

PRODIG

Partie 2: Guide du programmeur

CANQ
TR
TPM
123
V.2



Gouvernement du Québec
Ministère des Transports
Direction générale du transport
des personnes et des marchandises

REÇU
CENTRE DE DOCUMENTATION

MAR 14 1995

TRANSPORTS QUÉBEC

365989

PRODIG

Partie 2: Guide du programmeur

QMTRA

CANQ

TR

TPM

123

Vo2

mars 1988

Publication réalisée à la
Direction générale du transport
des personnes et des marchandises

Cet ouvrage a été préparé par
le Service des systèmes d'information

Coordination:
Pierre Tremblay, ing.

Rédaction:
Jean Larouche
Martin Nathanson, ing.
Pierre Tremblay, ing.

Programmation:
David Brown
Jean Larouche

Assistance technique:
Line Soucy

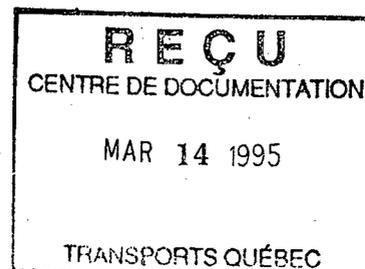


TABLE DES MATIÈRES

TABLE DES MATIÈRES

	Page
1. INTRODUCTION	1
2. STRUCTURE DU LOGICIEL	3
2.1 Le système de base	3
2.2 Hiérarchie des modules	3
2.3 Contrôle du programme	3
3. LES SOUS-ROUTINES	6
3.1 Répertoire	6
3.2 Bibliothèques graphiques	12
3.1.1 QPLOT	12
3.1.2 HCBS	12
4. LES FICHIERS	13
5. LIENS AVEC LE SYSTÈME	14
5.1 Communication avec le digimètre	14
5.2 Affichage graphique	14
5.3 Appel au système d'exploitation	15
5.4 Services d'interrupt	15
5.4.1 L'horloge	15
5.4.2 Le Print-Screen	16

Table des matières (suite)

	Page
6. PROCÉDURES INFORMATIQUES	17
6.1 Compilation	17
6.1.1 Programmes FORTRAN	17
6.1.2 Programmes ASSEMBLEUR	17
6.1.3 Fiches descriptives	17
6.2 Liaison des modules	17
6.3 Exécution	18
6.4 Archivage	19

LISTE DES ANNEXES

"A" Index des sous-routines	20
A1 Triées sur leur nom	20
A2 Triées par programme-source	22
A3 Triées par catégorie	24
"B" Fiches descriptives des routines	26
"C" Description des fichiers	130

1-INTRODUCTION

1. INTRODUCTION

Le présent document décrit un système de logiciel interactif graphique conçu par la D.G.T.T.P. pour la codification assistée par l'informatique des réseaux de transport urbain. Ce système appelé PRODIG (PROgramme de Digitalisation Interactive Graphique) vise à faciliter la saisie rapide et précise des données nécessaires à la modélisation des réseaux routiers et de transport collectif. La codification initiale d'un réseau est effectuée en superposant des cartes sur une table digitalisante. Des mises à jour ou modifications subséquentes peuvent être introduites directement sur un terminal graphique.

Le système crée des fichiers qui sont compatibles avec des logiciels de modélisation de transport urbain, notamment le système UTPS (du "U.S. Department of Transportation") ou EMME/2 (du Centre de recherche sur les transports de l'Université de Montréal). Les fichiers contiennent les attributs de noeuds et de liens, soit pour les réseaux routiers ou de transport en commun, qui permettent aux logiciels de simulation de compiler les temps et distances minimums de déplacements entre n'importe quelle paire de points. Cette combinaison de fichiers de réseau précis avec des logiciels de simulation constitue un pré-requis essentiel à la modélisation et à la planification du transport urbain.

PRODIG permet également la digitalisation de polygones de zones géographiques. Les polygones peuvent correspondre aux zones d'analyse ou à toute démarcation géographique, soit de nature politique ou physique. Les fichiers produits sont structurés afin de fournir des données qui se conforment aux logiciels graphiques comme EMME/2 ou SAS-GRAPH.

La documentation du logiciel se présente en deux parties. La première⁽¹⁾ est un guide d'utilisation, destiné à l'utilisateur. Quant à lui, ce manuel est un guide général de la structure informatique du logiciel PRODIG. Il est destiné aux programmeurs chargés du système

(1) D.G.T.T.P. PRODIG - partie I: Guide d'utilisation

et devant faire des modifications (ajouter des options, changer le format des fichiers, écrire une interface pour un équipement différent, etc...). Il contient des tableaux indiquant la hiérarchie des modules (sous-routines), le contrôle du programme (flow chart) et les descriptions de variables et fichiers utilisés. Il contient aussi une explication de l'interface "matériel", soit la communication avec la table, l'affichage graphique et les appels au système d'exploitation.

2-STRUCTURE DU LOGICIEL

2. STRUCTURE DU LOGICIEL

2.1 Le système de base

PRODIG est constitué d'une série de sous-routines, écrites en Fortran, qui exploitent les capacités graphiques à l'aide des routines de QPLOT développés au CRT⁽¹⁾. Ces routines, écrites pour un Tektronix en Fortran, ont été modifiées pour appeler des routines Assembleur sur PC écrites chez-nous. Le système a donc trois niveaux:

- 1) Le système principal, écrit en Fortran
- 2) Le système graphique, QPLOT, écrit en Fortran
- 3) L'interface avec le "matériel", écrite en Assembleur

2.2 Hiérarchie des modules

La figure 1 illustre la hiérarchie des modules principaux de PRODIG. Seules les routines FORTRAN y sont montrées; la plupart appellent à leur tour des sous-routines utilitaires en ASSEMBLEUR, par exemple, les appels I/O de bas niveau et des recherches dans les vecteurs de noeuds ou de liens. Ces sous-routines ne sont comprises dans aucun "overlay" et sont donc accessibles de partout.

2.3 Contrôle du programme

La figure 2 illustre le contrôle-type des programmes. La procédure de traitement des commandes est différente selon que le module de NOEUDS, LIENS, ZONES ou CODES POSTAUX est en opération. Les sous-routines NOEUD, LIEN, ZONE et CODPOS, décrites plus en détail dans cette section, correspondent respectivement aux routines principales de contrôle de ces modules.

(1) Spiess, Hienz: QPLOT set of subroutines for interactive-graphic network displays, Centre de recherche sur les transports, publication #232, Montréal, décembre 1981.

FIGURE 1
 ORGANISATION DES MODULES PRINCIPAUX

☐ = groupe de sous-routine en OVERLAY

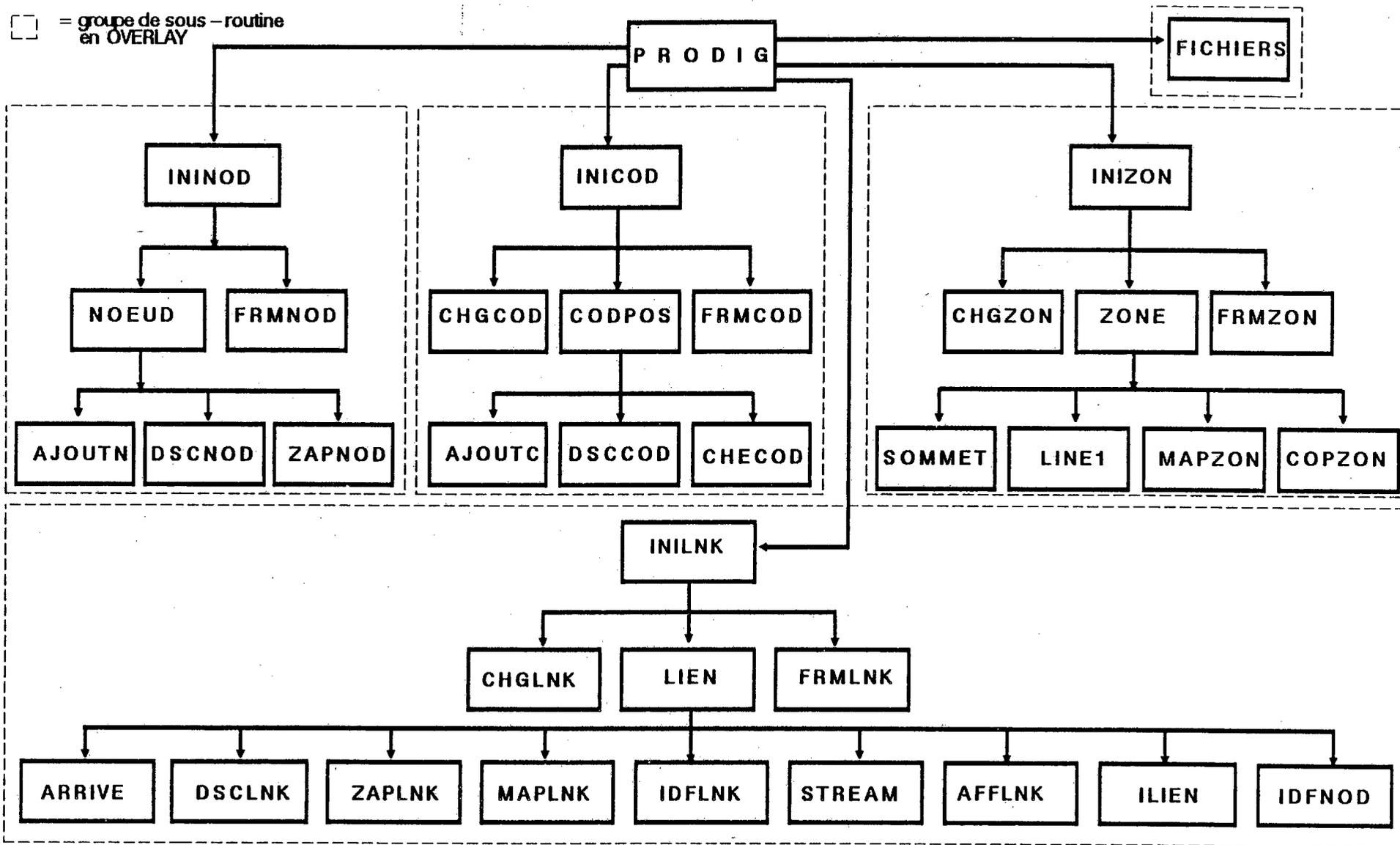
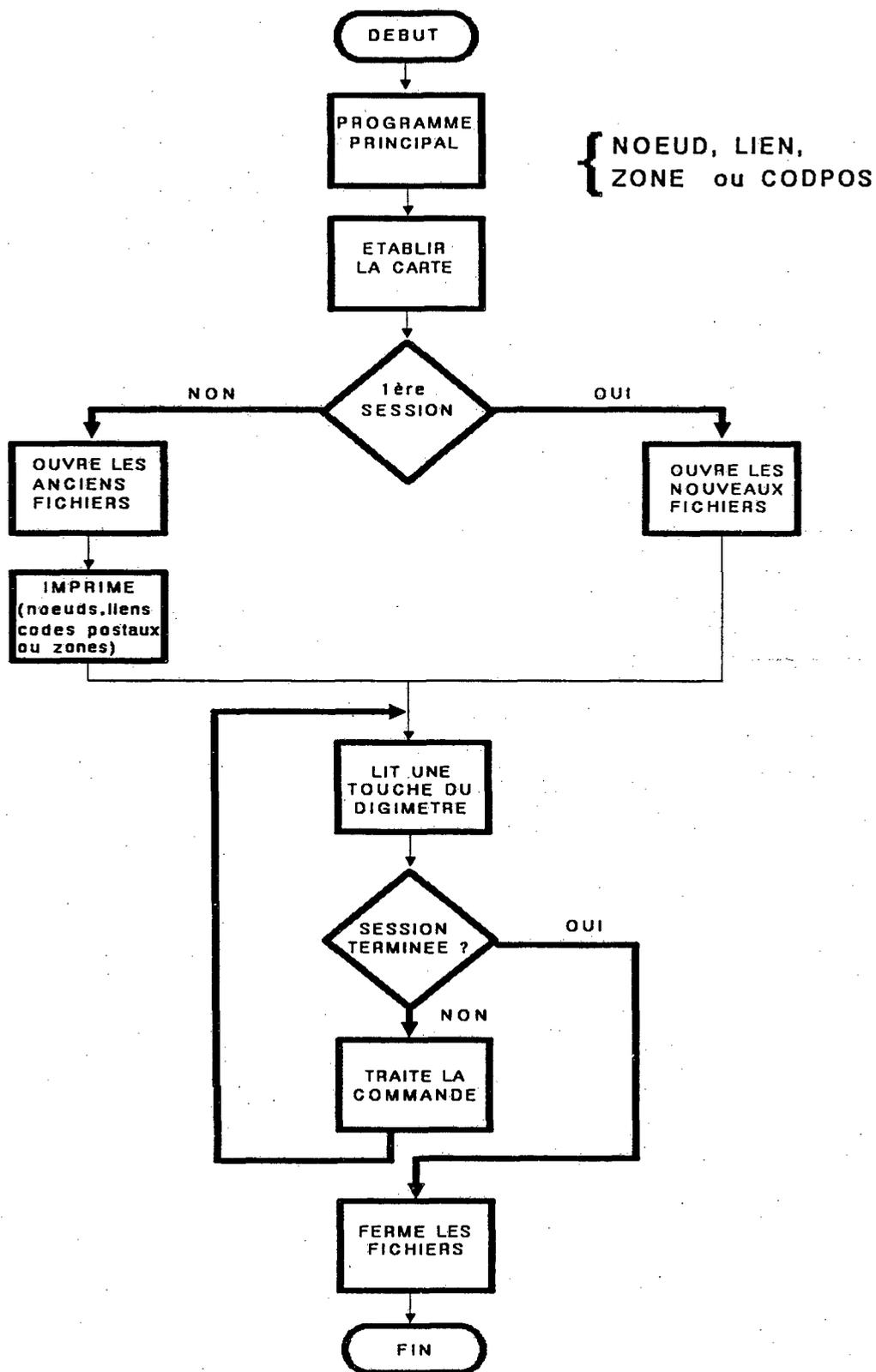


FIGURE 2
CONTRÔLE-TYPE DES PROGRAMMES



3-LES SOUS-ROUTINES

3. LES SOUS-ROUTINES

PRODIG comporte, au sens strict du terme, plus de 150 sous-routines, excluant celles des librairies graphiques QPLOT (C.R.T.) et HCBS (Cal-comp Inc.) qu'il doit aussi s'adjoindre.

La figure 3 montre la hiérarchie d'appel des routines FORTRAN pour l'ensemble du système. Les routines ASSEMBLEUR n'y apparaissent pas, étant appelés à tous les niveaux par la plupart des routines FORTRAN.

3.1 Répertoire

Afin de s'y retrouver, trois index sont fournis à l'annexe "A". Le premier répertorie toutes les sous-routines par leur nom, peu importe leur catégorie.

Le second index répertorie les sous-routines selon le nom du fichier où résident les programmes-source.

Le troisième index classe les sous-routines par catégorie, tel que définies ci-après:

N: Noeuds
L: Liens
Z: Zones
C: Codes postaux
G: Général

Les sous-routines sont décrites sur des fiches individuelles, classées par ordre alphabétique à l'annexe "B".

Ces fiches identifient chaque routine par son nom, indiquent le nom du fichier où se trouve sa programmation-source, sa catégorie d'appartenance, les arguments qu'elle utilise et une brève description de sa tâche et de son fonctionnement.

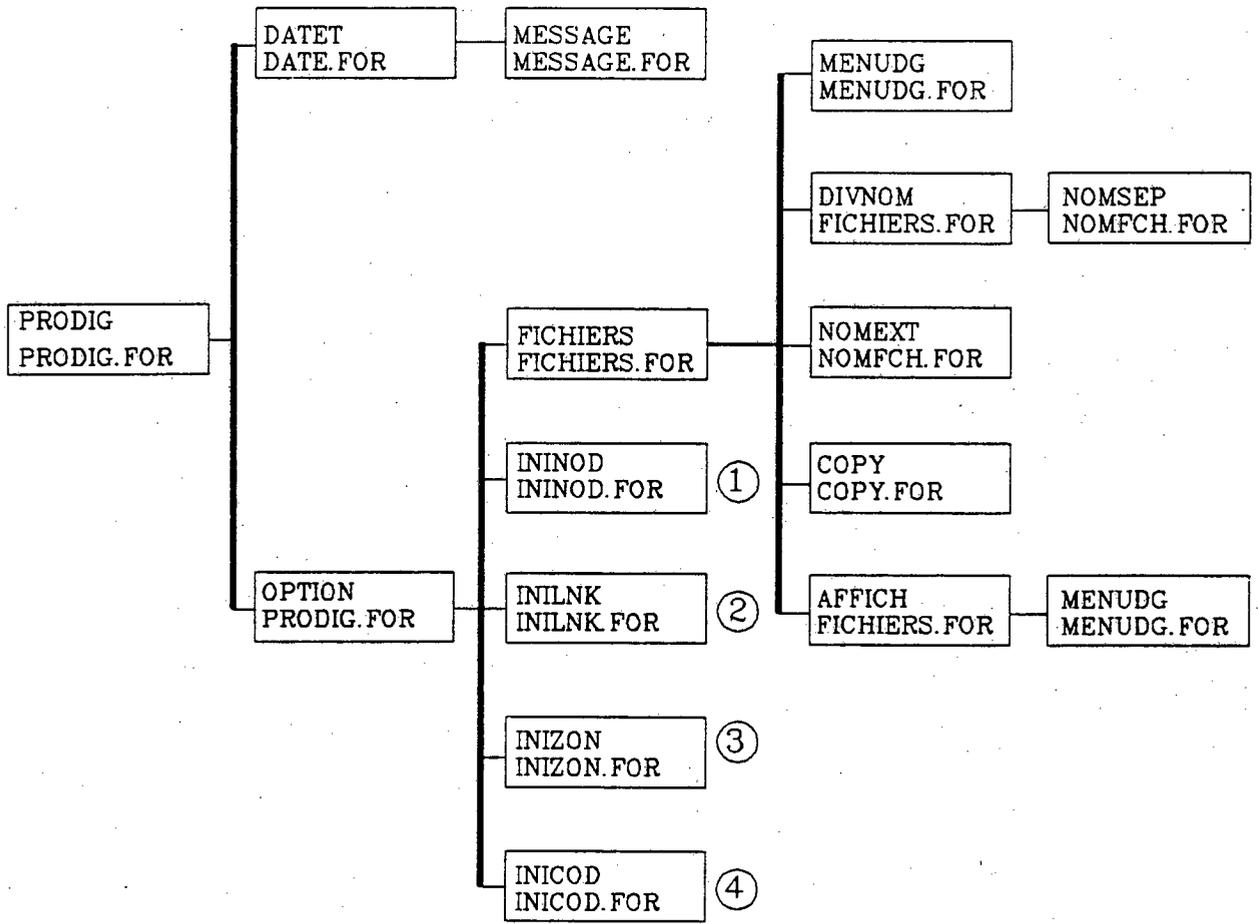
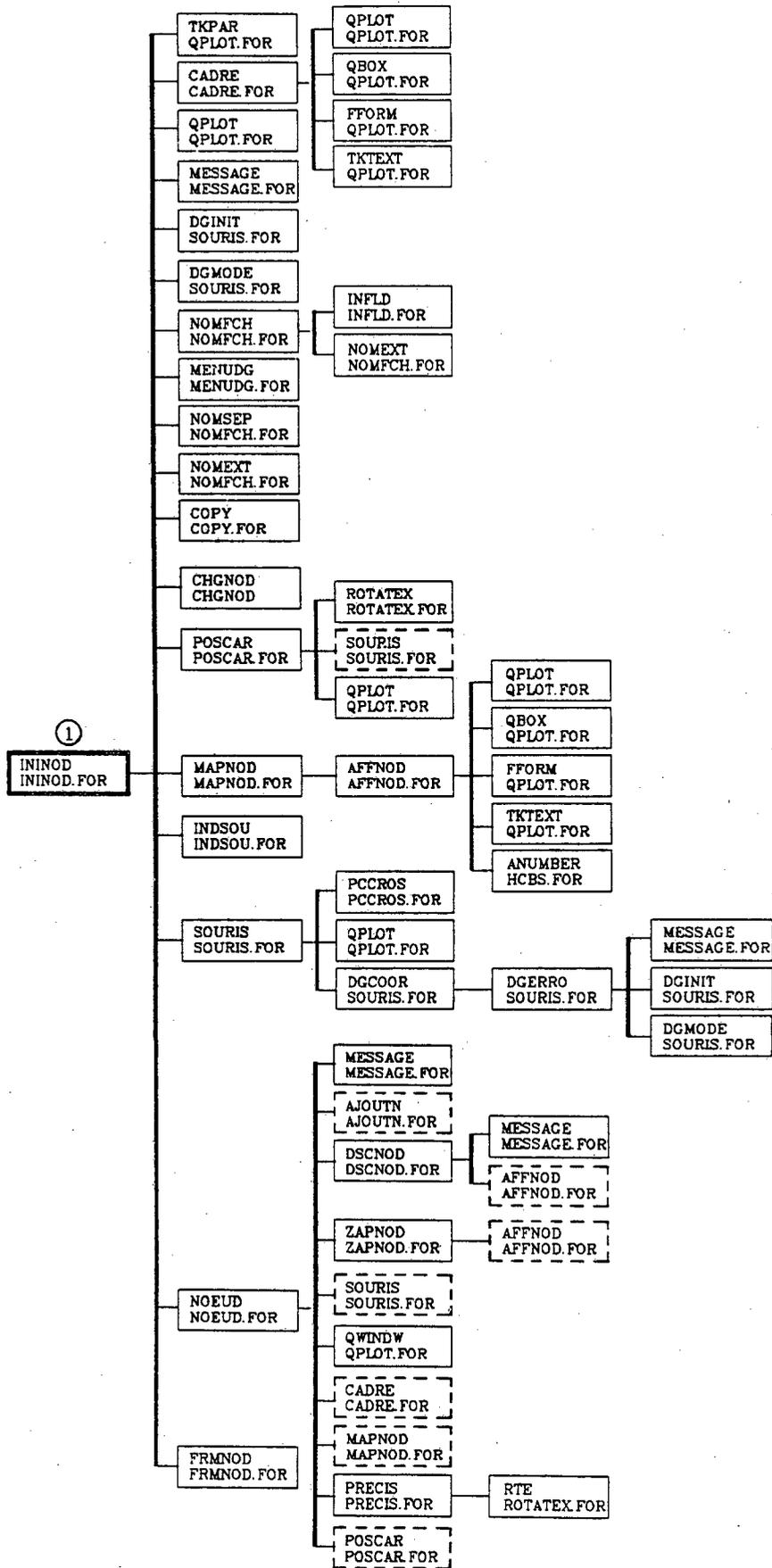
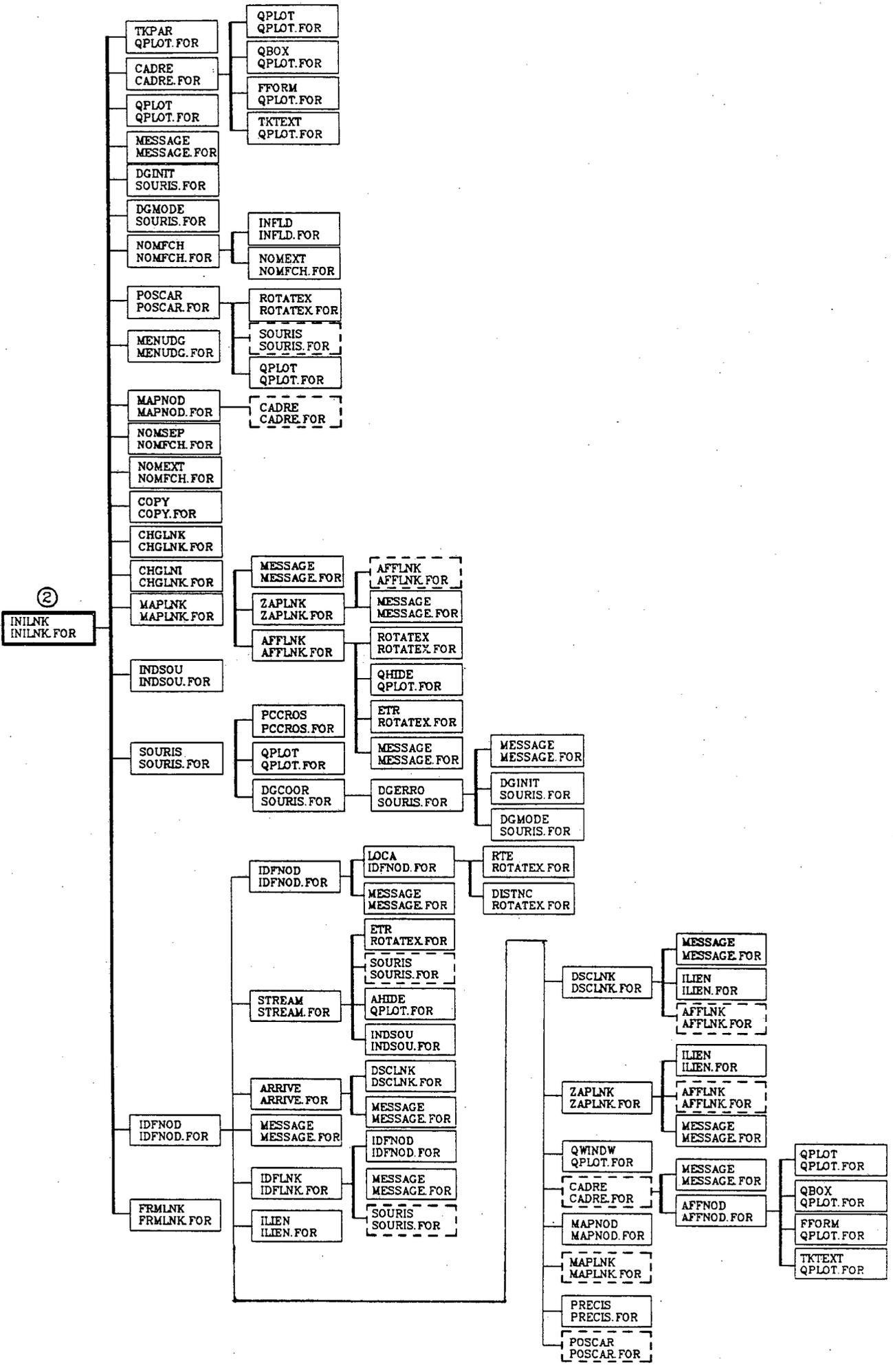
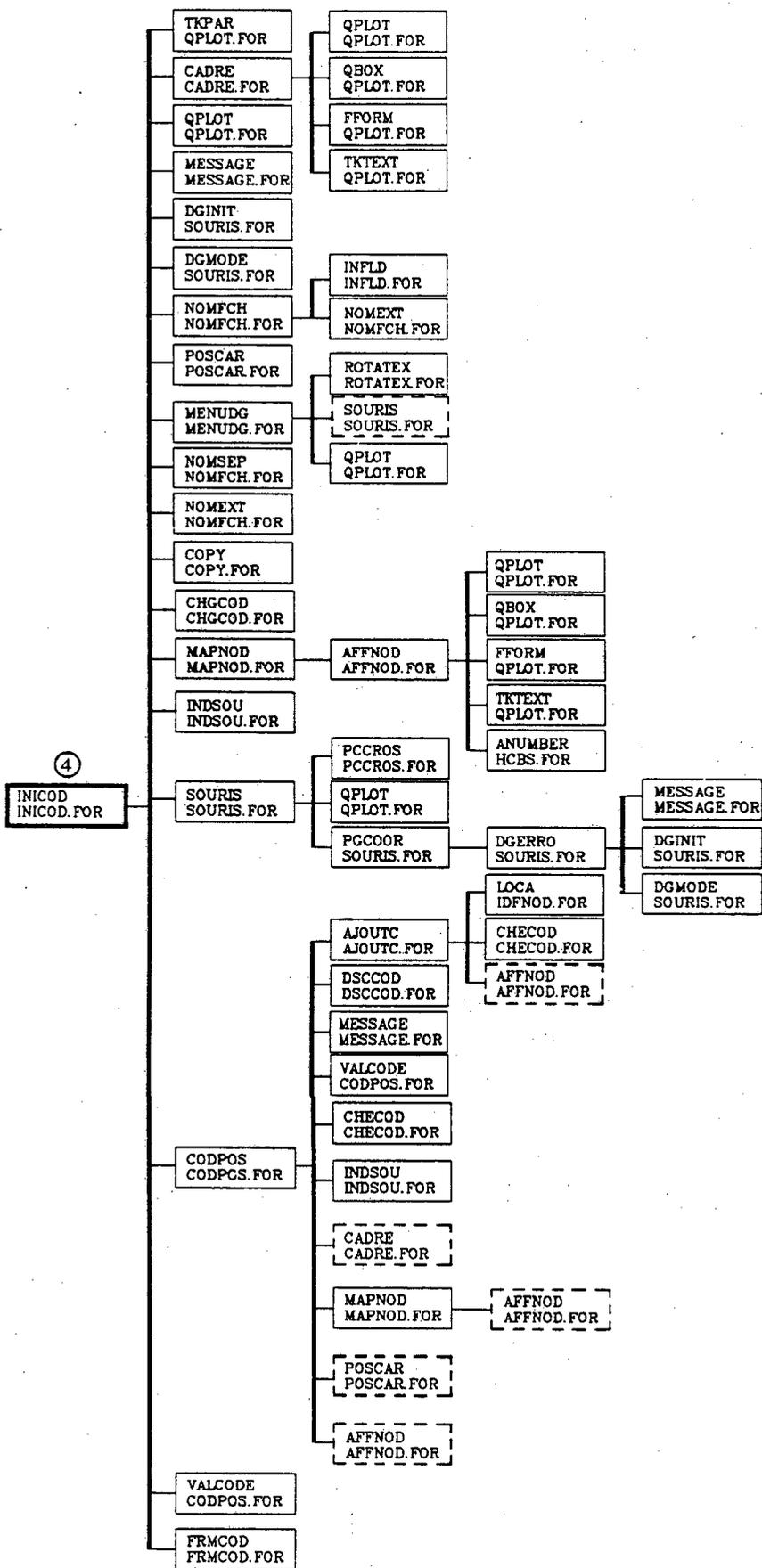


FIGURE 3
HIÉRARCHIE DES ROUTINES







3.2 Librairies graphiques

3.2.1 QPLOT

Les routines de QPLOT utilisées appellent à la limite nos routines WRDOT et LIGNE. Ces routines s'occupent de la définition des fenêtres graphiques et réelles, du "clipping" des lignes par rapport à la fenêtre graphique, du découpage des nombres en chaînes de caractères et de leur affichage. Leur utilisation est abondamment documentée dans les programmes-source eux-mêmes.

3.2.2 HCBS

Certaines routines de HCBS (PLOT, PLOTS et NUMBER) sont appelées lorsque le mode "CALCOMP" est activé sur la touche "PRTSCR" (Print Screen). Ces routines génèrent un "plot-file" qui, s'il est envoyé à un traceur CALCOMP (byte par byte), produira l'image-écran sur papier. Ici encore, l'auto-documentation des programmes-source permet d'en comprendre l'utilisation.

4-LES FICHIERS

4. LES FICHIERS

Les conventions retenues pour tous les suffixes des noms de fichiers sont résumées au tableau 1.

On retrouvera à l'annexe "C" l'ensemble des fiches descriptives des fichiers, avec leur structure d'enregistrements.

TABLEAU 1

EXTENSIONS DES FICHIERS DE PRODIG

FICHER PRINCIPAL

.NOD: Le fichier des noeuds
.LNK: Le fichier des liens
.ZON: Le fichier des zones
.SOM: Le fichier des sommets
.COD: Le fichier des codes postaux
.INT: Le fichier des points intermédiaires
.PLT: Le fichier des "plot file" du Calcomp

FICHER BACKUP

.BKN: Backup du fichier des noeuds
.BKL: Backup du fichier des liens
.BKZ: Backup du fichier des zones
.BKS: Backup du fichier des sommets
.BKC: Backup du fichier des codes postaux
.BKI: Backup du fichier des points intermédiaires

FICHER ASCII

.ASN: Le fichier des noeuds en ascii
.ASL: Le fichier des liens en ascii
.ASZ: Le fichier des zones en ascii
.ASS: Le fichier des sommets en ascii
.ASC: Le fichier des sommets en ascii
.ASI: Le fichier des points intermédiaires en ascii

5-LIENS AVEC LE SYSTÈME

5. LIENS AVEC LE SYSTÈME

5.1 Communication avec le digimètre

La communication avec la table se fait à l'aide de trois routines FORTRAN et deux routines ASSEMBLEUR. Ces routines devront être modifiées lorsqu'un digimètre autre que le CALCOMP 9000 doit être utilisé⁽¹⁾. Deux des routines FORTRAN, soit DGINIT et DGMODE servent à l'initialisation seulement. DGINIT ré-initialise la table et établit le délai entre les signaux. DGMODE établit le mode de fonctionnement de la table; PRODIG utilise toujours le "run mode".

La routine DGC00R s'occupe de retourner les coordonnées (x,y) et l'identification de la touche actionnée, le cas échéant. Il suppose que le digimètre retourne des records de longueur 13, dont le format est "CbXXXXXbYYYYY" où "C" est l'identification du bouton activé. Il extrait alors les informations pertinentes de leurs champs et les retourne en paramètres.

Les deux routines Assembleur, IOPORT, et INT14 contrôlent, à l'aide de l'interrupt 14 du BIOS, l'envoi et la réception d'informations au port sériel de communication. Ces routines sont paramétrisées pour fonctionner avec des longueurs de records variables. Il ne sera donc pas nécessaire de les changer pour l'utilisation d'une autre table.

5.2 Affichage graphique

L'affichage graphique est en tout temps fait à l'aide de routines ASSEMBLEUR qui, habituellement, font appel à l'interruption "10" du BIOS. Principalement, deux routines sont utilisées, LIGNE et WRDOT. LIGNE dessine toutes les lignes, que ce soit les côtés de boîtes de noeuds ou les flèches sur les liens, tandis que WRDOT

(1) Voir le manuel d'opération du digimètre: Calcomp 9000 Series Digitizer Operator's Manual, #50090-1

affiche les caractères à partir d'un pixel spécifié. Ces deux routines appellent à la base une fonction de l'interrupt "10" qui allume un pixel (AH=12). D'autres routines sont aussi utilisées: CWRIT pour afficher un caractère à une rangée, colonne et couleur spécifiées, CURLOC pour déplacer le curseur, etc.

Une routine spéciale, GRAF, n'appelle pas d'interrupt, mais recopie le "VIDEO-RAM" en mémoire pour pouvoir le rappeler rapidement. Ceci est utilisé, par exemple, pour sauver l'image graphique avant d'appeler le menu.

5.3 Appels au système d'exploitation

Un autre facteur qui dépend du système est l'appel au système d'exploitation (INT21H) au début de chaque module (NOEUDS, LIENS, ZONES et CODPOS). La fonction "block read" est utilisée dans les sous-routines CHGNOD, CHGLNK, CHGINT, CHGZON et CHGSOM pour rapidement lire le fichier existant.

Tout autre appel au système d'exploitation est fait par le compilateur FORTRAN.

5.4 Services d'interrupt

PRODIG intercepte deux interruptions, soit celui de l'horloge et celui du "Print Screen".

5.4.1 L'horloge (Cette fonction est inactive dans la version actuelle de PRODIG)

Au rythme de 18,2 fois par seconde, le PC génère l'interruption "1C". Le contrôle du programme est alors passé à une routine qui maintient un compteur du nombre d'interruptions. Quand ce compteur devient 18, l'heure est mise à jour et affichée au coin inférieur droit de l'écran. Un ajustement est fait à toutes les cinq secondes pour compenser pour le 0,2 restant. La sous-routine ONCLOCK

active l'interception et la routine OFFLOCK le retourne à son état inactif. Ces routines, ainsi que la routine de service, se trouvent dans le module TIME.ASM.

5.4.2 Le "Print Screen"

Lors de l'exécution d'un des modules de PRODIG, la clef "PRTSCR", qui est normalement utilisée pour envoyer le contenu de l'écran à l'imprimante, est redirigée vers le traceur CALCOMP. Pour ce faire, l'interruption "5" est interceptée et les routines d'impression MAPNOD, MAPLNK et MAPZON sont appelées selon le cas. Ces routines envoient alors des commandes graphiques à un "plot-file" par les routines de HCBS, décrites dans la section précédente.

Les routines ONSCR et OFFSCR activent et désactivent ce mode. Ces routines, ainsi que la routine de service, se trouvent dans le fichier PSCR.ASM. La routine de service appelle le sous-routine Fortran "CALC" qui s'occupe de l'ouverture des fichiers et des appels aux routines d'impression, selon le programme.

La routine de service n'est pas récursive; un appel à Print Screen, avant qu'un précédent ne soit complété, causera donc un "crash".

6-PROCÉDURES INFORMATIQUES

6. PROCÉDURES INFORMATIQUES

6.1 Compilation

6.1.1 Programmes FORTRAN

Tous les programmes-source (.FOR) et modules objets (.OBJ) des programmes faisant partie de PRODIG se trouvent sur la cartouche IOMEGA étiquetée "PRODIG-SOURCE". Le compilateur FORTRAN s'y trouve également et peut être invoqué par la commande:

C nom

où nom est le nom du fichier source (.FOR) à compiler.

6.1.2 Programmes ASSEMBLEUR

Tous les programmes-source (.ASM) et modules objet (.OBJ) se trouvent sur la même cartouche IOMEGA. L'assembleur peut être invoqué sur cette cartouche par la commande:

MASM nom

où nom est le nom du fichier source à compiler.

6.1.3 Fiches descriptives

Toujours sur la cartouche "PRODIG-SOURCE", on trouvera aussi l'ensemble des fiches descriptives des routines, dans le "directory" DOCUM.

6.2 Liaison des modules

PRODIG est "linké" avec les bibliothèques:

ASM.LIB,
ACSV20.LIB,
FILEV20.LIB;

qui se trouvent sur la cartouche "PRODIG-SOURCE". Lorsqu'un programme en ASSEMBLEUR est modifié, il doit être réintégré dans la librairie ASM.LIB, en utilisant l'utilitaire "LIBRARIAN" de MICROSOFT, avec les commandes:

LIB ASM + nom

où nom est le nom du module objet en question.

La liaison finale s'opère avec la commande:

NDIG

résidant toujours sur la cartouche "PRODIG-SOURCE", et qui utilise le fichier NDIG.ARF (Automatic Response File) pour identifier tous les modules objets (.OBJ) et librairies, ainsi que la structure d'overlay lors du "linkage".

Le fichier résultant, PRODIG.EXE, est le produit final, prêt à exécuter.

6.3 Exécution

PRODIG est exécuté en tapant la commande:

PRODIG

Le logiciel requiert la présence des fichiers auxiliaires suivants, sur le même support physique que PRODIG.EXE:

MENUG.DAT
MENUN.DAT
MENUT.DAT
MENUZ.DAT
MENUC.DAT
MENUF.DAT

L'annexe "D" montre le contenu de ces fichiers, associés aux différents menus affichés à l'écran par PRODIG. Leur format est expliqué à l'annexe "C".

Finalement, tous les fichiers de données eux-mêmes doivent également résider sur la même unité logique que le programme PRODIG.EXE.

6.4 Archivage

Le fichier PRODIG.EXE est trop grand pour être copié sur une seule disquette. On peut utiliser à cette fin les commandes BACKUP et RESTORE de PC-DOS.

ANNEXE A

Index des sous-routines

ANNEXE A-1

LISTE DES ROUTINES
TRIEES SUR LEUR NOM

ROUTINE	SOURCE	CATEGORIE
AFFLNK	AFFLNK .FOR	L
AFFNOD	AFFNOD .FOR	N
AJOUTC	AJOUTC .FOR	C
AJOUTN	AJOUTN .FOR	N
ARRIVE	ARRIVE .FOR	L
CADRE	CADRE .FOR	G
CALC	CALC .FOR	G
CHECOD	CHECOD .FOR	C
CHGCOD	CHGCOD .FOR	C
CHGLNI	CHGLNI .FOR	L
CHGLNK	CHGLNK .FOR	L
CHGNOD	CHGNOD .FOR	N
CHGSOM	CHGZON .FOR	Z
CHGZON	CHGZON .FOR	Z
CLBUF	MESSAGE .FOR	G
CLSG	CLSG .ASM	G
CODPOS	CODPOS .FOR	C
COPY	COPY .FOR	G
COPZON	COPZON .FOR	Z
CREAD	CREAD .ASM	G
CURLOC	CURLOC .ASM	G
CURRET	CURRET .ASM	G
CWRIT	CWRIT .ASM	G
DATEG	DATE .FOR	G
DATET	DATE .FOR	G
DGBEEP	SOURIS .FOR	G
DGCOOR	SOURIS .FOR	G
DGERRO	SOURIS .FOR	G
DGINIT	SOURIS .FOR	G
DGMODE	SOURIS .FOR	G
DISTNC	ROTATEX .FOR	G
DSCCOD	DSCCOD .FOR	C
DSCLNK	DSCLNK .FOR	L
DSCNOD	DSCNOD .FOR	N
ERASE	INFLD .FOR	G
ETR	ROTATEX .FOR	G
FICHIERS	FICHIERS .FOR	G
FRMCOD	FRMCOD .FOR	C
FRMLNK	FRMLNK .FOR	L
FRMNOD	FRMNOD .FOR	N
FRMSOM	FRMZON .FOR	Z
FRMZON	FRMZON .FOR	Z
GETDAT	GETDAT .ASM	G
GETMOD	GETMOD .ASM	G
GRAF	GRAF .ASM	G
HCBS	HCBS .FOR	G
IDFLNK	IDFLNK .FOR	L
IDFNOD	IDFNOD .FOR	N
ILIE	ILIE .FOR	L
INDSOU	INDSOU .FOR	G
INFLD	INFLD .FOR	G
INICOD	INICOD .FOR	C
INILNK	INILNK .FOR	L

ININOD	ININOD	.FOR	N
INIZON	INIZON	.FOR	Z
INT14	INT14	.ASM	G
IOPORT	IOPORT	.ASM	G
IRECO	IRECO	.ASM	G
IRECORD	IRECORD	.FOR	G
KBREAD	KBREAD	.ASM	G
LIEN	LIEN	.FOR	L
LIGNE	LIGNE	.ASM	G
LINE1	LINE1	.FOR	G
LOCA	IDFNOD	.FOR	G
MAPGIN	MAPGIN	.ASM	G
MAPLNK	MAPLNK	.FOR	L
MAPNOD	MAPNOD	.FOR	N
MAPZON	MAPZON	.FOR	Z
MENU DG	MENU DG	.FOR	G
MESSAGE	MESSAGE	.FOR	G
MODSET	MODSET	.ASM	G
NOEUD	NOEUD	.FOR	N
NOMEXT	NOMFCH	.FOR	G
NOMFCH	NOMFCH	.FOR	G
NOMSEP	NOMFCH	.FOR	G
NUMBER	INFLD	.FOR	G
OFFCB	CB	.ASM	G
OFFCLOCK	TIME	.ASM	G
OFFSCR	PSCR	.ASM	G
ONCB	CB	.ASM	G
ONCLOCK	TIME	.ASM	G
ONSCR	PSCR	.ASM	G
OPTION	PRODIG	.FOR	G
OUTFLD	INFLD	.FOR	G
PAGEN	PAGEN	.ASM	G
PCCROS	PCCROS	.FOR	G
PCGIN	PCGIN	.ASM	G
POSCAR	POSCAR	.FOR	G
PRECIS	PRECIS	.FOR	G
PRODIG	PRODIG	.FOR	G
PSCR	PSCR	.ASM	G
QPLOT	QPLOT	.FOR	G
RDBUF	RDBUF	.ASM	G
READC	MESSAGE	.FOR	G
ROTATEX	ROTATEX	.FOR	G
RTE	ROTATEX	.FOR	G
SKROL	SKROL	.ASM	G
SOMMET	SOMMET	.FOR	Z
SOUND	SOUND	.ASM	G
SOURIS	SOURIS	.FOR	G
STREAM	STREAM	.FOR	L
TIME	TIME	.ASM	G
VADRS	VADRS	.ASM	G
VALCOD	INICOD	.FOR	C
VLOAD	VLOAD	.ASM	G
WRDOT	WRDOT	.ASM	G
ZAPLNK	ZAPLNK	.FOR	L
ZAPNOD	ZAPNOD	.FOR	G
ZONE	ZONE	.FOR	Z

ANNEXE A-2

LISTE DES ROUTINES
TRIEES PAR PROGRAMME-SOURCE

ROUTINE	SOURCE	CATEGORIE
AFFLNK	AFFLNK .FOR	L
AFFNOD	AFFNOD .FOR	N
AJOUTC	AJOUTC .FOR	C
AJOUTN	AJOUTN .FOR	N
ARRIVE	ARRIVE .FOR	L
CADRE	CADRE .FOR	G
CALC	CALC .FOR	G
OFFCB	CB .ASM	G
ONCB	CB .ASM	G
CHECOD	CHECOD .FOR	C
CHGCOD	CHGCOD .FOR	C
CHGLNI	CHGLNI .FOR	L
CHGLNK	CHGLNK .FOR	L
CHGNOD	CHGNOD .FOR	N
CHGSOM	CHGZON .FOR	Z
CHGZON	CHGZON .FOR	Z
CLSG	CLSG .ASM	G
CODPOS	CODPOS .FOR	C
COPY	COPY .FOR	G
COPZON	COPZON .FOR	Z
CREAD	CREAD .ASM	G
CURLOC	CURLOC .ASM	G
CURRET	CURRET .ASM	G
CWRIT	CWRIT .ASM	G
DATEG	DATE .FOR	G
DATET	DATE .FOR	G
DSCCOD	DSCCOD .FOR	C
DSCLNK	DSCLNK .FOR	L
DSCNOD	DSCNOD .FOR	N
FICHIERS	FICHIERS .FOR	G
FRMCOD	FRMCOD .FOR	C
FRMLNK	FRMLNK .FOR	L
FRMNOD	FRMNOD .FOR	N
FRMSOM	FRMZON .FOR	Z
FRMZON	FRMZON .FOR	Z
GETDAT	GETDAT .ASM	G
GETMOD	GETMOD .ASM	G
GRAF	GRAF .ASM	G
HCBS	HCBS .FOR	G
IDFLNK	IDFLNK .FOR	L
IDFNOD	IDFNOD .FOR	N
LOCA	IDFNOD .FOR	G
ILIEN	ILIEN .FOR	L
INDSOU	INDSOU .FOR	G
ERASE	INFLD .FOR	G
INFLD	INFLD .FOR	G
NUMBER	INFLD .FOR	G
OUTFLD	INFLD .FOR	G
INICOD	INICOD .FOR	C
VALCOD	INICOD .FOR	C
INILNK	INILNK .FOR	L
ININOD	ININOD .FOR	N
INIZON	INIZON .FOR	Z

INT14	INT14	.ASM	G
IOPORT	IOPORT	.ASM	G
IRECO	IRECO	.ASM	G
IRECORD	IRECORD	.FOR	G
KBREAD	KBREAD	.ASM	G
LIEN	LIEN	.FOR	L
LIGNE	LIGNE	.ASM	G
LINE1	LINE1	.FOR	G
MAPGIN	MAPGIN	.ASM	G
MAPLNK	MAPLNK	.FOR	L
MAPNOD	MAPNOD	.FOR	N
MAPZON	MAPZON	.FOR	Z
MENUJG	MENUJG	.FOR	G
CLBUF	MESSAGE	.FOR	G
MESSAGE	MESSAGE	.FOR	G
READC	MESSAGE	.FOR	G
MODSET	MODSET	.ASM	G
NOEUD	NOEUD	.FOR	N
NOMEXT	NOMFCH	.FOR	G
NOMFCH	NOMFCH	.FOR	G
NOMSEP	NOMFCH	.FOR	G
PAGEN	PAGEN	.ASM	G
PCCROS	PCCROS	.FOR	G
PCGIN	PCGIN	.ASM	G
POSCAR	POSCAR	.FOR	G
PRECIS	PRECIS	.FOR	G
OPTION	PRODIG	.FOR	G
PRODIG	PRODIG	.FOR	
OFFSCR	PSCR	.ASM	G
ONSCR	PSCR	.ASM	G
PSCR	PSCR	.ASM	G
QPLOT	QPLOT	.FOR	G
RDBUF	RDBUF	.ASM	G
DISTNC	ROTATEX	.FOR	G
ETR	ROTATEX	.FOR	G
ROTATEX	ROTATEX	.FOR	G
RTE	ROTATEX	.FOR	G
SKROL	SKROL	.ASM	G
SOMMET	SOMMET	.FOR	Z
SOUND	SOUND	.ASM	G
DGBEEP	SOURIS	.FOR	G
DGCOOR	SOURIS	.FOR	G
DGERRO	SOURIS	.FOR	G
DGINIT	SOURIS	.FOR	G
DGMODE	SOURIS	.FOR	G
SOURIS	SOURIS	.FOR	G
STREAM	STREAM	.FOR	L
OFFCLOCK	TIME	.ASM	G
ONCLOCK	TIME	.ASM	G
TIME	TIME	.ASM	G
VADRS	VADRS	.ASM	G
VLOAD	VLOAD	.ASM	G
WRDOT	WRDOT	.ASM	G
ZAPLNK	ZAPLNK	.FOR	L
ZAPNOD	ZAPNOD	.FOR	G
ZONE	ZONE	.FOR	Z

ANNEXE A-3

LISTE DES ROUTINES
TRIEES PAR CATEGORIE

ROUTINE	SOURCE	CATEGORIE
PRODIG	PRODIG .FOR	
AJOUTC	AJOUTC .FOR	C
CHECOD	CHECOD .FOR	C
CHGCOD	CHGCOD .FOR	C
CODPOS	CODPOS .FOR	C
DSCCOD	DSCCOD .FOR	C
FRMCOD	FRMCOD .FOR	C
INICOD	INICOD .FOR	C
VALCOD	INICOD .FOR	C
CADRE	CADRE .FOR	G
CALC	CALC .FOR	G
CLBUF	MESSAGE .FOR	G
CLSG	CLSG .ASM	G
COPY	COPY .FOR	G
CREAD	CREAD .ASM	G
CURLOC	CURLOC .ASM	G
CURRET	CURRET .ASM	G
CWRIT	CWRIT .ASM	G
DATEG	DATE .FOR	G
DATET	DATE .FOR	G
DGBEEP	SOURIS .FOR	G
DGCOOR	SOURIS .FOR	G
DGERRO	SOURIS .FOR	G
DGINIT	SOURIS .FOR	G
DGMODE	SOURIS .FOR	G
DISTNC	ROTATEX .FOR	G
ERASE	INFLD .FOR	G
ETR	ROTATEX .FOR	G
FICHIERS	FICHIERS .FOR	G
GETDAT	GETDAT .ASM	G
GETMOD	GETMOD .ASM	G
GRAF	GRAF .ASM	G
HCBS	HCBS .FOR	G
INDSOU	INDSOU .FOR	G
INFLD	INFLD .FOR	G
INT14	INT14 .ASM	G
IOPORT	IOPORT .ASM	G
IRECO	IRECO .ASM	G
IRECORD	IRECORD .FOR	G
KBREAD	KBREAD .ASM	G
LIGNE	LIGNE .ASM	G
LINE1	LINE1 .FOR	G
LOCA	IDFNOD .FOR	G
MAPGIN	MAPGIN .ASM	G
MENUDG	MENUDG .FOR	G
MESSAGE	MESSAGE .FOR	G
MODSET	MODSET .ASM	G
NOMEXT	NOMFCH .FOR	G
NOMFCH	NOMFCH .FOR	G
NOMSEP	NOMFCH .FOR	G
NUMBER	INFLD .FOR	G
OFFCB	CB .ASM	G
OFFCLOCK	TIME .ASM	G

OFFSCR	PSCR	.ASM	G
ONCB	CB	.ASM	G
ONCLOCK	TIME	.ASM	G
ONSCR	PSCR	.ASM	G
OPTION	PRODIG	.FOR	G
OUTFLD	INFLD	.FOR	G
PAGEN	PAGEN	.ASM	G
PCCROS	PCCROS	.FOR	G
PCGIN	PCGIN	.ASM	G
POSCAR	POSCAR	.FOR	G
PRECIS	PRECIS	.FOR	G
PSCR	PSCR	.ASM	G
QPLOT	QPLOT	.FOR	G
RDBUF	RDBUF	.ASM	G
READC	MESSAGE	.FOR	G
ROTATEX	ROTATEX	.FOR	G
RTE	ROTATEX	.FOR	G
SKROL	SKROL	.ASM	G
SOUND	SOUND	.ASM	G
SOURIS	SOURIS	.FOR	G
TIME	TIME	.ASM	G
VADRS	VADRS	.ASM	G
VLOAD	VLOAD	.ASM	G
WRDOT	WRDOT	.ASM	G
ZAPNOD	ZAPNOD	.FOR	G
AFFLNK	AFFLNK	.FOR	L
ARRIVE	ARRIVE	.FOR	L
CHGLNI	CHGLNI	.FOR	L
CHGLNK	CHGLNK	.FOR	L
DSCLNK	DSCLNK	.FOR	L
FRMLNK	FRMLNK	.FOR	L
IDFLNK	IDFLNK	.FOR	L
ILIE	ILIE	.FOR	L
INILNK	INILNK	.FOR	L
LIEN	LIEN	.FOR	L
MAPLNK	MAPLNK	.FOR	L
STREAM	STREAM	.FOR	L
ZAPLNK	ZAPLNK	.FOR	L
AFFNOD	AFFNOD	.FOR	N
AJOUTN	AJOUTN	.FOR	N
CHGNOD	CHGNOD	.FOR	N
DSCNOD	DSCNOD	.FOR	N
FRMNOD	FRMNOD	.FOR	N
IDFNOD	IDFNOD	.FOR	N
ININOD	ININOD	.FOR	N
MAPNOD	MAPNOD	.FOR	N
NOEUD	NOEUD	.FOR	N
CHGSOM	CHGZON	.FOR	Z
CHGZON	CHGZON	.FOR	Z
COPZON	COPZON	.FOR	Z
FRMSOM	FRMZON	.FOR	Z
FRMZON	FRMZON	.FOR	Z
INIZON	INIZON	.FOR	Z
MAPZON	MAPZON	.FOR	Z
SOMMET	SOMMET	.FOR	Z
ZONE	ZONE	.FOR	Z

ANNEXE B

Fiches descriptives des routines

ROUTINE: AFFLNK

SOURCE: AFFLNK.FOR

CATEGORIE: L

ARGUMENTS: CALL AFFLNK(IC)

DESCRIPTION:

Cette sous-routine affiche un lien à l'écran.

Elle commence par faire une conversion des coordonnées réelles en coordonnées de l'écran et appelle ensuite QHIDE, une routine QPLOT, qui s'occupe de tracer la ligne aussi bien que la flèche au bout.

Si le lien est de type "stream", le fichier de points intermédiaires (unité 14) est examiné pour tracer des lignes intermédiaires, sans flèches.

Le paramètre "IC" sert à indiquer la "couleur" d'affichage; lorsque IC = 0, le lien est effacé.

ROUTINE: AFFNOD

SOURCE: AFFNOD.FOR

CATEGORIE: N

ARGUMENTS: CALL AFFNOD(XR,YR,NOEUD,IC)

DESCRIPTION:

Cette sous-routine affiche le numero "NOEUD", encadré aux coordonnées réelles (XR,YR). Les coordonnées réelles sont d'abord converties en coordonnées graphiques et une boîte est dessinée à l'aide de la sous-routine QBOX. Le numéro de noeud (NOEUD) est alors converti en chaîne de caractères avec FFORM et est affiché au a boîte avec TKTEXT.

ROUTINE: AJOUTC

SOURCE: AJOUTC.FOR

CATEGORIE: C

ARGUMENTS: CALL AJOUTC

DESCRIPTION:

Cette sous-routine permet d'ajouter un code-postal au système courant. Elle vérifie d'abord s'il existe déjà un point aux coordonnées graphiques (XR,YR), à l'aide de la sous-routine LOCA. Si c'est le cas, un message est affiché. Si le numéro de noeud est plus petit que 100, les centaines prennent la valeur des centaines du numéro de noeud précédent (NOEUD3).

ROUTINE: AJOUTN

SOURCE: AJOUTN.FOR

CATEGORIE: N

ARGUMENTS: CALL AJOUTN

DESCRIPTION:

Cette sous-routine permet d'ajouter un noeud au réseau courant. Elle vérifie d'abord s'il existe déjà un noeud aux coordonnées graphiques (XR,YR), à l'aide de la sous-routine LOCA. Si c'est cas, un message est affiché. Sinon, le système demande un numéro de noeud à l'utilisateur et l'emmagasine dans la variable NOEUD. Si l'utilisateur ne retourne aucune valeur (indiqué par "0"), la valeur par défaut (noeud précédent + 1) est utilisée (variable NOEUD2). Si le numéro de noeud est plus petit que 100, les centaines prennent la valeur des centaines du numéro de noeud précédent (NOEUD3). La sous-routine IRECORD est alors appelée pour vérifier que le numéro n'existe pas déjà, auquel cas un message est affiché.

ROUTINE: ARRIVE

SOURCE: ARRIVE.FOR

CATEGORIE: L

ARGUMENTS: CALL ARRIVE

DESCRIPTION:

Cette sous-routine est appelée lorsqu'un lien a été défini. Elle appelle "DSCLNK" pour prendre la description des attributs du lien. Si le lien est bidirectionnel, elle demande si la description est identique pour le sens contraire et appelle "DSCLNK" une deuxième fois, le cas échéant.

ROUTINE: CADRE

SOURCE: CADRE.FOR

CATEGORIE: G

ARGUMENTS: CALL CADRE

DESCRIPTION:

Cette sous-routine dessine le cadre extérieur de l'espace graphique de l'écran.

ROUTINE: CALC

SOURCE: CALC.FOR

CATEGORIE: G

ARGUMENTS: CALL CALC

DESCRIPTION:

Cette sous-routine est appelée par la routine assembleur "PSCR" pour créer un fichier-dessin de type "Calcomp" du réseau courant. Il lit et valide le nom du fichier. Puis il détermine l'échelle du dessin sur le CALCOMP 1075. Pour faire dessiner le fichier utilisez le logiciel NHCALP.COM.

ROUTINE: CHECOD

SOURCE: CHECOD.FOR

CATEGORIE: C

ARGUMENTS: CALL CHECOD

DESCRIPTION:

Cette sous-routine détermine si un code-postal existe déjà à l'intérieur du fichier. La méthode de recherche se fait en additionnant les caractères ascii du code-postal, et en les comparant au données du vecteur ICPOS. Lorsqu'une donnée est identique, il vérifie que les caractères des codes-postaux sont similaire.

ROUTINE: CHGCOD

SOURCE: CHGCOD.FOR

CATEGORIE: C

ARGUMENTS: CALL CHGCOD

DESCRIPTION:

Cette sous-routine charge les vecteurs TAB1, TAB2, IREC et ICPOS. Pour le vecteur ICPOS, la valeur ascii des caractères du code-postal sont additionné et ensuite chargé dans le vecteur. Les records du fichier sont copiés dans le fichier TEMP.DTA.

ROUTINE: CHGLNI

SOURCE: CHGLNI.FOR

CATEGORIE: L

ARGUMENTS: CALL CHGLNI

DESCRIPTION:

Cette sous-routine lit le fichier des points intermédiaires et détermine le nombre total de record.

Des routines qui appellent l'interrupt "21H" de DOS sont appelées pour établir le nom du fichier (ASCIIZ), pour établir l'adresse de transfert (SETXFR), pour ouvrir les fichiers (OPFL2), pour les initialiser (LSEEK), et finalement les fermer (CLFIL2). Ces routines font partie de la librairie "FILEV20".

ROUTINE: CHGLNK

SOURCE: CHGLNK.FOR

CATEGORIE: L

ARGUMENTS: CALL CHGLNK

DESCRIPTION:

Cette sous-routine lit le fichier des liens existants et charge les informations requises dans les vecteurs correspondants. Elle fonctionne sur le même principe que la routine "CHGNOD".

Des routines qui appellent l'interrupt "21H" de DOS sont appelées pour établir le nom du fichier (ASCIIZ), pour établir l'adresse de transfert (SETXFR), pour créer le fichier (CREAT), pour ouvrir les fichiers (OPFL2) et les initialiser (LSEEK), pour lire (REDFLD) et écrire (WRTFLD) aux fichiers, et finalement les fermer (CLFIL2). Ces routines font partie de la librairie de routines "FILEV20".

Le vecteur IRECL est modifié. il contient dans les deux derniers bytes le numero du noeud de départ et dans les deux autres le numero du noeud d'arrivé. Il fait le radical des deux nombres et multiplie le noeud de départ par 10000, pour qu'il n'utilise que deux bytes.

ROUTINE: CHGNOD

SOURCE: CHGNOD.FOR

CATEGORIE: N

ARGUMENTS: CALL CHGNOD

DESCRIPTION:

Cette sous-routine lit le fichier de noeuds de base et charge les informations requises dans les vecteurs TAB et IREC. S'il s'agit du programme de noeuds, (IPROG=1), le fichier est aussi recopié dans le fichier de travail "TEMP.DTA".

Des routines qui appellent l'interrupt "21H" de DOS sont appelées pour établir le nom du fichier (ASCIIZ), pour établir l'adresse de transfert (SETXFR), pour créer le fichier (CREAT), pour ouvrir les fichiers (OPFL2) et les initialiser (LSEEK), pour lire (REDFLD) et écrire (WRTFLD) aux fichiers, et finalement les fermer (CLFIL2). Ces routines font partie de la librairie de routines "FILEV20".

"CHGNOD" fonctionne comme suit:

- 1- Les fichiers sont ouverts et l'initialisation est faite.
- 2- Un bloc (jusqu'à 32 k-octets) est lu du fichier avec DMA (direct memory access) et est placé dans le vecteur "ARRAY".
- 3- S'il s'agit du module "NOEUDS", ce bloc est écrit au fichier de travail.
- 4- Les numéros de noeuds sont extraits et placés dans le vecteur "IREC", à l'aide des sous-routines "VADRS" et "VLOAD".
- 5- Les coordonnées sont placées dans le vecteur "TAB" de la même façon.
- 6- Les étapes 2 à 5 sont répétées jusqu'à ce que tout le fichier ait été lu.
- 7- Les fichiers sont fermés.

ROUTINE: CHGSOM

SOURCE: CHGSOM.FOR

CATEGORIE: Z

ARGUMENTS: CALL CHGSOM

DESCRIPTION:

Cette sous-routine lit le fichier des sommets existants et charge les informations requises dans les vecteurs correspondants. Elle fonctionne sur le même principe que la routine "CHGNOD".

Des routines qui appellent l'interrupt "21H" de DOS sont appelées pour établir le nom du fichier (ASCIIZ), pour établir l'adresse de transfert (SETXFR), pour créer le fichier (CREAT), pour ouvrir les fichiers (OPFL2) et les initialiser (LSEEK), pour lire (REDFLD) et écrire (WRTFLD) aux fichiers, et finalement les fermer (CLFIL2). Ces routines font partie de la librairie de routines "FILEV20".

ROUTINE: CHGZON

SOURCE: CHGZON.FOR

CATEGORIE: Z

ARGUMENTS: CALL CHGZON

DESCRIPTION:

Cette sous-routine lit le fichier des zones existantes et charge les informations requises dans les vecteurs correspondants. Elle fonctionne sur le même principe que la routine "CHGNOD".

Des routines qui appellent l'interrupt "21H" de DOS sont appelées pour établir le nom du fichier (ASCIIZ), pour établir l'adresse de transfert (SETXFR), pour créer le fichier (CREAT), pour ouvrir les fichiers (OPFL2) et les initialiser (LSEEK), pour lire (REDFLD) et écrire (WRTFLD) aux fichiers, et finalement les fermer (CLFIL2). Ces routines font partie de la librairie de routines "FILEV20".

ROUTINE: CLBUF

SOURCE: MESSAGE.FOR

CATEGORIE: G

ARGUMENTS: CALL CLBUF

DESCRIPTION:

Cette sous-routine vide le "buffer" du clavier.

ROUTINE: CLSG

SOURCE: CLSG.ASM

CATEGORIE: G

ARGUMENTS: CALL CLSG

DESCRIPTION:

Cette sous-routine vide l'écran de la partie graphique, par un "scroll" complet.

ROUTINE: CODPOS

SOURCE: CODPOS.FOR

CATEGORIE: C

ARGUMENTS: CALL CODPOS

DESCRIPTION:

Cette sous-routine est le programme de contrôle pour le module "CODES-POSTAUX". Elle appelle des routines de plus bas niveau, en fonction des options choisies sur la souris (variable IFLAG).

Lorsque IFLAG vaut "1", il s'agit de l'insertion d'un nouveau point, la routine "AJOUTC" est alors appelée pour afficher le code-postal. Si tout est en ordre, "DSCCOD" est appelée pour écrire le code postal au fichier. Lorsque l'utilisateur rentre ses codes-postaux à l'aide d'un fichier, il peut utiliser la touche "2" afin de sauter un code-postaux et la touche "7" pour revenir au début du fichier. Si IFLAG vaut "0", il s'agit d'une destruction de point. Ce dernier est définie par l'utilisateur et une vérification est faite. Si IFLAG vaut "A", l'utilisateur demande un changement de fenêtre d'affichage ou zoom. La routine "QWINDW" est alors appelée pour redéfinir la fenêtre, et "CADRE" et "MAPNOD" sont appelées pour respectivement dessiner le cadre extérieur et les points existants.

Si IFLAG vaut "C", l'utilisateur veut changer la fenêtre extérieure; une question est alors posée pour savoir s'il veut revenir à la fenêtre précédente: si OUI, "POSCAR" est appelée avec IOPTION=2; si NON, "POSCAR" est appelée avec IOPTION=0 pour redéfinir la position de la carte. Le tout se poursuit comme avec "A".

Si IFLAG vaut "#", l'utilisateur veut modifier la valeur de la tolérance pour la recherche (variable TOL).

Si IFLAG vaut "6", l'utilisateur veut voir le menu des commandes de la souris. L'écran graphique est alors sauvegardé et le menu rappelé par la routine "GRAF". Quand l'utilisateur fait "return", l'écran graphique est ramené.

Lorsque les traitements sont complétés, le contrôle est retourné à la sous-routine "INICOD" pour recevoir une autre commande de la souris.

ROUTINE: COPY

SOURCE: COPY.FOR

CATEGORIE: G

ARGUMENTS: CALL COPY(F1,F2)

DESCRIPTION:

Cette sous-routine recopie le fichier "F1" dans le fichier "F2". Elle fonctionne sur le même principe que la routine "CHGNOD".

Des routines qui appellent l'interrupt "21H" de DOS sont appelées pour établir le nom du fichier (ASCIIZ), pour établir l'adresse de transfert (SETXFR), pour créer le fichier (CREAT), pour ouvrir les fichiers (OPFL2) et les initialiser (LSEEK), pour lire (REDFLD) et écrire (WRTFLD) aux fichiers, et finalement les fermer (CLFIL2). Ces routines font partie de la librairie de "FILEV20".

ROUTINE: COPZON

SOURCE: COPZON.FOR

CATEGORIE: Z

ARGUMENTS: CALL COPZON

DESCRIPTION:

Cette sous-routine est appelée quand l'usager veut recopier une partie de zone. Elle demande à l'usager le numéro de zone à recopier et détermine alors si le point pointé est bien un sommet. Elle demande ensuite de pointer le point terminal et d'indiquer si la chaîne d'arêtes à être recopiée correspond à celle du sens de rotation des aiguilles d'une montre ou le sens inverse. Pour calculer le sens de la rotation de la zone pointée, il utilise la formule qui calcule la surface du polygone. Si le total est négatif, le polygone est dans le sens des aiguilles d'une montre Si il est positif c'est dans le sens contraire. Les sommets concernés sont alors placés dans le vecteur "GTAB".

N.B. Des cas spéciaux existent pour les premier et dernier points d'une zone.

ROUTINE: CREAD

SOURCE: CREAD.ASM

CATEGORIE: G

ARGUMENTS: CALL CREAD(IASCI)

DESCRIPTION:

Cette sous-routine lit le caractère "IASCI" à la position courante du curseur à l'écran.

ROUTINE: CURLOC

SOURCE: CURLOC.ASM

CATEGORIE: G

ARGUMENTS: CALL CURLOC(IR,IC,PAGE)

DESCRIPTION:

Cette sous-routine place le curseur à l'écran, à la ligne "IR", à la colonne "IC", et sur la page "PAGE" du "buffer" vidéo.

ROUTINE: CURRET

SOURCE: CURRET.ASM

CATEGORIE: G

ARGUMENTS: CALL CURRET(IR,IC,PAGE)

DESCRIPTION:

Cette sous-routine retourne la position du curseur à l'écran; "IR" est la ligne, "IC" la colonne et "PAGE" la page active du "buffer" vidéo.

ROUTINE: CWRIT

SOURCE: CWRIT.ASM

CATEGORIE: G

ARGUMENTS: CALL CWRIT(COLOR,CHAR,PAGE)

DESCRIPTION:

Cette sous-routine affiche le caractère "CHAR" à l'écran, avec la couleur "COLOR" et sur la page "PAGE" du "buffer" vidéo.

ROUTINE: DATET

SOURCE: DATE.FOR

CATEGORIE: G

ARGUMENTS: CALL DATET

DESCRIPTION:

Cette sous-routine affiche la date, en mode "texte", dans le coin inférieur gauche.

ROUTINE: DATEG

SOURCE: DATE.FOR

CATEGORIE: G

ARGUMENTS: CALL DATEG

DESCRIPTION:

Cette sous-routine affiche la date, en mode graphique, dans le coin inférieur droit.

ROUTINE: DGBEEP

SOURCE: SOURIS.FOR

CATEGORIE: G

ARGUMENTS: CALL DGBEEP

DESCRIPTION:

Cette sous-routine fait sonner la cloche du digimètre.

ROUTINE: DGCOOR

SOURCE: SOURIS.FOR

CATEGORIE: G

ARGUMENTS: CALL DGCOOR(XR,YR,IFLAG,CODE)

DESCRIPTION:

Cette sous-routine détermine la touche qui a été actionnée sur la souris du digimètre (IFLAG), retourne les coordonnées correspondantes (XR,YR) et le mode de fonctionnement de la table numérisante (CODE)

Elle est appelée exclusivement par la routine "SOURIS".

ROUTINE: DGERRO

SOURCE: SOURIS.FOR

CATEGORIE: G

ARGUMENTS: CALL DGERRO

DESCRIPTION:

Cette sous-routine affiche un message lorsqu'une erreur de type "hardware" se produit au digimètre.

ROUTINE: DGINIT

SOURCE: SOURIS.FOR

CATEGORIE: G

ARGUMENTS: CALL DGINIT

DESCRIPTION:

Cette sous-routine initialise le digimètre. Il identifie le port de communication pour la table numérisante. Vérifie la communication entre le PC et la table. Puis transmet les données pour l'initialisation de la table.

ROUTINE: DGMODE

SOURCE: SOURIS.FOR

CATEGORIE: G

ARGUMENTS: CALL DGMODE

DESCRIPTION:

Cette sous-routine établit le mode de fonctionnement du digimètre.

ROUTINE: DISTNC

SOURCE: ROTATEX.FOR

CATEGORIE: G

ARGUMENTS: CALL DISTNC(X,Y,X1,Y1,IVOI)

DESCRIPTION:

Cette sous-routine calcule la distance entre deux points. Si les deux points sont trop rapprochés (écart inférieur à la tolérance), le drapeau "IVOI" sera levé.

ROUTINE: DSCCOD

SOURCE: DSCCOD.FOR

CATEGORIE: C

ARGUMENTS: CALL DSCCOD

DESCRIPTION:

Cette sous-routine place les coordonnées dans le vecteur GTAB et le numéro de noeud dans le vecteur IREC. Les valeurs ASCII de chaque caractère du code-postal sont additionnées et le total est inséré dans le vecteur IREC. Les informations sont écrites au fichier de travail (unité 10), au record courant.

ROUTINE: DSCLNK

SOURCE: DSCLNK.FOR

CATEGORIE: L

ARGUMENTS: CALL DSCLNK(I)

DESCRIPTION:

Cette sous-routine interroge l'utilisateur pour enregistrer les informations sur le lien courant. Le numéro du lien est calculé et validé. Si le paramètre (J) est égal à 1, alors il s'agit d'une modification d'un lien existant, auquel les valeurs par défaut deviennent les anciennes.

ROUTINE: DSCNOD

SOURCE: DSCNOD.FOR

CATEGORIE: N

ARGUMENTS: CALL DSCNOD

DESCRIPTION:

Cette sous-routine interroge l'utilisateur, sauf si IFLAG est égal à "8", pour enregistrer les informations sur le noeud courant. Elle place alors les coordonnées dans le vecteur GTAB et le numéro de noeud dans le vecteur IREC. S'il s'agit d'une modification (paramètre MODI), les informations sont écrites au fichier de travail (unité 10), au record courant. Autrement, elles sont ajoutées à la fin du fichier.

ROUTINE: ERASE

SOURCE: INFLD.FOR

CATEGORIE: G

ARGUMENTS: CALL ERASE(IR,IC,JC)

DESCRIPTION:

Cette sous-routine efface un champs de l'écran, à la ligne "IR", entre les colonnes "IC" et "JC".

ROUTINE: ETR

SOURCE: ROTATEX.FOR

CATEGORIE: G

ARGUMENTS: CALL ETR(XR,YR,XG,YG)

DESCRIPTION:

Cette sous-routine convertit les coordonnées réelles "carte" (XR,YR), en coordonnées graphiques "table" (XG,YG).

ROUTINE: FICHIERS

SOURCE: FICHIERS.FOR

CATEGORIE: G

ARGUMENTS: CALL FICHIERS

DESCRIPTION:

Cette sous-routine s'occupe de la gestion des fichiers. Il affiche un menu et lit le nom du fichier. L'utilisateur choisit entre créer un fichier BACKUP ou ASCII, imprimer le fichier à l'imprimante, afficher le fichier à l'écran, changer de fichier ou revenir au menu principal.

Le fichier BACKUP est recopié avec la sous-routine COPY.

Le nom du fichier ASCII est déterminé avec la sous-routine NOMEXT.

Le fichier ASCII et le fichier binaire est ensuite ouvert,

Les fichiers binaire et LPT1 sont ouverts. Le fichier binaire est lu et retranscrit dans le fichier LPT1, qui envoie les données à l'imprimante.

Le fichier binaire est ouvert, lu et affiché à l'écran. La sous-routine (AFFICH) gère l'affichage des records à l'écran.

ROUTINE: FRMCOD

SOURCE: FRMCOD.FOR

CATEGORIE: C

ARGUMENTS: CALL FRMCOD

DESCRIPTION:

Cette sous-routine est appelée à la fin d'une session dans le module "CODPOS" pour fermer le fichier des codes-postaux. Elle lit, record par record, le fichier de travail (unité 10) et le recopie dans le fichier de base des codes-postaux.

Une vérification est faite pour les codes-postaux ayant été détruits, car leurs coordonnées sont égales à "0" et le record ne serait pas recopier.

La sous-routine tient l'utilisateur informé sur le déroulement du processus, en indiquant le nombre de records recopiés et détruits.

ROUTINE: FRMLNK

SOURCE: FRMLNK.FOR

CATEGORIE: L

ARGUMENTS: CALL FRMLNK

DESCRIPTION:

Cette sous-routine est appelée à la fin d'une session dans le module "LIENS" pour fermer le fichier des liens. Elle lit, record par record, le fichier de travail (unité 12) et le recopie dans le fichier de base des liens (unité 13).

Une vérification est faite pour les liens ayant été détruits (numéro de noeud mis à 32767). S'il s'en trouve, le record correspondant n'est pas recopié.

La routine tient l'utilisateur informé sur le déroulement du processus, en indiquant les records recopiés et détruits.

ROUTINE: FRMNOD

SOURCE: FRMNOD.FOR

CATEGORIE: N

ARGUMENTS: CALL FRMNOD

DESCRIPTION:

Cette sous-routine est appelée à la fin d'une session dans le module "NOEUDS" pour fermer le fichier des noeuds. Elle lit, record par record, le fichier de travail (unité 10) et le recopie dans le fichier de base des noeuds (unité 11).

Une vérification est faite pour les noeuds ayant été détruits (numéro de noeud mis à 32767). S'il s'en trouve, le record correspondant n'est pas recopié.

La routine tient l'utilisateur informé sur le déroulement du processus, en indiquant les records recopiés et détruits.

ROUTINE: FRMSOM

SOURCE: FRMSOM.FOR

CATEGORIE: Z

ARGUMENTS: CALL FRMSOM

DESCRIPTION:

Cette sous-routine est appelée à la fin d'une session dans le module "ZONES" pour fermer le fichier des sommets. Elle lit, record par record, le fichier de travail (unité 10) et le recopie dans le fichier de base des sommets (unité 13).

Une vérification est faite pour les sommets devenus inutiles (numéro de sommet mis à 0). S'il s'en trouve, le record correspondant n'est pas recopié.

La routine tient l'utilisateur informé sur le déroulement du processus, en indiquant les records recopiés et détruits.

ROUTINE: FRMZON

SOURCE: FRMZON.FOR

CATEGORIE: Z

ARGUMENTS: CALL FRMZON

DESCRIPTION:

Cette sous-routine est appelée à la fin d'une session dans le module "ZONES" pour fermer le fichier des zones. Elle lit, record par record, le fichier de travail (unité 12) et le recopie dans le fichier de base des zones (unité 11).

Une vérification est faite pour les zones devenues inutiles (numéro de zone mis à 0). S'il s'en trouve, le record correspondant n'est pas recopié.

La routine tient l'utilisateur informé sur le déroulement du processus, en indiquant les records recopiés et détruits.

ROUTINE: GETDAT

SOURCE: GETDAT.ASM

CATEGORIE: G

ARGUMENTS: CALL GETDAT(J,M,A)

DESCRIPTION:

Cette sous-routine retourne la date courante du système d'opération.

ROUTINE: GETMOD

SOURCE: GETMOD.ASM

CATEGORIE: G

ARGUMENTS: CALL GETMOD(MODE)

DESCRIPTION:

Cette sous-routine retourne le mode courant de l'écran.

ROUTINE: GRAF

SOURCE: GRAF.ASM

CATEGORIE: G

ARGUMENTS: CALL GRAF(OPT)

DESCRIPTION:

Cette sous-routine sauve ou retourne le contenu de la mémoire vidéo (video-RAM) en mémoire courante. On indique OPT=0 pour sauver une image et OPT=1 pour la rappeler.

ROUTINE: IDFLNK

SOURCE: IDFLNK.FOR

CATEGORIE: L

ARGUMENTS: CALL IDFLNK

DESCRIPTION:

Cette sous-routine identifie un lien en faisant appel à "IDFNOD" pour localiser le noeud de départ, suivi d'un appel à "SOURIS" pour relever le deuxième noeud et d'un dernier appel à "IDFNOD" pour identifier le noeud d'arrivé.

ROUTINE: IDFNOD

SOURCE: IDFNOD.FOR

CATEGORIE: N

ARGUMENTS: CALL IDFNOD(X,Y,N)

DESCRIPTION:

Cette sous-routine identifie le noeud pointé au digimètre, aux coordonnées réelles (X,Y), en tenant compte de la tolérance. Elle retourne alors son numéro "N", ainsi que ses véritables coordonnées.

Si aucun noeud n'est trouvé à l'intérieur de la tolérance, l'argument "N" prend la valeur "0".

FONCTION: ILIEN

SOURCE: ILIEN.FOR

CATEGORIE: L

ARGUMENTS: ILIEN(L)

DESCRIPTION:

Cette fonction retourne le numéro de record où se trouve le lien "NOLIEN" dans le fichier. Ce numéro de record correspond à l'indice du tableau IRECL. Si le lien n'existe pas, la fonction retourne la valeur "0".

ROUTINE: INDSOU

SOURCE: INDSOU.FOR

CATEGORIE: G

ARGUMENTS: CALL INDSOU

DESCRIPTION:

Cette sous-routine affiche dans le coin inférieur droit, l'indice utilisation de la souris "<+>", si le paramètre "CA" est égal à "ON". Et efface l'indice si le paramètre est égal à "FF".

ROUTINE: INFLD

SOURCE: INFLD.FOR

CATEGORIE: G

ARGUMENTS: CALL INFLD(LENFLD,LENSTR,STR)

DESCRIPTION:

Cette sous-routine lit à l'écran la chaîne de caractères "STR", de longueur "LENSTR", dans un champs de longueur "LENFLD".

ROUTINE: INICOD

SOURCE: INICOD.FOR

CATEGORIE: C

ARGUMENTS: CALL INICOD

DESCRIPTION:

Cette sous-routine est le programme d'initialisation et de fermeture d'une session dans le module "CODE-POSTAUX".

Elle exécute successivement les tâches suivantes:

- Initialise les valeurs de défaut;
- Définit les fenêtres hardware et software;
- Appelle l'affichage du cadre extérieur (CADRE);
- Détermine les dimensions de la table numérisante.
- Détermine le type d'écran monocrome ou couleur.
- Ouvre les fichiers de codes-postaux et de travail;
- Copie les points existants (s'il y a lieu) en mémoire et dans le fichier de travail (CHGNOD);
- Détermine la position de la carte (POSCAR);
- Affiche le menu du digimètre (MENU DG);
- Affiche les codes-postaux existants (MAPNOD)(s'il y a lieu);
- L'utilisateur indique s'il veut rentrer les codes-postaux au clavier ou avec l'aide d'un fichier.
- Si l'utilisateur utilise un fichier, le fichier est validé, puis la touche de la souris est lue (SOURIS).
- Si l'utilisateur rentre les codes-postaux au clavier, le code-postal est lu et validé (VALCOD).
- Appelle la sous-routine "CODPOS" qui contrôle le choix de l'utilisateur.

La dernière routine est appelée tant que l'option "D" (sortie) n'est pas choisie. Les fichiers sont alors fermés et le contrôle est retourné au programme principal "DIGIT".

ROUTINE: INILNK

SOURCE: INILNK.FOR

CATEGORIE: L

ARGUMENTS: CALL INILNK

DESCRIPTION:

Cette sous-routine est le programme d'initialisation et de fermeture d'une session dans le module "LIENS".

Elle exécute successivement les tâches suivantes:

- Initialise les valeurs de défaut;
- Définit les fenêtres hardware et software;
- Appelle l'affichage du cadre extérieur (CADRE);
- Teste si le fichier TEMP.DTA existe, si oui un message est affiché et la session est terminée.
- Détermine la dimension de la table numérisante.
- Détermine le type d'écran monocrome ou couleur.
- Ouvre les fichiers de noeuds, de liens et de travail;
- Copie les noeuds existants (s'il y a lieu) en mémoire et dans le fichier de travail (CHGNOD);
- Copie les liens existants (s'il y a lieu) en mémoire et dans le fichier de travail (CHGLNK);
- Détermine la position de la carte (POSCAR);
- Affiche le menu du digimètre (MENUUDG);
- Affiche les noeuds existants (MAPNOD) (s'il y a lieu);
- Affiche les liens existants (MAPLNK) (s'il y a lieu);
- Appelle la sous-routine "LIEN" qui contrôle les interventions de l'utilisateur à partir de la souris.

La dernière routine est appelée tant que l'option "D" (sortie) n'est pas choisie. Les fichiers sont alors fermés et le contrôle est retourné au programme principal "DIGIT".

ROUTINE: ININOD

SOURCE: ININOD.FOR

CATEGORIE: N

ARGUMENTS: CALL ININOD

DESCRIPTION:

Cette sous-routine est le programme d'initialisation et de fermeture d'une session dans le module "NOEUDS".

Elle exécute successivement les tâches suivantes:

- Initialise les valeurs de défaut;
- Définit les fenêtres hardware et software;
- Appelle l'affichage du cadre extérieur (CADRE);
- Teste si le fichier TEMP.DTA existe, si oui un message est affiché et la session est terminée.
- Détermine la dimension de la table numérisante.
- Détermine le type d'écran, monocrome ou couleur.
- Ouvre les fichiers de noeuds et de travail;
- Copie les noeuds existants (s'il y a lieu) en mémoire et dans le fichier de travail (CHGNOD);
- Détermine la position de la carte (POSCAR);
- Affiche le menu du digimètre (MENU DG);
- Affiche les noeuds existants (MAPNOD) (s'il ya lieu);
- Appelle la sous-routine "NOEUD" qui contrôle les interventions de l'utilisateur à partir de la souris.

La dernière routine est appelée tant que l'option "D" (sortie) n'est pas choisie. Les fichiers sont alors fermés et le contrôle est retourné au programme principal "DIGIT".

ROUTINE: INIZON

SOURCE: INIZON.FOR

CATEGORIE: Z

ARGUMENTS: CALL INIZON

DESCRIPTION:

Cette sous-routine est le programme d'initialisation et de fermeture d'une session dans le module "ZONES".

Elle exécute successivement les tâches suivantes:

- Initialise les valeurs de défaut;
- Définit les fenêtres hardware et software;
- Appelle l'affichage du cadre extérieur (CADRE);
- Détermine les dimensions de la table numérisante.
- Détermine le type d'écran, monocrome ou couleur.
- Ouvre les fichiers de zones, de sommets et de travail;
- Copie les zones existantes (s'il y a lieu) en mémoire et dans le fichier de travail (CHGZON);
- Détermine la position de la carte (POSCAR);
- Affiche le menu du digimètre (MENUDG);
- Affiche les zones existantes (MAPZON);
- Appelle la sous-routine "ZONE" qui contrôle les interventions de l'utilisateur à partir de la souris.

La dernière routine est appelée tant que l'option "D" (sortie) n'est pas choisie. Les fichiers sont alors fermés et le contrôle est retourné au programme principal "DIGIT".

ROUTINE: INT14

SOURCE: INT14.ASM

CATEGORIE: G

ARGUMENTS: CALL INT14(INTYPE,CHR)

DESCRIPTION:

Cette sous-routine fait la communication avec un port RS-232C;

Le "high-byte" de "INTYPE" correspond au numéro du port (0 ou 1), tandis que le "low-byte" vaut "1" pour écrire, "2" pour lire ou "3" pour obtenir le statut du port.

La variable "CHR" est le caractère transmis.

ROUTINE: IOPORT

SOURCE: IOPORT.ASM

CATEGORIE: G

ARGUMENTS: CALL IOPORT(IOTYPE,STRING)

DESCRIPTION:

Cette sous-routine fait la communication avec un port RS-232C;

Le "high-byte" de "IOTYPE" correspond au numéro du port (0 ou 1), tandis que le "low-byte" indique la longueur de la chaîne de caractères "STRING" à transmettre.

ROUTINE: IRECO

SOURCE: IRECO.ASM

CATEGORIE: G

ARGUMENTS: CALL IRECO(IND,IREC,IRES,NOEUD)

DESCRIPTION:

Cette sous-routine cherche la position du noeud "NOEUD", dans le vecteur "IREC", de dimension "IND". "IRES" est la position retournée, et vaut "0" si le noeud est inexistant.

FONCTION: IRECORD

SOURCE: IRECORD.FOR

CATEGORIE: G

ARGUMENTS: IRECORD(N)

DESCRIPTION:

Cette fonction retourne le numéro de record où se trouve le noeud "N" dans le fichier. Ce numéro de record correspond à l'indice du tableau GTAB. Si le noeud n'existe pas, la fonction retourne la valeur "0".

ROUTINE: KBREAD

SOURCE: KBREAD.ASM

CATEGORIE: G

ARGUMENTS: CALL KBREAD(IASCI)

DESCRIPTION:

Cette sous-routine lit le "buffer" du clavier pour identifier la touche actionnée "IASCI".

ROUTINE: LIEN

SOURCE: LIEN.FOR

CATEGORIE: L

ARGUMENTS: CALL LIEN

DESCRIPTION:

Cette sous-routine est le programme de contrôle pour le module des "LIENS". Elle appelle des routines de plus bas niveau, en fonction des options choisies sur la souris (variable IFLAG).

Lorsque IFLAG vaut "3" ou "4", le noeud de départ du lien a été pointé. Si IFLAG vaut "5", le lien sera de type "courbé" et la variable logique STR est alors mise à .TRUE. pour indiquer un traitement particulier. La routine "IDFNOD" est alors appelée pour reconnaître le noeud et les valeurs de défaut sont ajustées. S'il s'agit du mode "courbé", la routine "STREAM" est appelée pour codifier le lien et la routine "ARRIVE" est finalement appelée pour prendre la description du lien et l'inscrire au fichier.

Si IFLAG vaut "1", "2" ou "8", le noeud d'arrivée du lien a été pointé. Si IFLAG vaut "1", le lien est à sens unique. "IDFNOD" sera alors appelée pour reconnaître le noeud et "ARRIVE" pour prendre la description du lien. Si IFLAG vaut "2" le sens est bidirectionnel, un sous-dialogue sera affiché pour la description de la deuxième direction. Lorsque IFLAG vaut "8" il n'y a pas de description et le lien est à sens unique.

Si IFLAG vaut "9", l'utilisateur veut modifier la description d'un lien existant. La routine "IDFLNK" est alors appelée pour identifier ce lien, l'ancienne description est remplacée dans les valeurs par défaut courantes et "DSCLNK" est appelée avec un paramètre indiquant qu'il s'agit d'une modification.

Le traitement des touches "A", "C", "6" et "#" est identique à celui décrit pour la sous-routine "NOEUD".

Lorsque les traitements sont complétés, le contrôle est retourné à la sous-routine "INILNK" pour recevoir une autre commande de la souris.

ROUTINE: LINE1

SOURCE: LINE1.FOR

CATEGORIE: G

ARGUMENTS: CALL LINE1(XX,YY,XP,YP,IC)

DESCRIPTION:

Cette sous-routine dessine un trait à l'écran, entre (XX,XY) et (XP,YP); IC définit la commande comme "MOVE" ou "DRAW". Il fait une rotation translation des coordonnées et fait le trait avec QPLOT.

fonction: LOCA

SOURCE: IDFNOD.FOR

CATEGORIE: G

ARGUMENTS: LOCA(X,Y)

DESCRIPTION:

Cette fonction convertit les coordonnées de la table en coordonnées réelles (X,Y), et cherche dans le vecteur des coordonnées (TAB) un point situé en dedans de la tolérance.

Si un noeud ou sommet (selon le module) est trouvé, la fonction retourne l'indice correspondant du vecteur. Autrement, elle retourne la valeur "0".

ROUTINE: MAPGIN

SOURCE: MAPGIN.ASM

CATEGORIE: G

ARGUMENTS: CALL MAPGIN

DESCRIPTION:

Cette sous-routine deplace une croix comme curseur à l'écran.

ROUTINE: MAPLNK

SOURCE: MAPLNK.FOR

CATEGORIE: L

ARGUMENTS: CALL MAPLNK(I)

DESCRIPTION:

Cette sous-routine effectue l'affichage d'un ensemble de liens existants, à l'écran, à l'aide de la sous-routine "AFFLNK".

Elle donne à l'utilisateur l'option de limiter l'impression à un bloc de liens choisis, afin de diminuer le temps requis. L'utilisateur peut aussi interrompre l'affichage en cours de dessin.

Si les deux noeuds du lien ne font pas partie des noeuds déjà affichés à l'écran, "AFFLNK" n'est pas appelée, augmentant ainsi la vitesse d'exécution du dessin.

ROUTINE: MAPNOD

SOURCE: MAPNOD.FOR

CATEGORIE: N

ARGUMENTS: CALL MAPNOD(I)

DESCRIPTION:

Cette sous-routine effectue l'affichage d'un ensemble de noeuds existants, à l'écran, à l'aide de la sous-routine "AFFNOD".

Elle donne à l'utilisateur l'option de limiter l'impression à un bloc de noeuds choisis, afin de diminuer le temps requis. L'utilisateur peut aussi interrompre l'affichage en cours de dessin.

ROUTINE: MAPZON

SOURCE: MAPZON.FOR

CATEGORIE: Z

ARGUMENTS: CALL MAPZON

DESCRIPTION:

Cette sous-routine effectue l'affichage à l'écran du dessin des zones existantes. Il lit le fichier de zones, record par record, trace les arêtes entre les sommets et affiche le numéro de zone à son centre géométrique. Elle donne à l'utilisateur l'option de limiter l'impression à un bloc de noeuds choisis, afin de diminuer le temps requis. L'utilisateur peut interrompre l'affichage en cours de dessin. Les lignes d'arêtes sont tracées à l'aide de la routine "LINE1" et le texte est affiché à l'aide de "TKTEXT".

ROUTINE: MENU DG

SOURCE: MENU DG.FOR

CATEGORIE: G

ARGUMENTS: CALL MENU DG

DESCRIPTION:

Cette sous-routine affiche le menu de la souris du digimètre, correspondant au module courant. Les menus existent sur fichiers indépendants pour chaque module de PRODIG.

Une fois le module identifié, une vérification de l'existence du fichier correspondant est faite. En cas de problème, un message est affiché.

Le fichier est alors ouvert et les informations correspondantes aux records de ce fichier sont affichées.

Cette sous-routine n'est appelée qu'une fois par session dans un module donné. Une fois que l'image-écran est générée, elle est sauvée en mémoire par la routine "GRAF". Lorsque la touche "6" de la souris est actionnée, cette image est rappelée directement, sans repasser par le fichier de menu, ni par la présente sous-routine.

ROUTINE: MESSAGE

SOURCE: MESSAGE.FOR

CATEGORIE: G

ARGUMENTS: CALL MESSAGE(IL,IC,ICOL,IP,ARR)

DESCRIPTION:

Cette sous-routine affiche le message "ARR" d'une longueur maximum de 78. Si IL et IC sont égal à "0", alors la message seras écrit position courante du curseur. Chaque caractère est affiché avec la sous-routine "CWGIT".

ROUTINE: MODSET

SOURCE: MODSET.ASM

CATEGORIE: G

ARGUMENTS: CALL MODSET

DESCRIPTION:

Cette sous-routine établit le mode de l'écran.

ROUTINE: NOEUD

SOURCE: NOEUD.FOR

CATEGORIE: N

ARGUMENTS: CALL NOEUD

DESCRIPTION:

Cette sous-routine est le programme de contrôle pour le module des "NOEUDS". Elle appelle des routines de plus bas niveau, en fonction des options choisies sur la souris (variable IFLAG).

Lorsque IFLAG vaut "1", "2", "4", "5", "7" ou "8", il s'agit de l'insertion d'un nouveau noeud, qui aura comme attribut de type une valeur associée à IFLAG, de "1" à "6", respectivement. La routine "AJOUTN" est alors appelée pour obtenir le numéro de noeud. Si tout est en ordre, "DSCNOD" est appelée pour prendre la description des attributs, sauf si IFLAG est égal à "8". Si IFLAG vaut "9", il s'agit d'une modification de noeud. La routine "LOCA" est alors appelée pour convertir les coordonnées "table" en coordonnées "carte" et identifier le noeud pointé. Une question est alors posée pour déterminer s'il s'agit de changer le noeud de place;

- Si OUI, le noeud est détruit et "SOURIS" est appelée pour repérer la nouvelle position.
- Si NON, le système demande un nouveau numéro pour le noeud et reprend ses anciens attributs.
- Dans les deux cas, "DSCNOD" est appelée avec un paramètre signifiant qu'il s'agit d'une modification de description.

Si IFLAG vaut "0", il s'agit d'une destruction de noeud. Ce dernier est localisé et une vérification avec l'utilisateur est faite avant que ne soit appelée "ZAPNOD" pour l'éliminer.

Si IFLAG vaut "A", l'utilisateur demande un changement de fenêtre d'affichage ou zoom. La routine "QWINDW" est alors appelée pour redéfinir la fenêtre, et "CADRE" et "MAPNOD" sont appelées pour respectivement dessiner le cadre extérieur et les noeuds existants.

Si IFLAG vaut "C", l'utilisateur veut changer la fenêtre extérieure; une question est alors posée pour savoir s'il veut revenir à la fenêtre précédente: si OUI, "POSCAR" est appelée avec IOPTION=2; si NON, "POSCAR" est appelée avec IOPTION=0 pour redéfinir la position de la carte. Le tout se poursuit comme avec "A".

Si IFLAG vaut "#", l'utilisateur veut modifier la valeur de la tolérance pour la recherche (variable TOL).

Si IFLAG vaut "6", l'utilisateur veut voir le menu des commandes de la souris. L'écran graphique est alors sauvegardé et le menu rappelé par la routine "GRAF". Quand l'utilisateur fait "return", l'écran graphique est ramené.

Lorsque les traitements sont complétés, le contrôle est retourné à la sous-routine "ININOD" pour recevoir une autre commande de la souris.

ROUTINE: NOME XT

SOURCE: NOMFCH.FOR

CATEGORIE: G

ARGUMENTS: CALL NOME XT(A,B,EXT)

DESCRIPTION:

Cette sous-routine ajoute l'extension "EXT" au nom du fichier "A".
Les deux vecteurs sont validés et insérés caractère par caractère
dans le vecteur "B".

ROUTINE: NOMFCH

SOURCE: NOMFCH.FOR

CATEGORIE: G

ARGUMENTS: CALL NOMFCH

DESCRIPTION:

Cette sous-routine lit un nom de fichier. Un message est affiché à l'écran, indiquant à l'utilisateur le ou les extensions que vous devez prendre les noms des fichiers. Elle lit le nom du fichier (INFLD). Puis elle termine en rajoutant l'extension au nom du fichier (NOMEXT).

ROUTINE: NOMSEP

SOURCE: NOMFCH.FOR

CATEGORIE: G

ARGUMENTS: CALL NOMSEP(A,NOM,EXT)

DESCRIPTION:

Cette sous-routine sépare le nom du fichier "A" en une partie "NOM" et une partie "EXT", correspondant à l'extension.

ROUTINE: NUMBER		
SOURCE:	INFLD.FOR	CATEGORIE: G
ARGUMENTS: CALL NUMBER(STR,LENSTR,NUMB,IVALID)		
DESCRIPTION: Cette sous-routine convertit la chaîne de caractères "STR", de longueur "LENSTR", en une valeur "NUMB" entière de 4 octets. Si IVALID = 1, il y avait un non-chiffre dans la chaîne.		

ROUTINE: OFFCB

SOURCE: CB.ASM

CATEGORIE: G

ARGUMENTS: CALL OFFCB

DESCRIPTION:

Cette sous-routine désactive la commande "Ctrl-Break" du clavier.

ROUTINE: OFFCLOCK

SOURCE: TIME.ASM

CATEGORIE: G

ARGUMENTS: CALL OFFCLOCK

DESCRIPTION:

Cette sous-routine désactive la routine "TIME".

ROUTINE: OFFSCR

SOURCE: PSCR.ASM

CATEGORIE: G

ARGUMENTS: CALL OFFSCR

DESCRIPTION:

Cette sous-routine désactive le mode "CALCOMP" de la fonction "Prt-Sc" du clavier.

ROUTINE: ONCB

SOURCE: CB.ASM

CATEGORIE: C

ARGUMENTS: CALL ONCB

DESCRIPTION:

Cette sous-routine active la commande "Ctrl-Break" du clavier.

ROUTINE: ONCLOCK

SOURCE: TIME.ASM

CATEGORIE: G

ARGUMENTS: CALL ONCLOCK

DESCRIPTION:

Cette sous-routine active la routine "TIME".

ROUTINE: ONSCR

SOURCE: PSCR.ASM

CATEGORIE: G

ARGUMENTS: CALL ONSCR

DESCRIPTION:

Cette sous-routine active le mode "CALCOMP" sur la fonction "Prt-Sc" du clavier, qui appelle ainsi la routine "PSCR".

ROUTINE: OPTION

SOURCE: PRODIG.FOR

CATEGORIE: G

ARGUMENTS: CALL OPTION(I)

DESCRIPTION:

Cette sous-routine affiche le menu secondaire à l'écran et permet à l'utilisateur de choisir certaines options. Le contrôle dépend de l'option choisie ("I").

L'option "1" alterne le contrôle entre le digimètre et le clavier lui-même. Les deux premiers octets du bloc "common /INP/" indiquent l'état courant. Une valeur de (1) indique que le digimètre est actif, tandis que (-1) signifie que le contrôle est assigné au clavier.

L'option "2" contrôle le port d'entrée assigné au digimètre. Une valeur de (0) dans les deux premiers octets du bloc /PORT/ indique que COM1: est actif; une valeur de (1) est associée au port COM2: .

L'option "3" appelle l'exécution du module choisi préalablement, au menu principal (routines "NOEUD", "LIEN", "CODPOS" et "ZONE").

L'option "0" retourne l'utilisateur au menu principal.

ROUTINE: OUTFLD

SOURCE: INFLD.FOR

CATEGORIE: G

ARGUMENTS: CALL OUTFLD(LENFLD,STR)

DESCRIPTION:

Cette sous-routine écrit à l'écran la chaîne de caractères "STR", dans un champs de longueur "LENFLD". Ceci remplace un énoncé WRITE de FORTRAN.

ROUTINE: PAGEN

SOURCE: PAGEN.ASM

CATEGORIE: G

ARGUMENTS: CALL PAGEN

DESCRIPTION:

Cette sous-routine établit la page active courante pour l'écran.

ROUTINE: PCCROS

SOURCE: PCCROS.FOR

CATEGORIE: G

ARGUMENTS: CALL PCCROS(X,Y,IAL)

DESCRIPTION:

Cette sous-routine remplace la routine "TKCROS" de la librairie "QPLOT". Elle affiche un curseur à l'écran (PCGIN). Puis dépendamment de la touche pesé par l'utilisateur, le curseur se déplace vers le bas, le haut, la droite ou la gauche.

ROUTINE: PCGIN

SOURCE: PCGIN.ASM

CATEGORIE: G

ARGUMENTS: CALL PCGIN(X,Y,DIR,INC,BITMAP)

DESCRIPTION:

Cette sous-routine déplace une croix blanche en lieu de curseur. (résolution 640 x 200), à la position (X,Y), à partir du clavier.

"DIR" est la direction (U,D,R,L) et "INC" l'incrément de déplacement (4 ou 20 pixels). "BITMAP" garde en mémoire l'image sous la croix avant son passage et l'image est restituée après ce passage.

Cette routine est utilisée avec "PCCROS".

ROUTINE: POSCAR

SOURCE: POSCAR.FOR

CATEGORIE: G

ARGUMENTS: CALL POSCAR

DESCRIPTION:

Cette sous-routine supervise le positionnement de la carte sur le digimètre. Tous ses paramètres sont passés en "COMMON".

La variable "IOPTION" détermine s'il s'agit de repérer la carte sur la table ("0") ou de revenir à la fenêtre originale ("2"). Autrefois, IOPTION=1 était utilisé pour faire un zoom à partir du digimètre; PRODIG fabrique maintenant les zooms à partir du clavier.

D'autres paramètres indiquent si on est en mode calvier ou en mode digimètre ("NN"), si c'est une première session de travail ("REPS") et dans quel module on se trouve ("IPROG").

Les variables pour lesquelles des valeurs sont retournées par la routine sont les suivantes:

- X, Y, X1, Y1 : coordonnées des points inférieur-gauche et supérieur-droit sur le digimètre;
- XMIN, YMIN : coordonnée inférieure-gauche, en unités de la carte réelle;
- GAMMA : angle en radians, entre les systèmes de coordonnées "table" et "carte";
- F1, F2 : échelles horizontale et verticale.

Les valeurs originales de ces variables (avant changement de la fenêtre d'affichage) sont emmagasinées respectivement dans XT, YT, XT1, YT1, XMIN1, YMIN1, GAMMA1, FX et FY.

Si "POSCAR" est appelée avec IOPTION=2, ces valeurs originales sont replacées dans les variables actives et la fenêtre est redéfinie, à l'aide de "QPLOT".

Si "POSCAR" est appelée avec IOPTION=0 alors que l'entrée se fait en mode "clavier", le système demande à l'utilisateur d'indiquer les coins inférieur-gauche et supérieur-droit en coordonnées "carte". Si l'entrée est assignée au digimètre, et qu'il s'agit d'une première session de travail, l'utilisateur devra pointer deux points sur la carte et en donner les coordonnées. Si ce n'est pas une première session, l'utilisateur a aussi la possibilité de repérer la carte en pointant deux noeuds existants, dont le système cherchera les coordonnées dans le fichier.

Dans tous les cas où IOPTION=0, une vérification est faite des échelles horizontale et verticale et de l'angle. Si le système décèle une difficulté (échelles très différentes, division par zéro, etc.), ou si l'utilisateur le demande à travers le dialogue de vérification, le processus de repérage sera recommencé.

ROUTINE: PRECIS

SOURCE: PRECIS.FOR

CATEGORIE: G

ARGUMENTS: CALL PRECIS

DESCRIPTION:

Cette sous-routine permet à l'utilisateur de définir une fenêtre de "précision", correspondant à une partie de la carte constituant un agrandissement d'un secteur donné.

Le traitement qui est fait est identique à celui décrit pour la routine "POSCAR".

PROGRAMME: PRODIG

SOURCE: PRODIG.FOR

CATEGORIE: G

ARGUMENTS: -- PROGRAMME PRINCIPAL --

DESCRIPTION:

Ceci est le programme principal, à la base de toutes les opérations du logiciel PRODIG.

Il affiche le menu principal à l'écran, à partir du fichier séquentiel "MENU.DAT" et demande à l'utilisateur d'identifier le module qu'il veut utiliser. La réponse est emmagasinée dans les deux premiers octets du bloc "COMMON /DD/". La clé de correspondance est la suivante:

- 1- Module "NOEUDS"
- 2- Module "LIENS"
- 3- Module "CODES-POSTAUX"
- 4- Module "ZONES"
- 5- Module "FICHIERS"

Le contrôle est alors passé à la routine "OPTION".

ROUTINE: PSCR

SOURCE: PSCR.ASM

CATEGORIE: C

ARGUMENTS: CALL PSCR

DESCRIPTION:

Cette sous-routine remplace l'interrupt "Prt-Sc" du clavier sous DOS pour rediriger le contenu graphique dans un fichier de commandes type "Calcomp".

ROUTINE: RDBUF

SOURCE: RDBUF.ASM

CATEGORIE: G

ARGUMENTS: CALL RDBUF

DESCRIPTION:

Cette sous-routine vérifie si une touche du clavier a été actionnée et la retourne le cas échéant.

ROUTINE: READC

SOURCE: MESSAGE.FOR

CATEGORIE: G

ARGUMENTS: CALL READC(C,J)

DESCRIPTION:

Cette sous-routine lit une chaîne de caractères au clavier;
"C" est le nom du vecteur de caractères (50A1) et "J" sa
longueur nette.

ROUTINE: ROTATEX

SOURCE: ROTATEX.FOR

CATEGORIE: G

ARGUMENTS: CALL ROTATEX(XI, YI, ANGLE, XOR, YOR, XO, YO)

DESCRIPTION:

Cette sous-routine effectue la transformation du système de repérage cartographique, avec rotation des axes de "ANGLE" degrés et translation de l'origine à (XOR, YOR).

ROUTINE: RTE

SOURCE: ROTATEX.FOR

CATEGORIE: G

ARGUMENTS: CALL RTE(XG,YG,XR,YR)

DESCRIPTION:

Cette sous-routine convertit en coordonnées réelles "carte" (XR,YR), les coordonnées graphiques "table" (XG,YG).

ROUTINE: SKROL

SOURCE: SKROL.ASM

CATEGORIE: G

ARGUMENTS: CALL SKROL

DESCRIPTION:

Cette sous-routine opère le déroulement de la partie "texte" de l'affichage.

ROUTINE: SOMMET

SOURCE: SOMMET.FOR

CATEGORIE: Z

ARGUMENTS: CALL SOMMET

DESCRIPTION:

Cette sous-routine est appelée chaque fois que la touche "1" est actionnée au digimètre, dans le module "ZONES".

Elle commence par appeler "LOCA" pour convertir les coordonnées et vérifier si le sommet existe, à l'intérieur de la tolérance spécifiée. Si c'est le cas, elle affiche un message et sonne la cloche du terminal. Une vérification est alors faite pour vérifier s'il s'agit du début d'une zone. Si oui, un traitement additionnel est fait.

Une autre vérification est faite pour déterminer si la zone vient d'être refermée. Si ce n'est pas le cas, le sommet est inscrit au fichier, en même temps qu'au vecteur "GTAB", et une ligne est tracée au sommet précédent.

Si la zone est refermée, la description des attributs est prise, le numéro de zone est affiché à l'aide de "TKTEXT" et les informations sont inscrites au fichier des zones.

ROUTINE: SOUND

SOURCE: SOUND.ASM

CATEGORIE: C

ARGUMENTS: CALL SOUND

DESCRIPTION:

Cette sous-routine actionne l'émission d'un signal sonore par le haut-parleur du micro-ordinateur.

ROUTINE: SOURIS

SOURCE: SOURIS.FOR

CATEGORIE: G

ARGUMENTS: CALL SOURIS(XR,YR,IFLAG,IMODE)

DESCRIPTION:

Cette sous-routine contrôle les informations captées par la souris du digimètre.

Elle retourne les coordonnées (XR,YR) pointées au digimètre, avec l'identificateur (IFLAG) de la touche qui a été actionnée au même moment.

Elle déplace, simultanément aux mouvements de la souris, le curseur affiché à l'écran (PCGIN).

Le mode courant du digimètre (IMODE) est aussi indiqué avec chaque transmission.

ROUTINE: STREAM

SOURCE: STREAM.FOR

CATEGORIE: L

ARGUMENTS: CALL STREAM(X,Y)

DESCRIPTION:

Cette sous-routine est appelée pour la digitalisation des liens courbes. Ce mode est appelé "stream", un flot continu de points étant transmis par le digimètre.

Chaque fois qu'un point étant à une distance égale ou supérieure à la tolérance ("TOL") du point précédent, il est enregistré dans le fichier de points intermédiaires (unité 14). Le lien est complété quand un noeud existant est reconnu.

Le record du fichier des liens indique la présence d'un lien courbe ("stream") si les noeuds de départ et d'arrivée sont négatifs. Il y a alors pointeur au premier record correspondant du fichier des points intermédiaires.

ROUTINE: TIME

SOURCE: TIME.ASM

CATEGORIE: G

ARGUMENTS: CALL TIME

DESCRIPTION:

Cette sous-routine affiche l'heure continuellement à l'écran et est activée par la routine "ONCLOCK". Elle est effectuée à chaque fois où l'horloge interne interrompt la tâche en cours.

ROUTINE: VADRS

SOURCE: VADRS.ASM

CATEGORIE: G

ARGUMENTS: CALL VADRS(AD)

DESCRIPTION:

Cette sous-routine retourne l'adresse d'une variable.

ROUTINE: WRDOT

SOURCE: WRDOT.ASM

CATEGORIE: G

ARGUMENTS: CALL WRDOT(X,Y,CHR,COL,FONT)

DESCRIPTION:

Cette sous-routine écrit le caractère "CHR" en mode graphique, point par point, aux coordonnées (X,Y). La couleur "COL" est toujours mise à "1". Les caractères standards sont indiqués par FONT=0, tandis que FONT=1 appelle des petits chiffres seulement (pas de lettres).

ROUTINE: ZAPLNK

SOURCE: ZAPLNK.FOR

CATEGORIE: L

ARGUMENTS: CALL ZAPLNK(NOLIEN)

DESCRIPTION:

Cette sous-routine détruit le lien "NOLIEN" dans le fichier des liens et l'efface de l'écran.

Elle appelle "AFFLNK" avec un paramètre lui indiquant que le lien doit être détruit, et met ensuite le noeud de départ de ce lien à 32000, une valeur spéciale qui indique que le record doit être détruit. Ceci sera fait par la routine "FRMLNK", à la fin de la session de travail.

ROUTINE: ZAPNOD

SOURCE: ZAPNOD.FOR

CATEGORIE: G

ARGUMENTS: CALL ZAPNOD(NOEUD)

DESCRIPTION:

Cette sous-routine élimine le noeud "NOEUD" à l'effaçant de l'écran à l'aide de la routine "AFFNOD" et en indiquant qu'il doit être détruit du fichier en le renumérotant à 32767. Ceci sera fait par la routine "FRMNOD".

ROUTINE: ZONE

SOURCE: ZONE.FOR

CATEGORIE: Z

ARGUMENTS: CALL ZONE

DESCRIPTION:

Cette sous-routine est le programme de contrôle pour le module des "ZONES". Elle appelle des routines de plus bas niveau, en fonction des options choisies sur la souris (variable IFLAG).

Lorsque IFLAG vaut "1" il s'agit de l'insertion d'un point, alors la sous-routine "SOMMET" est appelé.

Si IFLAG vaut "5", il s'agit de recopié une partie d'une autre zone (COPZON). uméro de

Si IFLAG vaut "6" le menu est réaffiché à l'écran.

Si IFLAG vaut "8" l'utilisateur a l'option de détruire un ou plusieurs

Si IFLAG vaut "0", il s'agit d'une destruction de noeud. Ce dernier est localisé et une vérification avec l'utilisateur est faite avant que ne soit appelée "ZAPNOD" pour l'éliminer.

Si IFLAG vaut "A", l'utilisateur demande un changement de fenêtre d'affichage ou zoom. La routine "QWINDW" est alors appelée pour redéfinir la fenêtre, et "CADRE" et "MAPNOD" sont appelées pour respectivement dessiner le cadre extérieur et les noeuds existants.

Si IFLAG vaut "C", l'utilisateur veut changer la fenêtre extérieure; une question est alors posée pour savoir s'il veut revenir à la fenêtre précédente: si OUI, "POSCAR" est appelée avec IOPTION=2; si NON, "POSCAR" est appelée avec IOPTION=0 pour redéfinir la position de la carte. Le tout se poursuit comme avec "A".

Si IFLAG vaut "#", l'utilisateur veut modifier la valeur de la tolérance pour la recherche (variable TOL).

Lorsque les traitements sont complétés, le contrôle est retourné à la sous-routine "INILNK" pour recevoir une autre commande de la souris.

ANNEXE C

Description des fichiers

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: XXXXXXXX.COD

DATE: 87/01/06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: LE FICHIER CONTIENT DES CODES-POSTAUX ET LEURS
COORDONNEES.

ORGANISATION:

- sequentiel
- sequentiel indexe
- acces direct
- acces direct indexe
- autre: _____

FORMAT:

- Fixe
- Variable

ESPACE:

KB []

Longueur du record(LRECL): [20]

ALLOCATION:

- Fixe
- Dynamique

Type de fichier:

- Binaire
- ASCII
- EBCDIC

VOLUME:

- Disquette
- Cartouche iomega
- Ruban magnetique
- Hard disk
- Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position		Type	Long
		De	A		
C10101E1P		1	6	A	6
Description		LE CODE-POSTAL			
X		7	10	N	4
Description		LA COORDONNEE X DU CODE-POSTAL			
Y		11	14	N	4
Description		LA COORDONNEE Y DU CODE-POSTAL			
01111111111111111111		5	10	X	6
Description		CHAMP INITIALISE POUR L'INSTANT.			

Type: A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . N O D

DATE: 87/01/01

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI MICRO

DESCRIPTION: LE FICHIER CONTIENT LES NOEUDS, AINSI QUE SES
COORDONNEES ET SA DESCRIPTION.

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [43]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position		Type	Long
		De	A		
I, N, O		1	2	N	2
Description		LE NUMERO DE NOEUD.			
X		3	6	N	4
Description		COORDONNEE X DU NOEUD.			
Y		7	10	N	4
Description		COORDONNEE Y DU NOEUD.			
OIEISI1		11	15	A	5
Description		LE NOM DE LA RUE PRINCIPALE DU NOEUD.			
OIEISI2		16	20	A	5
Description		LE NOM DE LA RUE SECONDAIRE DU NOEUD.			
TIYP		21	21	A	1
Description		LE TYPE D'ENDROIT OÙ EST LE NOEUD.			
I		22	23	N	2
Description		TYPE DE NOEUD.			

Type: A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . L N K

DATE: 87 / 01 / 06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: LE FICHIER CONTIENT LES LIENS AINSI QUE LES MOEUD AUQUEL
ILS SE RATACHENT.

ORGANISATION:

- sequentiel
- sequentiel indexe
- acces direct
- acces direct indexe
- autre: _____

FORMAT:

- Fixe
- Variable

ESPACE:

KB []

Longueur du record(LRECL): [14]

ALLOCATION:

- Fixe
- Dynamique

Type de fichier:

- Binaire
- ASCII
- EBCDIC

VOLUME:

- Disquette
- Cartouche iomega
- Ruban magnetique
- Hard disk
- Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____ DATE: ___/___/___

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position	Type	Long
	De	A		
N10DIEP Description LE NUMERO DU NEUD DE DEPART DU LIEN.		1	2 N	2
N10IARI Description LE NUMERO DU NEUD D'ARRIVE DU LIEN.		3	4 N	2
I1DV Description TYPE DE ROUTE.		5	7 N	2
N10IVI Description C'EST LE NOMBRE DE VOIES.		7	8 N	2
D11SITI Description LA DISTANCE ENTRE LES DEUX NEUDS.		9	12 N	4
I1ORAIPI Description TYPE DE LIEN		13	14 N	2
Description				

Type:A: alphanumerique F: flottant I: fixe N: numerique
B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: x x x x x x x x . Z O N

DATE: 87 / 03 / 06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: LE FICHIER CONTIENT LES ZONES ET LES POINTS QUI DEFINIS-
SENT CHAQUES ZONES.

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [92]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___ / ___ / ___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position	Type	Long
	De	A		
U0		1	2 N	9
Description				
I		3	4 N	9
Description				
I TAB ₁ (L)		5	6 N	2
Description				

Type: A: alphanumérique F: flottant I: fixe N: numérique
B: binaire H: hexadécimale P: numérique condensée X: alphanumérique

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Description	Signe	Position		Type	Long
			De	A		
BILANIC	Dix ESPACES.		11	10	X	10
IJAIB(4)	LE NUMERO D'UN POINT DE LA ZONE LES 15 PROCHAINS BMS SONT IDEATIQUES.		12	12	N	12

Type:A: alphanumerique F: flottant I: fixe N: numerique
B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . S O M

DATE: 27 / 05 / 06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: LE FICHIER CONTIENT DES COORDONNEES DE POINTS.

ORGANISATION:

sequentiel

sequentiel indexe

acces direct

acces direct indexe

autre: _____

FORMAT:

Fixe

Variable

ESPACE:

KB []

Longueur du record(LRECL): [12]

ALLOCATION:

Fixe

Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette

Cartouche iomega

Ruban magnetique

Hard disk

Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position		Type	Long
		De	A		
X		1	4	N	4
Description		COORDONNEE X D'UN POINT.			
Y		5	8	N	4
Description		COORDONNEE Y D'UN POINT.			
DDMMYY		9	12	X	4
Description		CHAMP INITIALISE POUR INSTANT			
Description					
Description					
Description					
Description					
Description					

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . I N T

DATE: 87 / 01 / 06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: LE FICHIER CONTIENT DES COORDONNEES DE POINTS
INTERMEDIAIRES POUR FORMER UN LIEU COURBE.

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [14]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position		Type	Long
		De	A		
X R		1	4	N	4
Description		COORDONNEE X D'UN POINT			
Y R		5	8	N	4
Description		COORDONNEE Y D'UN POINT			
D I V M M Y		9	14	X	6
Description		CHAMP UTILISE SUR L'INSTANT			
Description					
Description					
Description					
Description					
Description					

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: XXXXXXXXX.A5N

DATE: 87/01/02

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST LE FICHIER DES NOEUDS EN ASCII

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [63]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position	Type	Long
	De	A		
DUMMY		1	1	1
Description			X	
INO		2	6	5
Description			A	
DUMMY		7	7	1
Description			X	
X		8	16	9
Description			N	
Y		17	25	9
Description			N	
DUMMY		26	26	1
Description			X	
DESI		27	41	15
Description			A	

Type: A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position De	Position A	Type	Long
DUMMIY		42	43	X	2
Description		DEUX ESPACES.			
DESIQ2		44	58	A	15
Description		C'EST LE NOM DE LA RUE SECONDAIRE DU NOEUD.			
DUMMIY		59	61	X	3
Description		TROIS ESPACES.			
TIYP		62	63	A	1
Description		LE TYPE D'ENDROIT QUI EST LE NOEUD.			
I		63	63	N	1
Description		LE TYPE DE NOEUD.			
Description					
Description					

Type:A: alphanumerique F: flottant I: fixe N: numerique
B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: x x x x x x x x . A S LDATE: 87 / 01 / 07

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -MINI MICRO DESCRIPTION: C'EST LE FICHIER DES LIENS EN ASCII.

ORGANISATION:

sequentiel
 sequentiel indexe
 acces direct
 acces direct indexe
 autre: _____

FORMAT:

Fixe
 Variable

ESPACE:

KB []

Longueur du record(LRECL): [38]

ALLOCATION:

Fixe
 Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
 Cartouche iomega
 Ruban magnetique
 Hard disk
 Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position De	Position A	Type	Long
Q U M M I Y		1	1	X	1
Description	UN ESPACE				
I		2	6	N	5
Description	LE NUMERO DU NOEUD DE DEPART DU LIEN.				
Q U M M I Y		7	7	X	1
Description	UN ESPACE				
J		8	12	N	5
Description	LE NUMERO DU NOEUD D'ARRIVE DU LIEN.				
K		13	17	N	5
Description	TYPE DE ROUTE.				
L		18	19	N	2
Description	C'EST LE NOMBRE DE VOIES.				
E		20	33		14
Description	LA LONGUEUR DU LIEN.				

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position De	Position A	Type	Long
D U M M Y					
Description UN ESPACE		314	317	X	1
M					
Description TYPE DE LIEN		315	318	N	4
Description					

Type:A: alphabetique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . A S Z

DATE: 27 / 01 / 07

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST LE FICHIER DE ZONE EN ASCII.

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [74]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position	Type	Long
	De	A		
N10				
Description		1	6	A
LE NUMERO DE LA ZONE.				
I				
Description		7	10	A
LE NUMBRE DE POINTS DE LA ZONE.				
ZONES.				
I I A B (L)				
Description		11	14	A
LE NUMERO D'UN POINT DE LA ZONE.				
LES 15 PROCHAINS CHAMPS SONT				
IDENTIQUES.				
Description				
Description				
Description				
Description				

Type: A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position	Type	Long
	De	A		
DIUMIMYI				
Description	SIX ESPACES.		X	
ITAB(L)				
Description	LE NUMERO D'UN POINT DE LA ZONE			
	LES 15 PROCHAINES CHAMPS SONT			
	IDENTIQUES.			
Description				
Description				
Description				
Description				
Description				

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: x x x x x x x x . A S S

DATE: 87 / 01 / 07

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST LE FICHIER DE SOMMETS EN ASCII.

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB [_____]

Longueur du record(LRECL): [37]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position	Type	Long
	De	A		
1 N 0				
Description		1	N	6
0 U M M Y				
Description		7	X	7
X				
Description		8	N	10
Y				
Description		8	N	10

Type:A: alphabetique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . A S I

DATE: 87 / 01 / 07

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST LE FICHIER DES POINTS INTERMEDIAIRE EN ASCII.

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [27]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___ / ___ / ___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position	Type	Long
	De	A		
QUMMY		1	X	1
Description		UN ESPACE		
LINDI		2	N	6
Description		C'EST LE NUMERO DE POINT INTERMEDIAIRE.		
XIR		18	N	10
Description		COORDONNEE X DU POINT INTERMEDIAIRE.		
YIR		18	N	10
Description		COORDONNEE Y DU POINT INTERMEDIAIRE.		
Description				
Description				
Description				
Description				
Description				

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: x x x x x x x x . A S C

DATE: 87/01/07

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST LE FICHIER DES CODES-POSTAUX EN ASCII.

ORGANISATION:

sequentiel
 sequentiel indexe
 acces direct
 acces direct indexe
 autre: _____

FORMAT:

Fixe
 Variable

ESPACE:

KB []

Longueur du record(LRECL): [99]

ALLOCATION:

Fixe
 Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
 Cartouche iomega
 Ruban magnetique
 Hard disk
 Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position	Type	Long
	De	A		
DUMMIY		1	X	1
Description				
UN ESPACE				
CODP		2	X	6
Description				
C'EST LE CODE - POSTAL				
DUMMIY		2	X	2
Description				
DEUX ESPACE				
X		10	N	10
Description				
COORDONNEE X DU CODE-POSTAL				
Y		20	N	10
Description				
COORDONNEE Y DU CODE-POSTAL				
Description				
Description				

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: x x x x x x y x . P L TDATE: 27/01/07

DIRECTORY: _____

NO-CLASSIFICATION
- - - - -MINI MICRO DESCRIPTION: C'EST LE FICHIER DE COMMANDE POUR LE CALCOMP 1075.

ORGANISATION:

sequentiel
 sequentiel indexe
 acces direct
 acces direct indexe
 autre: _____

FORMAT:

Fixe
 Variable

ESPACE:

KB []

Longueur du record(LRECL): [80]

ALLOCATION:

Fixe
 Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
 Cartouche iomega
 Ruban magnetique
 Hard disk
 Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS

Nom du Champ	Signe	Position		Type	Long
		De	A		
Description <u>C'EST UNE COMMANDE POUR LE</u> <u>CALCUL P.</u> <u>C'EST IDENTIQUE POUR LES AUTRES CHAMPS.</u>		17	21	N	9
Description					

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: M E N U N _ _ _ . D A TDATE: 27 / 01 / 02

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -MINI MICRO DESCRIPTION: C'EST UN FICHIER QUI CONTIENT UN MENU A FAIRE
AFFICHER A L'ECRAN.C'EST IDENTIQUE POUR LES FICHIERS MENU1.DAT, MENU2.DAT,
MENU3.DAT, MENU4.DAT ET MENU5.DAT

ORGANISATION:

sequentiel
 sequentiel indexe
 acces direct
 acces direct indexe
 autre:

FORMAT:

Fixe
 Variable

ESPACE:

KB []

Longueur du record(LRECL): [20]

ALLOCATION:

Fixe
 Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
 Cartouche iomega
 Ruban magnetique
 Hard disk
 Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

**DESCRIPTION DES CHAMPS
DES FICHIERS PERMANENTS**

Nom du Champ	Signe	Position		Type	Long
		De	A		
X					
Description		1	6	A	6
COORDONNEE X EN PICSELLE A L'ECRAN.					
Y					
Description		7	12	A	6
COORDONNEE Y EN PICSELLE A L'ECRAN.					
SIZE					
Description		13	16	A	4
LA LONGUEUR DE LA SUITE DE CARACTERE.					
NC					
Description		17	19	A	3
LA COULEUR DES CARACTERES					
TEXT(I)					
Description		20	20	A	1
UN CARACTERE A AFFICHER A L'ECRAN.					
C'EST IDENTIQUE POUR LE RISIE.					
Description					
Description					

Type:A: alphanumerique F: flottant I: fixe N: numerique
 B: binaire H: hexadecimal P: numerique condense X: alphanumerique

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X Y X Y X X . B K Z

DATE: 87 / 03 / 06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -MINI MICRO

DESCRIPTION: C'EST LE FICHIER "BACKUP" DU FICHIER DE ZONES.
POUR LES CHAMPS DES RECORDS, VOIR XXXXXXXX, ZON

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre:

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [47]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . D K SDATE: 27 / 03 / 06

DIRECTORY: \ _____

NO-CLASSIFICATION

MINI MICRO

DESCRIPTION: C'EST LE FICHIER "BACK UP" DU FICHIER DE SOMMETS,
POUR LES CHAMPS DES RECORDS, VOIR XXXXXXXX, SON

ORGANISATION:

sequentiel

sequentiel indexe

acces direct

acces direct indexe

autre: _____

FORMAT:

Fixe

Variable

ESPACE:

KB [_____]

Longueur du record(LRECL): [17]

ALLOCATION:

Fixe

Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette

Cartouche iomega

Ruban magnetique

Hard disk

Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: XXXXXXXXX.BKIDATE: 87/03/06

DIRECTORY: \ _____

NO-CLASSIFICATION
-----MINI MICRO

DESCRIPTION: C'EST LE FICHIER "BACKUP" DU FICHIER INTERMEDIAIRE
POUR LES CHAMPS DES RECORDS, VOIR XXXXXXXX.INT

ORGANISATION:

sequentiel

sequentiel indexe

acces direct

acces direct indexe

autre: _____

FORMAT:

Fixe

Variable

ESPACE:

KB []

Longueur du record(LRECL): []

ALLOCATION:

Fixe

Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette

Cartouche iomega

Ruban magnetique

Hard disk

Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: x x x x x x x . b k c

DATE: 87/02/06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST LE FICHIER "BACKUP" DU FICHIER DE CODE -
POSTAUX, POUR LES CHAMPS DES RECORDS, VOIR XXXXXX.COO

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [90]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X X . B K L

DATE: 87 / 01 / 06

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST LE FICHIER "BACKUP" DU FICHIER DE LIEN.
POUR LES CHAMPS DES RECORDS, VOIR XXXXXXXXX.LNK

ORGANISATION:

sequentiel
sequentiel indexe
acces direct
acces direct indexe
autre: _____

FORMAT:

Fixe
Variable

ESPACE:

KB []

Longueur du record(LRECL): [14]

ALLOCATION:

Fixe
Dynamique

Type de fichier:

Binaire ASCII EBCDIC

VOLUME:

Disquette
Cartouche iomega
Ruban magnetique
Hard disk
Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___/___/___

SYSTEME DE DOCUMENTATION
description des fichiers
mini et micro

NOM DU FICHIER: X X X X X X X . B K V

DATE: 82 / 01 / 02

DIRECTORY: \ _____

NO-CLASSIFICATION
- - - - -

MINI

MICRO

DESCRIPTION: C'EST UN FICHIER "BACKUP" DU FICHIER DE MARCHÉ
POUR LES CHAMPS DES RECORDS VOIR XXXXXXXXXXXX.NDD

ORGANISATION:

- sequentiel
- sequentiel indexe
- acces direct
- acces direct indexe
- autre: _____

FORMAT:

- Fixe
- Variable

ESPACE:

KB []

Longueur du record(LRECL): [43]

ALLOCATION:

- Fixe
- Dynamique

Type de fichier:

- Binaire
- ASCII
- EBCDIC

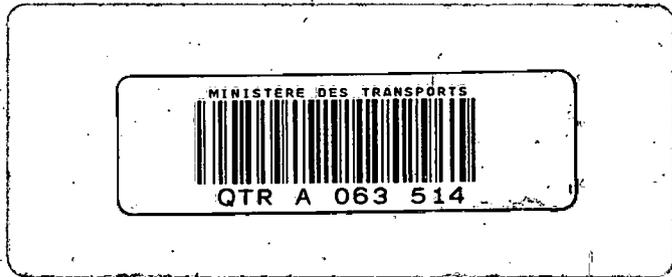
VOLUME:

- Disquette
- Cartouche iomega
- Ruban magnetique
- Hard disk
- Autre: _____

Nom ou No: _____

NOM DU RESPONSABLE: _____

DATE: ___ / ___ / ___



MINISTÈRE DES TRANSPORTS



QTR A 063 514