

## B-ASIC pour CPC+

### Logon System – Longshot

Vers.	Date	Evolutions
V1.1	??/04/1991	Creation du B-Asic pour A100% N°38 (mai/juin 1991) Instructions INKF, INKS, BORDERP, INKFRVB, INKSRVB, BORDERRVB
V1.2	??/05/1991	Version parue dans A100% N°39 (juin/août 1991) Instructions SPRON,SPROFF,SPRZOOM,SPRXY,SPRTDEF,SPRTON,SPRTOFF, SPRSWAP,SPRPLOT,SPRCOPY,SPRTURNX,SPRTURNY,SPRFILL
V1.3	??/08/1991	Version parue dans A100% N°40 (sept/oct 1991) - Correction bug avec MODE - Correction bug sur prg + de 16k - Modif instructions SPRTDEF, SPRTON, SPRTOFF - Instructions SPRTORAM, RAMTOSPR, SPRSAVE, SPRLOAD - Instructions RETARDX, RETARDY, MASQUE, SPLIT
V1.4	??/ ??/1992	Version parue dans A100% N°42 () Passage du B-Asic en banque (sans refonte du système RSX).
V2.1	??/ ??/1992	Version transitoire avec refonte du système d'interprétation des RSX
V3.1	??/??/1992	- 2eme page vidéo. Correction bugs mineurs. - Instructions POKEASIC, POKEVR, DEFORGXY, DEFORGSPR - Instructions SPRCATCH, INKCOPY, SPRLINK, SPRUNLINK, ERALINK
V3.2	25/02/2006	Correction bug calcul table dans interruption Correction bug firmware (ei avant fin interrupt)
V3.3	27/02/2006	02/2006 - Code du programme basic banque 2 transferé en banque 5. Pouvait poser problème sur des prog > 16 kb depuis la 3.1 - Instruction SPRWHERE
V3.4	24/05/2007	- Correction bug sur SPRWHERE (affectation variable entre 4000 et 7FFF) - Modification instruction SPRCOPY (qui accepte désormais des numéros de sprites entre 0 et 143). Les sprites 0 à 15 sont les sprites réels. Il y a désormais 128 sprites au lieu de 64 en plus. 64 sprites FAST et 64 sprites SLOW (les acces aux sprites 80 a 143 sont légèrement plus lents). Ces sprites additionnels n'occupent pas plus d'espace qu'avec les 64. - Instruction DOKE,adresse, valeur 16 bits Suppression des instructions RAMTOSPR et SPRTORAM, qui sont désormais inutiles (remplacées par SPRCOPY) - Instruction SPRTSTEP,n1,...,nx qui permet de jouer le chemin des sprites définis étape par étape. - Instruction DEFKBDSPR qui permet d'associer un ou deux sprites a des touches, avec gestion automatique ou manuelle. - Instruction CATCH devient SPRCATCH - Instruction SPRCOLL qui permet de gérer les détections de collisions entre 8 sprites. - Optimisations diverses

<b>Instruction</b>	<b>Description</b>
INKF,Encre, Couleur	Modification couleur d'une encre écran Encre=[0..15] Couleur=[0..4095]
INKFRVB,Encre, R, V, B	Modification couleur d'une encre écran définie par ses composantes rouge,vert,bleu Encre=[0..15] R/V/B = [0..15]
BORDERP,Couleur	Modification couleur du bord Couleur=[0..4095]
BORDERRVB,R,V,B	Modification couleur du bord définie par ses composantes rouge, vert, bleu R/V/B = [0..15]
INKS,Encre, Couleur	Modification couleur d'une encre sprite Encre=[1..15] Couleur=[0..4095]
INKSRVB,Encre, R,V, B	Modification couleur d'une encre sprite définie par ses composantes rouge, vert, bleu Encre=[1..15] R/V/B=[0..15]
SPROFF,NumSprite	Désactiver l'affichage d'un sprite NumSprite=[0..15]
SPRON,NumSprite	Activer l'affichage d'un sprite NumSprite=[0..15]
SPRZOOM,NumSprite, ZoomX, ZoomY	Zoomer un sprite Le sprite doit être activé pour être affiché NumSprite=[0..15] ZoomX/ZoomY=[0..3]
SPRXY,NumSprite, PosX, PosY	Positionner un sprite Le sprite doit être activé pour être affiché NumSprite=[0..15] PosX=[-256 ..767] PosY=[-256..255]
SPRTDEF,NumSprite, Adresse, Duree, DelaiDepart	Permet de définir un chemin pour un sprite Le chemin est défini par des coordonnées x, y définies en ram. Chaque coordonnée est définie sur 2 octets [donc 4 octets par position], avec l'octet de poids faible en 1 <sup>er</sup> et l'octet de poids fort en 2 <sup>nd</sup> . La dernière coordonnée x doit être FOF0 pour indiquer la fin du chemin avant bouclage. La durée indique le temps pendant lequel le sprite doit parcourir le chemin (0 pour toujours), et le délai de départ indique le temps à partir duquel le sprite commencera son chemin lorsque le chemin sera activé (voir instruction SPRTON) NumSprite=[0..15] Adresse=[0..65535] Duree=[0..65535], DelaiDepart=[0..65535] Les temps sont définis par pas de 1/50 <sup>ème</sup> de sec.
SPRTON,N1,N2, ....., Nx	Démarrage de 1 ou plusieurs chemins définis par SPRTDEF (simultanément). Le sprite doit être activé pour être affiché (SPRON) Nx=[0..15]
SPRTOFF,N1,N2,.....,Nx	Arrêt de 1 ou plusieurs chemins définis par SPRTDEF (simultanément) Nx=[0..15]
SPRTSTEP,N1,N2,.....,Nx	Jouer une étape d'un chemin défini par SPRTDEF. L'utilisateur reste maître de l'animation. Nx=[0..15]
SPRSWAP,NumSpr1, NumSpr2	Echange le contenu de 2 sprites NumSprX=[0..15]
SPRPLOT,NumSprite, PosX, PosY, Encre	Positionne un pixel dans un sprite NumSprite=[0..15] PosX=[0..15] PosY=[0..15] Encre=[0..15] (0 encre transparente)

SPRCOPY,NumSprDest, NumSprSource	Copie du contenu du sprite source dans le sprite dest. Il existe 144 sprites dans le B-Asic. Sprites 0 à 15 : Sprites réels Sprites 16 à 79 : Sprites Ram Fast Sprites 80 à 143 : Sprites Ram Slow NumSprX=[0..143]
SPRTURNX,NumSprite	Retourne un sprite horizontalement NumSprite=[0..15]
SPRTURNY,NumSprite	Retourne un sprite verticalement NumSprite=[0..15]
SPRFILL,NumSprite, Encre	Remplir un sprite avec une encre NumSprite=[0..15] Valeur=[0..15]
SPLIT,Ligne, Offset	Rupture Ligne =[0..255] , Offset=[0..65535] Cette instruction doit évoluer. Ne plus utiliser
RETARDX,Decalage	Gestion de l'avance retard horizontale Décalage=[0..15] Cette instruction doit évoluer. Ne plus utiliser
RETARDY,Decalage	Gestion de l'avance retard verticale Decalage=[0..7] Cette instruction doit évoluer. Ne plus utiliser
MASQUE,Type	Activation du masquage d'avance retard horiz. Type=0 ou 1 (inactif, actif) Cette instruction doit évoluer. Ne plus utiliser
SPRSAVE,[NoSprite],NoBlock,Filename	Sauvegarde d'un sprite ou d'un bloc de sprites Le B-Asic dispose d'une banque de 128 sprites en plus des 16 déjà en ram asic. Ils sont gérés comme 4 blocs de 32 sprites supplémentaires. Si le Numéro de sprite est précisé, alors 2 sprites sont sauvegardés (sprite x et sprite x+64), sinon le bloc de 32 sprites est sauvegardé. Le bloc 0 représente les sprites réels. <u>Remarque</u> : chaque bloc contient 16 sprites Fast et 16 sprites slow NoSprite=[0..15] champ facultatif NoBlock=[0..4], Filename=Nom du fichier sauvé
SPRLOAD,[NoSprite],NoBlock, Filename	Lecture d'un sprite ou d'un bloc de sprites Le B-Asic dispose d'une banque de 128 sprites en plus des 16 déjà en ram asic. Ils sont gérés comme 4 blocs de 32 sprites supplémentaires. Si le Numéro de sprite est précisé, alors 2 sprites sont chargés (sprite x et sprite x+64), sinon le bloc de 32 sprites est chargé. Le bloc 0 représente les sprites réels. <u>Remarque</u> : chaque bloc contient 16 sprites Fast et 16 sprites slow NoSprite=[0..15] champ facultatif NoBlock=[0..4], Filename=Nom du fichier à lire
DOKE, Adresse, Valeur 16 bits	Double Poke en mémoire pour stocker des valeurs 16 bits. Adresse=[&0000..&FFFF] Valeur16bits=[&0000..&FFFF]
SPRCATCH,NumSprite, PosX, PosY	Capture un sprite à partir d'un écran en mode 0 NumSprite=[0..15] PosX=[0..143] PosY=[0..184]
INKCOPY	Copie la palette de l'écran vers la palette sprites
SPRLINK,NumSpr1,NumSpr2, Xrel, Yrel	Attache un sprite à un autre sprite Le sprite 2 est attaché au sprite 1 avec des coordonnées relative. Si sprite 1 est en position X1,Y1, alors la position du sprite 2 est calculé comme suit X2=X1+Xrel et Y2=Y1+Yrel

SPRUNLINK,NumSpr1, NumSpr2	Détache un sprite d'un autre sprite.
ERALINK	Efface tous les attachements.
POKEASIC,Adresse, val1, ....., valn	Poke de 1 ou plusieurs valeurs dans les registres de l'asic. Adresse=[4000..7FFF] ValX=[0..FF] Réservé aux utilisateurs expérimentés
POKEVR,Adresse, val1, ..., valn	Poke de 1 ou plusieurs valeurs dans la Ram Video 2 Adresse=[4000..7FFF] ValX=[0..FF]
DEFORGXY, Sprite, X, Y	Définir la position absolue à partir de laquelle la position d'un sprite est définie. Si un sprite est positionné en 10, 10 et que l'instruction définit une position d'origine 30, 40, alors le sprite sera affiché réellement en 40, 50 Sprite = [0..15] X= [-256..767] Y=[-256..255]
DEFORGSPR, Sprite, SpriteAttache	Définir la position absolue d'un sprite par rapport à celle d'un autre sprite. Si un sprite est positionné en 10,10 et que le sprite d'attache est en 30, 40, alors le sprite attaché sera en 40, 50 Sprite et SpriteAttache = [0..15]
SPRWHERE,Sprite,@varx%,@vary% Attention à bien respecter le mode déclaratif des variables sous peine de résultats imprévisibles.  DEFKBDSPR, NoGestion, NoSprite, ToucheHaut, StepHaut, ToucheBas, StepBas, ToucheGauche, StepGauche, ToucheDroite, StepDroite DEFKBDSPR,NoGestion,255 DEFKBDSPR	Récupération des coordonnées d'un sprite. Sprite = [0..15] <b>IMPORTANT</b> : varx et vary doivent être préalablement définis comme entier (déclaration par initialisation avec %), puis ils doivent être passés par adresse à la fonction (@ devant le nom). Exemple : varx%=0 : vary%=0 lsprwhere,1,@varx%,@vary% Gestion d'un sprite par rapport au clavier/joystick NoGestion =[0..1] Il est possible de gérer 2 sprites NoSprite=[0..15] Numéro du sprite dirigé Touchexxxx=[Numéro de touche] Voir annexe NB : Une touche 0 équivaut à ne pas tester la touche Stepxxxx=Pas en pixel mode 2 de déplacement dans la direction correspondante Lorsque DEFKBDSPR est défini avec tous les paramètres, la gestion du sprite est automatique, et donc indépendante du programme basic. Il est possible de contrôler manuellement le déplacement en utilisant l'instruction sans paramètre (dès lors, le mode automatique est désactivé). Si on souhaite annuler une gestion (plutôt que de placer toutes les touches à 0), il est préférable de désactiver le sprite via DEFKBDSPR,NoGestion,255, ce qui décharge le B-Asic de faire des tests inutiles. Copie d'écran de la videoram 1 vers 2 (ou inversement) Sens : 0 pour une copie de l'écran principal vers le second écran (1 pour l'inverse)
SCREENCOPY, sens	

SPRCOLL,NoGestion,NoSprite1,DeltaX1,DeltaY1,  
NoSprite2,DeltaX2,DeltaY2,@collision%,PixelPerf

Gestion de la collision entre 2 sprites.  
NoGestion = [0..7] 8 collisions sont gérées  
actuellement  
NoSpritex=[0..15]  
DeltaXn, DeltaYn : Nombre de pixels mode 2 à  
partir de x,y du sprite pour lequel on doit gérer la  
collision. Cela permet d'étendre la collision à un  
super sprite ou à restreindre la collision à seulement  
une partie du sprite.  
PixelPerf : Non utilisé actuellement  
@collision% : variable qui prend la valeur 1  
lorsqu'une collision survient entre les 2 sprites.  
**IMPORTANT** : collision doit être préalablement  
défini comme entier (déclaration par initialisation  
avec %), puis ils doit être passé par adresse à la  
fonction (@ ou à (clavier français) devant le nom).  
Exemple : collision%=0  
Lorsque SPRCOLL est défini avec tous les  
paramètres, la gestion des collisions est  
automatique, et donc indépendante du programme  
basic. Il est possible de contrôler manuellement  
chaque test de collision en utilisant l'instruction  
sans paramètre (dès lors, le mode automatique est  
désactivé). Si on souhaite annuler une gestion de  
collision, il faut utiliser l'instruction avec 2  
paramètres : SPRCOLL,NoGestion,255 (cela  
décharge le B-Asic). **Voir note**

Ligne à ajouter dans le programme pour charger et initialiser le B-Asic

**10 IF PEEK(&BC0E)<>&CD THEN OUT &7F00,&C1 :LOAD «BASICxx.BIN »,&C000 :CALL &C000**

#### **NOTE TRES IMPORTANTE RELATIVE A L'INSTRUCTION SPRCOLL**

Cette instruction utilise une variable basic passée par adresse qui peut être affectées après coup par le B-Asic alors que la variable a changé de place. Il faut donc être vigilant dans les deux situations suivantes pour éviter une destruction de données de votre programme basic:

- Lorsque l'instruction SPRCOLL est en mode automatique, pensez à la replacer en mode manuel avant de modifier votre programme basic (ùSPRCOLL sans paramètre)
- Lorsque vous venez d'arrêter votre programme. Si vous le modifiez, ne faites jamais un GOTO vers une section de votre programme avec ùSPRCOLL seul sans avoir préalablement rappelé ùSPRCOLL avec tous les paramètres d'initialisation.

**ANNEXE 1 – Définition des touches pour DEFKBDSPR (touche en décimal)**

Numéro	Touche	Numéro	Touche	Numéro	Touche	Numéro	Touche
01	↑	02	→	03	↓	04	F9
05	F6	06	F3	07	ENTER	08	. (pavé n.)
17	←	18	COPY	19	F7	20	F8
21	F5	22	F1	23	F2	24	F0
33	CLR	34	[{	35	RETURN	36	]}
37	F4	38	SHIFT	39	\	40	CTRL
49	£^	50	-=	51	@	52	P
53	+ ;	54	* :	55	?/	56	>.
65	_ 0	66	) 9	67	O	68	I
69	L	70	K	71	M	72	< ,
81	( 8	82	' 7	83	U	84	Y
85	H	86	J	87	N	88	SPACE
97	&6 J2↑	98	%5 J2↓	99	R J2←	100	T J2→
101	G J2⊕	102	F	103	B	104	V
113	\$4	114	#3	115	E	116	W
117	S	118	D	119	C	120	X
129	!1	130	"2	131	ESC	132	Q
133	TAB	134	A	135	CAPS	136	Z
145	J1↑	146	J1↓	147	J1←	148	J1→
149	J1⊕	152	DEL				