peuvent être utilisées en réception ou en transmission mais pas simultanément.

B7 = 0			
B6 = 1	$\overline{\text{ready}}$ à l'état haut.	B6 = 0	$\overline{\text{wait}}$ est flottant.
B7 = 1			
B5 = 0	$\overline{\text{ready}}$ est haut quand le tampon de transmission est plein. $\overline{\text{wait}}$ est bas quand le tampon de transmission est plein et qu'un port data du SIO est sélectionné.	B5 = 1	$\overline{\text{ready}}$ est haut quand le tampon de réception est vide. $\overline{\text{wait}}$ est bas quand le tampon de réception est vide et qu'un port data du SIO est sélectionné.

$\overline{ready}$ est bas quand le tampon de transmission est vide.	$\overline{ready}$ est bas quand le tampon de réception est plein.
$\overline{wait}$ est flottant quand le tampon de transmission est vide.	$\overline{wait}$ est bas quand le tampon de réception est plein.

Registre d'écriture WR2

WR2 est le registre qui contient le vecteur d'interruption. Il n'existe que dans le canal B. Lors d'un accusé d'interruption, les bits B7, B6, B5, B4 et B0 sont retournés sans aucune modification. Seuls les bits B3, B2 et B1 sont modifiés si le bit vecteur d'affectation d'état du registre WR1 est égal à 1.

B7	B6	B5	B4	B3	B2	B1	B0
V7	V6	V5	V4	V3	V2	V1	V0

Registre d'écriture WR3

WR3 contient les bits et les paramètres de commande de la logique de réception.

B7	**B6**	**B5**	**B4**
réception bits car 1	réception bits car 0	auto validation	phase de recherche d'entrée
B3	**B2**	**B1**	**B0**
validation du CRC en réception	mode de recherche d'adresses	inhibition chargement caract sysnc	validation réception

B0 : validation de réception

Après avoir initialisé tous les paramètres de réception, le positionnement de B0 à 1 permet le début du fonctionnement en mode réception.

B1 : inhibition du chargement du caractère de sysnc

Si cette option est sélectionnée, le caractère de synchronisation précédant le message n'est pas chargé dans le tampon de réception.

B2 : mode de recherche d'adresses

Si le mode SDLC est sélectionné, les messages dont les adresses ne correspondent pas aux adresses programmées dans le registre WR6 ou à l'adresse globale (11111111) sont rejetés. En résumé, en mode de recherche adresses, aucune interruption sur la réception n'aura lieu s'il n'y a pas concordance d'adresses.

B3 : validation du CRC en réception

Si le bit B3 est positionné à 1, le calcul du CRC commence sur le début du dernier caractère transféré depuis le registre à décalage de réception vers la pile tampon, indépendamment du nombre de caractères présents dans la pile (*voir* les sections SDLC réception et CRC vérification du manuel technique du Z80-SIO de ZILOG).

B4 : phase de recherche d'entrée

Après un reset, le Z80-SIO lance automatiquement la phase de recherche. Cependant, cette dernière peut être réactivée si le caractère de synchronisation est perdu ou si le contenu d'un message n'est pas nécessaire (mode HDLC). La réactivation de la phase de recherche est obtenue en positionnant le bit B4 du registre WR3 à 1. Ceci positionne le bit B4 de RR0 (sync/recher).

B5 : auto validation

Si ce mode est sélectionné, les signaux $\overline{DCD}$ et $\overline{CTS}$ deviennent les signaux de validation de réception et de transmission. Si B5 est égal à 0, DCD et CTS positionnent simplement leur bit d'état respectif dans RR0.

B7-B6 : réception de bits/caractères

Ces deux bits déterminent le nombre de bits reçus pour former un caractère. Il est possible de les modifier durant l'assemblage d'un caractère et ce, avant que le nombre de bits programmés ne soit atteint.

B7	B6	Bits/caractères
0	0	5
0	1	7
1	0	6
1	1	8

Registre d'écriture WR4

WR4 contient les bits qui affectent la transmission et la réception du SIO.

B7	B6	B5	B4	B3	B2	B1	B0
fréquence horloge	fréquence horloge	modes sync	modes sync	stop bit	stop bit	parité paire	parité
1	0	1	0	1	0	impaire	

B0 : parité
Si ce bit est positionné à 1, un bit supplémentaire est additionné aux bits spécifiés par la commande (bits/caractères) du registre WR3. Dans le mode réception, le bit de parité reçu est transmis au CPU comme partie intégrante du caractère, sauf si un caractère de 8 bits est spécifié.

B5 : parité paire/$\overline{\text{impaire}}$

Dans le cas où la parité est spécifiée, ce bit est à 0 si elle est paire et à 1 si elle est impaire.

B3-B2 : stop bits

Ils déterminent le nombre de bits d'arrêt ajoutés à chaque caractère asynchrone envoyé. A la réception, la présence d'un stop bit est toujours vérifiée. Le mode spécial (00) signifie qu'un mode synchrone doit être sélectionné.

B3	B2	
0	0	Mode synchrone.
0	1	1 stop bit par caractère.
1	0	1 + 1/2 stop bits par caractère.
1	1	2 stop bits par caractère.

B5-B4 : modes synchrones

Ces bits sélectionnent les différentes options de caractère de synchronisation.

B5	B4	
0	0	Synchronisation sur 8 bits.
0	1	Synchronisation sur 16 bits.
1	0	Mode SDLC (01111110).
1	1	Mode synchronisation externe.

B7-B6 : fréquence horloge

Ces bits déterminent le coefficient de multiplication entre les horloges TxC et RxC. TxC représente l'horloge d'émission et RxC représente l'horloge de réception. Pour le mode synchrone, le coefficient 1 doit être spécifié. Toutes les fréquences de transmission sont permises en mode asynchrone. Dans tous les cas, l'horloge système doit être au moins 4,5 fois plus rapide que la fréquence des données. Lorsque le facteur de multiplication choisi est égal à 1, la synchronisation doit provenir d'une source extérieure.

B7	B6	
0	0	Fréq horloge x 1 = fréq data
0	1	Fréq horloge x 16 = fréq data
1	0	Fréq horloge x 32 = fréq data
1	1	Fréq horloge x 64 = fréq data

Registre d'écriture WR5

WR5 contient les bits de contrôle des opérations de transmission, à l'exception du bit B2 qui affecte la réception et la transmission.

B7	B6	B5	B4	B3	B2	B1	B0
DTR	TX BIT CAR N°1	TX BIT CAR N°2	ARRET IMMED TX	VALIDATION TX	CRC-16 SDLC	RTS	TX CRC VALIDAT

B0 : validation CRC en transmission

Ce bit détermine si le CRC est calculé sur un caractère spécial de transmission. S'il est égal à 1, le caractère est chargé du registre de transmission vers le registre à décalage de transmission et le CRC est calculé sur ce caractère. Le CRC n'est pas automatiquement transmis sauf si ce bit est positionné à 1 lorsqu'une condition de transmission UNDERRUN existe.

B1 : $\overline{RTS}$ ou demande d'envoi (Request To Send)

Lorsque le bit de commande RTS est égal à 1, la sortie RTS du SIO passe à un état bas et passe à un état haut après un reset. En mode asynchrone, la sortie RTS ne passe à un état haut qu'après transmission complète de tous les bits constituant le caractère et qu'une fois que le tampon de transmission est vide.

B2 : CRC-16/$\overline{SDLC}$

Ce bit sélectionne le type de polynôme du calcul du CRC pour la transmission et la réception. S'il est égal à 1, c'est le CRC-16 ($x16 + x15 + x12 + 1$) qui est sélectionné. S'il est égal à 0, c'est le polynôme CRC SDLC ($x16 + x15 + x5 + 1$) qui est sélectionné. Si le mode SDLC est utilisé, le générateur de CRC, ainsi que le système de vérification sont initialisés pour tout 1 et une vérification spéciale est appliquée. Si le mode SDLC n'est pas sélectionné, le générateur et le système de vérification sont initialisés pour tout 0 (pour les deux polynômes).

B3 : validation de transmission

Les données ne sont pas transmises tant que le bit B3 du registre WR5 est positionné à 1 et que l'état de la sortie de transmission est maintenu en marquage. Dès que ce bit est remis à 0, les données et les caractères de synchronisation en cours de transmission sont entièrement transmis. Si le transmetteur est arrêté pendant la transmission d'un caractère CRC, des caractères de synchronisation ou de drapeau sont envoyés à la place de CRC.

B4 : arrêt immédiat (BREAK)

Si ce bit est égal à 1, la sortie de transmission (Transmit Data) est placée à un état espacement (space) et ce, quelles que soient

les données transmises. Lorsqu'il est remis à 0, la sortie de transmission retourne à un état de marquage (marking).

B6-B5 : transmission bits/caractères

B6 et B5 déterminent le nombre de bits composant chaque octet transféré dans le tampon de transmission.

B6 : bit de transmission caractère n° 1	B5 : bit de transmission caractère n° 0	Nombre de bits par caractère
0	0	5 ou -
0	1	7
1	0	6
1	1	8

Les bits des octets à envoyer doivent être justifiés à droite. Le mode 5 peut transmettre 1 à 5 bits par caractère, mais le caractère doit être formaté comme ci-dessous :

B7	B6	B5	B4	B3	B2	B1	B0	
1	1	1	1	0	0	0	D	Envoie 1 bit de données.
1	1	1	0	0	0	D	D	Envoie 2 bits de données.
1	1	0	0	0	D	D	D	Envoie 3 bits de données.
1	0	0	0	D	D	D	D	Envoie 4 bits de données.
0	0	0	D	D	D	D	D	Envoie 5 bits de données.

B7 : $\overline{\text{DTR}}$ ou Terminal de Données Prêt (Data Terminal Ready)

Positionné à 1, le signal DTR est actif bas. Positionné à 0, il est inactif.

Registre d'écriture WR6

Ce registre est programmé pour contenir : soit le caractère de synchronisation en mode MONOSYNCHRONE ; soit les huit premiers bits des seize bits formant un caractère de synchronisation en mode BISYNCHRONE ; soit un caractère de synchronisation de transmission en mode SYNCHRONISATION EXTERNE.

En mode SDLC, il est programmé pour contenir le champ d'adresses secondaire destiné à être comparé avec le champ d'adresses de la trame SDLC.

B7	B6	B5	B4	B3	B2	B1	B0
sync7	sync6	sync5	sync4	sync3	sync2	sync1	sync0

Registre d'écriture WR7

Ce registre est programmé pour contenir les caractères de synchronisation de réception dans le mode MONOSYNCHRONE, les huit

bits de poids fort d'un caractère de synchronisation en mode BISYNCHRONE ou un caractère drapeau (01111110) en mode SDLC. Le registre WR7 n'est pas utilisé en mode de synchronisation externe.

B7	B6	B5	B4	B3	B2	B1	B0
sync 15	sync 14	sync 13	sync 12	sync 11	sync 10	sync9	sync8

Les registres de lecture

Le Z80-SIO possède trois registres de lecture qui sont RR0, RR1 et RR2. Ils permettent d'obtenir des informations sur l'état de chaque canal à l'exception de RR2 pour le canal B. Ces informations incluent les conditions d'erreurs ainsi que les signaux standards des interfaces de communication.

Pour lire un registre autre que RR0, il faut d'abord charger dans WR0 le pointeur de registre exactement de la même façon que pour une opération d'écriture.

Registre de lecture RR0

Ce registre contient l'état des tampons de réception et de transmission. Les entrées DCD, CTS et SYNC ; le latch de transmission UNDERRUN/EOM et le latch BREAK/ABORT.

B0 Caractère de réception disponible.
B1 Toute demande d'interruption positionne ce bit à 1, mais il ne peut être lu que par le canal A.
B2 Tampon de transmission vide. Est égal à 1 sauf si un caractère CRC en mode sync ou SDLC est envoyé.
B3 DCD détection de porteuse (Data Carrier Detect).
B4 Synchronisation/recherche. Présente un mode de fonctionnement différent suivant qu'on se trouve en mode synchrone ou asynchrone. Pour plus de détails, consulter le manuel ZILOG-Z80.
B5 CTS. Mettre à 0 pour envoyer (Clear To Send). Ce signal est similaire au DCD, excepté qu'il montre un état inverse du signal CTS actif bas.
B6 Transmission UNDERRUN et fin de message.
B7 Arrêt immédiat et abandon.

Registre de lecture RR1

Ce registre contient les bits d'état des conditions spéciales de réception et les codes résiduels pour le champ 1 en mode de réception SDLC.

B0 Tout est envoyé (All Send). Ce bit est toujours mis à 1 pour le mode synchrone.

B3-B1 Codes résiduels 0, 1 et 2 réception SDLC.
B4 Erreur de parité.
B5 Erreur overrun en réception.
B6 Erreur de CRC et de trame.
B7 Fin de trame (SDLC).

Registre de lecture RR2 - canal B uniquement

Ce registre contient le vecteur d'interruption écrit dans le registre WR2 si le bit d'affectation du vecteur d'état n'est pas mis à 1. Si le bit de commande est mis à 1, il contient le vecteur modifié par les conditions spéciales d'interruption. Lorsque ce vecteur est lu, il est retourné modifié par la condition d'interruption supérieure au moment de la lecture. S'il n'y a pas d'interruption en attente, le vecteur est modifié avec V1=1, V2=1 et V3=0.

B7	B6	B5	B4	B3	B2	B1	B0
V7	V6	V5	V4	V3	V2	V1	V0

V3, V2, V1 : modifiés si le bit du vecteur d'affectation d'état est validé.

Remarque : l'interprétation des registres d'écriture étant complexe, nous invitons le lecteur désireux d'approfondir la question à se référer au manuel technique du Z80-SIO.

CTC 8253

Généralités

Le **8253** est un circuit intégré fabriqué par INTEL. Il est utilisé pour résoudre la plupart des problèmes de comptage et de temporisation de la micro-informatique. Au lieu de programmer des boucles de temporisation, le programmeur peut spécifier simplement ses désirs au moyen de quelques instructions élémentaires d'entrée/sortie.

On retrouve le 8253 dans des applications aussi diverses que les générateurs de vitesse de transmission (BAUD RATE GENERATOR), les compteurs d'événements, les horloges en temps réel ou le contrôle complexe de moteurs.

Dans l'interface **RS232** du CPC, il est utilisé comme générateur d'horloge pour le SIO. C'est donc par l'intermédiaire du 8253 que l'on programme les vitesses de transmission et de réception de cet interface.

Le 8253 est composé de trois compteurs indépendants de 16 bits chacun. Ces compteurs sont appelés 0, 1 et 2. Les trois compteurs ont une structure totalement identique. Chaque compteur se comporte comme un simple "décompteur" (COUNTER DOWN) de 16 bits dont la valeur de départ peut être positionnée par le logiciel (PRESETABLE).

Ils peuvent compter en binaire (de 65535 à 0) ou en BCD (décimal codé binaire). Dans ce dernier cas, les 16 bits sont découpés en quatre tranches de quatre bits, chaque tranche pouvant contenir un nombre entre 0 et 9. Le compteur peut alors compter de 9999 à 0.

L'interfaçage de chaque compteur avec le monde extérieur est réalisé par l'intermédiaire de trois signaux physiques, l'horloge, la gachette et la sortie.

L'horloge reçoit les impulsions de comptage en provenance d'un dispositif oscillant, la gachette est un signal entrant qui permet de déclencher certaines fonctions et la sortie est le signal sortant qui indique l'état du compteur.

Interfaçage avec le CPC

L'interfaçage du 8253 se fait au moyen de quatre ports d'entrée/sortie.

Ports utilisés

Port	Fonction en écriture	Fonction en lecture
FBDCH	Ecriture compteur 0	Lecture compteur 0
FBDDH	Ecriture compteur 1	Lecture compteur 1
FBDEH	Ecriture compteur 2	Lecture compteur 2
FBDFH	Ecriture REGISTRE MODE	Non utilisable

Ces différentes fonctions seront décrites au point suivant.

Registre et programmation

Le REGISTRE MODE

Le 8253 se programme par l'intermédiaire d'un seul registre appelé REGISTRE MODE.

Ce registre est à écriture seule et il permet d'accéder séparément à chacun des trois compteurs.

Format du REGISTRE MODE

B7	B6	B5	B4	B3	B2	B1	B0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

SC0 et **SC1** : ces bits permettent de sélectionner un des trois compteurs selon la table suivante :

SC1	SC0	Compteur
0	0	0
0	1	1
1	0	2
1	1	illégal

RL0 et **RL1** : permettent de définir la sélection de la procédure de lecture ou d'écriture suivant la table figurant ci-dessous :

RL1	RL0	Procédure
0	0	Mémorisation temporaire du compteur.
0	1	Ecriture/lecture de l'octet le moins significatif.
1	0	Ecriture/lecture de l'octet le plus significatif.
1	1	Ecriture/lecture des deux octets consécutifs (le moins significatif en premier).

M0, M1 et M2 : permettent de définir le mode opératoire du compteur suivant la table que voici :

M2	M1	M0	MODE
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	2
1	1	1	3

Les six modes possibles sont définis au point "Définition des modes" ci-dessous.

BDC : ce bit permet de sélectionner le mode de fonctionnement du compteur.

Si BCD vaut 0, le compteur fonctionne en mode 16 bits.
Si BCD vaut 1, le compteur fonctionne en mode BCD.

Définition des modes

MODE 0 : interruption à la fin du comptage.

Dans ce mode, la sortie du compteur est positionnée à l'état bas après la sélection du mode. Après le chargement de la valeur du compteur, la sortie reste à l'état bas et le compteur commence à compter. Lorsque le comptage est terminé, la sortie passe à l'état haut et reste dans cet état jusqu'à ce qu'une nouvelle commande soit envoyée au compteur.

Si une nouvelle valeur de comptage est chargée pendant l'exécution, le chargement du premier octet arrête le compteur, le chargement du second octet redémarre le compteur qui commence à décompter en partant de la nouvelle valeur.

MODE 1 : génération d'une impulsion programmable.

Dans ce mode, la sortie passe à l'état bas après le changement d'état de la gachette. Le compteur commence alors à compter, à l'issue du comptage, la sortie repasse à l'état haut. Si une nouvelle transition apparaît sur la gachette, le compteur recommence son comptage (le compteur est dit "RETRIGGERABLE").

MODE 2 : générateur de vitesse de transmission.

Dans ce mode, le compteur se comporte comme un diviseur par n. La sortie passe à l'état bas pendant une période d'horloge et ce, à chaque passage à zéro du compteur qui se réarme automatiquement.

Si une nouvelle valeur est chargée pendant un cycle, elle commencera son action à la fin du cycle en cours.

Le passage à l'état bas de la gachette force la sortie à l'état haut. La transition de la gachette à l'état bas redéclenche le compteur.

MODE 3 : générateur de signaux carrés.

Dans ce mode, la sortie passe à l'état bas durant la moitié du comptage et à l'état haut durant l'autre moitié. Le compteur se réarme automatiquement. La sortie produit donc un signal dont l'état bas et l'état haut présentent la même période.

MODE 4 : impulsion générée par le logiciel.

Dans ce mode, la sortie, qui est à l'état haut au départ, passe à l'état haut pendant une période d'horloge à l'issue du comptage.

Si la gachette est à l'état bas, le compteur est inhibé.

MODE 5 : impulsion générée par le matériel.

Dans ce mode, le compteur commence à compter après la transition de la gachette vers l'état haut. A l'issue du comptage, la sortie passe à l'état bas pendant une période d'horloge. Si une nouvelle transition apparaît sur la gachette, le compteur recommence son comptage.

Ecriture dans un compteur

Le chargement d'une valeur dans un compteur doit se faire en fonction du choix effectué lors de la programmation du registre MODE (*voir* RL1 et RL0).

Le compteur doit donc être chargé avec un ou deux octets en fonction de l'état des bits RL1 et RL0.

Le chargement de la valeur ne doit pas nécessairement suivre la programmation du registre MODE. Elle peut être chargée à n'importe quel moment pour autant qu'elle reflète la sélection effectuée.

Si RL1 et RL0 sont positionnés à 1, les deux octets doivent être chargés dans le compteur. Le premier octet transmis prendra la place la moins significative (B0 à B7), l'autre occupera la place la plus significative (B8 à B15).

Lecture d'un compteur

Deux méthodes de lecture sont possibles :

- La première consiste à lire le port idoine, la première lecture fournissant l'octet le moins significatif, la seconde fournissant l'autre octet. Cette opération ne requiert qu'une seule précaution : il faut inhiber le compteur durant l'opération par l'intermédiaire de la gachette ou par l'interruption de l'horloge.
- La seconde utilise une des options des bits RL1 et RL0. Si ces deux bits sont à 0, lors de l'écriture dans le registre MODE, la valeur courante du compteur est copiée sur le port de lecture du compteur spécifié et cette valeur est maintenue jusqu'à ce qu'une lecture en prenne connaissance. Dans ce dernier cas, l'état des bits B4 à B0 du registre MODE n'a aucune importance.

Généralités

Le **PD765A** est un circuit LSI de la firme NEC utilisé pour contrôler et gérer jusqu'à quatre lecteurs de disques. Il est capable de supporter des formats de données au standard IBM 3740 (simple densité FM) ou au standard IBM 34 (double densité MFM) et ce, en simple ou en double face.

Le circuit PD765A possède des signaux permettant une liaison DMA. Si le mode DMA n'est pas choisi, le FDC peut travailler en mode interruption ou en mode polling. C'est cette dernière méthode qui a été choisie par le concepteur du CPC.

En résumé, l'interface lecteur du CPC utilise le contrôleur PD765A piloté par une horloge de 4 Mhz pour contrôler deux lecteurs interfacés aux adresses suivantes :

Adresse	Sortie	Entrée
#FA7E	Contrôle moteur	Non utilisé
#FB7E	Non utilisé	Registre d'état
#7B7F	Registre données	Registre données

Programmation et registres du FDC PD765A

Le PD765A possède deux registres accessibles par le CPU qui sont le registre d'état principal et le registre de données.

Le registre d'état principal est accessible en lecture à n'importe quel moment.

Le registre de données est en réalité constitué de différents registres rangés en pile dont un seul est présent sur le bus de données à un instant donné. Il manipule les données, les commandes, différents paramètres et le registre d'information d'état du lecteur.

Les octets de données sont lus ou écrits dans le registre de données suivant un ordre programmé pour obtenir une réponse précise après une commande particulière.

La relation entre le registre d'état/données et les signaux RD (read data) et WR (write data) et AO (adresse) est fournie par le tableau suivant :

AO	RD	WR	Fonction
0	0	0	Illégal.
0	0	1	Lecture du registre d'état principal.
0	1	0	Illégal.
0	1	1	Sans action.

A0	RD	WR	Fonction
1	0	0	Illégal.
1	0	1	Lecture du registre de données.
1	1	0	Ecriture du registre de données.
1	1	1	Sans action.

Séquence de commande

Le PD765A est capable d'exécuter **15 commandes différentes.** Chaque commande demande un transfert de plusieurs octets vers le CPU. Après exécution, le résultat est également retourné en plusieurs octets.

Vu la complexité des échanges entre le CPU et le PD765A, on considère que chaque commande s'effectue en trois phases :

- Phase commande. Le FDC reçoit toutes les informations nécessaires à une commande particulière.
- Phase exécution. Le FDC exécute la commande.
- Phase résultat. Le FDC renvoie au CPU des informations concernant le déroulement de l'exécution. Toutes les informations doivent être lues.

Remarque *:* la plupart des commandes requièrent jusqu'à 9 octets en phase commande et jusqu'à 7 octets dans les phases résultat.

Description des symboles utilisés dans les commandes

NP Numéro de piste.

ADT Adresse de tête 0 ou 1.

NS Numéro de secteur lecture/écriture.

LS Nombre de données écrites dans un secteur.

FP Numéro du dernier secteur d'une piste. Pendant une lecture ou une écriture, le FDC arrête de transférer les données après un EP.

GL Durant une lecture/écriture, cette valeur détermine le nombre d'octets au cours duquel le VCO doit rester inactif après deux octets CRC.

STP Pendant une opération de vérification, si STP=1, les données dans un secteur continu sont comparées, octet par octet, avec les données envoyées par le CPU (ou DMA) ; si le STP=2, les secteurs sont lus et comparés alternativement.

ET0
ET1
ET2
ET3
Durant la phase résultat, les registres mémorisent les états relatifs à une commande particulière. Ces registres ne doivent pas être confondus avec le registre d'état principal sélectionné par le fil d'adresse (A0=0).

NPC Numéro de piste après une commande d'interprétation d'état d'interruption.

SRT Step Rate du FDD de 1 à 16 ms par pas de 1 ms (délai entre deux impulsions de pas).

TCT Délai avant le chargement de la tête (de 2 à 254 ms par pas de 2 ms). Ce délai n'a de sens que pour le lecteur 8". Dans les petits lecteurs, la tête est presque toujours chargée avec le signal moteur OW.

TDT Délai de déchargement de la tête (de 16 à 240 ms par pas de 2 ms).

ND Opération en mode Non Data.

NNP Nouveau numéro de piste résultant d'une opération de recherche de piste.

SD0-SD1 Sélection du lecteur A ou B.

LD Lorsque LS est défini égal à 0, LD vaut la longueur de donnée qui sera utilisée pour la lecture et l'écriture dans un secteur.

ST Dans les lecteurs à deux têtes, la sélection de la face est effectuée par ce bit, Dans le cas de l'AMSTRAD, il est toujours égal à 0.

MF Sélection du mode MF ou MFM. S'il est égal à 1, le FDC travaille en double densité. Pour l'AMSTRAD, il est toujours égal à 1.

MT Egal à 1, il permet les opérations multipiste. A la fin des opérations de lecture/écriture sur la face 0, le FDC recherche automatiquement le secteur 1 sur la face 1 (valable uniquement pour les lecteurs à double tête). Toujours égal à 0 sur AMSTRAD.

TD Type de données qui doit être écrit dans le secteur.

NSP Indique le nombre de secteurs par piste.

SS Egal à 1, les secteurs effacés sont sautés. Toujours égal à 0 sur l'AMSTRAD.

W Ecriture.

R Lecture.

Les différentes commandes du PD765A

COMMANDE 1 - LECTURE DES DONNEES

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarques
Commande	W	MT	MF	SS	0	0	1	1	0	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	Information
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ID secteur
	W	NS	NS	NS	NS	NS	NS	NS	NS	avant
	W	LS	LS	LS	LS	LS	LS	LS	LS	exécution de
	W	FP	FP	FP	FP	FP	FP	FP	FP	la commande.
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	LD	LD	LD	LD	LD	LD	LD	LD	
Exécution	Transfert de données entre le FDD et le système.									
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	Information
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	d'état après
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	exécution de
	R	NP	NP	NP	NP	NP	NP	NP	NP	la commande.
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	Information
	R	NS	NS	NS	NS	NS	NS	NS	NS	ID secteur
	R	LS	LS	LS	LS	LS	LS	LS	LS	après commande.

COMMANDE 2 - LECTURE DE DONNEES EFFACEES.
On entend par données effacées des données d'un secteur marqué comme étant effacé.

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarques
Commande	W	MT	MF	SS	0	1	1	0	0	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	W	NS	NS	NS	NS	NS	NS	NS	NS	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	FP	FP	FP	FP	FP	FP	FP	FP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	LD	LD	LD	LD	LD	LD	LD	LD	
Exécution	Transfert de données entre FDD et le système.									
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 3 - ECRITURE DE DONNEES

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	
Commande	W	MT	MF	0	0	0	1	0	1	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	W	NS	NS	NS	NS	NS	NS	NS	NS	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	FP	FP	FP	FP	FP	FP	FP	FP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	LD	LD	LD	LD	LD	LD	LD	LD	
Exécution	Transfert de données entre FDD et le système.									
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 4 - ECRITURE DE DONNEES EFFACEES
Cette commande diffère de la commande d'écriture normale par le fait qu'elle concerne des données dont le "Data Adress Mark" contient un code disant que ces dernières sont effacées.

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarques
Commande	W	MT	MF	0	0	1	0	0	1	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	W	NS	NS	NS	NS	NS	NS	NS	NS	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	FP	FP	FP	FP	FP	FP	FP	FP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	LD	LD	LD	LD	LD	LD	LD	LD	
Exécution	Transfert de données entre FDD et le système.									
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 5 - LECTURE D'UNE PISTE
Cette commande lit tous les octets de données de la piste entre l'orifice d'indx et FP.

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	MF	SS	0	0	0	1	0	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	W	NS	NS	NS	NS	NS	NS	NS	NS	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	FP	FP	FP	FP	FP	FP	FP	FP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	LD	LD	LD	LD	LD	LD	LD	LD	
Exécution										
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 6 - LECTURE DE ID
Lecture sur la disquette de la prochaine ID possible.

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	MF	0	0	1	0	1	0	Code
	W	x	x	x	x	x	ST	SD1	SD0	
Exécution										
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 7 - FORMATAGE D'UNE PISTE

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	MF	0	0	1	1	0	1	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	NSP	NSP	NSP	NSP	NSP	NSP	NSP	NSP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	TD	TD	TD	TD	TD	TD	TD	TD	
Exécution										
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 8 - VERIFICATION D'UN SECTEUR EGALITE
Cette commande opère une vérification de l'identité ID entre les données écrites et les données à écrire.

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	MT	MF	SS	1	0	0	0	1	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	W	NS	NS	NS	NS	NS	NS	NS	NS	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	FP	FP	FP	FP	FP	FP	FP	FP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	MV	MV	MV	MV	MV	MV	MV	MV	
Exécution	Comparaison de données entre FDD et le système.									
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 9 - VERIFICATION D'UN SECTEUR (plus petit ou égal)

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	MT	MF	SS	1	1	0	0	1	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	W	NS	NS	NS	NS	NS	NS	NS	NS	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	FP	FP	FP	FP	FP	FP	FP	FP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	MV	MV	MV	MV	MV	MV	MV	MV	
Exécution	Comparaison de données entre FDD et le système.									
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 10 - VERIFICATION D'UN SECTEUR (plus grand ou égal).

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	MT	MF	SS	1	1	1	0	1	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NP	NP	NP	NP	NP	NP	NP	NP	
	W	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	W	NS	NS	NS	NS	NS	NS	NS	NS	
	W	LS	LS	LS	LS	LS	LS	LS	LS	
	W	FP	FP	FP	FP	FP	FP	FP	FP	
	W	GL	GL	GL	GL	GL	GL	GL	GL	
	W	MV	MV	MV	MV	MV	MV	MV	MV	
Exécution	Comparaison de données entre FDD et le système.									
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	
	R	ET2	ET2	ET2	ET2	ET2	ET2	ET2	ET2	
	R	NP	NP	NP	NP	NP	NP	NP	NP	
	R	ADT	ADT	ADT	ADT	ADT	ADT	ADT	ADT	
	R	NS	NS	NS	NS	NS	NS	NS	NS	
	R	LS	LS	LS	LS	LS	LS	LS	LS	

COMMANDE 11 - POSITIONNEMENT SUR LA PISTE 0

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	0	0	0	0	1	1	1	Code
	W	x	x	x	x	x	0	SD1	SD0	
Exécution										

COMMANDE 12 - INTERROGATION D'ETAT D'INTERRUPTION
Cette commande détermine les causes d'interruption

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	0	0	0	1	0	0	0	Code
Résultat	R	ET0	ET0	ET0	ET0	ET0	ET0	ET0	ET0	
	R	NPC	NPC	NPC	NPC	NPC	NPC	NPC	NPC	

COMMANDE 13 - SPECIFICATION
Cette commande permet d'adapter différents types de lecteurs au FDC.

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	0	0	0	0	0	1	1	Code
	W	SRT	SRT	SRT	SRT	TDT	TDT	TDT	TDT	
	W	TCT	TCT	TCT	TCT	TCT	TCT	TCT	ND	

COMMANDE 14 - INTERROGATION D'ETAT DES LECTEURS

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	0	0	0	0	1	0	0	Code
	W	x	x	x	x	x	ST	SD1	SD0	
Résultat	R	ET3	ET3	ET3	ET3	ET3	ET3	ET3	ET3	

COMMANDE 15 - RECHERCHE D'UNE PISTE
Cette commande permet le déplacement de la tête.

PHASE	R/W	B7	B6	B5	B4	B3	B2	B1	B0	Remarque
Commande	W	0	0	0	0	1	1	1	1	Code
	W	x	x	x	x	x	ST	SD1	SD0	
	W	NNP	NNP	NNP	NNP	NNP	NNP	NNP	NNP	
Exécution	La tête est positionnée sur la piste recherchée.									

COMMANDE X - COMMANDE INVALIDE Le FDC se met en attente.			
Résultat	R	ET0 ET0 ET0 ET0 ET0 ET0 ET0 ET0	ET0 = 80 en base 16.

Les registres d'état du FDC 765A

Le **PD765A** possède cinq registres d'état : le registre d'état principal qui peut être lu à tout instant ; les registres 0 à 2 à lecture séquentielle et le registre ET3 qui peut être lu par une instruction particulière.

Le registre d'état principal

B0-B3 FDD0-FDD3 (disque occupé).
Dès qu'une instruction de recherche de piste est lancée sur un des lecteurs, le bit correspondant est mis à 1. Aucune instruction de lecture ou d'écriture ne pourra être lancée sur ce lecteur tant que ce bit ne sera pas remis à 0 par la commande 12 (interrogation d'état interruption).

B4 CB (FDC occupé).
Positionné à 1, il signifie que le FDC est en train de traiter une commande de lecture ou d'écriture.

B5 ME (mode exécution).
Ce bit est uniquement positionné en mode non DMA. Dans ce mode, il est égal à 1 lors de la phase exécution et il est égal à 0 lors de la phase résultat.

B6 DES (données entrée/sortie).
Indique le sens de transfert des données entre le FDC et le registre données. Si DES = 0, le transfert s'effectuera du registre de données vers le CPU.

B7 DT (demande de transfert).
Si ce bit est égal à 1, il indique que le registre de données est prêt à transférer ou à recevoir des informations venant du CPU.

Le registre d'état ET0

B7-B6 CI (code interruption).

B7	B6	
0	0	Instruction terminée sans erreur.
0	1	L'exécution de la commande a été lancée, mais elle ne s'est pas terminée avec succès.

B7	B6	
1	0	Commande invalide.
1	1	Exécution abandonnée. Au cours de l'exécution de la commande, le signal ready du lecteur a changé d'état.

B5 F0
Lorsque le FDC a terminé une commande, il positionne ce bit à 1.

B4 VE (vérification d'équipement).
Ce bit signale si une erreur est envoyée au lecteur ou si la piste 0 n'a pas été trouvée lors de la commande 11.

B3 NPT (non prêt).
Lorsque le lecteur n'est pas prêt, ce bit est positionné pour une instruction de lecture/écriture.

B2 ADT (adresse de tête).
Ce bit est utilisé pour indiquer l'état de la tête sélectionnée au moment de l'interruption.

B1-B0 SD1-SD0.
Ces deux bits sont utilisés pour indiquer le lecteur sélectionné au moment de l'interruption.

Registre d'état ET1

B7 FP (fin de piste).
Il est égal à 1 si le FDC essaie d'accéder un secteur après une fin de piste.

B6 Non utilisé. Toujours égal à 0.

B5 ED (erreur donnée).
Ce bit est mis à 1 lors de la détection d'une erreur de CRC.

B4 OR (overrun).
Ce bit est égal à 1 si le transfert de données ne s'est pas déroulé dans un intervalle de temps précis.

B3 Non utilisé. Toujours égal à 0.

B2 ND (pas de données).
Ce bit est positionné si le secteur n'est pas trouvé ou si, lors d'une lecture ID secteur, le FDC ne peut pas lire un champ ID.

B1 NE (pas d'écriture).
Ce bit est positionné si, durant l'exécution d'une commande d'écriture, de lecture ou de formatage, le FDC détecte une protection d'écriture du FDD.

B0 MA (missing adress mark).
Il est égal à 1 si le FDC ne peut détecter l'identificateur ID après une rotation complète du disque ou si le FDC ne peut trouver un "data adress mark".

Registre d'état ET2

B7 Non utilisé. Toujours égal à 0.

B6 CM (control mark).
Ce bit est égal à 1 si le FDC trouve un secteur avec un "data adress mark" effacé.

B5 DD (erreur de données dans champ de données).
Ce bit est égal à 1 lors d'une détection d'erreur CRC dans les champs de données.

B4 PE (piste erronée).
Ce bit est positionné si le FDC constate une différence entre le numéro de la piste lue et le numéro de la piste prévue.

B3 VV (vérification secteur vrai).
Ce bit est égal à 1 si, lors d'une commande de vérification, le résultat est vrai.

B2 VF (vérification secteur faux).
Il est égal à 1 si, lors d'une commande de vérification quelconque (= ; >= ; <=), le résultat est faux.

B1 PD (piste défectueuse).
Ce bit est positionné à 1 si le numéro de piste lu dans ID est différent de celui prévu et que le numéro de piste lu est FF (base 16).

B0 MD (missing ADRESS MARK dans champ de données).
Ce bit est égal à 1 si le FDC ne peut trouver un "data adress mark" effacé ou un "data adress mark" lors d'une lecture.

Registre d'état ET3

B7 FT (faute).
Ce bit est utilisé pour indiquer une erreur signalée par le lecteur.

B6 PE (protection d'écriture).
Ce bit indique si la disquette est protégée.

B5 PT (prêt).
Ce bit est utilisé pour indiquer si la disquette est protégée.

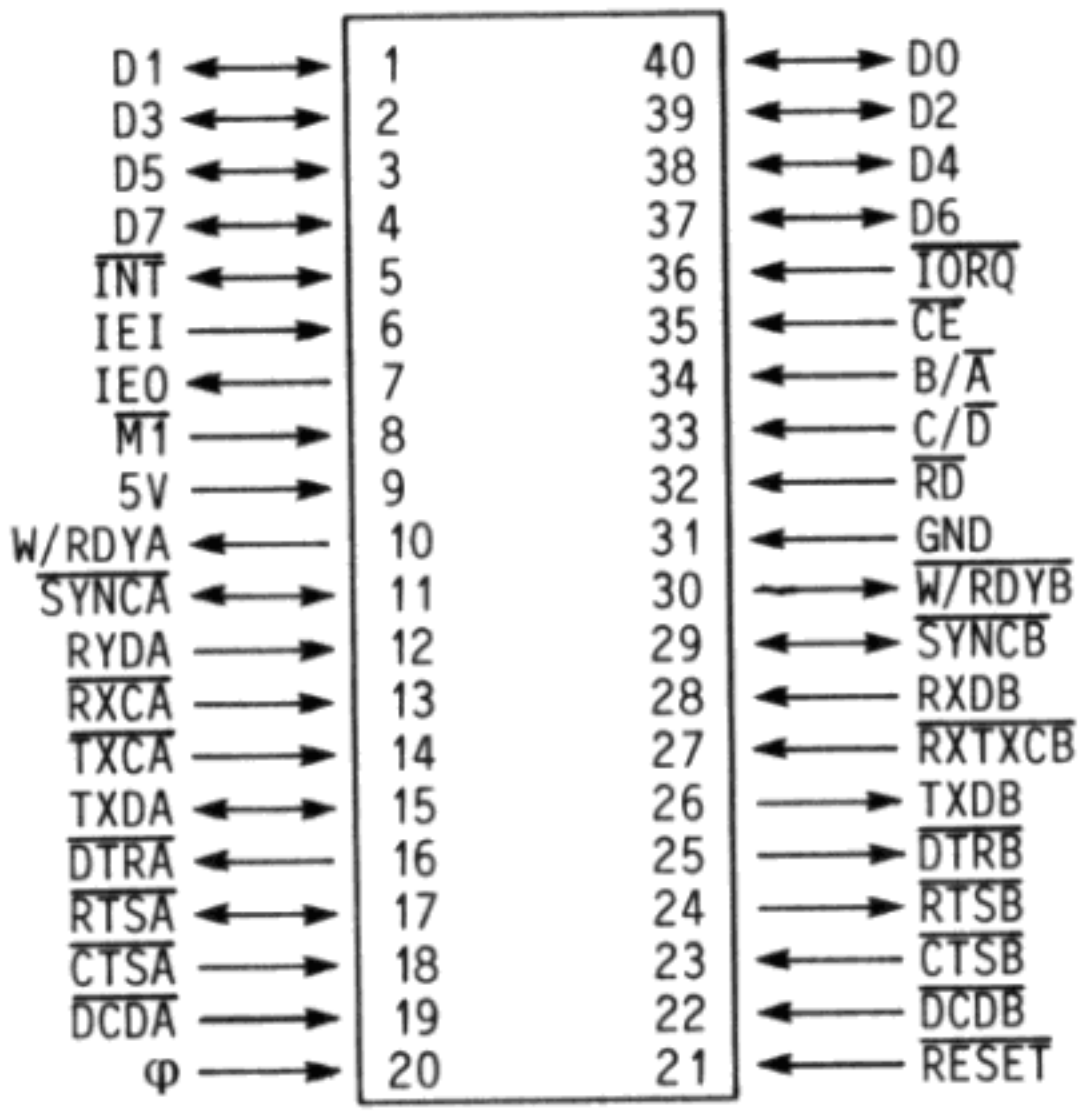
B4 PO (piste 0).
Par ce bit, le lecteur indique que sa tête est positionnée sur la piste 0.

B3 DF (double face).
Il permet de déterminer le type de lecteur (1 tête/2 têtes).

B2 ADT (adresse de tête).
Indique la tête sélectionnée.

B1-B0 SD1-SD0
Ces deux bits indiquent l'unité de disque sélectionnée.

BROCHAGE DU SIO



Nom de broche	Description	Type E=entrée S=sortie B=bidirect.
D0-D7	Bus de données.	B
B/$\overline{A}$	Sélection du canal B ou A.	E
C/$\overline{D}$	Ligne de sélection de commande ou de données.	E
$\overline{CE}$	Sélection du circuit.	E
Φ	Horloge du CPU.	E

BROCHAGE DU SIO

Nom de broche	Description	Type E=entrée S=sortie B=bidorect.
$\overline{M1}$	Cycle machine 1. $\overline{M1}$ et $\overline{RD}$ du SIO = 0 signifie recherche d'un code opération $\overline{M1}$ et $\overline{IORQ}$ du SIO = 0, le Z80 produit un accusé de réception d'interruption.	**S**
$\overline{IORQ}$	Demande d'entrée/sortie.	E
$\overline{RD}$	Lecture	E
$\overline{RESET}$	Arrête les transmissions sur les canaux A et B. Toutes les commandes modem sont mises à l'état logique 1. Les registres internes doivent être à nouveau programmés.	E
IEI	Entrée de validation d'interruption hardware (DAISY CHAIN).	E
IEO	Sortie de validation d'interruption hardware (DAISY CHAIN).	S
$\overline{INT}$	Demande d'interruption.	S
W/RDYA et **W/RDYB**	Fonctions WAIT et READY associées aux canaux A et B. Fonction WAIT sortie drain ouverte. Fonction READY sortie utilisée pour un contrôleur DMA.	S
$\overline{CTSA}$ et **$\overline{CTSB}$**	Clear To Send. Permet au circuit extérieur de démarrer l'opération de transmission.	E
$\overline{DCDA}$ et **$\overline{DCDB}$**	Détection de porteuse canal A et B.	E
$\overline{RTSA}$ et **$\overline{RTSB}$**	Request To Send. Demande d'envoi sur les canaux A et B.	S
$\overline{DTRA}$ et **$\overline{DTRB}$**	Data Terminal Ready. Terminal de données prêt pour les canaux A et B.	S
RXDA et **RXDB**	Entrée de données sur les canaux A et B.	E
TXDA et **TXDB**	Sortie de données sur les canaux A et B.	S

Nom de broche	Description	Type E=entrée S=sortie B=bidirect.
$\overline{\text{RXCA}}$ et **$\overline{\text{RXCB}}$**	Entrées horloge de réception.	E
$\overline{\text{TXCA}}$ et **$\overline{\text{TXCB}}$**	Entrées horloge de transmission. Les horloges peuvent être multipliées par un facteur 1, 16, 32, 64.	E

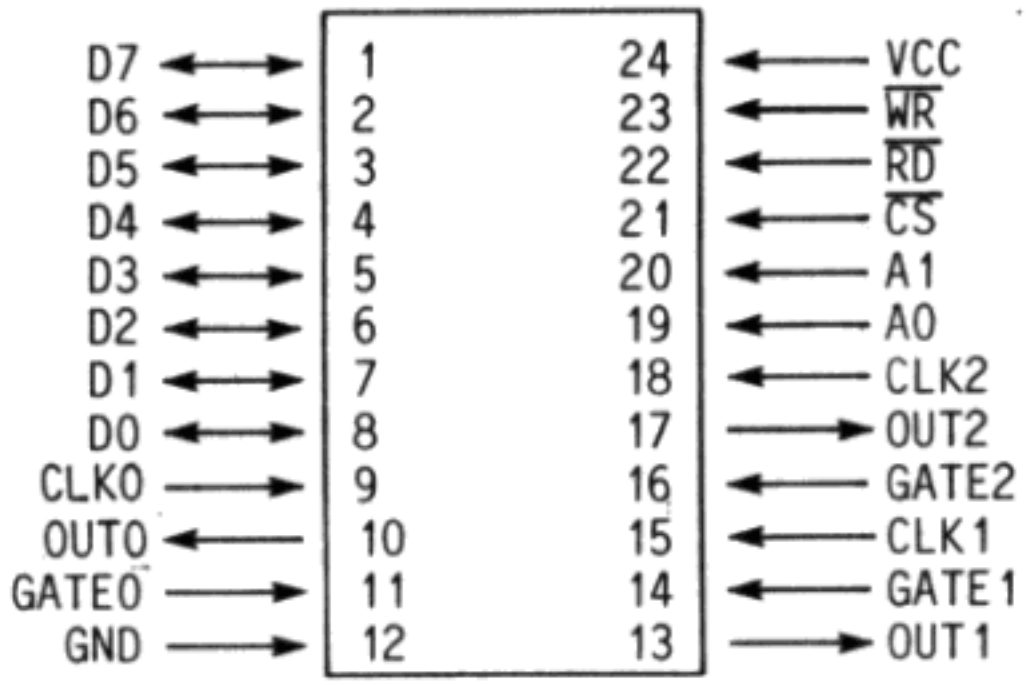
Remarque *:* il y a trois types de SIO : le SIO(0), le SIO(1) et le SIO(2).

Dans le SIO(0), les signaux $\overline{\text{TXCB}}$ et $\overline{\text{RXCB}}$ sont liés entre eux.

Dans le SIO(1), le signal $\overline{\text{DTRB}}$ est supprimé et le signal $\overline{\text{TXRXCB}}$ est dédoublé en $\overline{\text{TXCB}}$ et $\overline{\text{RXCB}}$.

Dans le SIO(2), le signal $\overline{\text{SYNCB}}$ est supprimé et le signal $\overline{\text{TXRXCB}}$ est dédoublé. $\overline{\text{DTRB}}$ est conservé.

BROCHAGE DU 8253

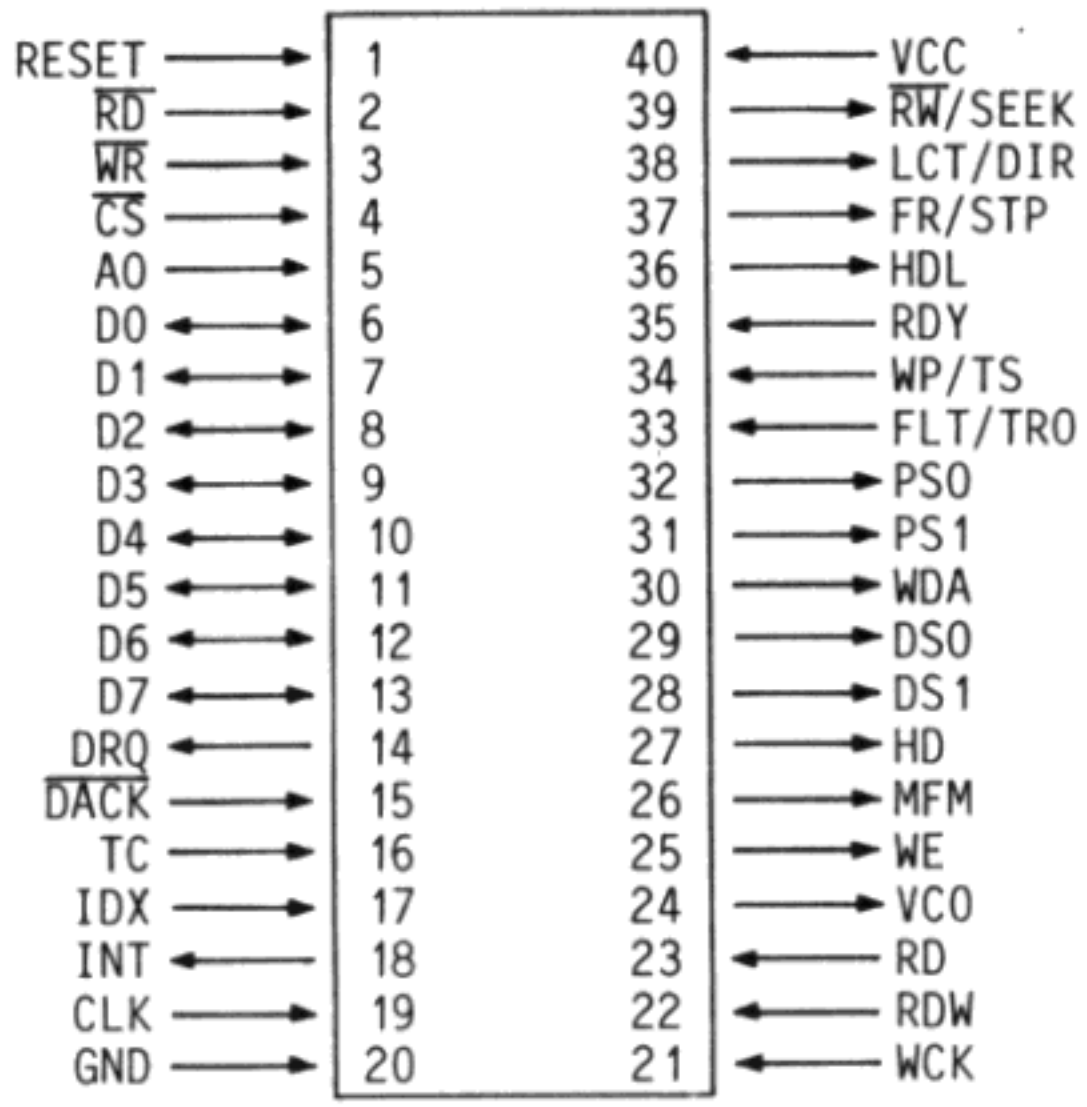


Nom de broche	Description	Type E=entrée S=sortie B=bidirect.
D0-D7	Bus de données.	B
VCC	Alimentation + 5 volts.	E
GND	Masse électrique.	E
$\overline{CS}$	Sélection du circuit (état bas).	E
A0-A1	Bits de sélection d'adresse. Ils permettent de sélectionner un des trois compteurs ou le registre MODE.	E
$\overline{RD}$	Indicateur de lecture. A l'état bas, il signale que le processeur lit le contenu d'un compteur.	E
$\overline{WR}$	Indicateur d'écriture. A l'état bas, il signale que le processeur écrit dans le registre MODE ou dans un compteur.	E
CLK0	Entrée de l'horloge du compteur 0.	E
CLK1	Entrée de l'horloge du compteur 1.	E
CLK2	Entrée de l'horloge du compteur 2.	E
GATE0	Gachette du compteur 0.	E
GATE1	Gachette du compteur 1.	E

BROCHAGE DU 8253

Nom de broche	Description	Type E=entrée S=sortie B=bidirect.
GATE2	Gachette du compteur 2.	E
OUT0	Sortie du compteur 0.	S
OUT1	Sortie du compteur 1.	S
OUT2	Sortie du compteur 2.	S

BROCHAGE DU PD765A



Nom de broche	Description	Type E=entrée S=sortie B=bidirect.
RESET	Place le FDC dans un état déterminé.	E
$\overline{RD}$	Transfert des données du FDC → CPU.	E
$\overline{WR}$	Transfert des données du CPU → FDC.	E
$\overline{CS}$	Sélection du circuit.	E
A0	A0=1, sélection du registre d'état principal. A0=1, sélection du registre de données.	E
D0-D7	Bus de données.	B
DRQ	DMA Request. Demande d'un transfert DMA	S
$\overline{DACK}$	DMA ACKNOWLEDGE. Accusé de réception d'une demande de transfert DMA.	E
TC	Terminal Count. Un niveau haut interrompt le transfert de données du FDC.	E

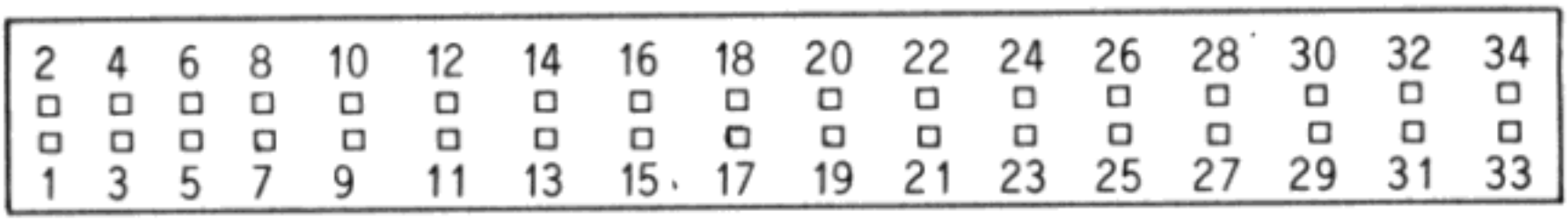
BROCHAGE DU PD765A

Nom de broche	Description	Type E=entrée S=sortie B=bidirect
IDX	Index. Signale au FDC le début physique d'une piste.	E
INT	Demande d'interruption.	S
CLK	Horloge du FDC. 4 MHZ pour les disques 5"1/4, 3"1/2 et 3". 8 MHZ pour les disques 8".	E
GND	Masse.	
WCK	Write Clok (horloge d'écriture). Pour le format MF, fréquence = 500 Khz. Pour le format MFF, fréquence = 1 Mhz.	E
RDW	Read Data Window (fenêtre de lecture de données).	E
RDD	Read Data. Ecriture dans le FDC des données lues sur la disquette.	E
VCO	Signal nécessaire pour le séparateur données.	S
WE	Write Enable. Autorise l'écriture des données sur la disquette.	S
MFM	MFM mode. A l'état haut, le contrôleur travaille en mode MFM et à l'état bas, en mode MF.	S
HD	Head Select. Sélection de tête pour les lecteurs possédant deux têtes de lecture/ écriture.	S
SD0-SD1	Sélection de l'unité disque 0 ou 1.	S
WDA	Write Data. Sortie d'écriture des données du FDC vers la disquette.	S
PS1-PS0	Indique l'état de précompensation à l'écriture en mode MFM. Trois états possibles : EARLY, NORMAL et LATE (plus tôt, normal, plus tard).	S

BROCHAGE DU PD765A

Nom de broche	Description	Type E=entrée S=sortie B=bidirect.
FLT/TRO	FAULT/TRACKO. Signale reçu en cas d'erreur du disque ou cas d'une recherche infructueuse de la piste 0.	E
WP/TS	WRITE PROTECT/TWO SIDE. Signale l'état de protection de la disquette en écriture. Signale le numéro de la face sélectionnée de la disquette en mode recherche de la piste (pour les lecteurs à deux têtes).	E
RDY	READY. Indique que le lecteur est prêt à envoyer ou à recevoir des données.	E
HDL	HEAD LOAD. Commande le chargement et le déchargement de la tête sur la disquette.	S
FR/STP	FIT RESET/STEP. - Exécute une remise à 0 du flip-flop d'erreur en mode lecture/écriture. - Commande des pas du moteur du lecteur en mode de recherche de piste.	S
LCT/DIR	LOW CURRENT/DIRECTION. Indique la direction de déplacement de la tête en mode de recherche de piste. La fonction Low Current permet de diminuer le flot d'écriture des données sur les pistes inférieures.	S
$\overline{RW}$/SEEK	A l'état haut, le FDC indique au lecteur qu'il utilise les signaux relatifs au mode recherche de piste. A l'état bas, il indique qu'il utilise les signaux relatifs au mode écriture/lecture.	S
VCC	+ 5 V.	

CONNECTEUR DU DISQUE SUPPLEMENTAIRE



Toutes les broches paires (2 à 34) sont raccordées à la masse (GND).

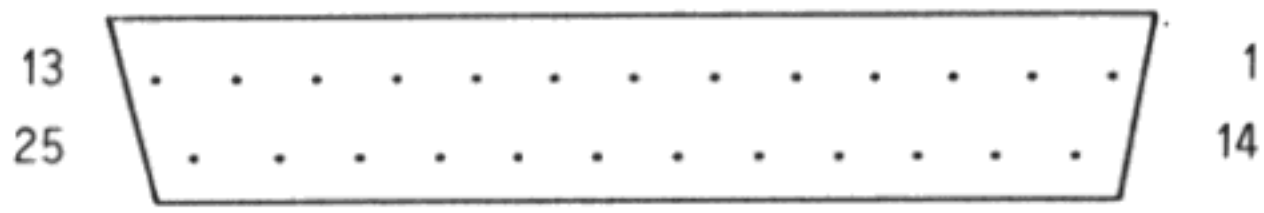
PIN	Nom	Sens du signal
1	READY	LECTEUR → ORDINATEUR
3	SIDE A SELECT	ORDINATEUR → LECTEUR
5	READ DATA	LECTEUR → ORDINATEUR
7	WRITE PROTECT	LECTEUR → ORDINATEUR
9	TRACK 0	LECTEUR → ORDINATEUR
11	WRITE GATE	ORDINATEUR → LECTEUR
13	WRITE DATA	ORDINATEUR → LECTEUR
15	STEP	ORDINATEUR → LECTEUR
17	DIRECTION	ORDINATEUR → LECTEUR
19	MOTOR ON	ORDINATEUR → LECTEUR
21	NON CONNECTE	
23	DRIVE 1 SELECT	ORDINATEUR → LECTEUR
25	NON CONNECTE	
27	INDEX	LECTEUR → LECTEUR
29	NON CONNECTE	
31	NON CONNECTE	
33	NON CONNECTE	

Signal	Fonction
READY	Ce signal est généré par le lecteur lorsqu'il est prêt à recevoir une instruction de lecture ou d'écriture.
SIDE 1 SELECT	Signal de sélection de la tête de lecture utilisé par les lecteurs double-face. Ce signal n'est pas utilisé dans un lecteur standard.
READ DATA	Ligne de transmission des données en série du lecteur vers l'ordinateur.

CONNECTEUR DU DISQUE SUPPLEMENTAIRE

Signal	Fonction
WRITE PROTECT	Signal émis par le lecteur pour signaler que le dispositif de protection en écriture est actif sur la disquette (média).
TRACK 0	Signal émis par le lecteur pour signaler qu'il a atteint la piste de départ (0).
WRITE GATE	Signal émis par l'ordinateur pour signaler qu'une écriture est en cours sur le média.
WRITE DATA	Ligne de transmission des données en série de l'ordinateur vers le lecteur.
STEP	Signal de commande qui fait avancer la tête du lecteur d'une piste dans le sens défini par le signal DIRECTION.
DIRECTION	Signal de sélection de la direction des déplacements de la tête du lecteur (piste par piste).
MOTOR ON	Signal d'activation du moteur de rotation du lecteur.
DRIVE 1 SELECT	Signal de sélection du lecteur 1 (lecteur externe).
INDEX	Signal en provenance du lecteur qui indique la détection du trou d'index présent sur la disquette (média) et par là, la présence effective de la disquette.

CONNECTEUR RS232



Pin	Signal	Nom	Fonction
1	FG	Frame Ground.	Terre générale.
2	TD	Transmit Data.	Emission des données.
3	RD	Receive Data.	Réception des données.
4	RTS	Request To Send.	Demande d'émission.
5	CTS	Clear to Send.	Autorisation d'émission (inverse de RTS).
6	DSR	Data Set Ready.	Signal inverse du DTR.
7	SG	Signal Ground.	Masse des signaux.
8	CD	Carrier Detect.	Détection de porteuse MODEM.
20	DTR	Data Terminal Ready.	Signal habituel de "handshaking".

Les autres signaux ne sont pas utilisés.

Sens des signaux

Pin	Nom	Sens
1	FG	--------- ---------
2	TD	ordinateur → périphérique
3	RD	ordinateur ← périphérique
4	RTS	ordinateur → périphérique
5	CTS	ordinateur ← périphérique
6	DSR	ordinateur ← périphérique
7	GND	--------- ---------
8	CD	ordinateur ← périphérique
20	DTR	ordinateur → périphérique

AMSDOS : RSXS DE LECTURE ET D'ECRITURE SECTEUR

Le petit programme suivant permet d'installer deux nouvelles instructions dans le Basic sous forme de RSXs.

1- Encodez le programme Basic.
2- Sauvez-le sur disque.
3- Lancez-le (RUN).
4- Effacez-le (NEW).
5- Deux nouvelles instructions sont à votre disposition.

¦LECTSEC,piste,secteur,tampon

Cette instruction permet de lire un secteur et d'écrire son contenu en mémoire centrale :

- piste est un nombre compris entre 0 et 39 qui indique le numéro de piste ;
- secteur est un numéro de secteur compris entre 0 et 8 qui indique le secteur logique à lire ;
- tampon représente l'adresse d'une zone mémoire vive disponible de 512 octets où l'on retrouvera le secteur lu.

¦ECRISEC,piste,secteur,tampon

Cette instruction permet d'écrire le contenu d'un tampon présent en mémoire vive (512 octets) dans un secteur spécifié.

Remarque *:* ces instructions fonctionnent uniquement pour les disquettes au format système ou données.

Vous trouverez le listing de la source assembleur de la routine à la suite du programme Basic.

Programme Basic

```
10 REM installe RSX LECTSEC ECRISEC
20 MEMORY &9FFF
30 FOR I=&A000 TO &A08F
```

AMSDOS : RSXS DE LECTURE ET D'ECRITURE SECTEUR

```
40 READ A$
50 POKE I,VAL("&"+A$)
60 NEXT I
70 CALL &A000
80 DATA 01,0A,A0,21,8E,A0,CD,D1,BC,C9,12,A0,C3,21,A0,
C3,5B,A0,4C,45,43,54,53,45,C3,45,43,52,49,53,45,C3,00,
F5,AF,32,8D,A0,F1,FE,03,20,38,DD,7E,00,DD,56,02,DD,6E,
04,DD,66,05,C6,41,1E,00,F5,0E,07,CD,0F,B9,F1,C5,4F,F5,
3A,8D,A0,FE,00
90 DATA 28,06,F1,CD,4E,C6,18,04,F1,CD,66,C6,C1,CD,18,
B9,C9,F5,AF,3C,32,8D,A0,18,C3,21,70,A0,7E,FE,00,C8,CD,
5A,BB,23,18,F6,4E,4F,4D,42,52,45,20,44,27,41,52,47,55,
4D,45,4E,54,53,20,49,4E,43,4F,52,52,45,43,54,00,00,00,
00
```

Listing Assembleur

```
Hisoft GENA3.1 Assembler. Page      1.

Pass 1 errors: 00

                        10 ; RSX LECTSEC & ECRISEC
                        20 ; D. MARTIN LIEGE 1985
                        30 ;
                        40 ; |LECSECT,TAMPON,PISTE,SECTEUR
                        50 ; |ECRISEC,TAMPON,PISTE,SECTEUR
                        60 ;
BCD1                    70 INIRSX: EQU  #BCD1
BB5A                    80 PRINT:  EQU  #BB5A
B90F                    90 SELROM: EQU  #B90F
B918                   100 DESROM: EQU  #B918
C666                   110 LECT:   EQU  #C666
C64E                   120 ECRIT:  EQU  #C64E
A000                   130         ORG  #A000
A000  010AA0           140         LD   BC,COMEXT
A003  218EA0           150         LD   HL,TAMPON
```

```
A006  CDD1BC      160          CALL INIRSX
A009  C9          170          RET
A00A  12A0        180 COMEXT:  DEFW TABLE
A00C  C321A0      190          JP   LSEC
A00F  C35BA0      200          JP   ESEC
A012  4C454354    210 TABLE:   DEFM "LECTSE"
A018  C3          220          DEFB "C"+#80
A019  45435249    230          DEFM "ECRISE"
A01F  C3          240          DEFB "C"+#80
A020  00          250          DEFB 00
A021  F5          260 LSEC:    PUSH AF
A022  AF          270          XOR  A
A023  328DA0      280          LD   (TYPE),A
A026  F1          290 MAIN:    POP  AF
A027  FE03        300          CP   3
A029  2038        310          JR   NZ,ERREUR
A02B  DD7E00      320          LD   A,(IX+0)
A02E  DD5602      330          LD   D,(IX+2)
A031  DD6E04      340          LD   L,(IX+4)
A034  DD6605      350          LD   H,(IX+5)
A037  C641        360          ADD  A,#41
```

Hisoft GENA3.1 Assembler. Page 2.

```
                  365 *E
A039  1E00        370          LD   E,0
A03B  F5          380          PUSH AF
A03C  0E07        390          LD   C,7
A03E  CD0FB9      400          CALL SELROM
A041  F1          410          POP  AF
A042  C5          420          PUSH BC
A043  4F          430          LD   C,A
A044  F5          440          PUSH AF
```

AMSDOS : RSXS DE LECTURE ET D'ECRITURE SECTEUR

```
A045  3A8DA0      450          LD    A,(TYPE)
A048  FE00        460          CP    0
A04A  2806        470          JR    Z,LIT
A04C  F1          480          POP   AF
A04D  CD4EC6      490          CALL  ECRIT
A050  1804        500          JR    SUITE
A052  F1          510 LIT:     POP   AF
A053  CD66C6      520          CALL  LECT
A056  C1          530 SUITE:   POP   BC
A057  CD18B9      540          CALL  DESROM
A05A  C9          550          RET
A05B  F5          560 ESEC:    PUSH  AF
A05C  AF          570          XOR   A
A05D  3C          580          INC   A
A05E  328DA0      590          LD    (TYPE),A
A061  18C3        600          JR    MAIN
A063  2170A0      610 ERREUR:  LD    HL,MESERR
A066  7E          620 ENVOI:   LD    A,(HL)
A067  FE00        630          CP    0
A069  C8          640          RET   Z
A06A  CD5ABB      650          CALL  PRINT
A06D  23          660          INC   HL
A06E  18F6        670          JR    ENVOI
A070  4E4F4D42    680 MESERR:  DEFM  "NOMBRE D'ARGUMENTS
A083  494E434F    690          DEFM  "INCORRECT"
A08C  00          700          DEFB  0
A08D              710 TYPE:    DEFS  1
A08E              720 TAMPON:  DEFS  4
```

AMSDOS : ANALYSE DE L'EN-TETE D'UN FICHIER

Ce programme permet d'analyser l'en-tête d'un programme sur disque ou sur cassette et affiche son type (BASIC, BASIC PROTEGE, FICHIER SEQUENTIEL ou IMAGE BINAIRE), son adresse d'implantation, sa longueur et son point d'entrée éventuel (IMAGE BINAIRE).

Les informations fournies par ce programme doivent vous permettre une recopie aisée sur disque des programmes binaires contenus sur cassette.

Remarque *:* si la lecture se fait sur disque, un nom de fichier est indispensable. Sur cassette, le nom de fichier à lire est optionnel.

Programme Basic

```
10 REM LECTURE DU HEADER
20 REM SUR DISQUE OU CASSETTE
30 REM DANIEL MARTIN LIEGE 1985
40 REM
50 MEMORY &8FFF
60 FOR i=&A000 TO &A011
70 READ a$
80 POKE i,VAL("&"+a$)
90 NEXT i
100 CLS
110 INPUT"source Cassette ou Disque (C/D) ";s$
120 s$=UPPER$(s$)
130 IF s$="C" THEN |TAPE.IN : F=1:GOTO 170
140 IF s$<>"D" THEN GOTO 100
150 F=0
160 |DISC
170 PRINT
180 N$=""
190 INPUT"nom du fichier a lire ";n$
200 IF N$="" AND F=0 THEN PRINT"SUR DISQUE UN NOM EST
INDISPENSABLE":GOTO 170
210 l=LEN(n$)
```

AMSDOS : ANALYSE DE L'EN-TETE D'UN FICHIER

```
220 POKE &A200,L
230 FOR I=1 TO L
240 POKE &A200+I,ASC(MID$(n$,i,1))
250 NEXT i
260 CALL &A000
270 PRINT:PRINT
280 HEADER=PEEK(&A220)+256*PEEK(&A221)
290 PRINT"ADRESSE DU HEADER ";HEX$(HEADER,4)
300 PRINT
310 TYPE=PEEK(HEADER+18)
320 PRINT "TYPE :";type
330 PRINT
340 IF TYPE=0 THEN PRINT"PROGRAMME BASIC NORMAL"
350 IF TYPE=1 THEN PRINT"PROGRAMME BASIC PROTEGE"
360 IF TYPE=2 THEN PRINT"FICHIER BINAIRE"
370 IF TYPE=22 THEN PRINT"FICHIER SEQUENTIEL"
380 ADC=PEEK(HEADER+21)+256*PEEK(HEADER+22)
390 PRINT
400 PRINT "ADRESSE CHARGEMENT :";HEX$(ADC,4)
410 LG=PEEK(HEADER+24)+256*PEEK(HEADER+25)
420 PRINT
430 PRINT "LONGEUR : ";HEX$(LG,4)
440 PRINT
450 PE=PEEK(HEADER+26)+256*PEEK(HEADER+27)
460 PRINT"POINT D'ENTREE : ";HEX$(PE,4)
470 DATA 21,00,A2,46,23,11,00,90,CD,77,BC,22,20,A2,CD,
7A,BC,C9
```

AMSDOS : CHARGEMENT DES FICHIERS OPTION P

Le programme suivant permet de charger en mémoire centrale des fichiers préalablement sauvés avec l'option P.

Le fichier à charger peut se trouver sur disque ou sur cassette.

Remarque *:* si le fichier cassette est trop long, vous éprouverez quelques problèmes de chargement malheureusement insolubles.

- Lancez le programme. Celui-ci s'installe puis s'efface de la mémoire.
- Tapez CALL £A400 et le chargement commence.

NOTE IMPORTANTE : ce programme est proposé en version 664 - 6128, les possesseurs de 464 doivent modifier la valeur : 66 de la ligne 230 en 83.

Vous trouverez, à l'issue du programme Basic, la source de la routine en langage machine contenue dans ce programme.

Programme Basic

```
10 REM LECTURE PROGRAMME BASIC OPTION P
20 REM SUR DISQUE OU CASSETTE
30 REM DANIEL MARTIN LIEGE 1985
40 REM
50 MEMORY &9FFF
60 CLS
70 INPUT"source Cassette ou Disque (C/D) ";s$
80 s$=UPPER$(s$)
90 IF s$="C" THEN |TAPE.IN : l=0 :GOTO 180
100 IF s$<>"D" THEN GOTO 60
110 |DISC
120 PRINT
130 INPUT"nom du fichier ";n$
140 l=LEN(n$)
150 FOR i=1 TO l
160 POKE &A430+I,ASC(MID$(n$,i,1))
170 NEXT i
180 FOR i=&A400 TO &A428
```

AMSDOS : CHARGEMENT DES FICHIERS OPTION P

```
190 READ a$
200 POKE i,VAL("&"+a$)
210 NEXT i
220 POKE &A401,1
230 DATA 06,00,21,31,a4,11,00,a0,cd,77,bc,30,18,c5,21,
70,01,cd,83,bc,c1,21,70,01,09,eb,21,66,ae,06,04,73,23,
72,23,10,fa,cd,7a,bc,c9
240 CLS
250 PRINT"tapez CALL &A400"
260 NEW
```

Listing Assembleur

```
Hisoft GENA3.1 Assembler. Page      1.

Pass 1 errors: 00

                        10 ; ROUTINE INTERNE DU PROGRAMME
                        20 ; LECBASP.BAS
                        30 ; DANIEL MARTIN LIEGE 1985
                        40 ;
A400                    50          ORG   #A400
0170                    60 BASIC:   EQU   #170
AE66                    70 FINBAS:  EQU   #AE66
                        80 ;        EQU   #AE83 POUR 464
                        90 ;        EQU   #AE66 POUR 664-6128
A431                   100 NOMFIC:  EQU   #A431
A000                   110 TAMPON:  EQU   #A000
BC77                   120 OPEN:    EQU   #BC77
BC83                   130 READ:    EQU   #BC83
BC7A                   140 CLOSE:   EQU   #BC7A
A400   0600             150          LD    B,0
A402   2131A4           160          LD    HL,NOMFIC
A405   1100A0           170          LD    DE,TAMPON
```

```
A408  CD77BC      180          CALL  OPEN
A40B  3018        190          JR    NC,FIN
A40D  C5          200          PUSH  BC
A40E  217001      210          LD    HL,BASIC
A411  CD83BC      220          CALL  READ
A414  C1          230          POP   BC
A415  217001      240          LD    HL,BASIC
A418  09          250          ADD   HL,BC
A419  EB          260          EX    DE,HL
A41A  2166AE      270          LD    HL,FINBAS
A41D  0604        280          LD    B,4
A41F  73          290 BOUCLE:  LD    (HL),E
A420  23          300          INC   HL
A421  72          310          LD    (HL),D
A422  23          320          INC   HL
A423  10FA        330          DJNZ  BOUCLE
A425  CD7ABC      340 FIN:     CALL  CLOSE
A428  C9          350          RET
```

AMSDOS : SUPERZAP

Ce programme est relativement important (160 lignes). Cependant, il vous aidera à faire des miracles.

Les applications en sont infinies :

- Lecture de secteurs endommagés.
- Réparation des répertoires (DIRECTORY).
- Analyse de la structure du système d'exploitation.
- Copie de secteurs.
- Modification des messages systèmes.
- ...

L'utilisation du programme est très simple.

A l'initialisation, un menu apparaît en bas d'écran. Dix options se présentent à vous. La sélection se fait en poussant la touche correspondante sur le clavier numérique.

F0 - Lecture d'un secteur dans le tampon

A l'issue de cette commande, le système vous demande le numéro de piste (0 à 39) et le numéro de secteur (0 à 8) à lire. Le secteur est transféré dans le tampon et les 256 premiers octets du secteur (qui en comporte 512) sont affichés en hexadécimal et en ASCII à l'écran.

F1 - Ecriture secteur

A l'issue de cette commande, le système vous demande le numéro de piste et le numéro de secteur où il doit écrire le contenu du tampon.

***** ATTENTION CETTE COMMANDE PEUT ENDOMMAGER LE *****
***** CONTENU DE VOTRE DISQUETTE *****

F2

FE - Modification du tampon

A l'issue de cette commande, le système se positionne en mode modification et permet de modifier directement le contenu du tampon en ASCII ou en HEXA.

Pour **sortir** du mode modification, appuyez sur **FB.** F8

La modification n'est effective que dans le tampon. Pour l'enregistrer de façon définitive, vous devez utiliser l'option d'écriture secteur (F1).

En cours de modification, vous pouvez utiliser les fonctions F3 et F4. Les autres options (F0, F1, F5, F6, F7 et F9) sont inopérantes.

F3 - Commutation du mode modification en ASCII ou en HEXA

Cette option n'est effective que lors du choix de l'option modification (F2).

F4 - Commutation de la première partie du secteur (00 à FFH) vers la seconde (100H à 1FFH) et vice versa.

F5 - Lecture du secteur suivant.

F6 - Lecture du secteur précédent.

F7 - Remplissage du tampon (512 octets) avec une valeur constante (0 à 255).

F8 - Sortie du mode modification.

F9 - Sortie du programme.

Vous devez utiliser cette fonction pour sortir du programme car elle reprogramme le contenu correct des touches de fonction.

Programme Basic

```
10 REM programme superzap
20 REM (c) Daniel Martin
30 REM Liege novembre 1985
40 REM
50 MEMORY &9FFF
60 DEFINT a-z
70 BUF=&A300
80 GOSUB 1540
90 FL=1
```

AMSDOS : SUPERZAP

```
100 MODE 2
110 LOCATE 12,1
120 PRINT"* * *   S U P E R Z A P   d e   D.  M A R
T I N  * * *"
130 LOCATE 12,2
140 PRINT STRING$(55,CHR$(95))
150 LOCATE 1,24
160 PRINT"F0 = Lecture  , F1 = Ecriture , F2 = Modi
fier , F3 = ASCII/HEX , F4 = P1 <-> P2"
170 PRINT"F5 = Sect +1  , F6 = Sect -1  , F7 = Remplir
, F8 = STOP/MODI , F9 = termine"
180 LOCATE 54,4:PRINT"PISTE"
190 LOCATE 54,8:PRINT"SECT."
200 LOCATE 54,12:PRINT"MODE"
210 LOCATE 54,13:PRINT"HEXA "
220 LOCATE 54,14:PRINT" P1"
230 FOR i=0 TO 9
240 KEY i,CHR$(i+1)
250 NEXT i
260 LOCATE 1,22 :PRINT STRING$(78," "):LOCATE 1,22: PR
INT"COMMANDE ?";
270 A$=INKEY$
280 CALL &BB81
290 IF a$="" THEN GOTO 270
300 CALL &BB84
310 a=ASC(a$)
320 IF a>10 THEN GOTO 270
330 ON a GOSUB 410,750,790,1280,1220,1330,1410,1450,
260,350
340 GOTO 260
350 REM termine
360 CLS
370 FOR i=0 TO 9
380 KEY i,CHR$(i+48)
```

```
390 NEXT i
400 STOP
410 REM LECTURE SECTEUR
420 POKE &A015,&66
430 GOSUB 450
440 GOTO 610
450 LOCATE 1,22
460 PRINT STRING$(78," ")
470 LOCATE 1,22
480 INPUT"PISTE :";PT$
490 PT=VAL(PT$)
500 IF PT<0 OR PT>39 THEN GOTO 450
510 LOCATE 55,5:PRINT PT;" "
520 LOCATE 20,22
530 INPUT"SECTEUR :";SC$
540 SC=VAL(SC$)
550 IF SC<0 OR SC>8 THEN GOTO 520
560 LOCATE 55,9:PRINT sc
570 DO=PT*256+SC+65
580 CALL &A000,DO
590 LOCATE 1,22:PRINT STRING$(78," ")
600 RETURN
610 DP=0
620 Y=3 : X=1
630 FOR I=0 TO 255
640 IF I MOD 16 = 0 THEN Y=Y+1 : X=1 : X1=60 : LOCATE
X,Y :
AD$=HEX$(I+DP,3):PRINT AD$:X=X+2
650 X=X+3 : X1=X1+1
660 VL=PEEK(BUF+DP+I)
670 VL$=HEX$(VL,2)
680 IF v1>31 THEN AS$=CHR$(VL) ELSE as$="."
690 LOCATE X,Y
```

```
700 PRINT VL$
710 LOCATE X1,Y
720 PRINT AS$
730 NEXT I
740 RETURN
750 REM ECRITURE SECTEUR
760 POKE &A015,&4E
770 GOSUB 450
780 RETURN
790 REM MODIFIER SECTEUR
800 X=0:Y=0
810 IF FL=1 THEN LOCATE 6+X*3,4+Y
820 IF FL=-1 THEN LOCATE 61+X,4+Y
830 Q$=INKEY$
840 CALL &BB81
850 IF Q$="" THEN GOTO 830
860 Q=ASC(Q$)
870 IF Q=243 AND X<15 THEN X=X+1 : GOTO 810
880 IF Q=243 AND X=15 THEN X=0:Y=Y+1
890 IF Y=16 THEN Y=0
900 IF Q=242 AND X>0 THEN X=X-1
910 IF Q=240 AND Y>0 THEN Y=Y-1
920 IF Q=241 AND Y<15 THEN Y=Y+1
930 IF Q=9 THEN CALL &BB84 :RETURN
940 IF Q=4 THEN GOSUB 1280 : GOTO 810
950 IF Q=5 THEN CALL &BB84:GOSUB 1220:GOTO 800
960 IF FL=1 THEN GOTO 1050
970 IF Q<32 OR Q>127 THEN GOTO 810
980 PRINT Q$
990 QH$=HEX$(Q)
1000 LOCATE 6+X*3,4+Y
1010 PRINT QH$
1020 LOCATE 61+X,4+Y
```

```
1030 POKE BUF+DP+X+Y*16,Q
1040 Q=243 : GOTO 870
1050 REM MODIFIER EN HEXA
1060 IF Q>70 THEN Q=Q AND 223
1070 IF Q<48 OR Q>70 OR (Q>57 AND Q<65) THEN GOTO 810
1080 PRINT CHR$(Q)
1090 LOCATE 7+X*3,Y+4
1100 Q2$=INKEY$
1110 IF Q2$="" THEN GOTO 1100
1120 Q2=ASC(Q2$)
1130 IF Q2>70 THEN Q2=Q2 AND 223
1140 IF Q2<48 OR Q2>70 OR (Q2>57 AND Q2<65) THEN GOTO
1100
1150 PRINT CHR$(Q2)
1160 VH=VAL("&"+Q$+Q2$)
1170 IF VH>31 THEN QA$=CHR$(VH) ELSE QA$="."
1180 LOCATE 61+X,Y+4
1190 PRINT QA$
1200 POKE BUF+DP+X+Y*16,VH
1210 Q=243:GOTO 870
1220 REM P1 <-> P2
1230 IF DP=0 THEN DP=256 ELSE DP=0
1240 GOSUB 620
1250 LOCATE 55,14
1260 IF dp=0 THEN PRINT"P1" ELSE PRINT"P2"
1270 RETURN
1280 REM ascii hexa
1290 FL=-FL
1300 LOCATE 54,13
1310 IF FL=1 THEN PRINT"HEXA " ELSE PRINT"ASCII"
1320 RETURN
1330 REM SECT+1
1340 SC=SC+1
```

```
1350 IF SC=9 THEN SC=0 : PT=PT+1
1360 IF PT=40 THEN PT=0
1370 POKE &A015,&66
1380 LOCATE 55,5:PRINT PT;" "
1390 GOSUB 560
1400 GOTO 610
1410 SC=SC-1
1420 IF SC=-1 THEN SC=8 : PT=PT-1
1430 IF PT=-1 THEN PT=0 : SC=0
1440 GOTO 1370
1450 REM REMPLIR AVEC PATTERN
1460 LOCATE 1,22
1470 PRINT STRING$(78," ")
1480 LOCATE 1,22
1490 INPUT "VALEUR DECIMALE ";VD$
1500 VD=VAL(VD$)
1510 IF VD<0 OR VD>255 THEN GOTO 1460
1520 CALL &A01C,VD
1530 GOTO 620
1540 REM routines langage machine
1550 FOR i=&A000 TO &A02B
1560 READ v$
1570 POKE i,VAL("&"+v$)
1580 NEXT i
1590 DATA DD,7E,00,DD,56,01,1E,00,21,00,A3,F5,0E,07,
CD,0F,B9,F1,C5,4F,CD,66,C6,C1,CD,18,B9,C9
1600 DATA DD,7E,00,21,00,A3,11,01,A3,01,00,02,77,ED,
B0,C9
1610 RETURN
```

AMSDOS : OUVERTURE FANTOME D'UN FICHIER

La gestion des variables alphanumériques en Basic produit un morcellement important de la mémoire centrale. Lors de l'ouverture d'un fichier, le système procède à une restructuration de la mémoire. Cette opération est relativement longue.

Pour éviter des périodes d'attente toujours désagréables, il suffit de réserver préalablement un tampon pour le fichier par l'intermédiaire d'une ouverture fictive.

De plus, l'utilisation de la commande **SYMBOL AFTER** modifie le pointeur **HIMEM** et, par là, la structure de la mémoire. Il est donc nécessaire de faire suivre cette commande de la séquence ci-dessous :

```
10 OPENOUT "FICTIF"
20 MEMORY HIMEM-1
30 CLOSEOUT
```

Ligne 10 : le fichier "FICTIF" est ouvert.

Ligne 20 : la mémoire disponible au Basic est modifiée par l'instruction MEMORY et le tampon du fichier "FICTIF" ne fait donc plus partie de la mémoire disponible.

Ligne 30 : l'instruction CLOSEOUT libère le tampon, mais ne rend pas l'espace au Basic.

CP/M - UTILITAIRE DE RAPPEL DU PROGRAMME COURANT

Pour CP/M 2.2

Ce petit utilitaire très simple à construire vous permettra de relancer un programme déjà en mémoire après une sortie involontaire.

Exemple : vous utilisez MBASIC et, après l'encodage d'un programme relativement long, vous tapez SYSTEM par erreur. Cette commande produit un retour au CP/M. Si vous relancez MBASIC, le tampon s'initialisera et vous aurez perdu le contenu de votre programme. L'utilisation de OUF vous permettra de retourner au Basic sans rien modifier.

Construction du programme OUF.COM

C'est très simple, il suffit d'écrire :

```
SAVE 0 OUF.COM
```

Le programme OUF.COM existe alors dans votre répertoire.

Remarque *:* le programme n'occupe pas de place sur disque.

Pour CP/M 3.0

Pour transformer votre **CP/M** en **AZERTY**, il suffit d'utiliser l'utilitaire **SETKEYS** en conjonction avec le fichier KEYS.AZE que vous construirez avec l'éditeur de la façon suivante :

```
ED KEYS.AZE

71 N "w"
71 S "W"
71 C "↑W"
67 N "a"
67 S "A"
67 C "↑A"
59 N "z"
59 S "Z"
59 C "↑Z"
69 N "q"
69 S "Q"
69 C "↑Q"
29 N "m"
29 S "M"
29 C "↑M"
38 N ":"
38 S "★"
38 C ":"
```

La commande SETKEYS KEYS.AZE commutera le clavier en AZERTY.

CP/M : MISE EN ROUTE AUTOMATIQUE DE L'IMPRIMANTE

Pour CP/M 2.2

La réalisation des deux programmes suivants vous permettra de disposer de deux nouvelles commandes **AVECIMP.COM** et **SANSIMP.COM** qui permettent de mettre en route et d'inhiber l'imprimante pendant une procédure BATCH (SUBMIT), par exemple.

Réalisation de AVECIMP.COM

a - appelez le DDT.
b - tapez A 100.
c - encodez le programme suivant.

```
    LHLD 1
    LXI D,9
    DAD D
    MVI M,CD
    RET
    retour chariot
```

d - tapez CTRL C.
e - tapez SAVE 1 AVECIMP.COM.

Réalisation de SANSIMP.COM

Répétez les points a et b.

c - encodez le programme suivant :

```
    LHLD 1
    LXI D,9
    DAD D
    MVI M,C3
    RET
    retour chariot
```

d - tapez CTRL C.
e - tapez SAVE 1 SANSIMP.COM.

LOGO : PROGRAMMATION DES NOTES MUSICALES

Voici une procédure **LOGO** qui permet de mémoriser les notes de l'octave médian dans une variable locale. Elle associe une période de tonalité à chaque nom de note.

```
?to periode :note
>local "octave
>make "octave [DO 478 DO# 451 RE 426 RE# 402 MI 379 FA 358
FA# 338 SOL 319 SOL# 301 LA 284 LA# 268 SI 253]
>op cherche :octave :note
>end
```

Définition de la procédure cherche

```
?to cherche :l :n
>if emptyp :l [op 0]
>if = :n first :l [op item 2 :l]
>op cherche bf bf :l :n
>end
```

ainsi ?periode "FA fournira 358.

Vous pourrez utiliser :

```
?sound (se 1 pediode "FA 200)
```

SCHEMA GUIDE

Pour savoir de quel modèle AMSTRAD traite chaque chapitre, vous pouvez consulter le schéma-guide suivant :

Matériels	**464/664**	**6128**	**8256**
Chapitre I	oui	oui	non
Chapitre II	oui	oui	non
Chapitre III	oui	oui	non
Chapitre IV	non	oui	oui
Chapitre V	oui	oui	oui
Chapitre VI	oui	oui	oui
Chapitre VII	oui	oui	oui
Chapitre VIII	seulement p. 197 p. 201 p. 203 p. 206 p. 213 p. 214 p. 216 P. 217	oui	seulement p. 214 p. 215 p. 216 p. 217

INDEX

Remarques *:* l'index est classé par ordre alphabétique. Au sein d'une même lettre, les mots en majuscules sont classés en premier et suivi des mots en minuscules. Les **RSX** (mots précédés du signe ¦) sont classés à la lettre qui suit le signe ¦.

$$$ 13,54
8253 32,153,166,188
* / + - 131
< > = 131
.APV 130
.DEF 130
.EMT 148
.ENL 148
.PRM 130
.REM 148
.contents 130
.deposit 130
.examine 130
.in 148
.out 148

¦A 17,34
AMSDOS.COM 122
ASM 54,69
ATT 110
AUXIN 83,90,95,96
AUXOUT 83,90,95,96
AZERTY 215
and 131
arctan 148
ascii 131

INDEX

¦B 17,34
BAK 13,54
BANK 87
¦BANKFIND 24
BANKING 79
BANKMAN 22
¦BANKOPEN 23
¦BANKREAD 24
¦BANKWRITE 23
BAS 13,54
¦BASIC 17
BAT 43
BATCH 75
BDOS 40,42,47,50,52,59,67,80,81,82,83,84,85,87,88,94,102,104
BIN 13,54
BIOS 40,41,42,45,55,57,80,81,82,90,104
BOOT 32,42,50,55,59,77,90,94,97
BUFFER FULL 32
BUSY 32
bf 131
bk 131
bl 131
buttonp 131

CAS... 34,36,37
CAT 14
CATALOG 34
CBIOS 52
CCB 107,108
CCP 40,41,42,50,67,69,80,83,84,103
CHAIN 14
CHAIN MERGE 14
CHKDISC 77
CLOAD 77
CLOSEIN 14
CLOSEOUT 14,213
COLD BOOT 42,82
CONFIGURATION 50
CONIN 32,55,83,90
CONNECTEURS 193,195
CONOUT 32,55,83,90
CONST 55,90
COPYDISC 77
COPYSYS 109
¦CPM 17,32
CPMLDR 82
CRC 153,154,156,164,162

CSAVE 77
catch 132
changef 148
char 132
clean 132
co 132
copyoff 148
copyon 148
cos 132
count 133
cs 133
ct 133
cursor 148

DATE 109,110
DDT 69
DEVICE 110
DEVTBL 92
DIR 67,110
¦DIR 12,17,34
DIRECTORY 12,28,51
DIRSYS 109,111
¦DISC 18,34
¦DISC.IN 18,34
¦DISC.OUT 18,34
DISCCHK 77
DISCOPY 78
DISCKIT2 78
DISCKIT3 122
DMA 63,65,84,99,101,103
DPB 33,37,46,64,86,87
DPH 33,35,36,37,46,86
¦DRIVE 17,18,34,110
DRIVER 43
DRIVERS.GSX 127
DRLKEYS 78
DRVTBL 92
DUMP 71,111
defaultd 148
define 148
dir 133
dirpic 148
dot 133
dotc 149

ECRISEC 197
ED 71,111

INDEX

EOF 14,73
ERA 67
¦ERA 18
ERASE 109,111
ERRACT 134
EXCLUDE 111
ed 133
edall 149
edf 149
emptyp 133
end 134
ent 134
env 134
er 134
erall 149
ern 134
error 135

FALSE 135
FCB 52,64,65,66,84,88,97,98,99,100,
101,102,104,105,106,108
FDC 33,38,171,190
FDOS 40
FF 111
FILECOPY 78
FLUSH 92
FORMAT 78
FULL 111
FWRESET 78
fd 135
fence 135
fill 149
first 135
fput 136
fs 136

GENCOM 83,112
GENGRAPH 127
GET 112
GSX 127
glist 136
go 136
gprop 136

HDLC 153,160
HELP 113

HEX 54,74,113
HOME 55,90
home 149
ht 136

IMP.COM 216
INITDIR 113
INP 73
INPUT 15
INT 54
if 136
int 137
item 137

JOCKER 24

keyp 137

LANGUAGE 122
LECTSEC 197
LENGHT 111
LIB 113
LINE INPUT 15
LINK 87,114
LIST 15,43,55,60,73,82,90,95
LISTST 56,91
LISTOUT 83
LDAD 15,72
LOADER 83
LOGO 35,129
LPT 43
label 137
last 149
lc 149
list 137
listp 149
load 137
loadpic 149
local 137
lput 149
it 138

MAC 115
MERGE 15,74

INDEX

MODE 167,168,169
MOVCPM 43,72
make 138
memberp 149

NUL 73
NOPAGE 111
NOSORT 111
namep 150
nodes 138
noformat 150
not 138
nowatch 150
numberp 150

OPENIN 15,34,36
OPENOUT 16,34,36
OUF.COM 214
OUT 73
OVERLAY 104
op 138

P 203
PALETTE 122
PATCH 115
PERLOG-PERPHY 75
PIP 73,116
PRINT 16
PROFILE.SUB 82
PUNCH 55
PUNCHER 32,33,43,59,73,82
PUT 116
paddle 138
pal 138
pause 139
pd 139
pe 139
piece 149
plist 139
po 139
poall 150
pons 150
pops 150
pots 139
pprop 139
pps 150

pr 140
pu 140
px 140

quotient 150

READ 56,64,91
READER 32,33,43,55,59,73,82
REDEFP 140
REN 68
¦REN 18,34
RENAME 109,117
RESET 43
RESTART 40,52,84
RMAC 117
RR0,RR1,RR2 164,165
RS232 195
RSX 11,83,104
RUN 16
random 140
rc 140
recycle 140
release 141
remainder 151
remprop 141
repeat 141
rerandom 151
rl 141
round 151
rq 142
rt 142
run 142

SANSIMP.COM 216
SAVE 16,68,109,117
SCB 84,103
SCRATCH 52
SCREENCOPY 22
SCREENSWAP 22
SDLC 153,156,159,162
SECTOR 87
SEEK 32,38
SELDSK 50,55,90
SET 117
SET24X80 123
SECTRAN 56

INDEX

SETDEF 118
SETDMA 55,90
SETKEYS 123,125
SETLST 124
SETSEC 55,90
SETSIO 124
SETTRK 55,90
SETUP 45,78
SHOW 119
SID 120
SIO 32,33,45,153,185
SIZE 111
STAT 43,75
STATUT 31
STEP RATE 45
SUBMIT 75,121
SUPERZAP 206
SYS 111
SYSGEN 76
SYSTEME 100
save 142
savepic 151
se 142
setbg 151
setcursor 151
setd 151
seth 142
setpal 142
setpc 142
setpos 143
setscrunch 151
setsplit 143
setx 151
sety 151
sf 143
show 143
shuffle 151
sin 143
sound 143
ss 144
st 144
stop 144

¦TAPE 18,34
¦TAPE.IN 19,34
¦TAPE.OUT 19,34
TICK 33
TIME 92

TIMER 46
TOPLEVEL 145
TPA 39,40,41,43,69,80,81,83
TRACK 87
TRUE 146
TTY 43
TYPE 68,109,121
text 151
tf 145
thing 15
throw 145
to 145
towards 152
trace 152
ts 146
tt 146
type 146

¦USER 19,34,36,64,68,74,75,100,103,109,111,121
USERF 93
uc 152

VAL 75

WARM BOOT 42,55,83,90
WILD CART 53
WR0 154
WR1 157
WR2 159
WR3 159
WR4 160
WR5 162
WR6 163
WR7 163
WRITE 16,56,91
WRITE PROTECT 64,100
wait 146
watch 152
where 152
window 146
word 146
wordp 147
wrap 147

INDEX

XFCB 88,106
XLT 47,86
XMOVE 93
XREF 121
XSUB 76

CONSEILS DE LECTURE

Pour mieux connaître le système des CPC 464, 664 et 6128 ainsi que du PCW 8256, P.S.I. vous propose une palette d'ouvrages utiles.

Pour mieux connaître le système des CPC et du PCW 8256

■ **CLEFS POUR AMSTRAD 1. SYSTEME DE BASE**
par Daniel Martin - *Editions du P.S.I.*

Mémento présentant synthétiquement le jeu d'instructions du Z80, les points d'entrée des routines système, les connecteurs et brochages, etc.
Le livre de chevet du programmeur sur Amstrad.

■ **LE LIVRE DE CP/M PLUS SUR AMSTRAD**
par Yvon Dargery - *Editions du P.S.I.*

Toutes les commandes CP/M et CP/M plus pour maîtriser le système des 6128 et 8256 : un ouvrage de référence illustré par de nombreux programmes.

■ **LE LIVRE DE L'AMSTRAD - TOME 1**
par Daniel Martin et Philippe Jadoul - *BCM diffusé par P.S.I.*

Ce livre, destiné aux programmeurs des CPC 464 et 664, donne une étude complète de tous les circuits internes, et analyse la structure interne du BASIC. Vous y trouverez, en outre, une étude complète des RSX, et des programmes de scrolling, de traçage de rectangles, de coloriage de surface et de manipulation vectorielle.

Pour être informé régulièrement de l'actualité des micros Amstrad

■ **MICROSTRAD**
Revue bimestrielle du Groupe Tests.

Pour exploiter au mieux les capacités de votre micro, vous trouverez au sommaire de chaque numéro un rendez-vous avec les rubriques clés :

- DECOUVREZ... LA FACE CACHEE DE VOTRE CPC : astuces, idées, conseils, tout pour comprendre votre micro, son anatomie, son fonctionnement, sa programmation et exploiter ses capacités graphiques et sonores.

- DOMPTEZ... VOTRE CPC 464, 664, 6128 OU PCW 8256 : passionnés, petits ou grands, spécialistes ou débutants, une information pratique et la compétence d'experts au service de votre micro.

- PROGRAMMEZ... VOTRE MICRO AMSTRAD : dans chaque numéro de MICROSTRAD, un cocktail de programmes (dessins, jeux, utilitaires, gestion, etc.) et des trucs de programmation.

Achevé d'imprimer en janvier 1986
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : janvier 1986

N° d'impression : 601002
N° d'édition : 86595-256-1
ISBN : 2-86595-256-8

Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

- Ce livre vous donne-t-il toute satisfaction?

_____

_____

_____

_____

_____

- Y a-t-il un aspect du problème que vous auriez aimé voir abordé?

_____

_____

_____

_____

_____

Comment avez-vous eu connaissance de ce livre?

☐ publicité
☐ catalogue
☐ boutique micro
☐ autres
☐ cadeau
☐ librairie
☐ exposition

Avez-vous déjà acquis des livres PSI?

lesquels? _____

_____

qu'en pensez-vous? _____

_____

_____

Nom _____ Prénom _____ Age _____
Adresse _____
Profession _____
Centre d'intérêt _____

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex

mémento

Un mémento qui s'ouvre à la bonne page et vous permet d'accéder efficacement à toutes les informations dont vous avez besoin : commandes, points d'entrée des routines disque, blocs de contrôle, programmation et brochages des circuits spécialisés. En outre, un chapitre complet est réservé au langage LOGO distribué avec le système disque. *Clefs pour Amstrad 2- système disque* est aussi un recueil d'astuces qui vous apprendront comment transférer des programmes de la cassette vers le disque, comment utiliser l'éditeur de secteur, comment convertir du CP/M en AZERTY... Ces programmes vous feront découvrir toute l'originalité de votre CPC 664-6128 ou 464 avec extension DD1. Les possesseurs d'Amstrad 8256 trouveront également dans ce livre toutes les informations techniques importantes.

CLEFS POUR AMSTRAD

CPC 464-664/6128 et PCW 8256 2.Système disque

9 782865 952564

Éditions du PSI
B.P. 86
77402 Lagny/Marne
France

ISBN : 2.86595.256.8

Couverture Mariane de Nayer

Imprimé en France

155 FF

Téléchargé sur
Le Vieux Manuel

Http://www.abandonware-manuels.org