

TELE-ASS

ASSEMBLEUR SYMBOLIQUE
pour
ORIC TELESTRAT

Logiciel et manuel réalisé par Pierre GRENET
(manuel re-saisi par F.FRANCES)

© 1987 ORIC INTERNATIONAL

Page	Contenu
1	Table des matières
2	Préliminaires - Syntaxe des valeurs et adresses
3	Programme-source - Les symboles
4	Programme-source - Mnémoniques et pseudo-opérations
5	Programme-source - Pseudo-opérations (suite)
6	Programme-source - Syntaxe opérandes des mnémoniques
7	Liste des commandes directes du moniteur
9	Commandes utilitaires diverses et commandes du DOS
11	Commandes du source
12	Commandes de mise au point
15	Commandes d'assemblage et gestion des symboles
17	Liste des messages d'anomalies
18	Différenciation symboles locaux et globaux

- Les commandes du moniteur et leur syntaxe ne nécessitent aucun espace dans le texte et peuvent s'écrire indifféremment en majuscules ou en minuscules.

- Sauf indication contraire, les adresses ou valeurs sur 2 octets utilisées dans la syntaxe des commandes directes ou dans le programme source peuvent être écrites dans les bases ou formats suivants:

. Décimal: Valeur de 0 à 65535

. Hexadécimal: Signe '#' ou '\$' suivi d'une valeur de 0 à FFFF
----- (uniquement '\$' dans un programme source)

. Binaire: Signe '%' suivi d'une valeur de 0 à 1111111111111111

. Symbole: Nom du symbole (global, local ou du moniteur).

- De même, les valeurs sur 1 octet, outre les formats ci-dessus, peuvent être écrites au format:

. ASCII: Double quote (") suivi d'1 caractère ASCII

. ou en extrayant la partie haute ou basse d'une valeur ou d'une adresse quelconque (en général d'un symbole) en la faisant précéder du signe '<' pour la partie basse et du signe '>' pour la partie haute.

- Ces adresses et valeurs peuvent former entre elles des expressions numériques par les seuls opérateurs '+' et '-'.

- Cette possibilité d'écriture de valeurs sur 1 octet est limitée au source et à quelques commandes directes.

- Pour effectuer une opération dont un des opérandes est un caractère ASCII, le premier opérande doit être numérique:

. ex: 150 motcle BYT "CHAIN",128+"E

. ex: ?HEX 128+"E

- La saisie d'une ligne source s'effectue de la même façon qu'en Basic c'est à dire en indiquant un numéro de ligne (en décimal) suivi du texte source.
- Une ligne du programme source peut être une ligne de commentaires à elle seule. Dans ce cas, la ligne débute par le signe quote (').
- Autrement, une ligne source doit être composée dans l'ordre:
 - . d'une étiquette (un symbole) en général facultatif,
 - . du nom correspondant à un code opération (mnémonique) ou à une pseudo opération (fonction spéciale) - obligatoire,
 - . des opérandes appropriés - obligatoire,
 - . d'un commentaire (facultatif) commençant par le signe quote (').

I - LES SYMBOLES:

- Tout NOM représentant une VALEUR ou une ADRESSE.
- Les symboles du MONITEUR sont présents en permanence.
- Les symboles GLOBAUX utilisateur sont créés par l'assemblage et peuvent rester présents en mémoire sous certaines conditions (Voir fonctions d'assemblage et de gestion table des symboles)
- Les symboles LOCAUX utilisateur sont créés par l'assemblage et ne sont présents en mémoire que si le programme source n'est pas modifié, qu'un autre programme source n'est pas chargé en mémoire, ou qu'une nouvelle demande d'assemblage n'est pas intervenue.
- Ces SYMBOLES doivent comporter un maximum de 6 caractères sans espace.
 - . Le 1er caractère DOIT être un caractère alphabétique majuscule ou minuscule ou un des caractères: . ? à [\] ^ £ ç
 - . Pour les caractères suivants, on peut utiliser, en plus, les chiffres 0 à 9.
 - . Les trois premiers caractères ne doivent pas correspondre à un nom de code opération ou de pseudo-opération.
 - . Une de ces causes peut générer le message "ETIQUETTE INVALIDE".

Exemples de symboles invalides: (la cause est soulignée)

```

BYTES  STXTPT  TAXES  %AUGMT  TST>01  hexa$  RESERV  PLAN?
---
```

Exemples de symboles valides:

```

.BYTE  stxTPT  tAXES  POURCT  TST.01  hexa  [RESV]  ?PLAN
```


. RES : Syntaxe : RES val ou étiquette RES val

Permet de réserver le nombre d'octets 'val' à partir de l'adresse courante pendant l'assemblage. Les octets existants dans cette plage de mémoire réservée seront conservés.

Exemple: si PC = \$204A: RES 256 indiquera qu'aucun octet n'est à assembler depuis \$204A jusqu'à \$2149 inclus. La valeur suivante de PC sera l'adresse \$214A.

. BYT : Syntaxe: BYT sym,val,"ASCII",... ou étiquette BYT sym,val,"ASCII",...

Place en mémoire les éléments séparés par une virgule à partir de l'adresse courante pendant l'assemblage.

Ces éléments doivent être sur 1 OCTET et peuvent être des expressions numériques dans une base quelconque, des symboles ou des chaînes ASCII entre double-quotes.

Exemple: si PC = \$204A : étiquette BYT 1,255,"ABCD",0,\$20,%11111 assignera à 'étiquette' la valeur \$204A et placera en mémoire à partir de \$204A les valeurs: 01 FF 41 42 43 44 00 20 1F

. DBT : Syntaxe: DBT sym,val,... ou étiquette DBT sym,val,...

Place en mémoire les éléments séparés par une virgule à partir de l'adresse courante. Ces éléments seront considérés comme étant sur 2 OCTETS et placés dans l'ordre OCTET HAUT puis OCTET BAS. Ces éléments doivent être numériques (symboles compris).

Exemple: si PC = \$204A : DBT \$CCD9,1,\$2F1,%11111,0 mettra en mémoire à partir de \$204A : CC D9 00 01 02 F1 00 1F 00 00

. WRD : Syntaxe: WRD sym,val,... ou étiquette WRD sym,val,...

Place en mémoire les éléments séparés par une virgule à partir de l'adresse courante. Ces éléments seront considérés comme étant sur 2 OCTETS et placés dans l'ordre OCTET BAS puis OCTET HAUT.

Ces éléments doivent être numériques (symboles compris).

Exemple: si PC = \$204A : étiquette WRD \$CCD9,1,\$2F1,%11111,0 assignera à 'étiquette' la valeur \$204A et placera en mémoire à partir de \$204A les valeurs: D9 CC 01 00 F1 02 1F 00 00 00

IV- LES OPERANDES DES CODES OPERATION (Syntaxe des mnémoniques)

- La syntaxe dépend du mode d'adressage désiré. Se reporter éventuellement à un ouvrage traitant de l'assembleur du 6502.
- Pour mémoire, le tableau ci-dessous donne quelques exemples de syntaxes suivant les différents modes d'adressage du 6502.

Modes d'adressage 6502	Exemples d'écriture		
Implicite et accumulateur	SEC	ASL	CLD
Immédiat	LDY #\$FF	EOR #%10000000	CMP #"A
Indirect pré-indexé X	STA (\$33,X)	LDA (TOTO,X)	SBC (01,X)
Indirect post-indexé Y	LDA (\$E9),Y	STA (50),Y	ADC (CHIF),Y
Relatif	BNE .étiqu	BCC +3	BMI Etiqu+3
Absolu	CMP \$9A	LSR \$2F1	ROR 2749
Absolu indexé X	LDA \$35,X	LDA CHAINE,X	CMP \$B800,X
Absolu indexé Y	STA \$BBD0,Y	STA ECRAN+40,Y	ROL 10,Y
Indirect pur	JMP (\$FFFE)	JMP (inter)	JMP (244)

NB. Le mode d'adressage 'accumulateur', qui concerne principalement les opérations de rotation et de décalage - LSR ASL ROR ROL -, est souvent syntaxé de la lettre 'A' (ex: ASL A, ROL A, ...). Pour des raisons d'interprétation, il a été retenu pour cet assembleur une syntaxe nulle identique au mode "implicite"(ex: ASL, ROL).

- . Le mode d'adressage de BRK a été modifié en mode "page 0" du fait de la particularité du TELESTRAT qui permet l'accès aux routines moniteur à l'aide de ce mnémonique.

```
ex:   10 affich LDA #"A
      20          BRK XWR0
```

-o-o-o-o-o-o-

Nota. L'écriture d'une ligne source s'effectue indifféremment en majuscules ou en minuscules mais toute référence à un symbole local ou global devra s'écrire tel que le symbole aura été saisi. Par contre, l'écriture d'une valeur hexa, d'un symbole moniteur ou d'un mnémonique peut être indifféremment majuscules ou minuscules.

- Liste des commandes et syntaxe. Les options sont indiquées entre parenthèses et l'option préfixe (L) sert à adresser l'imprimante au lieu de l'écran.
. val = valeur décimale, hexa (\$ ou #), binaire (%) ou symbole
. Les options sont entre parenthèses

..... Commandes utilitaires

- . QUIT
- . ?DEC val
- . ?HEX val
- . ?BIN val
- . ?CAR val
- . ?BANK (val)
- . FPAGE (LPRFY) (,LPRSY)
- . DPAGE
- . SLIGNE
- . MOVE adrdeb, adrfin (,Bval1) , adrcib (,Bval2)
- . Commandes clavier AZERTY, QWERTY, FRENCH, ACCSET, ACCOFF

..... Commandes du DOS

- . (L)DIR ("NFA")
- . LOAD "NF" (,A EN) (,V) (,N)
- . SAVE (O/U/M) "NF" (,A EN ,E EN (,T EN))
- . DEL "NFA"
- . DELBAK
- . EXT ("ext")
- . BACKUP
- . INIT

..... Commandes Source

- . (L)LIST (numdeb) (-) (numfin)
- . RENUM (num1) (,pas) (,numdeb) (,numfin)
- . NEW
- . OLD
- . DELETE (numdeb) (-) (numfin)
- . MERGE "NF"
- . SEEK "chaîne" (,L)
- . CHANGE "chaîne1","chaîne2"

..... Commandes mise au point

- . (L)DESAS adrdeb (,adrfin) (,Bval)
- . (L)DUMP adrdeb (,adrfin) (,Bval)
- . VREG (,Aval) (,Yval) (,Xval) (,Pval)
- . CALL adrdeb (,Aval) (,Yval) (,Xval) (,Pval) (,Bval)
- . MODIF adrdeb (,Bval)
- . TRACE adrdeb (,Sadrfin) (,H) (,E) (,Bval) (,Aval) (,Yval) (,Xval) (,Pval)

..... Commandes d'assemblage

- . (L)ASSEM (adrdeb) (,G) (,L) (,S)
- . (L)SYTAB (L) (,G) (,M)
- . SYDEF (adrdeb, adrfin (,C))
- . SYOLD adrdeb, adrlim
- . MINAS adrdeb

QUIT	QUIT - Sortie du moniteur, retour au Basic par JMP \$C000.
?DEC	?DEC val ou ?DEC (adr) - Conversion de la valeur 'val' ou du contenu de 'adr' en décimal
?HEX	?HEX val ou ?HEX (adr) - idem pour conversion en hexadécimal
?BIN	?BIN val ou ?BIN (adr) - idem pour conversion en binaire
?CAR	?CAR val ou ?CAR (adr) - idem pour conversion en caractère ASCII avec 'val' sur 1 octet
	NB. Aux valeurs "val" décrites plus haut, peuvent s'ajouter: - >val pour extraire la partie haute de "val", - <val pour extraire la partie basse de "val", - "car" pour la correspondance ASCII du caractère "car", - (val) pour le contenu de l'adresse "val", - des combinaisons d'additions et de soustractions.
BANK	BANK (val) - Initialisation (val=0 à 7) de la banque par défaut. Si aucune valeur n'est indiquée, banque 0 par défaut.
FPAGE	FPAGE (LPRFY) (,LPRSY) - Affichage ou initialisation du nombre de lignes par page de l'imprimante (LPRFY) et du numéro de ligne de référence pour effectuer un saut de page imprimante (LPRSY). - (LPRFY≤255) ≥ (LPRSY≤255) - Si initialisation, le compteur de ligne imprimante est remis à 0
DPAGE	DPAGE - Effectue un saut de page imprimante suivant position courante.

SLIGNE	<p>SLIGNE</p> <ul style="list-style-type: none"> - Effectue un saut de ligne à chaque appui sur RETURN. Dès qu'une autre touche est pressée, le processus s'arrête.
Commandes clavier	AZERTY, QWERTY, FRENCH, ACCSET, ACCOFF (idem BASIC)
MOVE	<p>MOVE adrdeb ,adrfin (,Bval1) ,adrcib (,Bval2)</p> <ul style="list-style-type: none"> - Déplace la zone mémoire de la banque val1 comprise entre adrdeb (incluse) et adrcib (exclue) vers la banque val2 à partir de l'adresse adrcib. Si un option est omise => Banque par défaut.
-----Commandes du DOS-----	
(L)DIR	<p>(L)DIR ("NFA")</p> <ul style="list-style-type: none"> - Listage du directory. Attention: si LDIR, l'imprimante n'est pas gérée (Refaire le cas échéant "FPAGE").
LOAD	<p>LOAD "NF" (,A EN) (,V) (,N)</p> <ul style="list-style-type: none"> - Chargement du fichier de nom "NF". - (,A EN) si chargement imposé à l'adresse EN. - (,V) si visualisation des adresses début-fin du fichier. - (,N) si chargement forcé sans exécution.
SAVE	<p>SAVE (O/U/M) "NF" (,A EN ,E EN (,T EN))</p> <ul style="list-style-type: none"> - Sauvegarde du fichier "NF". Options identiques STRATSED.
DEL	<p>DEL "NFA"</p> <ul style="list-style-type: none"> - Détruit le(s) fichier(s) "NFA". Idem STRATSED.
DELBAK	<p>DELBAK</p> <ul style="list-style-type: none"> - Détruit tous les fichiers d'extension ".BAK"
EXT	<p>EXT ("ext")</p> <ul style="list-style-type: none"> - Visualisation ou initialisation de l'extension par défaut(3 car)
BACKUP	<p>BACKUP</p> <ul style="list-style-type: none"> - Copie intégrale d'une disquette. Idem STRATSED.
INIT	<p>INIT</p> <ul style="list-style-type: none"> - Initialisation et formatage d'une disquette. Idem STRATSED.

(L)LIST	LIST (numdeb) (-) (numfin) - Listage du programme source présent en mémoire. Peut être interrompu momentanément par ESPACE et définitivement par CTRL C. - 'numdeb' et 'numfin' sont soit des numéros de lignes en décimal, soit des étiquettes du programme source.
RENUM	RENUM (num1) (,pas) (,numdeb) (,numfin) - Renumerotation des lignes du programme source de 'numdeb' à 'numfin' commençant par 'num1' avec un incrément de 'pas'. - 'numxxx' sont soit des numéros de ligne source en décimal, soit des étiquettes du programme source. ('pas' uniquement décimal)
NEW	NEW - "Efface" le programme source en mémoire et la table des symboles locaux. (Ne fait que réajuster les différents pointeurs)
OLD	OLD - "Réactive" le programme source en mémoire (après un NEW).
DELETE	DELETE (numdeb) (-) (numfin) - Delete les lignes source de "numdeb" à "numfin". - 'numdeb' et 'numfin' sont soit des numéros de ligne en décimal soit des étiquettes du source.
MERGE	MERGE "NF" - Met le programme source "NF" présent sur disquette à la suite du programme source en mémoire, les lignes sont renumérotées.
SEEK	SEEK "chaîne" (,L) - Recherche la chaîne entre double-quotes (obligatoires) dans le programme source et indique le nombre d'occurrences trouvées. - si l'option (,L) est indiquée, les lignes sources sont listées.
CHANGE	CHANGE "chaîne1","chaîne2" - Remplace toutes les occurrences de "chaîne1" trouvées dans le programme source par "chaîne2". (Chaînes entre double-quotes) (les chaînes ne peuvent comporter elles-mêmes de double-quotes)

(L) DESAS	<p>DESAS adrdeb (,adrfin) (,Bval)</p> <ul style="list-style-type: none">- Listage du désassemblage de la mémoire depuis l'adresse 'adrdeb' qui peut être interrompu momentanément par ESPACE et définitivement par ESCAPE ou CTRL C.- Si l'option (,adrfin) est indiquée, le listage s'arrêtera après désassemblage de cette adresse.- Si l'option (,Bval) est indiquée, le désassemblage se fera sur la banque "val", sinon sur la banque par défaut.
(L) DUMP	<p>DUMP adrdeb (,adrfin) (,Bval)</p> <ul style="list-style-type: none">- Listage du contenu hexadécimal et caractère de la mémoire depuis l'adresse 'adrdeb' qui peut être interrompu momentanément par ESPACE et définitivement par ESCAPE ou CTRL C.- Si l'option (,adrfin) est indiquée, le listage s'arrêtera après affichage complet de la ligne 'dumpée' incluant cette adresse.- Si l'option (,Bval) est indiquée, le désassemblage se fera sur la banque "val", sinon sur la banque par défaut.
VREG	<p>VREG (,Aval) (,Yval) (,Xval) (,Pval)</p> <ul style="list-style-type: none">- Visualisation ou/et imputation des valeurs dans les registres A, Y, X et P du 6502 où 'val' est une expression quelconque.- Ces registres seront réellement chargés lors de l'exécution d'une routine assembleur par les fonctions CALL ou TRACE qui mettent à jour ces registres.
CALL	<p>CALL adrdeb (,Bval) (,Aval) (,Yval) (,Xval) (,Pval)</p> <ul style="list-style-type: none">- Exécution de la routine "adrdeb" sur la banque "val" ou sur la banque par défaut si non précisée.- Les valeurs des registres 6502 créés par VREG ou par les options de CALL sont présents à l'entrée de la routine appelée et l'état de ces registres est indiqué en entrée et en sortie de cette routine- La routine appelée doit se terminer par un RTS et toute anomalie 'plantera' le 6502.

MODIF

MODIF adrdeb (,Bval)

- Modification pleine-page du contenu hexadécimal ou caractère de la mémoire depuis l'adresse 'adrdeb'. 216 octets sont affichés à l'écran sous les 2 formes. Après une modification, le curseur se déplace automatiquement vers la zone suivante, et le contenu de l'équivalent hexa ou caractère est modifié en conséquence.
- Si l'option (,Bval) est indiquée, les modifications se feront sur la banque "val" (si celle-ci est une RAM !)
- Gestion du déplacement du curseur et commandes disponibles:
 - . ↑ : déplacement vers le haut (scrolling si haut écran)
 - . ↓ : déplacement vers le bas (scrolling si bas écran)
 - . → : déplacement vers la droite
 - . ← : déplacement vers la gauche
 - . SHIFT → : positionne le curseur en fin de ligne courante
 - . SHIFT ← : positionne le curseur en début de ligne courante
 - . SHIFT ↓ : visualisation des 216 octets suivants
 - . SHIFT ↑ : visualisation des 216 octets précédents
 - . RETURN : positionne le curseur au début de la ligne suivante
 - .SHIFT RET: positionne le curseur au début de la ligne précédente
 - . ESCAPE : Sortie commande MODIF (ou CTRL C)
 - . CTRL B : Modification de la banque à traiter
 - . CTRL L : régénère l'écran avec les dernières valeurs mémoire
 - . CTRL A : positionne le curseur d'une zone hexadécimale vers la zone caractère correspondante et vice-versa (bascule)
 - . CTRL T : Bascule majuscules/minuscules
 - . CTRL V : Caractère ASCII en vidéo inverse

TRACE TRACE adrdeb (,Sadrstop) (,E) (,H) (,N) (,Aval) (.....) (,Bval)

- Effectue la trace d'une routine assembleur commençant à l'adresse 'adrdeb'. Quelles que soient les options précisées, la trace s'arrête dès le dernier mnémonique RTS rencontré qui correspond, logiquement, à la fin de la routine ou, éventuellement sur un message d'erreur (MNEMONIQUE INCONNU, PILE ERRONEE)

- Les options possibles sont les suivantes:

(,Sadrstop) = adresse d'arrêt momentané du listage en mode trace déroulante. Cette adresse doit correspondre à un mnémonique sur lequel la routine "passera".
(,E) = mode exécution sans affichage.
(,H) = mode trace en pas à pas (H comme Halte).
(,N) = ne liste que le désassemblage.
(,Bval) = pour tracer une routine sur la banque "val".
(,Aval)...= chargement préalable des registres (Cf.VREG)

- A chaque étape de la trace, l'état des registres et du sommet de la pile sont affichés ainsi que le désassemblage de la prochaine opération à effectuer. (sauf pour l'option N)

- En mode trace déroulante, possibilité d'arrêt momentané du listage par la touche ESPACE qui fonctionne en bascule (comme la commande LIST).

L'appui sur ESCAPE ou CTRL C stoppent le mode trace déroulante pour passer en mode trace pas à pas.

- En mode exécution sans affichage, l'appui sur ESCAPE ou CTRL C stoppe le mode exécution pour passer en mode trace pas à pas.

- En mode trace pas à pas, l'appui sur:

. ESPACE exécute l'instruction en cours,
. ESCAPE sort définitivement de la fonction TRACE (ou CTRL C),

et, si l'instruction à exécuter est un JSR, l'appui sur RETURN exécute (sans vérification) la routine indiquée et la trace reprend au retour de cette routine.

Si l'instruction n'est pas un JSR, l'instruction est exécutée comme après un appui sur ESPACE.

(L) ASSEM (L) ASSEM (adrdeb) (,G) (,L) (,S)

- Assemblage du programme source à l'adresse d'origine (qui doit être indiquée dans le programme source après ORG). Les différents calculs d'adresses sont TOUJOURS effectués depuis cette adresse origine.
- L'option (adrdeb) indique que l'implantation PHYSIQUE du programme objet en mémoire doit se faire à partir de cette adresse
- L'option (,G) permet de conserver les symboles globaux créés par les assemblages précédents ou d'en créer un.
- L'option (,L) permet d'avoir l'édition complète du désassemblage et du source) la fin de l'assemblage.
- L'option (,S) permet d'avoir l'édition complète des tables des symboles locaux et globaux.
- Avant assemblage physique, les adresses début et fin du programme source + des symboles locaux, du programme objet et de la table des symboles globaux sont indiquées et la question: 'Assemblage (O/N):' est posée. Au vu des adresses, vous pourrez facilement savoir si l'assemblage à l'endroit désiré risque de poser un problème ou non.

(L) SYTAB (L) SYTAB (,L) (,G) (,M)

- Liste les tables des symboles (ordre alphabétique sur 1 car.)
- Les options (,L), (,G) et (,M) sélectionnent respectivement la table des symboles (L)OCAUX, (G)LOBAUX ou du (M)ONITEUR. Si aucune option n'est indiquée, seule la table des symboles GLOBAUX est listée.
- Pour pouvoir lister les symboles locaux, le programme source les ayant engendrés doit être présent en mémoire et ne doit pas avoir été modifié.

SYDEF SYDEF (adrdeb, adrlim (,C))

- Sans option, SYDEF donne les adresses de la table des symboles GLOBAUX existante, sinon déplace cette table à l'adresse de début 'adrdeb' et lui donne 'adrlim' comme adresse limite absolue (Aucun symbole ne pourra être créé au-delà).
- L'option (,C) permet de recopier à partir de 'adrdeb' les symboles globaux existants.

- Si cette option est absente, la nouvelle table sera réinitialisée (sans symbole).
- A l'origine, cette table est en \$3000 (adrdeb) et sa limite est fixée à \$3FFF (adrlim). => création possible de 512 symboles.
- Seule vérification effectuée: le déplacement de cette table ne doit pas empiéter sur le programme source ni sur la table des symboles locaux en cours (adresse basse) ni dépasser l'adresse \$B400 (adresse haute) sinon le message "VALEUR HORS LIMITES" sera généré.

SYOLD

SYOLD (adrdeb, adrlim)

- Retrouve les symboles globaux (après sauvegarde et chargement par exemple) et réinitialise les adresses de début et de limite de la table avec 'adrdeb' et 'adrlim'. (valeurs courantes par défaut)

MINAS

MINAS adrdeb

- Mini-assembleur à assemblage direct (sans programme source) où 'adrdeb' est l'adresse de début d'implantation physique et logique du code objet.
- Après chaque RETURN, le prompt '→' apparaît et une ligne source peut être saisie. Un RETURN sur une ligne 'vide' fait sortir de cette commande.
- La syntaxe d'une ligne source est strictement identique à celle définie pour créer un programme source sans le numéro de ligne. Les lignes source entrées seront assemblées en séquence depuis l'adresse 'adrdeb' et le désassemblage apparaît à chaque ligne.
- Si des étiquettes sont indiquées, celles-ci seront toujours considérées comme des symboles GLOBAUX quelle que soit leur syntaxe et seront référencées dans la table correspondante.
- Si il est fait référence à un symbole, celui-ci doit exister dans la table des symboles GLOBAUX ou MONITEUR uniquement.
- Tous les mnémoniques et pseudo-opérations sont valides à l'exception de ORG (qui n'a aucun intérêt ici).
- Si une erreur est détectée, la ligne source incriminée n'a pas été assemblée et le processus s'arrête automatiquement.

- Les messages d'anomalies stoppent le déroulement de la commande en cours.
- Si un message est généré lors du processus d'assemblage, la ligne source incriminée sera listée.
- Ces messages s'affichent en haut et à gauche de l'écran et le curseur revient à l'endroit où l'interprétation n'a pas pu se faire.

ERREUR DE SYNTAXE par exemple oubli d'une parenthèse, de double-quotes, d'adresse obligatoire de commande, etc.

VALEUR ILLEGALE par exemple écriture erronée d'hexadécimal (ex: \$G22), de binaire (ex: %21001000), etc.

VALEUR HORS LIMITES par exemple une valeur ou addition de valeurs dépassant 2 octets (supérieure à \$FFFF), etc.

MEMOIRE PLEINE par exemple un programme source si long qu'à l'assemblage il n'y a plus de place pour créer des symboles, etc.

MNEMONIQUE NON DEFINI si une ligne source saisie ne comporte qu'une étiquette ou espace manquant entre étiquette et mnémonique.

ETIQUETTE ILLEGALE par exemple caractère '\$' ou '>' dans une étiquette.

ETIQUETTE EN DOUBLE pendant l'assemblage, si il est trouvé plusieurs fois la même étiquette locale.

SYMBOLE NON DEFINI pendant l'assemblage, si il est fait référence à une étiquette inconnue ou un symbole non défini.

BRANCHEMENT HORS LIMITES pendant la dernière passe de l'assemblage, un dépassement des capacités de saut.

MODE ADRESSAGE ILLEGAL par exemple une valeur de 2 octets pour l'adressage indirect indexé qui n'en demande qu'1 seul.

ORG NON DEFINI si la pseudo opération ORG (obligatoire) n'a pas été trouvée dans le programme source.

ORG DEJA DEFINI si la pseudo opération ORG est trouvée plusieurs fois dans le programme source.

PILE ERRONEE pendant un TRACE ou un EXEC si il est trouvé une opération qui dépile (PLA, PLP, RTS, RTI) et que la pile est insuffisante.

CODE INCONNU pendant un TRACE ou un EXEC si un code opération inconnu est rencontré.

+ les messages d'anomalies liés au DOS.

Le discours sur la différence entre symboles locaux et globaux n'a de sens que si l'on désire assembler plusieurs programmes source ayant des liens entre eux. Dans le cas contraire, on peut considérer qu'il y a équivalence entre symboles locaux et globaux et ne pas prêter attention à la syntaxe particulière de ces symboles.

Ces liens entre programmes source sont constitués de symboles, définis ou non, présents dans un programme source préalablement assemblé et auxquels on désire faire référence dans un autre programme source à assembler ultérieurement.

CREATION ET CONSERVATION DE CES SYMBOLES:

C'est l'option (,G) qui décide qu'il y a lieu lors de l'assemblage:

- . de faire la distinction entre symboles globaux et locaux, la différenciation s'effectuant par le 1er caractère du symbole qui doit être un point '.' pour un symbole local.
- . de prendre en compte les symboles globaux existants et, éventuellement, de les mettre à jour si des symboles existent en même temps dans le programme source et dans la table des symboles globaux.
- . de créer ceux qui n'existent pas dans cette table.

Ces symboles peuvent également être créés par la commande de mini-assemblage MINAS qui prend en compte les symboles prédéterminés par la pseudo-opération EQU ainsi que les symboles saisis en tant qu'étiquette.

- Pour 'vider' de son contenu la table des symboles, il suffit d'utiliser la commande SYDEF sans l'option 'C', la commande ASSEM sans l'option 'G', et, pour la reconstituer, la commande SYOLD.

EN RESUME:

- La table des symboles LOCAUX est détruite et recréée lors d'une demande d'assemblage quelles qu'en soient les options. Elle est également détruite dès qu'une modification du programme-source intervient, qu'un programme-source est chargé en mémoire ou que l'on a demandé à changer de place la table des symboles globaux.
Il n'est donc pas possible de faire référence à un symbole local d'un programme sur l'autre.
- La table des symboles GLOBAUX est créée, augmentée et mise à jour lors d'une demande d'assemblage avec option (,G) ou par la commande directe MINAS.
Il est donc possible de faire référence à un symbole global d'un programme sur l'autre ainsi que dans la syntaxe des commandes directes.
- Un symbole, quel qu'il soit, occupe 8 octets en mémoire.