

Canon

XP-140

MONITOR CARD

GUIDE TECHNIQUE

INTRODUCTION

Ce manuel procure l'information nécessaire pour utiliser le Canon X-07 afin de créer et d'exécuter des programmes en langage machine. Ce manuel est prévu pour être utilisé conjointement avec les autres publications suivantes.:

- “Mode d’emploi du X-07”
- “Programmation du X-07 pour les débutants”
- “Manuel de référence du X-07”
- “Mode d’emploi du XP-140”

Le présent manuel n’explique pas les procédures de programmation en langage machine. Pour de telles procédures, consulter l’un des nombreux manuels disponibles concernant le microprocesseur Z-80, avec son jeu d’instructions.

Table des Matières

INFORMATIONS GENERALES.....	3
1. APPEL DES PROGRAMMES EN LANGAGE MACHINE	
AU MOYEN DU BASIC.....	4
1.1 Réservation d'une zone dans la MEV pour les programmes en langage machine.....	4
1.2 Codage et introduction en mémoire des programmes en langage machine.....	5
2. LANCEMENT DES PROGRAMMES EN LANGAGE MACHINE INDEPENDAMMENT DU BASIC LORS DE L'ENCLENCHEMENT DE L'ALIMENTATION.....	8
2.1 Zone MEV utilisée par le programme en langage machine	8
(a) Zone MEV non disponible pour les programmes en langage machine.....	8
(b) Zone MEV d'unité centrale secondaire.....	8
(c) "Cacher" la zone de programme en langage machine du BASIC	9
2.1.1 Interface entre programmes en langage machine de la MEM et du BASIC	10
2.1.2 Les trois points d'introduction de programme en langage machine	11
3. POINTS A CONSIDERER LORS DU CODAGE DES PROGRAMMES EN LANGAGE MACHINE.....	12
3.1 Exceptions	12
3.1.1 Programmes en langage machine appelés depuis le BASIC.....	12
3.1.2 Programmes en langage machine appelés indépendamment du BASIC	13
(a) BREAK FLAG	13
(b) Contrôle de BREAK FLAG.....	13
(c) Traitement de secours	15
3.2 Pile d'introduction.....	17
3.3 Traitement de l'interruption de programme.....	18
(a) Vecteurs d'interruption	18
(b) Réécriture des programmes d'interruption.....	18
4. APPELS AU SYSTEME.....	20
4.1 Résumé des appels au système.....	20
4.1.1 Sous-programmes standards pour dispositifs et système	21
4.1.2 Sous-programmes de dispositifs	39
(a) Séquence d'initialisation de dispositif	39
(b) Utilisation des sous-programmes de commande de dispositif.....	41
(c) Exemple d'entrée/sortie	42
(d) Changement de l'adresse de branchement d'erreurs.....	47
(e) Augmentation du nombre de dispositif d'entrée/sortie	48

4.2	Commandes et fonctions d'unité centrale secondaire	54
4.2.1	Liste des commandes d'unité centrale secondaire	55
4.2.2	Explication des commandes d'unité centrale secondaire.....	58
4.2.3	Etat implicite de l'unité centrale secondaire	66
4.2.4	Secteur MEV d'unité centrale secondaire	67
(a)	MEV du système d'unité central secondaire (C00H à C00FFH).....	67
(b)	Adresses des touches définies par l'utilisateur (C800H à C9FFH)...	68
(c)	Adresses des caractères définis par l'utilisateur (CA00H à CBFFFH)	68
(d)	Adresse du registre tampon d'entrée de touche (C00H à CCFFH)...	69
(e)	Adresse du programme de lancement (C00H à CFFFH)	69
4.2.5	Matrice de clavier	70
5.	SEQUENCE D'ENCLENCHEMENT/DECLENCHEMENT D'ALIMENTATION DU X-07	71
5.1	Séquence d'enclenchement d'alimentation	71
5.2	Séquence de déclenchement d'alimentation	79
6.	EXEMPLE DE CODAGE DE PROGRAMME EN LANGAGE MACHINE	81
6.1	Traitement général des entrées	81
6.2	Exemples de traitement d'entrée de Monitor Card.....	84
6.3	Exemple d'expansion de dispositif au moyen de sous-programmes de contrôle de dispositif	88
7.	ANNEXES	89
7.1	Tableau d'adresse HOCK.....	89
7.2	Tableau d'adresses de symboles	90

INFORMATIONS GENERALES

Les programmes en langage machine peuvent être exécutés sur le X-07 au moyen de l'une des deux méthodes suivantes:

1. Appel des programmes en langage machine en tant que sous-programmes, à l'aide des instructions BASIC, USR ou EXEC.
C'est la méthode communément utilisée. Elle permet de réaliser facilement des procédures telles que le traitement d'erreurs, car les programmes en langage machine sont démarrés sous le contrôle du BASIC.
Le BASIC du X-07 procure un certain nombre de fonctions utiles pour travailler avec des programmes en langage machine. Elles comprennent la fonction de fichier MEV (permettant de sauvegarder des programmes en langage machine) et la fonction START qui lance automatiquement les programmes lors de l'enclenchement de l'alimentation.
2. Lancement des programmes en langage machine indépendamment du BASIC lors de l'enclenchement de l'alimentation. Avec cette méthode, les programmes en langage machine sont écrits dans la MEM (comme avec les cartes CANP) afin de séparer le X-07 du BASIC, ce qui permet de l'utiliser en tant que machine à application spéciale.

Il n'y a aucune restriction de programmation si l'utilisateur code lui-même entièrement sans avoir recours aux sous-programmes du BASIC; néanmoins, cela requiert une compréhension totale de la partie matérielle du X-07. Le présent manuel ne traite pas de l'utilisation directe du matériel; il part de l'hypothèse que toutes les commandes de la machines seront réalisées en appelant le système. Dans ce cas, l'accès à une partie de la zone de travail de la MEV est interdit.

1. APPEL DES PROGRAMMES EN LANGAGE MACHINE AU MOYEN DU BASIC

- (a) Réserver la zone MEV dans laquelle le programme en langage machine doit être conservé.
- (b) Coder et introduire le programme en langage machine dans la mémoire.

Après avoir réalisé ces procédures, appeler le programme en langage machine au moyen du BASIC.

1.1 Réservation d'une zone dans la MEV pour les programmes en langage machine

Utiliser l'instruction CLEAR pour réservier une zone dans la MEV destinée à l'utilisation du langage machine, et pour éviter que le contenu de la zone MEV ne soit modifié par l'interpréteur BASIC. (Pour plus de détails sur l'instruction CLEAR, se reporter au manuel de référence du X-07.

(Exemple 1) CLEAR 50, &HFFF


1-Adresse la plus élevée utilisable
par les programmes BASIC

Dans l'exemple 1, la zone débutant à l'adresse &H1000 (l'adresse suivant &HFFF) est réservée aux programmes en langage machine en positionnant l'adresse la plus élevée pouvant être utilisée avec le BASIC à &HFFF.

La zone réservée pour l'utilisation du langage machine par l'instruction CLEAR est réassignée au BASIC au moyen de l'instruction FSET, lors de l'enclenchement de l'alimentation. Par conséquent, prendre soin de réserver à nouveau la zone à l'aide de l'instruction CLEAR lors de l'enclenchement de l'alimentation ou après l'exécution de l'instruction FSET.

Si la taille de la zone MEV du X-07 est de 8K octets et que les instructions indiquées dans l'exemple 2 sont exécutées, l'espace mémoire est alloué de la manière indiquée à la figure 1.

(Exemple 2) FSET 2048
CLEAR 50, &HFFF

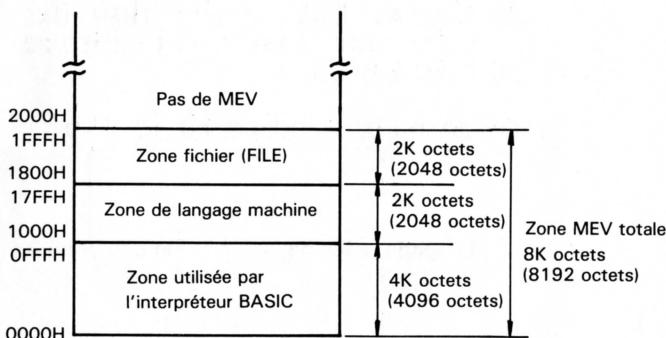


Figure 1

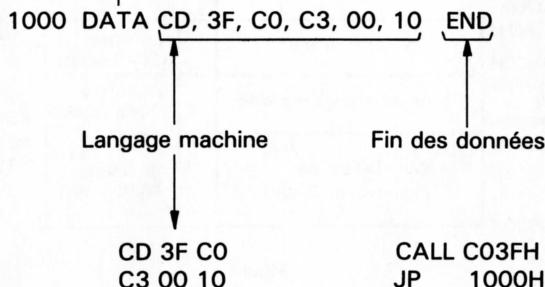
1.2 Codage et introduction en mémoire des programmes en langage machine

Les programmes en langage machine peuvent être introduits dans la zone de langage machine au moyen de l'une des trois méthodes suivantes:

1. Les codes d'instructions peuvent être inclus dans les instructions DATA du programme BASIC et écrits dans la zone de langage machine au moyen de l'instruction POKE. (Voir exemple 3).
2. Les codes d'instructions peuvent être écrits en tant que données dans la zone FILE, puis chargés dans la zone de langage machine. (Voir exemple 4).
3. Les programmes peuvent être introduits au moyen de la carte moniteur (Monitor Card). Pour plus de détails, se reporter au mode d'emploi de la carte moniteur (Monitor Card).

(Exemple 3) Utilisation des instructions DATA pour introduire les programmes en langage machine

```
10 CLEAR 50, &HFFF : I=&H1000
20 READ A$ : IF A$<>"END" THEN POKE
   I, VAL ("&H" + A$) : I=I+1 : GOTO 20
30 EXEC &H1000
```



Dans l'exemple 3, les données spécifiées par l'instruction DATA représentent un programme en langage machine.

La ligne 10 réserve la zone de langage machine et assigne &H1000 à I en tant que pointeur de zone de langage machine.

La ligne 20 mémorise les codes d'instructions de l'instruction DATA dans la variable AS et les introduit séquentiellement dans la zone débutant à l'adresse 1000. Les codes d'instructions sont lus et mémorisés jusqu'à ce que l'instruction END soit rencontrée.

La ligne 30 appelle l'adresse de départ (&H1000) du langage machine au moyen de l'instruction EXEC.

(Exemple 4) Mémorisation des programmes en langage machine en tant que données dans la zone FILE

```
(FSET 2000)
10 INIT #1, "MEV : MAC", 100
20 INPUT A$ : IF A$ = "END" THEN END
30 OUT #1, VAL ("&H" + A$) : GOTO 20
```

Tout d'abord, l'instruction FSET est exécutée pour spécifier la taille de la zone FILE requise pour la mémorisation des données.

La ligne 10 du programme réserve 100 octets de zone FILE pour créer un fichier MEV présentant le nom MAC. L'instruction INPUT de la ligne 20 introduit en mémoire les codes d'instruction du clavier et la ligne 30 les mémorise dans le fichier appelé MAC.

Par la suite, les codes d'instructions de MAC sont introduits dans la zone en langage machine au moyen du programme suivant.

```
10 CLEAR 50, &HFFF : INIT #1, "MEV : MAC"
20 FOR I=&H1000 TO &H1000+99
30 POKE I, INP (#1) : NEXT I
```

Les lignes 20 et 30 de ce programme introduisent 100 octets de MAC dans la zone de langage machine débutant à l'adresse &H1000.

2. LANCEMENT DES PROGRAMMES EN LANGAGE MACHINE INDEPENDAMMENT DU BASIC LORS DE L'ENCLENCHEMENT DE L'ALIMENTATION

Lorsqu'un programme en langage machine est exécuté indépendamment du BASIC, l'utilisateur est responsable de toutes les commandes d'alimentation, de gestion de mémoire et d'accès au dispositif d'entrée/sortie.

Les programmes en langage machine de ce type sont utilisés avec un logiciel qui est écrit dans la MEV de la carte mémoire ou dans la mémoire morte programmable (EPROM) de 8K octets installés dans l'orifice pour mémoire. L'utilisateur doit tout d'abord se familiariser avec le contenu de ce manuel avant de tenter de créer des programmes en langage machine de ce type.

2.1 Zone MEV utilisée par le programme en langage machine

(a) Zone MEV non disponible pour les programmes en langage machine

Les programmes en langage machine écrits par l'utilisateur ne peuvent pas être introduits dans la zone MEV (adresse 0000H-0551H) qui est utilisée pour réaliser des appels au système au moyen du BASIC.

La zone MEV pouvant être utilisée pour l'introduction de programmes en langage machine ou de données est celle débutant à l'adresse 0552H.

Voir "Exemples de traitement utilisant des appels au système" à la fin du présent manuel.

(b) Zone MEV d'unité centrale secondaire

La zone MEV d'unité centrale secondaire allant de C800H à C9FFH (dans l'espace d'adresse d'unité centrale secondaire) n'est pas utilisée par le système. Bien entendu, les programmes pour l'unité centrale principale ne peuvent pas être exécutés dans cette zone, mais elle peut être utilisée pour la mémorisation de données. Des données peuvent être introduites ou lues dans/depuis l'unité centrale secondaire en faisant appel aux sous-programmes de système RAMRD et RAMWRT, respectivement.

(c) **“Cacher” la zone de programme en langage machine du BASIC**

Lorsqu'un programme en langage machine est exécuté indépendamment du BASIC, le programme et les données relatives de la MEV sont normalement effacés si le BASIC est lancé ou si l'alimentation est coupée. Si cela présente un inconvénient, il existe une méthode permettant de “cacher” la zone de langage machine du BASIC afin de protéger son contenu. Cette méthode est en fait utilisée par les cartes CANP et autres.

Appels aux programmes en langage machine réalisés par ENTRY0, ENTRY1 et ENTRY2.

Lorsque l'alimentation est enclenchée, l'interpréteur BASIC recherche une zone FILE dans la MEV et si aucune n'est trouvée il établit l'adresse la plus élevée de la MEV en tant qu'adresse de départ de la zone FILE. Si une zone FILE a été réservée, l'interpréteur BASIC contrôle sa taille et alloue le reste de la MEV pour l'utilisation du BASIC. Par conséquent, une zone de langage machine peut être cachée du BASIC en assignant à ENTRY0, ENTRY1 et ENTRY2 des adresses supérieures à l'adresse de départ de la zone FILE.

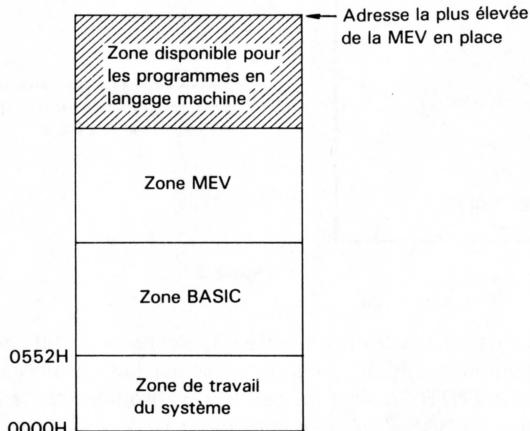


Figure 2

Un exemple de ce procédé est indiqué dans le listage du programme de carte moniteur (carte CANP) “Monitor Card” donné à la fin du présent manuel.

2.1.1 Interface entre programmes en langage machine de la MEM et le BASIC

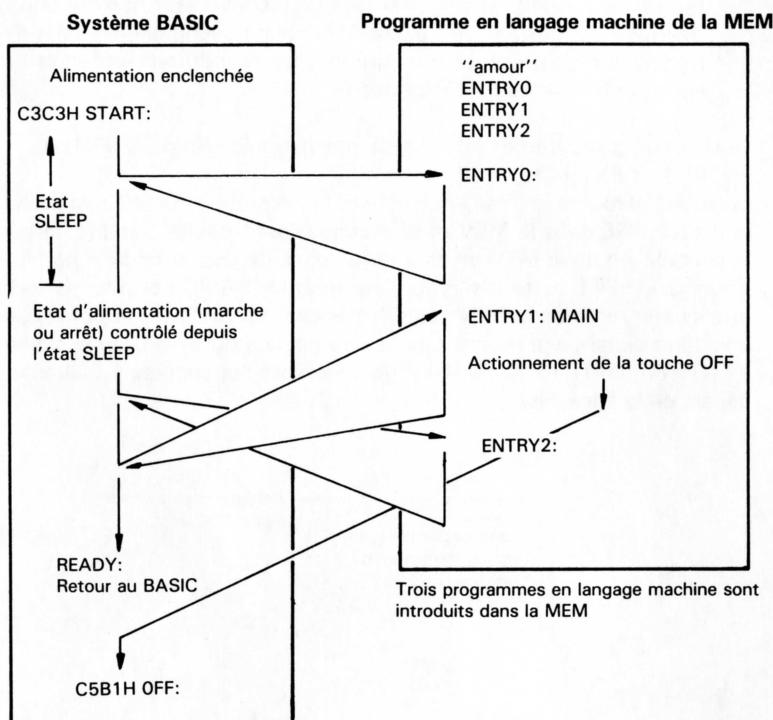


Figure 3

Lorsque l'alimentation est enclenchée, le système BASIC débute le traitement à l'adresse C3C3H, puis un appel est fait au programme en langage machine ENTRY0. A la fin de ce sous-programme, la commande est retournée au BASIC et l'alimentation est contrôlée depuis l'état SLEEP afin de déterminer si elle est enclenchée ou déclenchée. Si l'alimentation est enclenchée, le système appelle ENTRY2; la commande est retournée au BASIC lorsque le traitement de ENTRY2 a été terminé. Ensuite, un appel est réalisé à ENTRY1 (introduction principale de programme en langage machine). Lorsque le traitement du programme principal a été terminé, la commande retourne au BASIC, qui introduit l'état READY: et qui attend l'introduction de la commande BASIC.

Si la touche OFF est enfoncée pendant l'exécution de ENTRY1 (programme principal), la commande du traitement sera transférée à C5B1H OFF: du système BASIC.

2.1.2 Les trois points d'introduction de programme en langage machine

- ENTRY0:** Cette adresse d'introduction est appelée lorsque l'interpréteur BASIC a terminé l'initialisation de tous les dispositifs d'entrée/sortie et de la zone de travail minimum requise. Lorsque le X-07 est utilisé avec un matériel spécial, le sous-programme appelé doit initialiser ce matériel; autrement, une instruction RETURN est suffisante.
- ENTRY1:** Introduire le sous-programme en langage machine principal dans la zone appelée depuis ce point d'introduction. Si il est nécessaire de retourner la commande au BASIC depuis le programme en langage machine, l'exécution de l'instruction RETURN lancera le BASIC depuis le message "Copyright".
- ENTRY2:** C'est l'adresse d'introduction qui est appelée lorsque l'on quitte l'état SLEEP (état introduit lorsque la commande SLEEP est exécutée depuis le BASIC). Par conséquent, la commande doit être immédiatement retournée au BASIC au moyen d'une instruction RETURN.
Pour plus de détails, voir les explications de la séquence d'enclenchement d'alimentation du BASIC à la page 71 du présent manuel.

3. POINTS A CONSIDERER LORS DU CODAGE DES PROGRAMMES EN LANGAGE MACHINE

3.1 Exceptions

3.1.1 Programmes en langage machine appelés depuis le BASIC

- (a) Dans le X-07, les touches BREAK et OFF sont surveillées pendant l'exécution des instructions BASIC, mais non pendant l'exécution des programmes en langage machine. Par conséquent, la séquence CALL ABORT (code machine = CD, 3F, C00) doit être incluse dans le code machine si plus de 100 millisecondes doivent s'écouler depuis le moment où le programme en langage machine est appelé jusqu'au moment où la commande de temps est renvoyée au BASIC.

Le sous-programme ABORT surveille la touche OFF, la touche BREAK, l'état de la pile, etc. Si une interruption de programme est générée par l'un de ces facteurs, la commande est retournée au BASIC pour le traitement approprié. Si une boucle sans fin apparaît dans un programme en langage machine qui ne fait pas appel au sous-programme ABORT, les touches autres que le commutateur RESET seront ignorées.

- (b) Le sous-programme ABORT ne peut être utilisé que lorsque la commande est transférée au programme en langage machine depuis le BASIC au moyen des instructions EXEC ou USER. Une perte de commande apparaîtra si ce sous-programme est appelé depuis un programme en langage machine qui est exécuté indépendamment du BASIC, tel que dans les cartes CANP.
- (c) Il faut contrôler soigneusement que les programmes en langage machine ne contiennent pas d'erreurs, avant leur exécution. Une seule erreur suffit à provoquer une perte de commande, ce qui détruira tous les programmes et toutes les données présents dans la MEV. Si cela se produit, les touches autres que le commutateur RESET seront ignorées.

3.1.2 Programmes en langage machine appelés indépendamment du BASIC

(a) BREAK FLAG

Une zone d'information de 1 octet est prévue à l'adresse 002BH pour BREAK FLAG. L'information importante transférée à l'unité centrale principale lorsque des interruptions d'unité centrale secondaire apparaissent sera introduite dans cette zone par le programme d'interruption.

	D7	D6	D5	D4	D3	D2	D1	D0 (bit)
002BH	OFF	Arrêt de minuterie			Pile carte épuisée		Pile épuisée	BREAK

Figure 4

La figure 4 donne la signification de chaque bit de la zone d'octet de BREAK FLAG. Lors de l'écriture du programme en langage machine, l'utilisateur doit incorporer le traitement à réaliser lorsque ce bit (qui indique l'état de BREAK FLAG) est mis à 1 afin d'éviter de rendre invalide la touche OFF.

(b) Contrôle de BREAK FLAG

Lorsqu'un bit BREAK FLAG est contrôlé et qu'il se trouve à l'état 1, le traitement de secours correspondant doit être réalisé. Lorsque cela se produit dans le sous-programme d'entrée/sortie du système, le programme HOCK (ABORT HOCK = 00A8H) est appelé. L'utilisateur peut déterminer le traitement en cours de réalisation en présence d'un ABORT en plaçant un saut vers le sous-programme approprié dans ABORT HOCK.

(Référence 1)

L'indicateur d'interruption BREAK FLAG peut être contrôlé au moyen de l'appel au système TBREAK, mais ce programme ne contrôle que les bits de touche OFF, de batterie épuisée et de touche BREAK. En conséquence, il est préférable de contrôler tous les bits de BREAK FLAG se trouvant dans la boucle principale et d'utiliser l'appel TBREAK dans les sous-programmes.

LD A, (002BH)
 AND A
 JP NZ Programme BREAK FLAG

Contrôler BREAK FLAG avant de passer au programme principal

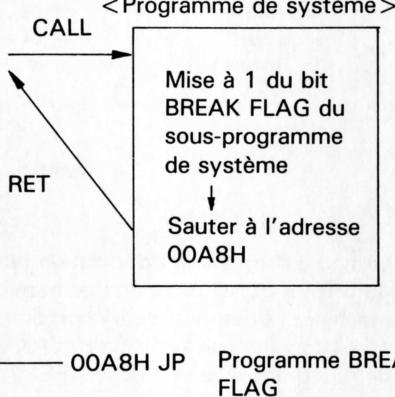
LD A, (002BH)
 AND A
 JP NZ Programme BREAK FLAG

Contrôler BREAK FLAG en chaque point de décision du programme

CALL Sous-Programme de système

Programme BREAK FLAG

Sous-programme de traitement de secours



(c) Traitement de secours

Les procédures de traitement de secours à utiliser en réponse aux différents bits BREAK FLAG sont indiquées ci-après. (Voir figure 4)

- Bit 0 = 1** indique que la touche BREAK a été enfoncée. Réalise le même traitement que la touche BREAK. Normalement, le traitement actuel est interrompu et la commande est renvoyée à la boucle principale.
- Bit 1 = 1** indique que la pile de l'unité principale du X-07 s'épuise. Il interrompt le traitement actuel et affiche un message indiquant l'état d'épuisement de la pile.
- Bit 3 = 1** indique que la pile de secours de la carte mémoire résidente s'épuise. Il interrompt le traitement actuel et affiche un message indiquant l'état d'épuisement de la pile dans la carte.
- Bit 6 = 1** indique qu'aucune des touches n'a été enfoncée pendant 15 minutes. Cela est normalement utilisé pour introduire automatiquement l'état hors service. En tenant compte des indications de la note 1, cela présentera le même fonctionnement que la touche OFF.
- Bit 7 = 1** indique que la touche OFF a été enfoncée. Avant d'introduire l'état hors service, réaliser le traitement nécessaire pour introduire l'état hors service dans un programme en langage machine et transférer la commande du traitement à l'appel de système OFF.

Note 1: Le bit 6 (spécifiant "arrêt de minuterie") sera mis à 1 si aucune des touches de l'unité centrale secondaire n'a été enfoncée pendant 15 minutes. Cela est utilisé pour déterminer si 15 minutes se sont écoulées depuis qu'une touche a été enfoncée. Par conséquent, même si ce bit a été réglé à 1 pendant l'exécution du programme de l'utilisateur, l'alimentation ne doit pas être déclenchée. Immédiatement avant que le programme de l'utilisateur n'entre dans l'état d'attente pour une touche, ce bit et la minuterie doivent être remis à zéro en utilisant l'appel au système PTMCLR, et il faut déterminer si l'état d'attente s'est poursuivi pendant 15 minutes avant qu'une touche n'ait été enfoncée. (PTMCLR = E327H)

Note 2: L'interrupteur marche-arrêt (situé à l'arrière de l'unité principale), utilisé lors du retrait ou de la mise en place d'une carte, est raccordé au même matériel que la touche OFF. Si l'état hors service ne peut pas être introduit dans les 200 millisecondes suivant le déclenchement de ce commutateur, il est possible qu'une perte de commande se produise si une carte a été introduite. Il est nécessaire de déterminer au moins chaque 100 millisecondes au moyen d'un programme si l'état hors service a été introduit.

3.2 Pile d'introduction

Lors de l'appel de chaque introduction, la pile est réglée quelque part entre D5H et 1D4H. Si l'information peut être couverte par environ 200 octets, la pile peut être utilisée telle quelle.

3.3 Traitement de l'interruption de programme

Le mode d'interruption ordinaire du X-07 est 0.

Lors de l'emploi de registres dans un programme, faire attention car le registre inverse est utilisé dans les programmes de traitement d'interruption.

(a) Vecteurs d'interruption

Le X-07 possède trois vecteurs d'interruption.

INTA (3CH).....Ce vecteur est utilisé pour les interruptions d'unité centrale secondaire, telles que pour les introductions au clavier.

INTB (34H)Ce vecteur est utilisé pour les interruptions séries et pour le compteur BEEP.

INTC (2CH)Ce vecteur n'est pas utilisé dans la configuration de base, mais il est utilisé pour l'interruption de minuterie (clignotement du curseur) de la MEM dans le poste d'extension.

Chaque vecteur contient des instructions de saut, au moyen desquelles la commande de traitement est transférée vers un sous-programme approprié. Par conséquent, la réécriture des instructions de saut pour chaque vecteur permet de transférer la commande du traitement vers une adresse optionnelle.

Dans ce cas, les appels au système ne peuvent pas réaliser leur traitement d'interruption d'origine. Par conséquent, après avoir terminé le traitement nécessaire, s'assurer que la commande du traitement est transférée à l'adresse d'origine, qui est rendu connue par les instructions de saut écrites dans le vecteur avant sa réécriture.

(b) Réécriture des programmes d'interruption

A titre d'exemple, la méthode d'interdiction de la touche BREAK est décrite ci-dessous.

Le vecteur d'interruption de l'unité centrale secondaire doit être réécrit pour interdire la touche BREAK. Les codes de langage machine (JP, C799H) ont été introduits dans le vecteur INTA.

003CH: C3 Code JP

003DH: 99] Adresse de départ du sous-programme d'interruption:
003EH: 07 C799H

Introduire tout d'abord dans la zone MEV les codes de langage machine (sous-programme d'interruption) indiqués dans la liste jointe en utilisant l'instruction POKE. Réécrire ensuite les adresses des programmes d'interruption qui ont été introduites aux adresses 003DH et 003EH, vers les adresses des sous-programmes d'interruption de l'utilisateur, en employant l'instruction POKE.

(Référence 2)

Si les codes de langage machine ont été écrits depuis 4000H, l'écriture de 00H et 40H à 003DH et 003EH, respectivement, sera convenable. Dans ce cas, la zone MEV dans laquelle les codes de langage machine seront introduits doit être protégée des programmes objets.

(Exemple 5)

LOCATION	OBJECT CODE	LINE	SOURCE LINE	
		1	"Z80" LIST XREF	
<C8C1>	2	INTEX	EQU 0C8C1H	
<C835>	3	ANSDA	EQU 0C835H	
<C80C>	4	NOTBRK	EQU 0C80CH	
<C8BD>	5	STROBE	EQU 0C8BDH	
<C884>	6	NOTO3	EQU 0C884H	
	7			
0000 D9	8	INTRPT :	EXX	
0001 08	9	EX	AF, AF	
0002 DBF2	10	IN	A,(0F2H) ;Vérifier indicateur	
0004 E601	11	AND	0IH	
0006 CAC8C1	12	JP	Z, INTEX	
0009 DBF0	13	TSBINT :	IN A, (0F0H) ;Vérifier type de données	
0008 E6C0	14	AND	0COH	
000D 280F	15	JR	Z, DATADA	
000F E680	16	AND	80H	
0011 CAC835	17	JP	Z, ANSDA	
0014 DBF1	18	IN	A, (0F1H) ;Lire données de commande	
0016 D605	19	SUB	5	
0018 C2C80C	20	JP	NZ, NOTBRK	
0018 C3C3BD-	21	Skip	STROBE	
	22	:		:Lire données d'unité centrale auxilliaire
001E DBF1	23	DATADA :	IN A, (0F1H)	
0020 FE03	24	CP	3	:C'est poussé?
0022 28F7	25	JR	Z, Skip	:Oui, poser indicateur C
0024 C3C884	26	JP	NOT03	
	27	END		

4. APPELS AU SYSTEME

4.1 Résumé des appels au système

Sous-programmes standard pour dispositifs et système	Page	
BEEPO	Actionne le haut-parleur	23
BEEP6	Actionne le haut-parleur	24
CMDPUT	Réglage des paramètres de commande de l'unité centrale secondaire	25
CONGET	Reçoit l'introduction de console	25
GPRBSY	Contrôle les bits d'état	26
GPRON	Envoie un caractère vers le point d'accès parallèle	27
GPROPN	Règle la taille, la position et le mode des caractères	27
GPRPUT	Envoie un caractère du registre A vers le point d'accès parallèle	27
IOCLS	Vide les points d'accès d'entrée/sortie	28
KBDSNS	Reçoit l'introduction au clavier	29
LCDPUT	Affiche un caractère	30
LPHYDSP	Affiche un caractère sur la position du curseur	30
NOPCMD	Règle les paramètres de commande de l'unité centrale secondaire	31
PRTOPN	Envoie CR et LF vers le point d'accès série	31
PUTTSB	Affiche un caractère sur la position du curseur et fait avancer la position du curseur	31
RAMRD	Lit un octet dans la mémoire d'unité centrale secondaire	32
RAMWRT	Introduit un octet dans la mémoire d'unité centrale secondaire	32
Programmes secondaires		
CHKCNTC	Contrôle les bits BREAK	33
CHKBAT	Contrôle les bits d'épuisement des piles de carte et d'unité	33
INDJMP	Provoque un saut dans le tableau de vecteurs	34
LSETCUR	Allume ou éteint le curseur	34
LBUFCLR	Vide le registre tampon d'entrée	35
MEMCADR	Contrôle l'interrupteur de prise	36
OFFO	Coupe l'alimentation	36
OFF1	Valide ou invalide le programme de lancement	36
OFFCHECK	Contrôle BREAK FLAG	37
PTMCLR	Initialise la minuterie d'alimentation	37
RAMCHK	Contrôle la MEV	37
SLEEP	Introduit l'état d'arrêt SLEEP	38
TBREAK	Essais de BREAKFLAG	38

4.1.1 Sous-programmes standards pour dispositifs et système

Le X-07 permet d'appeler directement de nombreux sous-programmes d'interpréteur BASIC depuis le programme en langage machine.

Conventions relatives à la notation

Désignation des
sous-programmes

nnnnH ← Cette adresse est appelée.

E
R
M

Information d'introduction (registre à charger avec un paramètre)

Information de retour (information de registre ou état d'indicateur sur le retour)

Information de modification (registre dont le contenu diffère avant et après l'appel)

Séquence d'appel au système

- Sauvegarder toute information "modification". (voir exemple 6)
- Pour les sous-programmes nécessitant une information d'introduction, fournir le paramètre requis dans le registre spécifié.
- L'information de retour est fournie par le sous-programme appelé.

(Exemple 6) Appel de GPRON

GPRON

CF11H—Cette adresse est appelée.

E
R
M

A—Registre dans lequel doit être placé le paramètre.

Aucun—Aucun indicateur ou autre information n'est nécessaire.

AF, B, E—Registres dont le contenu doit être sauvegardé dans la pile si nécessaire.

Programme d'utilisateur

```
F5      PUSH AF      * Sauvegarder ces registres si
C5      PUSH BC
D5      PUSH DE
3E 30,   LD A, 30H ; Placer les données de caractères 30H dans le registre A.
CD 11 CF CALL CF 11H
D1      POP DE
C1      POP BC
F1      POP AF
```

RET

Sous-programme
GPRON

L'exemple 6 utilise les décalages et les remontages. Lorsque les registres contenant l'information "modification" de retour sont assignés au même registre, il faut néanmoins, sauvegarder ce registre d'une autre manière.

BEEPO

C2DFH Actionne le haut-parleur

E	HL	IY
R	Aucun	
M	AF, BC, DE, HL, IY, EI	

BEEPO procure la même fonction que l'instruction BASIC BEEP (voir le manuel de référence du BASIC).

IY ← Tonalité
HL ← Durée

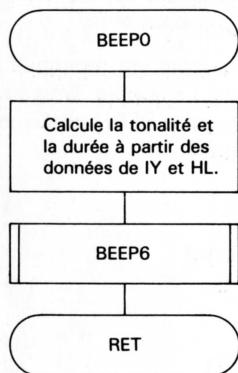


Figure 5

BEEP6

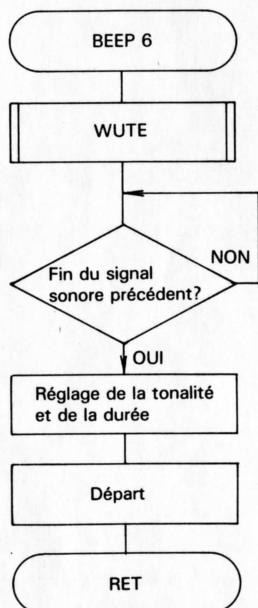
C331H

Actionne le haut-parleur.

E	HL	IY	C
R	Aucun		
M	AF, BC, DE, HL, EI		

Paramètres

IY ← Tonalité $f_0 = \text{INT}(19200/(IY + 1))$ (Hz)
HL ← Durée $t_0 = HL/f_0$ (sec)
C ← Haut-parleur marche/arrêt
Déclencher si C = FF.
Déclencher si C = 00.



Une tonalité d'avertisseur est terminée par une interruption de programme.

La fin d'un signal sonore est reconnue par le programme donné à l'exemple. Un zéro dans l'adresse d'entrée/sortie OFOH bit 5 indique la fin d'un signal sonore.

(Exemple) INA (OFOH)
BIT 5, A

CMDPUT

C92FH Réglage des paramètres de commande d'unité centrale secondaire.

E
R
M

A, BC, DE, HL

Aucun

AF, BC, HL

CMDPUT est utilisé pour générer un ordre destiné à l'unité centrale secondaire, ou pour introduire l'information d'introduction nécessaire pour obtenir l'information de réponse de l'unité centrale secondaire.

Paramètres

A←Numéro de commande. Régler le bit 7 si aucun paramètre n'est utilisé.

HL←Pointeur des paramètres, s'il y en a.

B←Comptage des octets des paramètres.

C←Comptage des octets de réponse. Régler 0 si aucun octet de réponse n'est reçu.

D←Pointeur du registre tampon utilisé pour mémoriser les octets de réponse.

CONGET

C8C5H Obtention de l'introduction de console.

E
R
M

Aucun

A, Indic-Z, Indic-C

AF

CONGET permet d'obtenir l'introduction du clavier, s'il y en a, dans le registre A et règle l'information de retour nécessaire. Il appelle CONSNS, et si une erreur d'entrée/ sortie est détectée, il saute au programme d'erreurs d'entrée/sortie.

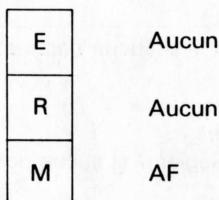
	Indic-Z	Indic-C	Registre A
Avec terminaison normale	Remise à zéro	Remise à zéro	/
Existence d'un état d'interruption	/	Mise à 1	/
Existence d'un état de piles épuisées	Mise à 1	/	0

Tableau 1

GPRBSY

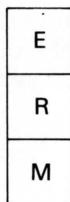
CF2AH

Contrôle les bits d'état.



GPRBSY attend que la borne BUSY du point d'accès parallèle passe à l'état bas. Il contrôle les bits OFF, BREAK, piles épuisées et contrôle S dans le secteur BREAKFLAG (à 002BH) et retourne immédiatement si l'un de ces bits est à 1.

GPRON



CF11H Envoie un caractère au point d'accès parallèle.

A

Aucun

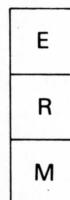
AF, B, E

Paramètres

A ← Donnée de caractères à envoyer au point d'accès parallèle.

GPRON sort le caractère du registre A vers le point d'accès parallèle.

GPROPN



CFB7H Règle la taille, la position et le mode de caractères.

Aucun

Aucun

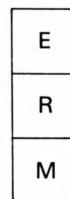
AF, BC, DE, HL

GPROPN ouvre le dispositif GPR:. Il règle le format (2), la position (TXT) et le mode de caractères.

||
TEXT

2

GPRPUT



CED6H Envoie la valeur du registre A vers le point d'accès parallèle.

A

Aucun

AF, BC, DE, HL

Paramètres

A → Données dont la dimension doit être contrôlée et envoyée au GPR

GPRPUT contrôle le format du caractère dans la position de sortie actuelle et appelle GPROUT pour envoyer les données de caractère du registre A vers le GPR. Si une interruption apparaît, le programme saute au sous-programme ABORT.

IOCLS

	C39DH	Vidage des points d'accès d'entrée/sortie.
E	Aucun	
R	Aucun	
M	AF, BC, DE	

IOCLS initialise (remet à zéro) les points d'accès parallèles, séries, optiques, imprimante et cassettes. IOCLS appelle WUTE.

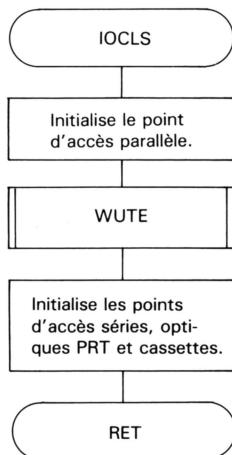


Figure 7

KBDSNS

C90AH Saisit l'entrée du clavier.

E
R
M

Aucun

A, Indic-Z

AF

KBDSNS saisit un octet du clavier dans le registre A s'il y a une introduction, et il règle l'indicateur Z s'il n'y a pas d'introduction au clavier.

LCDPUT

C1BEH Affiche un caractère.

E
R
M

A
Aucun
Aucun

Paramètres

A←Données de caractères à afficher.

LCDPUT place le caractère du registre A dans LCD et dans la MEV image d'écran (Zone de 80 octets débutant à 0214H).

LPHYDSP

C231H Affiche un caractère sur la position du curseur.

E
R
M

HL, C
Aucun
AF, BC, DE

Paramètres

C←Données de caractères à afficher
H←Position de coordonnée X du curseur
L←Position de coordonnée Y du curseur

LPHYDSP affiche le caractère du registre C à la position du curseur (origine 1) désignée par la paire de registres HL.

NOPCMD

E
R
M

E428H

A

Aucun

AF, C

Règle les paramètres de commande de l'unité centrale secondaire.

Paramètres

A ← Numéro de commande à envoyer à l'unité centrale secondaire

NOPCMD envoie le numéro de commande d'une commande n'ayant pas de paramètres et ne nécessitant pas d'information de réponse pour l'unité centrale secondaire.

PRTOPN

E
R
M

CFB0H

Aucun

Aucun

AF, BC, DE, HL

Envoie CR et LF vers le point d'accès parallèle.

PRTOPN envoie les codes CR et LF au point d'accès parallèle.

PUTTSB

E
R
M

C18AH

A

Aucun

Aucun

Affiche un caractère sur la position du curseur et fait avancer la position du curseur.

Paramètres

A ← Données de caractères à envoyer à l'unité centrale secondaire.

PUTTSB envoie le caractère du registre A vers l'unité centrale secondaire. L'unité centrale secondaire affiche le caractère sur la position actuelle du curseur et incrémente la position du curseur.

RAMRD

E348H Lit un octet dans la mémoire
d'unité centrale secondaire.

E
R
M

HL

A

AF, BC, DE

Paramètres

HL—Adresse de MEV d'unité centrale secondaire.

A—Données revenant de l'unité centrale secondaire.

RAMWRT

E334H Ecrit un octet dans la mémoire
d'unité centrale secondaire.

E
R
M

A, HL

Aucun

AF, BC, DE

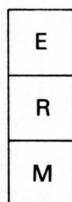
Paramètres

HL—Adresse de MEV d'unité centrale secondaire.

A—Données à écrire dans la MEV d'unité centrale secondaire.

CHKCNTC

C02DH Contrôle les bits BREAK.



Aucun

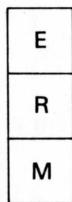
Indic-Z

Aucun

CHKCNTC interroge BREAKFLAG. Si le bit d'introduction est à 1, CHKCNTC le remet à zéro, il vide le registre tampon de touche et met à 1 le bit d'indication Z. Si le bit d'interruption est à zéro, CHKCNTC envoie la commande au programme appelant avec l'indicateur Z réglé.

CHKBAT

C5A6H Contrôle les bits d'épuisement de piles de cartes et de piles d'unité principale.



Aucun

A, Indic-Z

Aucun

CHKBAT interroge les bits BREAKFLAG. Si le bit de piles épuisées est à 1, il affiche le message "Low battery" (piles épuisées) et renvoie la commande au programme appelant avec l'indicateur Z remis à zéro. Si le bit de piles de carte épuisées est à 1, CHKBAT affiche le message "Card low battery" (piles de carte épuisées) et renvoie la commande au programme appelant avec l'indicateur Z remis à zéro. Si les deux bits sont à 1, CHKBAT renvoie la commande au programme appelant avec l'indicateur Z mis à 1.

* Le registre A est chargé avec les dernières données BREAKFLAG.

INDJMP

EE13H Provoque un saut dans le tableau du vecteur.

E	C, HL
R	HL
M	AF, BC, DE, HL

Paramètres

HL — Adresse de saut de tableau
C — Décalage

INDJMP provoque le passage de la commande de programme à l'adresse de l'introduction de saut de tableau désignée par les registres HL et C.



HL désigne l'adresse de départ du tableau de saut.
C = 1 désigne SYMBOL1, C = 2 désigne SYMBOL2, etc.

LSETCUR

C0A1H Allume et éteint le curseur.

E	A
R	Aucun
M	AF, BC, DE

Paramètres

A → 0 ou 1

L'appel de LSETCUR avec le registre A chargé de 0 affiche le curseur sur l'écran d'affichage. L'appel de LSETCUR avec le registre A chargé de 1 provoque la disparition du curseur de l'écran.

LBUFCLR

COBDH

Vide le registre tampon d'entrée.

E
R
M

Aucun

Aucun

AF

LBUFCLR vide le registre tampon d'entrée d'unité centrale secondaire et de clavier.

MEMCADR

C557H Contrôle le commutateur de prise.

E
R
M

Aucun
HL
AF, HL

MEMCADR charge la paire de registres HL avec 4000H si le commutateur de prise est réglé sur la position de mémoire vive (MEV), et avec 2000H si le commutateur est réglé sur la position de mémoire morte (MEM).

OFF0

C5C3H Coupe l'alimentation.

E
R
M

Aucun
Aucun
Tout

OFF0 traite l'information de mémoire devant être gérée par l'interpréteur BASIC et coupe l'alimentation du X-07 (voir la partie 5 "Séquence d'enclenchement d'alimentation du X-07" du présent manuel).

OFF1

C5CDH Valide ou invalide le programme de lancement.

E
R
M

A
Aucun
Aucun

Paramètres

A → Données spécifiant la validation ou l'invalidation du programme de lancement.

Lorsqu'il est appelé avec le registre A chargé de 1, OFF1 valide le programme de lancement et coupe l'alimentation du X-07. Lorsqu'il est appelé avec le registre A chargé d'un 2, OFF1 invalide le programme de lancement avant de couper l'alimentation du X-07.

OFFCHECK

C5A6H

Contrôle de BREAKFLAG.

E
R
M

Aucun

Indic-S

Aucun

OFFCHECK interroge le secteur BREAKFLAG et renvoie la commande au programme appelant si l'indicateur S est remis à zéro. Si l'indicateur S est à 1, OFFCHECK ramène tous les bits de BREAKFLAG à zéro et coupe l'alimentation du X-07.

PTMCLR

E327H

Initialise la minuterie d'
alimentation.

E
R
M

Aucun

Aucun

AF, B

PTMCLR initialise la minuterie d'alimentation de 15 minutes en remettant à zéro le bit d'arrêt de minuterie dans le secteur BREAKFLAG.

RAMCHK

C62EH

Contrôle pour la MEV.

E
R
M

HL

Indic-Z

AF, B

Paramètres

HL ← Adresse à interroger

RAMCHK réalise un contrôle pour déterminer si l'adresse indiquée par la paire de registres HL est une adresse de MEV. Si l'adresse est une adresse de MEV, il met l'indicateur Z à 1, sinon il met l'indicateur Z à zéro.

SLEEP

C59BH Introduit l'état SLEEP OFF.

E
R
M

Aucun
Aucun
Aucun

SLEEP sauvegarde l'indicateur de piles actuel à l'adresse 0266H et le contenu de la paire de registres HL à l'adresse 0264H, puis il introduit l'état SLEEP OFF.

TBREAK

E8E8H Interrogation de BREAKFLAG.

E
R
M

Aucun
Indic-Z
AF

TBREAK interroge le secteur de BREAKFLAG (à 002BH) et met l'indicateur Z à 1 si un bit quelconque se trouve à 1.

4.1.2 Sous-programmes de commande de dispositifs

a) Séquence d'initialisation de dispositif

Le BASIC du X-07 assigne les dispositifs d'entrée/sortie à cinq adresses de dispositifs et il les initialise lorsqu'il rencontre la commande INIT #. Les adresses et les paramètres de dispositifs nécessaires au contrôle des dispositifs d'entrée/sortie sont mémorisés dans le tableau d'adresses de dispositifs de la MEV de travail. Les fonctions de commande INIT # servant à cataloguer et à initialiser les adresses de dispositifs d'entrée/sortie dans le tableau d'adresses de dispositifs peuvent aussi être obtenues par la fonction d'appel au système EXOPN qui sera décrite ultérieurement.

Le reste de cette partie décrit la fonction d'initialisation de la commande INIT # (EXOPN).

- **CASI, CASO, OPT, COM et PRT**

Ces cinq dispositifs ne peuvent pas fonctionner simultanément car il n'y a qu'un ACIA et qu'un générateur de débit en bauds pour les commander. De plus, un mauvais fonctionnement apparaîtra si un second dispositif est démarré après avoir mis en marche le premier et si l'on tente ensuite de faire marcher à nouveau le premier dispositif. Lorsqu'un nouveau dispositif doit être initialisé, l'introduction concernant le dispositif ayant été préalablement assigné au ACIA est effacée du tableau d'adresses de dispositifs, ce qui provoque son débranchement du ACIA. Le registre tampon d'entrée est aussi vidé à ce moment-là pour les dispositifs d'entrée et la borne RTS est réglée à l'état haut pour le dispositif COM.

- **GPR**

Lorsque l'imprimante graphique est initialisée, la taille de caractère est réglée à 2 et une paire de codes CR/LF est envoyée à l'imprimante. Si l'imprimante est trouvée en mode graphique, elle est ramenée au mode de texte.

- **LPT**

Une paire de codes CR/LF est envoyée au point d'accès parallèle Centronics.

- **KBD**

Le registre tampon du clavier est vidé.

- **MEV**

Le fichier présentant le nom de fichier spécifié est recherché, et s'il est trouvé, les points d'écriture et de lecture sont placés au début du fichier. Le premier paramètre étant ignoré, il n'est pas nécessaire de le spécifier.

Si le fichier spécifié n'est pas trouvé, un nouveau fichier sera créé et les points d'écriture et de lecture seront placés au début du fichier. Dans ce cas, le premier paramètre est mis en référence.

- **CON**

Aucune action d'initialisation explicite n'est entreprise.

Note 5: Si l'on exécute une commande BEEP tout en utilisant CASI, CASO, OPT, COM ou PRT, les cinq dispositifs seront effacés du tableau d'adresses de dispositifs, car le programme BEEP produit la tonalité requise en utilisant le générateur de débit en bauds.

(Exemple 7)

- 10 INIT # 1, "COM :"Initialise le point d'accès d'entrée/sortie série à 4800 bauds.
- 20 INIT # 2, "OPT :"Initialise le dispositif OPT à 1200 bauds. Le dispositif COM précédemment assigné au dispositif N°1 est effacé du tableau d'adresses de dispositifs.
- 30 OUT # 1,65L'exécution de cette commande provoque une erreur NO (NOT OPEN).
- 10 INIT # 1, "COM :"Initialise le point d'accès d'entrée/sortie série à 4800 bauds.
- 20 BEEP5, 10Génère une tonalité d'avertisseur. Le dispositif COM précédemment assigné au dispositif N°1 est effacé du tableau d'adresses de dispositifs.
- 30 A=INP (# 1)L'exécution de cette commande cause une erreur NO (NOT OPEN).
- 10 INP # 1, "PRT :"Initialisation de l'imprimante thermique.
- 20 OUT # 1, IEnvoie I (données à 1 octet) vers l'imprimante.
- 30 BEEP10, 10Génère une tonalité d'avertissement.
- 40 INIT # 1, "PRT :"Initialise à nouveau l'imprimante thermique.
- 50 OUT # 1, I+1Envoie I + 1 vers l'imprimante.

b) Utilisation des sous-programmes de commande de dispositifs

Le sous-programme EXOPN est utilisé pour initialiser un dispositif d'entrée/sortie ou pour le cataloguer dans le tableau d'adresses de dispositifs.

Les paramètres à préparer pour l'appel de ce sous-programme diffèrent d'un dispositif à l'autre. Le tableau 2 donne la liste des paramètres devant être réglés pour l'initialisation des dispositifs d'entrée/sortie.

Tableau 2

Désignation du dispositif Registre	MEV	CON	KBD	GPR	LPT(Centronics)
DE	'MEVHND'	'CONHND'	'KBDHND'	'GPRHND'	'PRTHND'
A	00	—	—	—	—
B	Type de données 'A-X'	—	—	—	—
IX	Dimension des fichiers	—	—	—	—
IY	Pointeur d'adresses de dispositifs (Note 5)	Comme à gauche	Comme à gauche	Comme à gauche	Pointeur d'adresses de dispositifs
(ESCAPE)	Longueur du nom (longueur de fichier)	—	—	—	—
(ESCAPE + 1)	Pointeur vers le nom proprement dit	—	—	—	—

Désignation du dispositif Registre	CASI	CASO	OPT	COM	PRT (X-03)
DE	'CASTHND'	'CASHND'	'OPTHND'	'SIOHND'	'X04HND'
A	—	—	—	—	—
B	—	—	Mode ACIA	Comme à gauche	—
IX	—	—	Débit en bauds (Note 6)	Comme à gauche	Comme à gauche
IY	Comme à gauche	Comme à gauche	Comme à gauche	Comme à gauche	Comme à gauche
(ESCAPE)	—	—	—	—	—
(ESCAPE + 1)	—	—	—	—	—

- (Note 4)** Voir le tableau d'adresses de symboles donné à la fin du présent manuel pour les adresses réelles et les labels enfermés entre des apostrophes ('').
- (Note 5)** Utiliser en tant que pointeur d'adresses de dispositifs l'un des cinq labels symboliques suivants: 'KBNTBL', 'KBNTBL' + 8,..., 'KBNTBL' + 8 × n, 'KBNTBL' + 32.
- (Note 6)** La plage de débit en bauds de OPT est de: 100 bps ≤ débit en bauds ≤ 2400 bps.
 La plage de débit en bauds de COM est de: 100 bps ≤ débit en bauds ≤ 800 bps.

Il est possible de mettre en place l'environnement pour réaliser les opérations PUT et GET pour un dispositif d'entrée/sortie initialisé par le sous-programme EXOPN, en appelant le sous-programme DASSIGN spécifiant le pointeur d'adresses de dispositifs (même données que celles placées dans le registre IY lors de l'exécution de EXOPN) dans la paire de registres DE. Un appel ultérieur des programmes GETAD ou OUT-CHR place les données d'entrée dans le registre A ou envoie les données du registre A vers le dispositif d'entrée/ sortie, respectivement. Les données de tous les registres autres que AF sont garanties pendant l'exécution du sous-programme GETAD et tous les registres sont garantis pendant le sous-programme OUT-CHR.

c) Exemples d'entrée/sortie

Programme 1: Entrée/sortie depuis et vers l'imprimante graphique et le point d'accès d'entrée/sortie série.

Initialiser l'imprimante graphique.	LD DE,GPRHND	Charger l'adresse de tableau du module de gestion d'imprimante graphique dans DE.
	LD IY,KBNTBL	Régler l'adresse de départ de la zone MEV contiguë à 8 octets.
	CALL EXOPN	Les paramètres nécessaires au contrôle de l'imprimante graphique sont transmis à KBNTBL.

	LD DE,SIOHND LD IY,KBNTBL+8 LD B,'B'	Régler le mode ASCII 'B' (pas de parité).
Initialiser l'entrée/ sortie série.	LD IX,4800B CALL EXOPN	Régler le débit en bauds à 4800 bauds.
Préparer le trans- fert des données vers l'imprimante graphique	LD DE,KBNTBL CALL DASSIGN	Assigner le sous-programme de sortie à PUTAD
Envoyer 'G' vers l'imprimante gra- phique.	LD A,'G' CALL DASSIGN	Charger les données de sortie.
	}	
Envoyer 'P' vers l'imprimante gra- phique.	LD A,'P' CALL OUT CHR	Charger les données de sortie
	}	
Envoyer 'P' vers l'imprimante gra- phique	LD A,'R' CALL OUT CHR	Charger les données de sortie
	}	
Envoyer 'P' vers l'imprimante gra- phique	LD A,'R' CALL OUT CHR LD DE,KBNTB+8 CALL DASSIGN	Charger les données de sortie
	}	(Note 7: Etat de l'indicateur)
Préparer l'entrée/ sortie de l'entrée/ sortie série	CALL GETAD	Charger les données d'entrée (Note 8)
	}	
Envoyer les don- nées vers l'entrée/ sortie série	LD A,'S' CALL OUT CHR	Charger les données de sortie (Note 9)
	}	

Préparer le transfert des données vers l'imprimante graphique

LD DE,KBNTBL

CLL DASSIGN

}

Envoyer '1' vers l'imprimante graphique

LD A,'1'

CALL OUT CHR

(Note 7) Les états de l'indicateur pendant l'apparition de différentes erreurs d'entrée ou interruptions sont indiqués ci-dessous.

Etat	Registre/indicateur	A	C	Z
RETOUR NORMAL	Données d'entrée	0	0	0
ERREUR D'ENTREE	Imprévisible	0	1	
RETOUR DE RUPTURE	0	1	1	

Tableau 3

(Note 8) Le traitement d'entrée/sortie série étant réalisé en mode d'interruption-entraînement utilisant un registre tampon d'entrée à 31 octets, EXOPN prend toutes les données du registre tampon d'entrée au moyen d'appels au sous-programme GETAD. La commande RTS est accomplie par les six autres octets.

(Note 9) Les erreurs de sortie sous-programme ne peuvent apparaître que dans les dispositifs MEV (c'est-à-dire lorsque l'on tente d'écrire des données dépassant la dimension du fichier). La commande étant directement transmise vers le sous-programme ABORT en présence d'une interruption pendant le processus de sortie, l'exécution retourne à l'interpréteur BASIC au moyen du sous-programme BASIC ABORT, sauf si l'on fournit son propre programme, tel que celui indiqué ci-dessous.

Programme 2

LD A,0C3H

Préparer un sous-programme de destination pour recevoir la commande lorsque le programme est suspendu.

LD [ABTHOCK],A

LD HL,ABORT

LD [ABTHOCK + 1],HL

}

Opération d'entrée/sortie

}

LD A,0C9H

Charger A avec le code 0C9H pour rétablir la destination du sous-programme ABORT sur l'adresse d'origine.

LD [ABTHOCK],A

↓

Retourner au BASIC.

ABORT: ID SP,nnnn

l

Préparer à nouveau le pointeur de piles, car le programme ne connaît pas le niveau de la pile auquel une commande existe.

Suspendre le traitement.

l

JP MAIN

LD A,E

Le registre E contient un code d'erreurs.

CP SNERR

JR Znnnn

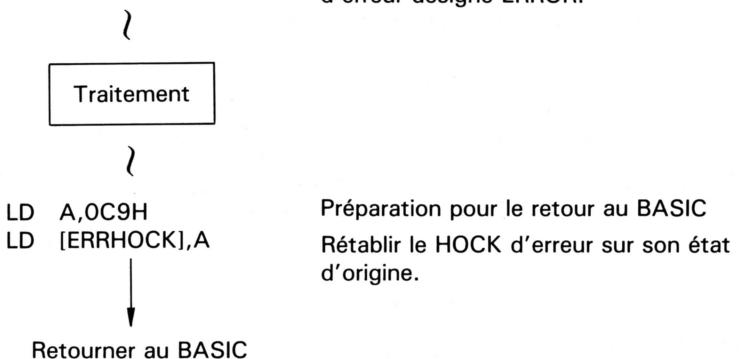
}

d) Changement de l'adresse de branchement d'erreurs

La commande du programme est directement transférée au sous-programme de traitement d'erreurs BASIC lorsqu'une erreur apparaît pendant l'exécution des sous-programmes de sortie de dispositifs EXOPN ou MEV. Cela provoque des problèmes si ces sous-programmes sont appelés à partir de programmes en langage machine. Pour éviter cela, un HOCK est prévu au début du sous-programme d'erreurs BASIC qui transfère la commande au sous-programme ERRHOCK situé dans la MEV, lors de l'introduction. Un programme HOCK, tel que le programme 3 ci-dessous, changeant l'adresse de branchement d'erreurs, sera utile pour éviter que la commande de programme ne soit retransférée au BASIC en présence d'une erreur pendant l'exécution d'un sous-programme BASIC appelé par le programme en langage machine (ERRHOCK = 00ABH).

Programme 3

LD A,OC3H	Ecrire le code de saut
LD [ERRHOCK],A	
LD HL,ERROR	En présence d'une erreur, la commande
LD [ERRHOCK + 1],HL	est transférée au sous-programme d'erreur désigné ERROR



ERROR: LD SP,xxxx	Régler à nouveau le pointeur de pile, car le programme ne connaît pas le niveau de la pile auquel existe une commande.
LD A,E	
CP SNERR	Le registre E contient un code d'erreurs identifiant le type d'erreurs réalisé.
JR Z,yyyy	

e) **Augmentation du nombre de dispositifs d'entrée/sortie**

Cette partie décrit la manière d'augmenter le nombre de dispositifs d'entrée/sortie pouvant être géré par les commandes et les instructions d'entrée/sortie BASIC du X-07, telles que POINT #n, LIST #n, DIR #n, OUT #n, INP (#n), SNS (#n), TRON (#n), SAVE "fichier" et LOAD "fichier". Cette possibilité de développement du X-07 permet de modifier le fonctionnement de dispositifs d'entrée/sortie prédéfini tel que COM:, RAM:, GPR:, et OPT:. Les préparatifs concernent les sous-programmes d'initialisation du matériel et du logiciel, les sous-programmes d'ouverture, de sortie à 1 octet et de mémorisation à 1 octet.

Le X-07 comporte une zone de travail désignée DEVHOCK, qui est un tableau consistant de quatre entrées à 4 octets. La troisième et la quatrième entrées de DEVHOCK ne sont pas disponibles pour l'utilisateur.

Lors de l'exécution d'une commande INIT, LOAD, ou SAVE, le BASIC balaie la liste des noms de dispositifs définie du système pour trouver une correspondance avec le paramètre de la commande, puis il accède au tableau DEVHOCK. Le premier octet de chaque entrée est une batterie de dispositif. Cet octet doit être mis à 0; pour que cet octet ait une signification, un matériel spécial serait nécessaire. Les trois octets restants de l'entrée sont utilisés pour contenir le code programme JP xxxx, qui transfère la commande à un sous-programme de traitement. Normalement, ce secteur contient un code de retour. Si l'on donne à ce secteur une instruction provoquant le branchement du programme sur un sous-programme de gestion de dispositif, la commande est passée à ce sous-programme lors de chaque exécution d'une commande INIT, SAVE ou LOAD. Ce HOCK peut être utilisé pour cataloguer un nouveau dispositif d'entrée/sortie dans le système.

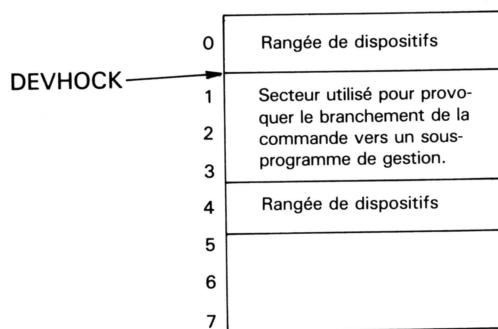


Figure 8

Un exemple de programme est donné ci-dessous afin d'illustrer la manière de cataloguer un nouveau dispositif d'entrée/sortie. Considérons qu'un point d'accès à sortie parallèle 8 bits est assigné à l'adresse d'entrée/sortie OH, et qu'un point d'accès d'entrée parallèle 8 bits est assigné à l'adresse d'entrée/sortie 1H. L'exemple ci-dessous donne au point d'accès de sortie la désignation PIO: et rend le point d'accès accessible aux commandes et aux fonctions BASIC. Placer tout d'abord le mot clé "amour" sur la frontière de 2K-octets au-dessus de 1800H. Cela permet au X-07 de sauter à l'un des trois points d'entrée lorsque son alimentation est enclenchée (pour plus de détails sur les trois points d'entrée, voir la partie 2.2.2). Le premier point d'entrée doit être assigné à un sous-programme qui initialise le matériel d'entrée/sortie. Le X-07 étant invalidé pour l'interruption au moment de l'enclenchement de l'alimentation, le mode LSI périphérique doit être réglé en ce point. L'exemple de programme ci-dessous assume des points d'accès d'entrée/sortie simples et envoie 0 au point d'accès de sortie (car l'état du point d'accès de sortie est imprévisible au moment de l'enclenchement de l'alimentation).

Le second point d'entrée doit être introduit immédiatement avant l'apparition du message de Copyright du BASIC. Réécrire l'entrée DEVHOCK dans ce but. Dans le code de programme ci-dessous, le code JP représente 0C3H. Lorsque ce programme est exécuté, le code JP DEVSCN est placé en DEVHOCK + 1.

```
LD A,JPcode
LD [DEVHOCK + 1],A
LD HL,DEVSCN
LD [DEVHOCK + 2],HL
```

Le troisième point d'entrée doit être introduit lorsque le X-07 est lancé après la terminaison de la commande SLEEP. Dans cette entrée, initialiser le matériel du X-07 et réécrire DEVHOCK.

Si l'on introduit la commande INIT = 1, "PIO:XYZ", 1000H, "T", le BASIC du X-07 explore les noms prédéfinis du système, et si une correspondance apparaît ou non, il exécute le code JP DEVSCN et saute à DEVSCN. En ce point, un 0 dans l'indicateur de report indique qu'un nom de dispositif correspondant a été trouvé. Un 1 indique qu'aucune correspondance n'est apparue. Un nouveau dispositif d'entrée/ sortie allant être défini, DEVSCN ne doit rien renvoyer si l'indicateur de report à mis à 0, c'est-à-dire si une correspondance apparaît. Si aucune correspondance n'apparaît, DEVSCN doit rechercher une correspondance avec le nom du nouveau dispositif à définir. Les données dans les paires de registres IX et BC ne doivent pas être détruites lorsque l'on sort de ce sous-programme. Bien que la recherche d'une correspondance de nom de dispositif puisse sembler être une tâche difficile, il est possible

de faire appel dans ce but à un sous-programme approprié dans la MEM BASIC. Il suffit de préparer alors un tableau pour définir le nouveau dispositif.

```
DEVSCN RET NC  
LD DE,DEVTBL  
CALL DTSCAN  
RET
```

Le tableau doit présenter le format suivant:

- (1) Désignation du dispositif: Bit 7 (bit le plus significatif) du dernier octet du nom doit être 1.
- (2) Données de rangées: Doit toujours être réglé à 1.
- (3) Code de dispositifs: Un code préservé au dispositif doit être développé. Le X-07 utilise déjà 13 entrées et réserve un total de 19 entrées. Il faut utiliser des entrées de 20 et plus. Le X-07 fonctionnera d'une manière imprévisible si l'on utilise une entrée se situant au-dessous de l'entrée de tableau 20.
- (4) Sous-programme de sortie à 1 octet
- (5) Sous-programme d'entrée à 1 octet
- (6) Sous-programme d'ouverture de dispositif
- (7) Programme optionnel

Terminaison de tableau: 0

- (1) DEFB "PIO", ":" + 128
- (2) DEFB 0
- (3) DEFB 21
- (4) DEFW PIOPUT
- (5) DEFW PISONS
- (6) DEFW PIOOPN
- (7) DEFW PIOOPT

```
DEFB "ABC", "D" + 128  
|  
DEFW ABCDOPT  
(8) DEFB 0
```

Il est possible de définir n'importe quel nombre de combinaisons de codes de (1) à (7) entre (7) et (8). Lorsque DTSCAN est appelé par l'adresse de départ du tableau ci-dessous dans la paire de registres DE, et qu'une correspondance apparaît avec le nom du dispositif "PIO:", DTSCAN renvoie la commande au sous-programme principal avec la paire de registres DE indiquant les données de rangées (2) et avec l'indicateur de report mis à 1. PIO: étant assigné à un point d'accès d'entrée dans cet exemple, l'indicateur de report est mis à 0 lorsqu'il est retourné de DEVHOCK.

Comme le montrent les explications ci-dessus, le nouveau dispositif est assigné à l'adresse de dispositif #n lorsque la commande est retournée de DEVHOCK, avec l'indicateur de report mis à 0 et la paire de registres DE chargée avec l'adresse de départ (et non pas le nom du dispositif; la paire de registres DE doit être dirigée vers les données de rangées) du tableau contenant les codes (2) à (7). Le BASIC explore alors les entrées DEVHOCK restantes et transfère la commande vers le sous-programme ouvert associé au tableau de correspondance, spécifiant les paramètres suivants:

IX: Contient la représentation binaire de la première donnée d'opérateur 1000.

B: Contient le code &H de la deuxième donnée d'opérateur "T".

A: Mode d'ouverture

O: Actionné par la commande INIT ou LOAD.

1: Actionné par la commande SAVE.

[ESCAPE]: Contient le nombre restant de caractères du nom entré. Ici 3 est indiqué car le nom entré est "XYZ".

[ESCAPE + 1][ESCAPE + 2]: Contient l'adresse des caractères du nom entré restant. [ESCAPE + 1] contient l'octet d'ordre inférieur de l'adresse et [ESCAPE + 2] contient l'octet d'ordre supérieur.

Ces paramètres sont ignorés car ils ne sont pas nécessaires dans ce programme. Il est possible de passer des paramètres entre le sous-programme d'ouverture et la commande (exemple, nombre de registres tampons dans le premier paramètre et mode de fonctionnement dans le second paramètre. Cela vaut pour les commandes SAVE et LOAD.

La fonction de ce sous-programme d'ouverture est d'envoyer 0 au point d'accès de sortie. Néanmoins, lors de l'utilisation de LSI périphérique, le sous-programme d'ouverture doit établir le mode de fonctionnement, réaliser les définitions d'entrée/sortie, initialiser les registres tampons (s'il y en a), et réservier les canaux d'interruption de programme. Le sous-programme d'ouverture sauvegarde le contenu des paires de registres HL, BC et DE aux emplacements KBNTBLn + 2, KBNTBLn + 4, et KBNTBLn + 6, respectivement. Cette information peut être utilisée en tant qu'indicateur d'entrée/sortie, registre tampon et bloc de commande. Néanmoins, dans cet exemple, elle n'est pas utilisée.

La description du fonctionnement des sous-programmes de gestion d'entrée/sortie activé lors de l'exécution d'une commande OUT = 1, &H12 est donnée ci-après. Lorsque cette commande est exécutée, le contrôle est transféré au sous-programme de sortie à 1 octet qui est catalogué dans la table de définitions du dispositif. Dans le programme de l'exemple, le contrôle est transféré à PIOPUT. Le registre A contenant les données de sortie, il est envoyé tel quel au point d'accès de sortie. Le sous-programme de sortie à 1 octet peut détruire le contenu de tous les registres. Il est à remarquer que si l'on utilise une boucle attendant que les dispositifs correspondant soient prêts, les touches d'interruption et d'arrêt ne fonctionneront pas, provoquant une boucle infinie. Pour éviter cela, introduire l'étape CALL ABORT dans la boucle d'attente. Ce sous-programme retourne sans modifier le contenu du registre si aucun des états de touche d'interruption, touche d'arrêt ou pile épuisée n'apparaît. En présence de l'un de ces états, la commande n'est pas renvoyée au sous-programme appelant, mais elle est transférée à un sous-programme de gestion approprié et provoque l'émission d'un message ABORT dans xxxx par l'interpréteur BASIC et l'attente d'une commande BASIC. Les données se trouvant dans les registres HL, BC et DE établies par le sous-programme d'ouverture sont sauvegardées aux endroits indiqués par ACSDCB avec des décalages d'octets de 2, 4 et 6, respectivement. En conséquence, il est possible de placer les données en différents endroits pour différentes adresses de dispositifs. Cela est illustré à la figure 9.

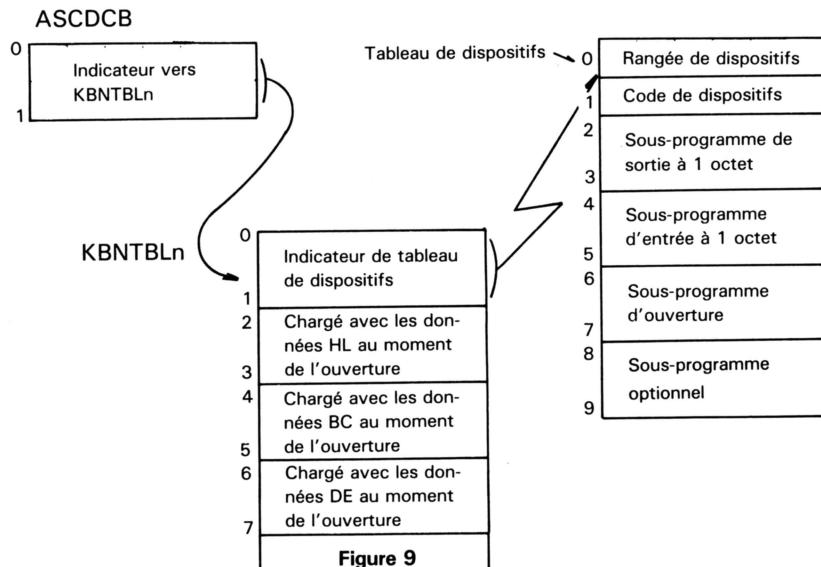


Figure 9

Le fonctionnement de la fonction INP(# 1) est décrit ci-dessous. Lorsque cette commande est exécutée, le contrôle est transféré au sous-programme d'entrée à 1 octet dans le tableau de dispositifs associés. Dans l'exemple de programme, le contrôle est transféré à PIOSNS. Ce sous-programme doit retourner immédiatement si aucune donnée n'est présente au point d'accès d'entrée. Sinon, il n'aurait pas de signification en tant que fonction. Dans cet exemple, on considère que les données sont toujours présentes à IPORT (en d'autres termes, il n'apparaît pas d'état sans entrée), et par conséquent le sous-programme retournera toujours les données introduites. S'il est nécessaire d'attendre que les données soient reçues au point d'accès d'entrée, le sous-programme devrait simplement retourner le contrôle au sous-programme appelant avec l'indicateur de report mis à 0 et l'indicateur Z mis à 1, au lieu de former une boucle dans le sous-programme lui-même. Le sous-programme INP(# n) appelle le sous-programme d'introduction fourni par l'utilisateur jusqu'à ce que des données soient reçues au point d'accès d'entrée. Il transfère automatiquement le contrôle au sous-programme ABORT si un état de touche d'interruption, touche d'arrêt ou pile épuisée apparaît, de façon à ce que le sous-programme d'introduction n'ait pas à contrôler de tels états. Si une erreur de dispositif d'introduction apparaît, le sous-programme d'introduction doit retourner avec l'indicateur de report puis à 1. Cela procure à la MEM BASIC la possibilité de détecter n'importe quelles erreurs d'entrée/sortie. La relation entre les états d'erreurs possibles et l'état des indicateurs et le contenu du registre A sont résumés dans le tableau 4.

Etat	Indicateur de report	Indicateur zéro	Reg A
Données du registre A	0	0	data
Aucun donnée présente	0	1	x
Erreur d'introduction	1	x	x

Tableau 4

Le sous-programme optionnel est un sous-programme spécial utilisé pour des dispositifs spéciaux. Il peut écrire une en-tête de cassette, créer un intervalle temporisé ou indiquer la position actuelle du curseur sur l'affichage de la console ou la position d'impression actuelle sur l'imprimante. Il n'est jamais appelé par les commandes INP(# n) ou OUT # n, et par conséquent il n'est pas décrit en détails ici.

Le X-07 ne prend pas en charge le sous-programme de fermeture. Si nécessaire, il faut réaliser son propre sous-programme de fermeture pour un nouveau dispositif utilisant la commande EXEC.

4.2 Commandes et fonctions d'unité centrale secondaire

L'unité centrale secondaire du X-07 est équipée de différentes fonctions. L'unité centrale principale (une NSC-800) utilise l'unité centrale secondaire en lui envoyant ses commandes.

Les commandes peuvent être acheminées à l'unité centrale secondaire en utilisant les appels de système BASIC suivants:

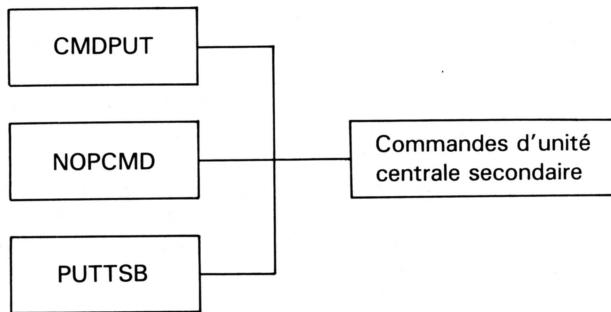


Figure 10

4.2.1 Liste des commandes d'unité centrale secondaire

N°	Fonction	Nombre d'octets (commande et opérateur)	Longueur de réponse (octets)	Traitement
00	Dummy	1	0	N° d'opération (s'applique également au n° de commande 47 et plus).
01	TIME CALL	1	8	Année (2 octets, mois (1 octet), date (1 octet))
02	STICK	1	1	Lecture des touches de curseur et renvoi d'une direction de 0 à 8 en code ASCII.
03	STRIG	1	1	Renvoi de TRUE si les touches F6 ou F12 sont enfoncées; sinon, renvoi de FALSE.
04	STRIG1	1	1	Renvoie TRUE pour un espace; sinon, renvoi de FALSE.
05	RAM READ	3	1	Mise des données dans A depuis l'adresse indiquée en HL.
06	RAM WRITE	4	0	Mise des données de A dans l'adresse indiquée en HL.
07	SCROLL SET	3	0	Envie les lignes de départ et de fin en format binaire.
08	SCROLL EXET	1	0	Défilement jusqu'à la limite de défilement sans déplacement du curseur
09	LINE CLEAR	2	0	Effacement de la ligne d'affichage spécifiée.
0A	TIME SET	9	0	Seconde (1 octet), minute (1 octet), heure (1 octet), Date (1 octet), mois (1 octet), année (L) (1 octet), année (H) (1 octet)
0B	Calcul et réglage du jour de la semaine	1	0	Après TIME SET, calcul et réglage du jour de la semaine.
0C	Réglage des données d'alarme	9	0	Comme pour 01, mais les secondes est un paramètre fictif; le bit le plus significatif est ignoré pour tous les paramètres.
0D	BUZZER OFF	1	0	Fait cesser le retentissement du vibreur par le CCU lorsque l'heure d'alarme est atteinte.
0E	BUZZER ON	1	0	Fait retentir le vibreur du CCU lorsque l'heure d'alarme est atteinte.
0F	Transfert de BIT IMAGE	2	120	Envoi de la ligne d'opérateur (1 ligne) à l'unité centrale.
10	POINT	3	1	Renvoie TRUE si un point est présent aux coordonnées (X, Y) spécifiées; sinon, renvoi de FALSE.
11	P SET	3	0	Positionnement d'un point aux coordonnées (X, Y) spécifiées.
12	PRE SET	3	0	Effacement du point aux coordonnées (X, Y) spécifiées.
13	PEOR	3	0	Inversion du posage aux coordonnées (X, Y) spécifiées.
14	LINE	5	0	Posage de tous les points d'une ligne entre (X1, Y1) et (X2, Y2).
15	CIRCLE	4	0	Traçage d'un cercle de rayon r autour du centre (X, Y).
16	Ecriture USER DEFINE KEY	2 + no. caractères + "00"	0	Nombre de touches définies + caractères + "00".

Tableau 5 (a)

N°	Fonction	Nombre d'octets (commande et opérateur)	Longueur de réponse (octets)	Traitement
17	Lecture de USER DEFINE KEY	2	Nombre de caractères + "00"	Envoi du numéro de la touche définissable à lire
18	UDK ON	1	0	Provoque le début d'envoi de UDK par l'unité centrale secondaire.
19	UDK OFF	1	0	Provoque l'arrêt d'envoi de UDK par l'unité centrale secondaire
1A	Ecriture de USER DEFIN CHR	10	0	Posage des caractères.
1B	Lecture de caractères	2	8	Envoi des codes de caractères depuis 20H.
1C	UDC INT	1	0	Renvoie à UDC (Allemand ou Français).
1D	Ecriture de START PROGRAM	1 + no. caractères + "00"	0	Ecriture des codes de caractères + "00" depuis l'adresse de départ.
1E	Ecriture de SP CONT	1 + no. caractères + "00"	0	Ecrit après la commande ID.
1F	SP ON	1	0	Débute la sortie du programme de lancement.
20	SP OFF	1	0	Termine la sortie du programme de lancement.
21	SP READ	1	N°. caractères + "00"	Lecture du programme de lancement.
22	ON STATE	1	1	Bit 0 = 1 (commutateur OFF, ON), bit 6 = 1 (SLEEP ON)
23	OFF REQUEST	1	0	Demande de coupure d'alimentation.
24	LOCATE	4	0	Déplace le curseur aux coordonnées (X, Y) et affiche les caractères.
25	Affichage du curseur.	1	0	Allumage du curseur.
26	Non affichage du curseur.	1	0	Extinction du curseur.
27	TEST KEY	3	1	Traverse la ligne K pour la ligne d'échantillonage spécifiée.
28	TEST CHR	2	1	Renvoie TRUE si le code ASCII spécifié est trouvé; sinon, renvoi de FALSE.
29	Remise à zéro des secondes.	1	0	Mise des secondes à "00".
2A	Réglage des données de date.	2	0	Ecriture des données de date
2B	Effacement de l'affichage	1	0	Coupure du signal LCD commun
2C	Allumage de l'affichage.	1	0	Sorties du signal LCD commun.
2D	KEY BUFFER CLEAR	1	0	Vidage du registre tampon de touches, sans affecter UDK et SP.
2E	Effacement de l'affichage	1	0	Effacement de la totalité de l'affichage (la limite de défilement est CHRS (OC)).
2F	HOME	1	0	Déplace le curseur à (0, 0) (la limite de défilement est CHRS(OB)).

Tableau 5 (b)

N°	Fonction	Nombre d'octets (commande et opérateur)	Longueur de réponse (octets)	Traitement
30	Affichage de UDK.	1	0	Affichage des labels de touches définies par l'utilisateur à ligne 3.
31	Non affichage de UDK.	1	0	Extinction de l'affichage des labels de touches définies par l'utilisateur.
32	REPEAT KEY ON	1	0	Validation d'entrée répétée (ne s'applique pas aux touches spéciales).
33	REPEAT KEY OFF	1	0	Invalidation d'entrée répétée (ne s'applique pas aux touches spéciales).
34	Réglage des caractères "kana"	1	0	Assignation des caractères "kana" à UDK (pour PC6000).
35	Ecriture de UDK CONT	2 + no. caractères + "00".	0	UDK no. caractères + "00".
36	Lecture d'alarme	1	8	Alarme pour commande 01.
37	BUZZER ZERO	1	1	Renvoi de TRUE pour 0; sinon FALSE.
38	CLICK OFF	1	0	Déclenchement de la fonction "click".
39	CLICK ON	1	0	Enclenchement de la fonction "click".
3A	LOCATE CLOSE	1	0	Ajustage de la position à l'extérieur des limites de défilement.
3B	Validation de sortie de touche d'unité centrale.	1	0	Invalidation d'entrée de touche.
3C	Invalidation de sortie de touche d'unité centrale.	1	0	Validation d'entrée de touche.
3D	Posage du programme de lancement	1	0	Provoque le départ d'exécution du programme lorsque l'alimentation est enclenchée.
3E	Remise à zéro du programme de lancement	1	0	Arrête l'exécution du programme de lancement lorsque l'alimentation est enclenchée.
3F	REQUEST SLEEP	1	0	Posage de l'indicateur SLEEP et coupure de l'alimentation.
40	UDK INIT	1	0	Rétablissement des assignations de défauts à UDK.
41	Ecriture des caractères	9	0	Affichage de 8 octets de données sur la position du curseur et avance du curseur.
42	Lecture de caractères.	1	8	Sortie de 8 octets de données sur la position du curseur.
43	SCANR	3	2	Renvoi du nombre de points posés depuis le point PAINT (X,Y) et RC.
44	SCANL	3	2	Renvoi du nombre de points posés depuis le point PAINT (X-,Y) et RC.
45	TIME CHK	1	1	Renvoi de TRUE si le résultat du contrôle des données de temps est 0.
46	ALM CHK	1	1	Renvoi de TRUE si le résultat du contrôle des données d'alarme est 0.

Tableau 5 (c)

4.2.2 Explication des commandes d'unité centrale secondaire

TIME CALL (01)

Cette commande renvoie les données de temps en format binaire, selon la séquence suivante: Année, Mois, Date, Jour de la semaine, Heure, Minute, Seconde

Un octet est renvoyé pour chaque article, l'exception de l'année qui se compose de deux octets (octet supérieur en premier, octet inférieur en second).

Le jour de la semaine est représenté par des bits de 6 à 0 pour les jours de dimanche à samedi. Le bit correspondant au jour de la semaine est remis à 0.

L'heure est représentée selon un cycle de 24 heures.

L'exemple ci-dessous indique les données envoyées lorsque la commande TIME CALL (01) est générée.

(Exemple 8) 01, 07,BF, 01, 02, BF, OC, 03, 1E

Année mois date jour heure minute seconde

Les données indiquées ci-dessus correspondent à: Dimanche, Janvier 1, 1983, 12:03:30.

STICK (02)

Lorsque cette commande est générée, la touche de curseur enfoncee est renvoyée à ANS.

←	↓	→	↑	ANS
0	0	0	0	30
0	0	0	1	31
0	0	1	0	33
0	0	1	1	32
0	1	0	0	33
0	1	0	1	35
0	1	1	0	34
0	1	1	1	33

←	↓	→	↑	ANS
1	0	0	0	37
1	0	0	1	38
1	0	1	0	30
1	0	1	1	31
1	1	0	0	36
1	1	0	1	37
1	1	1	0	35
1	1	1	1	30

Tableau 6

Les codes 30 à 38 correspondent aux codes ASCII 0 à 8. Ces codes indiquent la direction.

RAM READ (05)

Cette commande permet à l'unité centrale principale de lire l'espace mémoire de l'unité centrale secondaire. Lorsque les octets hauts et bas de l'adresse (dans cet ordre) sont spécifiés après la commande, le contenu de cette adresse est renvoyé vers l'accumulateur.

(Exemple 9) 05,06,C0—Lit l'adresse C006H de l'unité centrale secondaire et introduit le contenu de cette adresse dans l'accumulateur.

RAM WRITE (06)

Cette commande permet à l'unité centrale principale d'écrire des données dans l'espace mémoire de l'unité centrale secondaire. Lorsque les données et les octets hauts et bas de l'adresse (dans cet ordre) sont spécifiés après la commande, les données sont écrites à l'adresse spécifiée de l'unité centrale secondaire.

(Exemple 10) 06,01,1A,C0—Ecrit 01 l'adresse mémoire C01A de l'unité centrale secondaire.

SCROLL SET (07)

Cette commande pose l'étendue du défilement entre 0 et 3. La ligne de départ et la ligne de terminaison sont envoyées après la commande; seuls les deux bits de valeur inférieure envoyés sont effectifs.

TIMESET (0A)

Cette commande régle le temps. Les éléments du temps (seconde, minute, heure, jour, date, mois et année) sont envoyés séquentiellement après la commande. L'année est spécifiée par deux octets (octet inférieur en premier, octet supérieur en second). Le temps est représenté au moyen d'un cycle de 12 heures si la commande **[45]** est envoyée après le réglage du bit 6 de la donnée d'heure.

Calcul et pose du jour de la semaine (0B)

Cette commande calcule et pose le jour de la semaine correspondant à la date réglée en utilisant la commande TIME SET. Les bits 6 à 0 de l'octet qui indique le jour de la semaine sont mise à 0 l'un après l'autre pour indiquer les jours de dimanche à samedi; le bit 7 est toujours à 0.

Pose de la donnée d'alarme (OC)

Cette commande pose la donnée d'alarme. Elle est introduite de la même manière que la commande **[0A]**. Le deuxième paramètre doit être spécifié, mais il n'a pas de signification. Pour le jour de la semaine, tous les bits à l'exception du bit 7 et les bits correspondants aux jours d'alarme doivent être à "1". Si le bit le plus significatif d'un caractère est mis à "1", ce paramètre est ignoré.

(Exemple 11) Réglage de l'alarme pour qu'elle retentisse à 8:00 heure du matin chaque semaine du lundi au vendredi.

[OC], 00, 00, 08, 41, 80, 80, 82, F5

Seconde minute heure jour date mois année

Réglage de bit pour
jour de la semaine

Année, mois et jour sont ignorés
car le bit le plus significatif est "1".

01000001



Le code 01 doit être écrit à l'adresse C01AH du secteur d'unité central secondaire afin de valider la donnée d'alarme.

La sonnerie ou non du vibreur pendant l'alarme peut aussi être spécifiée par la commande indiquée dans l'exemple ci-dessous.

[06], 01, 1A, C0, [OE]

Commande RAM WRITE

Commande de marche du vibreur

Transfert BIT IMAGE (0F)

Cette commande sort la ligne spécifiée de l'écran LCD sous format de bit image (120 octets). L'opérateur se compose d'un numéro de ligne compris entre 0 et 3, et il est spécifié après la commande.

POINT (10)

TRUE est renvoyé en réponse à cette commande si un point se trouve aux coordonnées (X,Y) de l'écran LCD; sinon, FALSE est renvoyé. Les coordonnées X et Y sont spécifiées après la commande et le résultat est renvoyé sous forme d'octet unique.

X doit être spécifié en tant que nombre dans la plage de 0 à 79, et Y doit être spécifié en tant que nombre compris dans la plage de 0 à 31.

PSET (11)

Cette commande pose le point aux coordonnées (X,Y) spécifiées de l'écran. La spécification est la même que pour la commande **[10]**.

PRESET (12)

Cette commande remet à zéro le point des coordonnées (X,Y) spécifiées de l'écran. La spécification est la même que pour la commande **[10]**.

PEOR (13)

Cette commande inverse la pose du point sur les coordonnées (X,Y) spécifiées de l'écran. La spécification est la même que pour la commande **[10]**.

LINE (14)

Cette commande pose tous les points d'un ligne entre les coordonnées (X_1, Y_1) et (X_2, Y_2). Les coordonnées sont spécifiées selon la séquence X_1, Y_1, X_2, Y_2 après la commande.

CIRCLE (15)

Cette commande trace un cercle de rayon r autour du centre de coordonnées (X_0, Y_0). Les paramètres sont spécifiés selon la séquence X_0, Y_0, r , après la commande.

USER DEFINE KEY WRITE (16)

Cette commande assigne les définitions aux touches de fonction définies par l'utilisateur. Les paramètres spécifiés après la commande sont (dans l'ordre de spécification) le nombre UDK (nombre de 1 à 12), la chaîne de caractères devant être assignée à UDK et une terminaison fictive ("00"). Les trois premiers caractères sont utilisés en tant que labels de touche, et les caractères débutant au quatrième sont effectifs. La chaîne de caractères peut se composer de 1 à 42 caractères (y compris les trois caractères de labels et la terminaison fictive), un maximum de 38 d'entre eux étant effectifs.

(Note) Prendre soin d'éviter une entrée continue, car cela corromprait la définition des touches suivantes. En outre, un maximum de 42 caractères effectifs sont possibles avec les touches F6 et F12.

USER DEFINE KEY READ (17)

Cette commande est utilisée par l'unité centrale afin de lire les touches définies par l'utilisateur. Lorsque le nombre UDK est introduit après la commande, un maximum de 255 caractères de l'adresse de départ des touches vers la terminaison fictive ("00") sont renvoyés.

USER DEFINE CHARACTER WRITE (1A)

Cette commande permet de définir des caractères qui sont affichés pour les codes de caractères 80 à 9F et E0 à FF. Introduire le code de caractères et 8 octets de données de caractères (6 points horizontalement × 8 points verticalement) après la commande.

(Exemple 12) **1A**, 80, 20, F0, 28, 20, 30, 58, C8, 00
↑
Code de caractères

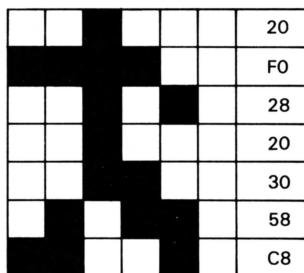


Figure 11

USER DEFINE CHARACTER READ (1B)

Cette commande est utilisée pour lire les définitions de caractères réalisées à l'aide de la commande 1A. Lorsque le code de caractères est introduit, la donnée CG est renvoyée après cette commande. Les codes de caractères doivent se situer dans la plage de 20H à FFH.

START PROGRAM WRITE (1D)

Cette commande définit un programme de lancement pouvant atteindre 511 octets. Le début du programme est posé à l'adresse de départ lorsque les données de chaîne et une terminaison ("00") sont introduites après la commande. Le programme de lancement est exécuté lors de l'enclenchement de l'alimentation après une demande OFF, et il a priorité sur les entrées de touches.

(Exemples 13) **[1D],00** ... Annulation du programme de lancement

[1E], 52, 55, 4E, 0D

↑ ↑ ↑ ↑
R U N CR

SP WRITE CONT. (1E)

Cette commande est écrite après la commande ID (voir l'exemple ci-dessus).

SPON (1F)

Cette commande permet l'exécution du programme de lancement. Lors de l'exécution de cette commande, le programme de lancement a priorité sur UDK et sur les autres entrées de touches. L'exécution du programme de lancement est invalidée par la commande **[20]**.

LOCATE (24)

Cette commande déplace le curseur sur la position spécifiée et affiche un caractère. Si 00 est spécifié en tant que code de caractères, le curseur est déplacé mais aucun caractère n'est affiché. Introduire les coordonnées X et Y et le code de caractères (dans cet ordre) après la commande. X doit se trouver dans la plage de 0 à 19 et Y dans la plage de 0 à 3.

(Exemple 14) **[24], 01, 05, 41**

Affiche le caractère "A" à la cinquième colonne de la première rangée.

Affichage du curseur (25)

Cette commande allume l'affichage du curseur. Normalement, le curseur ne devrait être affiché que si le système attend une entrée.

Non affichage du curseur (26)

Cette commande éteint l'affichage du curseur. La vitesse d'affichage est approximativement deux fois supérieure lorsque le curseur n'est pas affiché.

TEST KEY (27)

Lorsque la ligne d'échantillonnage de touches est spécifiée (en tant que nombre à 2 octets) après cette commande, la valeur d'échantillonnage de touches OR est renvoyée. Voir le tableau d'adresses d'essai de touches à la page 70.

(Exemple 15) Si 01 est renvoyé lors de l'introduction de **[27]**, 00,04, la touche "A" est enfoncée.

TEST CHR (28)

TRUE est renvoyé en réponse à cette commande si la touche du caractère envoyé est enfoncée; sinon, FALSE est renvoyé. Le caractère spécifié doit être un caractère pouvant être introduit dans le mode alphanumérique sans utiliser la touche SHIFT. La raison réside dans le fait que cette commande réalise son contrôle sans tenir compte de l'état des touches SHIFT ou GRPH.

Effacement de l'affichage (2B)

Cette commande coupe le signal commun du LCD. La MEV d'affichage peut être lue ou écrite, mais son contenu ne sera pas affiché.

Allumage de l'affichage (2C)

Cette commande rétablit l'affichage de LCD après l'exécution de la commande 2B.

UDK CONT WRITE (35)

Cette commande ajoute des données à la fin de la chaîne de caractères actuellement assignée à l'une des touches de fonction définies par l'utilisateur. Spécifier le nombre UDK, les données de chaîne et "00" après la commande.

ALARM READ (36)

Cette commande lit les réglages d'alarme; son format est le même que celui de la commande **[01]**.

LOCATE CLOSE (3A)

Lorsque le curseur a été déplacé à l'extérieur des limites de défilement à l'aide de la commande **[24]**, cette commande ramène le curseur dans les limites de défilement. Si le curseur se trouve au-dessus de la limite de défilement, il est déplacé vers l'extrémité de gauche de la ligne supérieure de la limite; s'il se trouve sous la limite de défilement, il est déplacé vers l'extrémité de gauche de la ligne inférieure de la limite.

Ecriture de caractère (41)

Cette commande affiche le motif de 6 × 8 points spécifié par l'utilisateur sur la position du curseur, puis elle fait avancer le curseur sur la position suivante. Le motif est spécifié au moyen de 8 octets de données suivant la commande; le format est le même que celui de la commande **[1A]**.

SCANR (43)

Cette commande renvoie 8 bits de données indiquant les réglages des points dans la zone 6×8 point sur l'emplacement du curseur. Le format des données est le même que celui de la commande **[1B]**

Lecture de caractères (42)

Ce sous-programme est utilisé par la commande PRINT. Il pose des points à la droite de la position (X, Y) jusqu'à ce qu'un point posé précédemment soit atteint ou que les points soient à l'extérieur de la limite. Le code de renvoi se compose de 2 octets, comme suit:

Premier octet: nombre de points posés

Deuxième octet: "00" - Opération terminée lorsqu'un point posé précédemment est atteint.

 "40" - Opération terminée lors de la pose d'un point à l'extérieur de l'écran.

 "80" - Un point était déjà posé à la position (X, Y).

SCANL (44)

Cette commande est utilisée conjointement avec la commande 43. Elle pose les points à la gauche de la position (X-1, Y). Les codes de renvoi sont les mêmes que ceux de la commande **[43]**

TIME CHK (45)

Cette commande contrôle si les réglages de date et d'heure actuelle se trouvent à l'intérieur des plages admissibles, et ils règlent l'affichage de l'heure selon un cycle de 24 heures.

Plage admissible	Etat d'erreur
$0 \leq \text{seconde} < 60$	Bit 0 = 1
$0 \leq \text{minute} < 60$	Bit 1 = 1
$0 \leq \text{heure} < 24$	Bit 2 = 1
$0 \leq \text{jour} \leq 28$	Bit 3 = 1
29	(Pour $1901 \leq \text{année} \leq 2099$)
30	
31	
$0 \leq \text{mois} \leq 12$	Bit 4 = 1
$1901 \leq \text{année} \leq 2099$	Bit 5 = 1, Bits 6 et 7 = 0

ALM CHK (46)

Cette commande contrôle les réglages d'alarme. (Aucune donnée n'est renvoyée pour les secondes). Une erreur apparaît lors de la pose du bit le plus significatif.

4.2.3 Etat implicite de l'unité centrale secondaire

Article	RESET	OFF KEY ON	SLEEP ON
Alarme	DISABLE		
Effacement d'écran	Effacé	Effacé	Effacé
Position du curseur	(0, 0)		
Affichage du curseur	Allumé		
Vidage du registre tampon de touches	Effacé	Effacé	Effacé
Répétition de touches	Allumé	Allumé	Allumé
Sortie des touches d'unité centrale	Validé		
UDK INIT	Initialisé		
UDC INIT	Initialisé		
UDK	Fermé	Fermé	Fermé
Programme de lancement	OFF	Ouvert	OFF
Etendue du défilement	0 ~ 3		
Vibreur	OFF		
Bit SLEEP	0	0	1
ONSTATE	0	1	1
Mode de caractères		Alphanumérique	Alphanumérique
TIME	00:00:00, Janvier 1, 2001		

Tableau 7

4.2.4 Secteur MEV d'unité centrale secondaire

(a) MEV du système d'unité centrale secondaire (C000H à C0FFH)

Adresse d'unité centrale secondaire	Contenu ouvert à l'utilisateur
C006H	Bit 0 = ONSTATE, Bit 6 = SLEEP, Bit 7 = ALARM simulé
C00CH	Mode de touches (00 = normal, 01 = "kana", 02 = graphique, 03 = numérique)
C01AH	Validation d'alarme si bit 0 = 1.
C040H C04EH	Seconde, minute, heure, jour de la semaine, date, mois et année (mémorisés dans cet ordre chaque deux octets)
C050H C05EH	Donnée d'alarme
C080H	Indicateur de curseur
C082H	Indicateur de touche définie par l'utilisateur
C084H	Compteur de vibreur (le vibreur cesse de retentir lorsque l'indicateur de report est posé)
C092H	Compteur de la minuterie d'alimentation
C096H	Fin de défilement
C09EH	Début de défilement

Tableau 8

(b) Adresses des touches définies par l'utilisateur (C800H à C9FFH)

UDK	Adresse MEV d'unité centrale secondaire
F1	C800H
F2	C82AH
F3	C854H
F4	C87EH
F5	C8A8H
F6	C8D2H
F7	C900H
F8	C92AH
F9	C954H
F10	C97EH
F11	C9A8H
F12	C9D2H

Tableau 9

(c) Adresses des caractères définis par l'utilisateur (CA00H à CBFFH)
Adresses pour les codes ASCII 80 à 9F

CA00H + (ASC-80) × 8

Adresses pour les codes ASCII E0 à FF

CB00H + (ASC-E0) × 8

(d) Adresse du registre tampon d'entrée de touche (CC00H à CCFFH)

127 octets

Adresse du registre tampon	Contenu
C086H	Indicateur de fin
C088H	Adresse de départ

(e) Adresse du programme de lancement (CE00H à CFFFH)

512 octets

L'adresse C092H indique les 8 bits inférieurs de l'adresse du programme de lancement actuel, et le bit 0 de C008H indique le bit 8 de l'indicateur.

4.2.5 Matrice de clavier

SFT	CTR								A1	0200H
HOME		GRP	NUM	OFF			ON		BZ	0100H
L	+	*	})	■	:	=	~		S8 0080H
>	?	●	¥	-	CR	O	P	.	{	S7 0040H
.	/	X	?				@	„	c	
F 1	F 2	F 3	F 4	F 5	F 6					S6 0020H
!	"	#	\$	%	&	'	(S5 0010H
I	2	3	4	5	6	7	8			
Q	W	E	R	T	Y	U	I			S4 0008H
A	S	D	F	G	H	J	K			S3 0004H
Z	X	C	V	B	N	M	.			S2 0002H
INS	DEL	→	—	↑	↓	SPC				S1 0001H
K1	K2	K3	K4	K5	K6	K7	K8			

Tableau 10

Voir les explications de la commande TEST KEY à la page 64.

5. SEQUENCE D'ENCLENCHEMENT/ DECLENCHEMENT D'ALIMENTATION DU X-07

5.1 Séquence d'enclenchement d'alimentation

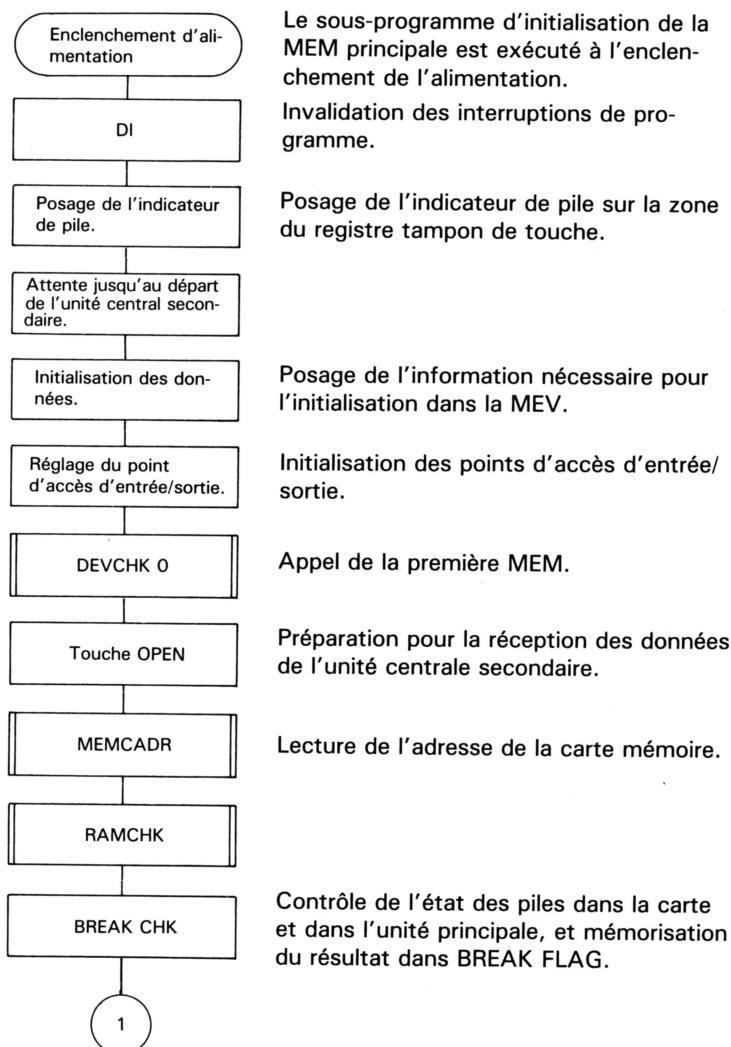


Figure 12 (a)

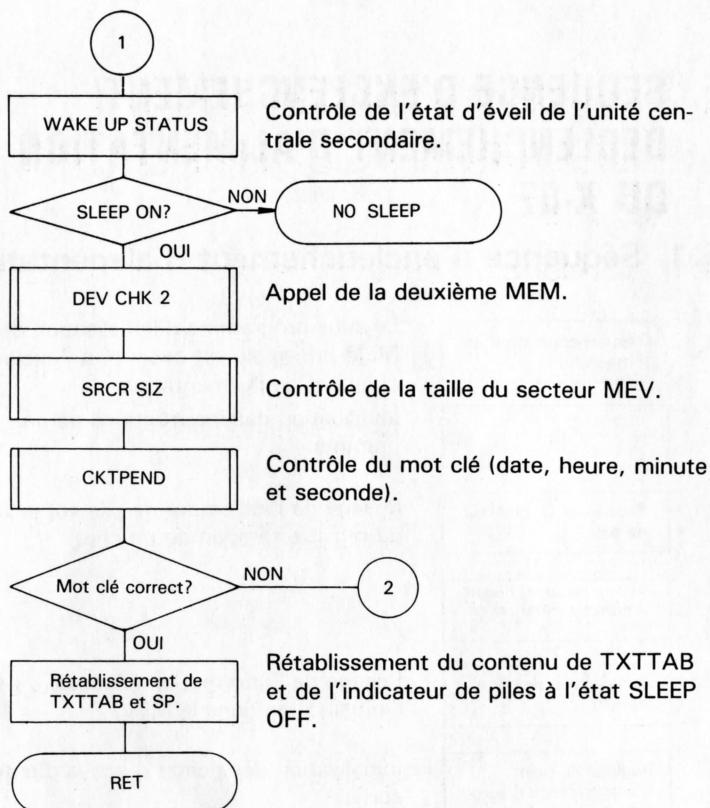


Figure 12 (b)

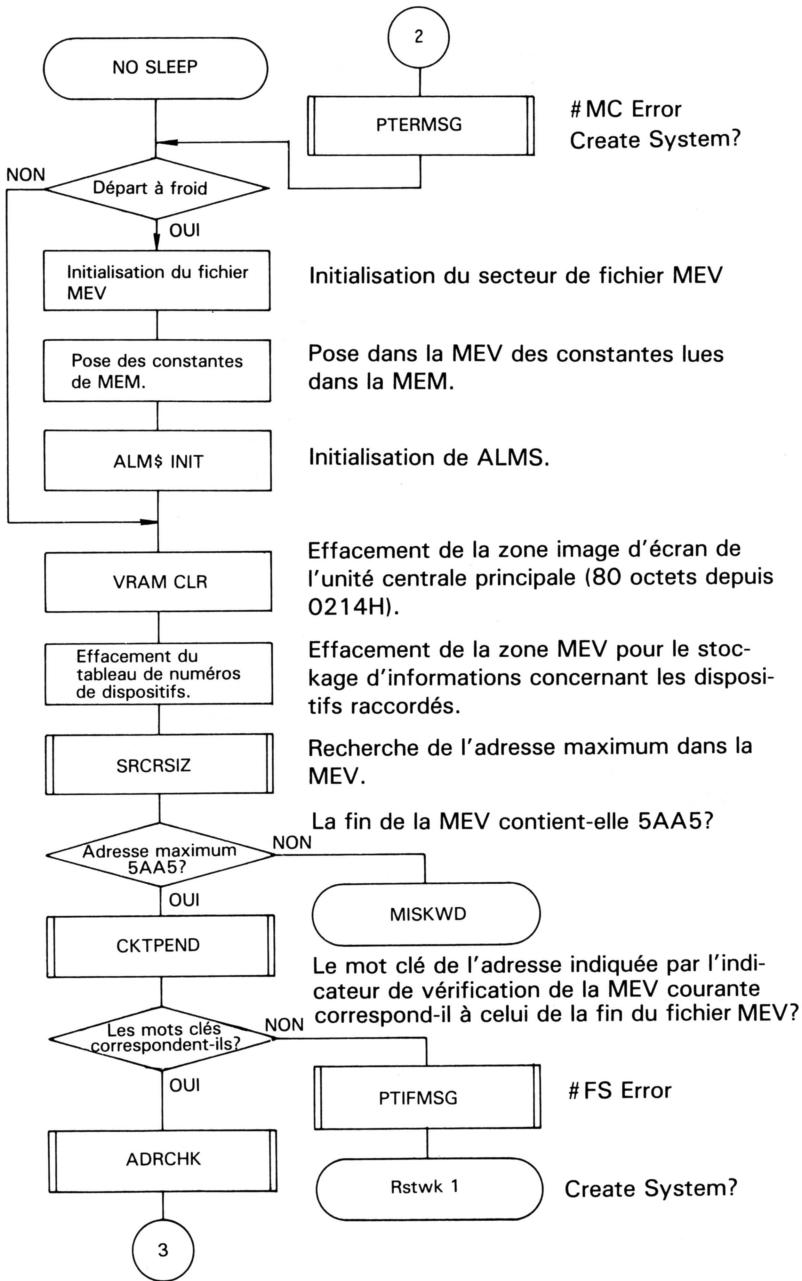


Figure 12 (c)

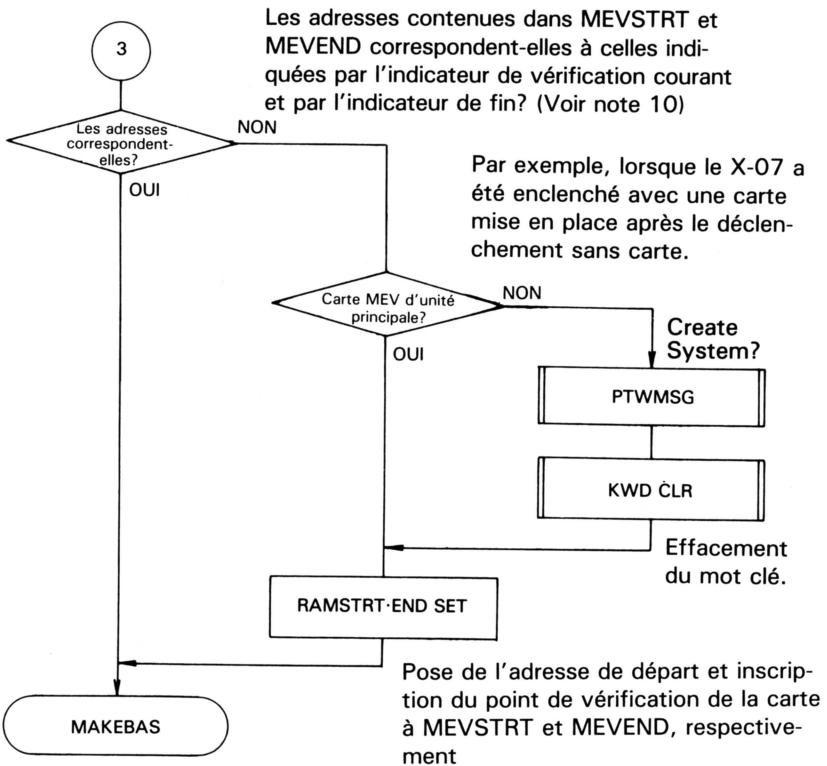


Figure 12 (d)

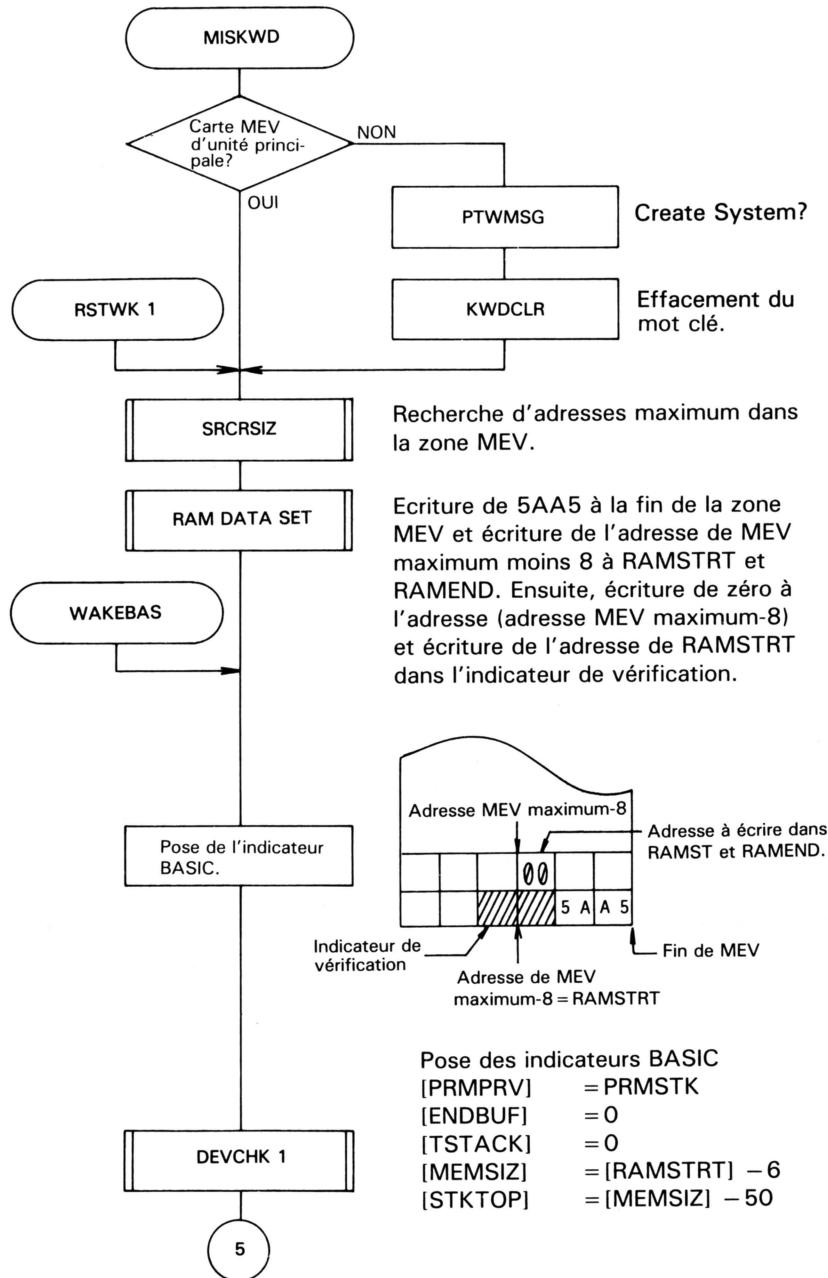


Figure 12 (e)

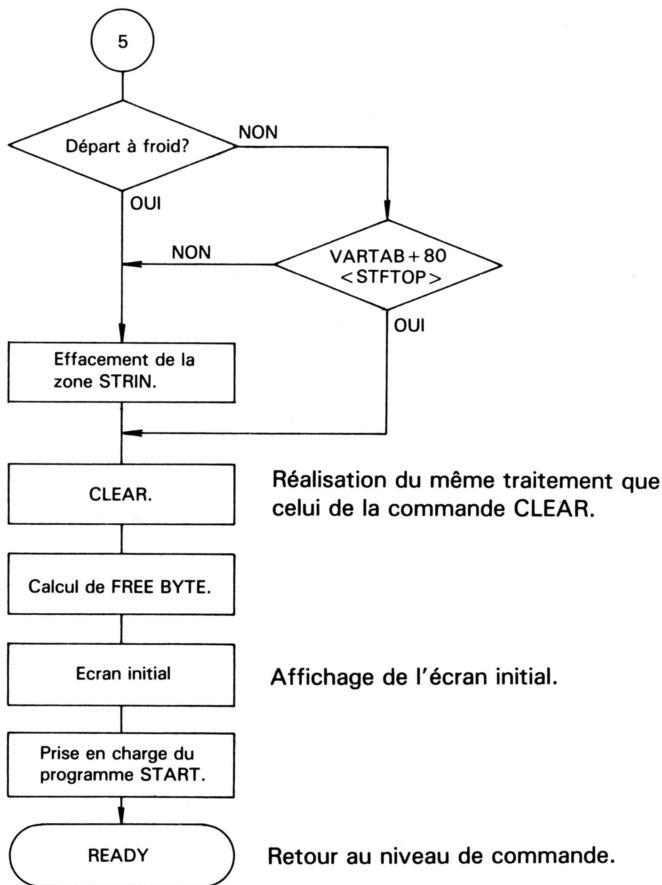


Figure 12 (f)

- (Note 10)** Lorsque le contenu de l'indicateur de vérification actuel et celui de l'indicateur de fin sont supérieurs à ceux de MEVSTRT et MEVEND de la MEV d'unité principale, le message "Create system?" est affiché pour demander à l'opérateur d'effacer le mot clé.
- Le mot clé doit être effacé pour la raison suivante: Considérons que des fichiers aient été créés avec la MEV principale uniquement. Si une carte fichier est utilisée pour poser une nouvelle zone fichier, les fichiers créés précédemment sur la MEV principale seront détruits, car la zone MEV pour ces fichiers est utilisée pour des piles, etc.
- Si le X-07 est alors utilisé sans carte fichier, le mot clé correspondra et le système fonctionnera (bien qu'incorrectement) car le contenu de la zone fichier a été détruit.

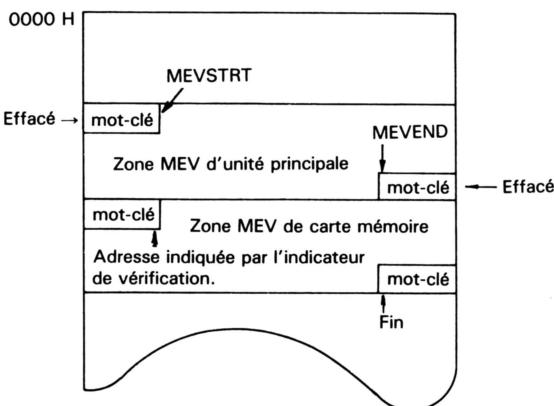


Figure 13

DEVCHK

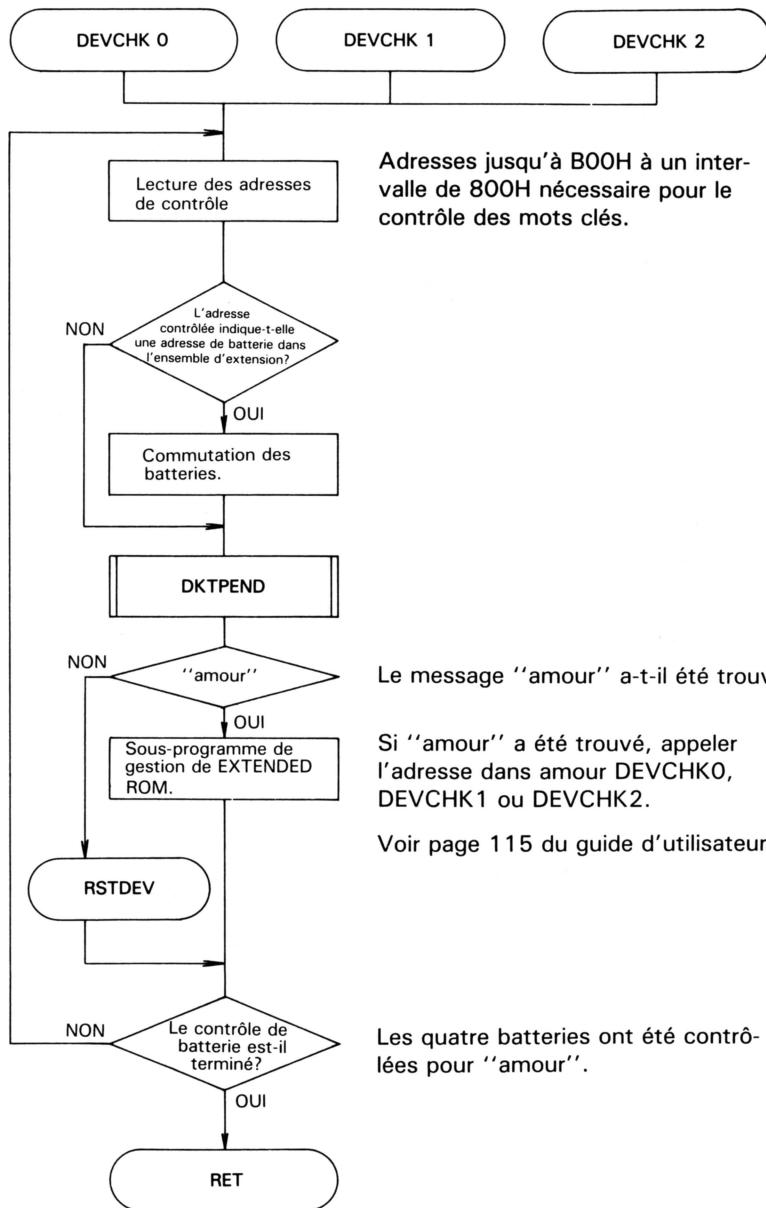
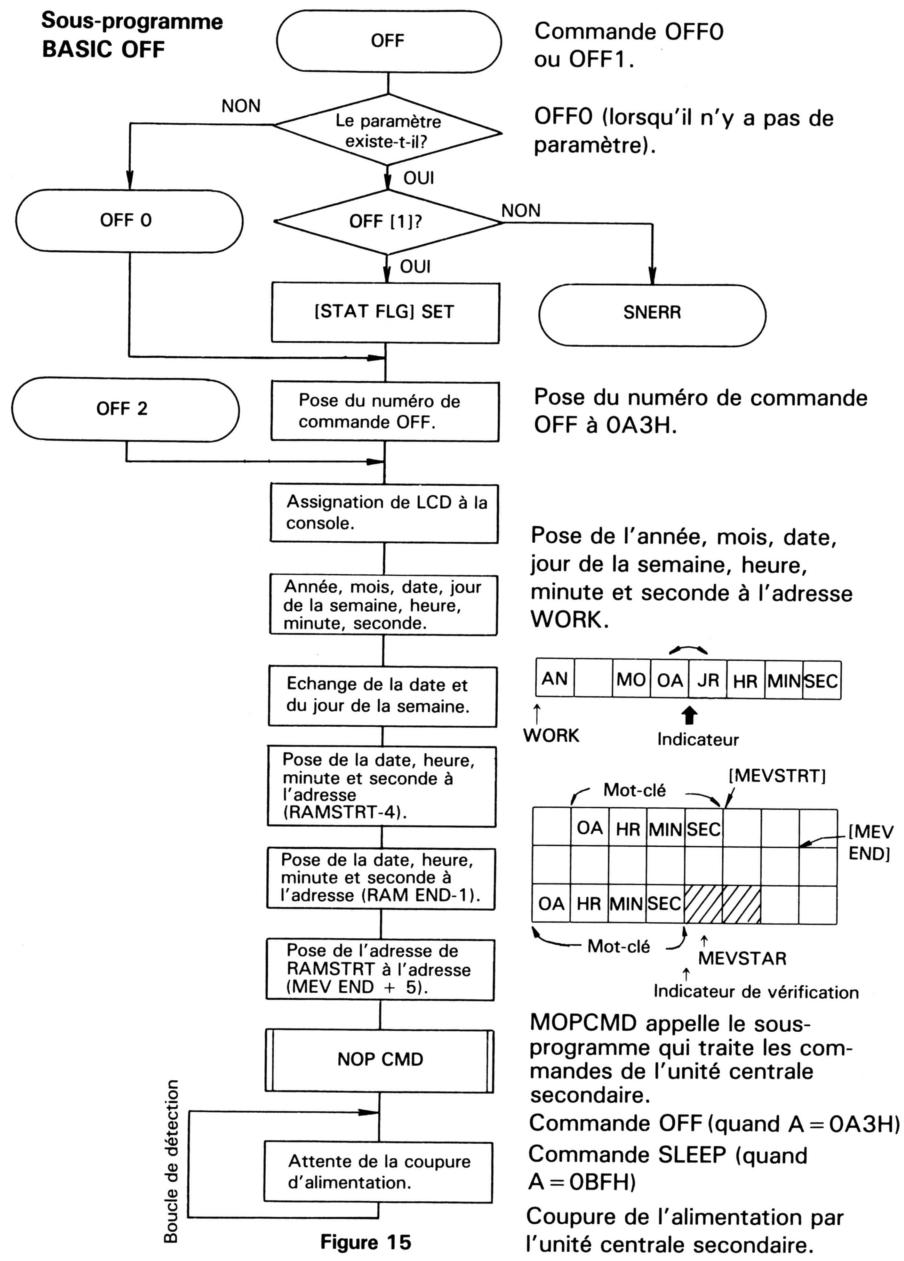


Figure 14

5.2 Séquence de déclenchement d'alimentation



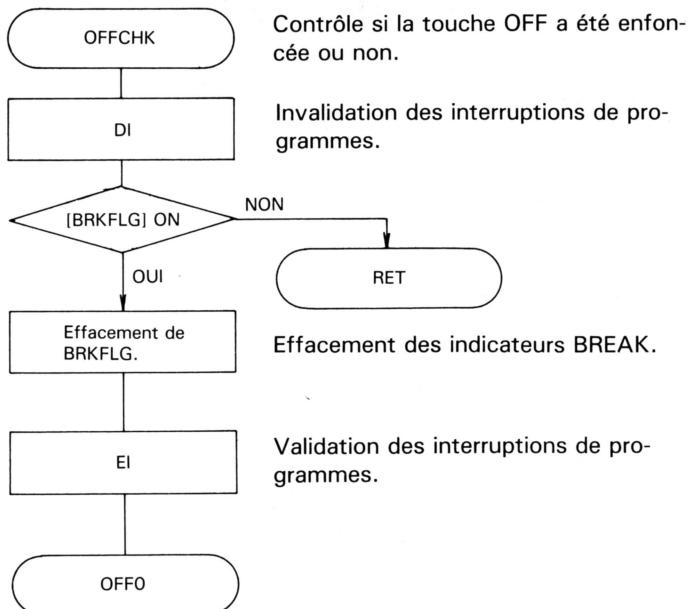


Figure 16

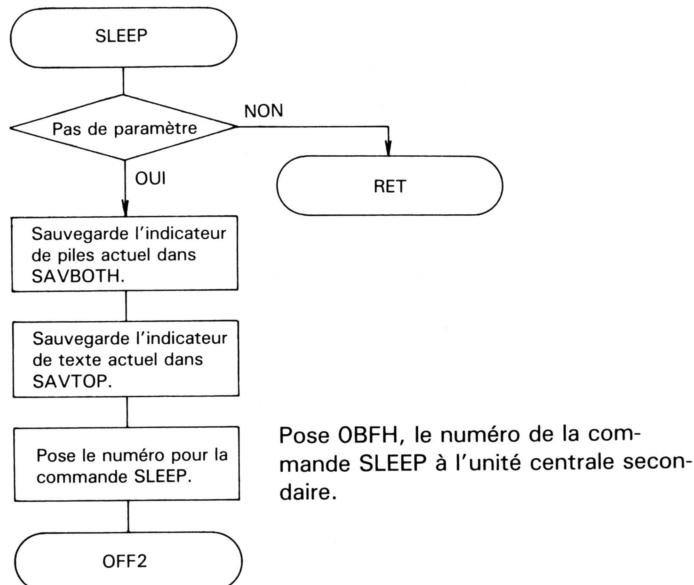


Figure 17

6. EXEMPLE DE CODAGE DE PROGRAMME EN LANGAGE MACHINE

6.1 Traitement général des entrées

(Lorsque la zone de programme en langage machine risque d'être détruite par le système BASIC).

“amour” et les trois adresses d'entrée suivantes agissent en tant qu'interfaces entre le système BASIC et la MEM.

ENTRY 0

L'adresse d'entrée qui est appelée immédiatement après l'initialisation de la zone de travail minimum et des étapes de contrôle d'entrée/sortie. Lors de cet état, les interruptions sont masquées.

ENTRY 1

Adresse de l'entrée qui est appelée immédiatement avant que le système BASIC ne produise le message de copyright. Ce qui suit doit être réalisé avant de renvoyer la commande au système BASIC.

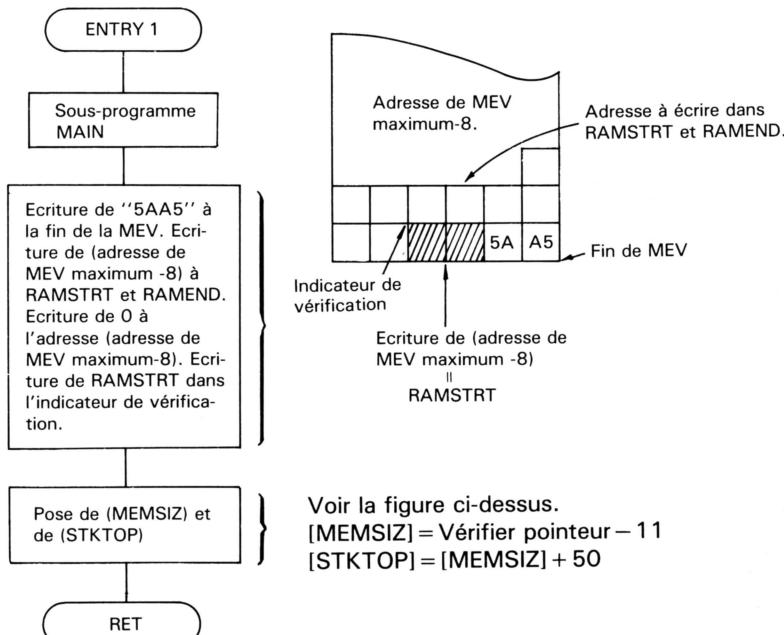
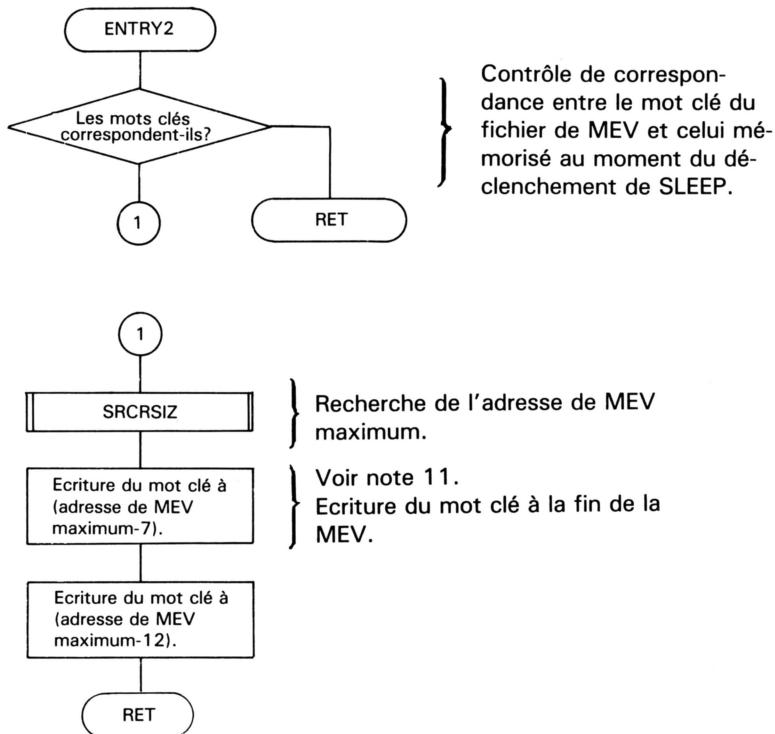


Figure 18

Voir page 113 du guide d'utilisateur.

ENTRY 2

Adresse de l'entrée immédiatement après le lancement depuis l'état SLEEP.



Note 11:

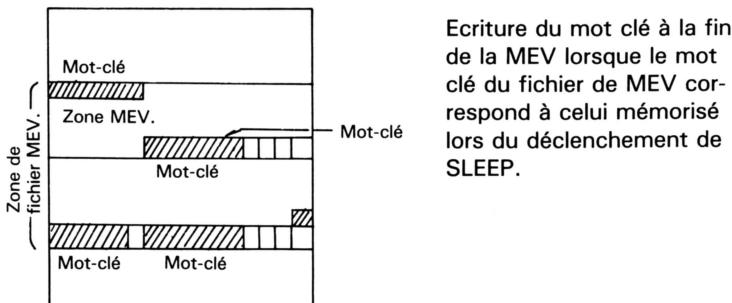


Figure 19

(Exemple 16)

	1	"Z80" LIST		
0000 6C6F7665	2	DEFB	'amour'	
0004 000A	3	DEFW	ENTRY0	enDEVCHK0
0006 000B	4	DEFW	ENTRY1	enDEVCHK1
0008 0036	5	DEFW	ENTRY2	enDEVCHK2
	6	EXT	RAMSTR, RAMEND, STKTOP, MEMSIZ, KEYWORD,	
	7;		CKTPEND, MOVE, SRCRSIZ	
000A C9	8	ENTRY0:	RET	Adresse d'entrée appelée
	9;			immédiatement après l'initialisation de la zone de travail
000B	10	ENTRY1:		minimum et des étapes de
	11;			contrôle d'entrée/sortie.
	12;			
	13;			
	14;			
	15;			
	16;			
	17;			
000B CD00000	18	CALL	SRCRSIZ	
000E 3EA5	19	LD	A, 0A5H	Recherche de la fin de la MEV.
0010 77	20	LD	[HL], A	Ecriture de "5AA5" à la fin
0011 2F	21	CPL		de la MEV.
0012 2B	22	DEC	HL	Adresse de MEV maximum.
0013 77	23	LD	[HL], A	
0014 E5	24	PUSH	HL	
0015 1FFF9	25	LD	DE, -7	[RAMSTR] = [RAMEND] = Fin
0018 19	26	ADD	HL, DE	de MEV-7
0019 220000	27	LD	[RAMSTR], HL	
001C 220000	28	LD	[RAMEND], HL	
001F AF	29	XOR	A	[Fin de MEV-7] = 0
0020 77	30	LD	[HL], A	Vérifier pointeur
0021 D1	31	POP	DE	[Fin de MEV-3] = Fin de
0022 EB	32	EX	DE, HL	MEV-8
0023 2B	33	DEC	HL	↑
0024 72	34	LD	[HL], D	RAMSTR
0025 2B	35	DEC	HL	
0026 73	36	LD	[HL], E	[MEMSIZ] = Fin de MEV-3-11
0027 1FFF5	37	LD	DE, -11	
002A 19	38	ADD	HL, DE	
002B 220000	39	LD	[MEMSIZ], HL	[STKTOP] = [MEMSIZ] - 50
002E 1FFCE	40	LD	DE, -50	
0031 19	41	ADD	HL, DE	Adresse d'entrée appelée lors
0032 220000	42	LD	[STKTOP], HL	du lancement depuis l'état
0035 C9	43	RET		SLEEP.
	44;			
0036 2A0000	45	ENTRY2:	LD, [RAMEND]	Le mot clé écrit alors que
0039 110005	46	LD	DE, 5	l'état SLEEP était introduit
003C 19	47	ADD	HL, DE	correspond-il à celui actuellement dans le fichier MEV?
003D 110004	48	LD	KEYWORD + 4	
0040 CD0000	49	CALL	CKTPEND	Recherche de la fin de la MEV.
0043 C0	50	RET	NZ	HL = adresse de MEV
0044 CD0000	51	CALL	SRCRSIZ	maximum-7
0047 E5	52	PUSH	HL	
0048 1FFF9	53	LD	DE, -7	Ecriture du mot clé à
004B 19	54	ADD	HL, DE	l'adresse (adresse de HEV
004C 110000	55	LD	DE, KEYWORD	maximum-7)
004F D5	56	PUSH	DE	
0050 CD0000	57	CALL	MOVE	Ecriture du mot clé à
0053 D1	58	POP	DE	l'adresse (adresse de MEV
0054 E1	59	POP	HL	maximum-12)
0055 01FFF4	60	LD	BC, -12	
0058 09	61	ADD	HL, BC	
0059 CD0000	62	CALL	MOVE	
005C C9	63	RET		
	64	END		

Error = 0

6.2 Exemples de traitement d'entrée de Monitor Card

(Exemple 17)

```
1      "Z80"
2      GLB      LDUHL, DATERR, error, MAIN1, REGDISP, ERRMSG, ERRMSG0
3      GLB      TMCODE, PUTHL, comput, MSG, AEXPR, SCAN, SYNERR, POOPHL, CKHEX
4      GLB      NSCAN, XXXXX, Pcomput, PPUTHL, CASC12, CASC11, CRLF, HCNVT
5      GLB      PSUHL, LDUSR, MSG0, SPK, SYNCER, GETRNG, PPUTA, SAVERG, COMAND
6      GLB      CKBRK
7      EXT     RSTHND, SETBRK, CANBRK, BPTOMEM, TRACE, NTRACE, RSCAN, NOPCMD
8      EXT     LINEND, LINWDT, BRKFLG, CONSNS, MEMTOBP, GETADR, WAITSTP
9      EXT     KEYWORD, MOVE, SETCUR, POUT, CSRX

<C579> 10 CKTPEND EQU 0C579H
<C620> 11 SRCRSIZ EQU 0C620H
<0210> 12 RAMSTRT EQU 0210H
<0212> 13 RAMEND EQU 0212H
<000E> 14 BEEPCNT EQU 000EH
<C262> 15 PUTQ    EQU 0C262H
<01DF> 16 MEMSIZ EQU 01DFH
<026C> 17 INTIMAG EQU 026CH
<030C> 18 RSERFLG EQU 030CH
<030B> 19 UCRIMAG EQU 030BH
<C2AA> 20 LFTQ    EQU 0C2AAH
<026D> 21 TERFLG EQU 026DH
<0016> 22 ANSPTR EQU 0016H
<C20F> 23 ANSCNT EQU 020FH
<009F> 24 Scomput EQU 09FH      ;PROGRAMME D'ECRITURE DE CONSOLE
<C805> 25 CONGET EQU 0C8C5H      ;PROGRAMME DE LECTURE DE CONSOLE
<EBFF> 26 PINLIN EQU 0EBFFH      ;COMMANDE DE LECTURE DE L'ECRAN
<00D5> 27 BUF     EQU 0D5H      ;COMMANDE DE PINUN VECTEUR
<003C> 28 INTVEC EQU 03CH      D'INTERRUPTION KBD
<0000> 29 RST0    EQU 0
<C92F> 30 CMDPUT EQU 0C92FH      ;COMMANDE DE LECTURE SUR TOSHIBA
<C799> 31 INTPPT EQU 0C799H      ;VECTEUR D'INTERRUPTION VIEUX 3CH
<01DD> 32 STKTOP EQU 01DDH      ;ADRESSE DE BAS DE MEMOIRE
<C528> 33 Pasure  EQU 0C528H      ;POINT D'ENTREE BASIC
<E906> 34 LDEHL   EQU 0E906H      ;LD-DE-HL
35 ;
36     INCLUDE HEAD
37 ;
```

	39	PROG	
0000 6C6F7665	40	DEFB	'amour'
0004 000A	41	DEFW	ENTRY0
0006 0070	42	DEFW	ENTRY1
0008 00EB	43	DEFW	ENTRY2
	44	EXT	ENT1
000A CD0000	45	ENTRY0	ENT1 ;Appel programme BASIC
000D 2101E9	46	LD	HL, KBDINT d'extension initial
0010 22003G	47	LD	[INTVEC+1], HL ;CHANGEMENT VECTEUR
0013 210000	48	LD	HL, RSTHND D'INTERRUPTION KBD
0016 220001	49	LD	[RSTO+1], HL ;Réglage de l'adresse RSTO
0019 3EC3	50	LD	A, Jump_code
001B 320000	51	LD	[RSTO], A
001E 32054D	52	LD	[MONFLG], A
	53 ::::		
	54	EXT	DEVHOCK, DEVSCN
0021 320001	55	LD	[DEVHOCK+1], A
0024 210000	56	LD	HL, DEVSCN
0027 220002	57	LD	[DEVHOCK+2], HL
002A CDC620	58	CALL	SRCRSIZ
002D E5	59	PUSH	HL
002E CD0056	60	CALL	MAKEF
0031 E1	61	POP	HL
0032 E5	62	PUSH	HL
0033 11FBF8	63	LD	DE, -1032 ;1024 + 8
0036 19	64	ADD	HL, DE
0037 ED5B0212	65	LD	DE, [RAMEND]
003B E7	66	RST	20H ;COMPAR
003C E1	67	POP	HL
003D 280C	68	JR	Z, Ent0
003F E5	69	PUSH	HL
0040 11F004	70	LD	DE, -4096 + 4
0043 19	71	ADD	HL, DE
0044 ED5B0210	72	LD	DE, [RAMSTRT]
0048 E7	73	RST	20H ;COMPAR
0049 E1	74	POP	HL
004A D0	75	RET	NC
004B 11FFF8	76	Ent0	Si il y a un fichier sur une Carte de 4K, ne pas toucher le pointeur.
004E 19	77	ADD	DE, -8
004F 220210	78	LD	[RAMSTRT], HL
0052 220212	79	LD	[RAMEND], HL
0055 C9	80	RET	
0056 3EA5	81	MAKEF:	LD A, 0A5H] Ecriture de "5AA5" à la fin
0058 77	82	LD	[HL], A de la MEV.
0059 2B	83	DEC	HL
005A 2F	84	CPL	
005B 77	85	LD	[HL], A] Ecriture de zéros sur les 10
005C 2B	86	DEC	octets en dessous de
005D 2B	87	DEC	(adresse de MEV
005E E5	88	PUSH	maximum-3).
005F AF	89	XOR	
0060 060A	90	LD	B, 10
0062 77	91	Clr	[HL], A] Ecriture de zéros sur les 10
0063 2B	92	DEC	octets en dessous de
0064 10FC	93	DJNZ	(adresse de MEV
0066 50	94	LD	maximum-3).
0067 1E05	95	LD	E, 5

0069 19	96	ADD	HL, DE	HL = BOTOM - 13 + 5
006A D1	97	POP	DE	DE = BOTOM - 3
006B EB	98	EX	DE, HL	
006C 73	99	LD	[HL], E	
006D 23	100	INC	HL	
006E 72	101	LD	[HL], D	
006F C9	102	RET		
	103			
0070 CDC620	104 <u>ENTRY1</u>	CALL	SRCRSIZ	
0073 11FC00	105	LD	DE, - 1024	Le Monitor a besoin de 1024 octets.
0076 19	106	ADD	HL, DE	
0077 3EAS	107	LD	A, 0ASH	
0079 BE	108	CP	[HL]	Ecriture de "5AA5" dans
007A 2B	109	DEC	HL	(BOTOM-1024)? sinon, aller
007B 201A	110	JR	NZ, Mis	à MIS.
007D 2F	111	CPL		
007E BE	112	CP	[HL]	
007F 2016	113	JR	NZ, Mis	
0081 E5	114	PUSH	HL	
0082 2B	115	DEC	HL	Indicateur de vérification de
0083 56	116	LD	D, [HL]	charge à DE.
0084 2B	117	DEC	HL	Comparaison de 4 octets du
0085 5E	118	LD	E, [HL]	clavier.
0086 D5	119	PUSH	DE	
0087 CDC579	120	CALL	CKTPEND	
008A D1	121	POP	DE	
008B E1	122	POP	HL	
008C 2009	123	JR	NZ, Mis	
008E 01FF99	124	LD	BC, - 7	Si NZ, erreur de fichier.
0091 09	125	ADD	HL, BC	Si Z, fichier normal.
0092 220212	126	LD	[RAMEND], HL	↓(BOTOM - 1024 - 8) - (RAMEND)
0095 1808	127	JR	OKfile	
0097 23	128 Mis	INC	HL	
0098 CD0056	129	CALL	MAKEF	1024 à la zone fichier.
009B ED530212	130	LD	[RAMEND], DE	
009F ED530210	131 Okfile	LD	[RAMSTRT], DE	
00A3 CDC620	132	CALL	SRCRSIZ	
00A6 11FFD0	133	LD	DE, - 48	----- < == = RAM MAX
00A9 19	134	ADD	HL, DE	35 + 13
00AA 22054E	135	LD	[WORK], HL	----- < == = [WORK]
00AD 11FFF0	136	LD	DE, - 16	16
00B0 19	137	ADD	HL, DE	----- < == = [BPAREA]
00B1	138	ST__HL__WK	BPARA	100
	139			
00B5 11FE9C	140	LD	DE, - 356	----- LIGNE DE
00B8 19	141	ADD	HL, DE	COMMANDE
00B9	142	ST__HL__WK	CMDLIN	----- 256
	143			----- < == = [CMDLIN]
	144			50 < USERSTK >
00BD 3E0D	145	LD	A, CR	----- ECHAPPEMENT
00BF 77	146	LD	[HL], A	TAMPON
00C0 11FECE	147	LD	DE, - 306	----- < == = [BUFESC]/[USP]
00C3 19	148	ADD	HL, DE	310-13
00C4	149	ST__HL__WK	USP	----- < == = [UPC]
00C8	150	ST__HL__WK	BUFESC	
00CC 11FED7	151	LD	DE, - 297	
00CF 19	152	ADD	HL, DE	

Réaliser le traitement nécessaire pour la zone
séparée de la zone BASIC.

TRAVAIL	35 + 13
POINT DE RUPTURE	16
PILE SYS	< == = [BPAREA]
LIGNE DE COMMANDE	256
PILE	< == = [CMDLIN]
UTILISATEUR	50 < USERSTK >
ECHAPPEMENT TAMPON	256
pile utilisateur	< == = [BUFESC]/[USP]
ZONE BASIC	310-13
	< == = [UPC]

00D0	153	ST__HL_WK	UPC	
00D4 2A0210	154	LD	HL, [RAMSTRT]	
00D7 11FFFA	155	LD	DE, -6	
00DA 19	156	ADD	HL, DE	
00DB 2201DF	157	LD	[MEMSIZ], HL	
00DE 11FFCE	158	LD	DE, -50	
00E1 19	159	ADD	HL, DE	
00E2 2201DD	160	LD	[STKTOP], HL	
00E5 CD011A	161	CALL	SETSYSK	;INITIALISATION SYSSTK
	162 :::::::			
00E8 C30000	163	EXT	CLRBP	
	164	JP	CLRBP	
	165 :			
00EB CDC620	166	ENTRY2:	CALL	SRCRSIZ
00EE E5	167	PUSH	HL	
00EF 11FBFD	168	LD	DE, -1032+5	
00F2 19	169	ADD	HL, DE	
00F3 110004	170	LD	DE, KEYWORD+4	
00F6 CDC579	171	CALL	CKTPEND	;Vérifier le mot-clé
00F9 E1	172	POP	HL	
00FA C0	173	RET	NZ	;S'il ne correspond pas, retour
00FB E5	174	PUSH	HL	
00FC 11FFF9	175	LD	DE, -7	
00FF 19	176	ADD	HL, DE	
0100 110000	177	LD	DE, KEYWORD	;Maintenant il faut prendre le mot-clé
0103 CD0000	178	CALL	MOVE	de la carte de mémoire
0106 E1	179	POP	HL	
0107 11FBF8	180	LD	DE, -1032	
010A 19	181	ADD	HL, DE	
010B 220212	182	LD	[RAMEND], HL	
010E 110005	183	LD	DE, 5	
0111 19	184	ADD	HL, DE	
0112 5E	185	LD	E, [HL]	
0113 23	186	INC	HL	
0114 56	187	LD	D, [HL]	
0115 ED530210	188	LD	[RAMSTRT], DE	
0119 C9	189	RET		
	190 :			
011A	191	SETSYSK	LD__HL_WK	BPARERA
011E 2B	192	DEC	HL	
011F 11013B	193	LD	DE, MAIN1	
0122 72	194	LD	[HL], D	
0123 2B	195	DEC	HL	
0124 73	196	LD	[HL], E	
0125 11FFFC	197	LD	DE, -4	;Utilisation du module RST (SAVERG)
0128 19	198	ADD	HL, DE	;Simuler les actions POP IX, POP IY, RET
0129	199	ST__HL_WK	SYSSTK	
012D C9	200	RET		
	201;			

6.3 Exemple d'expansion de dispositif au moyen de sous-programmes de contrôle de dispositif

(Exemple 18)

```

1           "Z80"
<E73C>    2 DTSCAN EQU      0E73CH   ;Balayage de dispositif
<0047>    3 DEVHOCK EQU     00047H   ;Hock de dispositif 1
<00C3>    4 JP code   EQU      0C3H     ;Code Z80 JP
<0000>    5 OPORT   EQU      0H       ;Adresse du point de sortie de l'utilisateur
<0001>    6 IPORT   EQU      1H
7;
0000 6C6F7665 8 START  DEF B   'love'
0004 0030    9 DEFW    PIOOPN   ;Initialisation dure de dispositif
0006 000D   10 DEFW    INIT     ;Initialisation douce
0008 000A   11 DEFW    INITS   ;De sleep
12;
000A CD0030 13 INIT S CALL    PIOOPN   ;Tous les bits = 0
000D 3EC3   14 INIT S LD      A, JPCODE ;Tableau à hock de dispositif
000F 320048 15 LD      [DEVHOCK + 1], A
0012 210019 16 LD      HL, DEVSCN
0015 220049 17 LD      [DEVHOCK + 2], HL
0018 C9   18 NOP    RET
19;
0019 D0   20 DEVSCN RET     NC      ;Retour si on trouve un autre dispositif
001A 110021 21 LD      DE, DEVtbl ;Ici sauvegarder les données IX, HL, BC
001D CDE73C 22 CALL    DTSCAN   ;Retour si le nom du dispositif est trouvé
0020 C9   23 RET
24;
0021 50494FBA 25 DEVtbl  DEF B   'PIO', `: + 128 ;Le dispositif balaye le nom (bit 7 = 1)
0025 00   26 DEFB    0        indique le caractère final du nom
0026 15   27 DEFB    21      ;Banque de dispositif = 0
0027 0320035 28 DEFW    PIOPUT, PIOSNS ;Code de dispositif
0028 0300018 29 DEFW    PIOOPN, NOP  ;Ecrire, lire
30;
002F 00   31 DEFB    0        ;Ouvrir, option
32;----- Paramètre d'entrée ouverte -----
33; <ESCAPE> = reste la longueur du nom
34; <ESCAPE + 1>, <ESCAPE + 2> = reste le nom du pointeur
35; IX = 1ère donnée d'opérande      implicite = 0
36; B = 2ème donnée d'opérande      implicite = 0
37; A = mode ouvert 0: appelé par la commande "INIT", "LOAD"
38;                   1: appelé par la commande "SAVE"
39;----- 
40; Nous pouvons modifier tous les registres
41; Les données de retour HL, BC, DE sont sauvegardées dans KBNtbln + 2,
42; .KBNtbln + 4, KBNtbln + 6, -----
0030 3E00 43 PIOOPN LD      A, 0      ;Initialise les entrées parallèles, toutes les entrées = 0
0032 D300 44 PIOPUT OUT    [OPORT], A ;Données de sortie aux points de sortie
0034 C9   45 RET
46;----- Condition de retour -----
47;      ---- SNSAD ----
48;      Report  Zéro  Reg A
49;          0      0      données  retour normal
50;          0      1      X      pas de données
51;          1      X      X      erreur io
52;----- Nous pouvons modifier tous les registres. ----- 
0035 DB01 53 PIOSNS IN      A, [IPORT] ;Prendre les données au point d'entrée
0037 A7   54 AND     A      ;report = 0 si il n'y a pas de statut d'erreur io
0038 0600 55 LD      B, 0
003A 04   56 INC     B
003B C9   57 RET      z = 0 pour état de données valide

```

7. ANNEXES

7.1 Tableau d'adresse HOCK

CONPUT	009FH	Définition du dispositif de sortie de console. Le dispositif implicite est le LCD.
CONSAIS	00A2H	Définition du dispositif d'entrée de console. Le dispositif implicite est le clavier.
DEVOCK	0047H ↓ 0056H	Il y a quatre HOCKS de dispositifs à 4 octets.
ABORT HOCK	00A8H	Adresse de HOCK à ABORT.
ERRHOCK	00ABH	Le code implicite est le code de retour.

7.2 Tableau d'adresses de symboles

ABORT	C03FH
BRKFLG	002BH
ABTHOCK	00ABH
INTA	003CH
INTB	0034H
INTC	002CH
KBNTBL	02C5H
EXOPN	E6A8H
DASSIGN	E827H
GETAD	E8D4H
OUT-CHR	E88FH
ERRHOCK	00ABH
TXTTAB	00B2H
RAMSTRT	0210H
RAMEND	0212H
PRMPRV	03ACH
PRMSTK	0344H
ENDBUF	01D7H
TSACK	0552H
MEMSIZ	01DFH
STKTOP	01DDH
VARTAB	0322H
DEVHOCK	0047H

Canon CANON INC.

7-1, 2-chome, Nishi-shinjuku, Shinjuku-ku, Tokyo 160, Japan
P.O. Box 5050, Shinjuku Dai-ichi Seimei Building, Tokyo 160, Japan

CANON U.S.A., INC.

HEAD OFFICE One Canon Plaza, Lake Success, N.Y. 11042, U.S.A.

CHICAGO 140 Industrial Drive, Elmhurst, Illinois 60126, U.S.A.

LOS ANGELES 123 Paularino Avenue East, Costa Mesa, California 92626, U.S.A.

ATLANTA 6380 Peachtree Industrial Blvd., Norcross, Georgia 30071, U.S.A.

DALLAS 2035 Royal Lane, Suite 290, Dallas, Texas 75229, U.S.A.

CANON CANADA INC.

HEAD OFFICE 6390 Dixie Road Mississauga, Ontario, L5T 1P7, Canada

CALGARY 2828, 16th Street, N.E. Calgary, Alberta, T2E 7K7, Canada

CANON EUROPA N.V.

P.O. Box 7907, 1008 AC Amsterdam, The Netherlands

CANON FRANCE S.A.

DIVISION Micro Informatique 93154 Le Blanc Mesnil, Cedex, France

CANON RECHNER DEUTSCHLAND GmbH.

Fraunhoferstrasse 14, Postfach 8033, München-Martinsried, West Germany

CANON UK LTD.

Waddon House, Stafford Road, Croydon CR9 4DD, England

CANON LATIN AMERICA, INC.

SALES DEPARTMENT P.O. Box 7022, Panama 5, Rep. of Panama

REPAIR SERVICE CENTER P.O. Box 2019, Colon Free Zone, Rep. of Panama

CANON HONG KONG TRADING CO., LTD.

Golden Bear Industrial Centre, 7th Floor, 66-82 Chai Wan Kok Street, Tsuen Wan,
New Territories, Hong Kong

CANON AUSTRALIA PTY. LTD.

1 Hall Street, Hawthorn East, Victoria 3123, Australia