raconte-moi...

LA MICRO-INFORMATIQUE

Ce programme nous a été proposé par M. Rangelian, du Kremlin-Bicêtre. L'auteur de cet article a toutefois apporté quelques modifications, pour en augmenter encore plus l'intérêt auprès des lecteurs, et surtout faire ressortir quelques aspects nouveaux.

C'est tout d'abord un programme «utilitaire». Au cours de la phase de mise au point d'un programme, quelques commentaires au début d'un listing permettent de le personnaliser et d'identifier aisément les différentes parties. De même, et à plus forte raison, avant l'édition d'un programme, si celui-ci comporte un titre suivi d'un court résumé de ses fonctions, il sera plus facilement utilisable ultérieurement : ainsi vous constituerez vous votre propre bibliothèque de programmes.

Le second aspect est essentiellement pédagogique. Nous suggérons quelques améliorations à la fin de cet article pour en augmenter les performances, mais c'est surtout parce que nous présentons un concept pratique, pas assez usité, et cependant très utile «Les programmes translatables»

PROGRAMMES TRANSLATABLES

Dans la plupart des cas, les programmes que nous écrivons ne sont pas «translatables».

A l'aide d'un exemple, nous allons présenter cet aspect, et en indiquer les avantages.

Considérons le programme (Fig.1) qui consiste à faire apparaître le message «Hello» sur les afficheurs. Il débute à l'adresse 1800 H et se termine à 180 CH. Introduit ainsi à partir de l'adresse 1800 H dans la RAM, il fonctionne.

Débranchez votre MPF-1 pour effacer le contenu de la RAM, et introduisez les mêmes codes hexadécimaux, non plus à partir de 1800 H, mais à partir d'une autre adresse, par exemple 1900 H. Ceci étant fait, relancez l'exécution en partant de 1900 H. Surprise ! la visualisation fait apparaître des caractères pour le moins étranges, qui ne ressemblent en rien au message initial.

Le programme proposé transforme votre MPF-1 en « journal lumineux » et, si vous disposez de l'option imprimante, en « machine à écrire », le texte est introduit au moyen du clavier, transformé pour la circonstance en clavier alphabétique, en utilisant la face verso du cache MPF-Basic.

A l'aide de la touche «MOVE», ramenez le programme à partir de l'adresse 1800 H (MOVE, 1900 ; début ; 190 C ; fin ; 1800 ; destination). Exécutez le programme en partant de 1800 H,le message «Hello» réapparaît.

Expliquons. Le programme comporte une table de données (cases mémoires 1807 H à 180 CH) contenant les codes du message à afficher. Pour accéder aux données du tableau nous avons utilisé l'adressage immédiat étendu en chargeant le registre index IX avec la première adresse du tableau, qui est 1807 H (première instruction DD 21 07 18, ou charger IX avec 1807 H).

En prenant comme origine l'adresse 1900 H, la table débutait à l'adresse 1907 H, tandis que le registre IX était toujours chargé avec 1807 H. Lors de l'exécution, IX pointe donc sur le contenu de la case 1807 H et non 1907 H, de sorte que les données trouvées sont aléatoires.

Conclusion, le programme n°1 n'est pas «translatable». Pour le faire fonctionner à partir d'une adresse d'origine différente de 1800 H, il faut au préalable recalculer la plupart des adresses exprimées en valeur absolue.

Introduisez le programme n°2 (Fig.2) à partir d'une adresse quelconque de la zone mémoire vive (1800 H à 1FFF) sans toutefois chevaucher la zone réservée à la pile. Essayez plusieurs adresses d'origine (en débranchant à chaque fois le MPF-1 pour effacer la mémoire vive), et vous constaterez que le programme s'exécute à partir de n'importe quelle adresse.

Ayant introduit le programme n°2 et noté l'adresse d'origine, placez un point d'arrêt (SBR) à l'adresse de départ «+6». Lancez le programme qui s'exécute jusqu'au chargement du registre DE. Examinez son contenu : vous devez y trouver l'adresse de l'«origine + 3». C'est cette valeur que nous utiliserons comme «référence» dans la suite de notre programme.

```
        PROGRAMME

CODES HEXADECIMAUX

1800 DD 21 07 18
1804 CD FE 05 00
1808 BD 85 85 8F
180C 37


CODES OBJET

1800 DD LD IX,1807
1804 CD CALL 05FE


 MESSAGE

1807 00 BD 85 85
180B 8F 37
```

Fig. 1

Le chargement du registre DE avec le contenu de PC ne peut pas s'effectuer immédiatement. En effet, bien que le jeu d'instructions du Z-80 soit relativement abondant, l'instruction Ld dd PC n'existe pas. Pour contourner la difficulté, nous provoquons une instruction de saut (CALL) qui entraîne le transfert du contenu de PC (adresse de l'instruction suivante) dans la pile. Le saut est effectué à l'adresse 00D3 H (dans le Moniteur) dont le contenu est C9 (RETour). Pendant cet «aller-retour» le pointeur de pile SP (Fig.3) est successivement décrémenté, puis incrémenté de 2. Donc apparemment aucun changement, si ce n'est que les cases mémoires d'adresses SP-1 et SP-2 contiennent les deux octets du compteur ordinal PC. La figure 3 explique le fonctionnement.

Avant de charger le registre DE à partir du contenu de la pile (instruction de dépillement), il faut diminuer le compteur SP de 2, après quoi l'adresse «Origine + 3» (adresse de l'instruction qui suit CALL) se trouve stockée dans le registre DE, et celle-ci constitue notre référence. Dans la suite du programme, chaque fois qu'il sera nécessaire de se rendre à une adresse donnée, début d'une table, par exemple, il faudra non plus utiliser **son adresse absolue mais son adresse relative par rapport à la référence,** c'est ce qu'on désigne par le terme «déplacement».

Ainsi, si l'adresse de référence est XX03, l'adresse «début de table» XX0F, la valeur du déplacement d est :

d = XX0F — XX03 = 000C

La valeur du déplacement d est chargée dans le registre IX (Ld IX, 0C : DD 21 0C 00), et l'addition des contenus de IX et de DE (ADD IX, DE) permet de reconstituer l'adresse absolue du début de table.

Puis la suite du programme 2 est identique au premier.

INTERET D'UN PROGRAMME TRANSLATABLE

Un programme en cours de développement se mémorise dans la mémoire vive. Il est ainsi très facile d'y apporter des modifications, des insertions ou des suppressions. Ensuite, si l'utilisateur souhaite le conserver d'une manière permanente, il doit l'introduire dans une EPROM dont les adresses sont dans la plupart des cas différentes de celles de la mémoire vive (les emplacements RAM et EPROM ne sont pas les mêmes). Avant de mémoriser un

```
        PROGRAMME
      TRANSLATABLE

CODES HEXADECIMAUX

1800 CD 3D 00 3B
1804 3B D1 DD 21
1808 0C 00 DD 19
180C CD FE 05 00
1810 BD 85 85 8F
1814 37


CODES OBJET

1800 CD CALL 003D
1803 3B DEC SP
1804 3B DEC SP
1805 D1 POP DE
1806 DD LD IX,000C
180A DD ADD IX,DE
180C CD CALL 05FE


  MESSAGE

180F 00 BD 85 85
1813 8F 37
```

Fig. 2

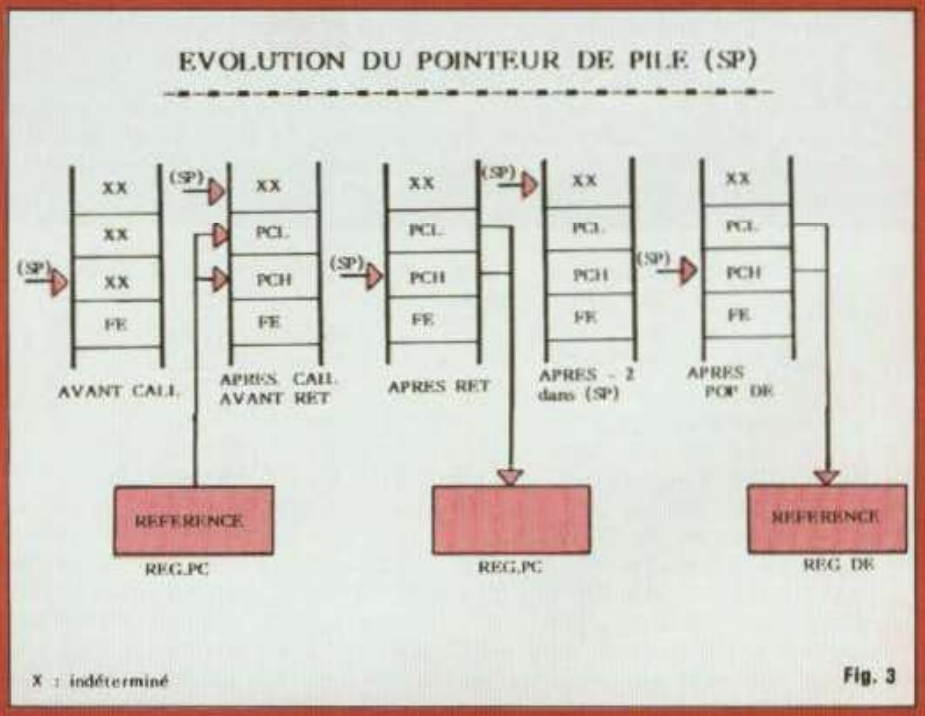

Fig. 3

raconte-moi...

LA MICRO-INFORMATIQUE

programme, il faut effectuer un transcodage pour les adresses absolues, avec le risque de ne constater une erreur ou un oubli que lorsque l'EPROM sera écrite.

Avec un programme translatable, plus aucun souci de calcul d'adresses : le programme peut être chargé **sans la moindre modification** dans l'EPROM. De plus, celle-ci peut être placée sur n'importe quel support disponible, comme nous l'avons fait pour le programme «machine à écrire» qui suit. En effet, celui-ci a été stocké dans une 2716, et l'EPROM placée sur le support U7 à côté de l'unité centrale. Nous avons aussi chargé une EPROM 2732 avec les mêmes données, et placé cette dernière sur la carte imprimante PRT/MPF. Cette deuxième solution est probablement la meilleure compte tenu de la finalité du programme.

Lorsque les sous-programmes sont translatables, chacun d'eux peut être testé indépendamment des autres puis ils peuvent être «assemblés» l'un après l'autre pour constituer l'application complète sans avoir à reprendre les adresses. Il n'est pas utile de «réserver» des zones libres entre les différentes parties pour d'éventuels débordements. En effet, en pareil cas, il suffit de faire «glisser» (fonction MOVE) le programme pour y insérer les modifications nécessaires.

Ainsi, utilisés conjointement avec la fonction MOVE, les programmes translatables sont plus aisés à mettre au point.

EXEMPLE

Nous avons appliqué ce concept au programme «machine à écrire et journal lumineux» dont l'édition commentée est présentée figure 4 (version mnémonique Z-80). La figure 5 présente l'édition en codes hexadécimaux que nous avons fait volontairement débuter à partir de l'adresse 0000. En réalité, le lecteur peut l'introduire à partir de n'importe quelle adresse dans la zone de mémoire vive.

Ayant lancé l'exécution du programme, la nouvelle affectation des touches est inscrite sur la face verso du cache (cache Basic du MPF-1B) comme le montre la figure 6. L'emploi de lettres de transfert (type Letraset) confère une bonne présentation.

En dehors des touches alphabétiques, la fonction des autres touches est la suivante :

IMP : Imprime le texte précédemment introduit. Le caractère fin de ligne est détecté automatiquement et provoque un retour à la ligne suivante.

FIN : Fin du texte (8. Code FF) qui n'apparaît que pendant l'introduction du texte.

DEF : Provoque le défilement de la droite vers la gauche. La vitesse se détermine par le contenu du registre B.

EFF : Touche d'effacement. L'action peut être répétée.

ESP : Espace

+1 : Défilement du texte caractère par caractère.

SUGGESTIONS

Nous ne disposons que des caractères alphabétiques. Pour introduire des chiffres et des symboles, il faut pouvoir affecter un deuxième caractère aux touches, qui se distingue du premier par la touche «USER KEY» (fonction SHIFT).

Ainsi conçue, la longueur des messages est de trois lignes (environ 60

```
     PROGRAMME
  MACHINE A ECRIRE
        ET
  JOURNAL LUMINEUX

  INITIALISATION

0000 21 LD HL,1E00
0003 E5 PUSH HL
0004 AF XOR A
0005 06 LD B,06
0007 77 LD (HL),A
0008 23 INC HL
0009 10 DJNZ 0007
000B DD POP IX
000D 3B DEC SP
000E 3B DEC SP
   ORIG DANS PG DE

000F CD CALL 003D
0012 3B DEC SP
0013 3B DEC SP
0014 D1 POP DE

   IDENTIFICATION
   DE LA FONCTION

0015 CD CALL 05FE
0018 FE CP 1C
001A 28 JR Z,0067
001C FE CP 1D
001E 20 JR NZ,0024
0020 3E LD A,FF
0022 18 JR 0048
0024 FE CP 1E
0026 28 JR Z,004F
0028 FE CP 1F
002A 20 JR NZ,0030
002C DD INC IX
002E 18 JR 0015
0030 FE CP 12
0032 20 JR NZ,0038
0034 DD DEC IX
0036 18 JR 0015
 TRANSCODAGE POUR
 AFFICHEURS
0038 21 LD HL,00D8
003B 19 ADD HL,DE
003C E5 PUSH HL
003D C1 POP BC
003E 21 LD HL,00BD
0041 19 ADD HL,DE
0042 ED CPD
0044 20 JR NZ,0042
0046 03 INC BC
0047 0A LD A,(BC)
0048 DD LD (IX+FF),A
004B DD DEC IX
004D 18 JR 0015
```

Fig. 4

caractères). Celle-ci pourrait être étendue à 100 lignes en modifiant la mémorisation du texte.
Une touche disponible peut faire office de retour de fin de ligne pour faciliter l'édition d'un texte.
Voici quelques suggestions. Vous en trouverez certainement d'autres ; n'hésitez pas à nous les communiquer pour que nous puissions en faire profiter tous nos lecteurs.

Philippe Duquesne

Fig. 6 : Clavier « alphabétique ».

```
      PROGRAMME
  MACHINE A ECRIRE
       ET
  JOURNAL LUMINEUX
CODES HEXADECIMAUX
0000 21 00 1E E5
0004 AF 06 06 77
0008 23 10 FC DD
000C E1 3B 3B CD
0010 3D 00 3B 3B
0014 D1 CD FE 05
0018 FE 1C 28 4B
001C FE 1D 20 04
0020 3E FF 18 24
0024 FE 1E 28 27
0028 FE 1F 20 04
002C LD 23 18 E5
0030 FE 12 20 04
0034 DD 2B 18 DD
0038 21 D8 00 19
003C E5 C1 21 BD
0040 00 19 ED A9
0044 20 FC 03 0A
0048 DD 77 FF DD
004C 2B 18 C6 DD
0050 E1 3B 3B DD
0054 2B 06 25 CD
0058 24 06 10 FB
005C DD 7E FF FE
0060 FF 20 F0 18
0064 EA F0 0F FD
0068 E1 3B 3B DD
006C E1 3B 3B 06
0070 13 0E 5B 21
0074 D8 00 19 DD
0078 7E FF FE FF
007C 28 1E FE 00
0080 20 04 3E 20
0084 18 05 ED A9
0088 20 FC 79 FD
008C 77 80 DD 2B
0090 FD 23 10 DD
0094 FD 36 80 0A
0098 FD 23 18 D3
009C FD 36 80 0D
00A0 DD E1 01 80
00A4 FF DD 09 CD
00A8 AC 65 21 EE
00AC FF 19 E5 C9
00B0 8F 0F F0 F0
00B4 0F 16 15 18
00B8 0C 0D 0E 0F
00BC 17 1A 1B 08
00C0 09 0A 0B 13
00C4 11 14 04 05
00C8 06 07 10 19
00CC 00 01 02 03
00D0 00 3F A7 8D
00D4 B3 8F 0F AD
00D8 37 89 B1 97
00DC 85 2B 23 A3
00E0 1F 3E 03 A6
00E4 87 B5 B7 A9
00E8 07 B6 8A
 DEFILEMENT
LE REG B DETERMINE
LA VITESSE
004F DD POP IX
0051 3B DEC SP
0052 3B DEC SP
0053 DD DEC IX
0055 06 LD B,25
0057 CD CALL 0624
005A 10 DJNZ 0057
005C DD LD A,(IX+FF)
005F FE CP FF
0061 20 JR NZ,0053
0063 18 JR 004F
   FIN  PARTIE  I
 IMPRIMANTE
0067 FD POP IY
0069 3B DEC SP
006A 3B DEC SP
006B DD POP IX
006D 3B DEC SP
006E 3B DEC SP
006F 06 LD B,13
0071 0E LD C,5B
0073 21 LD HL,00D8
0076 19 ADD HL,DE
0077 DD LD A,(IX+FF)
007A FE CP FF
007C 28 JR Z,009C
007E FE CP 00
0080 20 JR NZ,0086
0082 3E LD A,20
0084 18 JR 008B
 TRANSCODAGE EN
 CODE ASCII
0086 ED CPD
0088 20 JR NZ,0086
008A 79 LD A,C
008B FD LD (IY+80),A
008E DD DEC IX
0090 FD INC IY
0092 10 DJNZ 0071
0094 FD LD (IY+80),0A
0098 FD INC IY
009A 18 JR 006F
009C FD LD (IY+80),0B
00A0 DD POP IX
00A2 01 LD BC,FF80
 IMPRESSION
00A5 DD ADD IX,BC
00A7 CD CALL 65AC
00AA 21 LD HL,FFEE
00AD 19 ADD HL,DE
00AE E5 PUSH HL
00AF C9 RET

 FIN DE PROGRAMME
      TABLES
 DE CONVERSION
00B5 16 15 18 0C
00B9 0D 0E 0F 17
00BD 1A 1B 08 09
00C1 0A 0B 13 11
00C5 14 04 05 06
00C9 07 10 19 00
00CD 01 02 03 00
00D1 3F A7 8D B3
00D5 8F 0F AD 37
00D9 89 B1 97 85
00DD 2B 23 A3 1F
00E1 3E 03 A6 87
00E5 B5 B7 A9 07
00E9 B6 8A
```

Fig. 5