

DES POIGNE

Les principales applications en micro-informatique familiale se situent incontestablement au niveau ludique.

L'Oric 1, ce micro-ordinateur au prix et aux possibilités intéressants, est pourtant, dans sa version de base, dépourvu de l'interface « reine » en matière de jeu : celle destinée à recevoir des poignées de jeu (« joysticks » pour les Anglo-Saxons). Cette interface, dont la réalisation vous est présentée ici jusque dans les moindres détails, permet à l'ordinateur d'acquérir des informations sur la position d'un « manche à balai » et donc de pouvoir commander le déplacement d'un point sur l'écran. De plus, il vous sera possible, en élaborant toute adéquat, de dépasser le cadre limité des applications ludiques et de doter votre micro-ordinateur d'une « souris », à l'image des micro-systèmes de « haut de gamme ».

sur l'écran.

fois le logiciel

adéquates ludiques

et de doter votre micro-ordinateur d'une « souris », à l'image des micro-systèmes de « haut de gamme ».

D'un point de vue « électrique », une poignée de jeu est assimilable à cinq interrupteurs de type « poussoir » (ouverts au repos) qui viendront se fermer suivant les mouvements appliqués au « manche » (fig. 1a).

Par exemple, si le manche est en haut, alors I_1 sera fermé, tandis que si le manche est en bas à droite, c'est I_3 et I_2 qui seront fermés. Et ainsi de suite...

Si nous considérons que chaque interrupteur a l'une de ses bornes reliée à la masse (0 V) et l'autre au + 5 V via une résistance de 4,7 k Ω , alors une sortie S présentera une tension de + 5 V lorsque l'interrupteur sera ouvert et de 0 V lorsqu'il sera fermé (fig. 1b).

Nous voici avec cinq informations binaires qu'il va falloir introduire dans l'espace mémoire du micro-ordinateur. Il est à noter que ce procédé est utilisé de manière courante pour transformer les deux états d'un interrupteur en information digitale (cas des claviers).

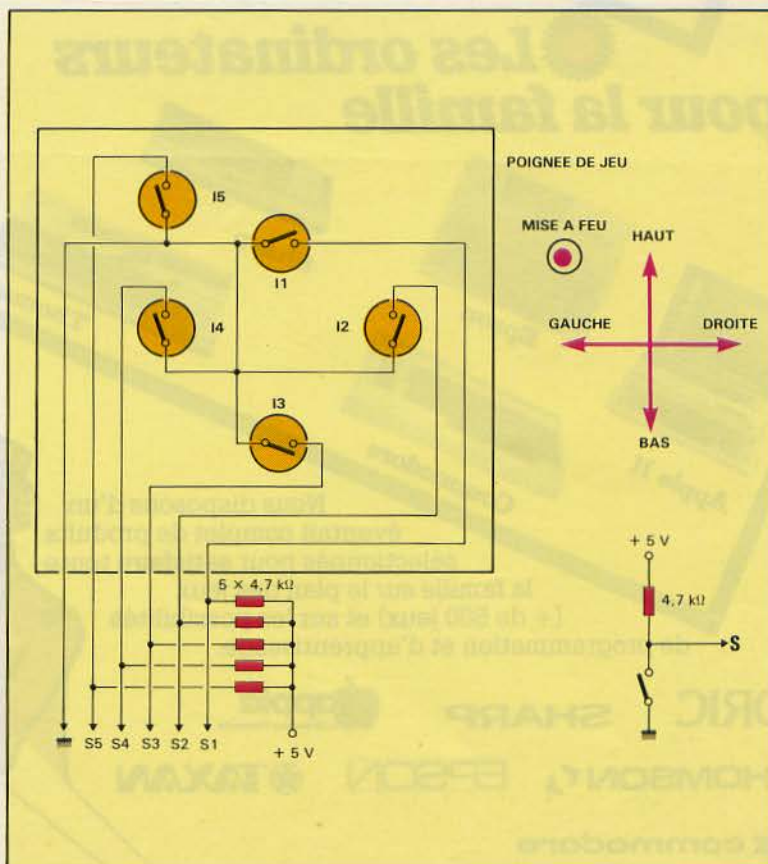


Fig. 1. - a) Une poignée de jeu peut être assimilée à cinq interrupteurs. - b) La sortie S est à + 5 V ou à 0 V selon la position de l'interrupteur.

ES DE JEU POUR ORIC 1

L'interface

Comme vous le savez certainement, le « cœur » de l'Oric 1 est constitué d'un microprocesseur de type 6502. Ce dernier, contrairement au Z 80, ne possède pas de port d'entrées/sorties. Il nous faut donc déterminer un emplacement mémoire disponible afin que le 6502 puisse saisir les cinq informations issues de la poignée de jeu.

A ce niveau, quatre solutions peuvent être envisagées car les 64 Ko adressables par le microprocesseur sont déjà occupés par le micro-ordinateur (48 Ko de mémoire vive et 16 Ko de mémoire figée).

Ces quatre possibilités sont les suivantes :

- Le « D.M.A. » (Direct Memory Access), c'est-à-dire ranger nos cinq bits dans un emplacement peu utilisé (en fin de R.A.M., par exemple). Cette solution, bien que séduisante, est onéreuse et délicate à mettre en œuvre pour un amateur. Laissons-la donc aux lecteurs de disquettes qui l'utilisent communément...
- Utiliser le port de l'imprimante, à condition, bien sûr, de modifier certains pointeurs, de réécrire une routine en langage machine, et... de ne pas se servir de l'imprimante. C'est une solution certes économique mais peu pratique : il faudrait écrire un programme d'utilisation et « poker » des octets çà et là...
- Trouver un moyen d'inhiber l'accès d'un octet de RAM afin d'y « glisser » notre interface. Dans ce cas, c'est simple et peu coûteux... Encore faut-il disposer d'une sortie du genre RAMDIS (RAM Disconnect) sur le connecteur de sortie...
- Essayer la même solution que précédemment, mais avec la ROM. Chacun sait que la ROM contient le programme moniteur de tout système et que supprimer

un octet est une opération prosaïque. A moins que nous puissions trouver dans cette ROM des octets ne servant à rien (quelques octets contenant, par exemple, un copyright).

Essayez donc le programme suivant :

```
10 FOR X = 60301 TO 60340
20 PRINT CHR$(PEEK(X)) ;
30 NEXT X
```

et vous verrez apparaître le message suivant :

« Software by Peter Halford and Andy Brown ».

Voilà 40 octets de ROM que nous pouvons utiliser pour nos interfaces. Cette fois-ci, le signal ROMDIS est bien disponible sur le connecteur de sortie (broche n° 2). Il nous faut maintenant choisir une adresse entre 60301 et 60340 puis la décoder.

Décodage d'un emplacement mémoire

Décoder un emplacement mémoire consiste à réaliser une fonction logique qui ne répond qu'à une seule condition, en l'occurrence l'apparition de l'adresse de cet emplacement sur le bus d'adresses.

Pour des raisons de commodité, nous avons choisi l'adresse 60319 (soit EB9F en hexadécimal). Pour connaître l'état du bus lors de l'apparition de cette adresse, il suffit de convertir 60319 en binaire, ce qui donne : 1110101110011111

Cela signifie que la ligne d'adresse n° 15 (A15) sera à 1 (+ 5 V), que A12 sera à l'état 0 (0 V), etc.

En examinant cette adresse (60319), nous nous apercevons que seulement quatre lignes (A12, A10, A6 et A5) sont à l'état bas. Il suffit donc d'inverser leur état pour n'avoir, sur les 16 lignes, que des états hauts (+ 5 V), que nous pouvons décoder avec deux circuits de type 74 LS 30 (fig. 2).

Le 74 LS 30 est une porte NON ET avec 8 entrées. La sortie de cette porte ne passera à 0 que si toutes les entrées sont à l'état 1.

Nous avons ainsi converti la présence de notre adresse sur le bus en deux états bas à la sortie des deux circuits 74 LS 30. Il reste à « réunir » ces deux états, ce qui est réalisé par une porte OU.



Notre interface est prête à recevoir une poignée de jeu au standard Commodore ou Atari.

Réalisation

Voilà notre décodage réalisé. L'apparition de l'adresse 60319 se concrétisera par la présence d'un état bas à la sortie de la porte OU. Celui-ci va servir à commander le circuit d'accès au bus de données et sera le « nerf » de notre interface.

Maintenant que notre adresse est décodée, il faut désactiver la ROM de l'Oric afin de libérer notre espace mémoire.

Désactivation de la ROM

L'opération de désactivation est simplement réalisée en appliquant le signal de décodage à la broche n° 2 (ROM DIS) du connecteur d'extension de l'Oric via un tampon constitué par une porte OU.

Le simple fait d'appliquer un état bas sur ROM DIS désactive la ROM de l'Oric et libère notre espace mémoire.

Presque tout est prêt pour avoir accès au bus de données. Mais il faut savoir que ce dernier est bidirectionnel, c'est-à-dire qu'il véhicule des données du microprocesseur vers les périphériques (mémoire, interface...) et vice versa.

Le microprocesseur indique le sens de circulation du bus de données par l'intermédiaire d'un signal baptisé Read/Write (lecture/écriture). Dans le cas de l'Oric (broche n° 7), cette sortie est à l'état haut lorsque le 6502 lit des données, et est à l'état bas lorsqu'il écrit des données.

En ce qui nous concerne, nous devons faire lire au 6502 les informations de notre poignée de jeu. C'est pourquoi l'accès au bus de données ne doit s'effectuer que lors d'une phase de lecture du microprocesseur (encore une nouvelle condition). Ce nouvel impératif se réalise simplement en inversant l'état du signal R/W, et en faisant un



Votre réalisation peut être « mise à l'abri » dans un boîtier de type Teko P2 (ou MMP 20).

« OU » avec le signal de décodage. Il ne nous reste plus qu'à synchroniser ce dernier signal avec l'horloge qui cadence toute action du microprocesseur.

Une donnée ne doit se trouver sur le bus que si l'horloge est dans sa phase active ($\Phi_2 = 1$). Même démarche que pour le signal R/W : l'état du signal d'horloge Φ_2 est inversé et nous effectuons encore un « OU » avec le signal précédent.

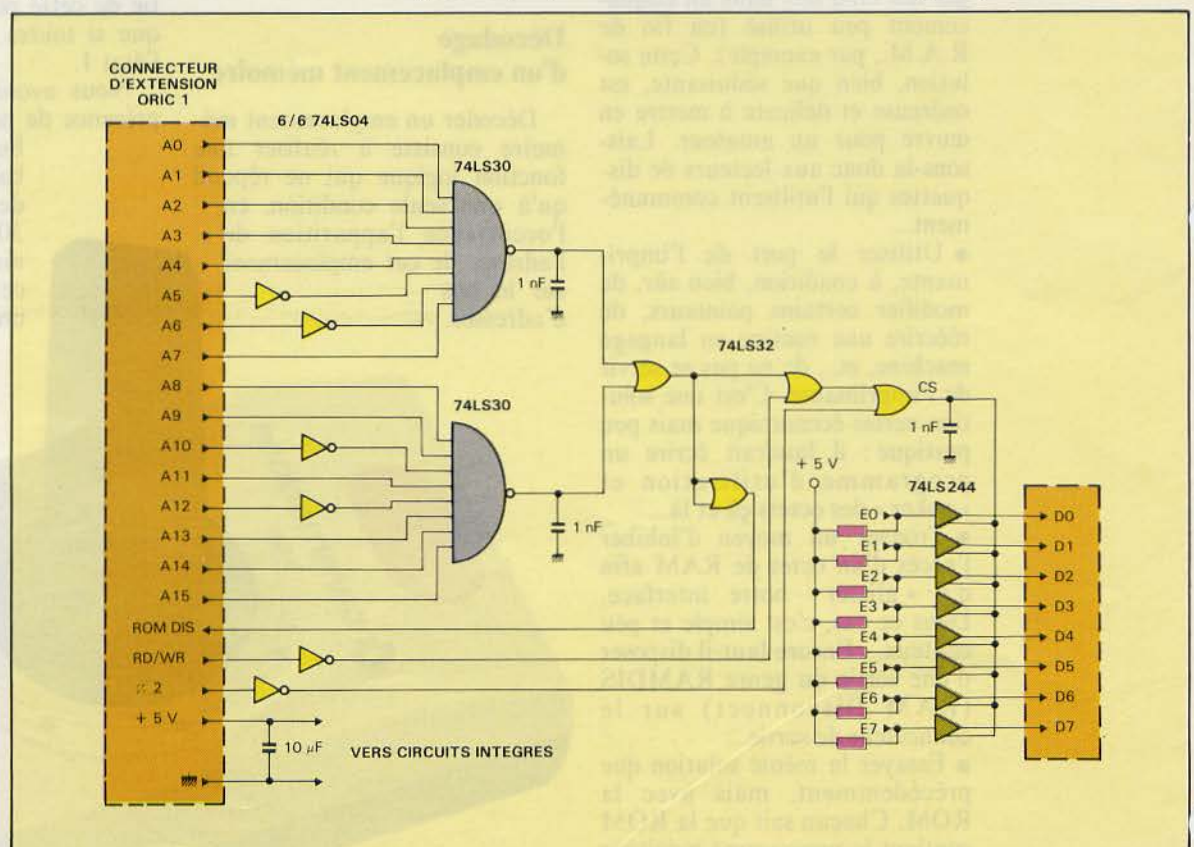


Fig. 2. - Schéma électrique complet du module d'interface.

Maintenant, nous avons élaboré notre signal de demande d'accès au bus de données (noté CS sur la **figure 2**) !

En résumé, CS sera actif (donc à 0) uniquement lorsque les trois conditions suivantes seront vérifiées :

- Bus d'adresse = 60319
- R/\overline{W} (broche n° 7) = 1
- Φ_2 (broche n° 3) = 1.

L'accès au bus de données s'effectuera grâce à un circuit intégré de type 74LS244. Celui-ci contient huit tampons dont les sorties peuvent se positionner en un état « haute impédance », n'émettant, de ce fait, aucun signal sur le bus de données.

La commande de l'état actif à l'état « haute impédance » a lieu par le signal CS.

Les huit entrées sont amenées à un potentiel de + 5 V par l'intermédiaire de résistances de 4,7 k Ω et sont ainsi prêtes à recevoir la poignée de jeu.

Une fois cette interface reliée à l'Oric, il suffira de taper PRINT PEEK (60319) pour connaître l'état électrique des huit entrées.

Réalisation pratique

L'ensemble des composants nécessaires à cette réalisation est résumé **tableau 1**.

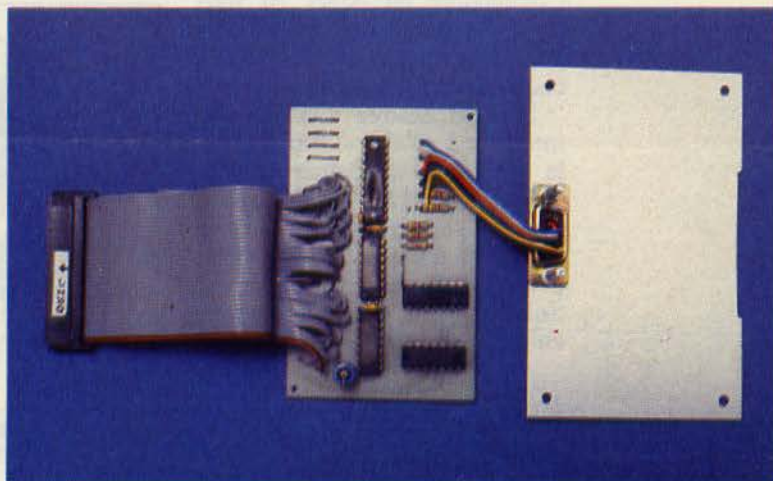
Le tracé du circuit imprimé est donné **figure 3** et le schéma d'implantation **figure 4**. Le câblage ne pose pas de problèmes à condition de commencer par les « straps » (simples fils électriques assurant les liaisons), puis les circuits intégrés (attention à leur orientation), les résistances et les condensateurs (ne pas oublier C_3 entre la broche n° 1 du 74LS244 et C_2).

Les résistances R_1 à R_5 seront légèrement surélevées afin de souder sur leurs pattes les fils de liaison du connecteur « D sub ».

Le seul point délicat est le câblage du connecteur reliant l'interface à l'Oric. La meilleure solution est d'acheter le connecteur de 2 x 17 broches

Références	Quantités	Type
Circuits intégrés		
I_1	1	74LS244
I_2, I_3	2	74LS30
I_4	1	74LS04
I_5	1	74LS32
Condensateurs		
C_1, C_2, C_3	3	1 nF (céramique)
C_4	1	10 μ F / 25 V
Résistances		
R_1 à R_8	8	4,7 k Ω 1/4 W
Divers		
<ul style="list-style-type: none"> • 1 connecteur 2 x 17 broches (2,54 x 2,54) femelle à sertir suivant la norme HE 10. • 1 connecteur D.Sub châssis, 9 broches mâle à souder. • 20 cm de câble 34 conducteurs à sertir. • 1 boîtier Teko ou MMP (facultatif). 		

Tableau 1. - Nomenclature du matériel utilisé.



L'interface et ses deux connecteurs assurant la liaison avec le micro-ordinateur et une poignée de jeu.

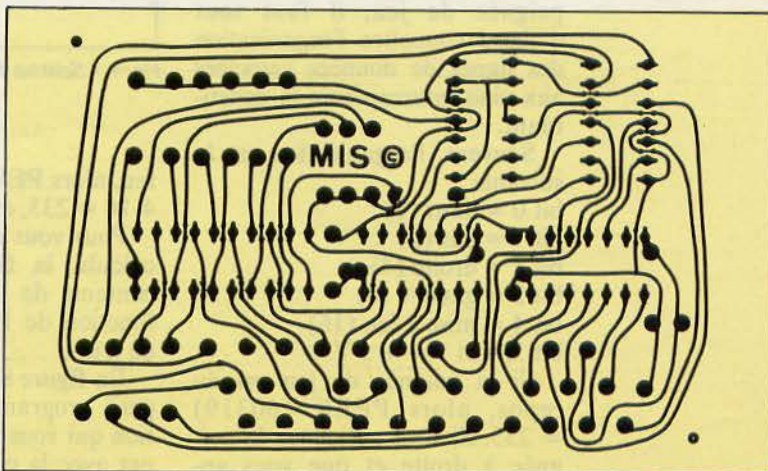


Fig. 3. - Circuit imprimé vu côté « soudures » (Ech. 1).

D'un point de vue électrique, une poignée de jeu est assimilable à cinq interrupteurs.

Réalisation

(voir **tableau 1**) dans un magasin qui vous le sertira avec 20 cm de câble spécial.

Ensuite, séparer les conducteurs, les dénuder et les étamer. Ce connecteur n'ayant pas de détrompeur, il va falloir repérer le fil n° 1 (broche n° 1 du connecteur de l'Oric).

Branchez le connecteur dans l'Oric. Le fil n° 1 sera celui situé à votre droite, c'est-à-dire celui qui est le plus près du connecteur pour l'imprimante. A ce moment, il ne vous reste plus qu'à câbler les 34 fils suivant l'ordre indiqué. Cette opération terminée, câblez les 6 fils au connecteur « D Sub » suivant le brochage indiqué **figure 5**. Ce connecteur est prêt à recevoir une poignée de jeu au standard Commodore ou Atari. Si vous disposez d'une autre poignée de jeu, il faudra modifier le brochage de ce connecteur.

Si vous désirez mettre votre interface à « l'abri », vous pouvez la loger dans un boîtier de type Teko P2 ou MMP 20. Le circuit imprimé se loge dans le fond et il n'y a qu'à découper la face avant pour y fixer le connecteur « D Sub » et faire ressortir le câble en nappe (**fig. 6**).

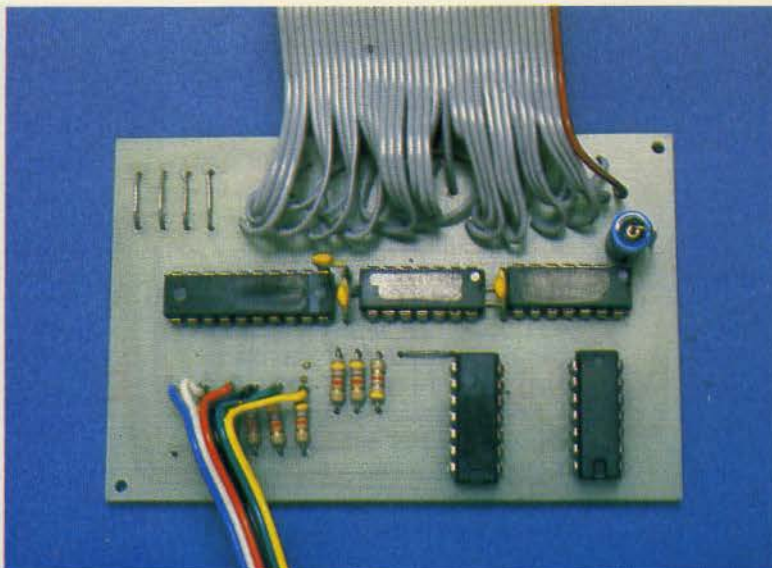
Un programme de démonstration

Afin de pouvoir écrire un programme mettant en œuvre une poignée de jeu, il faut tout d'abord connaître l'organisation des lignes de données associées aux cinq interrupteurs la constituant.

Souvent, l'organisation est la suivante :

- bit 0 = haut (1)
- bit 1 = bas (2)
- bit 2 = droite (4)
- bit 3 = gauche (8)
- bit 4 = mise à feu (16)
- bit 5, 6 et 7 = + 5 V.

Si la poignée de jeu est au repos, alors PEEK (60319) = 255. Si vous actionnez la poignée à droite et que vous appuyez sur le bouton de mise à



La carte électronique constituant l'interface.

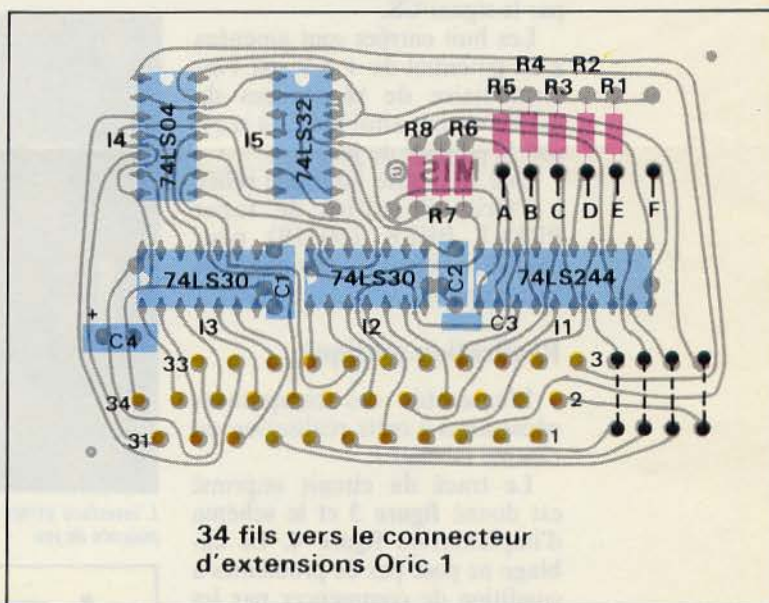


Fig. 4. - Schéma d'implantation des composants (Ech. 1).

feu, alors PEEK (60319) = 255-4-16 = 235, etc.

Pour vous éviter de fastidieux calculs, la **figure 7** résume le contenu de l'adresse 60319 en fonction de l'état de la poignée de jeu.

En **figure 8**, vous trouverez un petit programme de démonstration qui vous permettra de dessiner avec la poignée de jeu. Pour changer la couleur, il vous suf-

fira d'appuyer sur le bouton de « mise à feu ».

Ça y est, vous pouvez laisser les enfants jouer, ils ne martyriseront plus votre clavier. Et puis, il y a encore beaucoup d'autres possibilités. Ce petit logiciel n'est là que pour vous mettre « l'eau à la bouche » ou plutôt les doigts sur le clavier... ■

J.-P. PINTÉ

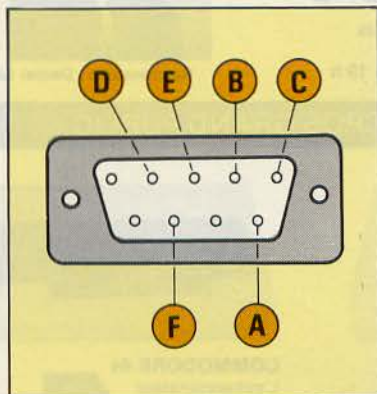


Fig. 5. - Le connecteur recevant la poignée de jeu, vu de dessous.

```

5 HIRES : INK 2 : PAPER 0 :
  X = 100 : Y = 90
10 A = PEEK (# EB9F)
15 IF A=25 THEN CURSET X,Y,0
20 IF A=254 THEN Y=Y-1
25 IF A=253 THEN Y=Y+1
30 IF A=251 THEN X=X+1
35 IF A=247 THEN X=X-1
40 IF A=246 THEN X=X-1:Y=Y-1
45 IF A=250 THEN X=X+1:Y=Y-1
50 IF A=249 THEN X=X+1:Y=Y+1
55 IF A=245 THEN X=X-1:Y=Y+1
60 IF A=239 THEN ZAP:INK (INT
  (RND(1)*6)+1)
65 CURSET X,Y,1:GOTO10
  
```

Fig. 8. - Un petit programme pour dessiner avec la poignée de jeu.

Cette réalisation vous intéresse ?

Si vous désirez vous procurer l'ensemble des composants nécessaires à la réalisation de cette interface (circuits intégrés, condensateurs, résistances ou matériels divers), écrivez-nous.

Nous regrouperons les commandes pour vous faire bénéficier de tarifs préférentiels.

MICRO-SYSTEMES
43, rue de Dunkerque
75010 PARIS



Les résistances R₁ à R₅ sont légèrement surélevées.

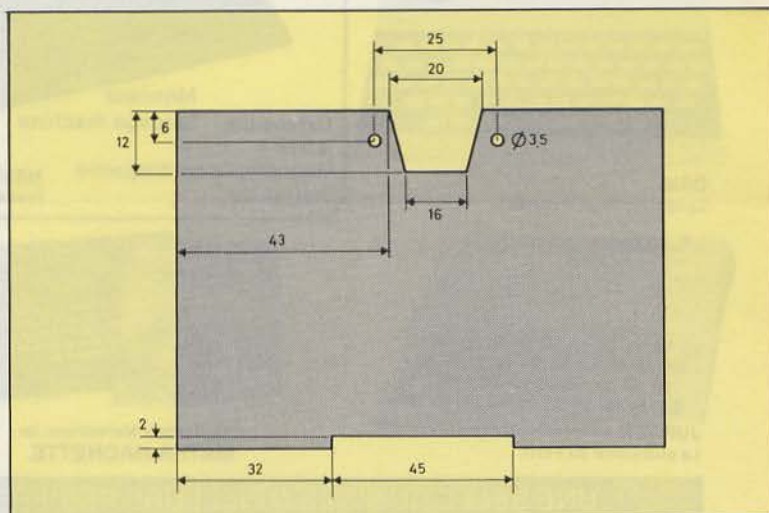


Fig. 6. - Les « cotes » d'un petit boîtier vous permettant de mettre votre réalisation « à l'abri ».

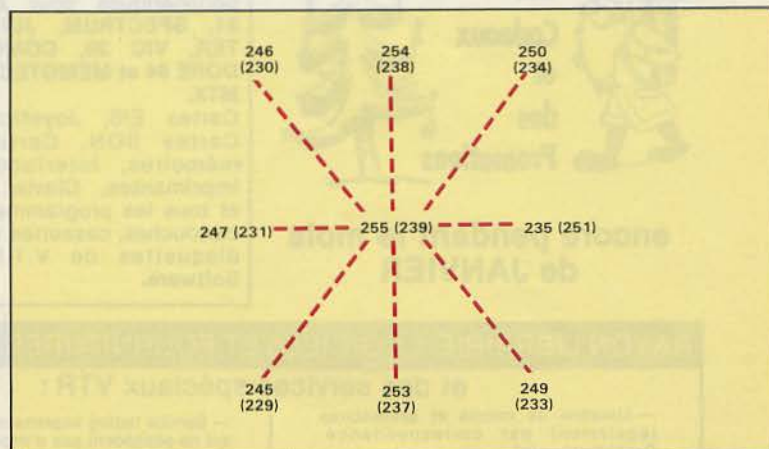


Fig. 7. - Valeurs contenues à l'adresse 60319 en fonction de la position du « manche à balai ». Les valeurs entre parenthèses sont à prendre en compte lorsque le bouton de « mise à feu » est enfoncé.