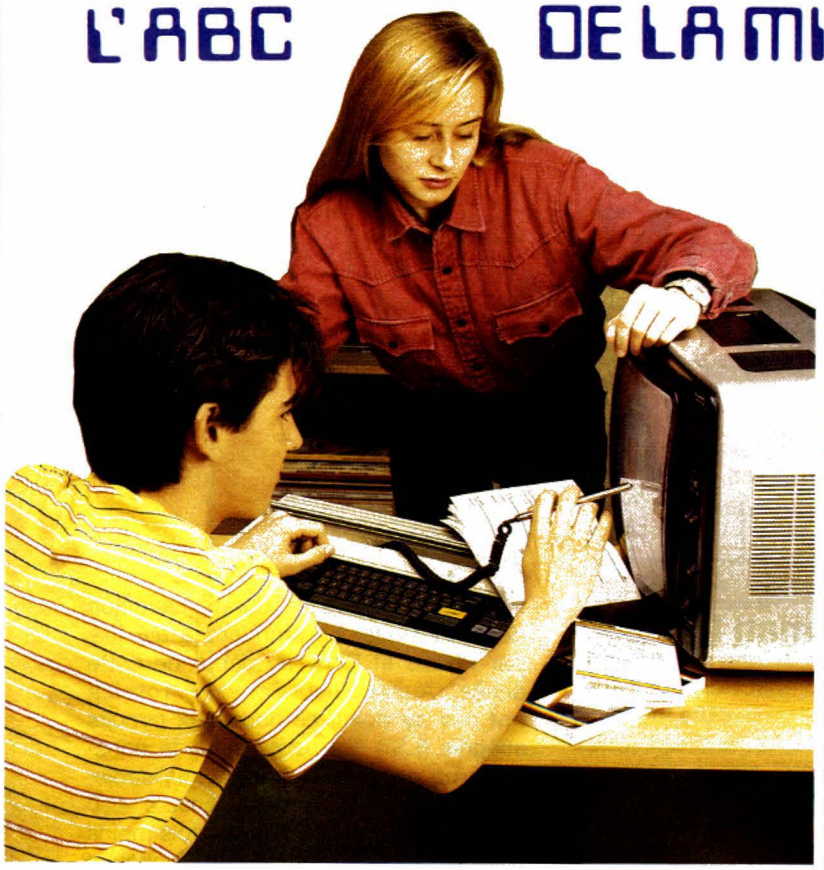
L'ABC DE LA MICRO-INFORMATIQUE



Nous poursuivons aujourd'hui nos exemples d'exploitation des instructions graphiques avec le tracé de fonctions mathématiques. Nous aborderons ensuite rapidement le cas particulier des instructions « sonores », avant de voir quelques exemples de programmes Basic utiles ou originaux.

Le tracé de fonctions mathématiques

Toute personne ayant approché, même de loin, les mathématiques s'est trouvé confrontée un jour ou l'autre au problème de la représentation graphique d'une fonction. Cette fonction pouvait très souvent être exprimée sous une des deux formes suivantes :
X = F (X), où F est une fonction quelconque ;
X = F (T) et Y = G (T), où F et G sont des fonctions quelconques.

La représentation graphique de ces fonctions est plus ou moins facile selon les instructions disponibles sur le micro-ordinateur utilisé mais, en principe, elle est toujours possible avec un peu de bon sens. En effet, comme nous l'avons vu le mois dernier, toutes les machines dotées de possibilités graphiques disposent au moins d'une instruction capable de positionner un point en fonction de ses coordonnées par rapport à un angle de l'écran utilisé comme origine. La transposition des équations ci-avant est donc relativement rapide, comme nous allons le voir sur quelques exemples adaptés pour l'EXL 100 d'Exelvision puisque c'est l'appareil que nous avons choisi pour « cobaye » le mois dernier. Nos deux formes d'équations vont pouvoir s'écrire de la façon très générale présentée en figures 1 et 2. La ligne 100 de la figure 1 précise la plage de variation de X tandis que la ligne 110 n'est autre que l'expression de la relation fonctionnelle entre X et Y. Bien sûr, il faut que cette relation puisse être exprimée avec les instructions scientifiques dont dispose l'interpréteur, ce qui est possible dans la majorité des cas (quitte à définir si nécessaire une fonction particulière avec un sous-programme ou un DEF FN comme expliqué dans nos précédents articles). Le « calcul » du tracé est alors réduit à la ligne 120 où l'instruction PLOT se charge de tout le travail.

La figure 2 est organisée de la même façon, la ligne 100 précise la plage de variation du paramètre T tandis que les lignes 110 et 120 indiquent les deux relations fonctionnelles.

Le seul problème qui reste à résoudre pour que ces exemples soient utilisables pratiquement est celui du cadrage. En effet, dans le cas de l'EXL 100 choisi en exemple, l'abscisse d'un point sur l'écran ne peut varier que de 0 à 319, et l'ordonnée de 0 à 10 × N–1 où N est le nombre de lignes mises en haute résolution. Le point de coordonnées 0.0 est, par ailleurs, en haut et à gauche de l'écran. De ce fait, les coordonnées utilisées dans les CALL PLOT des exemples précédents seront rarement X et Y directement, mais plutôt des valeurs du style (X–M)/N où M et N sont des facteurs de cadrage tenant compte des valeurs extrêmes de la fonction à représenter et des possibilités de l'écran.

Pour concrétiser cela, voici un exemple très simple de tracé de courbe du troisième degré avec représentation des axes sur l'écran, positionnement de l'origine au centre de l'écran et graduation des axes. Rassurez-vous, cela ne nécessite que quelques lignes comme vous pouvez le constater à la lecture du listing présenté figure 3. Signalons à

```
100 FOR X = X0 TO X1 STEP Z
110 Y = F(X)
120 CALL PLOT ("COULEUR",X,Y)
130 NEXT
```

FIGURE 1
Tracé de y = f(x) avec un EXL 100.

```
100 FOR T = T0 TO T1 STEP Z
110 X = F(T)
120 Y = G(T)
130 CALL PLOT ("COULEUR",X,Y)
140 NEXT
```

FIGURE 2
Tracé de x et y fonctions du paramètre t avec un EXL 100.

messieurs les lycéens qui nous lisent que ce programme à l'air ridicule résoud le problème classique : « Tracé de la courbe représentative de la fonction F(X) », comme quoi un micro-ordinateur domestique peut rendre bien des services !

L'analyse de ce programme est relativement simple compte tenu du fait que nous l'avons dépouillé à l'extrême. Tout d'abord, sachez que nous avons décidé de tracer la courbe représentative de la fonction
$F(X) = 2X^3 + 2X^2 - 5X + 1$.

```
100 CALL HRON ("Y",1,20)
110 CALL LINE ("M",160,0,160,199)
120 FOR Y=-10 TO 30
130 CALL LINE ("M",158,150-4*Y,162,150-4*Y)
140 NEXT
150 CALL LINE ("M",0,150,319,150)
160 FOR X=-2 TO 2
170 CALL LINE ("M",160+60*X,148,160+60*X,152)
180 NEXT
190 FOR X=-2.5 TO 2.5 STEP .01
200 Y=2*X↑3+2*X↑2-5*X+1
210 CALL PLOT ("b",160+60*X,150-4*Y)
220 NEXT
```

FIGURE 3

Exemple de tracé de courbe du troisième degré avec axes centrés et gradués.

```
100 CALL HRON ("Y",1,20)
110 FOR Y=-12 TO 37
120 CALL LINE ("G",0,150-4*Y,319,150-4*Y)
130 NEXT
140 FOR X=-2 TO 2
150 CALL LINE ("G",160+60*X,0,160+60*X,199)
160 NEXT
170 CALL LINE ("M",0,150,319,150)
180 CALL LINE ("M",160,0,160,199)
190 FOR X=-2.5 TO 2.5
200 Y=2*X↑3+2*X↑2-5*X+1
210 CALL PLOT ("b",160+60*X,150-4*Y)
220 NEXT
```

FIGURE 4

Le programme de la figure 3 avec quadrillage de tout l'écran pour faciliter la lecture des coordonnées des points.

La ligne 100 place l'écran en mode haute résolution en utilisant la taille maximum permise afin de disposer de la courbe la plus grande possible. La ligne 110 trace l'axe des ordonnées (ou axe des Y si vous préférez), que nous avons placé au centre de l'écran. La boucle des lignes 120 à 140 trace les graduations de cet axe au moyen de courts traits horizontaux. Le pas de la graduation qui est ici unitaire peut être modifié en ajoutant un STEP N après le FOR Y=−10 TO 30. La ligne 150 trace l'axe des abscisses qui est ensuite gradué lignes 160 à 180 selon la même méthode que celle utilisée pour les ordonnées.

Ensuite, compte tenu de la fonction choisie, de ses valeurs extrêmes et de la portion de courbe que l'on souhaite visualiser, le cadrage du tracé est réalisé ligne 190 pour les variations de X et ligne 210 pour la fonction elle-même. La détermination des constantes utilisées ligne 210 a été faite en calculant les valeurs extrêmes de la fonction pour les valeurs extrêmes de X et en prenant en compte les tailles horizontale et verticale de l'écran en nombre de points. En effet, X varie de − 2,5 à + 2,5 (parce que nous voulons étudier cette partie de la courbe) ; il faut donc, compte tenu de la position des axes, que pour X = 0 on obtienne une abscisse de 160, pour X = − 2,5, on obtienne une abscisse aussi proche de 0 que possible, et que pour X = + 2,5, on obtienne une abscisse aussi proche que possible de 319. D'où l'équation retenue 160 + 60*X qui satisfait au mieux ces conditions. Pour les ordonnées, le même procédé est utilisé en déterminant au préalable les valeurs extrêmes de la fonction (utilisation du Basic en mode direct pour ce faire par exemple).

Pour en terminer avec cet exemple et faciliter un peu son exploitation, nous vous proposons en figure 4 une version améliorée de son listing avec tracé des axes de graduations sur tout l'écran, ce qui permet de lire très facilement les coordonnées d'un point. L'utilisation de diverses couleurs permet de rendre ce tracé très agréable à lire et nous ne nous en sommes donc pas privés.

Comme vous pouvez le constater à la lumière de ces explications, le tracé de courbes représentatives de fonctions est très simple, même avec un micro-ordinateur disposant d'instructions graphiques très limitées, puisqu'il suffit de savoir allumer un point de l'écran en fonction de ses coordonnées. Le reste n'est plus qu'une question de cadrage.

Les instructions sonores

Les instructions « sonores » ou, pour être plus correct, celles relatives aux circuits de génération de sons dont sont équipés les micro-ordinateurs, souffrent du même mal que les instructions graphiques : l'absence de standardisation. Les micro-ordinateurs du marché sont en effet équipés de divers systèmes de génération de sons aux possibilités parfois très éloignées les unes des autres. Ainsi, pour reprendre nos trois exemples du mois dernier, le Thomson MO5, l'EXL 100 Exelvision et le « vieil » Oric, nous avons à faire à trois systèmes différents.

Sur le Thomson MO5, le plus pauvre en ce domaine, il n'existe qu'un générateur à une seule voie, capable de produire des notes de durée programmable sur une plage de 5 octaves. Une seule instruction est chargée de piloter ce générateur en fonction des informations contenues dans la chaîne de caractères qui la suit. Ainsi doit-on écrire PLAY « chaîne de caractères » où la chaîne de caractères peut contenir le nom des notes désirées (DO, RE, etc.), des instructions de silence (P), des instructions de changement d'octave (O1 à O5), des instructions de durée de note (L1 à L96), des instructions de

tempo (T_1 à T255) et une instruction de programmation de l'attaque de la note (A0 à A255). Cette syntaxe est tout à la fois souple par ses possibilités, mais assez lourde à manipuler selon l'effet désiré vu le nombre de caractères à utiliser.

Sur l'EXL 100, la situation est tout à fait différente puisque cet appareil est équipé d'un synthétiseur vocal. Il est donc très difficile de faire jouer facilement de la musique à cet appareil mais, par contre, il est théoriquement possible de le faire parler. Une instruction est prévue à cet effet : CALL SPEECH, suivie par une chaîne de caractères représentant le codage numérique des informations à destination du synthétiseur. Cette chaîne de caractères ne peut malheureusement pas être créée par l'utilisateur en fonction de ce qu'il veut faire dire à l'appareil et, à moins de s'en tenir aux exemples fournis dans le manuel ou de posséder un logiciel spécialement prévu à cet effet, il est quasiment impossible de faire parler un EXL 100. Quoi qu'il en soit, la syntaxe à utiliser n'a rien à voir avec celle du MO 5.

Notre dernier exemple, malgré son âge et son prix maintenant très compétitif, est celui qui dispose des fonctions de générations de sons les plus étendues. De plus, le circuit qu'il utilise pour ce faire est presque devenu un standard de l'industrie et se retrouve sur de nombreuses machines dont, en particulier, tous les micro-ordinateurs au standard MSX. Les instructions de pilotage sont cependant différentes d'une machine à une autre. Ce synthétiseur sonore dispose de trois voies indépendantes pouvant fonctionner simultanément, de générateurs de bruits blancs et d'un générateur d'enveloppes pouvant produire sept formes différentes. Deux instructions principales sont utilisées pour manipuler tout cela : MUSIC sous la forme MUSIC numéro de canal, octave, note, volume, et PLAY sous la forme PLAY canal de bruit, canal de son, enveloppe, période. Des tableaux définissent les relations entre les chiffres qui doivent suivre ces instructions et les différentes enveloppes et canaux utilisables. En outre, et c'est une bonne idée, un certain nombre de bruits « standards » sont prédéfinis et peuvent être appelés par un mot clé particulier (PING, ZAP, SHOOT, EXPLODE).

Comme vous pouvez donc le constater, il règne dans le monde des instructions sonores un manque de standardisation encore plus important que celui que nous avions rencontré dans le domaine des graphiques. Heureusement, l'utilisation de ces instructions n'est importante que dans les programmes de jeux, et il est donc possible soit de s'en passer, soit de transposer d'une machine à une autre vu le faible nombre de lignes de programme concernées.

Quelques exemples de programmes

Arrivé à ce stade de notre initiation au Basic et avant de passer à un autre sujet avec la programmation en langage machine ou assembleur (selon l'expression que vous préférez), nous vous proposons quelques exemples de programmes simples et utilisables sur la majorité des machines du marché.

Le Haut-Parleur étant avant tout une revue d'électronique, nous allons commencer par un programme pratique avec le...

Calcul des paramètres d'un monostable

Un circuit très utilisé en logique est le monostable de la famille TTL 74123. Ce circuit comporte, dans un seul boîtier, 16 pattes, deux monostables indépendants capables de générer des impulsions positives ou négatives de durées étagées de quelques nanosecondes à plusieurs secondes. Le réglage de la durée de l'impulsion se fait au moyen de deux composants externes : une résistance (de valeur inférieure à 50 kΩ si on veut des temps reproductibles) et un condensateur. Des abaques indiquent la relation durée de l'impulsion - valeur de la résistance - valeur du condensateur mais, outre le fait que leur tracé est assez peu précis, elles ne couvrent pas toujours la plage de valeurs que vous souhaiteriez utiliser. Nous avons donc fait appel à une formule empirique de détermination des éléments, formule empirique qui est admise comme correcte pour les circuits 74123 de la famille TTL normale. Cette formule est la suivante :

$$t = 0{,}28 \times R \times C\,(1 + 0{,}7/R)$$

avec t en nanosecondes, R en kilo-ohms et C en picofarads.

Il est très simple d'écrire un programme Basic qui calcule cette formule mais il est encore plus simple de faire cela avec n'importe quelle calculette de poche ; nous avons donc compliqué un peu le problème pour rendre le programme plus utile. En effet, on dispose généralement du temps et on souhaite calculer les éléments R et C, mais on tombe rarement sur des valeurs normalisées et encore plus rarement sur des valeurs de composants dont on dispose. La méthode de tout individu confronté à ce problème est donc la suivante : il calcule des valeurs de R et C pour un temps donné, il prend les valeurs calculées, cherche les valeurs les plus proches dans son stock et calcule le temps obtenu avec ces valeurs pour voir s'il est acceptable. Celui qui a pratiqué cette méthode une dizaine de fois de suite sait combien la frappe sur le clavier de la calculette devient source d'erreur au fur et à mesure que le temps passe. Notre programme peut donc calculer un des trois paramètres à partir des deux autres et ce, inlassablement et très rapidement. Comme c'est le premier programme que nous établissons, et pour ne pas faire mentir nos articles précédents, nous vous présentons en figure 5 son organigramme.

La première opération consiste à demander quel est le type de problème à résoudre ; selon la réponse fournie, une des trois branches du programme est choisie. Ces trois branches sont identiques dans leur esprit et très simples : nous n'en commenterons donc qu'une seule ; en effet, les deux paramètres nécessaires au calcul sont demandés, la formule leur est appliquée et le résultat est affiché. Les trois branches se rejoignent ensuite pour demander si un autre calcul est nécessaire et, si la réponse est oui, on revient en début de programme.

La concrétisation de cet organigramme est présentée figure 6 et suit rigoureusement la démarche décrite. Remarquez l'utilisation de commandes INPUT suivies de textes qui permettent d'économiser des PRINT suivis de INPUT, permettant ainsi de réaliser un programme plus compact. Le choix de la branche de programme a lieu lignes 30, 40 et 50 ; la ligne 60 faisant poser à nouveau la question initiale si aucune des réponses fournies ne convient.

Les trois branches commencent ensuite en 70, 120 et 170 pour se rejoindre en un point commun qui est ligne 210 et qui fait imprimer la question « un autre calcul ». Dans chaque branche la partie « active » du programme se trouve au niveau du LET (lignes 90, 140 et 190) où la formule empirique est calculée à partir des valeurs fournies. Toutes les autres lignes du programme constituent les lignes de dialogue. Comme vous pouvez le constater, et c'est le propre de 90 % des programmes Basic, les entrées/sorties ou, plus généralement, le dialogue avec l'opérateur occupe la majeure partie du programme, les lignes de calcul proprement dit étant très peu nombreuses.

Malgré sa simplicité, ce programme présente quelques particularités. Au niveau des lignes demandant une réponse, remarquez que le programme précise, grâce à une formulation aussi complète que possible de la question, le type de donnée qu'il attend ainsi. Par exemple, lorsque l'on demande la valeur de la résistance, la question précise que c'est en kilo-ohms qu'il faut la fournir : un détail qui fait toute la différence entre un programme agréable d'emploi et utilisable par tout un chacun sans notice et un programme hermétique affichant une suite de points d'interrogation lors des entrées de données. Nous aurions pu remplacer les lignes 60 et 70 par un INPUT R, C ; l'utilisateur du programme aurait alors vu s'afficher un point d'interrogation ; à lui de savoir s'il faut frapper R ou C en ohms ou en kilo-ohms, etc. Cet aspect conversationnel d'un programme n'est pas à négliger car il permet de réduire notablement, voire de rendre inutile, tout mode d'emploi du programme.

Remarquez, ligne 230, l'utilisation de LEFT qui permet de répondre O ou OUI avec le même effet.

Remarquez aussi que ce programme, pour être tout à fait opérationnel, doit être complété par quelques tests que nous vous laissons le soin d'ajouter ; en particulier il est nécessaire de vérifier que les valeurs frappées pour T, R et C sont positives, que R est inférieure ou égale à 50 kΩ (si l'on veut respecter les spécifications du 74123) et que C et T n'ont pas des valeurs démentes (imposer C inférieur à 1 000 µF par exemple et T à 100 secondes).

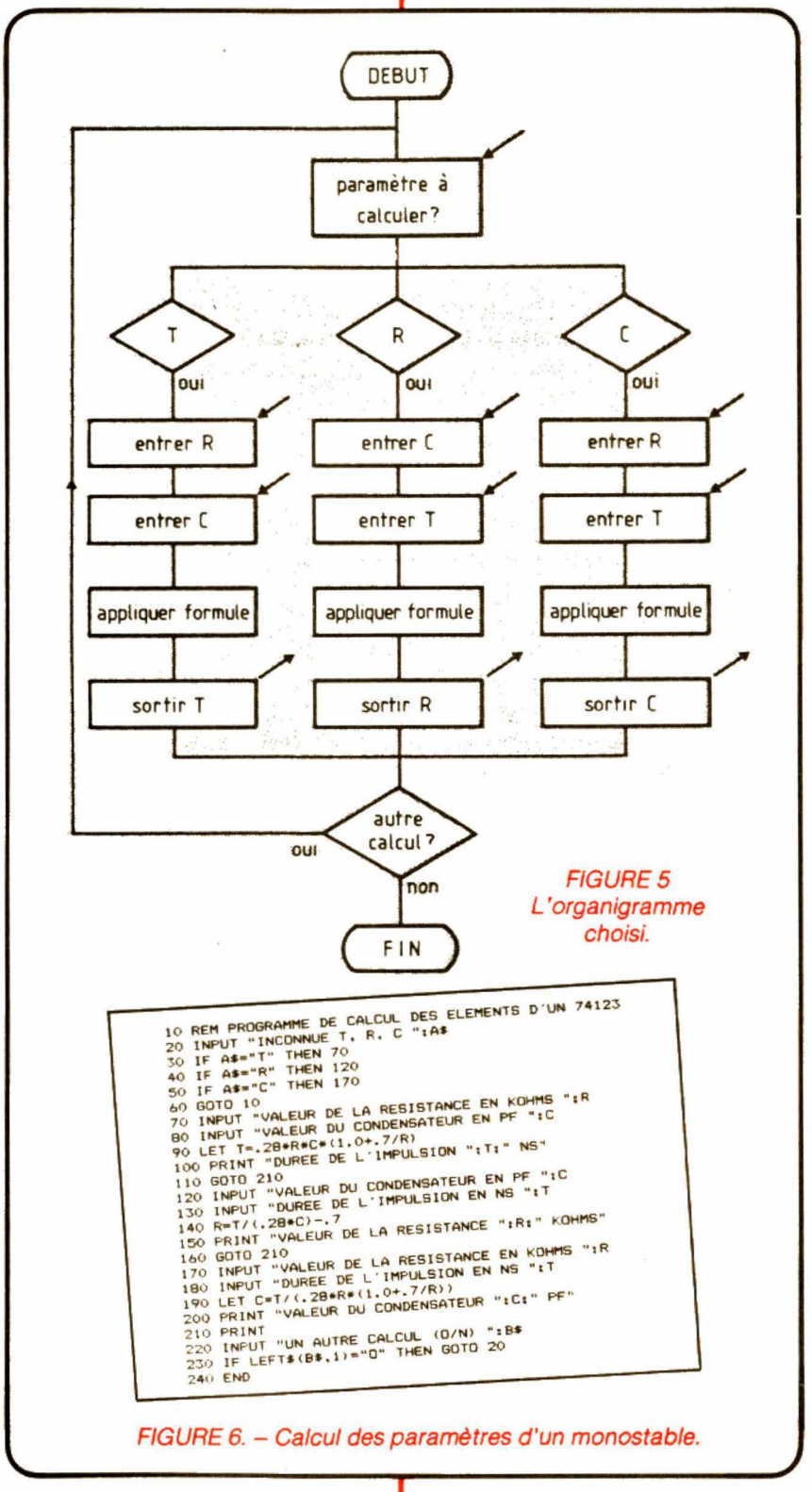


FIGURE 5
L'organigramme choisi.

```
10 REM PROGRAMME DE CALCUL DES ELEMENTS D'UN 74123
20 INPUT "INCONNUE T, R, C ";A$
30 IF A$="T" THEN 70
40 IF A$="R" THEN 120
50 IF A$="C" THEN 170
60 GOTO 10
70 INPUT "VALEUR DE LA RESISTANCE EN KOHMS ";R
80 INPUT "VALEUR DU CONDENSATEUR EN PF ";C
90 LET T=.28*R*C*(1.0+.7/R)
100 PRINT "DUREE DE L'IMPULSION ";T;" NS"
110 GOTO 210
120 INPUT "VALEUR DU CONDENSATEUR EN PF ";C
130 INPUT "DUREE DE L'IMPULSION EN NS ";T
140 R=T/(.28*C)-.7
150 PRINT "VALEUR DE LA RESISTANCE ";R;" KOHMS"
160 GOTO 210
170 INPUT "VALEUR DE LA RESISTANCE EN KOHMS ";R
180 INPUT "DUREE DE L'IMPULSION EN NS ";T
190 LET C=T/(.28*R*(1.0+.7/R))
200 PRINT "VALEUR DU CONDENSATEUR ";C;" PF"
210 PRINT
220 INPUT "UN AUTRE CALCUL (O/N) ";B$
230 IF LEFT$(B$,1)="O" THEN GOTO 20
240 END
```

FIGURE 6. – Calcul des paramètres d'un monostable.

Conclusion

Notre prochain numéro sera consacré à quelques autres exemples de programmes qui termineront cette partie consacrée au Basic. Nous pourrons ensuite aborder le langage machine, source d'inquiétude (à tort) pour nombre de programmateurs débutants.

C. TAVERNIER