

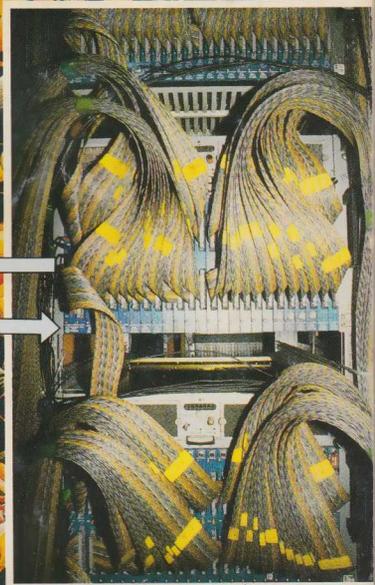
SCIENCE & VIE

MENSUEL

N° 823 AVRIL 1986

ISSN 0036 8369

**INTELLIGENCE
ARTIFICIELLE
CONTRE
INTELLIGENCE
NATURELLE**



**323 USINES DANGEREUSES
RECENSÉES EN FRANCE**
VOIR NOS CARTES PAGES 72, 77, 79, 81, 83

16 F

N° 823

SUISSE 5 FS
CANADA \$ 2,95
BELGIQUE 110 FB
ESPAGNE 450 Ptas
MAROC 17 Dh
TUNISIE 1,6 DT

Illustrations sonores pour Amstrad

INFORMATIQUE AMUSANTE

Sans illustration sonore un jeu reste bien terne. Afin donc, de donner un peu de vie à vos programmes, voici comment réaliser sur Amstrad quelques bruitages sélectionnés parmi les plus utilisés.

N'oublions pas que sur ce micro-ordinateur il existe aucun son programmé d'origine. Cependant grâce aux instructions ENV et SOUND, il nous sera possible de mettre au point de très bons bruitages. Rappelons que, pour paraître réel, un son devra comporter une ou plusieurs fréquences de base, plus, éventuellement, un bruit et son évolution contrôlée dans le temps. Les notes de base seront générées à l'aide de SOUND et nous déterminerons un format d'enveloppe (évolution dans le temps) en utilisant ENV.

Pour que ces illustrations sonores soient facilement accessibles — du moins pour vous les présenter — nous les avons regroupées dans diverses sous-routines directement disponibles depuis une page de présentation. Chaque fois qu'un son sera demandé, l'ordinateur rappellera, au bas de l'écran, le type de son choisi ainsi que l'emplacement dans le programme de la sous-routine employée.

Enfin dans chaque sous-routine un emplacement est réservé pour une animation. Celle-ci, pour ne pas perturber la génération du son, devra rester aussi courte que possible (modification de la position de quelques caractères présents sur l'écran par exemple).

Comme il est facile de le constater nous placerons toujours les animations après une instruction SOUND. Ceci nous permettra d'utiliser le "buffer" du générateur de sons et donc de laisser à l'ordinateur le temps de faire autre chose pendant qu'une note est émise.

Ces précisions apportées, pas-

- Bonjour, -

voici la liste des illustrations sonores que je vous propose.

0 BOMBE

1 MITRAILLEUSE

2 D.C.A.

3 REVOLVER

4 LOCOMOTIVE

5 AVION

6 SYNTHETISEUR 1

7 ORGUE

8 SYNTHETISEUR 2

9 HARMONICA

Taper la touche correspondant au son désiré.

SYNTHETISEUR 1: sous-routine 2000.

sons au programme. Nous commencerons par passer la machine en mode 1 (mode de fonctionnement standard pour l'affichage) puis l'écran sera effacé (ligne 5). Ensuite nous imprimerons la page de présentation. Une succession d'instructions PRINT sera utilisée à cet effet des lignes 10 à 60. Rien donc de bien nouveau ici. Ensuite une boucle nous permettra de savoir si une touche du clavier a été frappée.

Le programme reviendra systématiquement sur cette boucle tant qu'aucun son ne sera demandé, en raison du test de la ligne 80. En cas contraire, la variable SUB prendra une valeur liée à la touche tapée et l'instruction ON sub GOSUB de la ligne 90 aiguillera le programme vers la sous-routine désirée. Notons qu'après l'exécution de chaque sous-routine l'état du générateur de sons sera contrôlé par le test de la ligne 91 afin d'éviter de lui envoyer de nouvelles données tant que le son précédent n'aura pas été totalement exécuté.

Ce contrôle est indispensable pour éviter une saturation et donc un certain cafouillage du générateur de sons. Enfin, avant les diverses sous-routines, nous trouverons une suite de données contenues dans une chaîne DATA (ligne 100). Nous utiliserons ces dernières pour créer la mélodie des instruments de musique synthétisés. Les données seront lues par groupes de deux. La première de chaque groupe commandera le canal sonore numéro 1, la seconde

commandera le numéro 2.

Passons donc maintenant à l'écriture de chaque sous-routine. Le nombre d'illustrations sonores proposées étant de 10, il n'est pas surprenant de trouver 10 sous-routines. Notons qu'une onzième nous permettra de simplifier celles concernant les instruments de musique.

La bombe

Nous trouverons sa sous-routine entre les lignes 1100 et 1135. En premier lieu nous déterminerons une enveloppe sonore correspondant à une explosion.

L'ordre ENV sera utilisé à cet effet à la ligne 1100. Si vous désirez utiliser directement ce bruitage dans un jeu, la ligne 1101 devra être supprimée.

En effet son seul but est d'afficher sur l'écran le type et le numéro de la sous-routine employée. Les instructions SOUND des lignes 1110 et 1115 généreront le sifflement de la bombe durant sa chute, une courte animation pourra être logée après elles, puis les lignes 1125 et 1130 simuleront l'explosion.

Là encore, une animation pourra prendre place. Cette sous-routine se terminera par l'instruction RETURN de la ligne 1135.

La mitrailleuse

Ce bruitage utilise les lignes 1220 à 1230. Deux enveloppes seront créées: l'une sera utilisée pour le tir des balles, l'autre en fin de salve. Le nombre de projectiles tirés est fixé par la valeur de I à la ligne 1210-20 dans notre cas, mais

ce chiffre pourra aisément être modifié.

Vient ensuite la salve: ordre SOUND de la ligne 1215 avec utilisation de la première enveloppe, puis la fin de tir avec utilisation à la ligne 1225 de l'enveloppe numéro 2.

Cette sous-routine se terminera en ligne 1230 par une instruction RETURN. Comme précédemment, il sera possible de loger une courte animation après la ligne 1215.

La D.C.A.

La sous-routine des lignes 1300 à 1320 sera utilisée. Ici, contrairement aux autres cas nous ne définirons pas d'enveloppe. En effet, seule une modulation du générateur de bruits nous permettra de synthétiser cette arme. Là encore, le nombre de coups par salve sera fonction de la valeur donnée à I en ligne 1301.

F nous permettra de moduler le générateur de bruits en introduisant sa valeur à la fin de l'ordre SOUND de la ligne 1305. Si une animation est souhaitée elle devra être placée immédiatement à la suite de cette ligne. Cette sous-routine se terminera en 1320 par un ordre RETURN.

Le revolver

Nous simulerons ici le bruit du coup et le sifflement de la balle. Nous commencerons donc par créer une enveloppe correspondant au son produit par le coup (ligne 1400) puis celui-ci sera généré (ligne 1405). Comme pour les autres cas, une animation pourra prendre place entre les lignes 1405 et 1410.

Une légère pause sera introduite entre le coup et le sifflement de la balle à l'aide de la boucle de la ligne 1410. Une seconde boucle sera alors utilisée pour restituer le sifflement de la balle et contrôlera également la puissance du son émis. En effet I interviendra ici deux fois: à la fois pour la fréquence et pour le volume. Notons que, étant donné la vitesse d'exécution de cette boucle, il ne sera pas possible de lui insérer d'animation. Cette sous-routine se terminera en ligne 1430 par un ordre RETURN.

La locomotive

Cette illustration sonore utilisera la sous-routine comprise entre les lignes 1500 et 1545. Nous commencerons par définir une en-

```

5 MODE 1: CLEAR:CLS
10 PRINT " - Bonjour, - ":PRINT
15 PRINT "voici la liste des illustrations sonores"
20 PRINT "que je vous propose."
25 PRINT:PRINT
30 PRINT "0 BOMBE 5 AVION":PRINT
35 PRINT "1 MITRAILLEUSE 6 SYNTHETISEUR 1":PRINT
40 PRINT "2 D.C.A. 7 ORGUE":PRINT
45 PRINT "3 REVOLVER 8 SYNTHETISEUR 2":PRINT
50 PRINT "4 LOCOMOTIVE 9 HARMONICA":PRINT
55 PRINT:PRINT " Taper la touche correspondant":PRINT
60 PRINT "au son desire."
65 LET A$=INKEY$:LET sub =0
66 FOR K=48 TO 57
70 IF CHR$(K)=A$ THEN LET sub=K-47
75 NEXT K
80 IF sub =0 THEN GOTO 65
85 LOCATE 1,24
90 ON sub GOSUB 1100,1200,1300,1400,1500,1600,2000,2100,2200,2300
91 IF SQ(1)<>4 OR SQ(2)<>4 THEN GOTO 91
95 GOTO 65
100 DATA 239,190,478,379,239,190,358,268
1100 ENV 3,1,15,5,1,-2,10,13,-1,20
1101 PRINT "BOMBE: sous routine 1100."
1105 FOR I=40 TO 200
1110 SOUND 1,1,2,12,0,0,0
1115 SOUND 2,1+3,2,12,0,0,0
1116 REM *****
1117 REM - PLACER ICI VOTRE ANIMATION -
1118 REM *****
1120 NEXT I
1125 SOUND 1,2000,0,0,3,0,30
1130 SOUND 2,2050,0,0,3,0,25
1131 REM *****
1132 REM - PLACER ICI VOTRE ANIMATION -
1133 REM *****
1135 RETURN
1200 ENV 1,1,15,2,1,-5,3,2,-5,1
1205 ENV 2,1,15,2,1,-5,3,5,-2,10
1206 PRINT "MITRAILLEUSE: sous-routine 1200."
1210 FOR I=1 TO 20
1215 SOUND 1,0,0,0,1,0,(RND*4)+9
1216 REM *****
1217 REM - PLACER ICI VOTRE ANIMATION -
1218 REM *****
1220 NEXT I
1225 SOUND 1,0,0,0,2,0,9
1230 RETURN
1300 PRINT "D.C.A.: sous routine 1300."
1301 FOR I=1 TO 5
1302 FOR F=1 TO 20
1305 SOUND 1,0,1,7,0,0,F
1306 REM *****
1307 REM - PLACER ICI VOTRE ANIMATION -
1308 REM *****
1309 NEXT F
1310 NEXT I
1315 SOUND 1,0,0,0,2,0,20
1320 RETURN
1400 ENV 4,1,15,10,1,-3,3,6,-2,10
1401 PRINT "REVOLVER: sous-routine 1400."
1405 SOUND 1,0,0,0,4,0,12
1406 REM *****
1407 REM - PLACER ICI VOTRE ANIMATION -
1408 REM *****

```

veloppe (ligne 1500) puis deux boucles successives seront chargées de restituer l'effet de déplacement du train.

Elles commanderont avant tout le générateur de bruit du micro-ordinateur. Nous trouverons ces deux boucles respectivement entre les lignes 1505 et 1520 puis 1525 et 1540. Deux emplacements sont réservés pour y loger une animation. Cette sous-routine se terminera en ligne 1545 par RETURN.

L'avion

Ce bruitage correspond au bruit

que l'on peut entendre lorsque l'on se trouve dans un bimoteur à hélice. La sous-routine placée entre les lignes 1600 et 1620 sera utilisée. La durée de cette illustration sonore est directement liée à celle de I (ligne 1601); il sera donc aisé de la modifier.

Ici, nous n'utiliserons pas d'enveloppe mais deux fréquences proches sur les canaux 1 et 2 permettront, par effet de battement, de restituer le bruit d'un avion en vol. Un emplacement est prévu pour une animation en ligne 1612.

```

1410 FOR T=0 TO 200:NEXT T
1415 FOR I=40 TO 90
1420 SOUND 2,I,1,9-INT(I/10),0,0,0
1425 NEXT I
1430 RETURN
1500 ENV 5,10,1,1,1,1,1,2,-5,1
1501 PRINT "LOCOMOTIVE: sous-routine 1500.
1505 FOR I=1 TO 20
1510 SOUND 1,0,0,0,5,0,I+2
1515 SOUND 1,0,0,0,5,0,I
1516 REM *****
1517 REM - PLACER ICI VOTRE ANIMATION -
1518 REM *****
1520 NEXT I
1525 FOR I=20 TO 1 STEP -1
1530 SOUND 1,0,0,0,5,0,I+2
1535 SOUND 1,0,0,0,5,0,I
1536 REM *****
1537 REM - PLACER ICI VOTRE ANIMATION -
1538 REM *****
1540 NEXT I
1545 RETURN
1600 PRINT "AVION: sous routine 1600.
1601 FOR I=1 TO 100
1605 SOUND 1,2000,20,14,0,0,0
1610 SOUND 2,2021,20,14,0,0,0
1611 REM *****
1612 REM - PLACER ICI VOTRE ANIMATION -
1613 REM *****
1615 NEXT I
1620 RETURN
2000 ENV 6,1,15,2,1,-3,6,12,-1,1
2005 PRINT "SYNTHETISEUR 1: sous-routine 2000.
2010 LET P=5:LET W=1:GOSUB 3000
2015 RETURN
2100 ENV 6,5,3,1,1,0,10,3,-5,1
2105 PRINT "ORGUE: sous-routine 2100.
2110 LET P=0:LET W=1:GOSUB 3000
2115 RETURN
2200 ENV 6,1,15,7,10,-3,1,1,-15,1
2205 PRINT "SYNTHETISEUR 2 : sous-routine 2200.
2210 LET P=7:LET W=1:GOSUB 3000
2215 RETURN
2300 ENV 6,5,3,1,5,-1,2,2,-5,1
2305 PRINT "HARMONICA: sous-routine 2200.
2310 LET P=3:LET W=2:GOSUB 3000
2315 RETURN
3000 FOR I=1 TO 2
3005 FOR O=1 TO 2
3010 FOR L=1 TO 2
3015 RESTORE 100
3020 FOR M=0 TO 3
3025 READ N1:READ N2
3030 SOUND 3,12,2,P,0,0,1
3035 SOUND 1,N1/W,0,0,6,0,0
3040 SOUND 2,N2/(O*L*W),0,0,6,0,0
3045 REM *****
3050 REM - PLACER ICI VOTRE ANIMATION -
3055 REM *****
3060 NEXT M
3065 NEXT L
3070 NEXT O
3075 NEXT I
3080 RETURN

```

Cette sous-routine se terminera en 1620 par une instruction RETURN.

Les instruments de musique

Comme, dans tous les cas, la sous-routine d'exécution de la mélodie sera identique, nous l'avons logée dans une annexe située entre les lignes 3000 et 3080.

Etudions sa fonction avant de voir le cas de chaque instrument qui, somme toute, ne sera qu'une modification de l'enveloppe utilisée pour cette annexe. En premier lieu trois boucles permettront de jouer la mélodie avec diverses

transpositions (lignes 3000, 3005 et 3010 utilisant les variables I, O et L). Ensuite les données contenues dans la ligne 100 seront, comme nous l'avons indiqué plus haut, lues (ligne 3025) puis aiguillées vers les deux premiers canaux sonores (lignes 3035 et 3040).

La ligne 3040 prend en compte les variables O, L et W afin, à partir d'une mélodie on ne peut plus simple, d'agrémenter son orchestration. Pour modifier l'air joué il suffira de modifier les données contenues dans la ligne 100, mais

attention, les bornes de M devront être modifiées en conséquence (ligne 3020). Ces précisions apportées passons en revue l'ensemble des instruments que nous imiterons.

Le synthétiseur 1

Ici tout est permis. Nous utiliserons donc l'enveloppe déterminée en 2000 en lui ajoutant du bruit à l'aide de la variable P et en fixant la hauteur du son par W.

L'orgue

Ici seule l'enveloppe et le bruit seront modifiés (aucun bruit ne sera ajouté à la mélodie).

Le synthétiseur

L'enveloppe, ainsi que le niveau de bruits ajoutés seront modifiés de manière à obtenir un son plus syncopé.

L'harmonica

L'enveloppe sera de nouveau modifiée ainsi que la fréquence fixée par la lecture des données à l'aide de W. Cette sous-routine se terminera en 2315 par un ordre RETURN.

Ce programme, avant tout destiné à être utilisé sous-routine par sous-routine pour vos jeux, ne doit pas poser de problème de frappe particulier. Afin de faciliter son adaptation sur d'autres appareils nous n'avons pas utilisé, pour notre listing, le basic abrégé. Rappelons que pour l'Amstrad les instructions doivent être tapées caractère par caractère ou en abrégé (pour cela consultez votre fascicule d'emploi).

L'utilisation de ce programme reste également fort simple. Après avoir tapé "RUN" la page de présentation s'affichera et vous demandera le son désiré. Après sélection, la sous-routine concernée sera exécutée puis la page de présentation sera de nouveau affichée. Notons que, comme l'Amstrad possède un "buffer" de clavier capable de mémoriser la frappe d'une suite de touches, il sera possible de demander une série de sons, sans attendre leur exécution.

Il en sera de même quant à l'utilisation de ces illustrations sonores dans vos programmes; il sera inutile d'attendre la fin d'un son avant d'entamer l'opération suivante, sous réserve toutefois de ne pas saturer le générateur de sons.

Henri-Pierre Penel

Le "hard" n'est pas si dur qu'il en a l'air

INFORMATIQUE PRATIQUE

Voici la suite de cette rubrique mensuelle, commencée dans notre dernier numéro et dont le but est d'utiliser au maximum les ressources du micro-ordinateur dont vous pouvez être équipé à la maison.

Tout au long de ces lignes nous serons obligés d'utiliser des termes bien ésotériques, relatifs au fonctionnement d'un microprocesseur ; pour être plus précis, à celui qui équipe le "micro" que nous avons choisi pour la présente rubrique, le Z 80. Pour la clarté des exposés sur les diverses interfaces que nous réaliserons, il n'est pas inutile d'expliquer quelques-uns de ces termes et de voir, en bref, de quoi est fait un micro-ordinateur.

Comme chacun sait, le "cœur" de la machine est un microprocesseur. Mais seul, il ne serait pas d'une grande utilité. Il faut lui adjoindre des organes annexes : les plus connus sont la mémoire morte, qui contient généralement le langage basic et des sous-programmes assurant les diverses fonctions de la machine (affichage des caractères sur écran ou gestion du clavier) et la mémoire vive avec des données telles que les symboles utilisés ou le programme de l'utilisateur. Mais bien d'autres

organes — contrôleur vidéo, contrôleur de clavier ou modules d'entrée/sortie — font également partie de "l'architecture" de la machine.

Pour converser avec ces organes, le microprocesseur utilise un dispositif qui peut être comparé à notre réseau téléphonique. Il s'agit d'une série de fils — tracés sur le circuit imprimé de la machine — appelée "bus". Sur chacun de ces fils le microprocesseur pourra envoyer un "0" ou un "1", de manière à échanger les informations souhaitées.

Sur le bus, les connexions sont réparties en trois grandes catégories : les adresses, les données et

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1
soit :															
0	0	0	0	0	0	0	0	5	0	5	5	0	5	5	5
volts.															

les signaux de contrôle.

Sur tous les "micros", on peut accéder au bus depuis un connecteur placé à l'arrière de l'appareil (*voir photo*). Dans la plupart des cas il s'agit d'une simple fenêtre pratiquée dans le boîtier et laissant apparaître le circuit imprimé.

Les adresses.

Comme tout abonné au téléphone, chaque élément compo-

sant un "micro" possède en quelque sorte un numéro d'appel. Nous verrons plus loin que, grâce aux signaux de contrôle, le microprocesseur disposera d'éléments complémentaires pour appeler soit de la mémoire soit tout autre dispositif.

Le numéro d'appel est codé sur 16 fils. Pour les identifier, ils sont toujours numérotés de A0 à A15 sur le connecteur arrière ("A" pour "adresse", suivi du numéro de fil).

N'oublions pas qu'un ordinateur fonctionne en binaire et ne sait donc présenter que des 0 ou des 1 sur le bus. Pour trouver l'état des tensions (0 volt pour "0", 5 volts pour "1") présentes sur les fils d'adresses du bus pour un dispositif donné, il suffira de convertir un numéro d'appel choisi en binaire et de le reporter sur le bus en se rappelant que A0 représente le bit le plus faible (l'unité en quelque sorte). Par exemple, si l'adresse est 183, l'état du bus sera le suivant lorsqu'elle sera envoyée :

Notons qu'un périphérique peut avoir une adresse identique à celle d'une case mémoire. La distinction sera toujours effectuée grâce aux signaux de contrôle.

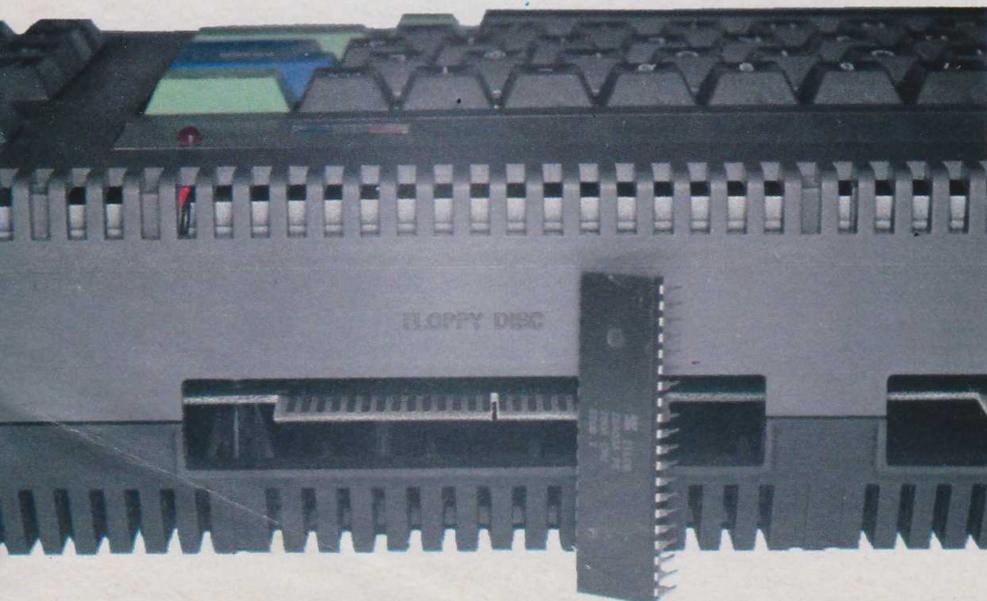
Comme, dans le cas de la mémoire, certaines adresses sont déjà utilisées par des éléments internes du micro-ordinateur tels que contrôleur vidéo, lecteur de cassettes ou de disquettes, contrôleur de clavier, imprimante, etc., avant d'envisager la réalisation d'une interface, il sera donc indispensable de vérifier si l'adresse que vous pensez lui donner est effectivement libre. En cas contraire, vous vous exposeriez à bien des surprises. Généralement le fascicule d'utilisation des micro-ordinateurs précise quelles sont les adresses déjà utilisées, souvent celles qu'il est judicieux d'employer.

Pour notre part nous tenterons toujours de réaliser des interfaces dont il sera facile de modifier l'adresse.

Les données.

Elles sont transmises sur huit fils. Le Z 80 ne sera donc capable de transmettre que huit bits simultanément, soit un octet. C'est sur

Au dos de votre "micro", le "bus", et, posée verticalement, l'unité centrale Z 80 qui équipe les machines que nous utiliserons.



ces fils que "la conversation" aura lieu. Dès que l'adresse aura été présentée, le Z 80 viendra échanger des informations sur les fils de données. Si, par exemple, c'est une case mémoire qui a été demandée il sera possible au microprocesseur, soit de lire son contenu, soit d'y enregistrer un octet. Il en serait de même pour un périphérique quelconque. D'une manière générale, les contacts correspondant aux données sont indiqués sur le schéma de brochage du connecteur par D0 à D7; D pour donnée. Comme dans le cas des adresses, le chiffre indiquera le numéro de fil. Tout comme précédemment, D0 correspondra au bit de poids faible. Si donc, à l'adresse dont nous avons parlé plus haut (adresse 183), l'information 17, par exemple, transite, nous trouverons simultanément : 0000000010110111 sur les adresses et 00010001 (soit 17) sur les données, répartis comme suit :

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	0	0	1
0	0	0	5	0	0	0	5

soit :
0 0 0 5 0 0 0 5 volts.

Comme on peut le constater, le principe de fonctionnement des fils de données est très proche de celui des fils d'adresses. Une différence fondamentale pourtant : si, sur les adresses, le microprocesseur ne peut qu'envoyer un message, sur les données par contre il peut soit en émettre (écrire), soit en recevoir (lire en mémoire).

Il s'agit donc de liaisons bidirectionnelles. La direction suivant laquelle seront acheminées les informations sera précisée, ici encore, par les signaux de contrôle. Il est donc évident qu'une interface devra prendre en compte la direction de conversation pour fonctionner correctement.

Les signaux de contrôle.

Ils sont nombreux. Cependant nous n'en utiliserons que quelques-uns. Ici, comme pour les adresses, il s'agit de signaux monodirectionnels ; c'est donc le microprocesseur qui les envoie ou les reçoit. Notons cependant que ces signaux sont inversés ; c'est-à-dire qu'ils prennent la valeur 0 lorsqu'il faut en tenir compte. Ce détail est indiqué en inscrivant une barre horizontale au-dessus de leur dénomination. Nous ne passerons ici en

revue que certains d'entre eux, qui nous seront le plus utile. Si, par la suite, nous sommes appelés à en utiliser d'autres nous indiquerons alors leur rôle précis.

- Le signal IORQ : Ce sera l'un des plus importants pour la réalisation d'interfaces. En effet lorsqu'il passe à 0, le microprocesseur indique par là qu'il va soit parler, soit écouter une interface. Son nom, peu évocateur en français il faut bien l'avouer, vient directement de l'abréviation de sa fonction en anglais : *Input Output ReQuest* ; c'est-à-dire demande d'entrée ou de sortie de données (via un périphérique ou une interface). Si, au contraire, le Z 80 avait voulu converser avec sa mémoire il aurait mis à 0 le signal MREQ : *Memory REQuest* ; demande mémoire.

Il s'agit donc là de deux signaux que le microprocesseur envoie.

- Le signal RD : il a pour effet d'indiquer le sens de la conversation sur les fils de données. Il signale que le microprocesseur est

INSTRUCTIONS PARTICULIÈRES POUR LE ZX 81

10 CLEAR 29999
15 REM routine OUT
20 POKE 30000,62
30 POKE 30001,0
40 POKE 30002,211
50 POKE 30003,254
60 POKE 30004,201
95 REM routine IN
100 POKE 30010,219
110 POKE 30011,254
120 POKE 30012,50
130 POKE 30013,68
140 POKE 30014,117
150 POKE 30015,201

à l'écoute et qu'il attend donc l'arrivée de données. Son appellation vient de *Read* ; lire, en anglais. Son complémentaire est le signal WR (*Write* ; écrire) qui indiquera que le microprocesseur va envoyer quelque chose sur les données ; donc interdiction absolue de tenter de parler. Là encore il s'agit de deux signaux émis par le microprocesseur.

- Le signal INT. Sa mise à 0 provoque l'arrêt momentané du Z 80. Il signifie INTerrupt et devra être utilisé avec prudence. Il s'agit d'un signal reçu par le Z 80.

Voici, pour les signaux qui nous intéresseront tout particulièrement.

Si le basic de votre "micro" comporte les instructions IN et OUT ;

elles se chargeront directement de piloter l'ensemble des signaux de BUS. IN sera utilisée pour questionner l'interface et OUT pour lui transmettre des données.

Cependant sur certaines machines, telles que le ZX 81 par exemple, ces instructions n'existent pas. Nous vous proposerons donc de réaliser un petit programme en langage machine pour les remplacer. Celui-ci est prévu pour fonctionner sur un "micro" comportant au moins 16 Koctets de mémoire vive (RAM).

Il commencera par une instruction CLEAR de manière à déclarer un RAMTOP, c'est-à-dire une zone mémoire où l'ordinateur n'aura pas le droit de modifier des données. Si vous disposez d'une machine dont la capacité mémoire est supérieure à 16 Koctets, la valeur du RAMTOP pourra être modifiée, mais il faudra alors prendre soin de changer l'adresse de chaque instruction POKE. Voici donc ce petit programme que nous commenterons au fur et à mesure :

- Mise en place du RAMTOP.

- Chargement du registre A du microprocesseur à la valeur 0, OUT de la valeur contenue dans A sur l'interface N.254. Retour au langage basic.

- IN : le registre A prend la valeur présentée par 254. Transfert de la donnée contenue dans A dans la case mémoire numéro 30020. Retour au langage basic.

Pour exécuter un OUT il faudra donc faire un POKE de l'adresse d'interface désiré en 30003, puis un POKE du contenu à transmettre en 30001 et enfin demander RANDOMISE, ou CALL, USR 30000.

De même pour un IN il suffira d'indiquer l'adresse de l'interface à questionner par un POKE de son adresse en 30011 et la donnée reçue sera obtenue en effectuant un PEEK en 30020. Ici l'instruction RAND, ou CALL, USR 30010 devra être insérée entre le POKE et le PEEK.

Nous sommes donc à présent prêts à attaquer notre première réalisation. Préparez vos outils à "wrapper", supports et connecteurs pour le mois prochain.

Henri-Pierre PENEL