

La plupart des ordinateurs domestiques ont le BASIC pour "langue maternelle". En fait, il serait plus exact de parler *des* BASIC, car presque chaque système à son dialecte propre. Ce qui rend à peu près impossibles les échanges directs (par modem ou cassette) de programmes entre ordinateurs différents (cette impossibilité existe même parfois entre systèmes issus d'une même famille). Dura lex, sed lex... c'est la loi du marché ! Plutôt que de nous y attarder, contournons cet obstacle avec le Basicode qui est un *standard de communication universel conçu pour l'échange direct de programmes en BASIC* entre systèmes par ailleurs sans aucune affinité élective et apparemment inconciliables. S'agit-il d'une velléité promise à un échec prochain ou est-ce le début d'un nouvel espéranto ? C'est à vous, lecteurs, d'en décider.

# basicode-2

un standard  
pour  
l'échange de  
programmes  
écrits en  
BASIC!

Il y a deux ans environ, les responsables d'une émission de radio d'une chaîne néerlandaise émettaient l'idée de créer un standard pour réconcilier les BASIC ennemis et les ordinateurs de marque et de conception différentes. Le premier obstacle rencontré surgit au niveau du vecteur de transmission utilisé, à savoir le type de modulation du signal émis et/ou enregistré pour le transfert de données. Le deuxième obstacle de taille est la diversité des dialectes BASIC et leur irréductible incompatibilité dès qu'il s'agit de programmes un tant soit peu élaborés. Fallait-il abandonner l'idée ?

Il faut dire qu'il y avait là de quoi décourager les plus obstinés... Le temps a passé, et il existe désormais le standard Basicode qui porte au premier chef sur la mise en forme du signal audio de transmission. Cette standardisation autorise n'importe quel micro-ordinateur à charger des programmes émis par n'importe quel autre micro-ordinateur. Mais le standard Basicode ne se contente pas de régir le format du signal audio enregistré sur bande ou émis par radio ou téléphone, mais aussi, et c'est non moins important, la structure même des programmes: il impose une limitation du jeu d'instructions utilisables, une numérotation normalisée des lignes, un choix des variables et une gestion déterminée de l'écran.

Bien que jeune, le Basicode existe déjà en deux versions, dont la seconde fait appel à un certain nombre de routines standardisées elles aussi. C'est à cette seconde version que nous nous attacherons, puisque l'on peut la considérer comme plus "universelle" que la première. D'où l'appellation "Basicode-2"...

## Le signal de transmission selon Basicode

Les deux fréquences utilisées par le format Basicode sont 1200 et 2400 Hz. Un niveau logique bas correspond à une période complète de 1200 Hz, tandis qu'un niveau logique haut correspond à deux périodes

complètes de 2400 Hz. La transmission est sérielle et le taux est de 1200 bauds. Le format utilisé est le suivant:

- 1 bit de départ (niveau logique bas)
- 8 bits de données (le bit de poids le plus faible en premier)
- 2 bits d'arrêt (niveau logique haut).

Le fichier BASIC est transmis caractère par caractère, exactement sous la forme obtenue à l'aide d'une instruction LIST. Il n'est pas fait usage de la notation interne au système: tous les chiffres, toutes les lettres et tous les autres caractères et signes apparaissent sous la forme de leur code ASCII (à noter que le huitième bit - de poids le plus fort - est toujours forcé au niveau logique haut). Chaque instruction BASIC doit être suivie d'un espace, tandis que chaque ligne doit être achevée par un CR (retour chariot; hex 8D).

Un programme complet est transmis comme suit:

- un signal d'en-tête de 5 secondes consistant en une fréquence fixe de 2400 Hz
- un caractère ASCII STX (start of text; début de texte) hex 82
- la séquence des codes ASCII de tous les caractères du programme BASIC
- un caractère ASCII EOT (end of text; fin de texte) hex 83
- une somme de vérification
- un signal de fin de transmission de 5 secondes consistant en une fréquence fixe de 2400 Hz.

La somme de vérification est un procédé bien connu qui permet, comme son nom l'indique, de comparer rapidement les caractères reçus aux caractères émis. La somme consiste en une opération EXOR appliquée systématiquement à chaque octet transmis sans qu'il soit tenu compte de la retenue, de sorte que la somme de vérification ne comporte toujours qu'un seul octet, quelque soit le nombre d'octets transmis et leur somme arithmétique réelle.

## Le protocole Basicode-2

### Conventions générales

Les instructions admises sont celles d'une police supposée commune à tous les sys-

tèmes susceptibles de communiquer à l'aide du protocole Basicode. Le tableau 1 en donne la liste (sur laquelle nous reviendrons plus tard).

Certains numéros de lignes sont réservés à des routines à fonction déterminée, notamment des opérations qu'il n'est pas possible d'effectuer directement en BASIC standard. C'est encore un point dont nous aurons à reparler. Ces routines ne sont pas transmises avec le programme BASIC; elles font partie du programme de conversion Basicode et devront être chargées séparément avant l'exécution du programme transmis.

Le format de l'écran est fixé à 24 lignes de 40 caractères. Comme il existe des systèmes aux performances inférieures à ce standard exigeant, il est préférable de ne pas utiliser plus de 16 lignes par écran, ainsi que les maintenir aussi courtes que possible. Une ligne ne doit pas comporter plus de 60 signes, numéro de ligne, espaces et CR inclus.

#### La structure des programmes

Le protocole de numérotation des lignes selon Basicode-2 est le suivant:

**0 à 999:** zone réservée aux routines ("obligées") spécifiques au système pour lequel elles ont été écrites. Elles sont fournies par le programme de conversion ou sont chargées séparément.

**1000:** première ligne du fichier proprement dit. Sa forme est obligatoirement la suivante:

1000 A = valeur: GOTO 20

REM nom du programme

où valeur est le nombre maximal de caractères utilisés par toutes les chaînes de caractères. Le saut vers la ligne 20 permet aux ordinateurs qui en ont besoin de réserver une zone de mémoire pour les chaînes de caractères.

**1010:** première ligne disponible pour le programme proprement dit.

**1010 à 32767:** zone disponible pour ce programme.

La structure interne des programmes n'a pas été uniformisée, bien qu'il existe quelques recommandations auxquelles il est conseillé d'accorder le plus possible

d'attention:

**1000 à 19999:** zone réservée au programme principal

**20000 à 24999:** zone réservée aux sous-programmes du programme principal, comportant des instructions interdites par Basicode-2.

**25000 à 29999:** données (data statements)

**30000-32767:** instructions REM et commentaires; à la fin, nom et adresse de l'auteur du programme.

L'incrémentement recommandé pour les numéros de ligne est de 10. Les sous-programmes de 20000 à 24999 doivent être évités autant que possible; s'il n'y a aucun moyen de s'en passer, il faut veiller à les commenter efficacement.

#### Les routines standard de Basicode-2

La structure de ces routines est, dans une large mesure, spécifique au système. De sorte que l'on ne trouvera aucun exemple détaillé ci-après, mais uniquement une description de leur fonction.

**GOSUB 100:** Cette routine efface le contenu de l'écran et remet le curseur en haut à gauche.

**GOSUB 110:** Avec cette routine, on place le curseur à un endroit déterminé; la position souhaitée doit figurer dans les variables HO (horizontal) et VE (vertical). Comme le format adopté est de 24 lignes de 40 caractères, les valeurs de HO et VE seront respectivement comprises entre 0 et 39, et entre 0 et 23. Les valeurs de HO et VE ne doivent pas être modifiées par cette routine.

**GOSUB 120:** Cette routine place les coordonnées du curseur dans les variables HO et VE; HO = 0 correspond à l'extrême gauche d'une ligne, tandis que VE = 0 correspond à la première ligne en haut de l'écran. Cette routine, associée à la précédente, pourra être utilisée pour déplacer le curseur.

**GOSUB 200:** Cette routine détermine si une touche est enfoncée et la mémorise, le cas échéant, à l'aide de la variable INS. Si aucune touche n'est enfoncée, INS est vide. En principe les caractères de commande peuvent être utilisés aussi, mais avec circonspection, car tous les systèmes ne les interprètent pas forcément de la même manière (à l'exception de la commande RETURN dont le code ASCII est 13 (décimal)).

**GOSUB 210:** Cette routine attend qu'une touche soit actionnée et la mémorise à l'aide de la variable IN\$. Notez que la routine précédente n'attend pas!

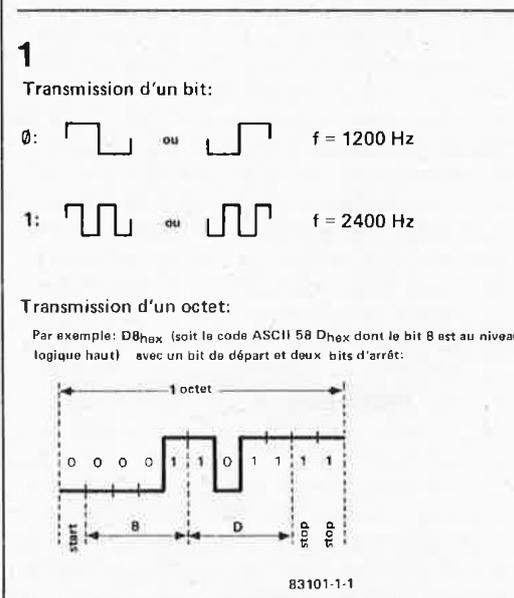
**GOSUB 250:** Lorsque c'est possible, cette routine assure l'émission d'un signal sonore dont la hauteur et la durée ne sont pas uniformisées.

**GOSUB 260:** Cette routine attribue à la variable RV un nombre aléatoire entre 0 et 1.

**GOSUB 270:** Cette routine calcule le nombre d'octets encore disponibles pour l'attribution de variables (les variables existantes ne sont pas effacées). La variable FR est attribuée à ce nombre.

**GOSUB 300:** cette routine fait de la variable SR une chaîne SR\$, laquelle ne doit comporter d'espace ni au début ni à la fin.

Figure 1. Le format de transmission selon Basicode-2. On notera que la transmission d'un octet commence toujours par le bit de poids le plus faible.



La déclaration STR\$ n'est pas autorisée par Basicode-2.

**GOSUB 310:** Cette routine établit une chaîne SR\$ de la façon suivante: le contenu de SR\$ est égal à la valeur de la variable SR, et ceci en notation à virgule fixe. La longueur totale de SR\$ compte CT caractères; le nombre de caractères après la virgule est déterminé par CN. Lorsque le nombre ne peut se conformer au format indiqué, SR\$ comporte CT astérisques. Si nécessaire, il y a arrondissement de SR. CT, CN et SR ne sont pas modifiés par cette routine. Par exemple: CT = 7, CN = 3 et SR = 0.6666; SRS sera " 0,667".

**GOSUB 350:** Cette routine assure l'impression de SR\$ sur l'imprimante, sans instruction de fin de ligne, de sorte qu'il reste possible d'accoler plusieurs chaînes de caractères sur une même ligne.

**GOSUB 360:** Emission des commandes CR et LF après une ligne à imprimer.

#### Variables

Pour garantir une interchangeabilité raisonnable des programmes, il a fallu fixer quelques règles d'attribution des variables:

- les variables numériques sont réelles et à simple précision;
- une variable ne sera jamais désignée par plus de deux caractères dont le premier doit toujours être une lettre. Pour les chaînes, on rajoute le signe "\$". L'utilisation des minuscules est prohibée dans la désignation des variables;
- pour les variables logiques, il n'est permis d'effectuer que des opérations "vrai ou faux", à l'exclusion de toute utilisation de valeurs de substitution éventuellement attribuées par l'ordinateur, comme par exemple + 1 pour "vrai", et 0 pour "faux";
- il ne convient pas de considérer que le lancement d'un programme assure automatiquement l'initialisation des variables;
- les chaînes ne peuvent comporter plus de 255 caractères;
- la première des deux lettres désignant une variable ne doit jamais être un "O", celui-ci étant réservé aux routines spécifiques au système, dites routines standard ou "obligées";
- les combinaisons de lettres suivantes sont interdites pour la désignation de variables: AS, AT, FN, GR, IF, PI, ST, TI, TI\$ et TO;
- pour les échanges entre le programme BASIC et les routines standard, il convient de faire usage des variables suivantes: HO, VE, FR, SR, CN, CT, RV, IN\$ et SR\$.

#### Autres limitations importantes

Le tableau 1 donne les instructions, opérateurs et déclarations utilisables avec Basicode-2. Même en-deçà de ces limites, les possibilités de variantes sont nombreuses, et une uniformisation est nécessaire. Le fond est toujours le même, aussi importe-t-il que la forme le soit aussi !

C'est pourquoi nous mentionnerons encore quelques détails importants. Ainsi, l'instruction GOSUB et l'instruction GOTO ne sont

pas acceptées avec une variable comme destination; la forme A = 1000: GOTO A est par conséquent à proscrire.

IF sera toujours suivi de THEN.

Par exemple IF...THEN A = 5 ou IF...THEN GOSUB 20000, ou encore IF...THEN 1000; tandis que la forme IF...THEN...ELSE est interdite.

Un INPUT ne doit jamais être suivi ni d'un texte à imprimer, ni de plus d'une seule variable.

Ainsi la forme INPUT "La valeur ="; A\$ est-elle interdite.

L'instruction RUN n'assure pas l'initialisation des variables. Il n'est pas permis de faire suivre l'instruction RUN par un numéro de ligne.

Avec l'instruction TAB, il ne faut pas oublier que certains ordinateurs commencent à compter à zéro, tandis que d'autres commencent à un !

#### La pratique

Voilà pour ce qui concerne l'essentiel des conventions à respecter si l'on veut utiliser le standard Basicode-2 pour la transmission de programmes en BASIC. Il resterait à examiner le programme de conversion et les routines "obligées" ou standard; mais comme celles-ci sont, par définition, spécifiques à chaque ordinateur, il est impossible de les détailler toutes. Un article entier a été consacré ailleurs dans ce numéro aux routines "obligées" du Junior Computer; nous y renvoyons le lecteur désireux de pratiquer le Basicode-2 avec le Junior Computer.

Aux Pays-Bas et, dans une moindre mesure, en République Fédérale d'Allemagne, la pratique du Basicode est généralisée, ou tout au moins bien rodée, alors qu'en France tout reste à faire. Dans les deux pays mentionnés ci-dessus, il existe des émissions de radio et de télévision consacrées (en partie) à l'émission de programmes à l'aide de ce standard.

Les responsables de l'émission Hobbyscoop de la radio néerlandaise, avec qui nous avons établi de bons contacts, nous ont déjà signalé que, ici ou là, des auditeurs français avaient manifesté leur intérêt. C'est bon signe ! En tous cas, voici les coordonnées de l'émission: dimanche soir, 19h10, Hilversum 2, NOS "Hobbyscoop", en modulation de fréquence (ne peut être reçue en France) et en modulation d'amplitude (ondes moyennes, 747 kHz) dont la réception est possible dans les régions septentrionales de notre pays et sans difficulté en Belgique. Ce numéro comporte également un article dans lequel nous décrivons un circuit de remise en forme du signal FSK spécialement conçu pour permettre une bonne transmission des programmes, même lorsque les conditions de réception ne sont pas idéales. Voici d'autre part l'adresse des responsables de l'émission: Hans G. Janssen/Hobbyscoop, Postbus 1200, 1200 BE Hilversum, Pays-Bas. C'est à cette adresse que l'on peut notamment se procurer un fascicule contenant le protocole Basicode-2 complet (n'oubliez pas de signaler que c'est Elektor qui vous a mis la puce à l'oreille !).

ABS  
AND  
ASC  
ATN  
CHR\$  
COS  
DATA  
DIM  
END  
FOR  
GOSUB  
GOTO  
INT  
IF  
INPUT  
LEFT\$  
LEN  
LET  
LOG  
MID\$  
NEXT  
NOT  
ON  
OR  
PRINT  
READ  
REM  
RESTORE  
RETURN  
RIGHT\$  
RUN  
SGN  
SIN  
SQR  
STEP  
STOP  
TAB  
TAN  
THEN  
TO  
VAL

+     ↑     <>  
-     =     <=  
\*     <     >=  
/     >

Tableau 1. Instructions, déclarations et opérateurs utilisables avec Basicode-2.