

4/2.7

Les banques ROM ou FIRMWARE

Cette partie décrit les micro-programmes installés dans les mémoires ROM occupant les adresses 0000H à 4000H.

Reportez-vous à la partie 2 pour avoir des détails sur l'organisation des mémoires de l'AMSTRAD.

Ces programmes élémentaires ont pour but de faciliter la gestion des mémoires et périphériques pour un programmeur qui travaille en ASSEMBLEUR.

Nous avons divisé les fonctions du firmware en 9 groupes :

- clavier,
- écran en mode texte,
- écran en mode graphique,
- gestion hard de l'écran,
- circuit sonore,
- gestion de l'interface cassette et disquette,
- kernel (« cœur » de l'ordinateur),
- imprimante et gestion bas niveau de la carte mère,
- bloc d'indirection.

Chaque fonction est présentée sous la forme d'un ou plusieurs mot(s)-clé(s) (par exemple, KM INITIALISE) et d'un point d'entrée en hexadécimal. Une description de la fonction est ensuite donnée. Cette description est accompagnée des paramètres en entrée nécessaires pour activer la fonction et des paramètres modifiés en sortie. Enfin, les fonctions en relation directe sont citées.

Prenons l'exemple de KM TEST KEY.

Cette fonction teste si une touche particulière ou le bouton du joystick est pressé.

En entrée, le registre A contient le code de la touche à tester. En sortie, le flag « Zéro » indique si une touche est pressée, et le registre C contient l'état des touches SHIFT et CONTROL. Enfin, les registres A et HL, et les autres flags sont écrasés.

Si nous voulons attendre l'appui sur une touche particulière, puis déclencher un traitement quand cette touche est actionnée, la fonction KM TEST KEY pourra être utilisée de la manière suivante :

```
ADR : EQU    $                ;Adresse de rebouclage
          LD    A,<Code touche a tester>
          CALL  OBB 1EH        ;Appel de KM TEST KEY
          JR    Z,ADR          ;La touche n'est pas pressée
TRA : EQU    $
          ;Traitement relatif a la touche
          RET
```

Ce sous-programme sera appelé par CALL ADR.

FONCTIONS CLAVIER

KM INITIALISE #BB00

Initialisation complète du clavier.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : KM RESET

KM RESET #BB03

Initialisation du clavier : Indirections et buffers.

Entrée : Aucune.

Sortie : AF, BC, DE, HL écrasés.
Les autres registres sont intacts.

En relation : KM DISARM
 KM EXP BUFFER
 KM INITIALISE

KM WAIT CHAR #BB06

Attend un caractère issu du clavier ou de l'extension courante.
Cette routine attend jusqu'à ce qu'un caractère soit disponible, s'il ne l'est pas immédiatement.

Entrée : Aucune.

Sortie : Carry à 1, les autres indicateurs sont écrasés.
A contient le caractère.
Les autres registres sont intacts.

En relation : KM CHAR RETURN
 KM READ CHAR
 KM WAIT KEY

KM READ CHAR #BB09

Essaie d'obtenir un caractère du buffer clavier ou de l'extension courante.
Cette routine n'attend pas que le caractère soit disponible s'il n'y en a aucun de disponible immédiatement.

Entrée : Aucune.

Sortie : Si un caractère était disponible :
Carry à 1.
A contient le caractère.

Si aucun caractère n'était disponible :
Carry à 0.
A écrasé.
Les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : KM CHAR RETURN
KM READ KEY
KM WAIT CHAR

KM CHAR RETURN #BBOC

Sauve un caractère pour un futur appel à KM READ CHAR ou KM WAIT CHAR.

Entrée : A contient le caractère à mettre dans le buffer.

Sortie : Tous les registres et les flags sont préservés.

En relation : KM READ CHAR
KM WAIT CHAR

KM SET EXPAND #BBOF

Valide une chaîne d'expansion associée à une clé d'expansion.

Entrée : B contient la clé d'expansion pour l'expansion à valider.
C contient la longueur de la chaîne
HL contient l'adresse de la chaîne.

Sortie : Si l'expansion peut être réalisée : Carry = 1.
Si la chaîne est trop longue ou le token invalide : Carry = 0.
Dans tous les cas : A, BC, DE et HL détruits.
Les autres registres sont intacts.

En relation : KM GET EXPAND
KM READ CHAR
KM WAIT CHAR

KM GET EXPAND #BB12

Lit un caractère d'une chaîne d'expansion.
Les caractères de la chaîne sont numérotés à partir de 0.

Entrée : A contient le code d'expansion.
L contient le numéro du caractère à lire.

Sortie : Si le caractère a été trouvé :
Carry = 1.
A contient le caractère.

Si la clé est invalidée ou si la chaîne n'est pas assez longue :
Carry = 0.
A est écrasé.

Dans tous les cas : DE et les autres flags sont écrasés.
Tous les autres registres sont intacts.

En relation : KM READ CHAR
KM SET EXPAND

KM EXP BUFFER #BB15

Déclaration d'un buffer d'expansion (adresse et longueur)

Entrée : DE contient l'adresse du buffer
HL contient la longueur du buffer.

Sortie : Si tout s'est bien passé, Carry = 1
Si le buffer est trop petit, Carry = 0.
Dans tous les cas, A, BC, DE, HL et les autres flags sont écrasés.
Les autres registres sont préservés.

En relation : KM GET EXPAND
KM SET EXPAND

KM WAIT KEY #BB18

Attend l'appui sur une touche du clavier.

Entrée : Aucune.

Sortie : Carry = 1.
A contient le caractère ou la clé d'expansion.
Les autres flags sont écrasés.
Les autres registres sont préservés.

En relation : KM READ KEY
KM WAIT CHAR

KM READ KEY #BB1B

Essaye de lire immédiatement une touche dans le buffer clavier.

Entrée : Aucune.

Sortie : Si le buffer n'est pas vide, Carry = 1.
A contient le caractère ou le code d'expansion.
Si le buffer est vide, Carry = 0.
A écrasé.
Dans tous les cas :
Les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : KM READ CHAR
KM WAIT KEY

KM TEST KEY #BB1E

Teste si une touche particulière ou le bouton du joystick est pressé.

Entrée : A contient le code de la touche à tester.

Sortie : Si la touche est pressée : Flag Zéro = 0.

 Sinon : Flag Zéro = 1.

 Dans tous les cas : Flag Carry = 0.

 C contient l'état des touches SHIFT et CONTROL,
 (Bit 7 = CONTROL, et Bit 5 = SHIFT)

 A, HL et les autres flags sont écrasés.

 Les autres registres sont préservés.

En relation : KM GET JOYSTICK

 KM GET STATE

 KM READ KEY

KM GET STATE #BB21

Teste l'état des touches CAPS LOCK et SHIFT LOCK.

Entrée : Aucune.

Sortie : L contient l'état de la touche SHIFT LOCK

 (#00 si non pressée, #FF si pressée).

 H contient l'état de la touche CAPS LOCK

 (#00 si non pressée, #FF si pressée).

 AF est écrasé.

 Les autres registres sont préservés.

En relation : KM TEST KEY

KM GET JOYSTICK #BB24

Teste l'état du joystick.

Entrée : Aucune.

Sortie : H contient l'état du joystick 0.

 L contient l'état du joystick 1.

 A contient l'état du joystick 0.

 Les flags sont écrasés.

 Les autres registres sont intacts.

 Etat du joystick :

 Bit 0 : Vers le haut,

 Bit 1 : Vers le bas,

 Bit 2 : Vers la gauche,

 Bit 3 : Vers la droite,

 Bit 4 : Bouton feu N° 2,

 Bit 5 : Bouton feu N° 1,

 Bit 6 : Inutilisé,

 Bit 7 : Toujours à 0.

 Un bit à 1 signifie que la direction ou la touche correspon-
 dante est activée.

En relation : KM TEST KEY

KM SET TRANSLATE #BB27

Force l'entrée à la table de translation normale.

Entrée : A contient le numéro de la touche.
B contient la nouvelle translation.

Sortie : AF et HL sont écrasés.
Les autres registres sont intacts.

Remarques :

- Un numéro de touche supérieur à #79 ne produit aucune action.
- La liste des numéros spéciaux est donnée ci-dessous :

#80 à #9F : codes d'expansion étendus à des chaînes de caractères
quand KM READ CHAR ou KM WAIT CHAR sont appelés.

#FD : CAPS LOCK (bascule)

#FE : SHIFT LOCK (bascule)

#FF : pour ignorer la prochaine touche.

En relation : KM GET TRANSLATE
KM SET CONTROL
KM SET SHIFT

KM GET TRANSLATE #BB2A

Attend une entrée de la table de translation normale.

Entrée : A contient un code de touche.

Sortie : A contient la translation.
HL et les flags sont effacés.
Les autres registres sont effacés.

Remarques : Se reporter à KM SET TRANSLATE.

En relation : KM GET CONTROL
KM GET SHIFT
KM SET TRANSLATE

KM SET SHIFT #BB2D

Force l'entrée au clavier avec la touche SHIFT appuyée.

Entrée : A contient un numéro de touche.
B contient la nouvelle translation.

Sortie : AF et HL sont écrasés.
Les autres registres sont intacts.

Remarques : Se reporter à KM SET TRANSLATE

En relation : KM GET SHIFT
KM SET CONTROL
KM SET TRANSLATE

KM GET SHIFT #BB30

Attend un caractère pour le traduire dans la table SHIFT

Entrée : A contient le code de la touche.

Sortie : A contient la translation.

HL et les flags sont écrasés.

Les autres registres sont intacts.

Remarques : Se reporter à KM SET TRANSLATE.

En relation : KM GET CONTROL
KM GET TRANSLATE
KM SET SHIFT

KM SET CONTROL #BB33

Force l'entrée au clavier avec la touche CONTROL appuyée.

Entrée : A contient le code de la touche.

B contient la nouvelle translation.

Sortie : AF et HL sont effacés.

Les autres registres sont intacts.

Remarques : Se reporter à KM SET TRANSLATE.

En relation : KM GET CONTROL
KM SET SHIFT
KM SET TRANSLATE

KM GET CONTROL #BB36

Attend un caractère pour le transfert dans la table de contrôle.

Entrée : A contient le code caractère.

Sortie : A contient la translation.

HL et les autres flags sont écrasés.

Les autres registres sont préservés.

Remarques : Se reporter à KM SET TRANSLATE.

En relation : KM GET SHIFT
KM GET TRANSLATE
KM SET CONTROL

KM SET REPEAT #BB39

Indique si une touche particulière peut être répétée dans la table des touches « répétables ».

Entrée : Répétition non autorisée : B = #FF.

Répétition autorisée : B = #00.

A contient le code de la touche concernée.

Sortie : AF, BC et HL sont effacés.

Les autres registres sont intacts.

En relation : KM GET REPEAT
KM SET DELAY

KM GET REPEAT #BB3C

Teste si une touche est autorisée en mode « AUTO REPEAT » dans la table des touches « répétables ».

Entrée : A contient le code de la touche à tester.

Sortie : Si la touche est « AUTO-REPEAT », Flag Zéro = 0.
Sinon , Flag Zéro = 1.

Flag Carry = 0

A, HL et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : KM SET REPEAT

KM SET DELAY #BB3F

Fixe le temps au bout duquel une touche est répétée pour la première fois et le temps entre deux répétitions.

Entrée : H contient le premier délai. (1)

L contient le deuxième délai. (2)

(1) par défaut = 30 cycles et 0.02 sec. par cycle.

(2) par défaut = 2 cycles

Sortie : AF écrasés.

Les autres registres sont préservés.

En relation : KM GET DELAY

KM SET REPEAT

KM GET DELAY #BB42

Donne le temps au bout duquel une touche est répétée pour la première fois et le temps entre deux répétitions.

Entrée : Aucune.

Sortie : H contient le premier délai.

L contient le second délai.

AF écrasés.

Les autres registres sont intacts.

En relation : KM SET DELAY

KM ARM BREAKS #BB45

Arme le mécanisme de BREAK.

Le prochain appel à KM BREAK EVENT génèrera une routine de BREAK.

Entrée : DE contient l'adresse de la routine de BREAK.

C contient l'adresse de ROM SELECT pour cette routine.

Sortie : AF, BC, DE et HL sont écrasés.

Les autres registres sont intacts

En relation : KM BREAK EVENT

KM DISARM BREAK

KM DISARM BREAK #BB48

Désarme le mécanisme de BREAK.

L'effet de KM BREAK EVENT devient nul suite à cette instruction.

Entrée : Aucune.

Sortie : AF et HL écrasés.

Les autres registres sont intacts.

En relation : KM ARM BREAK
KM BREAK EVENT

KM BREAK EVENT #BB4B

Si le BREAK est armé, provoque un déroutement sur BREAK :

Un caractère spécial est inséré dans le buffer clavier. A la prochaine lecture du buffer, ce caractère spécial génère un déroutement sur BREAK.

Entrée : Aucune.

Sortie : AF et HL sont écrasés.

Les autres registres sont préservés.

En relation : KM ARM BREAK
KM DISARM BREAK

FONCTIONS ECRAN EN MODE TEXTE

TXT INITIALISE #BB4E

Initialisation complète de l'écran : Toutes les variables texte et les indications relatives à l'écran sont perdues ainsi que le mode d'affichage de l'écran.

Entrée : Aucune.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

Remarques :

TXT DRAW CURSOR, TXT UNDRAW CURSOR, TXT WRITE CHAR, TXT UNWRITE CHAR et TXT OUT ACTION sont initialisés à leurs valeurs par défaut.

La table de contrôle est initialisée aux codes de contrôle par défaut.

La table de caractères redéfinis par l'utilisateur est réinitialisée.

Tous les champs sont à leur valeur par défaut.

En relation : SCR INITIALISE
TXT RESET

TXT RESET #BB51

Initialise les indirections de l'écran et la table de contrôle.

Entrée : Aucune.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

Remarques :

TXT DRAW CURSOR, TXT UNDRAW CURSOR, TXT WRITE CHAR, TXT UNWRITE CHAR et TXT OUT ACTION sont initialisés à leurs valeurs par défaut.

En relation : TXT INITIALISE

TXT VDU ENABLE #BB54

Permet d'afficher un caractère sur l'écran par l'ordre TXT OUTPUT ou TXT WR CHAR.

Entrée : Aucune.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : TX CUR ENABLE
TXT OUTPUT
TXT VDU DISABLE
TXT WR CHAR

TXT VDU DISABLE #BB57

Empêche l'affichage de caractères par les routines TXT OUTPUT et TXT WR CHAR.

Entrée : Aucune.

Sortie : AF écrasés.

Les autres registres sont intacts.

En relation : TXT CUR DISABLE
TXT OUTPUT
TXT VDU ENABLE
TXT WR CHAR

TXT OUTPUT #BB5A

Affiche un caractère à l'écran en obéissant aux codes de contrôle compris entre #00 et #1F.

Entrée : A contient le caractère à afficher.

Sortie : Tous les registres et les flags sont intacts.

En relation : GRA WR CHAR
TXT OUT ACTION
TXT SET GRAPHIC
TXT VDU DISABLE
TXT VDU ENABLE
TXT WR CHAR

TXT WR CHAR #BB5D

Affiche un caractère à l'écran. Les caractères de contrôle sont affichés mais non exécutés.

Entrée : A contient le caractère à afficher.

Sortie : AF, BC, DE et HL sont écrasés.

Les autres registres sont intacts.

En relation : GRA WR CHAR
TXT OUTPUT
TXT RD CHAR
TXT WRITE CHAR

TXT RD CHAR #BB60

Lit un caractère sur l'écran à la position occupée par le curseur.

Entrée : Aucune.

Sortie : Si un caractère connu (Code entre #00 et #FF) est trouvé,
Flag Carry = 1.

A contient le caractère lu.

Si le caractère n'est pas reconnu, Flag Carry = 0.

A = 0.

Les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : TXT UNWRITE
TXT WR CHAR

TXT SET GRAPHIC #BB63

Valide ou dévalide (basculer) l'option affichage graphique sur l'écran.

Entrée : Option affichage graphique validée : A < 0.

Option affichage graphique dévalidée : A = 0.

Sortie : AF effacés.

Les autres registres sont intacts.

En relation : GRA WR CHAR
TXT OUTPUT

TXT WIN ENABLE #BB66

Fixe les dimensions de la fenêtre de texte utilisée (largeur et hauteur).

Entrée : H contient la colonne d'un des côtés de la fenêtre.

D contient la colonne de l'autre côté de la fenêtre.

L contient la ligne du haut ou du bas de la fenêtre.

E contient l'autre ligne de la fenêtre.

Sortie : AF, BC, DE et HL sont écrasés.

Les autres registres sont intacts.

Remarques :

La fenêtre n'est pas effacée.

Le curseur est positionné en haut et à gauche de la fenêtre.

En relation : TXT GET WINDOW
TXT VALIDATE

TXT GET WINDOW #BB69

Indique la taille de la fenêtre utilisée.

Entrée : Aucune.

Sortie : Si la fenêtre occupe tout l'écran, Flag Carry = 0.

Sinon, Flag Carry = 1.

H contient le numéro de la colonne la plus à gauche.

D contient le numéro de la colonne la plus à droite.

L contient le numéro de la ligne la plus haute.

E contient le numéro de la ligne la plus basse.

A écrasé.

Les autres registres sont intacts.

En relation : TXT VALIDATE
TXT WIN ENABLE

TXT CLEAR WINDOW #BB6C

Efface la fenêtre

Entrée : Aucune.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

Remarque : Le curseur est positionné en haut et à gauche de la fenêtre.

En relation : GRA CLEAR WINDOW
SCR CLEAR
TXT SET PAPER
TXT WIN ENABLE

TXT SET COLUMN #BB6F

Fixe la position horizontale du curseur.

Le curseur est effacé de l'ancienne position et affiché à la nouvelle.

Entrée : A contient la colonne où l'on veut positionner le curseur.

Sortie : AF et HL sont écrasés.
Les autres registres sont intacts.

Remarque : La colonne 1 se trouve à l'extrémité gauche de la fenêtre.

En relation : TXT GET CURSOR
TXT SET CURSOR
TXT SET ROW

TXT SET ROW #BB72

Fixe la position verticale du curseur.

Le curseur est effacé de l'ancienne position et affiché à la nouvelle.

Entrée : A contient la ligne où l'on veut positionner le curseur.

Sortie : AF et HL écrasés.

Remarque :

La ligne 1 est située en haut de la fenêtre.
Les autres registres sont intacts.

En relation : TXT GET CURSOR
TXT SET COLUMN
TXT SET CURSOR

TXT SET CURSOR #BB75

Fixe les positions horizontale et verticale du curseur.

Le curseur est effacé de l'ancienne position et affiché à la nouvelle.

Entrée : H contient la colonne demandée.
L contient la ligne demandée.

Sortie : AF et HL sont écrasés.
Les autres registres sont intacts.

En relation : TXT GET CURSOR
TXT SET COLUMN
TXT SET ROW

TXT GET CURSOR #BB78

Donne la position actuelle du curseur.

Entrée : Aucune.

Sortie : H contient la colonne où se trouve le curseur.
L contient la ligne où se trouve le curseur.
A contient le « compteur de scroll ».
Ce compteur est décrémenté pour un scrolling vers le haut
et incrémenté pour un scrolling vers le bas.
Les flags sont écrasés.
Les autres registres sont intacts.

En relation : TXT SET COLUMN
TXT SET CURSOR
TXT SET ROW
TXT VALIDATE

TXT CUR ENABLE #BB7B

Permet l'affichage du bloc curseur à moins que le curseur ne soit éteint
par la commande TXT CUR OFF.

Entrée : Aucune.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : TXT CUR DISABLE
TXT CUR ON
TXT DRAW CURSOR
TXT UNDRAW CURSOR

TXT CUR DISABLE #BB7E

Interdit l'affichage du bloc curseur.

Entrée : Aucune.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : TXT CUR ENABLE
TXT CUR OFF
TXT DRAW CURSOR
TXT UNDRAW CURSOR

TXT CUR ON #BB81

Allume le bloc curseur. Le curseur apparaîtra à l'écran à moins que la commande TXT CUR DISABLE n'ait été émise.

Entrée : Aucune.

Sortie : Tous les registres et les flags sont intacts.

En relation : TXT CUR ENABLE
 TXT CUR OFF
 TXT DRAW CURSOR
 TXT UNDRAW CURSOR

TXT CUR OFF #BB84

Eteint le bloc curseur.

Entrée : Aucune.

Sortie : Tous les registres et les flags sont intacts.

En relation : TXT CUR DISABLE
 TXT CUR ON
 TXT DRAW CURSOR
 TXT UNDRAW CURSOR

TXT VALIDATE #BB87

Teste si une position d'écran est à l'intérieur de la fenêtre.

Entrée : H contient la colonne de la position à tester.
 L contient la ligne de la position à tester.

Sortie : Si cette position est à l'intérieur de la fenêtre,
 Flag Carry = 1. B écrasé.
 Si en écrivant à cette position, il se produit un scrolling vers
 le haut, Flag Carry = 0.
 B = #FF.
 Si en écrivant à cette position, il se produit un scrolling vers
 le bas, Flag Carry = 0.
 B = #00.
 H contient la colonne où le caractère sera écrit.
 L contient la ligne où le caractère sera écrit.
 A et les autres flags sont écrasés.
 Les autres registres sont intacts.

En relation : SCR HW ROLL
 SCR SW ROLL
 TXT GET CURSOR

TXT PLACE CURSOR #BB8A

Place un bloc curseur à la position occupée par le curseur.

Entrée : Aucune.

Sortie : AF écrasés

Les autres registres sont intacts.

Remarque : Cette commande sert lorsque l'on manipule plusieurs curseurs dans une même fenêtre.

En relation : TXT DRAW CURSOR

TXT REMOVE CURSOR

TXT REMOVE CURSOR #BB8D

Efface le bloc curseur à la position actuelle du curseur.

Entrée : Aucune.

Sortie : AF écrasés.

Les autres registres sont intacts.

Remarque : Cette commande sert lorsque l'on manipule plusieurs curseurs dans une même fenêtre.

En relation : TXT PLACE CURSOR

TXT UNDRAW CURSOR

TXT SET PEN #BB90

Permet de choisir la couleur d'encre du crayon utilisé en mode texte.

Entrée : A contient la couleur d'encre à utiliser.

Sortie : AF et HL sont écrasés.

Les autres registres sont intacts.

En relation : GRA SET PEN

SCR SET INK

TXT GET PEN

TXT SET PAPER

TXT GET PEN #BB93

Donne la couleur d'encre du crayon utilisé en mode texte.

Entrée : Aucune.

Sortie : A contient la couleur de l'encre.

Les flags sont détruits.

Les autres registres sont intacts.

En relation : GRA GET PEN

SCR GET INK

TXT GET PAPER

TXT SET PEN

TXT SET PAPER #BB96

Permet de choisir la couleur du fond de l'écran et/ou la couleur d'effacement de la fenêtre texte.

Entrée : A contient la couleur demandée.

Sortie : AF et HL sont écrasés.
Les autres registres sont intacts.

En relation : GRA SET PAPER
SCR SET INK
TXT GET PAPER
TXT SET PEN

TXT GET PAPER #BB99

Donne la couleur du fond de l'écran.

Entrée : Aucune.

Sortie : A contient la couleur du fond de l'écran
Les flags sont écrasés.
Les autres registres sont intacts.

En relation : GRA GET PAPER
SCR GET INK
TXT GET PEN
TXT SET PAPER

TXT INVERSE #BB9C

Echange les couleurs du fond de l'écran et du crayon.

Entrée : Aucune.

Sortie : AF et HL écrasés.
Les autres registres sont intacts.

En relation : TXT SET PAPER
TXT SET PEN

TXT SET BACK #BB9F

Activation des modes opaque et transparent.

Entrée : Mode opaque (fond apparent) : A = 0.
 Mode transparent (fond non apparents): A = 1.

Sortie : AF et HL écrasés.
Les autres registres sont intacts.

En relation : TXT GET BACK
TXT WR CHAR
TXT WRITE CHAR

TXT GET BACK #BBA2

Teste le mode d'affichage du fond (opaque ou transparent).

Entrée : Aucune.

Sortie : Mode opaque : A = 0.
 Mode transparent : A < > 0.
 DE, HL et les flags sont écrasés.
 Les autres registres sont intacts.

En relation : TXT SET BACK.

 TXT GET MATRIX #BBA5

Détermine si un caractère a été redéfini ou non.

Entrée : A contient le caractère à tester.

Sortie : Si le caractère a été redéfini, Flag Carry = 1.
 Si le caractère n'a pas été redéfini, Flag Carry = 0.
 HL contient l'adresse de la matrice de définition du caractère.
 A et les autres flags sont détruits.
 Les autres registres sont intacts.

En relation : TXT SET MATRIX.

 TXT SET MATRIX #BBA8

Signale à l'ordinateur qu'un caractère va être redéfini.

Entrée : A contient le caractère que l'on veut redéfinir.
 HL contient l'adresse de la matrice de redéfinition.

Sortie : Si le caractère peut être redéfini, Flag Carry = 1.
 Si le caractère ne peut être redéfini, Flag Carry = 0.
 A, BC, DE, HL et les autres flags sont détruits.
 Les autres registres sont préservés.

En relation : TXT GET MATRIX
 TXT SET M TABLE

 TXT SET M TABLE #BBAB

Définit la matrice de redéfinition d'un caractère.

Entrée : DE contient le premier caractère de la table.
 HL contient l'adresse du début de la nouvelle table.

Sortie : Si ce caractère n'a pas été précédemment redéfini,
 Flag Carry = 0.
 A et HL sont détruits.
 Si ce caractère a été précédemment redéfini, Flag Carry = 1.
 A contient le premier caractère de l'ancienne table.
 HL contient l'adresse de l'ancienne table de redéfinition.

BC, DE et les autres flags détruits.
Les autres registres sont intacts.

En relation : TXT GET M TABLE
TXT SET MATRIX

TXT GET M TABLE #BBAE

Donne l'adresse de redéfinition d'un caractère.

Entrée : Aucune.

Sortie : Si le caractère n'a pas été redéfini, Flag Carry = 0.
A et HL écrasés.
Si le caractère a été redéfini, Flag Carry = 1.
A contient le premier caractère de la table.
HL contient l'adresse du début de la table.
Les autres flags sont détruits.
Les autres registres sont intacts.

En relation : TXT GET MATRIX
TXT SET M TABLE

TXT GET CONTROLS #BBB1

Donne l'adresse de la table de contrôle des flux vidéo.

Entrée : Aucune.

Sortie : HL contient l'adresse de la table de contrôle.
Les autres registres et flags sont intacts.

Cette table de contrôle comporte trois octets pour chaque code de contrôle. Le premier donne le nombre de paramètres nécessaires pour le code de contrôle. Les deux suivants donnent l'adresse en RAM de la routine à exécuter quand tous les paramètres ont été reçus.

En relation : TXT OUTPUT.

TXT STR SELECT #BBB4

Sélectionne un flux vidéo.

Entrée : A contient le flux désiré.

Sortie : A contient le précédent flux.
HL et les flags sont écrasés.
Les autres registres sont intacts.

Remarque :

Plusieurs attributs de l'écran peuvent être définis indépendamment dans différents flux. Il est donc important de s'assurer que le flux correct est sélectionné quand un ou plusieurs attribut(s) sont différents.

Les attributs modifiables sont : Encre dans le crayon,
Encre du papier,
Position du curseur,
Dimensions de la fenêtre,
Validation/non du curseur,
Curseur allumé/éteint,
Ecran allumé/éteint,
Mode d'écriture caractère,
Mode d'écriture caractères graphiques.

En relation : TXT OUTPUT

TXT SWAP STREAM #BBB7

Echange l'état de deux flux vidéo.

Entrée : B contient un numéro de flux.
C contient l'autre numéro de flux.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

Remarque : Se reporter à TXT STR SELECT.

En relation : TXT STR SELECT

FONCTIONS ECRAN EN MODE GRAPHIQUE

GRA INITIALISE #BBBA

Initialisation de l'écran graphique :

- Remise à jour des indirections GRA PLOT, GRA TEST et GRA LINE,
- Fond graphique mis à INK 0,
- Crayon graphique mis à INK 1,
- Origine de l'écran en haut et à gauche,
- Position du curseur graphique sur l'origine,
- Fenêtre graphique occupe tout l'écran,
- Fenêtre non effacée.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA RESET
SCR INITIALISE

GRA RESET #BBBD

Initialisation des indirections GRA PLOT, GRA TEST et GRA LINE à leurs valeurs d'origine.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA INITIALISE.

GRA MOVE ABSOLUTE #BBCO

Définit la position graphique courante aux coordonnées absolues X,Y.

Entrée : DE contient l'abscisse absolue X.
HL contient l'ordonnée absolue Y.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

Remarque : La position absolue est donnée par rapport à l'origine définie par l'utilisateur.

En relation : GRA ASK CURSOR
GRA MOVE RELATIVE.

GRA MOVE RELATIVE #BBC3

Définit la position graphique courante aux coordonnées relatives X, Y.

Entrée : DE contient le déplacement relatif en abscisse X.
HL contient le déplacement relatif en ordonnée Y.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

En relation : GRA ASK CURSOR
GRA MOVE ABSOLUTE

GRA ASK CURSOR #BBC6

Donne la position graphique absolue courante.

Entrée : Aucune.

Sortie : DE contient l'abscisse absolue X.
HL contient l'ordonnée absolue Y.

Remarque : Comme toujours, la position graphique absolue courante est donnée par rapport à l'origine.

En relation : GRA MOVE ABSOLUTE
GRA MOVE RELATIVE

GRA SET ORIGIN #BBC9

Définit l'origine graphique de l'écran.

Entrée : DE contient l'abscisse absolue de l'origine X.
HL contient l'ordonnée absolue de l'origine Y.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

Remarque : La position (0,0) se situe en haut et à gauche de l'écran.

Le changement de mode par SCR SET MODE positionne l'origine en (0,0).

En relation : GRA GET ORIGIN.

GRA GET ORIGIN #BBCC

Donne les coordonnées de l'origine graphique.

Entrée : Aucune.

Sortie : DE contient l'abscisse absolue X de l'origine.
HL contient l'ordonnée absolue Y de l'origine.

En relation : GRA SET ORIGIN.

GRA WIN WIDTH #BBCF

Fixe la largeur de la fenêtre graphique.

Entrée : DE contient l'abscisse absolue d'un des côtés.
HL contient l'abscisse absolue de l'autre côté.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

Remarques : Selon le MODE choisi (0, 1 ou 2), les valeurs des registres DE et HL subissent les contraintes suivantes :

Mode 0 :

Le côté gauche a une abscisse multiple de 2.
Le côté droit a une abscisse multiple de 2 - 1.

Mode 1 :

Le côté gauche a une abscisse multiple de 4.
Le côté droit a une abscisse multiple de 4 - 1.

Mode 2 :

Le côté gauche a une abscisse multiple de 8.
Le côté droit a une abscisse multiple de 8 - 1.

Par défaut, la fenêtre graphique occupe tout l'écran.

En relation : GRA GET W WIDTH
GRA WIN HEIGHT

GRA GET W WIDTH #BBD5

Donne les abscisses des côtés droit et gauche de la fenêtre graphique.

Entrée : Aucune.

Sortie : DE contient l'abscisse absolue du côté gauche de la fenêtre graphique.
HL contient l'abscisse absolue du côté droit de la fenêtre graphique.
AF écrasés.
Les autres registres sont intacts.

En relation : GRA GET W HEIGHT
GRA WIN WIDTH

GRA GET W HEIGHT #BBD8

Donne les ordonnées des limites inférieure et supérieure de la fenêtre graphique.

Entrée : Aucune.

Sortie : DE contient l'ordonnée absolue du haut de la fenêtre graphique.
HL contient l'ordonnée absolue du bas de la fenêtre graphique.
AF écrasés.
Les autres registres sont intacts.

En relation : GRA GET W WIDTH
GRA WIN HEIGHT

GRA CLEAR WINDOW #BBDB

Efface la fenêtre graphique.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

Remarque : La position graphique courante est en haut et à gauche de la fenêtre graphique.

En relation : GRA SET PAPER
GRA WIN HEIGHT
GRA WIN WIDTH
SCR CLEAR
TXT CLEAR WINDOW

GRA SET PEN #BBDE

Définit la couleur du tracé graphique (points, lignes et caractères).

Entrée : A contient la couleur d'encre désirée.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : GRA GET PEN
GRA SET PAPER
SCR SET INK
TXT SET PEN

GRA GET PEN #BBE1

Donne la couleur du tracé graphique.

Entrée : Aucune.

Sortie : A contient la couleur de l'encre.
Les flags sont écrasés.
Les autres registres sont intacts.

En relation : GRA GET PAPER
GRA SET PEN
GRA GET INK
TXT GET PEN

GRA SET PAPER #BBE4

Définit la couleur de fond de l'écran graphique.

Entrée : A contient la couleur désirée.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : GRA SET PAPER
GRA SET PEN
SCR GET INK
TXT SET PAPER

GRA GET PAPER #BBE7

Donne la couleur du fond de l'écran graphique.

Entrée : Aucune.

Sortie : A contient la couleur de l'écran.
Les flags sont écrasés.
Les autres registres sont intacts.

En relation : GRA GET PEN
GRA SET PAPER
SCR GET INK
TXT GET PAPER

GRA PLOT ABSOLUTE #BBEA

Affiche un point élémentaire à une position absolue. La position graphique courante se trouve sur le point, et le point est affiché dans la couleur d'affichage graphique. Si le point sort de l'écran, aucune action n'est effectuée.

Entrée : DE contient l'abscisse X absolue.
HL contient l'ordonnée Y absolue.

Sortie : AF, BC, DE, HL écrasés.
Les autres registres sont intacts.

En relation : GRA PLOT
GRA PLOT RELATIVE
GRA TEST ABSOLUTE

GRA PLOT RELATIVE #BBED

Affiche un point élémentaire à une position relative. La position graphique courante se trouve sur le point, et le point est affiché dans la couleur d'affichage graphique. Si le point sort de l'écran, aucune action n'est effectuée.

Entrée : DE contient l'abscisse X relative signée.
HL contient l'ordonnée Y relative signée.

Sortie : AF, BC, DE, HL écrasés.
Les autres registres sont intacts.

En relation : GRA PLOT
GRA PLOT ABSOLUTE
GRA TEST RELATIVE

GRA TEST ABSOLUTE #BBF0

Donne la couleur d'un point situé aux coordonnées X, Y absolues.

Entrée : DE contient l'abscisse du point à tester.
HL contient l'ordonnée du point à tester.

Sortie : A contient la couleur d'encre du point testé.
BC, DE, HL et le flags sont écrasés.
Les autres registres sont intacts.

Remarque : Le curseur graphique est positionné aux coordonnées du point testé.

En relation : GRA PLOT ABSOLUTE
GRA TEST
GRA TEST RELATIVE

GRA TEST RELATIVE #BBF3

Donne la couleur d'un point situé à une position relative par rapport à la position graphique courante.

Entrée : DE contient un déplacement signé en abscisse.
HL contient un déplacement signé en ordonnée.

Sortie : A contient la couleur du point testé.
BC, DE, HL et les flags sont écrasés.
Les autres registres sont intacts.

Remarque : Le curseur graphique est positionné aux coordonnées du point testé.

En relation : GRA PLOT RELATIVE
GRA TEST
GRA TEST ABSOLUTE

GRA LINE ABSOLUTE #BBF6

Trace une ligne de la position graphique courante à une position absolue spécifiée.

Entrée : DE contient l'abscisse absolue X du point but.
HL contient l'ordonnée absolue Y du point but.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

Remarque : La ligne est tracée dans la couleur graphique courante.

En relation : GRA LINE
GRA LINE RELATIVE

GRA LINE RELATIVE #BBF9

Trace une ligne de la position graphique courante à une position qui lui est relative.

Entrée : DE contient l'abscisse signée X du point but.
HL contient l'ordonnée signée Y du point but.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

Remarque : La ligne est tracée dans la couleur graphique courante.

En relation : GRA LINE
GRA LINE ABSOLUTE

GRA WR CHAR #BBFC

Affiche un caractère à l'écran à la position graphique courante.

Entrée : A contient le caractère à afficher.

Sortie : AF, BC, DE et HL sont écrasés.
Les autres registres sont intacts.

Remarques : Le caractère est affiché de sorte que le point en haut et à gauche de sa matrice coïncide avec la position graphique courante.

Tous les caractères sont affichés, exceptés les caractères de contrôle de code ASCII compris entre #00 et #1F.

La position graphique courante est déplacée vers la droite de la largeur du caractère (32 points en MODE 0, 16 points en MODE 1 et 8 points en MODE 2).

Le caractère est écrit avec la couleur de stylo graphique sur un fond graphique.

En relation : TXT SET GRAPHIC
TXT WR CHAR

SCR INITIALISE #BBFF

Initialisation complète de l'écran :

Les indirections SCR READ, SCR WRITE et SCR MODE CLEAR sont forcées à leurs routines de défaut.

Les couleurs d'encre sont forcées à leurs valeurs par défaut.

Les périodes de flash sont forcées à leur valeur par défaut.

L'écran passe en MODE 1.

L'adresse d'écran commence en #C000 et finit en #FFFF.

Le décalage de l'écran est nul.

L'écran a une couleur de fond 0.

L'écriture graphique se fait en mode FORCE.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA INITIALISE
SCR RESET
TXT INITIALISE

SCR RESET #BC02

Initialisation de l'écran :

Indirections de l'écran, couleurs d'encre et vitesse de flash remises à leurs valeurs par défaut.

L'écriture graphique se fait en mode FORCE.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés
Les autres registres sont intacts.

En relation : SCR INITIALISE
SCR SET ACCESS
SCR SET FLASHING
SCR SET INK

SCR SET OFFSET #BC05

Définit le décalage du début de l'écran.

Entrée : HL contient le décalage désiré.

Sortie : AF et HL écrasés.
Les autres registres sont intacts.

Remarque : Le décalage ne peut excéder #07FE et est masqué à cette valeur pour empêcher tout débordement.

En relation : MC SCREEN OFFSET
SCR GET LOCATION
SCR HW ROLL
SCR SET BASE

SCR SET BASE #BC08

Définit l'adresse basse de la mémoire d'écran.

Entrée : A contient l'octet de poids fort de l'adresse de début d'écran.

Sortie : AF et HL écrasés.
Les autres registres sont intacts.

Remarque : Le poids fort du début de l'écran est masqué par #C0 pour être sûr d'avoir une zone de 16 KO consécutifs en RAM.

En relation : MC SCREEN OFFSET
SCR GET LOCATION
SCR SET OFFSET

SCR GET LOCATION #BC0B

Donne l'adresse de l'écran et son décalage.

Entrée : Aucune.

Sortie : A contient le MSB (Octet de poids fort) du début de l'écran.

HL contient le décalage de l'écran.

Les flags sont écrasés.

Les autres registres sont intacts.

En relation : SCR SET BASE
SCR SET OFFSET

SCR SET MODE #BB0E

Permet de choisir le mode d'affichage (0, 1 ou 2).

Entrée : A contient le mode choisi.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : MC SET MODE
SCR GET MODE

SCR GET MODE #BC11

Donne le MODE dans lequel se trouve l'écran (0, 1 ou 2).

Entrée : Aucune.

Sortie : Si le mode est 0 : Flag Carry = 1.

Flag Zéro = 0.

A = 0.

Si le mode est 1 : Flag Carry = 0.

Flag Zéro = 1.

A = 1.

Si le mode est 2 : Flag Carry = 0.

Flag Zéro = 0.

A = 2.

Les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : SCR SET MODE

SCR CLEAR #BC14

Efface l'écran avec l'encre 0.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : GRA CLEAR WINDOW
SCR MODE CLEAR
TXT CLEAR WINDOW

SCR CHAR LIMITS #BC17

Donne les dimensions de l'écran en caractères et donc détermine le MODE d'affichage.

Entrée : Aucune.

Sortie : B contient le numéro de dernière colonne de l'écran.

(La première colonne porte le numéro 0)

C contient le numéro de dernière ligne de l'écran.

(La première ligne porte le numéro 0)

AF écrasés.

Les autres registres sont intacts.

En relation : SCR GET MODE

SCR CHAR POSITION #BC1A

Convertit une position caractère sur l'écran en adresse écran.

Entrée : H contient la colonne pointée.

L contient la ligne pointée.

Sortie : HL contient l'adresse d'écran de la première ligne élémentaire du caractère (en haut et à gauche).

B contient la largeur en octets d'un caractère.

(largeur = 1 en MODE 2, largeur = 2 en MODE 1,

largeur = 4 en MODE 0).

AF écrasés.

Les autres registres sont intacts.

Remarque : L'adresse d'écran est calculée de la façon suivante :

AE = Adresse de base + (Déplacement MODULO #800)

avec Déplacement = $80 * \text{Ligne} + \text{Largeur} * \text{Colonne} + \text{Déplacement du début d'écran}$.

Et Largeur = 1 en MODE 2, 2 en MODE 1 et 4 en MODE 0.

En relation : SCR DOT POSITION

SCR NEXT BYTE

SCR NEXT LINE

SCR PREV BYTE

SCR PREV LINE

SCR DOT POSITION #BC1D

Convertit les coordonnées graphiques absolues en adresses écran.

Entrée : DE contient l'abscisse du point à convertir.

HL contient l'ordonnée du point à convertir.

Sortie : HL contient l'adresse écran du point défini en entrée.

C contient la marque du pixel.

B contient le nombre de pixels par octet - 1.

(2 en MODE 0, 4 en MODE 1 et 8 en MODE 2).

AF et DE sont écrasés.

Les autres registres sont intacts.

En relation : SCR CHAR POSITION
 SCR NEXT BYTE
 SCR NEXT LINE
 SCR PREV BYTE
 SCR PREV LINE

 SCR NEXT BYTE #BC20

Calcule l'adresse d'écran de l'octet situé à droite de l'octet pointé.

Entrée : HL contient l'adresse d'écran.

Sortie : HL contient la nouvelle adresse écran.

AF écrasés

Les autres registres sont intacts.

En relation : SCR CHAR POSITION
 SCR DOT POSITION
 SCR NEXT LINE
 SCR PREV BYTE
 SCR PREV LINE

 SCR PREV BYTE #BC23

Calcule l'adresse d'écran de l'octet situé à droite de l'octet pointé.

Entrée : HL contient l'adresse d'écran.

Sortie : HL contient la nouvelle adresse d'écran.

AF écrasés.

Les autres registres sont intacts.

En relation : SCR CHAR POSITION
 SCR DOT POSITION
 SCR NEXT BYTE
 SCR NEXT LINE
 SCR PREV LINE

 SCR NEXT LINE #BC26

Calcule l'adresse d'écran de l'octet situé une ligne élémentaire plus bas que l'octet pointé.

Entrée : HL contient l'adresse d'écran.

Sortie : HL contient la nouvelle adresse d'écran.

AF écrasés.

Les autres registres sont intacts.

En relation : SCR CHAR POSITION
 SCR DOT POSITION
 SCR NEXT BYTE
 SCR PREV BYTE
 SCR PREV LINE

SCR PREV LINE #BC29

Calcule l'adresse d'écran de l'octet situé une ligne élémentaire plus haut que l'octet pointé.

Entrée : HL contient l'adresse d'écran.

Sortie : HL contient la nouvelle adresse d'écran.

AF écrasés.

Les autres registres sont intacts.

En relation : SCR CHAR POSITION

SCR DOT POSITION

SCR NEXT BYTE

SCR NEXT LINE

SCR PREV BYTE

SCR INK ENCODE #BC2C

Convertit une couleur d'encre pour allumer tous les pixels d'un octet de cette couleur. Cette couleur d'encre peut être masquée pour revenir à l'état précédent en utilisant SCR INK DECODE.

Entrée : A contient la couleur d'encre.

Sortie : A contient la couleur d'encre codée.

Les flags sont écrasés.

Les autres registres sont intacts.

En relation : SCR INK DECODE

SCR INK DECODE #BC2F

Décode une couleur d'encre codée par SCR INK ENCODE.

Entrée : A contient la couleur d'encre codée.

Sortie : A contient la couleur d'encre décodée.

Les flags sont écrasés.

Les autres registres sont intacts.

En relation : SCR INK ENCODE

SCR SET INK #BC32

Définit les deux couleurs que peut prendre une encre.

Si les deux couleurs sont identiques, l'affichage avec cette encre sera continu. Dans le cas contraire, il sera intermittent et la vitesse de flash pourra être réglée par SCR SET FLASHING.

Entrée : A contient une couleur d'encre.

B contient la première couleur.

C contient la seconde couleur.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA SET PAPER
GRA SET PEN
SCR GET INK
SCR SET BORDER
SCR SET FLASHING
TXT SET PAPER
TXT SET PEN

SCR GET INK #BC35

Donne les deux couleurs utilisées pour une encre donnée.

Entrée : A contient une couleur d'encre.

Sortie : B contient la première couleur.
C contient la seconde couleur.
AF, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA GET PAPER
GRA GET PEN
SCR GET BORDER
SCR GET INK
TXT GET PAPER
TXT GET PEN

SCR SET BORDER #BC38

Définit les couleurs du bord de l'écran (BORDER).

Entrée : B contient la première couleur.
C contient la deuxième couleur.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : SCR GET BORDER
SCR SET FLASHING
SCR SET INK

SCR GET BORDER #BC3B

Donne les couleurs du bord de l'écran (BORDER).

Entrée : Aucune.

Sortie : B contient la première couleur.
C contient la seconde couleur.
AF, DE et HL sont écrasés.
Les autres registres sont intacts.

En relation : SCR GET INK
SCR SET BORDER

SCR SET FLASHING #BC3E

Définit les périodes d'affichage des deux couleurs possibles pour une encre.

Entrée : H contient la durée d'affichage de la première couleur
(en 1/50 sec. pour la France et 1/60 sec. pour la Grande-Bretagne).

L contient la durée d'affichage de la seconde couleur.

Sortie : AF et HL écrasés

Les autres registres sont intacts.

Remarque : La durée d'affichage de chaque couleur par défaut est de 1/5 sec. pour la France et de 1/6 sec. pour la Grande-Bretagne.

En relation : SCR GET FLASHING
SCR SET BORDER
SCR SET INK

SCR GET FLASHING #BC41

Donne les périodes d'affichage des deux couleurs possibles pour une encre.

Entrée : Aucune.

Sortie : H contient la durée d'affichage de la première couleur.

L contient la durée d'affichage de la deuxième couleur.

AF écrasés.

Les autres registres sont intacts.

En relation : SCR SET FLASHING

SCR FILL BOX #BC44

Remplit une surface rectangulaire définie au caractère près avec une encre donnée.

Entrée : A contient la couleur d'encre pour le remplissage.

H contient la colonne de gauche délimitant la surface à remplir.

D contient la colonne de droite délimitant la surface à remplir.

L contient la ligne du haut délimitant la surface à remplir.

E contient la ligne du bas délimitant la surface à remplir.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : SCR CLEAR
SCR FLOOD BOX
TXT CLEAR WINDOW

SCR FLOOD BOX #BC47

Remplit une surface rectangulaire définie à la ligne élémentaire près avec une encre donnée.

Entrée : C contient la couleur d'encre pour le remplissage.
HL contient l'adresse d'écran du coin en haut et à gauche du rectangle à remplir.
D contient la largeur de la surface à remplir.
E contient la hauteur de la surface à remplir.

En relation : GRA CLEAR WINDOW
SCR CLEAR
SCR FILL BOX

SCR CHAR INVERT #BC4A

Réécrit un caractère de l'écran en inverse vidéo.

Entrée : B contient une couleur d'encre.
C contient une autre couleur d'encre.
Tous les pixels du caractère pointé subiront un XOR avec la première couleur pour le motif, avec la seconde pour le fond.
H contient la colonne où se trouve le caractère à traiter.
L contient la ligne où se trouve le caractère à traiter.
(Colonne 0, Ligne 0 en haut et à gauche de l'écran.)

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : TXT PLACE CURSOR
TXT REMOVE CURSOR

SCR HW ROLL #BC4D

Produit un scrolling haut ou bas de l'écran de 8 pixels.

Entrée : Pour un scrolling bas : B = 0.
Pour un scrolling haut : B < > 0.
A contient la couleur de fond de la ligne qu'aura fait apparaître le scrolling.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

Remarque : Le scrolling est obtenu en modifiant le décalage de début d'écran.

En relation : SCR SET OFFSET
SCR SW ROLL

SCR SW ROLL #BC50

Produit un scrolling haut ou bas d'une partie de l'écran de 8 pixels.

Entrée : Pour un scrolling bas : $B = 0$.

Pour un scrolling haut : $B < > 0$.

A contient la couleur de fond de la ligne qu'aura fait apparaître le scrolling.

H contient la colonne la plus à gauche de la zone à scroller.

D contient la colonne la plus à droite de la zone à scroller.

L contient la ligne la plus haute de la zone à scroller.

E contient la ligne la plus basse de la zone à scroller.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : SCR HW ROLL

SCR UNPACK #BC53

Convertit une matrice en RAM en un caractère qui pourra être affiché sur l'écran.

Entrée : HL contient l'adresse de la matrice.

DE contient l'adresse de la RAM où va se faire le décompactage.

(En MODE 0, chaque octet de la matrice est converti en 4 octets, en MODE 1, chaque octet de la matrice est converti en 2 octets et en MODE 2, chaque octet est converti en un octet.)

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : SCR REPACK

SCR REPACK #BC56

Conversion d'un caractère de l'écran en une matrice en RAM qui représente chaque pixel allumé et chaque pixel éteint.

Entrée : A contient la couleur d'encre codée.

H contient la colonne où lire le caractère.

L contient la ligne où lire le caractère.

DE contient l'adresse où va être définie la matrice.

(Cette matrice fait 8 octets de long. Le premier octet représente la première ligne élémentaire du caractère, le 8^e la dernière ligne élémentaire du caractère.)

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

SCR ACCESS #BC59

Définit le mode d'affichage graphique (Normal, AND, OR ou XOR).

Entrée : A contient le mode d'affichage souhaité.

(0 : Mode normal : écrit par dessus l'ancien point,

1 : XOR : écrit en faisant un XOR sur le motif précédent,

2 : AND : écrit en faisant un AND sur le motif précédent,

3 : OR : écrit en faisant un OR sur le motif précédent.)

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : SCR WRITE

SCR PIXELS #BC5C

Ecrit un pixel sur l'écran en ignorant le mode d'écriture graphique défini par SCR ACCESS.

Entrée : B contient la couleur d'affichage du pixel.

C contient le masque du/des octets.

HL contient l'adresse écran du/des octets.

Sortie : AF écrasés

Les autres registres sont intacts.

En relation : SCR WRITE

SCR HORIZONTAL #BC5F

Trace une ligne horizontale dans le mode d'affichage graphique courant.

Entrée : A contient la couleur d'encre du tracé.

DE contient l'abscisse de départ de la ligne.

BC contient l'abscisse de fin de la ligne.

HL contient l'ordonnée de la ligne.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : GRA LINE ABSOLUTE

GRA LINE RELATIVE

SCR VERTICAL

SCR VERTICAL #BC62

Trace une ligne verticale dans le mode d'affichage graphique courant.

Entrée : A contient la couleur d'encre du tracé.

DE contient l'abscisse de la ligne.

HL contient l'ordonnée de départ de la ligne.

BC contient l'ordonnée de fin de la ligne.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : GRA LINE ABSOLUTE
GRA LINE RELATIVE
SCR HORIZONTAL

CAS INITIALISE #BC65

Initialise le driver cassette :

- tous les fichiers sont fermés,
- la vitesse est positionnée à sa valeur par défaut,
- les messages relatifs au lecteur de cassette sont validés.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont écrasés.

En relation : CAS IN ABANDON
CAS NOISY
CAS OUT ABANDON
CAS SET SPEED

CAS SET SPEED #BC68

Permet de choisir le taux d'échange de données avec le lecteur de cassettes.

Entrée : HL contient la longueur d'un signal logique sur la cassette ;
Cette valeur est liée à la vitesse en Bauds par :
 $Vitesse = 1/(3 * \text{longueur d'un signal logique})$.
Pour éviter des erreurs d'écriture ou de lecture, prendre des valeurs comprises entre 130 et 480 (microsec.).
A contient le taux de précompensation à appliquer :
Ce taux est de 333 microsec. par défaut et 25 microsec. de longueur de signal pour une vitesse de 1000 Bauds par défaut.

Sortie : AF et HL écrasés.
Les autres registres sont intacts.

En relation : CAS INITIALISE

CAS NOISY #BC6B

Valide ou interdit les messages relatifs au lecteur de cassettes.

Entrée : Pour valider les messages : A = 0.
Pour dévalider les messages : A < > 0.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : CAS INITIALISE

CAS START MOTOR #BC6E

Met en marche le moteur du lecteur de cassettes.

Entrée : Aucune.

Sortie : Si le moteur a pu être mis en marche : Flag Carry = 1.
Sinon, Flag Carry = 0.
A contient l'état précédent du moteur.
Les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : CAS RESTORE MOTOR
CAS STOP MOTOR

CAS STOP MOTOR #BC71

Arrête le moteur du lecteur de cassettes.

Entrée : Aucune.

Sortie : Si le moteur a pu être arrêté : Flag Carry = 1.
Sinon, Flag Carry = 0.
A contient l'état précédent du moteur.
Les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : CAS RESTORE MOTOR
CAS START MOTOR

**CAS RESTORE
MOTOR #BC74**

Restitue l'état précédent du moteur avant intervention sur CAS STOP MOTOR ou CAS START MOTOR.

Entrée : A contient l'état précédent du moteur.
(Cet état est donné par CAS START MOTOR ou CAS STOP MOTOR.)

Sortie : Si l'opération s'est bien déroulée, Flag Carry = 1.
Si l'utilisateur a appuyé sur ESCape, Flag Carry = 0.
A et les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : CAS START MOTOR
CAS STOP MOTOR

CAS IN OPEN #BC77

Ouvre un fichier en entrée et lit le premier bloc.

Entrée : B contient la longueur du nom du fichier à ouvrir.
HL contient l'adresse où se trouve le nom du fichier à ouvrir.
DE contient l'adresse d'un buffer de 2 KO. en RAM qui sera utilisé par le système.

Sortie : Si l'ouverture s'est bien passée :

Flag Carry = 1.

Flag Zéro = 0.

HL contient l'adresse du buffer contenant l'en-tête du fichier.

DE contient la position des données dans l'en-tête.

BC contient la longueur du fichier.

A contient le type de fichier.

Si le fichier est en cours d'utilisation :

Flag Carry = 0.

Flag Zéro = 0.

A, BC, DE et HL sont écrasés.

Si l'utilisateur a tapé ESCape :

Flag Carry = 0.

Flag Zéro = 1.

A, BC, DE et HL écrasés.

IX et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS IN ABANDON

CAS IN CHAR

CAS IN CLOSE

CAS IN DIRECT

CAS OUT OPEN

CAS IN CLOSE # BC7A

Ferme un fichier

Entrée : Aucune.

Sortie : Si le fichier a pu être fermé correctement (pas d'erreur de lecture) : Flag Carry = 1.

Si le fichier que l'on veut fermer n'était pas ouvert, Carry = 0.

A, BC, DE, HL et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS IN ABANDON

CAS IN OPEN

CAS OUT CLOSE

CAS IN ABANDON # BC7D

Ferme un fichier avant d'avoir fini sa lecture.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : CAS IN CLOSE

CAS IN OPEN

CAS OUT ABANDON

CAS IN CHAR #BC80

Lit un caractère dans un fichier ouvert en lecture.

Entrée : Aucune.

Sortie : Si le caractère a été correctement lu :

Flag Carry = 1.

Flag Zéro = 0.

A contient le caractère lu.

Si une fin de fichier (EOF) a été trouvée :

Flag Carry = 0.

Flag Zéro = 0.

A écrasé.

Si l'utilisateur a appuyé sur ESCape :

Flag Carry = 0.

Flag Zéro = 1.

A écrasé.

IX et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS IN CLOSE
CAS IN DIRECT
CAS IN OPEN
CAS OUT CHAR
CAS RETURN
CAS TEST EOF

CAS IN DIRECT #BC83

Lit le fichier ouvert dans un buffer.

Entrée : HL contient l'adresse où mettre les octets lus.

Sortie : Si tout s'est bien passé :

Flag Carry = 1.

Flag Zéro = 0.

HL contient l'adresse de l'en-tête.

Si le fichier n'a pas été ouvert préalablement :

Flag Carry = 0.

Flag Zéro = 0.

HL écrasé.

Si l'utilisateur a appuyé sur ESCape :

Flag Carry = 0.

Flag Zéro = 1.

HL écrasé.

A, BC, DE, IX et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS IN CHAR
CAS IN CLOSE
CAS IN OPEN
CAS OUT DIRECT

CAS RETURN #BC86

Met le dernier caractère lu par CAS IN CHAR dans le buffer de lecture : lorsque CAS IN CHAR sera réutilisée, ce dernier caractère sera lu.

Entrée : Aucune.

Sortie : Tous les registres et les flags intacts.

En relation : CAS IN CHAR

CAS TEST EOF #BC89

Teste si le caractère EOF (fin de fichier) a été rencontré.

Entrée : Aucune.

Sortie : Si oui : Flag Carry = 1.
 Flag Zéro = 0.

 Si non : Flag Carry = 0.
 Flag Zéro = 0.

 Si l'utilisateur a appuyé sur ESCape :
 Flag Carry = 0.
 Flag Zéro = 1.

 A, IX et les autres flags sont écrasés.
 Les autres registres sont intacts.

En relation : CAS IN CHAR

CAS OUT OPEN #BC8C

Ouvre un fichier cassette en écriture.

Entrée : B contient la longueur du nom du fichier.
 HL contient l'adresse où se trouve le nom du fichier à ouvrir.
 DE contient l'adresse d'un buffer de 2 KO en RAM.

Sortie : Si l'utilisateur appuie sur ESCape :
 Flag Carry = 0.
 Flag Zéro = 1.

 Si le fichier est déjà ouvert :
 Flag Carry = 0.
 Flag Zéro = 0.
 HL écrasé.

 Si tout s'est bien passé :
 Flag Carry = 1.
 Flag Zéro = 0.
 HL contient l'adresse du buffer où se trouve l'en-tête du fichier.

 A, BC, DE, IX et les autres flags sont écrasés.
 Les autres registres sont intacts.

En relation : CAS IN OPEN
CAS OUT ABANDON
CAS OUT CHAR
CAS OUT CLOSE
CAS OUT DIRECT

CAS OUT CLOSE #BC8F

Ferme un fichier ouvert en écriture.

Entrée : Aucune.

Sortie : Si le fichier s'est fermé correctement :

Flag Carry = 1.

Flag Zéro = 0.

Si le fichier n'était pas ouvert :

Flag Carry = 0.

Flag Carry = 0.

Si l'utilisateur a appuyé sur ESCape :

Flag Carry = 0.

Flag Zéro = 1.

A, BC, DE, HL, IX et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS IN CLOSE
CAS OUT ABANDON
CAS OUT OPEN

CAS OUT ABANDON #BC92

Ferme un fichier ouvert en écriture sans être parvenu en fin de fichier.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : CAS IN ABANDON
CAS OUT CLOSE
CAS OUT OPEN

CAS OUT CHAR #BC95

Ecrit un caractère dans un fichier ouvert en écriture.

Entrée : A contient le caractère à écrire

Sortie : Si tout s'est bien passé :

Flag Carry = 1.

Flag Zéro = 0.

Si le fichier n'a pas été ouvert :

Flag Carry = 0.

Flag Zéro = 0.

Si l'utilisateur a tapé ESCape :

Flag Carry = 0.

Flag Zéro = 1.

A, IX et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS IN CHAR
CAS OUT CLOSE
CAS OUT DIRECT
CAS OUT OPEN

CAS OUT DIRECT #BC98

Ecrit le contenu d'un buffer sur un fichier ouvert en écriture.

Entrée : HL contient l'adresse du buffer de données à écrire.

DE contient la longueur des données à écrire.

BC contient l'adresse d'entrée.

A contient le type de fichier.

Sortie : Si tout s'est bien passé

Flag Carry = 1.

Flag Zéro = 0.

Si le fichier n'a pas été ouvert en écriture :

Flag Carry = 0.

Flag Zéro = 0.

Si l'utilisateur a appuyé sur ESCape :

Flag Carry = 0.

Flag Zéro = 1.

A, BC, DE, HL, IX et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS IN DIRECT
CAS OUT OPEN
CAS OUT CLOSE

CAS CATALOG #BC9B

Donne la liste des fichiers contenus dans une cassette.

Entrée : DE contient l'adresse d'un buffer de 2 KO.

Sortie : Si le catalogue a été établi correctement :

Flag Carry = 1.

Flag Zéro = 0.

Si un des fichiers lus est en cours d'exploitation :

Flag Carry = 0.

Flag Zéro = 0.

Si une erreur intervient :

Flag Carry = 0.

Flag Zéro = 1.

A, BC, DE, HL, IX et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : CAS NOISY

CAS WRITE #BC9E

Ecrit un enregistrement sur un fichier cassette.

Entrée : HL contient l'adresse du buffer de données à écrire.

DE contient la longueur des données à écrire.

A contient le caractère de synchronisation à écrire en fin de chaque enregistrement. (#2C pour le premier enregistrement, #16 pour les autres).

Sortie : Si tout s'est bien passé :

Flag Carry = 1.

A écrasé.

Si une erreur est intervenue ou si l'utilisateur a tapé ESCape :

Flag Carry = 0.

A contient le code d'erreur :

0 si ESCape a été pressé,

1 l'écriture est impossible.

BC, DE, HL, IX écrasés.

Les autres registres sont intacts.

En relation : CAS CHECK

CAS READ

CAS READ #BCA1

Lit un enregistrement sur cassette.

Entrée : HL contient l'adresse d'un buffer où stocker les données lues.

DE contient la longueur des données à lire.

A contient le caractère de synchronisation de fin d'enregistrement : (#2C pour le premier enregistrement, #16 pour les autres).

Sortie : Si tout s'est bien passé :

Flag Carry = 1.

A écrasé.

Si une erreur est survenue ou si l'utilisateur a tapé ESCape :

Flag Carry = 0.

A contient le code erreur :

- 0 : ESCape a été pressé.
- 1 : Débordement dans la lecture.
- 2 : Mauvais fonctionnement du lecteur de cassettes.

En relation : CAS CHECK
CAS WRITE

CAS CHECK #BCA4

Compare un enregistrement sur cassette avec le contenu d'un buffer en RAM.

Entrée : HL contient l'adresse RAM du block à vérifier.
DE contient la longueur du block à vérifier.
A contient le caractère de synchronisation de fin d'enregistrement : (#2C pour le premier enregistrement et #16 pour les autres).

Sortie : Si la vérification est positive :
Flag Carry = 1.
A écrasé.

Si une erreur est survenue ou si l'utilisateur a tapé ESCape :
Flag Carry = 0.
A contient un code d'erreur.
0 : ESCape a été tapé.
1 : Débordement pendant la lecture.
2 : Erreur de lecture due au magnétophone.
3 : La comparaison est négative.
BC, DE, HL, IX et les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : CAS READ
CAS WRITE

SOUND RESET #BCA7

Remet à zéro le circuit sonore.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : SOUND HOLD

SOUND QUEUE #BCAA

Essaye d'ajouter un son dans le buffer sonore sur un ou plusieurs canaux.

Entrée : HL contient l'adresse du son à ajouter.

Sortie : Si le son a pu être mis dans le buffer :

Flag Carry = 1.

HL écrasé.

Si le buffer sonore était plein :

Flag Carry = 0.

HL intact.

A, BC, DE, IX et les autres flags sont écrasés.

Les autres registres sont intacts.

Remarques :

— Pour créer un son, il est nécessaire de transmettre à cette routine une suite de données qui a cette allure :

Octet 0 : Canaux à utiliser et rendez-vous :

Bit 0 : Utilisation du Canal A.

Bit 1 : Utilisation du Canal B.

Bit 2 : Utilisation du Canal C.

Bit 3 : Rendez-vous avec le canal A.

Bit 4 : Rendez-vous avec le canal B.

Bit 5 : Rendez-vous avec le canal C.

Bit 6 : Mise en attente de l'émission du son.

Le son sera émis par l'appel à SOUND RELEASE.

Bit 7 : Vide le buffer sonore et fait jouer immédiatement la note courante.

Octet 1 : Amplitude de l'enveloppe à utiliser (0..15).

Octet 2 : Enveloppe de ton à utiliser (0..15).

Octets 3 et 4 : Période de ton (0..4095).

Octet 5 : Période de bruit (0..31).

Octet 6 : Amplitude initiale (0..15).

Octets 7 et 8 : Durée de l'enveloppe (1..32768).

En relation : SOUND ARM EVENT

SOUND CHECK

SOUND RELEASE

SOUND CHECK #BCAD

Vérifie si le buffer sonore est plein ou non.

Entrée : A contient le canal à tester :

Bit 0 = 1 → Test Canal A.

Bit 1 = 1 → Test Canal B.

Bit 2 = 1 → Test Canal C.

Sortie : A contient le statut de test :

Bits 0..2 : Nombre de places disponibles dans le buffer.

Bit 3 : Le canal attend un rendez-vous avec le Canal A.

Bit 4 : Le canal attend un rendez-vous avec le Canal B.

Bit 5 : Le canal attend un rendez-vous avec le canal C.

Bit 6 : Le canal est en attente

Bit 7 : Le canal est en cours d'activation
BC, DE, HL et les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : SOUND ARM EVENT
SOUND QUEUE

SOUND ARM EVENT #BCB0

Arme le bloc de données qui sera activé quand le buffer musical sera vide.

Entrée : A contient le code du canal à armer :

Bit 0 pour le Canal A,
Bit 1 pour le Canal B,
Bit 2 pour le Canal C,
Un seul canal à la fois peut être armé.

HL contient l'adresse du bloc de données.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : KL INIT EVENT
SOUND CHECK
SOUND QUEUE

SOUND RELEASE #BCB3

Permet aux sons mis en attente d'être exécutés.

Entrée : A contient le code du canal à débloquer :

Bit 0 pour le Canal A,
Bit 1 pour le Canal B,
Bit 2 pour le Canal C.

Sortie : AF, BC, DE, HL et IX écrasés.
Les autres registres sont intacts.

En relation : SOUND QUEUE

SOUND HOLD #BCB6

Stoppe immédiatement l'émission d'un son quelconque. Le générateur sonore peut être relancé par l'appel de SOUND CONTINUE.

Entrée : Aucune.

Sortie : Si un son était actif, Flag Carry = 1.
Sinon, Flag Carry = 0.

A, BC, HL et les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : SOUND CONTINUE
SOUND RESET

SOUND CONTINUE #BCB9

Permet aux sons stoppés par la commande SOUND HOLD d'être réémis à partir de l'endroit où ils ont été stoppés.

Entrée : Aucune.

Sortie : AF, BC, DE et IX écrasés.
Les autres registres sont intacts.

En relation : SOUND HOLD
SOUND RELEASE

SOUND AMPL VOLUME #BCBC

Règle l'amplitude de volume.

Entrée : A contient un numéro d'enveloppe.
HL contient l'adresse d'un buffer en RAM qui servira à définir l'amplitude.
Octet 0 : Nombre de partie dans l'enveloppe (0..5).
Octet 1..3 : Première section.

Pour une enveloppe SOFT :
Octet 1 : Pas de comptage.
Octet 2 : Taille du pas.
Octet 3 : Temps de pause entre deux pas.

Pour une enveloppe HARD :
Octet 1 : Forme d'enveloppe.
Octets 2 et 3 : Période d'enveloppe.

Octets 4..6 : Deuxième section.
Octets 7..9 : Troisième section.
Octets 10..12 : Quatrième section.
Octets 13..15 : Cinquième section.

Sortie : Si tout s'est bien passé :
Flag Carry = 1.
HL contient l'adresse du bloc de données + 16.
A et BC écrasés.

Si le numéro d'enveloppe est invalidé :
Flag Carry = 0.
A, B et HL intacts.

DE et les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : SOUND A ADDRESS
SOUND TONE ENVELOPPE

SOUND TONE ENVELOPPE #BCBF

Définit une enveloppe de ton (une parmi les quinze possibles).

Entrée : A contient un numéro d'enveloppe.

HL contient l'adresse du bloc de données pour définir l'enveloppe.

Le bloc de données a la structure suivante :

Octet 0 : Nombre de parties dans l'enveloppe (0..5).

Bit 7 = 1 → Répétition de l'enveloppe.

Octets 1..3 : Première section d'enveloppe.

Octet 1 : Pas de comptage.

Octet 2 : Taille du pas.

Octet 3 : Temps de pause entre deux pas.

Octets 4..6 : Deuxième section d'enveloppe.

Octets 7..9 : Troisième section d'enveloppe.

Octets 10..12 : Quatrième section d'enveloppe.

Octets 13..15 : Cinquième section d'enveloppe.

En relation : SOUND AMPL ENVELOPPE
SOUND T ADDRESS

SOUND A ADDRESS #BCC2

Donne l'adresse du buffer de définition de l'amplitude de volume.

Entrée : A contient le numéro d'amplitude (1..15).

Sortie : Si tout s'est bien passé :

Flag Carry = 1.

HL contient l'adresse de l'amplitude de volume.

BC contient la longueur de l'enveloppe.

Si le numéro d'enveloppe n'est pas correct :

Flag Carry = 0.

HL écrasés.

BC intacts.

A et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : SOUND AMPL ENVELOPPE
SOUND T ADDRESS

SOUND T ADDRESS #BCC5

Donne l'adresse du buffer de définition de l'amplitude de ton.

Entrée : A contient le numéro d'amplitude (1..15).

Sortie : Si tout s'est bien passé :

Flag Carry = 1.

HL contient l'adresse de l'amplitude de ton.

BC contient la longueur de l'enveloppe.

Si le numéro d'enveloppe n'est pas correct :

Flag Carry = 0.

HL écrasés.

BC intacts.

A et les autres flags écrasés.

Les autres registres sont intacts.

En relation : SOUND A ADDRESS
SOUND TONE ENVELOPPE

KL CHOKE OFF #BCC8

Remise à zéro du KERNEL (IT synchrones, timers ne concernant pas le générateur de son et le clavier).

Entrée : Aucune.

Sortie : B contient l'adresse de la ROM basse (foreground).

DE contient l'adresse du point d'entrée de la ROM basse.

C contient l'adresse de la RAM utilisateur (foreground).

AF et HL écrasés.

Les autres registres sont intacts.

En relation : MC BOOT PROGRAM
MC START PROGRAM

KL ROM WALK #BCCB

Initialisation des ROM Background.

Entrée : DE contient l'adresse du premier octet utilisable.

HL contient l'adresse du dernier octet utilisable.

Sortie : DE contient l'adresse du premier octet utilisable après init.

HL contient l'adresse du dernier octet utilisable après init.

AF et BC écrasés.

Les autres registres sont intacts.

En relation : KL FIND COMMAND
KL INIT BACK
KL LOG EXT

KL INIT BACK #BCCE

Initialise une ROM en particulier.

Entrée : C contient le ROM select de la ROM à initialiser (1..7).
DE contient l'adresse du premier octet utilisable.
HL contient l'adresse du dernier octet utilisable.

Sortie : DE contient l'adresse du premier octet utilisable après init.
HL contient l'adresse du dernier octet utilisable après init.
AF et B écrasés.
Les autres registres sont intacts.

En relation : KL FIND COMMAND
KL LOG EXT
KL ROM WALK

KL LOG EXT #BCD1

Définit une extension RSX dans le logiciel de base.

Entrée : BC contient l'adresse de la table de commande RSX en RAM.
HL contient l'adresse d'une zone de 4 octets en RAM.

Sortie : DE écrasés.
Les autres registres sont intacts.

En relation : KL FIND COMMAND
KL INIT BACK

KL FIND COMMAND #BCD4

Recherche d'une extension RSX en ROM pour l'exécuter.

Entrée : HL contient l'adresse de la commande à chercher.

Sortie : Si la RSX a été trouvée :
Flag Carry = 1.
C contient l'adresse de ROM select.
HL contient l'adresse de la routine.
Si la RSX n'a pas été trouvée :
Flag Carry = 0.
C et HL écrasés.

A, B et DE écrasés.
Les autres registres sont intacts.

En relation : KL INIT BACK
KL LOG EXT
KL ROM WALK
MC START PROGRAM

KL NEW FRAME FLY #BCD7

Initialise et positionne un bloc dans la file d'attente d'IT.

Cette file d'attente fait 9 octets de long et se trouve en RAM.

Les 7 derniers octets du bloc sont les paramètres B, C et DE de la commande KL INIT EVENT.

Entrée : HL contient l'adresse du bloc.

B contient le type d'interruption.

C contient le ROM select de la routine d'IT.

DE contient l'adresse de la routine d'IT.

Sortie : AF, DE et HL écrasés.

Les autres registres sont intacts.

En relation : KL ADD FRAME FLY

KL DEL FRAME FLY

KL INIT EVENT

KL ADD FRAME FLY #BCDA

Met un bloc dans la file d'attente d'IT.

Entrée : HL contient l'adresse du bloc de définition d'IT.

Sortie : AF, DE, HL écrasés.

Les autres registres sont intacts.

En relation : KL DEL FRAME FLY

KL INIT EVENT

KL NEW FRAME FLY

KL DEL FRAME FLY #BCDD

Sort un bloc de la file d'attente d'IT.

Entrée : HL contient l'adresse du bloc de définition d'IT.

Sortie : AF, DE et HL écrasés.

Les autres registres sont intacts.

En relation : KL ADD FRAME FLY

KL NEW FRAME FLY

KL NEW FAST TICKER #BCE0

Initialise et met un bloc dans la file d'attente d'IT rapide.

(Ces IT sont activées toutes les 1/300 sec.).

Entrée : HL contient l'adresse du bloc d'IT en RAM.

B contient le type d'IT.

C contient le ROM select de la routine d'IT.

DE contient l'adresse de la routine d'IT.

Sortie : AF, DE et HL écrasés.
Les autres registres sont intacts.

En relation : KL ADD FAST TICKER
KL ADD TICKER
KL DEL FAST TICKER
KL INIT EVENT
KL TIME PLEASE

KL ADD FAST TICKER #BCE3

Place un bloc dans la file d'attente d'IT rapide.

Entrée : HL contient l'adresse du bloc d'IT rapide.

Sortie : AF, DE et HL écrasés.
Les autres registres sont intacts.

En relation : KL ADD TICKER
KL DEL FAST TICKER
KL INIT EVENT
KL NEW FAST TICKER
KL TIME PLEASE

KL DEL FAST TICKER #BCE6

Dévalide un bloc dans la file d'attente d'IT rapide.

Entrée : HL contient l'adresse du bloc d'IT rapide.

Sortie : AF, DE et HL écrasés.
Les autres registres sont intacts.

En relation : KL ADD FAST TICKER
KL DEL TICKER
KL NEW FAST TICKER

KL ADD TICKER #BCE9

Met un bloc dans la file d'attente d'IT.

Entrée : HL contient l'adresse du bloc d'IT.
DE contient la valeur initiale du compteur.
BC contient la valeur de réinitialisation du compteur.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

Remarques :

Tous les 1/50 sec. la valeur du compteur est décrémentée. Si le compteur passe à zéro, le compteur est réinitialisé avec la valeur de réinitialisation et l'IT est générée.

En relation : KL ADD FAST TICKER
KL DEL TICKER
KL INIT EVENT

KL DEL TICKER #BCEC

Sort un bloc de la file d'attente d'IT.

Entrée : HL contient l'adresse du bloc d'IT.

Sortie : Si le bloc a été trouvé
 Flag Carry = 1 et
 DE contient la valeur du compteur.
 Si le bloc n'a pas été trouvé, Flag Carry = 0 et DE écrasé.
 A, HL et les autres flags écrasés.
 Les autres registres sont intacts.

En relation : KL ADD TICKER
KL DEL FAST TICKER

KL INIT EVENT #BCEF

Initialise un bloc d'IT.

Entrée : HL contient l'adresse du bloc d'IT.
 B contient le type d'IT.
 C contient le ROM select de la routine d'IT.
 DE contient l'adresse de la routine d'IT.

Sortie : HL contient l'adresse du bloc d'IT + 7.
 Les autres registres sont intacts.

Remarque :

Le type d'IT est défini comme suit :
 Bit 0 : Prochaine adresse.
 Bits 1..4 : IT synchrone prioritaire.
 Bit 5 : Obligatoirement à 0.
 Bit 6 : IT express.
 Bit 7 : IT asynchrone.

En relation : KL DEL SYNCHRONOUS
KL DISARM EVENT
KL EVENT
KL NEW FAST TICKER
KL NEW FRAME FLY
KL NEW TICKER
KL SYNC RESET

KL EVENT #BCF2

Arme une IT dans le bloc d'IT.

Entrée : HL contient l'adresse du bloc d'IT.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : KL INIT EVENT
KL NEXT SYNC
KL POLL SYNCHRONOUS
KL SYNC RESET

KL SYNC RESET #BCF5

Efface le bloc d'IT.
Entrée : Aucune.
Sortie : AF et HL écrasés.
Les autres registres sont intacts.

En relation : KL DEL SYNCHRONOUS
KL NEXT SYNC
KL POLL SYNCHRONOUS

KL DEL SYNCHRONOUS #BCF8

Sort une IT synchrone du bloc d'IT.
Entrée : HL contient l'adresse du bloc d'IT.
Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : KL DISARM EVENT
KL INIT EVENT
KL SYNC RESET

KL NEXT SYNC #BCFB

Si une IT synchrone se trouve dans le bloc d'IT dont la priorité est supérieure à la priorité actuelle, cette IT synchrone est sortie du bloc d'IT, et sa priorité devient la priorité courante.

Entrée : Aucune.

Sortie : Si une IT synchrone de priorité > priorité actuelle est trouvée :
Flag Carry = 1.
HL contient l'adresse du bloc d'IT.
A contient la priorité précédente.

Si aucune IT synchrone de prior. > prior. actuelle n'est trouvée :
Flag Carry = 0.
A et HL écrasés.
DE écrasés.
Les autres registres sont intacts.

En relation : KL DONE SYNC
 KL DO SYNC
 KL EVENT
 KL INIT EVENT
 KL POLL SYNCHRONOUS

KL DO SYNC #BCFE

Active une routine d'IT.

Entrée : HL contient l'adresse du bloc d'IT.

Sortie : AF, BC, DE et HL écrasés.
 Les autres registres sont intacts.

En relation : KL DONE SYNC
 KL NEXT SYNC

KL DONE SYNC #BD01

Permet de sortir correctement de l'activation d'une IT synchrone après appel à la routine KL DO SYNC.

Entrée : A contient l'ancienne priorité.
 HL contient l'adresse du bloc d'IT.

Sortie : AF, BC, DE et HL écrasés.
 Les autres registres sont intacts.

En relation : KL DO SYNC
 KL NEXT SYNC

KL EVENT DISABLE #BD04

Empêche les IT synchrones d'être exécutées, mais permet l'exécution d'IT synchrones express.

Entrée : Aucune.

Sortie : HL écrasés.
 Les autres registres sont intacts.

En relation : KL DISARM EVENT
 KL EVENT ENABLE
 KL NEXT SYNC
 KL POLL SYNCHRONOUS

KL EVENT ENABLE #BD07

Valide les IT synchrones normales.

Entrée : Aucune.

Sortie : HL écrasés.
Les autres registres sont intacts.

En relation : KL EVENT DISABLE
KL NEXT SYNC
KL POLL SYNCHRONOUS

KL DISARM EVENT #BD0A

Interdit les IT en positionnant les compteurs à des valeurs négatives.

Entrée : HL contient l'adresse du bloc d'IT.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : KL DEL SYNCHRONOUS
KL INIT EVENT

KL TIME PLEASE #BD0D

Donne le nombre de 1/300 sec. écoulés depuis la mise en route de la machine ou l'appui sur les touches CTRL + ENTER + BREAK.

Entrée : Aucune.

Sortie : DE et HL contiennent le décompte des 1/300 sec.
D est le poids le plus fort, L le poids le plus faible.

En relation : KL TIME SET

KL TIME SET #BD10

Met à jour le compteur de temps exprimé en 1/300 sec.

Entrée : DE et HL représentent la base de temps que l'on veut programmer.
D est le poids le plus fort et L le poids le plus faible.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : KL TIME PLEASE

MC START PROGRAM #BD16

Initialisation du système et exécution d'un programme.

Entrée : HL contient le point d'entrée.
C contient le ROM select (#FC..#FF).

Sortie : Aucune !

En relation : MC BOOT PROGRAM
RESET ENTRY (RTS 0).

MC WAIT FLYBACK #BD19

Attend un signal vidéo « FRAME FLYBACK ».

Ce signal indique le début d'une période où l'écran n'est pas en période d'affichage.

Entrée : Aucune.

Sortie : Tous les flags et registres sont intacts.
Le retour de cette routine s'effectue quand le FRAME FLYBACK est détecté.

En relation : KL ADD FRAME FLY

MC SET MODE #BD1C

Définit le MODE d'affichage (0, 1 ou 2).

Entrée : A contient le MODE demandé.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : SCR SET MODE

MC SCREEN OFFSET #BD1F

Définit le décalage du début de l'écran.

Entrée : A contient le poids fort de l'adresse de début d'écran, masqué par #C0.
HL contient le décalage demandé (masqué par #07FE).

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : SCR SET BASE
SCR SET OFFSET

MC CLEAR INKS #BD22

Définit le bord de l'écran et toutes les encres de la même couleur.

Entrée : DE contient l'adresse d'un vecteur d'encre qui a la forme :
Octet 0 : Couleur du BORDER,
Octet 1 : Couleur de toutes les encres.

Sortie : AF écrasés.

En relation : MC SET INKS

MC SET INKS #BD25

Définit les couleurs de toutes les encres et du BORDER.

Entrée : DE contient l'adresse du vecteur d'encre qui a la forme :

Octet 0 : Couleur du BORDER,

Octet 1 : Couleur de l'encre 1,

Octet 2 : Couleur de l'encre 2,

...

Octet 16 : Couleur de l'encre 16.

En relation : MC CLEAR INKS
SCR SET BORDER
SCR SET INK

MC RESET PRINTER #BD28

Positionne l'indirection de l'imprimante à sa valeur par défaut.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : MC WAIT PRINTER
MC PRINT CHAR

MC PRINT CHAR #BD2B

Essaye d'envoyer un caractère sur le port CENTRONICS.

Entrée : A contient le caractère à envoyer (bit 7 non envoyé).

Sortie : Si tout s'est bien passé, Flag Carry = 1.
Si l'imprimante n'a pas reçu le caractère, Flag Carry = 0.
A et les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : MC RESET PRINTER
MC WAIT PRINTER

MC BUSY PRINTER #BD2E

Teste si l'imprimante CENTRONICS est occupée ou non.

Entrée : Aucune.

Sortie : Si l'imprimante est occupée : Flag Carry = 1.
Sinon, Flag Carry = 0.
Les autres flags sont écrasés.
Les autres registres sont intacts.

En relation : MC SEND PRINTER

MC SEND PRINTER #BD31

Envoie un caractère à l'imprimante CENTRONICS.

Entrée : A contient le caractère à envoyer sur 7 bits (bit 7 ignoré).

Sortie : Flag Carry = 1.

A et les autres flags sont écrasés.

Les autres registres sont intacts.

En relation : MC BUSY PRINTER
MC PRINT CHAR

MC SOUND REGISTER #BD34

Envoie une donnée au chip sonore.

Entrée : A contient le numéro du registre à accéder.

C contient la donnée à envoyer.

Sortie : AF et BC écrasés.

Les autres registres sont intacts.

En relation : Aucune routine.

JMP RESTORE #BD37

Remet à jour le bloc standard de saut.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : GRA RESET
KM RESET
MC RESET PRINTER
SCR RESET
TXT RESET

INDIRECTIONS DU FIRMWARE

Ce groupe rassemble les points d'entrée du bloc d'indirection du Firmware. Ce bloc d'indirection occupe les adresses comprises entre \$BDCD et \$BDF3. Il consiste en une suite de JMP ou de RST (si une des ROM hautes ou la RAM basse doit être accédée).

IND: TXT DRAW CURSOR #BDCD

Affiche le bloc curseur à l'écran si l'affichage est possible (curseur validé et affiché).

Entrée : Aucune.

Sortie : AF effacé.

Les autres registres sont intacts.

En relation : TXT PLACE CURSOR
TXT UNDRAW CURSOR

IND: TXT UNDRAW CURSOR #BDD0

Efface le bloc curseur de l'écran si le curseur est validé et affiché.

Entrée : Aucune.

Sortie : AF effacé.

Les autres registres sont intacts.

En relation : TXT DRAW CURSOR
TXT REMOVE CURSOR

IND: TXT WRITE CHAR #BDD3

Affiche un caractère à l'écran à la position courante spécifiée.

Entrée : A = Caractère à afficher.

H = Colonne où afficher le caractère.

L = Ligne où afficher le caractère.

Sortie : AF, BC, DE et HL effacés.

Les autres registres sont intacts.

En relation : TXT OUTPUT
TXT UNWRITE
TXT WR CHAR

IND: TXT UNWRITE #BDD6

Essai de lecture d'un caractère sur l'écran à la position spécifiée.

Entrée : H = Colonne où lire le caractère.

L = Ligne où lire le caractère.

Sortie : Si un caractère a pu être lu :

- l'indicateur Carry est positionné à un ;
- A contient le code ASCII du caractère.

Si aucun caractère n'a pu être lu :

- l'indicateur Carry est positionné à zéro ;
- A est positionné à zéro.

BC, DE, HL et les indicateurs sont écrasés.

Les autres indicateurs sont intacts.

En relation TXT RD CHAR
 TXT WRITE CHAR

IND: TXT OUT ACTION #BDD9

Affiche un caractère ou un code de contrôle (#00 à #1F) à l'écran.

Entrée A = Caractère à afficher.

Sortie AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : TXT OUTPUT
 TXT WR CHAR

IND: GRA PLOT #BDDC

Affiche un point élémentaire à l'écran avec la couleur d'encre par défaut.

Entrée : DE = Abscisse du point à afficher.
 HL = Ordonnée du point à afficher.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA PLOT ABSOLUTE
 GRA PLOT RELATIVE
 GRA TEST
 SCR WRITE

IND: GRA TEST #BDDF

Donne la couleur du pixel spécifié s'il se trouve dans la fenêtre d'affichage.

Entrée : DE = Abscisse du point à afficher.
 HL = Ordonnée du point à afficher.

Sortie : A contient la couleur d'encre du point spécifié.
BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA PLOT
GRA TEST ABSOLUTE
GRA TEST RELATIVE
SCR READ

IND: GRA LINE #BDE2

Trace une ligne entre le point courant et le point spécifié en coordonnées absolues.

Entrée : DE = Abscisse du point final.
HL = Ordonnée du point final.

Sortie : AF, BC, DE et HL écrasés.
Les autres registres sont intacts.

En relation : GRA LINE ABSOLUTE
GRA LINE RELATIVE
SCR HORIZONTAL
SCR VERTICAL

IND: SCR READ #BDE5

Donne la couleur du pixel (accédé par son adresse d'écran et un masque d'octet) s'il se trouve dans la fenêtre d'affichage.

Entrée : HL = Adresse d'écran du pixel.
C = Masque pour accéder au pixel dans l'octet.

Sortie : A = Couleur du pixel.
Indicateurs écrasés.
Les autres registres sont intacts.

En relation : GRA TEST
SCR WRITE

IND: SCR WRITE #BDE8

Affiche un pixel à l'adresse d'écran spécifiée, en utilisant le masque d'octet spécifié.

Le pixel est affiché dans le mode d'affichage courant (Forcé, XOR, AND ou OR).

Entrée : HL = Adresse d'écran où afficher le pixel.
C = Masque pour accéder au pixel.
B = Encre d'affichage du pixel.

Sortie : AF écrasé.
Les autres registres sont intacts.

En relation : GRA PLOT
GRA ACCESS
SCR PIXELS
SCR READ

IND: SCR MODE CLEAR #BDEB

Efface l'écran avec l'encre 0.

Entrée : Aucune.

Sortie : AF, BC, DE et HL effacés.
Les autres registres sont intacts.

En relation : SCR CLEAR
SCR SET MODE

IND: KM TEST BREAK #BDEE

Teste si la touche ESC est pressée ou si les touches ESC, SHIFT et CONTROL sont pressées simultanément.

Si la touche ESC n'est pas pressée, aucune action n'est effectuée. Dans le cas contraire, un événement Break est mémorisé. Si les touches ESC, SHIFT et CONTROL sont pressées simultanément, le système est initialisé.

Entrée : Interruptions dévalidées.
C = Etat des touches SHIFT et CONTROL
(bit 7 = 1 → touche CONTROL pressée,
bit 5 = 1 → touche SHIFT pressée).

Sortie : AF et HL écrasés.
Les autres registres sont intacts.

En relation : KM BREAK EVENT

IND: MC WAIT PRINTER #BDF1

Attend que le port CENTRONICS ne soit plus occupé pour émettre un caractère. Si le port CENTRONICS est occupé pendant un temps trop long, un « time-out » se produit et aucun caractère n'est émis.

Entrée : A = Caractère à émettre.

Sortie : Si le caractère a pu être émis :
Indicateur Carry = 1.
Si le caractère n'a pu être émis :
Indicateur Carry = 0.
A et BC écrasés.
Les autres registres sont intacts.

En relation : MC BUSY PRINTER
MC PRINT CHAR
MC SEND PRINTER

BLOCS D'INDIRECTION DU KERNEL

Ce bloc d'indirection s'étend entre les adresses #B900 et #B921. Il permet de valider/dévalider les ROM de l'Amstrad. Les entrées de ce bloc d'indirection se trouvent en mémoire RAM, mais ne doivent pas être modifiées sous peine d'altérer le fonctionnement de l'ordinateur.

HI: KL U ROM ENABLE #B900

Valide une des ROM supérieures (située aux adresses supérieures à #C000). Une lecture à une telle adresse après avoir activé la routine KL U ROM ENABLE donne le contenu de la mémoire ROM correspondante.

Entrée : Aucune.

Sortie : A = Etat précédent de la ROM.
Indicateurs écrasés.
Les autres registres sont intacts.

En relation : KL L ROM ENABLE
KL ROM RESTORE
KL ROM SELECT
KL U ROM DISABLE

HI: KL U ROM DISABLE #B903

Dévalide une des ROM supérieures (située aux adresses supérieures à #C000). Une lecture à une telle adresse après avoir activé la routine KL U ROM DISABLE donne le contenu de la mémoire RAM correspondante (souvent la mémoire d'écran).

Remarque : il est possible de dévalider la ROM supérieure lorsqu'un programme est en train de s'exécuter dans la ROM supérieure.

Entrée : Aucune.

Sortie : A = Etat précédent de la ROM.
Indicateurs écrasés.
Les autres registres sont intacts.

En relation : KL L ROM ENABLE
KL ROM RESTORE
KL U ROM ENABLE

HI: KL L ROM ENABLE #B906

Valide la ROM inférieure (située aux adresses inférieures à #3FFF). Une lecture à une telle adresse après avoir activé la routine KL L ROM ENABLE donne le contenu de la mémoire ROM correspondante.

Remarque : il est impossible de dévalider la ROM inférieure lorsqu'un programme est en train de s'exécuter dans la ROM inférieure.

Entrée : Aucune.

Sortie : A = Etat précédent de la ROM.
Indicateurs écrasés.
Les autres registres sont intacts.

En relation : KL L ROM DISABLE
KL ROM RESTORE
KL U ROM ENABLE

HI: KL L ROM DISABLE #B909

Dévalide la ROM inférieure (située aux adresses inférieures à #3FFF).
Une lecture à une telle adresse après avoir activé la routine KL L ROM
DISABLE donne le contenu de la mémoire RAM correspondante.

Remarque : il est impossible de dévalider la ROM inférieure lorsqu'un pro-
gramme est en train de s'exécuter dans la ROM inférieure.

Entrée : Aucune.

Sortie : A = Etat précédent de la ROM.
Indicateurs écrasés.
Les autres registres sont intacts.

En relation : KL L ROM ENABLE
KL ROM RESTORE
KL U ROM DISABLE

HI: KL ROM RESTORE #B90C

Restitue l'état de validation des ROM précédents. Cet état est retourné
par une des routines suivantes :

- KL L ROM ENABLE
- KL U ROM DISABLE
- KL ROM SELECT
- KL L ROM ENABLE
- KL L ROM DISABLE

Entrée : A = Etat précédent des ROM.

Sortie : AF écrasés.
Les autres registres sont intacts.

En relation : KL U ROM DISABLE
KL L ROM ENABLE
KL ROM SELECT
KL U ROM DISABLE
KL U ROM ENABLE

HL: KL ROM SELECT #B90F

Sélectionne une des ROM supérieures et valide les ROM supérieures.

Entrée : C = Adresse de la ROM à sélectionner.

Sortie : C = Adresse de la ROM précédemment sélectionnée.

B = Etat précédent de la ROM.

AF écrasés.

Les autres indicateurs et registres sont intacts.

En relation : KL CURR SELECTION

KL PROBE ROM

KL ROM DESELECT

KL ROM RESTORE

HI: KL CURR SELECTION #B912

Indique quelle ROM supérieure est actuellement utilisée.

Entrée : Aucune.

Sortie : A = Adresse de ROM SELECT de la ROM actuellement utilisée.

Les autres registres et indicateurs sont intacts.

En relation : KL PROBE ROM

KL ROM SELECT

HI: KL PROBE ROM #B915

Indique le type et la version de ROM utilisée.

Entrée : C = Adresse de ROM SELECT de la ROM à tester.

Sortie : A = Type de ROM utilisée,

(0 = ROM en premier plan,

1 = ROM en arrière plan,

2 = ROM d'extension en premier plan,

#80 = ROM basic).

L = Numéro de ROM utilisée.

H = Numéro de version de la ROM utilisée.

B et les indicateurs sont écrasés.

Les autres registres sont intacts.

En relation : KL ROM SELECT

KL CURR SELECTION

HI: KL ROM DESELECT #B918

Restitue l'état précédent d'utilisation des ROM supérieures (avant que KL ROM SELECT n'ait été utilisé).

Entrée : C = Adresse précédente de ROM SELECT.

B = Adresse précédente des ROM supérieures.

Sortie : C = Adresse de ROM SELECT de la ROM actuellement sélectionnée.
 B = écrasé.
 Les autres registres et indicateurs sont intacts.

En relation : KL CURR SELECTION
 KL ROM RESTORE
 KL ROM SELECT

 HI: KL LDIR #B91B

Instruction LDIR (LoaD Increment and Repeat) avec les ROM inférieures et supérieures dévalidées.

Entrée : BC, DE et HL en accord avec l'instruction LDIR.

Sortie : F, BC, DE et HL en accord avec l'instruction LDIR.
 Les autres registres sont intacts.

En relation : KL LDDR
 RST 4 (RAM LAM)

 HI: KL LDDR #B91E

Instruction LDDR (LoaD Decrement and Repeat) avec les ROM inférieures et supérieures dévalidées.

Entrée : BC, DE et HL en accord avec l'instruction LDDR.

Sortie : F, BC, DE et HL en accord avec l'instruction LDDR.
 Les autres registres sont intacts.

En relation : KL LDIR
 RST 4 (RAM LAM)

 HI: KL POLL SYNCHRONOUS #B921

Teste si un événement de priorité supérieure à la priorité actuelle est survenu (se trouve dans la queue).

Entrée : Aucune

Sortie : Si un événement de priorité supérieure existe :
 Indicateur Carry = 1.
 Si aucun événement de priorité supérieure n'existe :
 Indicateur Carry = 0.
 A et les autres indicateurs sont écrasés.
 Les autres registres sont intacts.

En relation : KL EVENT
 KL DONE SYNC
 KL DO SYNC
 K NEXT SYNC

Ce bloc d'indirection s'étend entre les adresses #0000 et #003F. Il rassemble des routines et des RESTART. Généralement, les routines qu'il renferme sont accessibles aussi bien en RAM qu'en ROM (la ROM inférieure peut donc être validée ou dévalidée lors de l'appel d'une de ces routines).

Remarque : L'instruction RST 6 est réservée à l'utilisateur.

L'instruction RST 7 est réservée aux interruptions et n'est pas utilisable dans un programme.

LOW: RESET ENTRY #0000 RST 0

Routine d'initialisation de l'Amstrad. Cette routine est exécutée à chaque mise sous tension de l'ordinateur, ou à chaque saut à l'instruction #0000 ou RST 0 (par exemple, lors de l'appui simultané sur les touches CONTROL, SHIFT et ESCape). Cette routine initialise le hard et le firmware.

Entrée : Aucune.

Sortie : Aucune

En relation : MC START PROGRAM

LOW: LOW JUMP #0008 RST 1

Saut à une adresse située dans la RAM ou la ROM située entre les adresses #0000 et #3FFF. Cette instruction doit être suivie d'une adresse sur un mot (2 octets).

La signification des bits du mot qui suit un LOW JUMP est la suivante :

- Les bits 0 à 13 représentent l'adresse à accéder.
- Le bit 15, s'il est à un dévalide la ROM supérieure,
s'il est à zéro valide la ROM supérieure.
- Le bit 14, s'il est à un dévalide la ROM inférieure,
s'il est à zéro valide la ROM inférieure.

Entrée : Les valeurs des registres et indicateurs sont passées à la routine activée par le RST 1.

Sortie : Les valeurs des registres et indicateurs sont restituées par la routine activée par le RST 1.

En relation : RST 3 (FAR CALL)
RST 5 (FIRM JUMP)
KL FAR ICALL
KL FAR PCHL
KL LOW PCHL

LOW: KL LOW PCHL #000B

Appel à une routine qui se trouve dans la ROM ou la RAM inférieure. L'adresse de la routine à activer est codée sur les bits 0 à 13. Les bits 14 et 15 ont la signification suivante :

- Le bit 15, s'il est à un dévalide la ROM supérieure, s'il est à zéro valide la ROM supérieure,
- Le bit 14, s'il est à un dévalide la ROM supérieure, s'il est à zéro valide la ROM inférieure.

Entrée : HL = Adresse de la routine à activer.

Les registres et indicateurs sont passés à la routine tels qu'ils sont au moment de l'appel.

Sortie : Les registres et indicateurs sont restitués tels qu'ils sont à la sortie de la routine qui a été activée par KL LOW PCHL.

En relation : KL FAR ICALL
KL FAR PCHL
LOW JUMP (RST 1)
Instruction PCHL

LOW: PCBC INSTRUCTION #000E

Cette routine effectue un saut à l'adresse contenue dans la paire de registres BC. Les registres et indicateurs sont passés tels qu'ils sont lors de l'appel à cette routine.

Entrée : (BC) = Adresse à exécuter.
Registres et indicateurs.

Sortie : Registres et indicateurs modifiés par la routine appelée.

En relation : KL FAR PCHL
KL LOW PCHL
KL SIDE PCHL
Instruction PCDE
Instruction PCHL

LOW: SIDE CALL #0010 RST 2

Accès à une des ROM d'extension. Cette instruction est suivie de deux octets qui indiquent le déplacement après l'adresse de base #C000 de la routine à accéder en ROM. Les bits 14 et 15, ajoutés à l'adresse de ROM SELECT de la première ROM, donnent l'adresse de ROM SELECT de la ROM à accéder.

Entrée : Les registres et indicateurs sont passés tels qu'ils sont lors de l'appel au RST 2.

Sortie : IY intact.

Les autres registres et indicateurs sont positionnés par la routine activée par le RST 2.

En relation : RST 3 (FAR CALL)
KL SIDE PCHL

LOW: KL SIDE PCHL #0013

Accès à une des ROM d'extension. L'adresse de la ROM à accéder est spécifiée dans la paire de registres HL (déplacement après l'adresse de base #C000 de la routine à accéder en ROM). Les bits 14 et 15 de HL, ajoutés à l'adresse de ROM SELECT de la première ROM, donnent l'adresse de ROM SELECT de la ROM à accéder.

Entrée : HL = Adresse de la routine à activer.

Les registres et indicateurs sont passés tels qu'ils sont lors de l'appel à KL SIDE PCHL.

Sortie : IY intact.

Les autres registres et indicateurs sont positionnés par la routine activée par le RST2.

En relation : RST 3 (FAR CALL)

KL FAR ICALL

KL FAR PCHL

LOW: PCDE INSTRUCTION #0016

Cette routine effectue un saut à l'adresse contenue dans la paire de registres DE. Les registres et indicateurs sont passés tels qu'ils sont lors de l'appel de cette routine.

Entrée : (DE) = Adresse à exécuter.

Registres et indicateurs.

Sortie : Registres et indicateurs modifiés par la routine appelée.

En relation : KL FAR PCHL

KL LOW PCHL

KL SIDE PCHL

Instruction PCBC

Instruction PCHL

LOW: FAR CALL #0018 RST 3

Cette instruction permet d'activer une routine qui se trouve n'importe où en RAM ou en ROM.

Trois octets suivent l'instruction RST 3. Les deux premiers indiquent l'adresse de la routine à accéder. Le troisième indique le ROM SELECT, codé de la façon suivante :

0..251 : ROM supérieure validée, ROM inférieure dévalidée,

252 : validation ROM supérieure inchangée,
ROM inférieure validée,

253 : validation ROM supérieure inchangée,
ROM inférieure dévalidée,

254 : validation ROM supérieure inchangée,
ROM inférieure validée,

255 : validation ROM supérieure inchangée,
ROM inférieure dévalidée.

Entrée : Registres et indicateurs.

Sortie : IY inchangé.

Registres et indicateurs modifiés par la routine appelée.

En relation : KL FAR ICALL
KL FAR PCHL
RST 1 (LOW JUMP)
RST 2 (SIDE CALL)

LOW: KL FAR PCHL #001B

Appel d'une routine située en RAM ou en ROM.

Entrée : HL = Adresse de la routine à activer.

C = Octet de ROM SELECT.

Les autres registres et indicateurs sont passés tels quels à la routine activée.

Sortie : IY inchangé.

Registres et indicateurs modifiés par la routine appelée.

L'octet de ROM SELECT est codé dans le registre C de la façon suivante :

0..251 : ROM supérieure validée, ROM inférieure dévalidée,
252 : validation ROM supérieure inchangée,
ROM inférieure validée,
253 : validation ROM supérieure inchangée,
ROM inférieure dévalidée,
254 : validation ROM supérieure inchangée,
ROM inférieure validée,
255 : validation ROM supérieure inchangée,
ROM inférieure dévalidée.

En relation : RST 3 (FAR CALL)
KL FAR ICALL
KL FAR PCHL
KL LOW PCHL
KL SIDE PCHL

LOW: PCHL INSTRUCTION #001E

Cette routine effectue un saut à l'adresse contenue dans la paire de registres HL. Les registres et indicateurs sont passés tels qu'ils sont lors de l'appel à cette routine.

Entrée : (HL) = Adresse à exécuter.

Registres et indicateurs.

Sortie : Registres et indicateurs modifiés par la routine appelée.

En relation : KL FAR PCHL
KL LOW PCHL
KL SIDE PCHL
Instruction PCBC
Instruction PCDE

LOW: RAM LAM #0020 RST 4

Cette instruction permet de lire un octet en mémoire, toutes les ROM étant dévalidées. RAM LAM est équivalent à l'instruction LD A,(HL), à ceci près que RAM LAM lit toujours en RAM, quels que soient les états de validation des diverses ROM.

Entrée : HL = Adresse de la mémoire RAM à lire.

Sortie : A = Valeur lue en RAM à l'adresse pointée par HL
Les autres registres et indicateurs sont intacts.

En relation : KL LDDR
KL LDIR

LOW: KL FAR ICALL #0023

Cette instruction permet d'accéder à une routine située en RAM ou en ROM.

Entrée : HL = Adresse de la routine à activer.

Les registres et indicateurs sont passés à la routine activée tels qu'ils sont au moment de l'appel à KL FAR ICALL.

Sortie : IY inchangé.

Les autres registres et indicateurs sont initialisés par la routine activée.

L'adresse pointée par HL contient 3 octets dont la signification est la suivante :

- les deux premiers octets contiennent l'adresse de la routine à activer ;
- le troisième octet contient le ROM SELECT, codé de la façon suivante :

- 0..251 : ROM supérieure validée, ROM inférieure dévalidée,
- 252 : validation ROM supérieure inchangée,
ROM inférieure validée,
- 253 : validation ROM supérieure inchangée,
ROM inférieure dévalidée,
- 254 : validation ROM supérieure inchangée,
ROM inférieure validée,
- 255 : validation ROM supérieure inchangée,
ROM inférieure dévalidée.

En relation : KL FAR CALL
KL FAR PCHL

LOW: FIRM JUMP #0028 RST 5

Saut à une adresse de la ROM inférieure. L'instruction RST 5 est suivie d'un octet dans lequel se trouve l'adresse de la routine à accéder (entre #0000 et #3FFF).

Entrée : Registres et indicateurs.

Sortie : Registres et indicateurs modifiés par la routine appelée.

En relation : RST 1 (LOW JUMP)

LOW: USER RESTART #0030 RST 6

Les octets compris entre #0030 et #0037 sont réservés à l'utilisateur qui peut y mettre les valeurs qu'il veut, par exemple pour personnaliser le jeu d'instructions du Z80 en fabriquant une nouvelle instruction.

Si la ROM inférieure est validée pendant l'exécution d'une instruction RST 6, le firmware dévalide la ROM inférieure et donne le contrôle au programme situé en #0030. Si la ROM inférieure est dévalidée lors de l'exécution d'une instruction RST 6, le firmware donne le contrôle au programme situé en #0030.

Si les octets #0030 à #0037 ne sont pas modifiés par l'utilisateur, l'exécution d'une instruction RST 6 est équivalente à un SYSTEM RESET (RST 0).

Entrée : Définie par l'utilisateur.

Sortie : Définie par l'utilisateur.

LOW: INTERRUPT ENTRY #0038 RST 7

Point d'entrée d'une interruption Hard en mode d'interruption 1.

Entrée : Aucune.

Sortie : Tous les registres et les indicateurs sont intacts.

En relation : EXT INTERRUPT

LOW: EXT INTERRUPT #003B

Traitement des interruptions externes. Les octets #003B à #003F peuvent être définis par l'utilisateur dans le but de personnaliser le traitement de ses interruptions externes. Lorsqu'une interruption externe est détectée, la ROM inférieure est dévalidée, et le contrôle est passé au programme qui se trouve à l'adresse #003B.

Entrée : Aucune.

Sortie : AF, BC, DE et HL écrasés.

Les autres registres sont intacts.

En relation : INTERRUPT ENTRY
KL EVENT

Résumé des fonctions du FIRMWARE

Fonctions clavier

KM INITIALISE	#BB00	Initialisation du Key Manager
KM RESET	#BB03	RAZ du Key Manager
KM WAIT CHAR	#BB06	Attend un caractère au clavier
KM READ CHAR	#BB09	Lit « au vol » un caractère au clavier
KM CHAR RETURN	#BB0C	Sauve un caractère pour READ ou WAIT
KM SET EXPAND	#BB0F	Valide une chaîne d'expansion
KM GET EXPAND	#BB12	Lit un caractère d'une chaîne d'expansion
KM EXP BUFFER	#BB15	Déclare un buffer d'expansion
KM WAIT KEY	#BB18	Attend l'appui sur une touche du clavier
KM READ KEY	#BB1B	Lit au vol une touche particulière du clavier
KM TEST KEY	#BB1E	Teste si une touche particulière est pressée
KM GET STATE	#BB21	Teste l'état de CAPS LOCK et SHIFT LOCK
KM GET JOYSTICK	#BB24	Teste l'état du Joystick
KM SET TRANSLATE	#BB27	Positionne l'entrée de la table de translation
KM GET TRANSLATE	#BB2A	Attend une entrée dans la table de translation
KM SET SHIFT	#BB2D	Force l'appui sur la touche SHIFT
KM GET SHIFT	#BB30	Attend un caractère et le translate dans la table SHIFT
KM SET CONTROL	#BB33	Force l'appui sur la touche CONTROL
KM GET CONTROL	#BB36	Attend un caractère et le translate dans la table CONTROL
KM SET REPEAT	#BB39	Indique si une touche peut être répétée
KM GET REPEAT	#BB3C	Teste si une touche fonctionne en Auto-Repeat

KM SET DELAY	# BB3F	Fixe le temps d'attente de l'Auto-Repeat
KM GET DELAY	# BB42	Lit le temps d'attente de l'Auto-Repeat
KM ARM BREAK	# BB45	Arme le mécanisme de BREAK
KM DISARM BREAK	# BB48	Désarme le mécanisme de BREAK
KM BREAK EVENT	# BB4B	Provoque un déroutement sur BREAK

Fonctions écran en mode texte

TXT INITIALISE	# BB4E	Initialisation de l'écran
TXT RESET	# BB51	Initialisation des indirections de l'écran
TXT VDU ENABLE	# BB54	Permet d'afficher un caractère par TXT OUTPUT ou TXT WR CHAR
TXT VDU DISABLE	# BB57	Interdit l'affichage de caractères par TXT OUTPUT ou TXT WR CHAR
TXT OUTPUT	# BB5A	Affiche un caractère, y compris les codes de contrôle
TXT WR CHAR	# BB5D	Affiche un caractère, sauf les codes de contrôle
TXT RD CHAR	# BB60	Lit un caractère à l'écran
TXT SET GRAPHIC	# BB63	Switch option affichage graphique
TXT WIN ENABLE	# BB66	Dimensions de la fenêtre texte
TXT GET WINDOW	# BB69	Donne la taille de la fenêtre texte
TXT CLEAR WINDOW	# BB6C	Efface la fenêtre
TXT SET COLUMN	# BB6F	Position horizontale curseur
TXT SET ROW	# BB72	Position verticale curseur
TXT SET CURSOR	# BB75	Positions horizontale et verticale du curseur
TXT GET CURSOR	# BB78	Donne la position courante du curseur
TXT CUR ENABLE	# BB7B	Affichage bloc curseur autorisé

TXT CUR DISABLE	#BB7E	Affichage bloc curseur non autorisé
TXT CUR ON	#BB81	Bloc curseur allumé
TXT CUR OFF	#BB84	Bloc curseur éteint
TXT VALIDATE	#BB87	Teste une position d'écran par rapport à la fenêtre
TXT PLACE CURSOR	#BB8A	Place un bloc curseur à la position courante du curseur
TXT REMOVE CURSOR	#BB8D	Efface le bloc curseur de sa position actuelle
TXT SET PEN	#BB90	Positionne la couleur d'affichage en mode texte
TXT GET PEN	#BB93	Donne la couleur d'affichage en mode texte
TXT SET PAPER	#BB96	Positionne la couleur du papier en mode texte
TXT GET PAPER	#BB99	Donne la couleur du papier en mode texte
TXT INVERSE	#BB9C	Inversion vidéo
TXT SET BACK	#BB9F	Modes opaque/transparent
TXT GET BACK	#BBA2	Donne le mode d'affichage (opaque ou transparent)
TXT GET MATRIX	#BBA5	Indique si un caractère est redéfini
TXT SET MATRIX	#BBA8	Permet de redéfinir un caractère
TXT SET M TABLE	#BBAB	Initialise la matrice de redéfinition d'un caractère
TXT GET M TABLE	#BBAE	Donne l'adresse de la table de redéfinition d'un caractère
TXT GET CONTROLS	#BBB1	Donne l'adresse de la table de contrôle des flux vidéo
TXT STR SELECT	#BBB4	Sélectionne un flux vidéo
TXT SWAP STREAM	#BBB7	Echange l'état de deux flux vidéo

Fonctions écran en mode graphique

GRA INITIALISE	#BBBA	Initialisation de l'écran graphique
GRA RESET	#BBBD	Initialisation des indirections graphiques

GRA MOVE ABSOLUTE	#BBC0	Position absolue graphique courante
GRA MOVE RELATIVE	#BBC3	Position relative graphique courante
GRA ASK CURSOR	#BBC6	Donne la position absolue graphique courante
GRA SET ORIGIN	#BBC9	Définit l'origine de l'écran
GRA GET ORIGIN	#BBCC	Renvoie l'origine de l'écran
GRA WIN WIDTH	#BBCF	Fixe la largeur de la fenêtre
GRA WIN HEIGHT	#BBD5	Fixe la hauteur de la fenêtre
GRA GET W WIDTH	#BBD8	Donne les abscisses des côtés de la fenêtre
GRA CLEAR WINDOW	#BBDB	Efface la fenêtre graphique
GRA SET PEN	#BBDE	Définit la couleur de tracé graphique
GRA GET PEN	#BBE1	Donne la couleur de tracé graphique
GRA SET PAPER	#BBE4	Définit la couleur de fond graphique
GRA GET PAPER	#BBE7	Donne la couleur de fond graphique
GRA PLOT ABSOLUTE	#BBEA	Affiche un pixel à une position absolue
GRA PLOT RELATIVE	#BBED	Affiche un pixel à une position relative
GRA TEST ABSOLUTE	#BBF0	Donne la couleur d'un point exprimé en coordonnées absolues
GRA TEST RELATIVE	#BBF3	Donne la couleur d'un point exprimé en coordonnées relatives
GRA LINE ABSOLUTE	#BBF6	Trace une ligne entre la position courante du curseur et une position absolue
GRA LINE RELATIVE	#BBF9	Trace une ligne entre la position courante du curseur et une position relative
GRA WR CHAR	#BBFC	Affiche un caractère à la position graphique courante

Fonctions écran en mode graphique ou non

SCR INITIALISE	#BBFF	Initialisation de l'écran
SCR RESET	#BC02	Initialisation des indirections de l'écran
SCR SET OFFSET	#BC05	Définit le décalage de début d'écran
SCR SET BASE	#BC08	Définit l'adresse basse de la mémoire d'écran
SCR GET LOCATION	#BC0B	Donne l'adresse de l'écran et son décalage
SCR SET MODE	#BC0E	Définit le mode d'affichage de l'écran (0, 1 ou 2)
SCR GET MODE	#BC11	Donne le mode d'affichage (0, 1 ou 2)
SCR CLEAR	#BC14	Efface l'écran avec l'encre 0
SCR CHAR LIMITS	#BC17	Donne les limites de l'écran
SCR CHAR POSITION	#BC1A	Convertit une position graphique en adresse d'écran
SCR DOT POSITION	#BC1D	Convertit des coordonnées graphiques absolues en adresse écran
SCR NEXT BYTE	#BC20	Calcule l'adresse d'écran située à droite de l'octet pointé
SCR PREV BYTE	#BC23	Calcule l'adresse d'écran située à gauche de l'octet pointé
SCR NEXT LINE	#BC26	Calcule l'adresse d'écran de l'octet située sur la ligne suivant la ligne courante
SCR PREV LINE	#BC29	Calcule l'adresse d'écran de l'octet située sur la ligne précédent la ligne courante
SCR INK ENCODE	#BC2C	Conversion d'une couleur d'encre
SCR INK DECODE	#BC2F	Décode une couleur d'encre
SCR SET INK	#BC32	Définit les deux couleurs d'une encre
SCR GET INK	#BC35	Donne les deux couleurs d'une encre
SCR SET BORDER	#BC38	Définit les couleurs du bord de l'écran
SCR GET BORDER	#BC3B	Donne les couleurs du bord de l'écran
SCR SET FLASHING	#BC3E	Définit les périodes d'affichage de chaque couleur en mode flash

SCR GET FLASHING	# BC41	Donne les périodes d'affichage de chaque couleur en mode flash
SCR FILL BOX	# BC44	Remplit une surface rectangulaire
SCR FLOOD BOX	# BC47	Remplit une surface rectangulaire
SCR CHAR INVERT	# BC4A	Inversion vidéo d'un caractère
SCR HW ROLL	# BC4D	Scrolling haut ou bas de 8 pixels de tout l'écran
SCR SW ROLL	# BC50	Scrolling haut ou bas de 8 pixels d'une partie de l'écran
SCR UNPACK	# BC53	Conversion d'une matrice en un caractère
SCR REPACK	# BC56	Conversion d'un caractère en une matrice
SCR ACCESS	# BC59	Définit le mode d'affichage graphique (Normal, AND, OR ou XOR)
SCR PIXELS	# BC5C	Affiche un pixel sur l'écran
SCR HORIZONTAL	# BC5F	Trace une ligne horizontale
SCR VERTICAL	# BC62	Trace une ligne verticale

Fonctions cassette ou disquette

CAS INITIALISE	# BC65	Initialisation du driver cassette
CAS SET SPEED	# BC68	Initialiation de la vitesse du lecteur de cassettes
CAS NOISY	# BC6B	Switch des messages du lecteur de cassettes
CAS START MOTOR	# BC6E	Met en marche le moteur du lecteur de cassettes
CAS STOP MOTOR	# BC71	Arrête le moteur du lecteur de cassettes
CAS RESTORE MOTOR	# BC74	Restitue l'état du moteur du lecteur de cassettes
CAS IN OPEN	# BC77	Ouvre un fichier en lecture
CAS IN CLOSE	# BC7A	Ferme un fichier ouvert en lecture
CAS IN ABANDON	# BC7D	Ferme un fichier avant de l'avoir totalement parcouru
CAS IN CHAR	# BC80	Lit un caractère dans un fichier
CAS IN DIRECT	# BC83	Lit le fichier ouvert
CAS RETURN	# BC86	Met le caractère lu par CAS IN CHAR dans un buffer
CAS TEST EOF	# BC89	Teste si le caractère EOF a été rencontré

CAS OUT OPEN	# BC8C	Ouvre un fichier en écriture
CAS OUT CLOSE	# BC8F	Ferme un fichier ouvert en écriture
CAS OUT ABANDON	# BC92	Ferme un fichier avant de l'avoir totalement parcouru
CAS OUT CHAR	# BC95	Ecrit un caractère dans un fichier
CAS OUT DIRECT	# BC98	Ecrit le contenu d'un buffer dans un fichier ouvert en écriture
CAS CATALOG	# BC9B	Liste des fichiers d'une cassette
CAS WRITE	# BC9E	Ecrit un enregistrement
CAS READ	# BCA1	Lit un enregistrement
CAS CHECK	# BCA4	Vérifie un enregistrement

Fonctions sonores

SOUND RESET	# BCA7	Initialisation du circuit sonore
SOUND QUEUE	# BCAA	Ajoute un son dans la queue sonore
SOUND CHECK	# BCAD	Vérifie si la queue sonore est pleine ou non
SOUND ARM EVENT	# BCB0	Arme un bloc de données sonores
SOUND RELEASE	# BCB3	Exécute un son mis en attente
SOUND HOLD	# BCB6	Stoppe l'émission du son courant
SOUND CONTINUE	# BCB9	Continue un son stoppé
SOUND AMPL VOLUME	# BCBC	Règle l'amplitude du volume
SOUND TONE ENVELOPPE	# BCBF	Définit une enveloppe de ton
SOUND A ADDRESS	# BCC2	Donne l'adresse du buffer de définition de l'amplitude de volume
SOUND T ADDRESS	# BCC5	Donne l'adresse du buffer de définition de l'amplitude de ton

Fonctions Kernel

KL CHOKE OFF	# BCC8	Initialisation du Kernel
KL ROM WALK	# BCCB	Initialisation des ROM basses
KL INIT BACK	# BCCE	Initialisation d'une ROM particulière
KL LOG EXT	# BCD1	Définit une extension RSX
KL FIND COMMAND	# BCD4	Recherche d'une extension RSX pour l'exécuter

KL NEW FRAME FLY	#BCD7	Initialise un bloc dans la file d'interruptions
KL ADD FRAME FLY	#BCDA	Place un bloc dans la file d'attente d'interruptions
KL DEL FRAME FLY	#BCDD	Sort un bloc de la file d'attente d'interruptions
KL NEW FAST TICKER	#BCE0	Initialise en positionnant un bloc dans la file d'attente d'interruptions rapide
KL ADD FAST TICKER	#BCE3	Place un bloc dans la file d'attente d'interruptions rapide
KL DEL FAST TICKER	#BCE6	Dévalide un bloc dans la file d'attente d'interruptions rapide
KL ADD TICKER	#BCE9	Place un bloc dans la file d'attente d'interruptions
KL DEL TICKER	#BCEC	Efface un bloc de la file d'attente d'interruptions
KL INIT EVENT	#BCEF	Initialise un bloc d'interruption
KL EVENT	#BCF2	Arme une interruption
KL SYNC RESET	#BCF5	Efface le bloc d'interruption
KL DEL SYNCHRONOUS	#BCF8	Efface une interruption synchrone du bloc d'interruption
KL NEXT SYNC	#BCFB	Efface une interruption synchrone du bloc d'interruption dans certaines conditions
KL DO SYNC	#BCFE	Active une routine d'interruption
KL DONE SYNC	#BD01	Finit une routine d'interruption
KL EVENT DISABLE	#BD04	Interdit l'exécution d'IT synchrones
KL EVENT ENABLE	#BD07	Autorise l'exécution d'IT synchrones
KL DISARM EVENT	#BD0A	Interdit certaines interruptions
KL TIME PLEASE	#BD0D	Donne le temps écoulé depuis la mise sous tension
KL TIME SET	#BD10	Initialise le compteur de temps écoulé depuis la mise sous tension

Fonctions hardware

MC START PROGRAMM	#BD13	Initialisation du système et activation d'un programme
MC WAIT FLYBACK	#BD16	Attend un signal « FRAME FLYBACK »
MC SET MODE	#BD19	Définit le mode d'affichage (0, 1 ou 2)
MC SCREEN OFFSET	#BD1C	Définit le décalage de début d'écran
MC CLEAR INKS	#BD1F	Initialise toutes les encres à la même valeur
MC SET INKS	#BD22	Définit toutes les couleurs
MC RESET PRINTER	#BD25	Initialisation de l'indirection d'imprimante
MC PRINT CHAR	#BD28	Envoie un caractère sur le port CENTRONICS
MC BUSY PRINTER	#BD2B	Teste si l'imprimante parallèle est occupée
MC SEND PRINTER	#BD2E	Envoie un caractère vers l'imprimante CENTRONICS
MC SOUND REGISTER	#BD31	Envoie une donnée au chip sonore

Accès au bloc de sauts

JMP RESTORE	#BD34	Initialisation du bloc de sauts
-------------	-------	---------------------------------