ZX SPECTRUM
EDITEUR-ASSEMBLEUR

1. INTRODUCTION

Ce logiciel a pour but de faciliter l'utilisation du langage machine sur le ZX SPECTRUM grâce à l'emploi des mnémoniques Z80 et des étiquettes symboliques.

Il comprend deux parties :

— Un éditeur de textes permettant d'écrire et de modifier facilement les programmes écrits en langage d'assemblage.

— Un assembleur effectuant la traduction du langage d'assemblage en langage machine et générant le code binaire sur cassette ou directement en mémoire.

Il fonctionne sur ZX SPECTRUM 16K ou 48K, mais il n'est utilisable que sur ZX SPECTRUM 48K, où la taille mémoire disponible est suffisamment importante pour permettre l'écriture de programmes.

Il est entièrement écrit en langage machine pour assurer une plus grande efficacité et une plus grande souplesse d'utilisation.

Il occupe environ 8Ko en mémoire vive et il est fourni sur une cassette protégée.

Dans tout ce qui suit, les termes entre crochets ([et]) désignent des termes optionnels et les termes entre symboles d'inégalité (< et >) désignent des touches du clavier.

2. CHARGEMENT DE LA CASSETTE

Pour charger le programme, placez la cassette dans votre magnétophone et tapez sur votre SPECTRUM la commande suivante :

LOAD"" <ENTER>

Le programme se charge alors pendant quelques minutes, puis il démarre automatiquement.

Si une erreur de chargement se produit, le message « Tape loading error » s'affiche. Appuyez alors sur <BREAK> et recommencez le chargement en changeant au besoin le niveau de sortie de votre magnétophone.

Au début de l'exécution, le nom du programme et le message de copyright s'affichent en haut de l'écran. En dessous, apparaît le symbole > suivi du curseur invitant l'utilisateur à taper une commande.

Chaque fois que ce symbole sera affiché, cela indiquera que l'éditeur/assembleur est en attente d'une commande.

3. LE CLAVIER ET LES TOUCHES DE CONTRÔLE

Pour permettre l'écriture des programmes en langage d'assemblage, la gestion du clavier a été entièrement revue.

Toutes les touches du clavier ne génèrent que des caractères ASCII et non plus des instructions BASIC. L'appui de <SYMBOL SHIFT> et d'une touche permet de générer le symbole rouge dessiné sur la touche. L'appui de <CAPS SHIFT> et d'une lettre permet de générer le caractère minuscule correspondant.

La touche <CAPS LOCK> (<CAPS SHIFT> + <2>) commande le passage du mode majuscule au mode minuscule et inversement. En mode majuscule on obtient les majuscules directement et les minuscules en appuyant sur <CAPS SHIFT>; en mode minuscule on obtient les minuscules directement et les majuscules en appuyant sur <CAPS SHIFT>.

La touche <DELETE> (<CAPS SHIFT> + <O>) efface le caractère situé juste avant le curseur.

La touche <TAB> (<CAPS SHIFT> + <SYMBOL SHIFT>) provoque le saut du curseur sur la prochaine tabulation de l'écran. Les tabulations sont situées sur les colonnes 6, 14, 22, 30. Elles sont utilisées en langage d'assemblage pour faciliter la lisibilité des programmes.

La touche <BREAK> (<CAPS SHIFT> + <SPACE>) permet d'interrompre n'importe quelle commande exécutée par l'assembleur.

La touche <ENTER> indique la fin de l'écriture d'une ligne de commande ou de texte.

La touche <SPACE> permet de geler momentanément l'affichage sur l'écran. Elle sera utilisée avec les commandes A et P de l'éditeur qui font défiler du texte, pour permettre à l'utilisateur d'examiner ce qui s'affiche sur l'écran. L'affichage normal reprendra par l'appui sur n'importe quelle touche sauf <BREAK>.

4. L'ÉDITEUR DE TEXTES

L'éditeur de textes permet la manipulation de 10000 lignes, constituées d'au plus 127 caractères ASCII,et numérotées de 0000 à 9999. L'accès à une ligne se fait par son numéro qui fait office de clé.

Les commandes de l'éditeur sont constituées d'un caractère qui est l'initiale du nom de la commande, suivi d'un certain nombre de paramètres qui peuvent être des numéros de lignes.

L'éditeur gère une ligne courante qui représente la dernière ligne manipulée. Cette ligne est généralement prise par défaut dans les diverses commandes si l'utilisateur omet un numéro de ligne. Elle peut être spécifiée également par l'emploi du symbole ".".

Par ailleurs les symboles "#" et "$" représentent respectivement la première et la dernière ligne du texte.

Les touches ↓ (<CAPS SHIFT> + <6>) et ↑ (<CAPS SHIFT> + <7>) permettent de commander respectivement l'affichage de la ligne suivante et de la ligne précédente du texte. Ces touches ne sont effectives que si elles sont tapées au début de la ligne de commande.

Toutes les commandes de l'éditeur peuvent être écrites en minuscules ou en majuscules et peuvent contenir un nombre quelconque d'espaces ou de tabulations qui ne seront pas pris en compte. De même, les caractères situés après la commande ne sont pas pris en compte.

Voici la liste détaillée des commandes de l'éditeur de texte :

4.1. Branch

Syntaxe : B adresse

Provoque un branchement du micro—processeur à l'adresse indiquée. Cette adresse peut être notée en décimal, en hexadécimal (suiviede "H") ou en octal (suivie de "0").

Exemples :

B33450	Branchement à l'adresse 33450 en décimal
B7FOOH	Branchement à ladresse 7F00 en hexadécimal
B5E53H	Retour à l'assembleur avec effacement du programme en mémoire
B5E61H	Retour à l'assembleur sans effacement du programme en mémoire

Cette commande sera particulièrement utilisée pour exécuter des programmes en langage machine obtenus par assemblage en mémoire. En plaçant une instruction JP 5E61H à la fin de votre programme, vous reviendrez à l'éditeur/assembleur à la fin de son exécution.

4.2. Copy

Syntaxe : C ligne1,[ligne2],[ligne3]

Recopie du groupe de lignes compris entre ligne1 et ligne2, bornes incluses, juste après la ligne3. Tout le texte est ensuite renuméroté avec un incrément de un.

Le message «Paramètres incorrects» sera affiché si l'une des conditions suivantes se produit :

— ligne1 > ligne2
— ligne1 < = ligne3 <ligne2
— ligne3 n'existe pas

Si ligne2 est omise, l'éditeur prend la valeur ligne1 par défaut. Par conséquent seul ligne1 est copiée.

Si ligne3 est omise, la ligne courante est prise par défaut.

Exemple : C30,150,710 Recopie des lignes 30 à 150, juste après la ligne 710

4.3. Delete

Syntaxe : D [ligne1] [,ligne2]

Destruction des lignes comprises entre ligne1 et ligne2, bornes comprises.

Si ligne1 > ligne2, le message "Paramètres incorrects" est affiché.

Si ligne1 est omise, l'éditeur prend la ligne courante par défaut.

Si ligne2 est omise, il prend la valeur ligne1 par défaut.

Exemple : D100 Destruction de la ligne 100

4.4. Edit

Syntaxe : E [ligne]

Passage en mode édition sur la ligne spécifiée.

Si la ligne n'existe pas, le message "Ligne inexistante" est affiché.

Si la ligne est omise, l'éditeur prend la ligne courante par défaut.

En mode édition, vous pouvez déplacer le curseur sur la ligne à modifier grâce aux touches ← et →, détruire un caractère par <DELETE> ou insérer des caractères en frappant la touche correspondante.

La touche <ENTER> provoque l'arrét du mode édition avec affichage de la ligne modifiée.

Exemple : E Edition de la ligne courante

4.5. Find

Syntaxe : F [/chaîne]

Recherche de la chaîne spécifiée dans le texte à partir de la ligne suivant la ligne courante, et affichage de la ligne contenant la chaîne.

Si la chaîne n'est pas trouvée, le message "Chaîne non trouvée est affiché.

Si "/" est omis le message "Paramètres incorrects" est affiché.

Si la chaîne est omise, l'éditeur prend comme chaîne, la dernière chaîne recherchée. Ainsi, pour rechercher toutes les occurences d'une chaîne dans le texte, il suffit de faire un F/chaîne lorsque la ligne courante est la première ligne du texte, pour rechercher la première chaîne puis F pour rechercher les suivantes.

Exemple : F/DEB Recherche de la chaîne DEB

Remarque : Il est possible de placer un caractère de tabulation dans la chaîne à rechercher. Ainsi la commande F/DEB<TAB> permettra de rechercher la ligne où est définie l'étiquette DEB.

4.6. Hard

Syntaxe : H [ligne1] [,ligne2]

Affichage sur imprimante du bloc de lignes compris entre ligne1 et ligne2 bornes comprises.

Si ligne1 est omise, la ligne courante est prise par défaut.

Si ligne2 est omise, ligne1 est prise par défaut.

Si ligne1 > ligne2, le message "Paramètres incorrects" est affiché.

Si ligne2 est omise et si ligne1 n'existe pas l'éditeur affiche la ligne suivante.

Exemple : H#,$ Affichage de tout le texte sur imprimante

4.7. Insert

Syntaxe : I [ligne1] [,incrément]

Passage en mode insertion de texte à partir de la ligne1 si elle n'existe pas ou à partir de ligne1 + incrément si elle existe.

En mode insertion, l'éditeur affiche successivement tous les numéros des lignes à insérer suivis du curseur invitant l'utilisateur à écrire la ligne. Le passage d'un numéro de ligne à un autre se fait par l'ajout de la valeur incrément.

Pour arrêter le mode insertion, tapez sur <BREAK>.

Le mode insertion sera également interrompu avec affichage du message "Plus de place entre les lignes", si la ligne à insérer a un numéro supérieur à la prochaine ligne du texte.

Si ligne1 est omise, la ligne courante est prise par défaut.

Si l'incrément est omis, l'éditeur prend la dernière valeur de l'incrément utilisée. Au départ l'incrément est fixé à 10.

Exemple : I30,1 Insertion à partir de la ligne 30 avec un incrément de 1

4.8. Load

Syntaxe : L [nom]

Charge le programme source dont le nom est spécifié.

L'éditeur charge le premier programme de la cassette dont le nom commence par celui qui est inscrit dans la commande. Ce programme se place à la suite de celui qui était en mémoire sans l'effacer.

Pour conserver une numérotation croissante des numéros de lignes, il y aura intérêt à exécuter la commande Number à la suite du chargement lors de la concaténation de deux programmes.

Si le nom est omis, l'éditeur charge le prochain programme de la cassette.

Si le prochain programme de la cassette n'est pas un programme source en langage d'assemblage, le message "Type du fichier incorrect" est affiché.

Lors du chargement, le nom du programme trouvé s'affiche sur l'écran et le bord de l'écran clignote. Il est possible d'interrompre le chargement en appuyant sur <BREAK>.

Si une erreur de chargement se produit, le message "Erreur de chargement" sera affiché.

Exemple : LTEST Chargement du programme TEST

4.9. Number

Syntaxe : N [ligne1] [,[ligne2] [,[incrément]]]

Renumérotation du texte source à partir de ligne1 qui prend le numéro ligne2 et jusqu'à la fin du texte, en utilisant un pas valant incrément entre chaque ligne renumérotée.

Si ligne1 > ligne2, le message "Paramètres incorrects" est affiché.

Si ligne2 est inférieure à la ligne précédent ligne1, le message "Paramètres incorrects" est affiché.

Si les paramètres de la commande sont tels que le numéro de la dernière ligne soit supérieur à 9999, le message "Numero de ligne trop grand" est affiché et la renumérotation n'est pas effectuée.

Si ligne1 est omise, l'éditeur renumérote tout le texte.

Si ligne2 est omise, la valeur ligne1 est prise par défaut.

Si incrément est omis, la dernière valeur de l'incrément est utilisée.

Exemple : N,10,10 — Numérotation de tout le texte qui est placé à partir de la ligne 10 avec un incrément de 10

4.10. Print

Syntaxe : P [ligne1] [,ligne2]

Affichage en continu sur l'écran du bloc de lignes compris entre ligne1 et ligne2.

La touche <SPACE> sera utile dans cette commande pour geler momentanément l'affichage.

Si ligne1 > ligne2, le message "Paramètres incorrects" est affiché.

S'il n'y a pas de texte en mémoire le message "Pas de texte" est affiché.

Si ligne2 est omise et si ligne1 n'existe pas, l'éditeur affiche la ligne suivante.

Si ligne1 est omise, la ligne courante est prise par défaut.

Si ligne2 est omise, ligne1 est prise par défaut.

Si ligne1 et ligne2 sont omises, l'éditeur affiche la ligne courante et les 23 lignes suivantes du texte.

Exemple : P.,$ — Affichage du texte à partir de la ligne courante et jusqu'à la fin

4.11. Replace

Syntaxe : R [ligne1] [,incrément]

Effacement de ligne1 si elle existe et passage en mode insertion à partir de cette ligne avec l'incrément spécifié.

La syntaxe est identique à la fonction Insert.

Exemple : R,1 Effacement de la ligne courante et passage en mode insertion avec un incrément de un

4.12. Save

Syntaxe : S [nom]

Sauvegarde sur cassette du texte source en mémoire sous le nom spécifié.

L'éditeur affiche le message "Préparer la cassette". Préparez donc la cassette en n'oubliant pas de débrancher la prise EAR. Appuyez sur <ENTER> une fois que cela sera terminé.

Durant la sauvegarde le bord de l'écran clignote.

Il est possible d'interrompre la sauvegarde en appuyant sur <BREAK>.

Exemple : Stest Sauvegarde du programme test

4.13. User

Syntaxe : 0

Affichage de la taille occupée par le texte en mémoire et de la taille de la zone mémoire libre, sous la forme :

xxxxx Octets utilisés (Nombre d'octets libres)
yyyyy Octets libres (Nombre d'octets occupés)

5. L'ASSEMBLEUR

5.1. Format d'une ligne assembleur

Une ligne assembleur doit se présenter sous la forme suivante, où les termes entre crochets sont optionnels.

[étiquette] [TAB code-op[TAB opérandes]] [TAB] [;commentaires]

Elle peut être écrite en minuscules ou en majuscules. Toutes les lettres seront transformées en majuscules par l'éditeur, à l'exception des commentaires et des caractères situés entre apostrophes.

5.2. Etiquette symbolique

Le terme "étiquette" placé au début de la ligne assembleur représente une étiquette symbolique contenant au plus six caractères.

Le premier caractère ne peut être qu'une lettre ou un symbole dont le code ASCII est supérieur à 63. De plus, les chiffres sont acceptés dans les cinq autres caractères.

Une étiquette est utilisée pour représenter l'adresse d'implantation de la ligne assembleur dans le programme objet ou pour représenter une constante fixe (voir EQU).

Si une étiquette est incorrecte, l'assembleur affichera le message "Etiquette incorrecte".

Exemple d'étiquettes : LOOP1 LA_BAS ©READ

5.3. Tabulation

Le terme "TAB" désigne un nombre quelconque mais non nul d'espaces ou de caractères de tabulation (obtenus par <SYMBOL SHIFT> + <CAPS SHIFT>).

Pour faciliter la lisibilité des programmes en langage d'assemblage, il est conseillé d'utiliser pour TAB un seul caractère de tabulation qui n'occupe qu'un octet en mémoire.

5.4. Code opératoire

Le terme "code-op" de la ligne assembleur désigne le code d'une instruction Z80 ou le code d'une directive d'assemblage (voir ce paragraphe).

L'assembleur reconnaît tous les codes du Z80 définis par ZILOG.

Si l'utilisateur emploie un autre code, l'assembleur affichera le message d'erreur "Code illégal".

Exemples : LD ADD LDIR ORG

5.5. Opérandes

Le terme "opérandes" décrit le mode d'adressage utilisé et les registres employés selon la syntaxe définie par ZILOG.

Si le mode d'adressage est interdit, l'assembleur affichera le message Mode d'adressage illégal".

Exemple : A(HL) HL,DE A,6

5.6. Commentaires

Les commentaires se placent à la fin de la ligne assembleur et sont précédés du symbole ″;″.

Exemples : 0010 ; Ceci est un commentaire
0020 RRA ;Autre commentaire

5.7. Expressions arithmétiques

Dans tous les opérandes numériques, il est possible d'utiliser une expression arithmétique.

Les quatre opérateurs arithmétiques que reconnaît l'assembleur sont les suivants :

+ addition
— soustraction
* multiplication
/ division

Une expression arithmétique contenant ces opérateurs est évaluée en tenant compte de la priorité de opérateurs * et / par rapport à + et —.

Ainsi 5 + 3*6 est évalué comme 5 + (3*6)

Les opérandes situées entre les opérateurs peuvent être :

— Le symbole $ qui indique le compteur ordinal

— Une étiquette symbolique

— Une constante de 8 ou 16 bits sous forme ASCII entre apostrophes (Ex :'Yt' vaut 5974H, 'k' vaut 6BH)

— Une valeur numérique en décimal, en octal ou en hexadécimal.

Pour indiquer qu'un nombre est en octal ou en hexadécimal, il faut faire suivre ce nombre respectivement de la lettre ″O″ (pour octal) ou ″H″ (pour hexadécimal). Si aucune lettre ne suit le nombre ou si c'est la lettre ″D″, il sera considéré comme un nombre décimal.

Les nombres hexadécimaux qui commencent par une lettre (A, B, C, D, E, F) doivent être précédés d'un zéro afin que l'assembleur les distingue d'une étiquette symbolique.

Si une expression est incorrecte, l'assembleur affichera le message ″Expression illégale″ et s'il y a débordement des calculs il affichera le message ″Débordement″.

Exemples d'expressions : 45H + 3*410/5 — 64D vaut 24
— 34 + 'hY'*2 + 5 vaut 53397

5.8. Directives d'assemblage

Les directives d'assemblage sont des pseudo opérateurs utilisés pour agir sur le déroulement de l'assemblage ou pour générer des octets quelconques.

Voici la liste détaillée de toutes les directives d'assemblage reconnues par le ZX SPECTRUM EDITEUR/ASSEMBLEUR:

DEFB ou DB

[étiquette]TAB DEFB TAB expression1[,expression2[,...]]
DB

Génère un octet de valeur expression pour chacune des expressions qui suivent cette directive.

Exemple : 0010 OCTGEN DEFB 3,56H,'g',O
Cette ligne génère les octets suivants en hexadécimal : 03 56 67 00

Remarque : Si une expression est vide, l'éditeur génère un octet zéro. Ainsi: DB 3,,4 génère 03 00 04

DEFM ou DM

[étiquette] TAB DEFM TAB 'suite de caractères ASCII'
DM

Génère un octet correspondant au code ASCII de chacun des caractères tapés. Cette directive sera donc utilisée pour définir des messages.

Pour inclure une apostrophe dans la chaîne de caractères il faut doubler cette apostrophe.

Exemple : 0010 MESSAG DEFM 'L' 'hiver'
Cette ligne génère : 4C 27 68 69 76 65 72

DEFS ou DS

[étiquette] TAB DEFS TAB expression
DS

Réserve un nombre d'octets égal à l'expression qui suit la directive. Cette directive sera donc utilisée pour définir des variables dans le programme.

Exemple : 0020 VAR1 DEFS 2
défini une variable de 2 octets

DEFW ou DW

[étiquette] TAB DEFW TAB expression1[,expression2[,...]]
DW

Cette directive est identique à DEFB, mais elle agit sur des mots de deux octets au lieu d'agir sur des octets.

Exemple : 0040 WORDS DEFW 45−4,,'gT' ,−1
Cette ligne génère : 2900 0000 5467 FFFF

END

[étiquette] TAB END TAB [expression]

Indique la fin du texte à assembler. L'expression représente l'adresse d'exécution du programme. Celle-ci sera affichée à la fin de l'assemblage par le message :
xxxxx est l'adresse de lancement

La valeur par défaut de l'expression est zéro.

Si aucune directive END n'a été rencontrée lors de l'assemblage, le message "Pas de END" sera affiché.

Exemple : 0100 END DEBUT + 35H

EQU

étiquette TAB EQU TAB expression

Donne la valeur expression à l'étiquette.

Il n'est pas possible de définir plus d'une fois une étiquette par la directive EQU.

Les références aux étiquettes définies par EQU ne peuvent apparaître qu'après la définition de ces étiquettes. Par contre les références aux autres étiquettes peuvent se faire n'importe quand. Le non respect de cette règle entraîne le message "Symbole indéfini".

Exemple : 1000 ICI EQU 7F00H
donne la valeur 7F00H à l'étiquette ICI

IF et ENDIF

[étiquette] TAB IF TAB expression
[étiquette] TAB ENDIF

Assemblage conditionnel

Si la valeur expression est nulle, toutes les instructions suivant la directive IF jusqu'à la directive ENDIF ne seront pas assemblées. Si l'expression est non nulle, cette directive est sans effet sur l'assemblage.

Il est possible d'utiliser jusqu'à 256 niveaux de directives IF imbriquées.

S'il y a plus de directives ENDIF que de directives IF, le message "ENDIF sans IF" sera affiché.

Exemple : 0100 IF 1—US
0300 ENDIF

LIST

[étiquette] TAB LIST TAB ON
[étiquette] TAB LIST TAB OFF

LIST OFF arrête le listing sur l'écran ou l'imprimante et LIST ON le fait reprendre.

ORG

[étiquette] TAB ORG TAB expression.

Défini l'implantation en mémoire des instructions qui suivent. Après l'exécution de cette directive, le code objet sera implanté à l'adresse donnée par expression.

Exemple : 1000 ORG 9000H
implantation du code en 9000H

5.9. Commande d'assemblage

La commande "A" de l'éditeur permet d'appeler l'assembleur.

Syntaxe : A [nom] [—option1 [—option2[—…]]]

Cette commande provoque l'assemblage du texte en mémoire. Les options d'assemblage sont les suivantes :

- — NL (No listing) Pas de listing
- — LP (Line Printer) Sortie du listing sur imprimante
- — WS (Write Symbol) Affichage de la table des symboles classée
- — WE (Wait on Error) Attente à chaque erreur
- — WO (Write Output) Sortie du binaire sur cassette en utilisant le nom spécifié
- — IM (Into Memory) Assemblage en mémoire

Les options WO, WS et IM sont incompatibles.

"nom" represente le nom du fichier sur cassette destine a recevoir le code objet. Si ce nom dépasse 10 caractères, le message "Nom trop long" est affiché.

5.10. Résultat de l'assemblage

L'assembleur sort sur l'écran ou sur l'imprimante (option LP) successivement chaque ligne du texte précédée de l'adresse d'implantation et du code généré. Si la ligne comporte des erreurs, les messages d'erreur seront affichés juste avant celle-ci. Dans l'option WE, l'assembleur attend que l'utilisateur frappe une touche chaque fois qu'une erreur est rencontrée.

A la fin de l'assemblage, le nombre d'erreurs est affiché et l'adresse d'exécution du programme. Si l'option IM a été spécifiée, l'assembleur charge le programme en mémoire à condition que l'adresse d'implantation se situe en haut de mémoire, après le texte de l'utilisateur. Si l'option WO a été spécifiée, l'assembleur affiche le message "Préparer la cassette". Il génère alors le code objet sur un fichier cassette qui pourra être lu par la commande LOAD "nom" CODE du BASIC. Si le programme est très gros, il se peut que l'assembleur soit obligé de fractionner le programme objet en plusieurs fichiers qui porteront le nom spécifié suivi d'un numéro.

6. MESSAGES D'ERREUR

6.1. Messages d'erreur émis par l'éditeur

Lorsque l'éditeur détecte une erreur, il affiche le message d'erreur correspondant sur l'écran et interrompt immédiatement la commande. Les messages d'erreur qu'il peut afficher sont les suivants :

MESSAGE	SIGNIFICATION
Commande illégale	Commande inconnue
Paramètres incorrects	La commande n'est pas utilisée avec des paramètres corrects
Ligne inexistante	Appel d'une ligne inexistante
Numéro de ligne trop grand	Numéro de ligne supérieur à 9999
Plus de place entre les lignes	Le numéro de la prochaine ligne à insérer est supérieur au numéro de la ligne suivante du texte
Pas assez de mémoire	Il n'y a plus de place en mémoire pour insérer du texte
Chaîne non trouvée	Chaîne de caractères non trouvée (commande Find)

Increment nul	L'incrément utilisé est nul
Pas de texte	Demande d'affichage d'une ligne alors qu'il n'y a pas de texte en mémoire
Type du fichier incorrect	Tentative de charger un fichier qui ne contient pas un programme source en assembleur
Erreur de chargement	Erreur de chargement
Nom trop long	Nom du fichier sur cassette dépassant 10 caractères

6.2. Messages d'erreur émis par l'assembleur

Lorsque l'assembleur détecte une erreur dans une ligne du texte, il affiche juste avant celle-ci le, ou les messages d'erreur sur l'imprimante (option LP) ou sur l'écran même si l'option NL a été demandée.

Suivant la gravité de l'erreur, il génère ou ne génère pas de code pour cette ligne. Dans les deux cas, l'assemblage n'est pas interrompu. Il ne s'arrêtera que lors de la rencontre de la directive END ou à la fin du texte si cette directive est absente. Seul l'erreur "Débordement table des symboles" arrête l'assemblage.

Erreurs mineures (code généré)

MESSAGE	SIGNIFICATION
Fin de ligne illégale	Des caractères illégaux apparaissent juste après les opérandes
Expression illégale	Expression arithmétique incorrecte
Débordement	Débordement dans une expression arithmétique
Débordement de champ	Opérande numérique supérieur à 255 ou à 127 respectivement dans un adressage d'octet ou dans un adressage relatif
Branchement trop éloigné	Branchement relatif hors des limites
Etiquette incorrecte	Etiquette incorrecte
Symbole indéfini	Référence à une étiquette non définie
Définition multiple	Définition multiple d'une étiquette
Symbole multiplement défini	Référence à une étiquette définie plusieurs fois
Pas de END	Absence de directive END

Erreurs majeures (code non généré)

MESSAGE	SIGNIFICATION
Code illégal	Code opératoire illégal
Mode d'adressage illégal	Mode d'adressage illégal
Adresse incorrecte	Adresse trop basse lors d'un assemblage en mémoire. Le code est affiché sur l'écran, mais n'est pas implanté en memoire
Trop de IF imbriqués	Utilisation de plus de 256 niveaux de IF imbriqués
ENDIF sans IF	Directive ENDIF non associée à une directive IF
Débordement table des symboles	Il n'y a pas assez de place en mémoire pour contenir la table des symboles. L'assemblage est interrompu.

7. EXEMPLE

```
5E61          0010 WBOOT  EQU  5E61H   ;Adresse de retour
9000          0020        ORG  9000H   ;Implantation du prog.
9000 210040   0030 DEBUT  LD   HL,4000H ;Début mémoire écran
9003 110140   0040        LD   DE,4001H ;Début mémoire écran+1
9006 01FF17   0050        LD   BC,17FFH ;Taille mémoire écran
9009 36AA     0060        LD   (HL),0AAH ;10101010 en binaire
900B EDB0     0070        LDIR           ;Remplir l'écran de AAH
900D 3E7F     0080 TEST   LD   A,7FH     ;
900F DBFE     0090        IN   A,(0FEH)  ;Lecture touche BREAK
9011 IF       0100        RRA            ;Test BREAK
9012 38F9     0110        JR   C,TEST    ;Si pas de BREAK
9014 C3B15E   0120        JP   WBOOT     ;Retour à l'assembleur
9000          0130        END  DEBUT     ;Fin
00000 Erreur(s)
36864 est l'adresse de lancement
```

Ce petit programme provoque un striage de l'écran en allumant un point sur deux de l'écran, puis il attend que l'utilisateur appuie sur <BREAK> pour revenir à l'éditeur/assembleur.