

## DESSINEZ EN 4096 COULEURS

**C'est une exclusivité Amstrad Cent Pour Cent !**  
**Votre magazine préféré vous en donne toujours plus (c'est le cas de le dire) en créant pour vous cette nouvelle rubrique ou comment exploiter toutes les caractéristiques techniques de votre CPC Plus à l'intérieur de vos programmes Basic ou Assembleur.**

Vous êtes l'heureux possesseur d'un CPC Plus ? Bravo ! Vous étiez très content de votre CPC (tout court) et avez décidé de vous offrir un CPC Plus car celui-ci, outre le fait d'être compatible avec la vaste logithèque que vous possédez déjà, offre de nouvelles caractéristiques techniques : absolument alléchantes (sprites, palette des couleurs étendue, scrolling hard, etc). Malheureusement, lorsque vous allumez votre ordinateur, force vous est de constater qu'il ne s'agit ni plus ni moins, d'un simple CPC.

### MAIS OU SONT MES PLUS ?

Ça c'est vu du côté de l'utilisateur. Effectivement, il faut savoir que le CPC Plus est bel et bien doté de nouvelles fonctionnalités mais celles-ci sont encodées quelque part dans la mémoire de la machine et sont inaccessibles au commun des utilisateurs lambda. Le guide de l'utilisateur ne fait nullement référence à ces possibilités cachées... Celles-ci ne peuvent être exploitées que dans le cadre de jeux ayant pour support les fameuses cartouches qui viennent s'enficher dans le port prévu à cet effet sur la machine. Seuls les éditeurs ayant passé un accord de développement avec Amstrad sont autorisés à produire des jeux sur ce format et exploiter ces capacités supplémentaires dissimulées au fin fond de la bécanne.

Amstrad ayant bien fait les choses, il faut disposer d'une séquence de codes (protégée par copyright) pour faire comprendre à la machine que nous désirons utiliser ses pleines capacités. Cette séquence n'est bien sûr connue que des seuls éditeurs ayant le droit de développer des jeux en cartouche.

### LE POURQUOI DU COMMENT

En tant que simple utilisateur, vous avez parfaitement le droit de chercher (et peut-être trouver) comment exploiter le plein potentiel de votre machine puisque vous vous en êtes porté acquéreur en la payant. C'est là que se trouve le petit oubli d'Amstrad. La société anglaise, désireuse d'offrir des logiciels de meilleure qualité aux « CPCistes », a permis aux éditeurs de créer de très beaux jeux mais n'a pas permis du même coup à l'utilisateur final d'exploiter à son compte les Plus de la machine. C'est le pourquoi de cette série d'articles qui n'a d'autres prétentions que de vous permettre enfin des réalisations aussi belles que Burnin' Rubber...

Même pour un programmeur averti, un rapide tour de la Rom Basic (qui se trouve sur la cartouche livrée avec la machine) ne lui permet pas de déceler la moindre trace d'une gestion de 4096 couleurs.

Et pourtant, sur cette même cartouche se trouve justement Burnin' Rubber qui, lui, utilise tous les Plus de la machine. C'est là qu'il faut chercher... Mais l'absence de toute documentation vous fera défaut pour savoir où et comment utiliser toutes ces nouvelles caractéristiques.

Nous vous proposons donc de faire le tour des nouveautés du Plus en commençant par sa large palette des couleurs.

### LA FAMEUSE SEQUENCE

Il s'agit de 17 octets envoyés sur le port &BC00 (habituellement le registre de

sélection du CRTIC) qui provoquent l'activation de la nouvelle fonction du Gate Array : la fonction RMR2.

Voici cette séquence de code :  
 255,0,255,119,179,81,168,212,98,  
 57,156,70,43,21,138,205,238

La nouvelle fonction du Gate Array, RMR2, sera décrite dans un prochain article, mais sachez qu'elle permet, entre autres, de déverrouiller une page de registre I/O située entre &4000 et &7FFF. Contrairement à ce que l'on a l'habitude de voir sur un Z80A au niveau des entrées/sorties, l'accès aux périphériques (ici l'Asic) est fait grâce à un procédé de Mapping Ram I/O. Celui-ci évite l'emploi des instructions d'I/O conventionnelles du Z80A. Il suffit, pour adresser un périphérique, de placer un octet en mémoire I/O.

### LE LOGONBASIC

Pour étendre vos possibilités de programmation avec de nouvelles instructions, complémentaires à celles du Basic que vous connaissez déjà, nous avons demandé aux Logon System de créer des commandes RSX. Vous voici donc avec un Basic enrichi que nous appellerons le LogonBasic (Ou B-Asic), grâce auquel vous pourrez désormais exploiter pleinement les capacités de votre Plus...

Un chapitre technique sera réservé à la fin de chaque article pour ceux qui veulent aller plus loin et programmer leur Plus en Assembleur !

Mais tout d'abord, qu'est-ce que c'est-ye que le logon il a préparé ce mois-ci ? Hein ?

Eh bien pour commencer, nous allons nous attaquer aux couleurs...

Tapez le listing ci-contre, sauvez-le sous le nom de logonbas (Save "Logonbas"), puis tapez Run (c'est pas difficile jus-

```

10 REM LOGON SYSTEM / AMSTRAD 100 %
20 REM
30 REM PROGRAMME GENERANT LOGONBAS.COD
40 REM
50 REM A EXECUTER UNE SEULE FOIS !!
60 REM
70 REM PLACEZ VOUS EN MODE 2 CA IRA MIEUX....
80 REM
100 MODE 2:BORDER 0:INK 0,0:INK 1,26
110 PRINT "Generation LOGONBAS.COD"
120 RESTORE:ADR=41808
130 FOR I=0 TO 28
135 S=0
140 FOR J=0 TO 15
150 READ A$:A=VAL("&"+A$):S=S+A
160 POKE ADR,A:ADR=ADR+1
170 NEXT J
180 READ CHK
190 IF CHK<>S THEN PRINT:PRINT "Erreur Ligne ";2000+(I*10):END
200 NEXT I
210 SAVE "LOGONBAS.COD",B,&A350,&1C4,&1234
220 PRINT:PRINT "LOGONBAS.COD Sauve !"
230 END
1997 REM
1998 REM OULA LA LA ! TOUT CA A TAPER ?
1999 REM
2000 DATA 3A,F9,A3,B7,C0,01,FA,A3,21,53,A4,CD,D1,BC,01,09, 2151
2010 DATA A4,21,57,A4,CD,D1,BC,01,13,A4,21,5B,A4,CD,D1,BC, 2124
2020 DATA 01,20,A4,21,5F,A4,CD,D1,BC,01,2D,A4,21,63,A4,CD, 1802
2030 DATA D1,BC,01,3C,A4,21,67,A4,CD,D1,BC,01,46,A4,21,6B, 1899
2040 DATA A4,CD,D1,BC,F3,1E,11,21,D1,A3,01,00,BC,7E,ED,79, 2134
2050 DATA 23,1D,20,F9,2A,39,00,11,CE,A3,E5,ED,A0,ED,A0,ED, 2090
2060 DATA A0,4E,06,00,23,22,F8,A3,09,22,F5,A3,D1,21,CB,A3, 1783
2070 DATA ED,A0,ED,A0,ED,A0,FB,C3,CA,A3,C9,C3,E2,A3,00,00, 2787
2080 DATA 00,FF,00,FF,77,B3,51,A8,D4,62,39,9C,46,2B,15,8A, 1852
2090 DATA CD,EE,F5,C5,06,F5,ED,78,1F,D2,F1,A3,3E,FF,32,F8, 2753
2100 DATA B7,C1,F1,08,DA,00,00,C3,00,00,FF,A3,C3,6F,A4,43, 1993
2110 DATA 4F,50,59,52,49,47,48,D4,00,0E,A4,C3,9A,A4,49,4E, 1600
2120 DATA 4B,C6,00,18,A4,C3,BE,A4,49,4E,4B,46,52,56,C2,00, 1668
2130 DATA 25,A4,C3,E3,A4,42,4F,52,44,45,52,D0,00,32,A4,C3, 1850
2140 DATA EF,A4,42,4F,52,44,45,52,52,56,C2,00,41,A4,C3,FB, 1886
2150 DATA A4,49,4E,4B,D3,00,4B,A4,C3,07,A5,49,4E,4B,53,52, 1598
2160 DATA 56,C2,00,00,00,00,00,00,00,00,00,00,00,00,00, 280
2170 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,21, 33
2180 DATA 7B,A4,7E,B7,C8,CD,5A,BB,23,18,F7,0D,0A,42,2D,41, 1783
2190 DATA 53,49,43,20,42,79,20,4C,6F,67,6F,6E,20,53,79,73, 1336
2200 DATA 74,65,6D,20,31,39,39,31,2E,00,DD,7E,02,E6,0F,21, 1243
2210 DATA 00,64,17,01,B8,7F,ED,49,4F,DD,7E,01,E6,0F,57,DD, 1725
2220 DATA 5E,00,06,00,00,09,73,2C,72,01,A0,7F,ED,49,C9,DD,7E, 1528
2230 DATA 06,E6,0F,21,00,64,17,01,B8,7F,ED,49,4F,06,0F,DD, 1350
2240 DATA 7E,00,A0,5F,DD,7E,04,A0,87,87,87,87,B3,5F,DD,56, 2013
2250 DATA 02,18,CF,01,B8,7F,ED,49,0E,20,21,00,64,18,BA,01, 1245
2260 DATA B8,7F,ED,49,0E,20,21,00,64,18,D2,DD,7E,02,E6,0F, 1628
2270 DATA 21,20,64,B7,20,9C,C9,DD,7E,06,E6,0F,B7,21,20,64, 1683
2280 DATA 20,B4,C9,00,00,00,00,00,00,00,00,00,00,00,00, 413
    
```

que-là !)... Si tout se passe bien, un fichier Logonbas.cod est créé sur la disquette...

Vous pourrez désormais utiliser le logonbasic en incluant en première ligne de vos programmes :

```

10 IF PEEK(&A350)<>&3A THEN
MEMORY &A34F:LOAD "LOGON-
BAS.COD":CALL &A350
    
```

Vous disposez alors de 6 fonctions supplémentaires. En voici la description :  
 INKF,numéroencres,couleur  
 Permet de fixer une des 16 encres de fond avec une des 4096 couleurs.

"numéroencres" est compris entre 0 et 15. "couleur" est compris entre 0 et 4095

INKS,numéroencres,couleur  
 Fixer une des 15 encres des Sprites avec une des 4096 couleurs.

"numéroencres" est compris entre 1 et 15  
 "couleur" est compris entre 0 et 4095

N.B. L'encore 0 des sprites étant l'encore transparente.  
 BORDERP,couleur  
 Fixer la Couleur du Bord avec une des 4096 couleurs.

"couleur" est compris entre 0 et 4095  
 INKFRVB,numéroencres, Rouge, Vert, Bleu

Fixer une des 16 encres de fond avec une des 4096 couleurs choisie en fonction des composantes Rouge,Vert et Bleu.

"numéroencres" est compris entre 0 et 15

"Rouge" est compris entre 0 et 15  
 "Vert" est compris entre 0 et 15  
 "Bleu" est compris entre 0 et 15  
 INKSRVB,numéroencres, Rouge, Vert, Bleu

Fixer une des 15 encres des sprites avec une des 4096 couleurs choisie en fonction des composantes rouge,vert,bleu.

"numéroencres" est compris entre 1 et 15  
 "Rouge" est compris entre 0 et 15  
 "Vert" est compris entre 0 et 15  
 "Bleu" est compris entre 0 et 15  
 BORDERRVB,Rouge,Vert,Bleu  
 Fixer la couleur du bord avec une des 4096 couleurs choisie en fonction des composantes rouge,vert et bleu.  
 "Rouge" est compris entre 0 et 15  
 "Vert" est compris entre 0 et 15  
 "Bleu" est compris entre 0 et 15

Voilà pour le descriptif des 6 nouvelles instructions du B-Asic.

Je rappelle que vous devez taper le caractère | (Sur le clavier, celui-ci apparaît comme un ù à l'écran...) devant les nouvelles instructions, par exemple : ùBORDERP,567

## UN EXEMPLE COLORE

Pour mettre tout de suite en application l'une de vos nouvelles commandes, vous pouvez taper le programme de démo qui se trouve dans le coin. Il vous permet d'attribuer des couleurs différentes aux 16 encres disponibles sous le mode 0 (basse résolution) du CPC.

Déplacez simplement les « flèches-curseur » avec les touches flèches de votre pavé numérique puis incrémentez ou décrémente les valeurs de rouge, vert et bleu avec les touches - et + de votre clavier.

Libre à vous de vous amuser comme vous le souhaitez avec ces nouvelles commandes dans vos propres créations. Attention toutefois, celles-ci devront rester du domaine public puisque la séquence de codes utilisée pour exploiter les Plus de votre machine est sous copyright Amstrad.

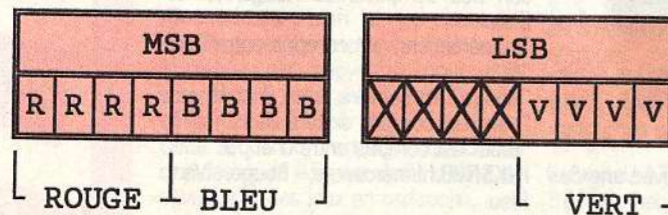
Et bien sûr, nous serons ravis de recevoir des programmes ou des pages graphiques exploitant ces RSX. Les réalisations originales seront publiées.

Une remarque encore... Lorsque vous utilisez les nouvelles instructions relatives aux couleurs, il vous est possible de vous servir des anciennes commandes du Basic Amstrad (INK et BORDER).

Cependant, leurs utilisations auront pour effet immédiat d'annuler complètement la palette définie avec LogonBasic et vous vous retrouverez instantanément avec l'ancienne palette. Mon conseil est donc : laissez tomber les vieilles instructions...

Voilà, répétons encore que vous pouvez désormais agrémenter vos jeux, programmes ou dessins de couleurs plus adaptées puisque le choix vous en est maintenant offert. Faites de beaux rêves... en 4096 couleurs !!!

FOND	SPRITES	NUMERO ENCRE	I/O
6400-6401		0 TRANSPARENT	R/W
6402-6403	6422-6423	1	R/W
6404-6405	6424-6425	2	R/W
6406-6407	6426-6427	3	R/W
6408-6409	6428-6429	4	R/W
640A-640B	642A-642B	5	R/W
640C-640D	642C-642D	6	R/W
640E-640F	642E-642F	7	R/W
6410-6411	6430-6431	8	R/W
6412-6413	6432-6433	9	R/W
6414-6415	6434-6435	A (10)	R/W
6416-6417	6436-6437	B (11)	R/W
6418-6419	6438-6439	C (12)	R/W
641A-641B	643A-643B	D (13)	R/W
641C-641D	643C-643D	E (14)	R/W
641E-641F	643E-643F	F (15)	R/W
6420-6421		BORDER	R/W



## LA REVUE TECHNIQUE

Je rappelle que cette partie est réservée à ceux qui veulent aller plus loin dans la programmation du Plus... Tout d'abord, abordons le principal problème rencontré par les nouvelles instructions couleurs : les perturbations du système avec les couleurs. En effet, le système gère habituellement le rafraîchissement des couleurs sous interruptions, ce qui est fâcheux lorsqu'on essaie une instruction d'entrée/sortie directe sur le Gate Array (via le OUT du Basic). Le problème était donc le même avec l'Asic. Il s'agissait donc d'empêcher le système de remettre à jour le Gate Array avec ses tables de couleurs.

La solution était simple : déviation du système d'interruption existant pour permettre l'antidécimation du compteur système de flash des encres. Ainsi, à chaque VBL, le compteur est-il forcé à sa valeur maximale et ne peut donc, par conséquent, jamais atteindre 0... Il suffisait d'y penser !  
 Abordons maintenant les vecteurs I/O de l'Asic concernant les couleurs...

Il vous suffit juste pour affecter une encres (ou un registre de l'Asic), de connecter la page I/O (OUT &7F00,&B8), d'affecter les registres à modifier en pokant les valeurs aux adresses voulues, puis de déconnecter la page I/O (OUT &7F00, &A0). Je reviendrai sur la façon d'utiliser le registre RMR2 du Gate Array/Asic prochainement !  
 Voilà, il ne nous reste qu'à vous souhaiter tout plein de rasters...colorés !

### NEXT TIME

Le mois prochain, nous continuerons bien sûr sur notre lancée avec de nouvelles instructions du LogonBasic. Nous verrons notamment comment exploiter les sprites hard de la machine. Avec la possibilité de déplacer 16 lutins sur votre écran, vous réaliserez des programmes intéressants sans aborder le problème complexe de la gestion de ces sprites.

Logon System, Poum et Robby

```

10 '
20 '*****
30 '* *
40 '* Demo pour les "plus" *
50 '* *
60 '* Par Poum & Logon *
70 '* *
80 '*****
90 '
100 IF PEEK(&A350)=0 THEN MEMORY &A34F:
LOAD "LOGONBAS.COD":CALL &A350
110 '
120 DIM rvb(3,15)
130 FOR i=1 TO 15:FOR j=1 TO 3
140 rvb(j,i)=15
150 NEXT j,i
160 PEN 1:MODE 0:LOCATE 10,1:PRINT "R V
B"
170 PRINT:FOR i=0 TO 15:PEN i
180 PRINT STRING$(4,143);" ";
190 PEN 1:PRINT HEX$(i,1):NEXT
200 FOR j=1 TO 3:FOR i=0 TO 15
210 LOCATE 8+j*2,3+i:PRINT HEX$(rvb(j,i
),1)
220 NEXT i,j
230 BORDER 0:FOR y=0 TO 15
240 ùINKFRVB,y,rvb(1,y),rvb(2,y),rvb(3,
y)
250 NEXT
260 x=1:y=1
270 '
280 LOCATE 8,y+3:PRINT CHR$(243);
290 LOCATE 8+x*2,20:PRINT CHR$(240);
300 '
310 cla$=""
320 WHILE cla$="":cla$=INKEY$:WEND
330 IF cla$=CHR$(240) AND y<>0 THEN 410
340 IF cla$=CHR$(241) AND y<>15 THEN 430
350 IF cla$=CHR$(242) AND x<>1 THEN 450
360 IF cla$=CHR$(243) AND x<>3 THEN 470
370 IF cla$="+" THEN 490
380 IF cla$="-" THEN 530
390 GOTO 310
400 '
410 LOCATE 8,y+3:PRINT CHR$(32)
420 y=y-1:GOTO 280
430 LOCATE 8,y+3:PRINT CHR$(32)
440 y=y+1:GOTO 280
450 LOCATE 8+x*2,20:PRINT CHR$(32)
460 x=x-1:GOTO 280
470 LOCATE 8+x*2,20:PRINT CHR$(32)
480 x=x+1:GOTO 280
490 IF rvb(x,y)=15 THEN 310
500 rvb(x,y)=rvb(x,y)+1
510 LOCATE 8+x*2,3+y:PRINT HEX$(rvb(x,y
),1);
520 ùINKFRVB,y,rvb(1,y),rvb(2,y),rvb(3,
y):GOTO 310
530 IF rvb(x,y)=0 THEN 310
540 rvb(x,y)=rvb(x,y)-1
550 LOCATE 8+x*2,3+y:PRINT HEX$(rvb(x,y
),1);
560 ùINKFRVB,y,rvb(1,y),rvb(2,y),rvb(3,
y):GOTO 310

```