SOFT 4000

EASI – AMSCODE

INITIATION AU LANGAGE MACHINE Z 80

AMSOFT & J.DUBLANCHET

INTRODUCTION AU LANGAGE MACHINE ET ASSEMBLEUR

1- LANGAGE MACHINE

Un programme en langage machine est une succession d'instructions codées sur 1, 2, 3 ou 4 octets, que le microprocesseur comprend et execute directement. (L'interprèteur basic est un programme en langage machine). Un programme est lancé à partir d'une adresse précise (adresse début) et retourne au point de lancement a la première instruction "RET" (équivalente à "RETURN" en basic).

COMMENT ENTRER UN PROGRAMME MACHINE DANS LA MEMOIRE DE L'ORDINATEUR ?

Soit le programme suivant: 21,ØØ,9Ø, 11,ØØ,AØ,Ø1,FF,ØØ,ED,BØ,C9

Pour entrer ce programme à partir du basic, il faudra "POKER" chaque valeur à une adresse précise; si nous prenons l'adresse &6ØØØ, cela donnera:

POKE &6ØØØ,&21
POKE &6ØØ1,&ØØ
POKE &6ØØ2,&9Ø etc...

Rassurez-vous! ce logiciel est fait pour simplifier considérablement ce travail et vous évitera ces "Pokages" rébarbatifs.

GENERALITES

EASI - AMSCODE est un puissant logiciel d'aide à la programmation en langage machine. Il a été étudié pour faciliter la création et la mise au point de petits programmes, mais aussi pour permettre au débutant d'apprendre le langage assembleur et de comprendre plus facilement les instructions du Z 80. Dans ce sens, cette notice est faite en deux parties:

La première partie est une introduction au langage machine avec démonstration d'un petit programme; ce chapitre est destiné au débutant qui pourra, par l'exemple, comprendre le principe même d'un programme machine et apprendre à l'aide d'un livre (Nous conseillons fortement " Z 80 programmation en langage assembleur" de L.A. Leventhal chez les éditions radio), le langage assembleur. Quelques notions de notation hexadécimale et binaire sont demandées.

La deuxième partie est d'avantage un manuel de référence, destiné aux initiés. Ce chapitre explique chaque fonction du moniteur en détail, avec des exemples.

COMMENT APPELER UN PROGRAMME MACHINE A PARTIR DU BASIC ?

```
10 PRINT "Début de programme"
20 REM Appel d'un programme machine à partir de l'adresse &6000
30 CALL &6000
40 PRINT "Suite du programme"
```

2- L'ASSEMBLEUR

Afin de mieux visualiser, écrire et analyser un programme machine (Qui en lui-même, est hermétique) il existe un autre langage: l'assembleur. Notre suite de nombres hexadécimaux de l'exemple précédent deviendra:

Codes machines (dit codes objets)	Assembleur (dit prog. source)
21009	LD HL,#9000
1100A0	LD DE,#A000
01FF00	LD BC,#00FF
EDB0	LDIR
C9	RET

Ceci est déja un peu plus "parlant" !

3- DEMONSTRATION

Cette partie est destinée à montrer de façon claire et par un exemple précis, l'utilisation de ce logiciel.

Programme qui servira d'exemple:

```
# 6000   212060        LD HL, 6020
# 6003   7E            LD A,(HL)
# 6004   0608          LD B,8
# 6006   0E30          LD C,"0"
# 6008   23     CONV:  INC HL
# 6009   71            LD (HL),C
# 600A   17            RLA
# 600B   3001          JR NC,COMPT
# 600D   34            INC (HL)
# 600E   10F8   COMPT: DJNZ CONV
# 6010   C9            RET
```

Ce programme convertit une valeur 8 bits située à l'adresse &6020 en une chaine ASCII commençant à l'adresse &6021. (L'explication détaillée de ce programme dépassant le cadre de cette notice, nous conseillons au débutant de se référer au livre cité dans le chapitre précédent).

1- COMMENT ENTRER CE PROGRAMME DANS LA MEMOIRE DE L'ORDINATEUR ?

Choisir l'option "CHARGER" à partir du menu en frappant la touche "C". A la question "ADRESSE DEBUT? " entrons simplement 6000. Remarquez qu'il n'est pas necessaire de valider avec la touche "ENTER". A partir de 4 caractères, le programme validera automatiquement l'adresse demandée. Si vous avez fait une erreur avant le 4ème caractère, il vous est toujours possible d'effacer avec la touche "DEL". Si votre adresse fait moins de 4 caractères, alors il faudra la valider vous-même avec la touche "ENTER". Enfin, si vous changez d'avis, appuyez sur la touche "M" pour retourner au menu.

Maintenant, vous voyez apparaitre à l'écran:

#6ØØØ ØØ >

Il s'agit de l'adresse, du contenu de l'adresse puis du signe " > ", indiquant l'endroit où s'affichera la nouvelle valeur que vous allez entrer.

Pour entrer notre programme d'exemple, tapez chaque caractère ASCII des codes objets, soit: 2,1,2,Ø,6,Ø etc...

Quand tous les codes seront entrés, terminez par la touche "ENTER" pour revenir à "ATTENTE D'UN ORDRE".

Si vous avez entre une mauvaise valeur, il vous suffit de revenir à l'adresse précédente avec la touche "DEL", ou passer à l'adresse suivante en appuyant sur la barre d'espacement.

2- COMMENT VERIFIER SI LES CODES OBJETS CORRESPONDENT BIEN AU PROGRAMME SOURCE ?

- Choisir l'option "DESASSEMBLER" en frappant la touche "D"
- Adresse début ? 6ØØØ
- Adresse fin ? 6Ø2Ø (laissons de la marge)
- Imprimante ? (O/N) N
- Voici maintenant qu'apparait à l'écran la première ligne de notre programme avec l'adresse, les codes objets, et les mnémoniques assembleur. Pour voir les lignes suivantes, appuyez sur une touche quelconque.
- Si votre programme ne correspond pas, cherchez le ou les code(s) machines faux et corrigez-les avec l'option "CHARGER".

3- COMMENT EXECUTER LE PROGRAMME EN "VOYANT CE QUE CA FAIT" ?

- Choisir l'option "EXECUTION PAS A PAS" en frappant la touche "P".
- Adresse début ? 6ØØØ
- Pour suivre l'execution du programme, il suffit d'appuyer sur une touche quelconque, une fois pour executer l'instruction et une seconde fois pour passer à l'instruction suivante.
- Nous ne pouvons pas détailler içi l'effet de chaque instruction, mais le débutant comprendra l'intérêt d'une telle option qui permet de "voir" à l'intérieur du microprocesseur et par conséquent de comprendre beaucoup plus rapidement l'effet d'une instruction.
- La page "ATTENTE D'UN ORDRE" revindra automatiquement quand le programme executera l'instruction "RET".

4- COMMENT VOIR LE RESULTAT ?

- Rappellons que le programme convertit une donnée 8 bits située à l'adresse &6Ø2Ø, en une chaine ASCII commençant à l'adresse &6Ø21. Regardons donc à l'adresse &6Ø21:

- Choisir l'option "DUMP ASCII" en frappant la touche "A"
- Adresse début? 6Ø21
- Adresse fin? 6Ø3Ø
- Imprimante (O/N) N
- Si vous voyez apparaître à l'écran : #6Ø21 ØØØØØØØØ....., c'est que votre programme "tourne"! En effet nous n'avons pas modifié le contenu de l'adresse &6Ø21, et tout le monde sait que &00 hexa. font &XØØØØØØØØ en binaire. Vous pouvez faire d'autres essais avec d'autres valeurs; commencez d'abord par modifier l'octet &6Ø2Ø avec l'option charger, puis exécuter le programme de la même manière que précédemment ou de façon normale par l'option "EXECUTION NORMALE".

5- COMMENT SAVOIR SI LE RESULTAT EST JUSTE?

- Choisir l'option "CONVERSION" en frappant la touche "K"
- Donnée (Mettre la valeur de l'adresse &6Ø2Ø).

Attention les données doivent être entrées avec la même syntaxe que pour le basic :

- &ØØØØ ;Pour une donnée hexadécimale
- ØØØØØ ;Pour une donnée décimale
- &xØØØØØØØØ ;Pour une donnée binaire
- Validez avec la touche "ENTER"
- Maintenant, comparez la valeur binaire avec la chaîne ASCII commençant à l'adresse &6Ø21.

NOTICE D'UTILISATION

1- GENERALITES

EASI - AMSCODE occupe toute la mémoire jusqu'à l'adresse &4FFF. L'utilisateur dispose donc de l'espace &5ØØØ > &AB7F, soit plus de 22ko. Il permet en outre d'accéder à la ROM comme à la RAM avec les options "desassembler, dump ascii, dump hexa. exécution normale et pas à pas.
Toutes les adresses doivent être entrées en hexadécimal. Entre 1 et 3 caractères, il sera toujours possible de corriger avec la touche "DEL" ou de valider avec la touche "ENTER". Au 4e caractère l'adresse sera automatiquement validée.

2- D = DESASSEMBLER

Desassemble une partie de la mémoire entre les adresses début et fin. "F" provoque la fin et

l'attente d'une autre commande. N'importe quelle autre touche provoque la suite de l'affichage à l'écran.

3- C = CHARGER

Modifie la mémoire à partir de l'adresse début avec les données entrées en hexadécimal. "ENTER" provoque la fin et l'attente d'une autre commande, "SPACE" passe à l'adresse suivante et "DEL" retourne à l'adresse précédente.

4- P = EXECUTION PAS A PAS

Exécute instruction pas instruction, une partie de la mémoire à partir de l'adresse début jusqu'à la première instruction "RET". Ceci se passe en deux parties: premièrement l'exécution de l'instruction affichée en haut de l'écran avec affichage des registres, deuxièmement, le désassemblage de l'instruction suivante. Pour passer de l'une à l'autre, il suffit de presser n'importe quelle touche, sauf "F", qui elle, provoquera l'arrêt et l'attente d'une autre commande.
Attention à certaines instructions comme "EXX", "OUT", ainsi qu'à certaines manipulations de pile

qui, mal employées, risqueraient de "planter" le programme.

5- E = EXECUTION NORMALE

Cette option commence par charger les registres de leur contenu sauvegarde lors de la précédente exécution (normale ou pas à pas), puis exécute une partie de la mémoire de l'adresse début jusqu'à la première instruction "RET". Ensuite elle sauvegarde les nouvelles valeurs des registres, qui pourront être visualisées par l'instruction "VISUALISATION DES REGISTRES".

6- A = DUMP ASCII

Montre la mémoire sous forme ASCII, à partir de l'adresse début jusqu'à l'adresse fin.

7- H = DUMP HEXADECIMAL

Même chose que précédemment, mais sous forme hexadécimale.

8- L = CHARGEMENT D'UN PROGRAMME

Charge en mémoire un programme binaire à partir de la cassette. L'adresse début est obligatoire et

doit se situer après &4FFF, pour ne pas "écraser" Z 8Ø.

9- S = SAUVEGARDE D'UN PROGRAMME

Enregistre sur cassette une partie de la mémoire située entre les adresses début et fin (obligatoirement au-dessus de &4FFF). L'adresse entrée doit se situer entre début et fin.

10- O = CHANGEMENT DES COULEURS

Modifie les couleurs du bord, du papier, de l'encre et des bandes situées en haut et en bas de l'écran. Suivre les instructions affichées à l'écran.

11- T = TRANSLATION

Duplique une partie de la mémoire située entre les adresses début et fin, à l'adresse implantation. L'option "Avec modification" corrigera toutes les adresses situées dans le bloc à dupliquer.

Exemple :

Imaginons ce programme commençant à l'adresse &8ØØØ :

```
#8ØØØ 21Ø58Ø    LD HL, #8ØØ5
#8ØØ3 11ØØAØ    LD DE, #AØØØ
```

```
#8ØØ6 Ø1FFØØ   LD BC, #ØØFF
#8ØØ7 C9        RET
#8ØØ8 CDØØ8Ø    CALL #8ØØØ
#8ØØB 18Ø1      JR #8ØØE
#8ØØD AF        XOR A
#8ØØE C3Ø88Ø    JP #8ØØ8
```

En le duplicant de l'adresse &9ØØØ, il deviendra :

```
#9ØØØ 21Ø59Ø    LD HL, #9ØØ5
#9ØØ3 11ØØAØ    LD DE, #AØØØ
#9ØØ6 Ø1FFØØ    LD BC, #ØØFF
#9ØØ7 C9        RET
#9ØØ8 CDØØ9Ø    CAL #9ØØØ
#9ØØB 18Ø1      JR #9ØØE
#9ØØD AF        XOR A
#9ØØE C3Ø89Ø    JP #9ØØ8
```

Remarquez que toutes les adresses situées entre &8ØØØ et &8Ø1Ø sont incrémentées de &1000, aussi bien les adresses de saut et d'appel que les adresses directes.

L'option "Sans modification" dupliquera le bloc, simplement.

12- K = CONVERSION

Convertit une donnée 16 bits. Les données doivent être entrées selon la même syntaxe que le basic:

Décimale sans préfixe, (de Ø à 65535)
Hexadécimale: préf. "&". (de &Ø à &FFFF)
Binaire: préf. "&X"
(de &XØ à &X1111111111111111)

13- V = VISUALISATION DES REGISTRES

Montre les registres de la même manière qu'avec l'option "EXECUTION PAS A PAS". Cette option sert à connaître l'effet d'un programme exécuté normalement.

14- R = RECHERCHE D'OCTETS

Permet de rechercher dans la mémoire (RAM ou ROM), un bloc d'octets (entre 2 et 255), situé entre les adresses début et fin. Entrez les octets de la même manière qu'avec l'option "CHARGER"