



# D.A.M.S

## MANUEL D'UTILISATION

Assembleur, Moniteur, Désassembleur Z80

# Index

<b>Présentation.....</b>	<b>3</b>
<b>Structure.....</b>	<b>5</b>
<b>chargement.....</b>	<b>6</b>
<b>Le moniteur.....</b>	<b>7</b>
<b>L'éditeur.....</b>	<b>12</b>
Les commandes générales.....	12
Les commandes de saisie du texte.....	13
Mode de travail de l'éditeur.....	14
Échanges de fichier texte.....	14
<b>L'ASSEMBLEUR.....</b>	<b>16</b>
Les options.....	16
La table des symboles.....	17
Pseudo-instructions d'assemblage.....	17
L'assemblage par blocs.....	19
Les expressions .....	19
Les erreurs au cours de l'assemblage.....	20
<b>LA TRACE.....</b>	<b>21</b>
La commande "T".....	21
La commande "R".....	22
La commande "J".....	22
<b>LES MESSAGES D'ERREUR.....</b>	<b>23</b>
<b>EXEMPLE D'UTILISATION.....</b>	<b>26</b>
<b>ADRESSES UTILES DE D.A.M.S.....</b>	<b>28</b>

## PRÉSENTATION

**D**.A.M.S est un logiciel permettant d'écrire et mettre au point des programmes en langage machine. Il contient entre autre un éditeur, un **assembleur**, un **désassembleur**, un **dump** et une **trace**. Sa simplicité d'utilisation, ses nombreuses commandes et sa rapidité d'exécution permettront de développer d'importants programmes en langage machine.

D.A.M.S est un logiciel intégrant un assembleur, un moniteur et un désassembleur symbolique pour développer et mettre au point facilement des programmes en langage machine sur les micro ordinateurs **AMSTRAD**. Les trois modules sont co-résidents en mémoire ce qui assure une grande souplesse d'utilisation. Vous pouvez notamment utiliser un éditeur plein écran, un assembleur immédiat, un désassembleur symbolique, une trace et beaucoup d'autres fonctions très puissantes. D.A.M.S est entièrement relogeable et est bien évidemment écrit en langage machine.

- L'éditeur est du type plein écran et sans numéro de ligne. Des commandes spéciales permettent la recherche ou la substitution de mots ou de phrases, l'effacement, la duplication et le déplacement de blocs de textes.
- L'assembleur Z80 est doté de pseudos instructions d'assemblage telles que **IF**, **THEN**, **ELSE**, et **DEFB**, **DEFW** etc... Il permet l'assemblage de plusieurs blocs de texte source. La table des labels générée peut être ensuite utilisée par le moniteur, le désassembleur et le mode trace.

- Le moniteur comprend plus de 15 commandes pour utiliser trace, dump, affichage et modification des registres, exécution d'un programme...Le désassembleur peut créer du texte source à partir d'un programme en langage machine avec génération automatique de labels et DEFB.. Le source peut être modifié à partir de l'éditeur et réassemblé.
- La trace comprend un mode rapide pour mettre au point des routines importantes. Chaque instruction est analysée avant son exécution pour éviter tout blocage système.

## STRUCTURE

**D** A.M.S occupe **10485** octets. Il contient sa propre pile machine et sa propre routine d'affichage, ce qui lui garantit son immunité lors d'exécution de routines par la trace. Il fait néanmoins directement appel à deux routines en **ROM** qui permettent de saisir les touches du clavier.

Les autres appels de routines extérieures se font lors d'exécutions de commandes utilisant un périphérique et toujours par les vecteurs du système d'exploitation de manière à pouvoir dévier ces routines. Par exemple, ceci permet d'utiliser un lecteur de cassette ou de disquette lors d'exécution des commandes **P** ou **G**.

A la suite de **D.A.M.S** se trouve le fichier texte et une éventuelle table de symboles. Entre ces 2 zones se trouve une plage de **256** octets qui est toujours dans la **RAM** centrale de **32K** et qui contient la pile machine et quelques routines. L'adresse de fin de cette zone est donnée par la valeur de **Hmem** : commande **X**.

## CHARGEMENT

**D** AMS est entièrement relogeable et peut-être chargé de l'adresse 1000 à 28000 de plusieurs façons.

### **RUN "DAMS"**

Ceci va charger un programme basic qui va charger D.A.M.S à l'adresse #4000. Si pendant le chargement du programme basic vous laissez la **touche espace enfoncée jusqu'à l'exécution**, alors l'amstrad va vous demander à quelle adresse voulez-vous charger D.A.M.S.

**! Cette méthode de chargement présente l'inconvénient d'effacer un éventuel programme basic déjà existant.**

Pour éviter cela, il suffit de charger directement les codes machine à l'adresse nn que vous choisissez :

```
MEMORY nn-1:LOAD "DAMS.BIN", nn:CALL nn
```

Dans tous les cas, pour réaccéder à D.A.M.S, il suffira de faire **CALL nn**, nn étant l'adresse où vous l'avez chargé.

De même, un programme exécuté par D.A.M.S grâce à la commande , peut se terminer non pas par un RET mais par un JP nn.

Ce point d'entrée sauve le contenu des registres du processeur.

Un autre point d'entrée qui peut être utilisé est l'adresse du moniteur équivalente à nn+2354. Ce point d'entrée ne mémorise cependant pas le contenu des registres.

## LE MONITEUR

**D**ès que l'on a accédé à D.A.M.S, le moniteur prend le contrôle et attend une des commandes. Celles-ci peuvent être tapées en majuscule ou en minuscule.

**<CAPS LOCK>** Passe de majuscule à minuscule et inversement.

D.A.M.S accepte indifféremment les majuscules et les minuscules. L'assembleur accepte aussi les minuscules.

**<CTRL> R** Fait basculer l'indicateur de haut de page sur **LRAM** ou **ROM**.

- **LRAM** signifie que de **0** à **#3FFF**, il y a de la RAM
- **ROM** signifie que de **0** à **#3FFF**, il y a de la ROM

**!** Vérifiez bien que l'indicateur est bien positionné avant toute commande telle que **D, H, K, T, R, J\$** etc...

**<CTRL> E** Fait basculer l'indicateur de haut de page sur **URAM** ou **EXT:n**

- **URAM** signifie que de **#C000** à **#FFFF**, il y a de la RAM
- **EXT:n** signifie que la même zone est occupée par la ROM numéro n. [Voir commande **E**]

**!** Comme pour la commande **<CTRL> R**, vérifiez bien cet indicateur.

**<CTRL> B** Bascule la base d'affichage. Tous les nombres affichés par D.A.M.S seront dans la base [10 ou 16] sélectionnée par

**<CTRL> B**.

**Ls** Permet d'accéder à l'éditeur au chapitre du texte spécifié par la chaîne s. s doit être un label du texte. Dès que les caractères de s correspondent aux premiers caractères du label, le texte est affiché. Si la chaîne s est un ";", le texte est affiché à partir du début. Si la chaîne s est **null**e le texte est

affiché à partir de la dernière position du curseur ">". Si l'imprimante a été sélectionnée [] alors le texte est transmis à l'imprimante. Si la chaîne **s** n'est pas trouvée dans le texte alors la commande n'a pas d'effet.

Élimine le fichier texte et la table des symboles. Pour ressusciter un fichier texte effacé par un **new** malencontreux, **poker 59** dans l'adresse de "Text" [donnée par la commande ].

Affiche une table de symboles [créée par un assemblage]. Elle peut être imprimée si auparavant l'imprimante a été sélectionnée.

**X0** ou **X** donne différentes adresses occupées par **D.A.M.S** :

- **Text** : c'est l'adresse du 1er octet du fichier texte.
- **End** : c'est l'adresse du dernier octet du fichier texte.
- **Hmem** : c'est l'adresse du dernier octet occupé par **D.A.M.S** [Voir commande .

Expression Numérique. Évalue l'expression et affiche le résultat. Tous les types de nombres et d'opérateurs décrits dans ce manuel peuvent être utilisés. Si une table de symboles existe, des labels peuvent figurer dans l'expression numérique.

Retour au Basic.

Poke la valeur **n** à l'adresse **nn**. Si **nn** est occupée par **D.A.M.S**, l'erreur **12** est lancée. **n** et **nn** peuvent être des expressions numériques.

Même effet que  mais sur **2** octets.



**Mnn** Fixe la dernière adresse mémoire occupée par D.A.M.S [Hmem]. Toute la zone mémoire comprise entre l'adresse de début de DAMS et Hmem est exclusivement réservée à D.A.M.S. Toute commande telle que **Q**, **W**, **A**... qui tente de modifier un octet de mémoire occupé par D.A.M.S, est clôturée par l'erreur **12**. La valeur de Hmem est visualisée par la commande **X**. Plus la valeur Hmem sera basse et moindre sera l'espace réservé au fichier texte et inversement. Toutefois, la valeur de Hmem devra se situer dans la Ram centrale de 32 K. Vous pouvez essayer différentes valeurs pour nn. Si cette valeur est trop basse l'erreur **12** est lancée, si elle est trop haute c'est l'erreur **8**. Cette commande détruit la table des symboles.

**Pn, s** Sauvegarde le fichier ayant pour nom la chaîne s.

- Si n=0 : Sauve le fichier texte.
- Si n=1 : Sauve le fichier texte, à partir de la dernière position du curseur ">".
- Si n=2 : Après un assemblage, sauve le code objet.

**Gn, s** Charge le fichier texte ayant comme nom la chaîne s. Le fichier texte est toujours chargé à la suite de celui déjà existant [s'il y en a un].

**! La valeur n n'a pas d'effet.**

**Sn, s** D.A.M.S possède en mémoire 2 chaînes de 16 caractères utilisées par l'éditeur [Voir commande **<COPY>** de l'éditeur]. n spécifie la chaîne concernée [1 ou 2] et s est la nouvelle chaîne. Si n est égal à 0, le contenu des chaînes est tout simplement affiché. La chaîne 1 est la chaîne de recherche, et la chaîne 2 est la chaîne de substitution. Lors de la saisie, ces 2 chaînes sont analysées comme des phrases du fichier texte, car c'est avec lui qu'elles sont en relation. Donc, un mnémonique doit être précédé d'un espace, et une chaîne qui n'est ni un mnémonique ni un label doit être précédée de 2 espaces. Après chaque commande

, le contenu des 2 chaînes est affiché de manière à voir s'il a été accepté [et si les chaînes peuvent donc se trouver dans le fichier texte].

**An,s** Assemble le fichier texte avec l'option **n** et **s** comme nom du 1er bloc. Ces 2 arguments ne sont pas obligatoires. [Voir assembleur]

**Fn,s** Assemble le fichier texte qui se trouve sur support magnétique avec l'option **n** et dont le 1er bloc a pour nom la chaîne **s**.

**Dnn** Désassemble pas à pas à partir de l'adresse **nn**. Si l'imprimante est sélectionnée, D.A.M.S demande l'adresse de fin. La touche "Q" ou **<CTRL> C** arrête le désassemblage. Toute autre touche continue le désassemblage instruction par instruction.

**Knn** Liste la mémoire à partir de l'adresse **nn**. Le bit 7 des caractères affichés a été remis à 0. "Q" ou **<CTRL> C** clôture le listage.

**Tnn** Dans un premier temps, affiche le contenu des registres, puis trace pas à pas [voir trace]. "Q" ou **<CTRL> C** revient au moniteur.

**Rnn** Trace en mode rapide à partir de l'adresse **nn** [voir trace]. **<CTRL> C** ramène au mode Trace.

**Jnn** Exécute le programme à l'adresse **nn**. Après un assemblage, si un ENT se trouvait dans le texte, la commande **J** sans argument exécute le programme à l'adresse déclarée dans le ENT. Si **nn** existe le programme est lancé en RAM. Indifféremment des indicateurs, le registre BC' prend la valeur #7F8E et F' est remis

à 0. Il est possible de transmettre une autre valeur dans les registres BC' et F' à un programme en RAM en faisant J\$-\$+nn. [voir la commande J\$ dans le paragraphe Trace].

Permet de modifier le contenu des registres du processeur. Peut être utile avant de lancer un programme [avec

ex : .BC,2 Charge BC avec la valeur 2

ex : ..BC,2 Charge B avec la valeur 2

ex : ...BC,2 Charge C avec la valeur 2

ex : .BC',#7F8E Charge BC' avec la valeur #7F8E

Cette commande désassemble la mémoire à partir de l'adresse nn et inclue au fichier texte à partir de la dernière position du curseur ">", le texte du désassemblage. Le désassemblage se fait en 2 passes de manière à créer une table de symboles. La mémoire écran est obligatoirement utilisée. Après Hnn, D.A.M.S demande : Text ?



Tous les octets qui suivront l'adresse que vous donnerez seront inclus au fichier texte sous forme de DEFB.


Ensuite D.A.M.S demande : Code ?

S'il y a de nouveau du langage machine après les DEFB alors rentrez l'adresse et ainsi de suite. Si vous ne répondez pas à ces questions en appuyant sur  alors D.A.M.S vous demande l'adresse de fin.


Lorsque l'indicateur URAM est sur EXT, la ROM concernée est fixée par la commande E. n doit être compris entre 0 et 251. 0 fixe la ROM basic et 7 fixe la ROM du lecteur de disquette.


## L'ÉDITEUR


On y accède par la commande  du moniteur. Il permet de créer, modifier, ajouter, effacer et dupliquer du texte qui sera soumis à l'assembleur. Dans cet éditeur 2 modes sont à considérer : le premier, celui auquel on accède grâce à la commande  permet les commandes générales. Le second permet la saisie du texte.


**!** Une fois la saisie du programme source terminée et quitter l'éditeur afin de retourner au programme principale, il suffi de valider par  sur une ligne vide

### Les commandes générales

 **[flèche gauche]** : Accède au 2ème mode d'édition. Il permet la saisie du texte à partir de la ligne pointée par le curseur ">". Après chaque retour de chariot, une nouvelle ligne est attendue. Un retour de chariot dans une ligne vide permet de sortir de ce mode.

 **[flèche droite]** : Permet de corriger la ligne pointée par le curseur ">". Un retour de chariot termine la correction. Si à ce moment là la ligne est vide, alors aucune modification n'est portée sur la ligne et l'ancienne ligne est affichée.

 **[flèche haut]** : Fait monter le curseur ">" d'une ligne ou descend le texte si le curseur ">" est en haut.

 **[flèche bas]** : Descend le curseur ">" d'une ligne ou monte le texte si le curseur est déjà en bas.



Efface la ligne pointée par le curseur ">".

**<COPY>** Recherche la chaîne de caractères **S1** [définie par la commande **S** du moniteur] à partir de la position du curseur ">".

Si la chaîne **S1** est trouvée, la page de texte où elle se trouve est affichée et la ligne est pointée par le curseur "\$". A ce moment là, la touche "S" permet de remplacer la chaîne **S1** du texte par la chaîne **S2** [définie elle aussi par la commande **S** du moniteur]. Ensuite la nouvelle phrase est affichée et la recherche de la chaîne **S1** se poursuit. Si la chaîne **S2** du moniteur, substituée à la chaîne **S1** du texte a donné une phrase incorrecte, la recherche ne se poursuivra qu'après que la phrase ait été corrigée. Toute autre touche que **S** appuyée quand le curseur "\$" est affiché fera réapparaître le curseur ">".

**!** Les trois commandes suivantes utilisent des pointeurs de texte matérialisés par le caractère "@". Ils doivent être les premiers et uniques caractères d'une ligne. L'assembleur les ignore.

**<CTRL> D** Efface tout le texte compris entre les 2 premiers pointeurs de texte qui sont eux même effacés.

**<CTRL> C** Duplique le texte se trouvant entre les premiers pointeurs de texte à la position actuelle du curseur ">". Les pointeurs de texte ne sont pas dupliqués.

**<CTRL> K** Détruit tous les pointeurs de texte.

## Les commandes de saisie du texte

**<DEL>** Efface le caractère avant le curseur.

**→** Avance le curseur d'un caractère.

**←** Recule le curseur d'un caractère.

## Mode de travail de l'éditeur

Il accepte des phrases de 80 caractères qui sont divisées en plusieurs zones LABELS, MNEMONIQUES, OPERANDES, COMMENTAIRES.

Ex : LOOP ADD A,[HL] ; *additionne à A la valeur de [HL]*

Un label doit être écrit en début de ligne, commencer par un caractère alphabétique et ne pas être le nom d'un registre. Sa longueur maximale ne doit pas excéder 8 caractères. Un mnémonique doit toujours être précédé d'un espace. La longueur maximale des opérandes est de 26 caractères. Après chaque retour de chariot, l'éditeur fait une analyse de la phrase au cours de laquelle il supprime les espaces inutiles et transforme les mots en codes. Ceci permet de réduire considérablement la taille du fichier texte et d'accélérer l'assemblage. Si une phrase n'est pas correcte, l'éditeur renvoie le curseur dans la zone erronée.


## Échanges de fichier texte

Pour assembler avec DAMS un fichier texte créé par un autre éditeur, il est nécessaire d'adapter ce fichier au format de D.A.M.S. L'exemple suivant montre comment, de façon générale, est codée une ligne.

Ex : LOOP LD HL, [LOOP + 5 ]

```
L O O P      L D H L , [ L O O P + 5 ]  
4C 4F 4F 50   80 48 4C 2C 28 4C 4F 4F 50 2B 35 29 0D
```

#0D est le caractère de fin de ligne et zéro est le caractère de fin de texte. Les mnémoniques du Z80 sont codés sur 1 octet d'après une table se trouvant à l'adresse de TEXT-242. Le premier mnémonique de cette table est "LD" dont le code est #80 et le dernier est la pseudo-instruction d'assemblage "END" qui a pour

code **#CC**. Le bit 7 du dernier caractère de chaque mnémonique est mis à **1** pour signaler la fin de mot. **#FF** est le code du ";" [commentaire] et peut être suivi par une chaîne de caractères alphanumériques. Pour plus de renseignements, écrivez une ligne à l'aide de l'éditeur et dumppez la avec la commande . Dans tous les cas, un simple programme basic utilisant la table des mnémoniques, permet d'adapter et de copier à l'adresse de **TEXT** un fichier texte, de manière à le rendre utilisable par **D.A.M.S**.

## L'ASSEMBLEUR


Les commandes  A et  F du moniteur permettent d'assembler le fichier texte. Cette opération consiste à lire le fichier texte [créé avec l'éditeur] qui constitue le programme source et à le transformer en une suite de codes [programme objet] qui pourront être directement exécutés par le microprocesseur. L'assemblage se fait en deux passes, le texte est examiné deux fois de suite. L'argument s des commandes  A et  F signale que le texte est organisé en blocs et qu'il est sur support magnétique. s est le nom du premier bloc. La commande  A signifie que le 1er bloc est déjà en mémoire et qu'il est inutile de le charger. Il doit aussi se trouver sur support magnétique pour la 2ème passe. La commande  F signifie que le 1er bloc n'est pas en mémoire et qu'il faut donc le charger.

### Les options

L'argument n des commandes  A et  F définit les options choisies.


1. Indique à l'assembleur qu'il doit utiliser la mémoire écran de 16 Ko pour y loger la table des symboles. Ceci est utile lorsque la taille du fichier texte est très importante ou lorsque D.A.M.S est exécuté à une adresse haute. Quand cette option est sélectionnée, le premier label absent arrête l'assemblage. Si la mémoire écran ne suffisait pas à loger tous les labels, l'assemblage serait stoppé par **l'erreur 9**.
2. Indique à l'assembleur qu'il doit déposer le code objet à la suite de la table des symboles ou si l'option l'est aussi sélectionnée, à la suite de l'adresse Hmem donnée par la commande  X. Toutefois, le texte sera assemblé pour être



exécuté à l'adresse donnée par le ORG figurant dans le texte. Lorsque l'on a accédé à l'assemblage par la commande , **l'option 2** est obligatoirement sélectionnée. Dans ce cas, l'assembleur demande à quelle adresse il doit déposer le code objet.

**! Ces 2 options peuvent être sélectionnées en même temps en les additionnant.**

## La table des symboles

C'est la table qui contient les labels et leurs valeurs. Elle est créée au cours de la lère passe de l'assemblage et elle se trouve à l'adresse de Hmem ou à #C000 si **l'option 1** est sélectionnée. Après un assemblage, elle est encore utilisable et peut être visualisée [et imprimée] par la commande . Si l'option 1 est sélectionnée, elle est détruite à la fin de l'assemblage. Au cours de sa création, si elle tentait de surcharger le code objet ou de dépasser HRAM elle serait alors immédiatement recopiée dans la mémoire écran de manière à ne pas stopper l'assemblage.

## Pseudo-instructions d'assemblage

Outre le langage d'assemblage standard du Z80, D.A.M.S accepte les commandes d'assemblage suivantes. Elles doivent être entrées comme des mnémoniques et non comme des labels.

**ORG nn** : Définit l'origine ou l'adresse à laquelle devra être assemblé le texte. Si l'option 2 est sélectionnée. ORG ne sert qu'à donner une valeur aux labels. Si ORG est omis ou si l'option 2 est sélectionnée, D.A.M.S affiche l'adresse où il dépose le code objet.

**Label EQU nn** : Le symbole "Label" prend la valeur nn, nn doit être une constante ou un label déjà défini.

**DEFB n1, n2, n3...** : Chaque octet de mémoire prend la valeur **n**.

**DEFW nn1, nn2, nn3...** : Chaque groupe de 2 octets prend la valeur **nn**.

**DEFS nn, n** : Saute **nn** octets. Si **n** est présent, alors les **nn** octets sont remplis par la valeur **n**.

**DEFM s** : Les octets suivants sont remplis par la chaîne **s**.

**ENT nn** : Signale à la commande  du moniteur que le point d'entrée du programme sera l'adresse **nn**.

**IF *expression*** : Évalue l'expression. Si l'expression est égale à **0**, alors l'assemblage est invalidé jusqu'à l'apparition d'un **ELSE** ou d'un **END**.

**ELSE** : Valide ou invalide l'assemblage.

**END** : Valide l'assemblage des lignes qui suivent.

## L'assemblage par blocs

Lorsque le fichier texte dépasse une dimension de **20 Ko**, il devient nécessaire de le découper en blocs pour l'assembler, sinon il n'y a plus assez de mémoire pour contenir le code objet et la table des symboles. Chaque bloc à assembler doit se terminer par la commande d'assemblage **\*F,s** [la chaîne **s** étant le nom du bloc suivant]. Cette commande d'assemblage doit être frappée en zone label et doit être la seule instruction de la ligne.

**! Le dernier bloc ne doit pas comporter cette commande d'assemblage.**

**! Notez qu'un lecteur de disquette est presque indispensable pour l'assemblage par blocs, car chaque bloc doit être chargé 2 fois pour les 2 passes.**

## Les expressions

Les expressions sur lesquelles DAMS travaille doivent être une suite de termes séparés par des opérateurs. Les termes et les opérateurs acceptés par D.A.M.S sont les suivants :

- **Termes :**

Ex : **12**            Constante numérique décimale.

Ex : **#20**           Constante numérique hexadécimale.

Ex : **%10110**      Constante numérique binaire.

Ex : **"a"**            Constante numérique code **ASCII** du caractère a

Ex : **VALEUR**      Étant un label auquel un assemblage a donné une valeur.

Ex : **\$**              Représente la valeur actuelle du compteur d'adresse.

Ex : `::x` Donne le contenu de l'octet adressé par `x`  
[PEEK].

Ex : `:::x` Donne le contenu des **2** octets adressés par `x`  
[DEEK].

• **Opérateurs :**

+ addition

- soustraction

\* multiplication

/ division

& élément logique ET

~ élément logique OU

! élément logique OU exclusif

? fonction modulo,  $a ? b = a - b * |N| [a / b]$

## Les erreurs au cours de l'assemblage

Dès que D.A.M.S rencontre une erreur il stoppe l'assemblage et affiche un message d'erreur. **Accédez à l'éditeur par un "L" et la ligne pointée par le curseur ">" est la ligne où est advenue l'erreur.** Si l'option 1 n'est pas utilisée, un label absent ne stoppe pas l'assemblage et est simplement signalé.

## LA TRACE

**E**lle permet d'exécuter un programme instruction par instruction et de visualiser le contenu des registres du processeur après chaque instruction. C'est un moyen efficace pour trouver des erreurs dans un programme en langage machine.


### La commande "T"

Lorsque vous entrez "T", le contenu des registres et les **16** premiers octets de pile sont affichés. Vous pouvez appuyer sur "Q" ou  pour revenir au moniteur. Si vous appuyez sur une autre touche, l'instruction adressée par le registre PC est désassemblée, exécutée, et le contenu des registres est à nouveau affiché. Ainsi, instruction après instruction, vous pouvez voir si votre routine a l'effet souhaité. Avant d'être exécutée, chaque instruction est testée afin d'éviter la destruction de D.A.M.S. Toute tentative de modification de la zone de mémoire comprise entre l'adresse de début de DAMS et Hmem, entraîne l'erreur **12**.

Les instructions qui modifient la configuration RAM/ROM modifient aussi les indicateurs de haut de page. Au cours d'une trace, vous pouvez revenir au moniteur pour, par exemple, modifier une valeur dans un registre et continuer de tracer en rentrant "T"; mais si vous avez utilisé une commande qui modifie le compteur d'adresse, comme D, A, K, H, il faudra continuer de tracer en rentrant "Tnn", nn étant l'adresse où vous aviez interrompu le traçage.

## La commande "R"

Permet de tracer en mode rapide. Le désassemblage et le contenu des registres ne sont pas affichés mais les instructions sont testées avant d'être exécutées. Lorsque vous entrez cette commande, D.A.M.S mémorise la valeur du registre SP et le retour au mode "trace normale" se fera lorsque le registre SP aura la **valeur mémorisée plus 2**, c'est-à-dire, lorsqu'il aura exécuté un RET ou un POP etc...

La commande "R" n'est donc pas critique et peut être lancée dans de nombreux cas sans danger. Si "R" ne revenait pas, [par exemple le programme boucle],  permettrait de le stopper et de revenir au mode "trace normal".

## La commande "J"

"J" permet d'exécuter une routine en langage machine en mémoire RAM. J\$ exécute une routine à l'adresse que contient le PC, et dans ce cas, \$ indique que la routine se trouve en ROM ou en RAM suivant les indicateurs de haut de page et la valeur du registre PC.

**! J\$ ne modifie pas la valeur des registres BC' et AF'.**

Pendant une trace il est donc préférable d'utiliser J\$. Avant d'exécuter une commande J, s'assurer que le registre **SP a une valeur comprise dans les 32 Ko** centraux de RAM.

**! R peut être exécuté à n'importe quel endroit d'une routine, mais J doit être exécuté au début d'une routine qui n'a pas d'erreur.**

## LES MESSAGES D'ERREUR

L'affichage d'un message d'erreur provoque le retour au moniteur. Voici la liste des messages d'erreurs et des cas dans lesquels ils peuvent apparaître :

### 0. Syntax Error

- une commande n'a pas été reconnue ou sa syntaxe est erronée.
- au cours d'un assemblage, une ligne est erronée.

### 1. Redefine label Error

- au cours d'un assemblage.

Un label porte déjà le même nom et un label ne peut avoir 2 valeurs.

### 2. Relative number Error

le déplacement relatif d'un JR ou l'index d'un registre IX ou IY a une valeur hors intervalle.

### 3. Integer out of range Error

cette erreur se produit dans tous les cas où devrait figurer un nombre compris dans les intervalles 0, +255 ou 0, +65535 et vous n'avez pas respecté cet intervalle.

### 4. Illegal character Error

au cours d'un assemblage, un caractère d'un label ou de la ligne est illégal.

## 5. Illegal reference Error

- au cours d'un assemblage :
  - 1] Une expression qui suit un **EQU**, un **ORG** ou un **DEFS**, comporte un label qui n'a pas été défini.
  - 2] Il a été trouvé la commande d'assemblage **\*F,s** alors que le fichier texte n'est pas assemblé par bloc.

## 6. Bad ORG Error


- au cours d'un assemblage :
  - 1] La valeur qui suit le **ORG** est une adresse occupée par **D.A.M.S**.
  - 2] L'adresse d'origine est inférieure à **D.A.M.S** mais l'ampleur du code objet a tenté de surcharger **D.A.M.S**.


## 7. I/O Error


clôture toute opération avec un périphérique qui a échoué.

## 8. Bad memory Error

- lors du chargement d'une source :

La taille du fichier texte que vous essayez de charger est supérieure à la taille de mémoire actuelle [commande ].
- sous l'éditeur :

Il n'y a plus de place pour rajouter cette ligne. Vous pouvez peut être faire de la place avec la commande .
- pendant un assemblage :

Le code objet a tenté de dépasser la limite supérieure de la mémoire [**HRAM**]. Pour y remédier, réassemblez avec l'option  ou rechargez **D.A.M.S** à une adresse plus basse ou modifiez **HRAM**.



- l'argument de la commande **M** est trop élevé et est supérieur à HRAM :

Lors de la création de D.A.M.S, HRAM a été fixé à **#A67C** mais cette valeur peut être changée pour assembler des programmes à une adresse plus haute ou pour réserver de la place pour un autre logiciel. L'adresse de la variable système qui contient l'adresse de HRAM se trouve à l'adresse de **D.A.M.S +80**.

## 9. No Table space Error

- au cours d'un assemblage :

L'option **1** a été sélectionnée et la table de symboles nécessite plus de **16 Ko**.

## 10. Break Error

vous avez interrompu une opération

## 11. Label absent Error

cette erreur ne se produit jamais au cours d'un assemblage mais lorsque vous entrez une commande dont une expression contient un label qui n'est pas dans la table des symboles.

## 12. Bad location Error

- une commande telle que **Q** ou **W** a tenté de modifier un octet occupé par D.A.M.S.
- au cours d'une trace, [commande **T** ou **R**], une instruction telle que **LD (HL),a**, **LDDR** etc... aurait détruit ou compromis D.A.M.S.
- l'argument de la commande **M** est trop bas.

## EXEMPLE D'UTILISATION

**V**ous venez de charger D.A.M.S, le fichier texte est vide, le moniteur attend une commande.

Entrez "L" et tapez le programme suivant :

```
;
      ORG 40000 ; le programme sera logé à l'adresse 40000
DEBUT
      ENT $
;
      LD A,"A" ;
      LD B,26
LOOP
      CALL #BB5A ; Affiche le cc contenu dans A
      INC A
      DJNZ LOOP ; Boucle 26 fois
      RET
FIN
```

- Assemblez-le : A <ENTER>
- Exécutez-le : J <ENTER>
- Désassemblez-le : DDEBUT <ENTER>
- Tracez-le : TDEBUT <ENTER>

Appuyez sur <ENTER> jusqu'à ce que le PC affiche **#BB5A**. A ce moment là, si vous avez le temps, continuez à appuyer sur <ENTER> et vous allez tracer la routine de la ROM qui affiche un caractère à l'écran, sinon, appuyez sur "Q" et puis R <ENTER>. Faites 2 ou 3 fois le tour de la boucle de cette façon, puis après le INC A tapez "Q" et R <ENTER>. La boucle va alors se tracer seule jusqu'à ce que le registre B soit égal à zéro. A ce moment là, le contenu des registres s'affiche et le PC a pris la valeur qui était au sommet de la pile avant de tracer la routine. Le RET a donc été exécuté.

**! Pour revenir au moniteur, tapez "Q".**

Tapez maintenant : **HDEBUT** <ENTER>

**D.A.M.S** demande : Text ? tapez <ENTER>

**D.A.M.S** demande : End ? tapez **FIN** <ENTER>

Aussitôt vous vous retrouvez dans l'éditeur et vous pouvez constater que les codes machines ont été désassemblés et inclus au texte, à partir de la dernière position du curseur ">". Un label a aussi été créé. Un label commençant par un "R" est une adresse à laquelle aboutit un saut relatif. Si un label occupe une ligne à lui seul c'est que plusieurs sauts relatifs aboutissent à cette adresse.

**! Une adresse appelée par un call ou un jump engendre une ligne  
● commentaire vide et un label commençant par un "W".**

## ADRESSES UTILES DE D.A.M.S

**D**AMS étant relogeable ce ne sont pas les adresses qui sont données mais les déplacements relatifs à l'adresse d'implantation suivis du nom de la variable et du nombre d'octets qu'elle occupe.


### +37 PTEXT 2 octets

Stocke l'adresse du début de la ligne pointée par le curseur ">".

### +42 FLG1 1 octet

Le bit 1 de cette variable est à 1 si un assemblage a été lancé avec l'option 2. L'expression "IF : FLG1&2" dans le fichier texte validera l'assemblage des lignes qui suivent si l'option 2 est sélectionnée. L'expression "IF FLG1 & 2 | 2" invalidera l'assemblage si l'option 2 est sélectionnée. **Le bit 5 à 1 indique que l'assemblage se fait par blocs.**

### +80 HRAM 2 octets

Stocke la dernière adresse à laquelle D.A.M.S peut déposer du code objet ou peut fixer la valeur de Hmem [commande 

### +43 FLG2 1 octet

Le bit 1 est le flag "décimal / hexadécimal". Le bit 7 est le flag "Printer allumé".

### +10243

C'est l'adresse de début de la table qui contient les mnémoniques.

### +2354

C'est le point d'entrée du moniteur.