

SUITE A L'ARTICLE "EXPRESSION EN RÉPONSE A INPUT" (Bulletin E.P.I. n°45, pages 163-1364)

C. GIRARDON

Les nombreux M05 du nanoréseau présentent le même inconvénient corrigé depuis dans le BASIC 128 par les fonctions CRUNCH\$ et EVAL. Celles-ci permettent de saisir dans une variable alphabétique une expression puis de l'évaluer comme si elle était tapée en mode direct au clavier. Plus de problème pour SQR(2), SIN(72), etc. On évite ainsi d'écrire de longues lignes de BASIC qui ralentissent l'exécution des programmes. Les logiciels traçant des courbes peuvent être réduits de moitié.

Je propose donc une solution donnant l'équivalent pour M05.

Le procédé est pratiquement identique à celui utilisé pour le T07/70. Il fait appel à un pointeur et deux routines de la ROM BASIC.

Le pointeur situé en \$21B3 et \$21B4 reçoit l'adresse où les deux routines effectueront leur travail. La fonction USR nous passe dans le registre X l'adresse de la chaîne à traiter. La routine située en \$DE3A se charge de traduire les codes ASCII en codes BASIC, et par extrême gentillesse nous laisse l'adresse de sa traduction dans notre pointeur (moins un octet toutefois).

La seconde routine se lancera alors dans son évaluation et nous placera son résultat dans l'accumulateur flottant (\$214C) dont l'adresse sera déposée par nos soins dans le registre X afin que la fonction USR nous le restitue dans la variable de notre choix.

Le programme est entièrement relogeable.

Programme :

TITLE INPUT.MO5

— *
— *
— *

21X4 CARCOU			EQU	\$21X4	
214C FAC			EQU	\$214C	
-	*				
-	*				
-	*				
8000			ORG	\$8000	
8000 AE	01	DÉPART	LDX	1,X	Adresse chaîne
8002 10BE	21X4		LDY	CARCOU	Sauve caract. cour.
8006 34	20		PSHS	Y	
8008 BD	DE3A		JSR	\$DE3A	Traduction
800B 4F			CLRA		
8000 A7	CO		STA	U+	Zéro de précaution
800E 7C	21X4		INC)CARCOU	Début chaîne à évaluer
8011 BD	21D9	JSR \$21D9			Evaluation
8014 35	20		PULS	Y	
8016 10BF	21X3		STY	\$21X3	
801A 86	04		LDA	44	Renvoi du résultat
801C 8E	214C		LDX	#FAC	par la fonction
801F 39	RTS		USR	0000	END

Mise en place du programme :

```

10 CLEAR ,&H7FFF
20 FOR N=&H8000 TO &H801F:READ C$:POKE N,VAL("&H"+C$):NEXT N
30 END
40 DATA AE,01,10,BE,21,B4,34,20,BD,DE,3A
50 DATA 4F,A7,CO,7C,21,B4,BD,21,D9,35
60 DATA 20,10,BF,21,B3,86,04,8E,21,4C,39

```

Pour illustration, un programme sans prétention :

Simulation d'une calculette :

```

10 CLEAR ,&H7FFF
20 LOADM "INPUT"
C. GIRARDON

```

```
30 DEFUSRO=&H8000
40 CLS:X1=5:Y1=5:ATTRB 1,O:A$="9"
50 LOCATE X1,Y1,O:PRINT A$+CHR$(24)
60 X$=INPUT$(1):GOSUB 1000
70 GOTO 50
900 '
910 ' ANALYSE DES TOUCHES
920 '
1000 IF X$="=" THEN GOSUB 2000:RETURN
1010 IF X$=CHR$(12) THEN GOSUB 3000:RETURN
1015 IF ASC(X$)<&H28 OR ASC(X$)>H39 THEN RETURN
1020 A$=A$+X$
1030 RETURN
1900 '
1910 ' CALCUL
1920 '
2000 A=USRO(A$)
2010 A$=STR$(A)
2020 RETURN
2900 '
2910 ' CLR
2920 '
3000 RUN 30
```

C. GIRARDON
Collège de la Verneuil
MONTIGNY LES CORMEILLES