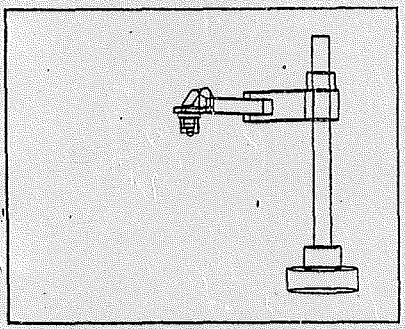
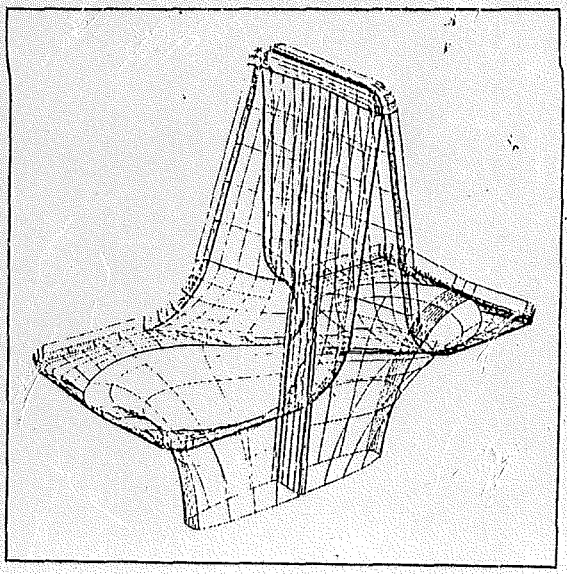
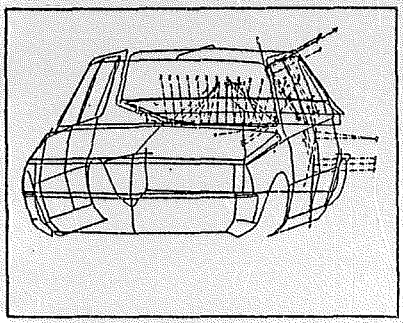
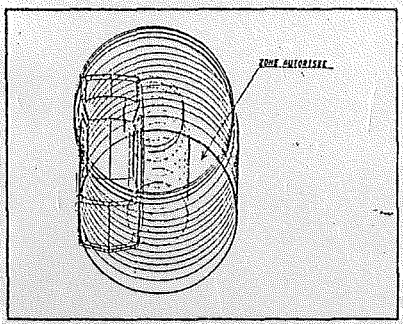
MEM/PLOT 6200

micro
informatique
service









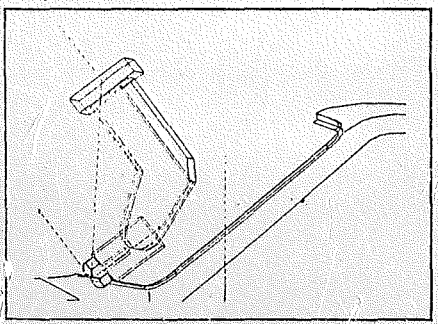


TABLE DES MATIERES

PROLOGUE

Vous avez fait l'acquisition d'un langage graphique très performant et très puissant, fonctionnant sous le système d'exploitation MEM/DOS 6502.

Nous vous félicitons et espérons que vous puissiez en tirer le meilleur parti.

Pour vous aider à vous familiariser avec MEM/PLOT 6502, nous avons écrit ce manuel de programmation qui vous fera, petit à petit, entrer dans le monde du traitement graphique.

Que vous soyez un novice ou un programmeur expérimenté, nous vous conseillons de lire attentivement ce manuel qui pourra servir par la suite de guide de programmation.

Le présent manuel est composé de deux parties :

La première partie vous fera découvrir MEM/PLOT 6502 en tant que langage graphique sous le système d'exploitation MEM/DOS 6502, vous vous familiariserez donc à sa philosophie, ce qui vous permettra d'attaquer la seconde partie sans trop de difficultés.

La deuxième partie concerne la programmation. Elle vous fera découvrir petit à petit les possibilités graphiques MEM/PLOT. Elle complètera ainsi le manuel de référence et le guide de programmation.

Tout au long du manuel, vous trouverez des exemples simples accompagnant les définitions.

Un manuel d'exemples de programmes permettant de compléter vos connaissances est en cours de réalisation.

Ce manuel s'adresse à toute catégorie d'utilisateur, l'aspect technique du produit n'est pas présenté ici.

Là encore, un manuel technique permettra au programmeur expérimenté d'interfacer lui-même des périphériques graphiques.

L'annexe A donne l'ensemble des notations utilisé dans le manuel.

CHAPITRE I : PRESENTATION GENERALE DU PRODUIT MEM/PLOT 6502

I - A - PRESENTATION GENERALE DU PRODUIT MEM/PLOT 6502 :

Si l'on doit donner une définition pour qualifier MEM/PLOT 6502, on emploiera le mot langage.

Pourtant, MEM/PLOT est plus qu'un simple langage de programmation puisqu'il sait gérer des périphériques graphiques, on pourrait donc l'associer à un système graphique comprenant deux couches dont l'une serait le langage, et l'autre, la gestion de l'environnement graphique.

MEM/PLOT fonctionnant sous MEM/DOS 6502, l'utilisateur possède un système très complet de 40 Koctets de soft en mémoire morte, lui permettant de faire du traitement graphique sur n'importe quelle mémoire de masse dans n'importe quelle configuration, que ce soit du monoposte, multiposte ou en réseau local réparti, sans oublier bien sur, la puissante gestion de fichiers et de masques du MEM/DOS 6502.

Il est donc certain que tout utilisateur y trouvera son compte, que ce soit le financier qui voudra analyser la croissance de sa société à l'aide de graphiques en histogramme, " camembert " ou courbe, ou le concepteur de petit circuit imprimé qui pourra modifier à volonté son circuit sur un écran graphique pour le sortir ensuite sur du mylar à l'aide d'une table traçante, ou enfin l'architecte bien content d'éditer ses plans sous n'importe quelle échelle avec des légendes de la taille et dans la direction souhaitées.

Il n'est certes pas possible d'énumérer la totalité des applications possibles de MEM/PLOT, tellement les possibilités sont grandes et dépassent même l'imagination de ceux qui, peut-être comme vous, ne connaissent pas encore MEM/PLOT 6502.

I - B - PHILOSOPHIE DU SYSTEME MEM/PLOT 6502 :

A l'image des handlers de MEM/DOS 6502 qui servent à gérer n'importe quel périphérique de masse, MEM/PLOT fonctionne aussi de cette manière. Il est composé d'un corps en ROM autour duquel il y a les handlers de gestion des périphériques graphiques.

Sous MEM/PLOT, on peut charger trois handlers en mémoire, ce qui donne accès à trois périphériques quasi simultanément. A chaque handler est associé un buffer de 130 octets, qui permet la sauvegarde et la restauration des contextes de chaque périphérique , c'est grâce à cela que l'on a accès aux trois handlers à la fois.

Les lecteurs intéressés par l'aspect technique de MEM/PLOT pourront se reporter au manuel technique.

CHAPITRE II : L'INSTALLATION ET LA MISE EN ROUTE DU PRODUIT MEM/PLOT 6502

II - A - GENERALITES :

MEM/PLOT 6502 se présente sous la forme d'une carte ROM enfichable dans l'un des slots de l'APPLE.

Pour " démarrer " cette carte, il faut utiliser la disquette de démarrage qui contient aussi les handlers des périphériques graphiques dont vous avez besoin.

N'oubliez pas que MEM/PLOT 6502 ne fonctionne que sous MEM/DOS 6502. Il faut par conséquent que la carte MEM/DOS (20 Ko), soit présente dans l'un des slots !

La disquette de démarrage lance en premier le système MEM/DOS, puis MEM/PLOT et charge les handlers.

II - B - INSTALLATION :

Ouvrez le capot de l'APPLE et placez la carte MEM/PLOT dans l'un des slots libres, sauf dans le slot = 7.

Vous avez donc à ce moment deux cartes ROM dans votre APPLE :

- une carte MEM/DOS 6502
- Une carte MEM/PLOT 6502

Enfichez s'il y a lieu une carte d'interfaçage pour gérer un périphérique externe (vous pouvez en mettre 1, 2 ou 3).

II - C - DEMARRAGE :

Booter la disquette de démarrage MEM/PLOT. Cette disquette contient les handlers dont vous avez besoin. Ces derniers s'autochargeront.

ATTENTION :

La disquette de démarrage MEM/PLOT démarre aussi le MEM/DOS. Si la carte MEM/PLOT est absente, une erreur sera générée. Peu de temps après, un masque de chargement des handlers apparaitra à l'écran et restera le temps du chargement des handlers.

Les opérations de chargement des handlers sont affichées dans le masque au fur et à mesure de leur bon déroulement.

Ensuite, l'écran se vide et le message :

MEM/PLOT 6502
(c) M.I.S.

apparait.

MEM/PLOT est booté, vous pouvez travailler.

Bonne chance ...

CHAPITRE III : INTRODUCTION AU LANGAGE MEM/PLOT 6502

III - A - GENERALITES :

Quand la carte MEM/PLOT est en service, le BASIC de l'APPLE possède un jeu très complet d'instructions graphiques en plus du standard APPLESOFT.

Pour les habitués de l'APPLESOFT, il est à signaler que les instructions graphiques telles que HGR, HCOLOR, TEXT, etc., sont toujours accessibles sous MEM/PLOT 6502 mais ne sont d'aucune utilité pour la programmation graphique. Leur utilisation conjointe avec les instructions de MEM/PLOT est d'ailleurs interdite car elles n'ont pas l'effet attendu par le système.

Le langage MEM/PLOT 6502 se rapproche de la nouvelle norme graphique ANSI qui pourrait prochainement devenir un standard ANSI.

Cependant, le jeu d'instructions graphiques de la norme a été complétée sous MEM/PLOT, de ce qui nous a semblé être indispensable pour une programmation aisée. Vous trouverez en annexe B , l'ensemble des commandes de MEM/PLOT.

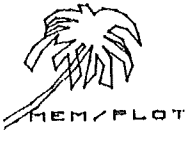
III - B - UN TOUR D'HORIZON RAPIDE DU SYSTEME MEM/PLOT 6502 :

Voici quelques exemples commentés qui vous permettront de juger immédiatement de la puissance du système.
Ces exemples ne suffisent pas à la compréhension du système. Vous vous appercevrez d'ailleurs avec ces exemples, que MEM/PLOT 6502 est un LANGAGE DE PROGRAMMATION GRAPHIQUE et non un simple logiciel d'application !

1) construisons un générateur aléatoitre d'aire pleine :

```
1  REM  £££ GENERATEUR ALEATOIRE D'AIRE PLEINE £££
10  GR "HA",1,1: REM  ON LANCE LE HANDLER 1
12 SET"DE",1,3: REM  ON ARME L'ECRAN
14 ASK"WI",G,B,C,D: REM  ON DEMANDE LA TAILLE DE L'ECRAN
16 SET"AREA",INT(RND(1)*G),INT(RND(1)*B),INT(RND(1)*C),INT(RND(1)*D):ON DEFINIT UNE
     AIRE ALEATOIRE
18  GR "C A","FILL": REM  ON LA REMPLIE
20  GOTO 16: REM  ON BOUCLE

]
```



2) traçons une sinusoîde amortie :

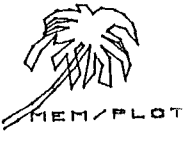
```
]LIST

1  REM  £££ TRACE D'UNE SINUSOIDE AMORTIE £££
10  GR "HA",1,1:SET"DE",1,3
12 SET"WINDOW", - 2 * 3.14,2 * 3.14, - 1.5,1.5: REM  MISE A L'ECHELLE
14 SET"TIC",10,10: REM  ON DEFINIT LA TAILLE DES BATONNETS
16  GR "XAXIS",0, - 2 * 3.14,2 * 3.14,3.14: REM  ON TRACE L'AXE DES X
18  GR "YAXIS",0, - 1.5,1.5,.5: REM  ON TRACE L'AXE DES Y
20  FOR X =  -6.28 TO 6.28 STEP .1
22  GR "PLOT",(X, SIN (X) / X):: NEXT X
24  END

]
```

3) Traçons des caractères alpha à l'écran :

```
1  REM  £££ TRACE DES CARACTERES DE L'ECRAN APPLE II £££
10  GR "HA",1,1:SET"DE",1,3
12 SET"WINDOW",0,10,0,10: REM  MISE A L'ECHELLE 10,10
14 SET"POSITION",2,8
16 SET"CHARACTER",.8,.5,1: REM  ON DEFINIT LA TAILLE DES CARACTERES
18 SET"TEXT ANGLE",0: REM  ON ECRIT HORIZONTALEMENT
20  GR "PRINT","MEM/PLOT 6502": REM  ON ECRIT MEM/PLOT 6502
22 SET"POS",5,5
24 SET"TEXT ANGLE",1.57
26  GR "PRINT","BONJOUR"
28  END
```



III - C - LE FORMAT DES INSTRUCTIONS SOUS MEM/PLOT :

Les instructions graphiques sont composées de trois parties :

un préfixe une commande des paramètres

III - C - 1 - LES PREFIXES :

Il y a trois préfixes possibles :

GR SET ASK

" SET " et " ASK " servent à mettre à jour les paramètres de tracé, ou à interroger l'état de ces derniers.

" GR " est utilisé pour toute instruction ayant une incidence directe sur le tracé ou les handlers.

On dira que SET et ASK sont des préfixes qui, employés avec les commandes adéquates, sont des instructions de gestion des paramètres graphiques.
Alors que GR donnera lui des instructions à incidence directe sur le tracé ou les handlers.

III - C - 2 - LA COMMANDE :

La commande peut être considérée comme l'instruction proprement dite.
Mais dans l'instruction générale composée de trois parties, la commande se comporte comme un paramètre :

Exemple :

Soit l'instruction : SET" DEVICE ", 1

- SET est un préfixe
- " DEVICE " est la commande
- 1 est le paramètre de la commande " DEVICE "

On pourrait donc faire :

```
D$ = " DEVICE " : D = 1

SET D$, D
```

D$ et D sont donc deux paramètres de l'instruction "SET" dont le premier est une commande.

Ainsi est-t-il possible de générer une séquence d'instruction graphique par un autre programme sous MEM/PLOT.

Dans la forme numéro 2, on voit l'intérêt de la virgule entre la commande et le paramètre. Celle-ci est absolument obligatoire quelque soit la forme de l'instruction utilisée.

Sous MEM/PLOT comme sous MEM/DOS, la commande peut-être abrégée pour faciliter la frappe.

Ainsi " DEVICE " devient " DE ", toute commande étant réductible à deux caractères minimum. Notez bien les cotes autour de la commande, elles sont obligatoires car une commande est toujours une chaîne de caractères.

```
Ainsi : SET " DE ",1
        SET " DEV ",1
        SET " DEVIC ",1
        SET " DEVICE ",1

       où  A = 1   B$ = " DEV " ----è    SET B$,A

       où  SET LEFT$(B$,2),A
```

ont la même signification: SET " DEVICE ",1

Il existe 32 commandes sous MEM/PLOT qui, combinées avec les préfixes SET, ASK et GR triplent quasiment le nombre des fonctions graphiques exécutables sous MEM/PLOT.
L'annexe ... donne la liste des instructions de MEM/PLOT et renvoie aux pages correspondantes de ce présent manuel, le lecteur désireux d'approfondir une fonction particulière.

III - C - 3 - LES PARAMETRES :

Les paramètres constituent la troisième partie des instructions de MEM/PLOT 6502.
Le nombre de paramètres varie selon l'instruction et vont en général de 0 à 4.
Chaque paramètre doit être séparé obligatoirement par une virgule.
Plusieurs paramètres constituent une " liste " de paramètres.

Sous MEM/PLOT 6502, il y a 4 types de paramètres :

- les flottants ou entiers
- les chaines de caractères
- les entiers compris entre 0 et 255
- les couples de coordonnées.

Tout paramètre peut être traité sous forme de variable.

En examinant la liste des paramètres, le lecteur sera peut être surpris par le dernier type de paramètre : couple de coordonnées !
Un couple de coordonnées s'écrit : (x,y) où x et y sont deux paramètres flottants.
Dans l'instruction de tracé GR " PLOT " les paramètres sont une liste de couples de coordonnées :
exemple GR " PLOT ", (x1,y1), (x2,y2), ..., (xn,yn).

ATTENTION :

En aucun cas on ne peut dire

```
A = (1,2)      GR "PLOT ",A
```

car BASIC ne sait pas reconnaître le type couple de coordonnées.
En fait, on peut associer ce type à une paire de flottants entre paranthèses séparées par une virgule.

Dans toutes les instructions de MEM/PLOT, les paramètres peuvent être des expressions calculées : exemple :

```
1) GR " PLOT ", (cos(I), sin(I))
                 expression calculée

2) A$ = " DEVICE N0 1 "
   SET LEFT$ (A$,2),1
   expression calculée

3) ASK " WINDOW ",A,B,C,D
   SET " VIEWPORT ",10,3aA,E,E * 2
```

MEM/PLOT effectue un contrôle sur le nombre et le type des paramètres à la saisie, et génère une erreur quand il y a incompatibilité.

III - D - UN LANGAGE ENTIEREMENT PARAMETRABLE :

MEM/PLOT 6502 fonctionne selon un système d'interrogation et de mise à jour des paramètres graphiques.

Il faut entendre par " paramètres graphiques " les éléments qui permettent la mise à l'échelle, qui indique la taille d'un écran ...

- l'interrogation de ces paramètres se fait par l'instruction " ASK "

- la mise à jour de ces paramètres se fait par l'instruction " SET "

Quand on arme un handler, ce dernier génère automatiquement les paramètres par défaut qui permettent à MEM/PLOT de faire les calculs.
Il est bien sûr possible de modifier ces paramètres par l'instruction " SET ".

Le chapitre VI décrit les techniques de mise à l'échelle d'un périphérique à l'autre en utilisant les valeurs par défaut.

III - E - MODE DIRECT OU PROGRAMME :

Toute instruction de MEM/PLOT peut s'exécuter soit en mode programmé, soit en direct. Sous MEM/DOS, pour travailler en direct, il faut exécuter l'instruction LET")M" pour ne pas que le système ferme automatiquement les fichiers.
Sous MEM/PLOT, l'utilisation du mode direct se confond totalement avec le mode programmé.
Le contexte de chaque périphérique étant sauvegardé dans un buffer, tous les paramètres sont protégés et ne peuvent être modifiés que volontairement. L'instruction RUN ou CLEAR du BASIC n'a aucune incidence sur ces paramètres car ils ne sont pas sauvegardés sous forme de variables BASIC.

La modification des paramètres de tracé se fait soit par l'instruction " SET " soit par le RESET de handler qui remet alors les paramètres d'origine par défaut.

Sous MEM/PLOT, le mode direct est en fait bien pratique, il permet de savoir à tout moment l'emplacement exact de la plume en " coordonnée du problème " par exemple, ou il permet d'aller modifier ou interroger l'ensemble des paramètres, ce qui est bien utile au niveau du debuggage des grosses applications graphiques.

Nous insistons sur le fait qu'il n'y a aucune barrière entre le mode direct et l'exécution programmée, à la différence que :

les erreurs sont automatiquement gérées en mode direct, alors qu'en mode programme, il est possible de les récupérer. (voir chapitre C.)

III - F - LA GESTION DES ERREURS :

En mode programme, si un "on err goto" a été exécuté en début de séquence, le système n'affiche pas l'erreur courante mais met seulement un code en 189 (peek (189) code).
(voir annexe C : les erreurs)

CHAPITRE IV : GENERALITES SUR LE BASIC GRAPHIQUE

Les instructions de MEM/PLOT 6502 qui viennent s'adjoindre au BASIC transforment ce dernier en un "BASIC GRAPHIQUE " orienté vers la nouvelle norme ANSI.

Cette norme introduit les notions générales qu'il est nécessaire de connaître pour comprendre la suite de ce manuel d'utilisation.

IV - A - L'AIRE DE TRACE GRAPHIQUE :

MEM/PLOT étant capable de piloter tous types de périphériques graphiques, il convient de généraliser le nom de l'aire de tracé qui peut être soit le plateau d'une table traçante, soit l'écran, soit le rouleau...
Le terme général que l'on emploiera par la suite est : "ECRAN", quelque soit l'aire de tracé du périphérique graphique.

IV - B - LE FAISCEAU :

Dans un même souci de standardisation, le faisceau ou "BEAM " en anglais, identifie à la fois la plume d'une table traçante, le point lumineux d'un écran ou le faisceau d'un crayon optique.
Sous MEM/PLOT, tous ces éléments ont une seule dénomination : BEAM.

IV - C - LE TRAITEMENT ALPHANUMERIQUE :

MEM/PLOT 6502 possède son propre jeu de caractères qui fonctionne sur tous les périphériques graphiques.
Là aussi, cela est fait dans le but de standardisation car le jeu de caractères inclus dans les tables traçantes ou les écrans ne sont jamais identiques.

Le jeu de MEM/PLOT permet la manipulation des caractères sous toutes leurs formes : hauteur, largeur, sens et direction de l'écriture ...

De plus, la taille des caractères est toujours proportionnelle à l'écran sur lequel on écrit.

Le chapitre VIII décrit le fonctionnement de ce jeu de caractères standard.

IV - D - SYSTEME DE COORDONNEES :

MEM/PLOT sait gérer les systèmes de coordonnées par des calculs d'échelle dépendant des paramètres saisis.

Quand on établit une échelle sous MEM/PLOT, on passe en "coordonnées problèmes" et toute référence aux coordonnées d'un point ou d'un vecteur sera faite dans cette échelle, jusqu'à ce que l'on change d'échelle.

IV - E - UN PEU DE VOCABULAIRE CONCERNANT "L'ESPACE GRAPHIQUE" :

Tout tracé graphique dans un écran se fait à l'intérieur d'une limite que l'on appelle " VIEWPORT ".

Cette notion s'applique à tout périphérique graphique actif sous MEM/PLOT 6502.

L'échelle qui vient se calquer sur le " VIEWPORT " s'appelle WINDOW.

Ce terme représente la " fenêtre " générale de tracé à l'intérieur d'un écran, cette aire pouvant elle-même devenir le VIEWPORT en cours. Cette aire se nomme : AREA.

Nous verrons par la suite, que la combinaison de "WINDOW" et de "AREA" permet de faire des zooms de sous-programmes graphiques.

Les trois commandes : VIEWPORT, WINDOW et AREA permettent donc la gestion de l'espace graphique sur tout périphérique actif sous MEM/PLOT 6502.

CHAPITRE V : LA PROGRAMMATION DU CONTEXTE

V - A - GENERALITES :

Sans cette phase, les instructions de MEM/PLOT ne s'adresseront à personne ou plutôt, à aucun périphérique.
La programmation d'un contexte consiste en la mise en route d'un ou de plusieurs handlers.

V - B - QU'EST-CE QU'UN HANDLER ?

Le handler, c'est tout simplement le programme qui gère le périphérique graphique. Chaque périphérique graphique est associé à un handler bien précis qui réalise l'interface entre lui-même et MEM/PLOT 6502.
Comme nous l'avons déjà cité plus haut, trois handlers peuvent coéxister en mémoire.
MEM/PLOT s'occupe de gérer ces handlers et les buffers associés aux contextes de chaque périphérique.
Quand on "boot" MEM/PLOT, on charge en ram les handlers, mais ceux-ci ne seront activés que sur demande d'un programme ou en mode direct par l'utilisateur ; sans cela, MEMP/PLOT ignorera ce handler et n'adressera donc jamais la parole au périphérique correspondant !
L'opération qui consiste à faire reconnaître à MEM/PLOT, l'existence d'un handler en mémoire s'appelle : LANCER un handler.
Une fois les handlers armés, il faut sélectionner un handler particulier pour travailler sur le périphérique concerné.
Cette opération s'appelle : ARMER un handler.
Trois opérations importantes sont donc nécessaires pour pouvoir travailler sous MEM/PLOT :

1- Charger un handler.
2- Lancer un handler.
3- Armer un handler.

L'opération 1 est en principe réalisé par le boot,
l'opération 2 se fait par la commande HANDLER,
l'opération 3 se fait par la commande DEVICE.

V - C - LA COMMANDE HANDLER :

V - C - 1 - SYNTAXE :

GR"HA[NDLE]",P1(,P2,...,Pn)

où P1 : numéro de handler. léPlé3

où P2,...Pn : paramètres propres au handler.

V - C - 2 - DESCRIPTION :

La commande "handler" lance un handler existant en ram donc préalablement chargé.
Le numéro de handler peut-être 1, 2 ou 3,
- en règle générale, le handler 1 est réservé à un écran graphique,
- le handler 2 est réservé à une table traçante
- et le handler 3 est libre : table traçante, écran graphique ou imprimante traçante...

Bien sûr, il s'agit là d'une configuration par défaut que l'utilisateur est libre de charger ; il devra pour cela, se reporter au manuel technique.

Le paramètre P1 représente le numéro du handler.
Les paramètres suivants P2,...,Pn dépendent du handler sélectionné.
En règle générale, s'il s'agit d'un périphérique extérieur à l'unité centrale comme une table traçante par exemple, le paramètre P2 représente un numéro de slot sur lequel est monté l'interface de communication : série, parallèle ou IEEE.
Pour le Handler de l'écran graphique de l'Apple II, P2 représente le numéro de page (1 ou 2) sur laquelle sera effectuée le traitement graphique.

V - C - 3 - EXEMPLES :

1) Lancement d'un handler écran : Apple II page 1

```
GR"HA",1,1
```

2) Lancement d'un handler de table traçante série connectée en slot 5 :

```
GR,"HA",2,5
```

V - C - 4 - INCIDENCE SUR LE SYSTEME :

Quand on lance un Handler par la commande "HANDLER", il se produit les évènements suivants :

1) Reset du périphérique associé au Handler :

- curseur ou plume en 0,0
- clear page graphique
- ...

2) Mise aux valeurs par défaut des paramètres graphiques correspondant au Handler lancé.

L'opération de lancement d'un handler est irrémédiable. C'est-à-dire que l'on ne peut annuler celui-ci sans rebooter le système.
Par contre, il est possible de lancer plusieurs fois le même handler, ce qui a pour effet de remettre à chaque fois les valeurs par défaut des paramètres concernés.

Ces paramètres sont :

Les quatre valeurs de VIEWPORT
Les quatre valeurs de WINDOW
Les quatre valeurs de AREA...
et tous les états des instructions à résultats binaires :
BEAM, CLIP, AREA...

Les chapîtres correspondants à chaque instruction donneront plus de détails sur ces valeurs par défaut.

Pour avoir la valeur des paramètres générés par défaut quand on lance un handler, il suffit d'utiliser l'instruction ASK "commande", ou la commande peut-être WINDOW, VIEWPORT, AREA...

Exemple : avoir lancer l'écran Apple II

```
par GR"HA",1,1

Si l'on tape ASK"VIEWPORT",A,B,C,D
on obtient : A = 0       B = 279
             C = 0       D = 191
```

(Voir le chapître concernant VIEWPORT pour avoir la signification de ces valeurs)

V - C - 5 - LE CONTROLE DES ERREURS :

P 1 : si P1 > 3 ou P1 < 0

illegal quantity error

si le handler invoqué n'existe pas

Handler not present error

P 2 : Pn : aucun contrôle car ces paramètres dépendent du Handler.

V - D - LA COMMANDE "DEVICE" :

V - D - 1 - SYNTAXE :

1) Avec le préfixe SET

SET"DEVICE"[,P1(,P2,...,Pn)]

2) Avec le préfixe ASK

ASK"DEVICE",V1

où P1 : numéro de device
P2,...,Pn : voir commande "Handler"

V1 : variable numérique

V - D - 2 - DESCRIPTION :

La commande "DEVICE" permet d'armer un handler pour pouvoir travailler sur le périphérique associé. C'est la troisième phase de mise en route (chargement, lancement, armement). "DEVICE" est la première commande que nous rencontrons, pouvant être employé à la fois avec "SET" et "ASK". Ceci demande un éclaircissement :

- toutes commandes ayant une incidence directe sur un dossier ou un handler s'utilisent avec le préfixe GR.

- toutes les autres commandes s'utilisent avec les préfixes "SET" et "ASK". Ces commandes servent à manipuler les variables.

Récapitulation :

Deux types de commandes :

1) Commande à incidence directe :

GR"commande", paramètre

2) Commande de manipulation des paramètres :

SET"commande", paramètre

ASK"commande", variables

Dans les commandes à incidence directe, les paramètres sont des variables que l'on donne au système.
Dans les commandes de manipulation des paramètres, les paramètres sont des variables que l'on donne au système dans le cas de l'utilisation avec le préfixe "SET", et se sont des variables que l'on récupère, venant du système, dans le cas de l'utilisation avec le préfixe ASK.

Exemple : SET"WINDOW",0,279,0,191

```
ASK"WINDOW",A,B,C,D
MEM/PLOT met dans A,B,C,D les valeurs de WINDOW

?A,B,C,D

0          279    0        191
(réponse de la machine)
```

V - D - 3 - EXEMPLES :

Après avoir lancé l'écran Apple II par l'instruction :

```
GR"HA",1,1
```

on arme ce handler par :

```
SET"DE",1,1
```

pour connaître le numéro de Handler en cours :

```
ASK"DE",X

?X
1  réponse de la machine
```

V - D - 4 - INCIDENCE SUR LE SYSTEME :

L'armement d'un Handler a pour but de faire envoyer tout traitement graphique vers le périphérique dont le Handler est armé.
Une fois un handler armé, les commandes à incidence directe sur le tracé seront envoyées vers le périphérique graphique et les commandes de manipulation des paramètres agiront sur le Buffer du handler correspondant.

Il est possible de désarmé un handler, pour cela, il suffit d'en armer un autre à la place.
On peut aussi armer deux handlers à la fois pour travailler sur deux périphériques simultanément, il faut alors taper :

```
SET"DEVICE",P1,P2
```

Notez que ASK"DEVICE", P1, P2 sont impossibles. On ne peut récupérer que la valeur du premier handler armé.
L'armement de deux handlers à la fois est une facilité qui doit être utilisé le plus rarement possible car le traitement graphique en est fort ralenti !

Pour désarmer tous les handlers, il faut taper :

```
SET"DE",0
```

- Supposons que l'on ait deux handlers en mémoire :

1 - écran Apple II
2 - table traçante, slot 5

Voici la manière de traiter un sous-programme graphique pour qu'il tourne avec les deux périphériques l'un après l'autre:

```
1 - GR"HA",1,1  : GR"HA",2.5 :

2 - SET"DE",1,1 : GOSUB 1000

3 - SET"DE",2.5 : GOSUB 1000

4 - SET"DE",0

5 - STOP

1000  en SOUS-PROGRAMME GRAPHIQUE
2000  RETURN
```

Commentaires :

Ligne 1 : on lance les deux handlers.

Ligne 2 : on arme le handler 1 écran et on exécute le sous-programme en 1000.

Ligne 3 : on arme le handler table ce qui désarme par défaut le handler écran et on exécute le sous-programme en 1000.

Ligne 4 : on désarme le handler 2 sans en réarmer un autre.

Ligne 1000 à 2000 : sous-programme graphique quelconque.

- Supposons maintenant que l'on veuille effectuer le traitement sur les deux périphériques en même temps :

```
1 - GR"HA",1,1 : GR"HA",2.5

2 - SET"DE",1,1 & 2.5 remplace les lignes 2 et 3
                      programme précédent.

3 - GOSUB 1000

4 - SET"DE",0
```

V - D - 5 - TRAITEMENT DES ERREURS :

C'est le même que pour la commande HANDLER.

V - E - LA COMBINAISON DES COMMANDES HANDLER ET DEVICE :

V - E - 1 - UNE DIFFERENCE FONDAMENTALE ENTRE CES DEUX COMMANDES :

Les paragraphes V-C et V-D vous ont faits découvrir les commandes de gestion du contexte de MEM/PLOT 6502.
Mais il ne faut pas confondre ces deux commandes qui se ressemblent beaucoup.
Handler ne s'exécute qu'une seule fois au début du traitement d'un programme.
S'il y a trois handlers en mémoire et que vous voulez utiliser les trois périphériques associés à ce handler, alors vous exécuterez 3 commandes "HANDLER" correspondant aux trois Handler en mémoire et c'est tout.
Device permet de commuter d'un Handler à l'autre, à condition que ce dernier ait été "lancé" par la commande "Handler" au préalable !

Ainsi ces deux instructions sont-elles indispensables pour le bon déroulement du traitement graphique.

Une erreur du type "Handler", not existant error, vient à coup sûr d'une mauvaise utilisation de ces 2 commandes.

V - E - 2 - COMMENT REGENERER LES PARAMETRES PAR DEFAUT DANS UN HANDLER PARTICULIER ?

Vous n'êtes pas sans savoir que les paramètres graphiques dans le buffer sont protégés de toute manipulation "Basic" tel que RUN CLEAR...
Pour régénérer les paramètres de base, il suffit de relancer le Handler par la commande "HANDLER".
C'est le seul cas de figure ou on peut relancer un handler, mais rappelez-vous : vous perdez l'ensemble des paramètres de votre problème qui sera remplacé par les paramètres de base !

Si vous êtes sûr d'avoir chargé un Handler, mais que le système renvoie une erreur quand vous essayez de l'armer ou de le lancer, c'est que ce dernier a été endommagé en mémoire, il faut donc le recharger.

Chapître VI : LA PROGRAMMATION DE L'ESPACE GRAPHIQUE

VI - A - GENERALITES :

La programmation de l'espace graphique concerne l'ensemble des commandes de MEM/PLOT qui définissent la taille et l'échelle d'un écran.

Il est peut-être utile de vous rappeler qu'un "écran" est employé ici en tant que terme général pour désigner une surface graphique qui a soit un véritable écran graphique ou le plateau d'une table traçante.

Ces commandes sont au nombre de cinq :

VIEWPORT : définition de la taille générale de l'écran.

WINDOW : définition d'une échelle.

AREA : définition d'une fenêtre interne.

C-AREA : commande de gestion de la fenêtre.

CLIP : troncature du tracé en bordure de fenêtre.

Le chapître IV concernant les généralités sur le basic graphique vous ont déjà introduits les notions d'aire de tracé et d'échelle.

VI - B - LA COMMANDE "VIEWPORT" :

VI - B - 1 - SYNTAXE :

1) avec le préfixe SET:

SET"VI[EWPORT]", P1x,P2x,P1y,P2y

2) avec le préfixe ASK :

ASK"VI[EWPORT]", V1,V2,V3,V4

où : P1 : point inférieur gauche de l'écran

P1x,P1y : coordonnées X et Y de P1

P2 : point supérieur droit de l'écran

P2x,P2y : coordonnées de X et Y de P2

(voir figure VI-1)

et V1,V2,V3,V4 sont des variables flottantes.

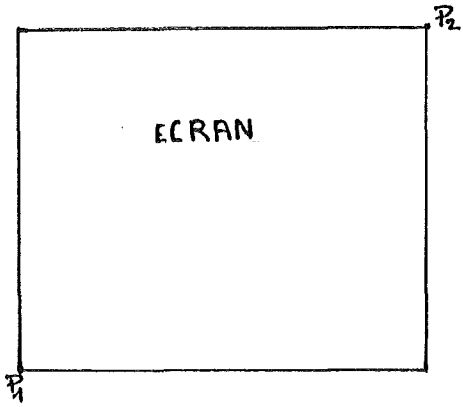


fig. VI-1

VI - B - 2 - DESCRIPTION :

Cette commande permet la modification physique de la taille d'un écran.
Une fois l'écran redéfini par la commande VIEWPORT, le faisceau ne sort pas de cette limite sauf cas exceptionnel (commande CLIP).

Quand on lance un Handler, le système donne des valeurs par défaut aux paramètres "VIEWPORT". Elles constituent la taille maximum que l'on peut obtenir physiquement sur l'écran.
On peut donc changer ces valeurs pour réduire ou augmenter cette surface de tracé.

ATTENTION : Les paramètres de VIEWPORT sont en coordonnées réelles même si une échelle a déjà été définie.

VI - B - 3 - EXEMPLES :

Soit le Handler no 1 correspondant à l'écran graphique de l'Apple II.

1) on lance le Handler : GR"HA",1,1

2) on arme le Handler : SET"DE",1,1

3) on demande la taille physique de l'écran :

ASK"VI",A,B,C,D

4) on regarde les variables : ?A,B,C,D

0.279 0 191

ce qui signifie que P1 a pour coordonnées :

P1 (0,0)

et P2 a pour coordonnées :

P2 (279,191)

La taille est donc : 0.....280 points sur l'axe X

et 0.....192 points sur l'axe Y

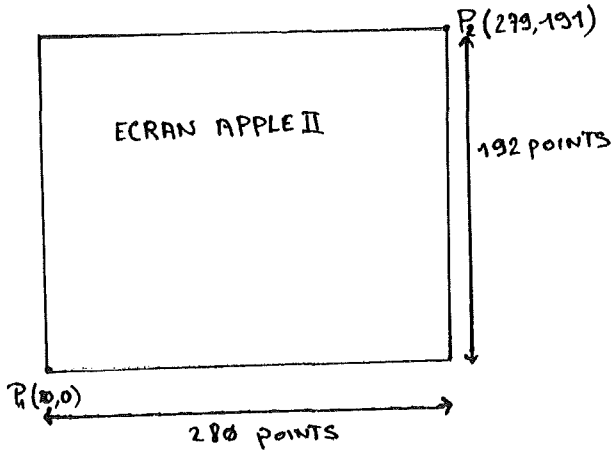


fig. VI-II

Nous allons maintenant modifier la taille physique de l'écran

5) réduisons le VIEWPORT au plus grand carré possible : soit 191 & 191

P1 aura donc comme coordonnées (0,0)

P2 aura comme coordonnées (191,191)

donc on exécutera :

SET"VI",0,191,0,191

la figure VI-3 donne le résultat physique après l'opération 5 :

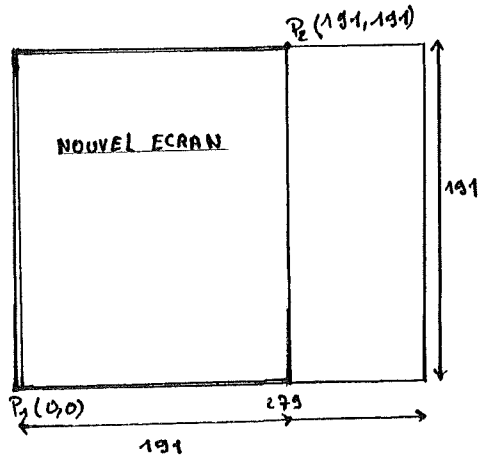


fig. VI-3

on peut centrer cet écran carré au centre de l'écran physique.

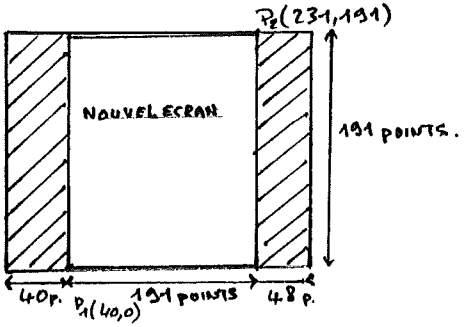
MEM/PLOT

6) déplaçons le VIEWPORT au centre de l'écran :

GR"VI",40,231,0,191

La figure VI-4 donne le résultat physique après l'opération 5 :

fig. VI-4



Il est possible de visualiser physiquement le cadre de l'écran en cours. Nous donnons ici la façon de le faire pour mieux comprendre l'effet de VIEWPORT, mais cette nouvelle commande fait l'objet du chapître VII.

7) GR"FRAME","VIEWPORT"

a pour effet d'encadrer avec le faisceau l'écran physique défini par VIEWPORT.

VI - B - 4 - INCIDENCE SUR LE SYSTEME :

Quand on exécute l'instruction : SET"VI",A,B,C,D, les paramètres du "VIEWPORT" sont mis à jour dans le buffer et les valeurs précédentes de ces paramètres sont perdues.
Pour récupérer les valeurs par défaut, il faut relancer le Handler correspondant.
Toute échelle ou manipulation d'objet graphique sera traité dans le nouveau "VIEWPORT".
Mais attention, si on redéfinit un VIEWPORT, il faut redéfinir l'échelle sinon elle est fausse car elle reste étalonné sur l'ancien VIEWPORT.

C'est la raison pour laquelle "VIEWPORT" n'est exécutable qu'une seule fois dans un programme.

En mode direct, comme dans un programme, pour définir une deuxième fois VIEWPORT, il faut soit relancer le Handler, soit exécuter l'instruction BASIC : CLEAR.

VI - B - 5 - TRAITEMENT DES ERREURS :

- La commande VIEWPORT demande obligatoirement 4 paramètres pour définir les points P1 et P2.
Si le nombre de paramètres dans l'instruction ASK"VI" ou le nombre de variables dans l'instruction SET"VI" n'est pas respecté, le système génère une erreur du type :

?WRONG NUMBER OF PARAMETER ERROR

- Si l'instruction SET"VI" est exécuté plus d'une fois sans avoir pris les précautions citées plus haut, le système génère une erreur du type :

?VIEWPORT ALREADY DEFINED ERROR

VI ~ C ~ LA COMMANDE WINDOW :

VI ~ C ~ 1 ~ SYNTAXE :

1) avec le préfixe SET :

SET"WI[NDOW]",W1x,W2x,W1y,W2y

2) avec le préfixe ASK :

ASK"WI[NDOW]",V1,V2,V3,V4

où W1x,W2x : échelle sur l'axe X

W1y,W2y : échelle sur l'axe Y

(voir figure VI~5)

et V1,V2,V3,V4 : variables flottantes.

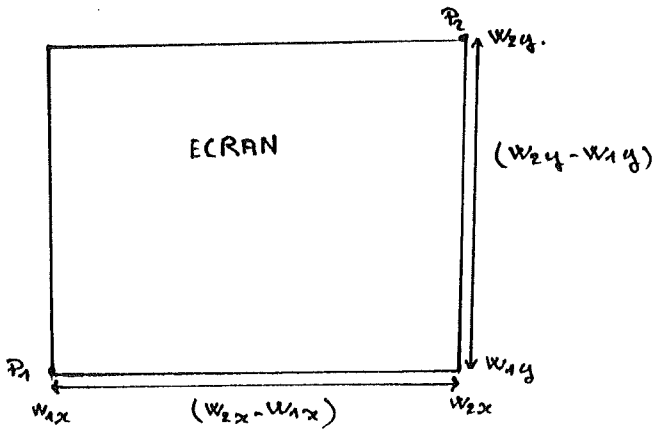


fig. VI~5

VI - C - 2 - DESCRIPTION :

La commande WINDOW permet de définir une échelle associée à l'écran en cours.
L'intérêt étant de pouvoir travailler dans un système de coordonnées correspondant le mieux possible au problème à traiter.
Le premier paramètre de WINDOW est associé au premier paramètre de VIEWPORT ainsi que les trois suivants de WINDOW sont associés aux trois suivants de VIEWPORT.

Ainsi change-t-on de système de coordonnées en définissant une échelle.
Quand on lance un Handler, les valeurs par défaut des paramètres de WINDOW sont égales aux valeurs du VIEWPORT.
Ceci permet à l'utilisateur de travailler immédiatement à l'échelle 1/1 que nous appelerons "unité graphique" car un point correspond à une unité.
Quand l'utilisateur redéfinit l'échelle avec la commande WINDOW, il passe en "unité utilisateur" ou "unité problème" car une unité ne correspond plus à un point mais à un groupe ou une fraction de points.

Notez bien que si l'on définit une échelle 10 & 10 sur un écran, l'image ne sera pas pour autant carrée car le point (10,10) correspondra au point P2 : la taille de l'écran n'est pas modifié (voir exemple plus bas).

VI - C - 3 - EXEMPLES :

Soit l'écran Apple II lancé et armé par les instructions :

```
GR"HA",1,1

SET"DE",1,3
```

Les quatre paramètres de VIEWPORT et de WINDOW sont donc :

```
0  279        0     191
```

Si on exécute : GR"PLOT",(0,0),(279,191)

on obtient une droite diagonale qui traverse l'écran de l'Apple II . (voir figure IV-6)
(la commande "PLOT" est décrite au chapître...)

Nous avons donc tracé à l'échelle 1/1.

Etablissons maintenant une échelle 10 & 10 par :

```
SET"WINDOW",0,10,0,10
```

et retraçons le trait cette fois-ci par :

```
GR"PLOT",(0,0),(10,10)
```

on obtient le même trait (fig. VI-7)

Etablissons maintenant une échelle 2/1 par :

```
SET"WINDOW",0,279 & 2, 0,191 & 2
```

et retraçons le trait jusqu'en (279,191),

on obtient un trait diagonale jusqu'à la moitié de l'écran: (voir fig. VI-8).

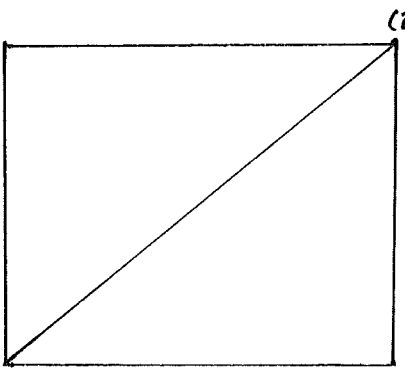


fig. VI-6

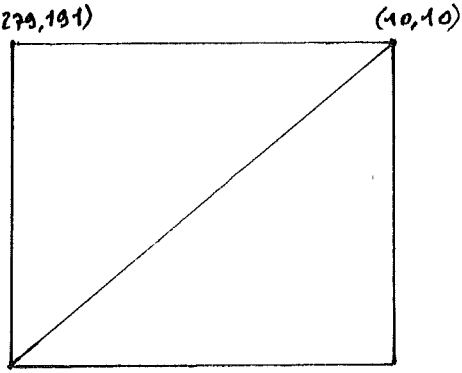


fig. VI-7

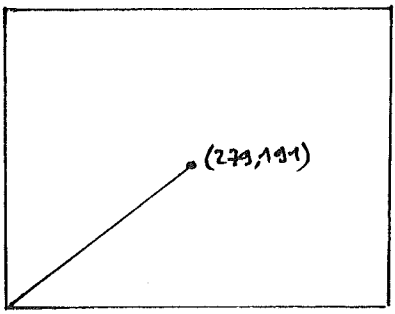


fig. VI-8

VI - C - 4 - INCIDENCE SUR LE SYSTEME :

Quand on exécute l'instruction : SET"W1",A,B,C,D.
Les paramètres "WINDOW" sont soit à jour dans le buffer et les valeurs précédentes de ces paramètres sont perdues.
Dès qu'une échelle est définie, tous les calculs de coordonnées qui suivront, seront faits dans cette nouvelle échelle. Pour travailler en unité graphique, il faut obtenir l'échelle avec les mêmes valeurs que le VIEWPORT.

Il est possible de changer d'échelle à tout moment.
Le système en tiendra compte et adaptera automatiquement ses paramètres de calcul.

VI - C - 5 - TRAITEMENT DES ERREURS :

La commande WINDOW demande obligatoirement quatre paramètres pour définir l'échelle sur les deux axes.
Si ce nombre de paramètres n'est pas respecté, alors le système génère l'erreur :

?WRONG NUMBER OF PARAMETER ERROR

VI - D - LA COMMANDE AREA :

VI - D - 1 - SYNTAXE :

1) avec le préfixe SET :

```
SET"AREA",P'1x,P'2x,P'1y,P'2y
```

2) avec le préfixe ASK :

```
ASK"AREA",V1,V2,V3,V4
```

où P'1x,P'1y : coordonnées du point P'1

et P'2x,P'2y : coordonnées du point P'2

P'1 et P'2 sont en coordonnées problèmes, il faut donc tenir compte quand on traite l'instruction SET"AREA"...

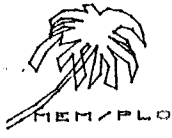
VI - D - 2 - DESCRIPTION :

La commande AREA ressemble, dans sa syntaxe, à la commande VIEWPORT.
Elle permet de définir une aire de tracé auxiliaire qui se comportera comme le VIEWPORT en cours si on en donne l'ordre au système par l'instruction SET"C-AREA", "ON" (voir VI-E).

Il est possible d'encadrer l'aire ainsi définie par l'instruction GR"FRAME","AREA" (voir.....)
On peut connaître les paramètres de "AREA" en utilisant le préfixe ASK avec la commande AREA.

Quand on lance un handler, la valeur des paramètres de la commande AREA est égale à celle du VIEWPORT. Ainsi à la mise en route, l'aire de tracé auxiliaire est égale à l'ECRAN en cours.

Le comportement de cette aire de tracé comme l'écran en cours sera traité au chapitre VI-E avec la commande "C-AREA".



VI - D - 4 - EXEMPLES :

Soit le handler de l'écran Apple II armé et lancé par les séquences :

```
GR"HA",1,1
SET"DE",1,3
```

1) interrogeons les paramètres de AREA :

```
ASK"AREA",A,B,C,D

?A,B,C,D

0,279,0,191
```

2) changeons ces paramètres :

```
SET"AREA",50,150,60,180
```

3) puis encadrons cette aire de tracé auxiliaire :

```
GR"FRAME","AREA"
```

4) encadrons l'aire de tracé VIEWPORT :

```
GR"FRAME","VIEWPORT"
```

on obtient la figure VI-9 :

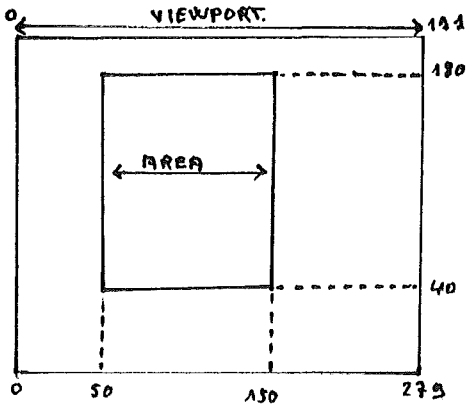


fig. VI-9

VI - D - 5 - INCIDENCE SUR LE SYSTEME :

Quand on crée une aire de tracé auxiliaire avec la commande AREA, les paramètres sont sauvegardés dans le buffer associé au handler en cours et les anciennes valeurs de ces paramètres sont écrasées.

On ne peut en effet définir qu'une seule aire de tracé auxiliaire à la fois, mais en jouant avec les préfixes "SET" et "ASK", on peut par soft en gérer d'une façon quasiment illimitéeen sauvegardant les paramètres dans des variables basic.

La création d'une AREA n'a pas d'incidence sur la taille de l'écran en cours tant que l'on a pas exécuté la commande "C-AREA" décrite en VI-E.

L'utilisation conjointe de "SET-AREA" et le GR"FRAME","AREA" permet de tracer des histogrammes dont chaque colonne est une aire de tracé auxiliaire temporaire qui l'on encadre.
Ceci fait l'objet du programme 01 de l'annexe....

VI - D - 6 - TRAITEMENT DES ERREURS :

La commande "AREA" demande obligatoirement quatre paramètres pour définir les points P'1 et P'2. Si cela n'est pas respecté, alors le système génère l'erreur :

?WRONG NUMBER OF PARAMETERS ERROR

VI - E - LA COMMANDE "C-AREA" :

VI - E - 1 - SYNTAXE :

1) avec le préfixe SET :

```
SET"C-AREA",["ON"
            "OFF","FILL"
            A$
```

2) avec le préfixe ASK :

```
ASK"C-AREA",V$
```

où "ON" et "OFF" sont les deux états possible avec SET

et V$, "ON" ou "OFF" en réponse de l'instruction AREA avec ASK.

V$ est obligatoirement une chaîne de caractères.

VI - E - 2 - DESCRIPTION :

La commande "C-AREA" s'utilise conjointement avec la commande AREA.
Quand une AREA a été créée, elle n'a aucune incidence sur le système tant que l'on a pas exécuté la commande "C- AREA".
"C-AREA" signifie "commande AREA".
Les paramètres de cette commande sont "ON" ou "OFF".
Quand on exécute SET"C-AREA", "ON", l'aire de tracé auxiliaire devient le VIEWPORT en cours.

Ceci signifie que l'échelle du VIEWPORT est reporté sur l'area et toute instruction à incidence directe sur le tracé aura effet sur l'aire auxiliaire et non sur le VIEWPORT ; jusqu'à ce que la commande SET"C AREA", "OFF" soit exécutée.

Tout tracé étant proportionnel au VIEWPORT en cours, si l'on retrace une même figure dans une aire de tracé auxiliaire qui est plus petite que le VIEWPORT, ce dernier sera proportionnellement plus petit.
Ceci est valable pour des tracés de droite, des axes et des caractères.
Quand on lance le Handler "C AREA" est à "OFF" par défaut.
Quand le paramètre de "C-AREA" est "FILL", MEM/PLOT remplit avec la couleur de fond l'aire définie même si celle-ci n'est pas "ON".

VI - E - 3 - EXEMPLES :

Soit le Handler de l'écran Apple II lancé et armé par la séquence :

```
GR"HA",1,1 et
SET"DE",1,3
```

1) Définissons une échelle :

```
SET"W1",0,10,0,10
```

et traçons une droite horizontale allant de

(0,5) à (5,5)

2) GR"PLOT",(0,5),(5,5)

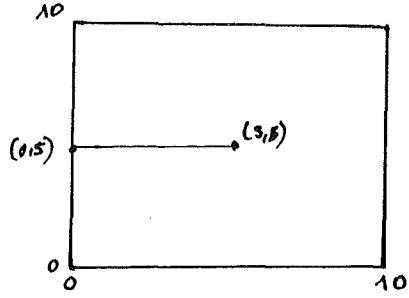


fig. VI-10

3) Définissons maintenant une AREA centrée sur l'écran :

```
SET"AREA",2,8,2,8
```

4) et encadrons cette aire :

```
GR"FRAME","AREA"
```

5) puis retraçons la même droite :

```
GR"PLOT",(0,5),(5,5)
```

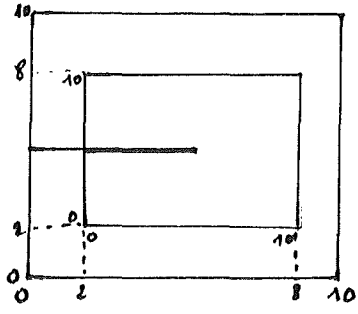


fig. VI-11

Remplissons maintenant cette aire par la commande :

GR"C A","FILL"

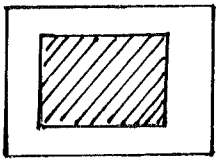


fig. VI-12

VI - E - 4 - INCIDENCE SUR LE SYSTEME :

Quand on exécute SET"AREA", "ON", le VIEWPORT en cours est remplacé par l'AREA précédemment définie. Tout calcul d'échelle et les tracés tiennent compte de cette nouvelle définition de VIEWPORT jusqu'à ce que l'on ait exécuté SET"C-AREA", "OFF".

VI - E - 5 - TRAITEMENT DES ERREURS :

Les paramètres autorisés sont "OFF" et "ON".
Ce sont obligatoirement des chaînes de caractères.
Si les paramètres sont autre chose que "ON" ou "OFF", le système génère "syntaxe error".

VI - F - LA COMMANDE "CLIP" :

VI - F - 1 - SYNTAXE :

1) avec le préfixe SET :

```
SET"CLIP"[,]"ON"
          "OFF"
          A$
```

2) avec le préfixe ASK :

```
ASK"CLIP"[,]"ON"
          "OFF"
          A$
```

où "ON" et "OFF" sont les deux états possibles avec SET

et V$ = "ON" ou "OFF" en réponse de l'instruction CLIP avec ASK.

V$ est obligatoirement une chaîne de caractères.

VI - F - 2 - DESCRIPTION :

Quand on redéfinit le VIEWPORT plus petit que la taille maximum, il est possible d'autoriser le tracé en dehors de l'écran.
Il est aussi possible de l'en empêcher.
Cela est aussi valable avec AREA quand celui-ci devient le VIEWPORT par la commande SET"C-AREA","ON".
Cette option s'appelle le "CLIPPING" d'où l'instruction CLIP qui peut prendre les deux états "ON" et "OFF".

SET"CLIP","ON" empêche le tracé en dehors.

SET"CLIP","OFF" permet le tracé en dehors.

A l'initialisation, "CLIP" est à l'état "OFF" par défaut.

VI - F - 3 - EXEMPLES :

Reprenons toujours notre écran graphique Apple II armé et lancé par la séquence :

```
GR"HA",1,1
SET"DE",1,3
```

1) Nous allons réduire le VIEWPORT :

```
SET"VI",0,150,0,191 : GR"FRAME,VI"
```

ce qui donne la figure VI-11.

2) la commande "CLIP" étant "ON" par défaut, traçons un trait qui va de (0,40) à (279,40)

```
GR"PLOT",(0,40),(279,40)
```

la figure VI-12 montre un trait qui va jusqu'au trait du cadre.

3) enlevons maintenant le clipping :

```
SET"CLIP","OFF"
```

4) et retraçons le même trait :

```
GR"PLOT",(0,40),(279,40)
```

nous voyons sur la figure VI-13 que le trait est sorti du VIEWPORT en cours.

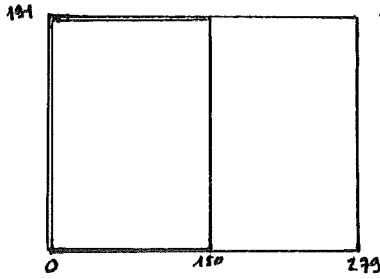


fig. VI-13-1

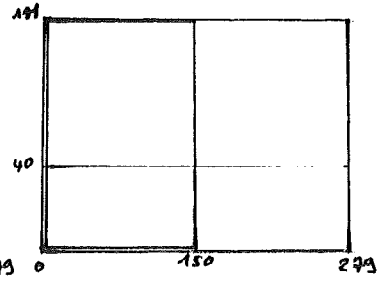


fig. VI-13-2

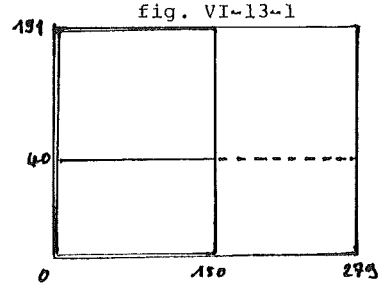


fig. VI-13-3

VI - F - 4 - INCIDENCE SUR LE SYSTEME :

La commande CLIP a une incidence sur le calcul des coordonnées pour tout type de tracé.
Si la taille de l'écran Apple II est maximal et que l'état de "CLIP" est "OFF", il se produit un "WRAP AROUND" quand on sort de l'écran par un tracé (plot) ; ce phénomène qui consiste en une réapparition du trait tracé de l'autre côté de l'écran paut justement être éviter en exécutant l'instruction SET"CLIP","ON".

Quand la taille de l'écran est plus petite que la taille maximale, le "WRAP AROUND" ne peut avoir lieu que si le trait dépasse la taille maximale de l'écran.

VI - F - 5 - TRAITEMENT DES ERREURS :

Les paramètres autorisés sont "OFF" et "ON" ; ce sont obligatoirement des chaînes de caractères.
Si les paramètres sont autre chose que "ON" ou "OFF", le système génère : "SYNTAX ERROR".

VI - G - LA COMMANDE "CLEAR" :

VI - G - 1 - SYNTAXE :

GR"CLEAR"[]
sans paramètre.

VI - G - 2 - DESCRIPTION :

"CLEAR" agit de différentes façons suivant le handler en cours :

- sur un écran graphique, la page en cours est vidée et le faisceau se positionne en (0,0).

- sur une table traçante sans dispositif rouleau :
la plume se positionne en (0,0).

- sur une table à rouleau :
le papier avance et la plume se positionne en (0,0).

VI - G - 3 - EXEMPLES :

GR"CLEAR" vide l'écran en cours.

VI - G - 4 - INCIDENCE SUR LE SYSTEME :

Le point courant devient toujours le point (0,0) en coordonnée problème.

VI - G - 5 - TRAITEMENT DES ERREURS :

La commande CLEAR n'accepte pas de paramètre.

CHAPITRE VII : LE TRACE

VII - A - GENERALITES :

Jusqu'à présent, nous avons étudiés la façon dont on programmait l'espace graphique.
Les instructions que nous avons eues l'occasion d'utiliser, n'avaient en général aucune influence directe sur le tracé puisqu'elles servaient à définir les paramètres de taille, d'échelle...

Nous allons voir dans ce chapître la manipulation du "faisceau". Ce terme qui peut paraître un peu "barbare" est employé dans un souci de standardisation puisque telle est la vocation principale de MEM/PLOT.
Sur un écran : "faisceau" est entièrement justifié ; c'est le faisceau lumineux qui se déplace en laissant un tracé derrière lui.
Sur une table traçante à plume, le faisceau, c'est la plume qui sait se déplacer ou tracer.
Sur une imprimante traçante, le faisceau, c'est la matrice de point de l'imprimante...
Nous allons aussi voir la notion de déplacement et de tracé avec les commandes "PLOT" et "POSITION".
Nous ne verrons cependant que le tracé de trait, le tracé de caractère faisant l'objet du chapître VIII.
Les commandes que nous allons étudier dans ce chapître sont :

BEAM : manipulation de la plume.

PLOT : déplacement ou tracé.

POSITION : déplace le faisceau sur l'écran.

COLOR :

BACKGROUND COLOR :

FRAME : encadrement du VIEWPORT ou de l'aire de tracé en cours.

En ce qui concerne les deux dernières commandes, des explications s'imposent.

Dans un souci de standardisation, "COLOR" signifie la couleur que l'on utilise.
C'est bien évidemment la plume d'une table traçante puisqu'une plume est associée à une couleur.

Ainsi une ambiguité peut-elle apparaître entre "COLOR" et "BEAM".
Il convient donc de dire que "COLOR" est employé pour "PLUME", et "BEAM" ou "FAISCEAUX" sont employés pour le résultat du tracé de la plume.

VII - B - LA COMMANDE "BEAM" :

VII - B - 1 - SYNTAXE :

1) avec le préfixe SET :

```
SET"BEAM"[,"ON"
         "OFF"
         A$
```

2) avec le préfixe ASK :

```
ASK"BEAM",V$
```

où "ON", "OFF" et A$ sont des chaînes de caractères.
V$ est une chaîne mise à jour par le système.

VII - B - 2 - DESCRIPTION :

La commande BEAM sert à modifier l'état du faisceau en cours; cette commande n'a pas une incidence directe sur le tracé, mais modifie cependant immédiatement l'état de la plume ou d'un faisceau lumineux.

"BEAM" peut prendre soit l'état "ON"
soit l'état "OFF"

quand l'état de BEAM est "ON"
le "faisceau est allumé"

quand l'état de BEAM est "OFF"
le "faisceau est éteint".

En utilisant le préfixe "ASK", on peut connaître à tout moment l'état du faisceau.
Par exemple, sur un écran graphique, on ne peut pas connaître l'état du faisceau d'une manière physique alors que sur une table traçante, on peut très bien voir si la "plume" est basse ou haute.
Avec ASK"BEAM",V$, on récupère dans V$ l'état de la plume sous forme d'une chaîne de caractères.
L'utilisation conjointe de "BEAM" et de "PLOT" (qui fait l'objet du paragraphe suivant) permet de tracer ou de déplacer le faisceau sans tracer.

Nous verrons par la suite que l'on peut exécuté un SET"BEAM" ou à l'aide d'un simple point-virgule dans les paramètres de la commande PLOT !

A la mise en route du Handler, "BEAM" est "OFF" par défaut.

VII - B - 3 - EXEMPLES :

SET"BEAM","ON" met le faisceau en position de tracé.
(plume basse sur une table)

```
ASK"BEAM",V$

?V$
ON          la plume est bien "ON"
```

VII - B - 4 - INCIDENCE SUR LE SYSTEME :

BEAM modifie l'état du faisceau de manière physique et d'une manière "SOFT".
En effet, MEM/PLOT doit être capable de se rappeler de l'état de la plume à tout moment.
Un flag lui est donc réservé dans le Buffer système associé au Handler en cours.

VII - B - 5 - TRAITEMENT DES ERREURS :

Si le paramètre de "BEAM" n'est ni "ON", ni "OFF", alors le système génère l'erreur : SYNTAXE ERROR.

VII - C - LA COMMANDE "PLOT" :

VII - C - 1 - SYNTAXE :

GR"PLOT"[], liste de coordonnées, (;)
où la liste de coordonnées est : (x1,y1), (x2,y2)...
le (;) est une option (voir description)

VII - C - 2 - DESCRIPTION :

"PLOT" est une commande à syntaxe multiple.
Nous n'en voyons cependant qu'un seul aspect dans ce chapître.
Sachez tout de même que la commande "PLOT", outre sa fonction de tracé et de déplacement, sert à appeler des "images" sous forme de sous-programme. Mais ce point étant si vaste à étudier qu'il fait l'objet d'un chapître entier dans la troisième partie : "programmation avancée".

La commande "PLOT" permet de tracer une liste de vecteurs dont les coordonnées problèmes sont précisées en paramètres.
A chaque déplacement ou tracé de vecteur, les coordonnées du point courant deviennent celles qui correspondent au point d'arrivée du tracé.

La différence entre le tracé et le déplacement réside dans l'état du faisceau.
Le faisceau étant manipulé soit avec la commande "BEAM" étudiée au paragraphe précédent, soit avec un petit artifice au niveau de la syntaxe de la commande "PLOT" que nous allons voir :

Cas 1) Si aucun paramètre ne suit la commande "PLOT" :

quelque soit l'état antécédent du faisceau, celui-ci passe à l'état : "OFF"

* si BEAM = "ON"

GR"PLOT" BEAM"OFF"

* si BEAM = "ON"

GR"PLOT" BEAM"OFF"

Cas 2) Si aucun paramètre ne suit la commande "PLOT" mais qu'il y ait juste un point-virgule.

quelque soit l'état antécédent du faisceau celui-ci passe à l'état "ON".

si BEAM = "ON"

GR"PLOT" BEAM"ON"

si BEAM = "OFF"

GR"PLOT" BEAM"ON"

Cas 3) Le système réagit de la même manière quand une liste de paramètres est présentée :

a) BEAM = "ON"

GR"PLOT",(x1,y1)...(xn,yn)

termine par BEAM "OFF"

GR"PLOT",(x1,y1)...(xn,yn);

termine par BEAM "ON"

b) BEAM"OFF"

GR"PLOT",(x1,y1)...(xn,yn)

se termine par BEAM"OFF"

GR"PLOT",(x1,y1)...(xn,yn);

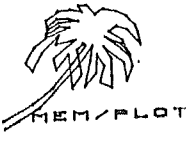
se termine par BEAM"ON"

quand une liste de paramètres est présente, le système trace les vecteurs correspondants à cette liste, en respectant la règle suivante :

- le premier couple de coordonnées est manipulé suivant l'état du faisceau en cours :
("ON" tracé / "OFF" déplacé)

- les couples suivants sont tracés
(BEAM"ON"/"BEAM")

pour clarifier ces points très importants, les exemples suivants traitent de tous les cas concrètement possibles.



VII - C - 3 - a) EXEMPLES :

Nous allons travailler sur l'écran graphique Apple II au Handler 1 :

Initialisation : GR"HA",1,1
SET"DE",1,3

Echelle : SET"W1",0,10,0,10

On obtient la configuration graphique suivante :

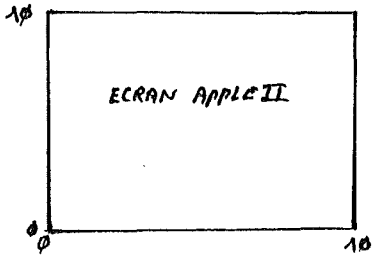


fig. VII-1

1) Déplaçons le faisceau, donc sans tracer ; au point P1 de coordonnées (5,5), le point courant étant par défaut en (0,0).
Assurons-nous d'abord que le faisceau est "OFF" par

SET"BEAM","OFF"

puis exécutons GR"PLOT",(5,5).

Nous n'avons pas mis de point-virgule donc le faisceau est resté sur OFF.

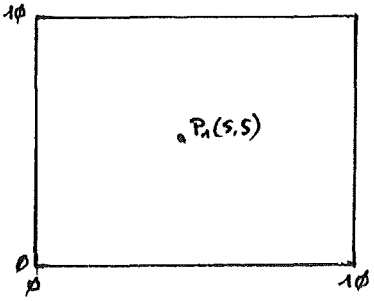


fig. VII-2

2) Nous voulons maintenant tracer un trait qui va de P1 à P2 (10,0).
Mais nous savons que le faisceau est "OFF".
Il faut donc le mettre sur "ON" :

```
SET"BEAM","ON"
```

puis GR"PLOT",(10,0) :

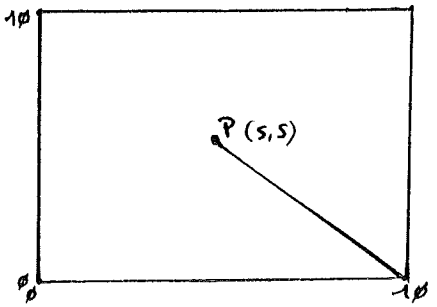


fig. VII-3

La séquence complète que nous avons exécuté pour obtenir le trait qui va de P1 à P2 est donc :

```
1 - SET"BEAM","OFF"

2 - GR"PLOT",(5,5)

3 - SET"BEAM","ON"

4 - GR"PLOT",(10,0)
```

On peut immédiatement voir que dans cette séquence d'instructions, le niveau 3 sert à mettre le faisceau à "ON" car le niveau 2 l'a laissé à "OFF".

Si nous mettons un ";" à la fin de l'instruction 2, l'instruction 3 ne sera plus utile puisque le faisceau sera mis à "ON" par ce point-virgule ce qui donne comme nouvelle séquence :

```
1 - SET"BEAM","OFF"

2 - GR"PLOT",(5,5);

3 - GR"PLOT",(10,0)
```

Appliquons maintenant le dernier point vu au VII ~ C ~ 2 : qui est :

Dans une liste de couple : le premier est traité avec l'état du faisceau en cours.
Les suivants sont traités avec le faisceau à "ON".
On peut donc réunir les instructions 2 et 3, à condition que le faisceau soit "OFF" au départ, soit :

```
1 ~ SET"BEAM","OFF"

2 ~ GR"PLOT",(5,5),(10,0)
```

qui donneront le même résultat que la figure VII~3.

On arrive donc à passer de la séquence :

```
1 ~ SET"BEAM","OFF"          1 ~ SET"BEAM","OFF"
                      à
2 ~ GR"PLOT",(5,5)           2 ~ GR"PLOT",(5,5),(10,0)

3 ~ SET"BEAM","ON"

4 ~ GR"PLOT",(10,0)
```

Voyons l'effet du même tracé sur l'instruction 1, qui met le faisceau à "ON" soit :

```
1 : SET"BEAM","ON"

2 : GR"PLOT",(5,5),(10,0)
```

Le faisceau étant à "ON" dès le début de l'instruction 2, on obtient un tracé de (0,0) à (5,5) puis de (5,5) à (10,0).

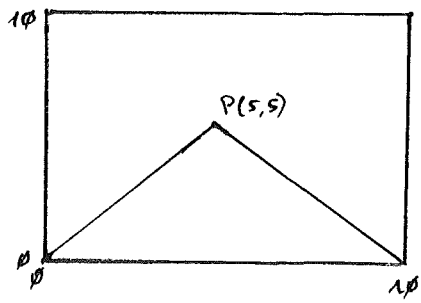


fig. VII~4

Vous voyez maintenant à quel point il est important de connaître les règles générales de manipulation du faisceau qui sont résumées à la page suivante.

Notez que l'instruction SET"BEAM","OFF" peut-être remplacé par : GR"PLOT"
et que l'instruction SET"BEAM","ON" peut-être remplacé par: GR"PLOT";

Notez que l'on peut exécuter une commande "PLOT" avec des coordonnées calculées telles que :

GR"PLOT",(sin(i)/4*sqr(i),i*cos(i*2)) par exemple.

VII - C - 3 - b) RESUME DES RÈGLES DE MANIPULATION DU FAISCEAU :

"Un point-virgule" à la fin d'une commande "PLOT" allume le faisceau (BEAM"ON").

"Une commande "PLOT" se terminant sans point-virgule éteint le faisceau (BEAM"OFF").

Une commande "PLOT" sans paramètre laisse le faisceau au même endroit mais allume le faisceau s'il y a un point-virgule et éteint le faisceau s'il n'y a pas de point-virgule.

Une commande "PLOT" avec des paramètres, fonctionnent de la manière suivante :

- le premier vecteur partant du point courant est tracé si le faisceau est "ON" et déplacé si le faisceau est "OFF" ;

- les vecteurs suivants sont obligatoirement tracés jusqu'au dernier.

VII - C - 4 - INCIDENCE SUR LE SYSTEME :

La commande "PLOT" quand elle est utilisée avec des paramètres, modifie la position du faisceau ainsi que son état s'il y a lieu.
A la mise en route du Handler, la position par défaut est (0,0).
Quand un déplacement, susceptible de modifier la position du faisceau, est exécuté, le système effectue ce déplacement en tenant compte de l'échelle en cours, du clipping, de la couleur et du type de ligne à tracer.
La commande "PLOT" ayant une incidence directe sur le tracé, s'exécute avec le préfixe "GR".

VII - C - 5 - TRAITEMENT DES ERREURS :

Les paramètres de la commande "PLOT" sont des couples de coordonnées du type (x,y).
La syntaxe générale de la liste étant un ensemble de couple séparé par une virgule, avec un ..... peuvent être un point-virgule ou rien du tout.
Le non-respect de cette syntaxe entraîne l'erreur :

?PICTURE NOT EXISTANT ERROR

La raison de ce message d'erreur étant la suivante : si une parenthèse n'existe pas comme premier caractère de la liste, MEM/PLOT prend ce paramètre comme un nom d'image (.....)

VII - D - LA COMMANDE "POSITION" :

VII - D - 1 - SYNTAXE :

1) Avec le préfixe SET :

```
SET"PO[S]ITION",(x,y)
```

2) Avec le préfixe ASK :

```
ASK"PO[S]ITION",x,y
```

où (x,y) est un couple de coordonnées,
et x,y sont deux variables flottantes.

VII - D - 2 - DESCRIPTION :

La commande "POSITION" s'approche beaucoup de "PLOT" dans une de ses formes :

```
SET"BEAM","OFF";GR"PLOT",(x,y)
```

Cette séquence peut tout simplement être remplacée par l'instruction :

```
SET"POS",(x,y)
```

qui déplace le faisceau sans s'occuper de l'état du faisceau en cours.
Avec le préfixe ASK, on peut connaître à tout moment la position du faisceau en coordonnées problèmes.
Il s'agit là, en fait, d'une digitalisation automatique qui fait l'objet du chapître...

VII - D - 3 - EXEMPLES :

Soit l'écran Apple II armé et lancé
(voir exemple VII-C-3).

Nous voulons nous déplacer en 5,5 :

```
SET"POS",5,5
```

Nous voulons connaître la position du point courant :

```
ASK"POS",Q,N
?Q,W
5,5  <--- réponse
```

On peut remplacer la dernière séquence du VII-C-3 par :

1) `SET"POS",5.5 : SET"BEAM","OFF"`

2) `GR"PLOT",(10,0)`

VII - D - 4 - INCIDENCE SUR LE SYSTEME :

La commande "POSITION" ne fait que modifier la position du faisceau.
Elle ne modifie pas l'image.
Quand on déplace le faisceau avec la commande "DOS", le tracé suivant partira du nouveau point obtenu.

Sur une table traçante, on verrait immédiatement la plume se déplacer jusqu'au point dont les coordonnées sont précisées dans la commande DOS.

Nous verrons par la suite que la commande "DOS" permet de positionner le faisceau pour écrire une chaîne de caractères.

VII - D - 5 - TRAITEMENT DES ERREURS :

Avec le préfixe SET, "DOS" doit être utilisé avec un paramètre couple de coordonnées, sinon le système génère :

?SYNTAX ERROR

VII - E - LA COMMANDE "COLOR" :

VII - E - 1 - SYNTAXE :

1) Avec le préfixe SET :

SET"COLOR",n

2) Avec le préfixe ASK :

ASK"COLOR",V

où n est un numéro de couleur possible.

V est une variable mise à jour par le système.

VII - E - 2 - DESCRIPTION :

La commande "COLOR" sert à charger la couleur du tracé sur un écran ou une table traçante.

Au niveau de la table traçante, SET"COLOR", provoque un changement de plume, alors que la même commande sur un écran graphique provoque le changement de couleur pour les prochains tracés.

Pour connaître le nombre de couleurs possibles, sur une table, on le voit au nombre de plumes sur un écran, il faut exécuter un utilitaire.
Voir la notice sur les utilitaires réunis sur la disquette.

La commande "COLOR" avec le préfixe ASK permet de connaître la couleur en cours.

VII - E - 3 - EXEMPLES :

Nous voulons travailler avec la couleur no 3 :

```
SET"COLOR",3
```

Nous voulons connaître la couleur en cours :

```
ASK"COLOR",C

?C
3.......réponse
```

VII - E - 4 - INCIDENCE SUR LE SYSTEME :

La commande "COLOR" charge physiquement la couleur et garde une image Soft de cette couleur pour l'interrogation.
Toute commande à incidence directe sur le tracé sera faite avec la dernière couleur définie sauf : le remplissage et le hachurage de figure qui se feront dans la couleur de fond (voir paragraphe suivant).
A la mise en route du système, la couleur par défaut est le 1.

VII - E - 5 - TRAITEMENT DES ERREURS :

Le système génère "ILLEGAL QUANTITY ERROR" si le numéro de la couleur sélectionnée n'existe pas.

VII - F - LA COMMANDE "BACKGROUND COLOR" :

VII - F - 1 - SYNTAXE :

1) Avec le préfixe "SET" :

SET"BACKGROUND COLOR",n

2) Avec le préfixe "ASK" :

ASK"BACKGROUND COLOR",V

où n est le numéro de la couleur.

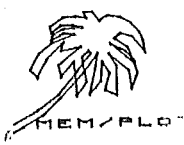
V est une variable numérique mise à jour par le système.

VII - F - 2 - DESCRIPTION :

La commande "BACKGROUND COLOR" ressemble beaucoup à la commande "COLOR".
Elle a le même effet à la différence que la couleur concernée, est utilisée pour le fond.
On en voit très vite l'utilité sur un écran graphique couleur où il est possible de gérer la couleur de fond.

En dehors de ce cas, la couleur de fond est toujours utilisée pour les remplissages (commande spéciale, voir chapître...)

Quand un périphérique graphique n'a qu'une seule couleur possible (ex: écran Apple), la couleur de fond est égale à la couleur de tracé et les commandes COLOR et BACKGROUND COLOR ne servent à rien.



VII - F - 3 - EXEMPLES :

Sélection de la couleur de fond no 2 :

```
SET"BA",2
```

Quelle est la couleur de fond actuelle ?

```
ASK"BA",Q
?Q
2 <---- réponse
```

VII - F - 4 - INCIDENCE SUR LE SYSTEME :

La commande "BACKGROUND COLOR" charge physiquement la couleur et garde une image Soft de cette couleur pour l'interrogation.

A la mise en route, la couleur de fond par défaut, dépend du Handler.

VII - F - 5 - TRAITEMENT DES ERREURS :

Le système génère "ILLEGAL QUANTITY ERROR" quand le numéro de la couleur sélectionnée n'existe pas.

Pour connaître le nombre .....de BACKGROUND COLOR, il faut exécuter un utilitaire (voir notion utilitaires).

En principe, le nombre de couleurs possibles est le même avec les commandes "COLOR" et "BACKGROUND COLOR".

VII - G - LA COMMANDE "FRAME" :

VII - G - 1 - SYNTAXE :

```
GR"FRAME","V"
          "A"
```

où V = VIEWPORT

et A = AREA

VII - G - 2 - DESCRIPTION :

La commande "FRAME" que nous avons déjà rencontrée au cours d'exemples des chapîtres précédents, permet d'encadrer le VIEWPORT ou l'AREA, avec la couleur en cours.

VII - G - 3 - EXEMPLES :

Encadrons le VIEWPORT de l'écran :

```
GR"HA",1,1 : SET"DE",1,3 : SET"DE",1,6
GR"FR","V"
```

Encadrons le VIEWPORT de l'aire 0,100,0,100

```
SET"AREA",0,100,0,100
GR"FRAME","A"
```

VII - G - 4 - INCIDENCE SUR LE SYSTEME :

Le point courant est modifié après la commande FRAME, il convient ..... de le sauvegarder avant par :

```
ASK"POS",A,B
```

et de le restaurer après par :

```
SET"POS",A,B
```

VII - G - 5 - TRAITEMENT DES ERREURS :

Si le paramètre de FRAME n'est ni V ni A, le système n'exécute rien du tout et ne renvoie pas d'erreur (extension future).

VII - H - UTILISATION CONJOINTE DES COMMANDES DE TRACE : PLOT, BEAM, POSITION, COLOR. (ou tracé d'un histogramme simple)

VII - H - 1 - GENERALITES :

Comme nous avons pu le voir dans les exemples de chapître, la commande PLOT peut s'utiliser dans plusieurs formes, tout en donnant le même résultat, ce qui peut sembler un peu déroutant pour le novice.

L'exemple qui suit, est un tracé d'histogramme simplifié qui montre une application concrète de ces commandes.
Simplifié car nous n'utiliserons ni le hachurage ni les labels puisque nous ne les avons pas encore vus.

VII - H - 2 - POSITION DU PROBLEME :

Nous allons tracer 12 colonnes qui représentent les 12 mois de l'année.
Chaque colonne étant l'image d'une valeur en francs.

Nous tracerons les axes et le cadre dans une couleur et l'histogramme dans une autre. Nous supposerons donc être sur un périphérique à deux couleurs, par exemple une table traçante en Handler 2.

VII - H - 3 - LE PROGRAMME :

```
10      GR"HA",2,5                  ; on arme le handler 2
15      SET"DE",2,5                 ; on lance ce handler
20      GR"FR","V"                  ; on encadre la feuille
30      SET"WI",0,12,0,10           ; on définit une
                                    échelle de 12 unités
                                    sur X (mois)
                                    et 10 u sur Y (milliers
                                    de francs)
40      SET"COLOR",2                ; on change de plume
50      SET"DOS",0,0                ; on se positionne en
                                    0,0
60      ......                      ; dans I position
                                    colonne
70      READ N                      ; dans N : hauteur
                                    colonne
80      GR"PLOT",(I,0),(I,N),(I+1,N),(I+1,0)
                                    ; on trace la colonne
90      NEXT N : GO TO 1000         ; on boucle pour le N
                                    suivant
100     DATA 4,6,1,0,2,8,9,10,10,4
                                    ; hauteur des colonnes
1000    SET"COLOR",1:END            ; on sépare en couleur1
```

Nous allons maintenant simplifier ce programme en ne touchant qu'aux instructions encadrées.

- la ligne 50 va tout simplement être supprimée car elle va devenir inutile à cause de la ligne 80.

- la ligne 80 est trop complexe. Nous allons utilisé la commande "AREA" pour définir la colonne et "FRAME","AREA" pour le tracé :

```
80 SET"AREA",I,I+1,0,N : GR"FR","A"
```

Le programme devient alors :

```
10      GR"FR",2,5                  : SET"DE",2,5

20      GR"FR","V"                  : SET"WI",0,12,0,10

30      SET"COLOR",2

40      FOR I = 1 to 12             : SET"AR",I,I+1,0,N

50      GR"FR","A"                  : NEXT N : GO TO 1000

60      DATA 4,6,1,0,2,8,9,10,10,4

1000    SET"COLOR",1                : END
```

Nous verrons par la suite que l'on peut rendre cet histogramme plus attrayant aux yeux et surtout plus compréhensible en lui rajoutant des axes, des labels, du remplissage.

CHAPITRE VIII : LE TRAITEMENT ALPHANUMERIQUE

VIII - A - GENERALITES :

Le système MEM/PLOT possède son propre générateur de caractères qui peut être utilisé d'une manière standard sur tous périphériques graphiques interfacés avec le système.

Ces caractères peuvent être tracés dans n'importe quelle direction, sens, hauteur et largeur. De plus, l'utilisateur peut, avec un utilitaire fournit avec la disquette de démarrage, créer ses propres caractères personnalisés; ceci se fait au dépend de la mémoire RAM (MEV) disponible.

Un jeu de caractères (préprogrammés) de MEM/PLOT est en ROM (MEM).

Chaque caractère est une suite de vecteur dont l'origine est l'arrivée du vecteur précédent les caractères en ROM possède entre 5 et 20 vecteurs, ce qui donne une précision assez grande pour avoir de beaux caractères dans n'importe quelle position.

Dans ce chapître, nous allons voir la manipulation de l'alphanumérique avec les commandes :

POSITION
TEXT ANGLE
CHARACTER SHAPE
PRINT

VIII - B - LA COMMANDE " POSITION " EN ALPHA :

VIII - B - 1 - LA COMMANDE "POSITION" EN ALPHA :

Nous connaissons déjà cette commande et sa syntaxe, voyons son utilisation sous le traitement alphanumérique.

VIII - B - 2 - DESCRIPTION :

" POS " permet de se positionner à un endroit précis de "L'ECRAN" pour "écrire" des caractères avec la commande PRINT. Quand on exécute:
"POS",x,y
(x,y) sera la position du premier caractere tracé.

Suivant le caractère de terminaison du PRINT, le curseur se repositionne au point (x,y) ou reste au niveau du dernier caractere tracé.(voir exemple).

Il faut signaler que le curseur est un objet fictif qui n'existe pas physiquement, il représente le point de départ d'un futur tracé.

La commande " POS " n'est pas une obligation pour se positionner à un endroit pour " écrire ". On peut très bien rester où l'on est ou se déplacer avec la commande "PLOT". Mais l'utilisation de " POS " est tout de même conseillée pour la clarté des programmes.

VIII - B - 3 - EXEMPLE :

On veut écrire à la position (10,10) :

```
SET "POS ", 10,10
```

On veut savoir où l'on va écrire si on ne se déplace pas:

```
ASK "POS ",x,y

? x,y
10,10
```

VIII - B - 4 - INCIDENCE SUR LE SYSTEME :

Voir VII - D - 4 (POS)

VIII - B - 5 - TRAITEMENT DES ERREURS :

Voir VII - D - 5 (POS)

VIII - C - LA COMMANDE "TEXT ANGLE" :

VIII - C - 1 - SYNTAXE :

1) avec le préfixe SET :

SET"TE[XT ANGLE]",A

2) avec le préfixe ASK :

ASK"TE[XT ANGLE]",V

où A est un angle en RADIAN

et V est la variable en retour en RADIAN.

VIII - C - 2 - DESCRIPTION :

La commande "TEXT ANGLE" permet de changer l'orientation de tracé des caractères. Le paramètre à donner est l'angle en Radian, compris entre 0 et 2... radians.
(si angle è 2....., MEM/PLOT prend le module 2....)
Quand on lance le handler, l'orientation est nulle : horizontale.

VIII - C - 3 - EXEMPLE :

- Supposons que l'on veuille écrire verticalement :

```
SET"TE",1.57
```

- Quelle est l'orientation du tracé ?

```
ASK"TE",V
?V
1.57
```

Voir l'utilisation conjointe de "TEXT ANGLE" et de "PRINT" au chapître VIII

VIII - C - 4 - INCIDENCE SUR LE SYSTEME :

Quand on exécute la commande SET"TEXT ANGLE", tout tracé de caractère par la commande "PRINT" sera affecté par l'angle de tracé A jusqu'à ce que l'on change l'angle ou que l'on fasse un reset du Handler auquel cas l'angle devient nul (position horizontale).

VIII - C - 5 - TRAITEMENT DES ERREURS :

Aucune erreur n'est générée avec cette commande. Le système effectuant l'angle module 2*Pi si ce dernier est supérieur à 2*Pi
Il faut mettre un seul paramètre.

VIII - D - LA COMMANDE "CHARACTER SHAPE" :

VIII - D - 1 - SYNTAXE :

1) avec le préfixe SET :

SET"CHA[RACTER SHAPE]",A,B,C

2) avec le préfixe ASK :

ASK"CHA[RACTER SHAPE]",V1,V2,V3

où A : largeur du caractère

B : longueur du caractère

C : espacement entre caractères

et V1,V2,V3 : variables réponse du système.

VIII - D - 2 - DESCRIPTION :

La commande "CHAR" permet de changer la forme des caractères tracés avec la commande PRINT.
Aussi est-il possible de travailler la forme en largeur et hauteur du caractère.

Le troisième paramètre définit l'espacement entre deux caractères dans une chaîne à écrire par "PRINT" (il est optionnel car l'espacement peut rester standard quelque soit la taille du caractère).

A l'initialisation du Handler, les paramètres par défaut sont les suivants:

largeur.5
hauteur.5
espacement 1

Les paramètres s'expriment en pourcentage du VIEWPORT en cours ; sachant que la matière de base est 7 & 7, on peut l'augmenter ou la diminuer en donnant un paramètre pour :

- la largeur : un pourcentage de P2x - P1x
- la hauteur : un pourcentage de P2y - P1y
- l'espacement : un pourcentage de P2x - P2y

Ces paramètres se comportant comme des facteurs .

Par exemple, si l'on exécute SET"CHA",1,1,1
Chaque vecteur constituant le caractère sera multiplié par :

1% de P2x - P1x pour X
1% de P2y - P1y pour Y

L'espacement de base : 1% de P2x - P1x
(espacement de base = 2u)

VIII - D - 3 - EXEMPLE :

Soit le Handler de l'écran Apple II armé et lancé.
Si l'on veut que la matrice de base soit multipliée par 10% de l'aire du VIEWPORT, il faut exécuter SET"CHA",10,10,1 (on garde l'espacement à 1%).

Quelle est la taille des caractères en cours ?

```
ASK"CHA",A,B,C
?A,B,C
10,10,1 <---- réponse
```

VIII - D - 4 - INCIDENCE SUR LE SYSTEME :

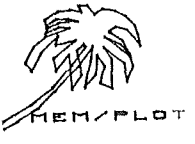
Quand on exécute cette commande, on modifie les paramètres de la forme de caractère pour ..... tracé postérieur, jusqu'à la redéfinition suivante ou le reset du Handler.

VIII - D - 5 - TRAITEMENT DES ERREURS :

Cette commande demande 3 paramètres. Le dernier paramètre est optionnel, l'espacement pouvant rester standard quelque soit la taille des caractères.

S'il n'y a pas au moins deux paramètres, le système génère :

?WRONG NUMBER OF PARAMETER ERROR?



VIII - E - LA COMMANDE "PRINT" :

VIII - E - 1 - SYNTAXE :

GR"PRINT","....."/A$

A$ étant une chaîne de caractères.

VIII - E - 2 - DESCRIPTION :

La commande "PRINT" permet d'écrire des caractères avec le générateur de MEM/PLOT ou des caractères spéciaux programmés par l'utilisateur avec le format et l'espacement programmés avec les commandes "CHAR" et "TEXT ANGLE".

Le tracé se fait toujours dans la couleur courante.
Pour invoquer les caractères utilisateurs, il faut les "encadrer" par le caractère " " (code ASC II 94).
Voir exemple.

Si un caractère ";" suit la chaîne A afficher, le curseur..... à la position finale de la chaîne, sinon, il revient où il était avec ce print.

VIII - E - 3 - EXEMPLES :

1) écrire "Bonjour"

```
GR"PRINT","Bonjour"
```

où A$ = "Bonjour"

```
GR"PRINT",A$
```

2) Si le caractère A a été redéfini comme étant caractère utilisateur, il faut l'invoquer par .

Soit écrire "Bonjour" puis le caractère spécial dont le code est A.

```
GR"PRINT","Bonjour^A^"
```

où A$ = "Bonjour"

```
GR"PRINT","Bonjour"+^"A"
```

3) Supposons que l'on ait défini un jeu complet de minuscules, chaque lettre majuscule a donc son équivalent en minuscule :

```
"A" = a
"B" = b
:   = :
```

pour écrire "Bonjour" en minuscules :

```
GR"PRINT","B^ONJOUR"
```

pour écrire Bonjour Monsieur Dupond :

```
GR"PRINT","B^ONJOUR M^ONSIEUR ^D^UPOND"
```

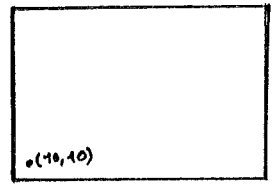
Le caractère . ^ . sert donc à alterner les deux jeux de caractères possibles :

- le jeu de MEM/PLOT
- le jeu de l'utilisateur.

4) Supposons que le point courant soit en (10,10).

```
SET"POS",10,10
```

fig. VIII-1

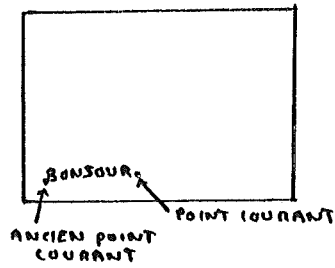


5) Nous allons écrire "Bonjour" avec les paramètres par défaut :

```
GR"PRINT","Bonjour";
```

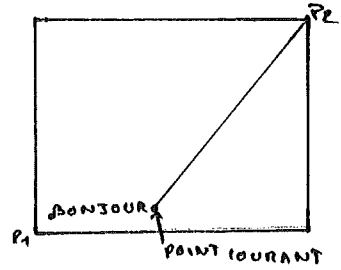
Ayant mis un point-virgule au bout de l'instruction, le point courant devient la fin du R.

fig. VIII-2



Si maintenant on veut tracer un trait jusqu'en P2 (279,191) :

fig. VIII-3



Recommençons la même séquence sans le point-virgule :

```
GR"CLEAR": SET"DOS",10,10
GR"PRINT","Bonjour"
SET"BEAM","ON"
GR"PLOT",(279,191)
```

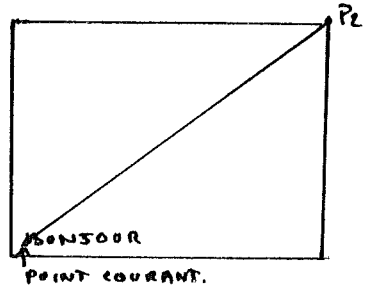


fig. VIII-4

VIII - E - 4 - INCIDENCE SUR LE SYSTEME :

La commande "PRINT" trace sur l'écran du périphérique en cours les caractères définis en paramètres, avec deux jeux possibles : le jeu ASC II complet, plus le jeu utilisateur programmable.

Le point courant à la fin de l'exécution du "PRINT" devient le dernier point tracé si l'instruction est suivit d'un point-virgule, ou redevient le point courant précédent avant l'exécution du "PRINT" si l'instruction n'est pas suivit d'un point-virgule.

VIII - E - 5 - TRAITEMENT DES ERREURS :

Une chaîne de caractères doit obligatoirement suivre la pommande "PRINT", sinon le système génère "?SYNTAX ERROR".

VIII - F - UTILISATION CONJOINTE DES COMMANDES : "CHAR", "TEXT ANGLE" ET "PRINT"

Les petits programmes qui vont suivre, montrent comment on peut utiliser conjointement les commandes que nous avons vues dans ce chapître.

VIII - F - 1 - PROGRAMME NO 1 :

Ce programme écrit les valeurs 0,pi/4,pi/2,pi...

dans leurs directions respectives, sur l'écran Apple II.

```
10 GR"HA",1,1 : SET"DE",1,3
12 SET"WINDOW",-10,10,-10,10
15 SET"POS",0,0
20 FOR I = 0 to 2*3.14 step 3.14/4
25 SET"TE",I : GR"PRINT",str$(i)
30 NEXT I
40 END
```

COMMENTAIRES :

Ligne 10 et 12, armement, lancement, échelle.

Ligne 15, positionnement en 0,0.

Ligne 20 - 30, boucle de 0 à 2*3.14 pas 3.14/4

Ligne 25 : on positionne l'angle I et on écrit sa valeur à l'écran dans l'orientation précédemment définit.

VIII - F - 2 - PROGRAMME NO 2 :

Ce programme écrit le jeu de caractères ASC II à l'écran de l'Apple II :

```
10 GR"HA",1,1: SET"DE",2,3:SET"DE",1,6

12 SET"WI",0,10,0,10

13 SET"POS",2,8

15 SET"CHA",.5,.5,1

20 FOR I = 32 TO 52

30 GR"PRINT",CHR$(I);

40 NEXT I

45 SET"PO",2,5

50 FOR I = 46 to 66

52 GR"PRINT",CHR$(I);

60 NEXT I

65 SET"POS",2,3

67 FOR I = 67 to 96

70 GR"PRINT",CHR$(I);

72 NEXT I

80 END
```

Commentaires :

On écrit par groupe de 20 caractères, l'équivalent ASC II des codes de 32 à 96, en changeant de DOS à chaque fois.

L'écriture des caractères en groupe de 20 est dûe au fait que le système ne gère pas les retours .....en graphique.

VIII - G - EXTENSION DU JEU DE CARACTERES GRAPHIQUES :

VIII - G - 1 - AU NIVEAU DE LA COMMANDE "PRINT" :

MEM/PLOT possède une indirection qui permet de gérer un second jeu de caractères propre à l'utilisateur.
Ce second jeu de caractères se déclanche avec un CHR$(94) ou un chapeau.

Exemple :

GR"PRINT","Bonjour [monsieur ["

Si aucun jeu utilisateur n'est présent, le caractère "[" est sauté et a un effet nul.

Si un jeu utilisateur est présent, le caractère "[" commute les jeux jusqu'au "[" suivants.

VIII - G - 2 - LA GESTION DU JEU DE CARACTERES UTILISATEURS :

La possibilité de créer un jeu de caractères auxiliaires n'est pas directement gérée par MEM/PLOT.

Deux solutions s'offrent à vous :

1) utiliser un utilitaire capable de gérer les jeux de caractères.

2) créer vous-même un jeu de caractères (voir le Bulletin Technique correspondant).

CHAPITRE IX : LES AXES

IX - A - GENERALITES :

Ce chapître concerne la génération automatique d'axes avec des bâtonnets dans l'écran du périphérique graphique en cours.

Ces commandes ne font pas parti du projet de nouvelle norme pour le basic graphique. Elles servent à simplifier la programmation des axes qui n'est pas évidente avec l'utilisation de la commande "PLOT" uniquement.

La génération automatique d'axe se fait avec les commandes suivantes :

TIC LENGTH : longueur des bâtonnets

XAXIS : tracé de l'abscisse

YAXIS : tracé de l'ordonnée

La combinaison de TIC LENGTH avec "XAXIS" et "YAXIS" permet de générer des axes avec des repères espacés de l'unité désirée.
Ces axes sont variables en longueur et commencent et finissent où l'on veut.

Cette génération automatique d'axe ne permet pas l'écriture automatique des labels sous les repères, ceci étant dû à l'immense variété d'écriture possible. Nous laissons donc ce soin à l'utilisateur. Des exemples lui montreront cependant la façon de la faire aisément.

IX - B - LA COMMANDE "TIC LENGTH" :

IX - B - 1 - SYNTAXE :

1) avec le préfixe SET :

SET"TIC[LENGTH]",A,B

2) avec le préfixe ASK :

ASK"TIC[LENGTH]",V1,V2

où A,B sont des variables numériques.

V1 et V2 sont les variables réponses du système.

IX - B - 2 - DESCRIPTION :

Cette commande permet de changer la longueur des bâtonnets qui représentent les "repères unités" sur chaque axe, que l'on trace avec XAXIS et YAXIS. C'est avec cette commande que l'on peut transformer les axes en grille qui couvre toute la page. Il suffit que ces bâtonnets montent et descendent jusqu'à la limite correspondante de la page.

Le paramètre A représente la taille de la partie supérieure du bâtonnet (au dessus de l'axe), alors que B représente la taille de la partie inférieure (sous l'axe).

Ces paramètres ne sont pas libres, ils ne peuvent en effet prendre que 7 valeurs : 1,10,20,30,40,50 ou 0.

- Si A ou/et B sont égales à 1, MEM/PLOT génère une grille (voir exemple).

- Si A ou/et B sont égales à 10,20,30,40 ou 50, le système génère des bâtonnets dont la longueur est respectivement égale à 10%, 20%, 30%, 40% ou 50% du côté du VIEWPORT correspondant.

A l'initialisation du Handler, A = B = 0 :
aucun tic ne peut-être tracé avec XAXIS et YAXIS.

Avec le préfixe ASK, l'utilisateur peut connaître à tout moment la taille des bâtonnets.

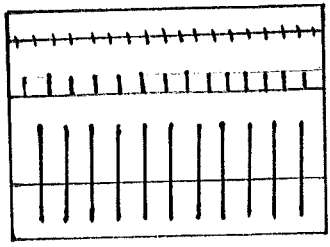
IX - B - 3 - EXEMPLES :

Ces exemples ne tracent pas les axes donc à fortiori pas les bâtonnets (voir VIII-C-3). Les AXES. sont là à titre indicatif pour montrer l'effet de la commande "TIC LENGTH" sur le tracé des bâtonnets.

Soit l'écran de l'Apple II armé et lancé.
Supposons que l'on ne travaille que sur l'axe des X.

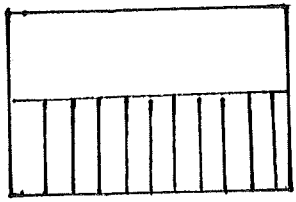
1) SET"TIC",10,10
donne des bâtonnets de 10% de 191 = 1.91

fig. IX-1



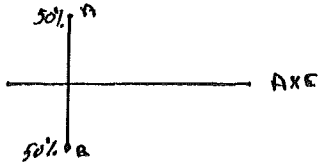
2) SET"TIC",20,0
donne des bâtonnets : 20% de 199 = 3,82 pour A et 0 pour B

fig. IX-2



3) SET"TIC",0,1
donne des bâtonnets 0% pour A et jusqu'à la limite de Viewport pour B.

4) SET"TIC",50,50
donne des bâtonnets de 50% pour A et B.



IX - B - 4 - INCIDENCE SUR LE SYSTEME :

Les valeurs données en paramètres dans la commande TIC sont sauvegardées dans le Buffer associé au Handler en cours.

Rappelons que les paramètres de "TIC LENGTH" n'agissent que sur les bâtonnets lors du tracé d'axe par les commandes "XAXIS" et "YAXIS".

IX - B - 5 - TRAITEMENT DES ERREURS :

La commande "TIC LENGTH" a besoin de deux paramètres. S'ils ne sont pas présents, le système génère l'erreur :

"?WRONG NUMBER OF PARAMETERS ERROR"

IX - C - LA COMMANDE XAXIS :

IX - C - 1 - SYNTAXE :

GR"XA[XIS]",y,A,B,T

où y : position sur Y
A et B : début et fin de l'axe : optionnel
T : espacement des Tics : optionnel

IX - C - 2 - DESCRIPTION :

Cette commande permet de tracer un axe des X en tenant compte des paramètres qui suivent et qui sont respectivement :

- la position en X en unité problème.
- le début de l'axe.
- la fin de l'axe.
- l'espacement des bâtonnets ou "TIC".

Les trois derniers paramètres étant optionnels.

- Si A, B et T sont omis,
une simple droite est tracé en guise d'axe allant d'un bout à l'autre du Viewport.

- Si B et T sont omis, une droite commençant en A est tracée jusqu'à l'autre bout de Viewport.

- Si T est omis, une simple droite est tracée de A à B.

- Si rien n'est omis :
une simple droite est d'abord tracé d'un bout à l'autre du Viewport.

Puis les Tics sont tracés avec un espacement T (en unité problème) et la longueur donnée avec la commande "TIC LENGTH" :

- si TIC LENGTH = 0,0 :
aucun TIC n'est tracé

- si TIC LENGTH = 1,1 :
une grille verticale est tracé.

- si TIC LENGTH = (10,20,30,40 ou 50) :
les TICS sont en % du Viewport.

Si l'on veut, par exemple, tracer un axe en bas du VIEWPORT, les tics inférieurs ne sont pas nécessaires.

IX - C - 3 - EXEMPLES :

Supposons que nous travaillons sur l'écran Apple II armé et lancé.

```
GR"HA",1,1 : SET"DE",1,3
SET"WI",-10,10,-10,10
```

1)
```
SET"TIC",0,0
GR"XAXIS",0
```

2)
```
SET"TIC",0,0
GR"XAXIS",0,-5
```

3)
```
T"TIC",0,0
GR"XAXIS",0,-5,7
```

4)
```
SET"TIC",0,0
GR"XAXIS",0,-5,7,1
```

5)
```
SET"TIC",10,10
GR"XAXIS",0,-5,7,1
```

6)
```
SET"TIC",1,1
GR"XAXIS",0,-10,10,1
```

7)
```
SET"TIC",20,0
GR"XAXIS",1,-10,0,0.5
```

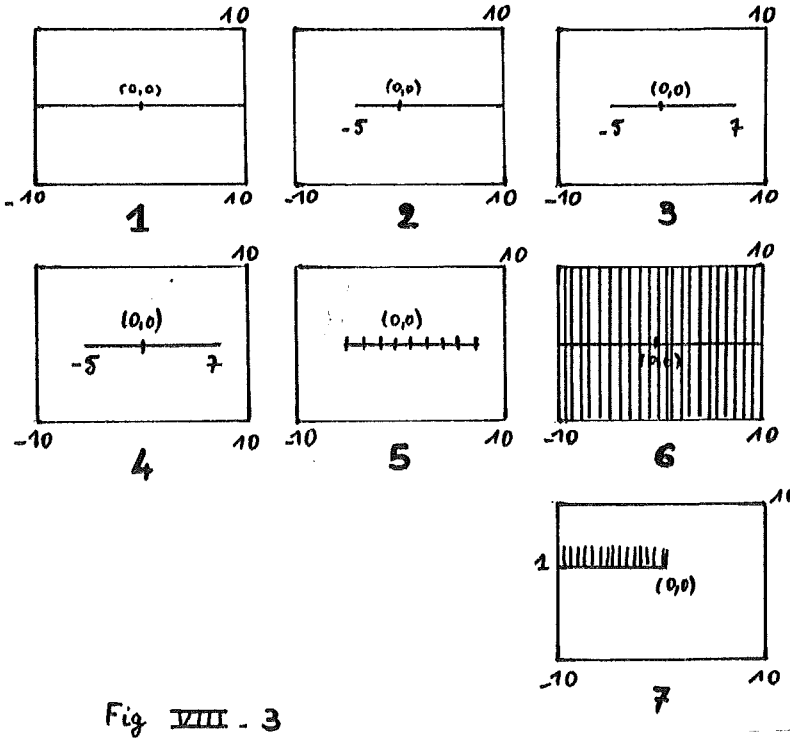


Fig VIII - 3

MEM/PLOT

IX - C - 4 - INCIDENCE SUR LE SYSTEME :

La commande "XAXIS" trace immédiatement sur l'écran du périphérique armé ; l'axe X.

Après l'exécution de la commande, le point courant n'est pas chargé.

IX - C - 5 - TRAITEMENT DES ERREURS :

Si un paramètre n'est pas au moins présent, le système génère :

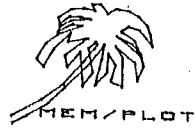
"?WRONG NUMBER OF PARAMETERS ERROR"

IX - D - LA COMMANDE "YAXIS" :

IX - D - 1 - SYNTAXE :

GR"YAXIS"[,] x,A,B,T

où x : position sur x
A et B : début et fin de l'axe (optionnel)
T : espacement des tics : optionnel.



IX - D - 2 - DESCRIPTION :

Cette commande permet de tracer un axe des y en tenant compte des paramètres qui suivent et qui sont respectivement :

- la position en x en unité problème.
- le début de l'axe.
- la fin de l'axe.
- l'espacement entre les tics.

Les trois derniers paramètres étant optionnels.

- Si A,B et T sont omis :
  une simple droite est tracé en guise d'axe, allant d'un bout à l'autre du Viewport.

- Si B et T sont omis :
  une droite commençant en A est tracé jusqu'à l'autre bout du Viewport.

- Si T est omis :
  une simple droite est tracé de A à B.

- Si rien est omis :
  une simple droite est d'abord tracé d'un bout à l'autre du Viewport. Puis les tics sont tracés avec un espacement T (en unité problème) et la longueur donnée avec la commande "TIC LENGTH".

  - Si TIC LENGTH = 0,0 :
    aucun TIC n'est tracé.

  - Si TIC LENGTH = 1,1 :
    une grille verticale est tracée.

  - Si TIC LENGTH = (10,20,30,40 ou 50) :
    les tics sont en % du Viewport.

Si l'on veut, par exemple, tracé un axe à gauche du Viewport, les tics de gauche ne sont pas nécessaires.

IX - D - 3 - EXEMPLES :

Supposons que nous travaillons sur l'écran Apple II armé et lancé :

```
GR"HA",1,1 : SET"DE",1,3

SET"WI",-10,10,-10,10
```

1)
```
SET"TIC",0,0
GR"YAXIS",0
```

2)
```
SET"TIC",0,0
GR"YAXIS",0,-5
```

3)
```
SET"TIC",0,0
GR"YAXIS",0,-5,7
```

4)
```
SET"TIC",0,0
GR"YAXIS",0,-5,7,1
```

5)
```
SET"TIC",10,10
GR"YAXIS",0,-5,7,1
```

6)
```
SET"TIC",1,1
GR"YAXIS",0,-10,10,1
```

7)
```
SET"TIC",20,0
GR"YAXIS",1,-10,0,0.5
```

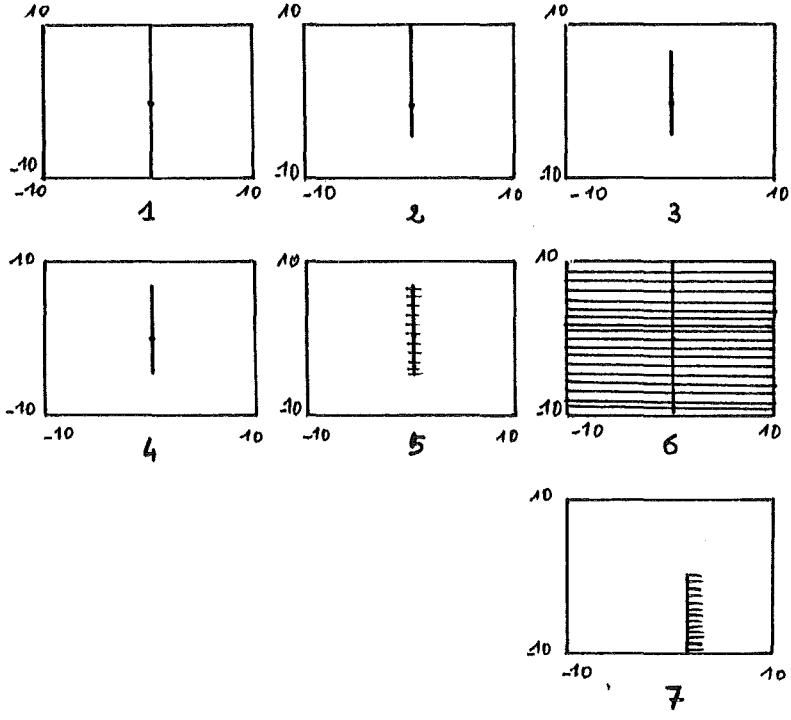


Fig VIII-4

IX - D - 4 - INCIDENCE SUR LE SYSTEME :

La commande "YAXIS" trace immédiatement sur l'écran du périphérique armé, l'axe y après exécution de la commande, le point courant n'est pas changé.

IX - D - 5 - TRAITEMENT DES ERREURS :

Si un paramètre n'est pas au moins présent, le système génère :

"? WRONG NUMBER OF PARAMETERS ERROR"

IX - E - MANIPULATION DE "XAXIS", "YAXIS" ET "PRINT" :

Le paragraphe donne un exemple concret de manipulation de "TIC LENGTH", "XAXIS", "YAXIS" et "PRINT" sous écran Apple II.

1) Traçons un axe des abscisses de -10 à +10 avec les labels -10,-9.....0,1,2.....8,9,10 sous chaque bâtonnet :

```
10 SET"HA",.5,-5,1

20 SET"WI",-10,10,-10,10

30 SET"T",10,10

40 GR"XAXIS",0,-10,10,1

50 FOR I:-10 to 10:SET"POS",I,0:GR"PRINT",?????

60 NEXT:END
```

Par l'axe des ordonnées, le programme est exactement le même sauf qu'il faut remplacer "XAXIS" par "YAXIS" et SET"POS",I,0 par SET"POS",0,1.

Il est donc possible sous MEM/PLOT de mettre des labels sous les repères unités des axes.

Et en jouant avec la commande "TEXT", on peut écrire verticalement ou horizontalement les labels que l'on veut.

Remarquez bien la présence des doubles cotes dans la commande :

```
GR"PRINT","" + STR$(I)
```

Celles-ci sont obligatoires car sinon, le paramètre STR$(I) est mal interprété.
Bien sûr, quand le paramètre n'est pas calculé, les cotes sont optionnelles.

PARTIE III :

PROGRAMMATION AVANCEE SOUS MEM/PLOT

CHAPITRE X : LES "PICTURES"

X - A - GENERALITES :

Les pictures sont des sous-images graphiques du même type que les sous-programmes BASIC, CALL FN du MEM/DOS.

Ce chapître décrit le fonctionnement des "pictures". L'utilisateur pourra se rapporter au Bulletin Technique, réf. 82T0047 du 21/09/82, décrivant l'utilisation des CALL FN.

La notion de sous-image graphique est très importante car elle permet la structuration des logiciels graphiques et les transformations d'image (scale, shift, rotate), cette dernière notion fait l'objet du chapître XI.

X - B - QU'EST-CE QU'UNE "PICTURE" ?

X - B - 1 - DESCRIPTION :

"Une picture" n'est autre qu'un sous-programme gonflé, admettant des paramètres sous forme de variables locales, globales et résultat, ce sont des sous-programmes invocables par un nom et permettant la récursivité.

Une picture est encadrée par une déclaration de début de picture et par une déclaration de fin de picture.

- le début de picture est constitué du mot PICTURE, plus le nom entre cotes puis les éventuels paramètres.

Exemple : PICTURE"TRIANGLE"(A,B)

- la fin de picture est constitué des mots : END PICTURE.

Pour écrire une sous-image graphique, il faut donc l'encadrer par PICTURE"nom" et END PICTURE.

X - B - 2 - L'INVOCATION :

Pour invoquer une "PICTURE", il suffit d'indiquer son nom en paramètre dans la commande PLOT.

Exemple : GR"PLOT","TRIANGLE"
ou
GR"PLOT",(0,4),(8,2),"CERCLE"

Ce dernier exemple montre la manière dont on peut tracer un symbole à la fin d'un vecteur (molécule de chimie par exemple).

Le nom d'une picture ne peut dépasser 15 caractères.

- on peut indiquer 253 niveaux de pictures.

X - B - 3 - GESTION DES "PICTURES" :

Le but des pictures étant la possibilité d'avoir en mémoire des objets graphiques que l'on a besoin de reproduire plusieurs fois, il est possible de savoir le nom de l'ensemble des pictures présents en mémoire en exécutant : LIST, PICTURE.

Le système liste alors le nom des pictures en mémoire :

Exemple : LIST PICTURE

```
PICTURE"TRIANGLE"
PICTURE"CERCLE"
```

Il est aussi possible de lister le contenu d'une picture par:

```
LIST PICTURE"nom"
```

Exemple : LIST PICTURE"TRIANGLE"

```
100 PICTURE"TRIANGLE"
110 PICTURE"TRIANGLE"
120 PICTURE"TRIANGLE"
150 END PICTURE
```

Lorsqu'au cours d'un déroulement de programme, MEM/PLOT rencontre PICTURE sans avoir eu d'appel préalable, le système génère "SYNTAX ERROR".

X ~ C ~ LA GESTION DES PARAMETRES :

X ~ C ~ 1 ~ LE PARAMETRAGE :

Les pictures permettent de passer des paramètres qui peuvent être :

~ en entrée : variable locale ou globale.
~ en sortie : variable résultat.

Une variable globale étant la même dans le programme principal et dans la "picture".
Une variable locale étant spécifique à la "picture".

Exemple :

```
~ programme principal : A = 1
~ picture : variable locale : A = 5
~ retour au programme principal
?A
1
```

~ Ces variables peuvent être de tout type sauf le type vecteur interne à MEM/PLOT.

~ Les variables en entrée et en sortie, peuvent être des tableaux (; après un nom : A$; représente le tableau A$).
Les variables locales ne peuvent être des tableaux.

~ L'appel d'une picture sans paramètre est équivalent à un ... avec nom.

~ Il n'est pas possible de définir un sous~programme avec des variables locales uniquement, il faut créer dans ce cas un paramètre fictif (entrée ou sortie).

X - C - 2 - ..... DES PARAMETRES :

PICTURE"nom" (S1,S2...=e1,e2.../l1,l2...)

où S1 : variable en sortie
e1 : variable en entrée
l1 : variable locale

un ; après les S1 ou les e1 indique que la variable est un tableau Basic.

X - D - LE TRAITEMENT DES ERREURS :

- plus de 253 niveaux imbriqués entraînent un "OUT OF MEMORY ERROR".

- trop de paramètres en appel entraînent le message : "SYNTAX ERROR".

- un "END PICTURE" entraîne "?OUT OF MEMORY ERROR".

- une invocation d'une "picture" inexistante entraîne "PICTURE NOT PRESENT ERROR".

- le passage du système sur le mot picture entraîne "SYNTAX ERROR".
(mettre END avant les "pictures")

X - E - LES EXEMPLES :

Bien qu'il soit possible de traiter des sous-programmes Basic classique avec les "PICTURES", les exemples suivants ne concernent que les applications graphiques qui sont notre préoccupation ici.

Soit l'écran Apple II armé et lancé.

```
GR"HA",1,1 : SET"DE",1,1
SET"WI",-10,10,-10,10
```

1) Définissons une "Picture" triangle.
   Pour cela, il faut tracer un triangle :
   Soit GR"PLOT",(0,0),(3,0),(1.5,1.5),(0,0)

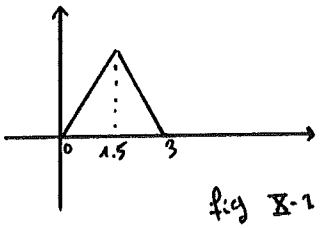


fig X-1

Mettons-le en Picture :

```
1000 Picture"TRIANGLE"
1002 GR"PLOT",(0,0),(3,0),(1.5,1.5),(0,0)
1005 END PICTURE
```

Passons en mode graphique et invoquons "TRIANGLE" :

```
SET"DE",1,3
GR"PLOT","TRIANGLE"
```

Vous avez sûrement remarqué que le sous-programme "Triangle" n'offre pas beaucoup d'intérêt car il reste en place fixe. Nous allons donc introduire des paramètres qui donneront le point relatif de départ du triangle.

La picture "Triangle" devient donc :

```
SET"DE",1,0

1000 PICTURE"TRIANGLE",(x,y)
1002 GR"PLOT",(x,y),(x+3,y),(x+1.5,y+1.5),(x,y)
1004 END PICTURE
```

Invoquons maintenant "Triangle" en donnant les coordonnées du point de départ en paramètre :

```
SET"DE",1,3
GR"PLOT","TRIANGLE",(0,0)
GR"PLOT","TRIANGLE",(-5,4)
```

(voir fig. X-1)

Il est donc possible de faire de la manipulation dynamique d'objet en deux dimensions sur une surface graphique.

Pour lister le nom de l'ensemble des pictures en mémoire : faire LIST PICTURE.
Pour lister la picture "TRIANGLE" : faire LIST PICTURE "TRIANGLE".

2) Définissons un cercle pour terminer les vecteurs pour tracer des molécules en chimie par exemple.

```
NEW

1000 PICTURE"CERCLE",(x,y)

1002 FOR I = 0 to 6.30 step .4

1004 GR"PLOT",(x+SIN(I)/10,y+COS(I)/10):NEXT I

1010 END PICTURE
```

Traçons un cercle après le vecteur qui va de (0,0) à (2,2).

```
SET"DE",1,3:GR"CLEAR"
GR"PLOT",(0,0),(2,2),"Cercle" (2,2)
```

On peut simplifier l'appel en introduisant dans le sous-programme picture "CERCLE" la demande de position courante pour tracer le cercle.

```
1000 PICTURE"CERCLE"

1002 ASK"POS",x,y

1004 FOR I = 0 to 2*3.14 step .4

1006 GR"PLOT",(x+.5*SIN(I)/10,y+COS(I)/10);

1008 NEXT I

1010 END PICTURE
```

L'invocation se fait maintenant par :

```
GR"PLOT",(0,0),(2,2),"CERCLE"
```

et on obtient la figure X-3 :

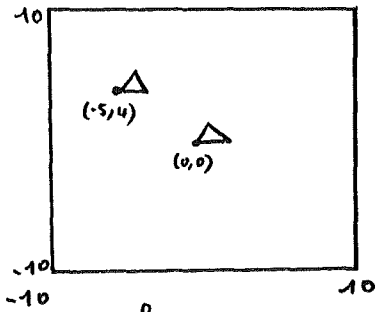


fig X-2

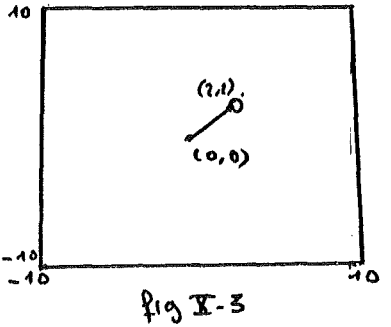


fig X-3

CHAPITRE XI : LES TRANSFORMATIONS

XI - A - GENERALITES :

Un des grands atouts de MEM/PLOT est la possibilité de réaliser des transformations de "picture", grâce à un module de traitement matricielle en deux dimensions pour le moment.

Les transformations se font en invoquant une sous-image graphique, elles sont au nombre de trois :

- SCALE :
- SHIFT : décalage sur chaque axe.
- ROTATE : rotation.

L'effet de transformation n'a lieu que pendant le tracé d'une picture et que sur les commandes de tracé (plot).

L'invocation d'une transformation se fait en même temps que l'appel picture avec le mot clé WITH.

Exemple :

```
GR"PLOT","TRIANGLE","WITH","SCALE",(2,5)
```

Il est possible de combiner jusqu'à trois transformations dans le même appel en les séparant par une * .

Exemple :

```
GR"PLOT","TRIANGLE","WITH","SCALE",(2,5)*"ROTATE",(1,5)*"SH
IFT",(-2,1)
```

Les transformations dans la même invocation doivent être groupées par type.
On ne peut en effet invoquer deux fois les transformations "SCALE",(2) et "SCALE",(3).
Il faut invoquer "SCALE",(6).

XI - B - LA TRANSFORMATION "SCALE" :

XI - B - 1 - SYNTAXE :

```
WITH  "SCALE",(a,b)
*
```

où a est le facteur scale sur X (et Y),
et b est le facteur scale sur Y.

XI - B - 2 - DESCRIPTION :

La transformation "SCALE" permet d'agrandir ou de diminuer par un facteur constant au x, sur y ou sur les deux axes un objet.
Selon l'échelle en cours, "SCALE" aura soit un effet d'agrandissement, soit un effet de zoom (voir exemple).

Si un seul paramètre est précisé, le facteur sera le même sur les deux axes.
Pour diminuer la taille d'une image, il faut donner des paramètres en 1/n ou n est le facteur de diminution.
(exemple : réduire 2.fois.=1/2..)

XI - B - 3 - EXEMPLE :

Soit la picture triangle que nous avons déjà utilisée au chapitre précédent :

```
1000 PICTURE"TRIANGLE"
1002 GR"PLOT",(0,0),(3,0),(1.5,3),(0,0)
1004 END PICTURE
```

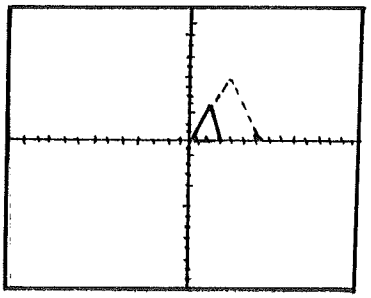
et soit le Handler Apple II initialisé :

```
GR"HA",1,1:SET"DE",1,3
SET"WI",-10,10,-10,10
```

Traçons des axes pour mieux voir l'effet de la transformation :

```
SET"TIC",10,10
GR"XA",0,-10,10,1
GR"YA",0,-10,10,1
```

et traçons la picture (fig. XI-1)



Traçons maintenant le triangle agrandi 2 fois :

```
GR"PLOT","TRIANGLE","WITH","SCALE",(2)
```

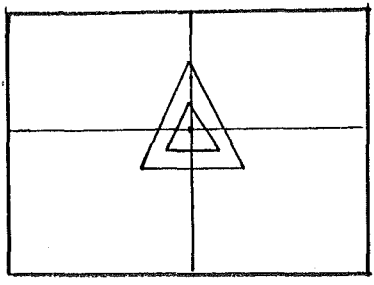
(fig. XI-1 EN POINTILLÉ)

Si par exemple :

la ligne 1002 de la picture avait été :

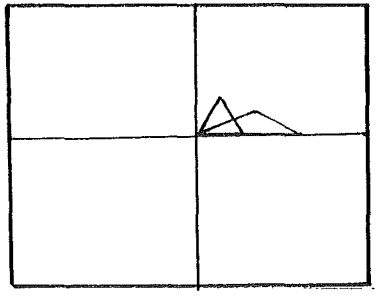
```
GR"PLOT",(-2,-1),(2,-1),(0,1),(-2,-1)
```

le triangle aurait été centré en (0,0) et l'agrandissement n'aurait laissé aucun point invariant : il y aurait donc eu effet de zoom.



Agrandissons le triangle de 2 fois sur X et de .5 fois sur Y :

```
GR"PLOT","TRIANGLE","WITH","SCALE",(2,.5)
```



XI - B - 4 - INCIDENCE SUR LE SYSTEME :

L'effet "SCALE" n'a lieu que pendant le tracé de la sous-image graphique.

XI - B - 5 - TRAITEMENT DES ERREURS :

La transformation "SCALE" n'accepte que deux paramètres maximum.
Les coordonnées du point courant sont modifiées.

XI - C - LA COMMANDE "SHIFT" :

XI - C - 1 - SYNTAXE :

```
WITH  "SH[F]T",(a,b)
*
```

où a est le décalage sur X (et Y),
et b est le décalage sur Y.

XI - C - 2 - DESCRIPTION :

La transformation "SHIFT" permet de décaler une sous-image graphique dans n'importe quelle direction en jouant avec les paramètres de décalage sur X et sur Y.

Si un seul paramètre est précisé, le décalage a lieu sur les axes X et Y avec le même facteur.

XI - C - 3 - EXEMPLE :

Restons fidèle à notre exemple du triangle :
traçons le triangle de référence par :

```
GR"PLOT","TRIANGLE"
```

décalons le triangle de 2 unités vers le haut et de 2 unités vers la gauche :

```
GR"PLOT","TRIANGLE","WITH","SHIFT",(2,-2)
```

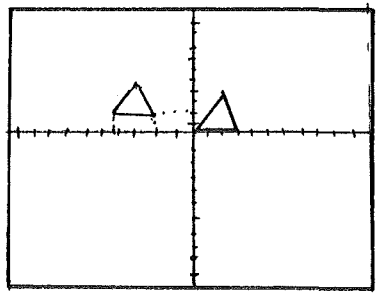


fig XI-4

XI - C - 4 - INCIDENCE SUR LE SYSTEME :

Le décalage n'est effectif que pendant le tracé de la sous-image graphique.
Les coordonnées du point courant sont modifiées.

XI - C - 5 - TRAITEMENT DES ERREURS :

La transformation "SHIFT" n'accepte que deux paramètres maximum.

XI - D - LA TRANSFORMATION "ROTATE" :

XI - D - 1 - SYNTAXE :

```
WITH "ROTATE",(a)
*
```

où a est l'angle de rotation en radian.

XI - D - 2 - DESCRIPTION :

La transformation "ROTATE" permet d'exécuter la rotation d'angle a d'une sous-image graphique.
Si l'image passe par le point (0,0), celui-ci reste invariant.
Si l'objet dans l'image a comme origine le point (0,0), la rotation se fera autour de ce point.

XI - D - 3 - EXEMPLE :

Reprenons notre exemple du triangle et traçons le comme base :

GR"PLOT","TRIANGLE".

Effectuons une rotation de à cette sous-image :

GR"PLOT","TRIANGLE","WITH","ROTATE",(1.57)

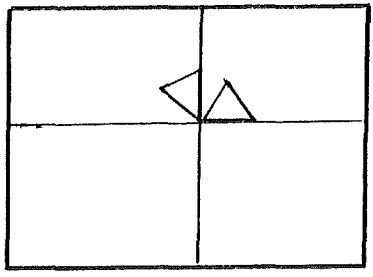
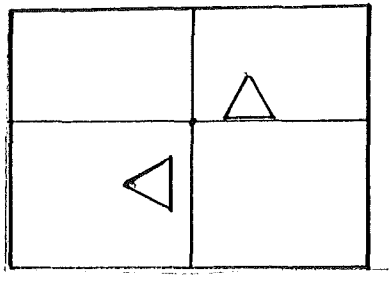


fig XI·5

Si la ligne 1002 de la picture "TRIANGLE" avait été :

1002 GR"PLOT",(1,1),(1,4),(2,4),(1,1)

la même rotation de ..... aurait donné :



Le point invariant (0,0) étant à l'extérieur de la figure, la rotation n'est plus relative à l'origine de la figure.

XI - D - 4 - INCIDENCE SUR LE SYSTEME :

La rotation n'est effectuée que pendant le tracé de la sous-image graphique.
Les coordonnées du point courant sont modifiées.

XI - D - 5 - TRAITEMENT DES ERREURS :

La transformation "ROTATE" n'accepte qu'un seul paramètre.

XI ~ E ~ LA COMPOSITION DES TRANSFORMATIONS :

Il est possible comme nous l'avons vu au XI~A de composer 2 ou 3 transformations.
Pour cela, il suffit de séparer la deuxième de la première et la troisième de la seconde par le caractère * qui signifiera dans cette commande : "composé avec".

Exemple :

```
Composons : SCALE(2)
            SHIFT(~1,1)
            ROTATE....
```

avec le picture triangle.

```
GR"PLOT","TRIANGLE","WITH","SCALE",(2)*"SHIFT",
(~1,1)*"ROTATE",(1.57)
```

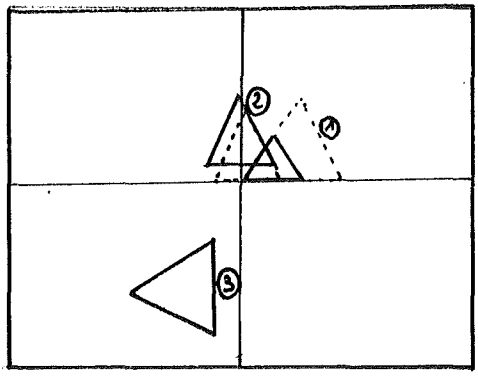


fig XI.7

CHAPITRE XII : LA DIGITALISATION

XII - A - GENERALITES :

La digitalisation permet de récupérer des informations (coordonnées et STATUS du faisceau) venant d'un périphérique.

Il existe deux catégories de périphériques qui savent digitaliser :

1) les tables traçantes qui ont l'option digitalisation.

2) les vrais périphériques de digitalisation :

- table à digitaliser,
- crayon optique.

Ce chapître s'adresse aux lecteurs qui désirent digitaliser sur un périphérique quelconque pourvu que le handler permette la digitalisation.

La variété des commandes et des possibilités exploitables des digitalisateurs du commerce étant si importante, MEM/PLOT possède une commande Status qui permet le déclanchement de certaines opérations propres au Handler.

Les exemples de ce chapître sont faits avec une table traçante HP 7470 qui peut renvoyer des informations grâce à l'interface SERIE BIDIRECTIONNELLE.

XII - B - LA GESTION DU STATUS DE DIGITALISATION :

XII - B - 1 - SYNTAXE :

1) avec le préfixe SET :

SET"ST[ATUS]","commande"

2) avec le préfixe ASK :

ASK"ST[ATUS]","commande"

XII - B - 2 - DESCRIPTION :

Un Handler de digitalisation est guidé par le STATUS qui est une chaîne de caractères.

Il convient de mettre à jour ce status pour digitaliser, par la séquence :

SET"STATUS","commande"

Pour connaître les commandes STATUS associées au Handler du périphérique que vous possédez, reportez-vous au Bulletin Technique correspondant.

XII - B - 3 - EXEMPLES :

Soit le Handler 2 armé et lancé avec comme périphérique associé la table traçante HP 7470 A.

Les commandes associées au STATUS sont :

* ⟶ digitalisation immédiate.
^M ⟶ digitalisation table.
*C ⟶ digitalisation par caractère clavier.

SET"ST","*"

La digitalisation se fera sans attendre d'intervention humaine (ce mode existe dans pratiquement tous les handlers).

SET"ST","^*"

Le renvoi des données se fera quand l'opérateur aura appuyé sur la touche d'envoi de la table.

SET"ST","*C"

Le renvoi des données se fera quand l'opérateur aura frappé une lettre au clavier du système.

Ces trois modes de digitalisation existent en principe dans tous les handlers pour raison de compatibilité.
Il peut cependant y en avoir d'autre, plus au moins complexe dépendant du périphérique digitaliseur.

XII - B - 4 - INCIDENCE SUR LE SYSTEME :

Quand on met à jour un Status de digitalisation, il est sauvegardé et utilisé à chaque appel de digitalisation.

Attention : il est commun à tous les handlers à la fois.

Un CLEAR, un NEW ou un NON annulent le status de digitalisation.

XII - B - 5 - TRAITEMENT DES ERREURS :

A la saisie du status, aucun contrôle n'est effectué.

XII - C - LA COMMANDE "INPUT" :

XII - C - 1 - SYNTAXE :

GR"INPUT",A,B,C,D$

où A,B sont les coordonnées x,y du point courant.

C est le status de la plume ou du faisceau.

D$ est la chaîne résultant d'une éventuelle touche frappée au clavier.

XII - C - 2 - DESCRIPTION :

La commande "INPUT" permet de digitaliser des données venant d'un périphérique graphique bidirectionnel dans le mode indiqué par le status de la commande SET"ST","mode". (La commande ST est expliquée au paragraphe précédent.).

La commande INPUT est surtout utilisée avec les périphériques digitaliseur (table à digitaliser, stylo optique...)

On se refferera au notice spécifique de chaque périphérique pour le mode de digitalisation.
Il est cependant possible de digitaliser avec des périphériques traçant ; exemple: table hp7470

XII - C - 3 - EXEMPLE :

Soit le Handler de la table traçante HP7470 armé et lancé:

Nous allons digitaliser à l'aide du bouton "enter" de la table :

```
set"ST","^*"

gr"input",x,y,t
```

Il suffit maintenant de se positionner à l'endroit désiré, puis d'appuyer sur la touche enter de la table pour débloquer le système.

On retrouve ensuite dans les variable x , y et t les coordonnées de plume son état: 1 = baissée/ 0 = levée.

XII - C - 4 - Incidence sur le système :

Quand on digitalise sous MEM/PLOT, si le status est donnée

tel maniere à ce que les données ne soit lachée qu'après une manipulation de l'opérateur, le système se met en attente jusqu'à ce que l'opération ait été effectuée !

XII - C - 5 - Traitement des erreurs :

Si au moment de la digitalisation , le status n'est pas conforme à ce que réclame le handler, le système génère l'erreur :

?digit status error .

NOTATIONS UTILISEES DANS CE MANUEL

1) crochet ouvert ... crochet fermé
Ce qui se trouve entre crochet n'a pas besoin d'être tapé pour être compris par Mem/Plot.

Exemple : SET"DEVICE,1,1,
non nécessaire
obligatoire

ASK"VIEWPORT",A,B,C,D

GR"HANDLER",1

2) (...) parenthèse ouverte .... parenthèse fermée
Ce qui se trouve entre parenthèses est optionnel ; des paramètres non obligatoires par exemple.

Exemple : SET"DEVICE",1,1

Cette instruction entièrement décrite donnerait :

SET"DEVICE",1,1

3) ~~~~~~~ soulignement en pointillé
Ce qui est souligné dépend du handler :

Exemple : GR"HANDLER",1,5,...
paramètre dépendant du handler

Ce qui donne au complet :

GR"HANDLER",1,5,...

SET"DEVICE",1,2,...

4) p1,p2,...pn liste des paramètres où :
p1 : premier paramètre, p2,...,pn : nème paramètre

V) LISTE DES COMMANDES MEM/PLOT 6502 :

COMMANDE	DESCRIPTION
GR"HANDLER",a,b	lance un handler
SET"DEVICE",a,b ASK"DEVICE",a,b	arme un périphérique pour utilisation
SET"VIEWPORT",a,b,c,d ASK"VIEWPORT",a,b,c,d	met à jour ou demande la taille du périphérique en cours
SET"WINDOW",a,b,c,d ASK"WINDOW",a,b,c,d	met à jour ou demande l'échelle en cours
SET"AREA",a,b,c,d ASK"AREA",a,b,c,d	détermine ou demande les limites de l'aire de tracé temporaire
SET"CLIP","on"/"off"/A$ ASK"CLIP","on"/"off"/A$	permet ou interdit l'interruption de tracé en bordure d'écran
GR"CLEAR"	vide la page écran
SET"BEAM","on"/"off"/A$ ASK"BEAM","on"/"off"/A$	lève ou baisse la plume en cours ou allume le faisceau
GR"PLOT",(x0,y0),...,(xn,yn)	trace en les reliant les points dont les coordonnées figurent en paramètres
GR"PLOT","nom"	trace une sous-image graphique
SET"position",a,b ASK"position",a,b	positionne ou demande la position du point
SET"COLOR",a ASK"COLOR",a	sélectionne ou demande la plume en cours

SET"BACKGROUND COLOR",a ASK"BACKGROUND COLOR",a	sélectionne ou demande la couleur de fond
GR"FRAME","v"/"a"	encadre le VIEWPORT ou l'AREA en cours
SET"TEXT ANGLE",a ASK"TEXT ANGLE",a	donne ou demande l'angle de tracé des caractères
SET"CHARACTER SHAPE",a,b,c ASK"CHARACTER SHAPE",a,b,c	détermine ou demande la taille (hauteur + largeur + écartement) des caractères
GR"PRINT",a$	écrit la chaîne a$, (caractère ASCII)
SET"TIC LENGTH",a,b ASK"TIC LENGTH",a,b	détermine ou demande la taille des bâtonnets sur les axes
GR"XAXIS",a,b,c,d	trace l'axe des X
GR"YAXIS",a,b,c,d	trace l'axe des Y
PICTURE"nom"(param) END PICTURE	délimitation d'une sous-image graphique
WITH/* "ROTATE"(a) "SHIFT",(a,b) "SCALE",(a,b)	utilisable avec list picture et GR "plot", "nom", transformation associée au picture, respectivement : rotation, décalage et zoom ou scaling
SET"ST",a$	met à jour ou demande le statut de digitalisation
GR"INPUT",a,b,c	ordre de digitalisation
GR"RELEASE"	donne des renseignements sur la version utilisée (sur l'écran Apple II)
SET"C-AREA",on	l'AREA devient le VIEWPORT en cours
SET"C-AREA","OFF"	commande inverse de la précédente
SET"C-AREA","FILL"	remplissage de l'AREA

ANNEXE C: Les erreurs

1 - Handler non existant error :

apparait quand la commande gr"HA" est éxécutée avec un numéro de handler non éxistant.

2 - bad dimension array error

apparait quand un mat plot est utilisé avec des indices erronés.

3 - plot without picture error

essais d'appel d'une picture inéxistante

4 - viewport already defined error

apparait lorsque l'on essaie de redéfinir le viewport une deuxieme fois.

5 - digit status error

Essai de digitalisation avec un status non mis à jour par la commande set"st"

6 - time out error

dépassement de temps dans une communication avec un périphérique (ieee...)

En cas d'erreur de syntax dans une commande MEM/PLOT, le système génère la meme erreur que dans BASIC soit : Syntax error.

En mode direct, les erreurs sont affichées en entier sur l'écran. En mode programme, si l'instruction On ERR Goto a été éxecuté au début du programme, Mem/Plot 6502 prend en compte le déroutement de l'erreur.