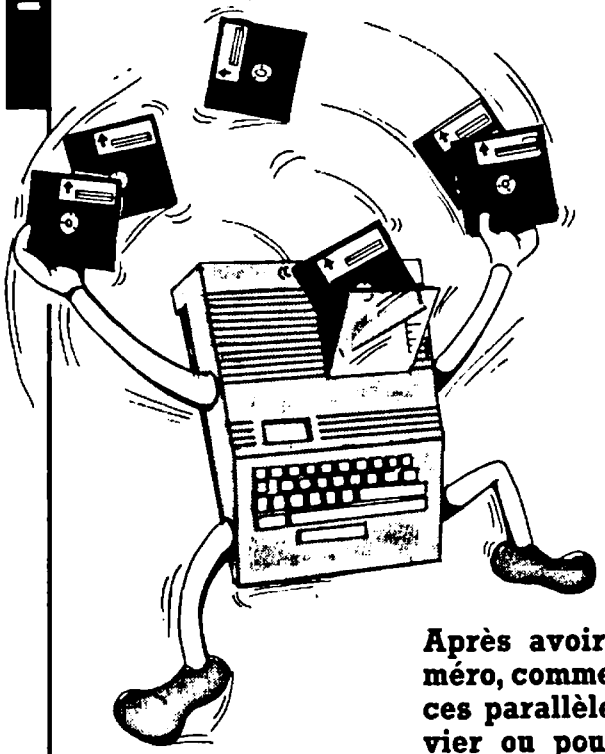
L'ABC DE LA MICRO-INFORMATIQUE



LES INTERFACES PARALLELES

Après avoir vu, dans notre précédent numéro, comment utiliser des circuits d'interfaces parallèles pour lire des touches de clavier ou pour commander des relais, nous allons aujourd'hui examiner des fonctions un peu plus « nobles » avec la commande d'afficheurs.

LES AFFICHEURS 7 SEGMENTS

Même si cette application n'est pas celle qui vous vient immédiatement à l'esprit lorsque l'on vous parle de circuit d'interface parallèle, c'est pourtant l'une des utilisations majeure de ces composants dans de très nombreux montages. En effet, hormis dans les micro-ordinateurs où le dialogue avec l'utilisateur s'établit généralement au moyen d'un écran, dans de très nombreuses applications, ce sont de simples afficheurs numériques ou alphanumériques qui sont chargés de cette fonction. Il vous suffit de regarder au rayon bricolage de n'importe quel grand magasin pour voir des thermostats programmables, des programmateurs ménagers, des arrosages automatiques ; tous ceux qui sont à base de microprocesseurs utilisent un affichage numérique ou alphanumérique, sous forme d'afficheurs à LED ou à cristaux liquides. De la même façon que nous avons vu comment utiliser un clavier le mois dernier, nous allons étudier maintenant comment piloter des afficheurs, que nous avons choisis du type à LED 7 segments pour simplifier un peu notre exposé.

Rappelons que de tels afficheurs ont tous la structure présentée figure 1. Chaque segment est en fait une (ou plusieurs) diode électroluminescente qui s'éclaire lorsqu'elle est correctement alimentée et polarisée. Toutes les anodes de ces diodes sont reliées entre elles dans un afficheur à anodes communes, alors que ce sont toutes les cathodes dans un afficheur à cathodes communes. Les appellations des segments, de a à f, sont normalisées, ainsi que celles du point décimal DP, ce qui simplifie grandement le travail des concepteurs de schémas. Dernier

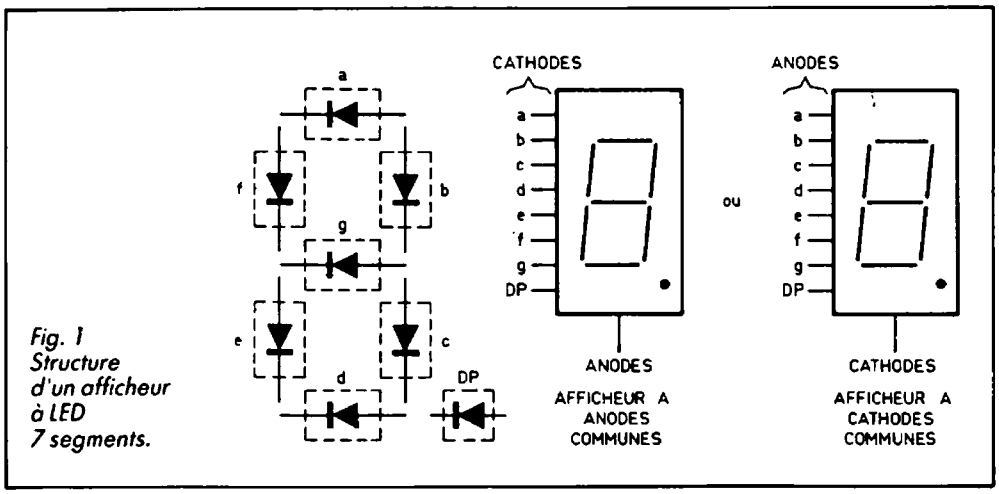


Fig. 1 Structure d'un afficheur à LED 7 segments.

point important à préciser, pour obtenir une luminosité correcte de l'afficheur : il faut faire passer au moins 10 mA par segment en continu.

LA METHODE DIRECTE

Lorsque l'on doit commander seulement un ou deux afficheurs et que l'on n'est pas trop limité au niveau du nombre de lignes d'entrées/sorties parallèles disponibles, on peut utiliser la méthode directe schématisée figure 2.

Comme vous pouvez le constater, on commande chaque segment de l'afficheur (ou des afficheurs s'il y en a plusieurs) avec une ligne d'un port parallèle, après amplification de courant ; sauf sur de rares circuits, ces derniers ne peuvent bien souvent pas fournir les 10 mA nécessaires. Il est évident que le logiciel de commande d'une telle configuration est fort simple, puisqu'il suffit de présenter sur les lignes du port concerné le code binaire correspondant directement aux segments à allumer. Dans notre exemple, un 1 logique provoque l'allumage du segment et un 0 son extinction.

Remarquez que, avec cette façon de faire et contrairement à la solution des décodeurs 7 segments traditionnels (7447 en TTL ou 14511 en CMOS par exemple), on peut faire afficher « n'importe quoi » à l'afficheur et non pas seulement la suite de chiffres de 0 à 9. C'est très utile dans certaines applications telles que celles évoquées dans le précédent paragraphe.

Pour simple et efficace qu'elle soit, cette méthode est cependant limitée à la commande de un ou deux afficheurs au maximum ; au-delà, la « consommation » en lignes d'entrées/sorties devient prohibitive, et plutôt que de multiplier les circuits d'interfaces parallèles, on préfère faire appel à la technique de l'affichage multiplexé. Ce choix est d'autant plus judicieux que l'on peut faire du multiplexage par logiciel, ce qui dispense d'utiliser les boîtiers spécialisés habituellement rencontrés avec les affichages de ce type.

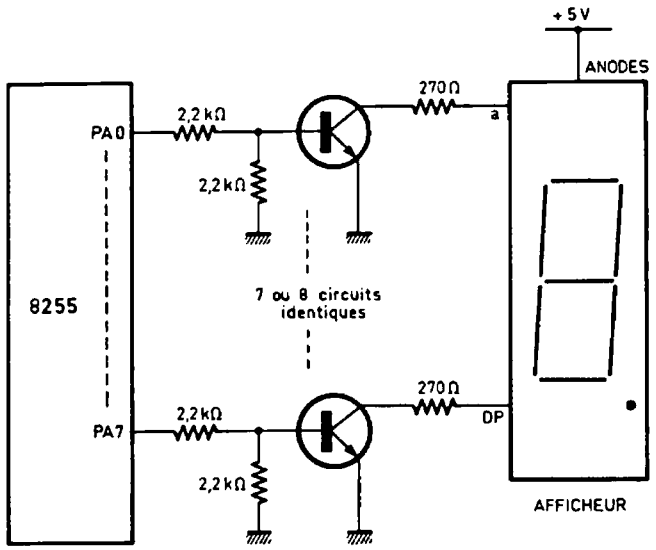


Fig. 2 Commande d'un afficheur par un circuit d'interface parallèle, méthode « classique ».

L'AFFICHAGE MULTIPLEXE

Un affichage de ce type utilise une propriété bien connue de l'œil humain : la persistance des impressions rétiniennes. Cette propriété fait que, pour que l'œil ait une sensation de vision continue, il n'est pas nécessaire de lui présenter des images permanentes, mais il suffit de lui renouveler celles-ci avec une fréquence suffisante. Pour éviter tout effet de papillotement, le renouvellement doit intervenir au maximum toutes les 40 ms. On peut renouveler plus souvent, mais c'est inutile car cela n'apporte rien. En revanche, le fait de

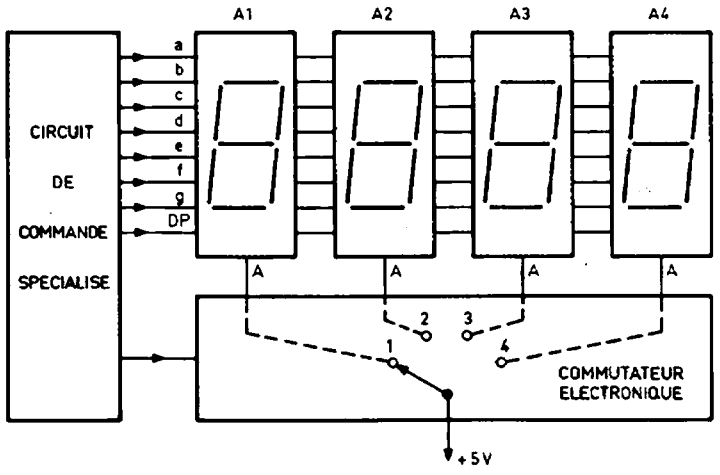


Fig. 3. – Principe d'un affichage multiplexé.

renouveler moins souvent conduit à un papillotement qui va en s'amplifiant avec le ralentissement du renouvellement.

Cela étant précisé, examinez la figure 3 qui représente le schéma général d'un affichage multiplexé. Nous avons représenté quatre afficheurs, mais le schéma et le principe sont extensibles à un nombre quelconque d'afficheurs. Le fonctionnement est le suivant.

A l'instant T_1, le commutateur électronique est en position 1, l'afficheur 1 est donc alimenté ; simultanément, le circuit de commande présente sur ses pattes a à f et DP le code du symbole à afficher, qui devient donc visible sur cet afficheur et sur celui-là seulement, les autres n'étant pas alimentés.

A l'instant T_2, le commutateur électronique passe sur 2, alimentant ainsi l'afficheur 2. Simultanément, le circuit de commande délivre sur ses sorties a à f et DP le code du symbole à afficher sur l'afficheur 2 qui est le seul à s'allumer puisque c'est le seul à être alimenté.

Le processus se poursuit ainsi jusqu'à l'afficheur 4, pour recommencer ensuite à partir du premier.

Si la vitesse de commutation est suffisante pour que chaque afficheur se trouve allumé au moins une fois toutes les 40 ms, l'œil de l'observateur aura l'impression que tous les afficheurs sont allumés en permanence, et le tour sera joué. Bien sûr, plus il y a d'afficheurs à commander, plus il faut « que ça tourne vite » afin de satisfaire cette contrainte physiologique de l'œil. Cela explique pourquoi, sur certains circuits intégrés de voltmètres numériques ou de fréquencemètres, on trouve des fréquences de multiplexage de 10 à 100 kHz et au-delà.

La réalisation matérielle d'un tel affichage nécessite de nombreux composants lorsque l'on souhaite la faire soi-même, ou alors elle passe par l'emploi de circuits intégrés à grande échelle comportant en interne tous les éléments nécessaires. Nous allons voir qu'avec notre circuit d'interface parallèle et un peu de logiciel, c'est un jeu d'enfant.

La figure 4 vous présente le schéma typique à utiliser, schéma dessiné ici pour des afficheurs à anodes communes, mais transposable sans difficulté pour des afficheurs à cathodes communes. Pour rester cohérent avec l'exemple précédent, nous avons choisi un affichage à 4 chiffres, particulièrement bien adapté à notre 8255 puisque les 8 lignes du port A commandent les segments, alors que 4 lignes du port C, placées en sorties, commandent les anodes, *via* des transistors amplificateurs bien sûr.

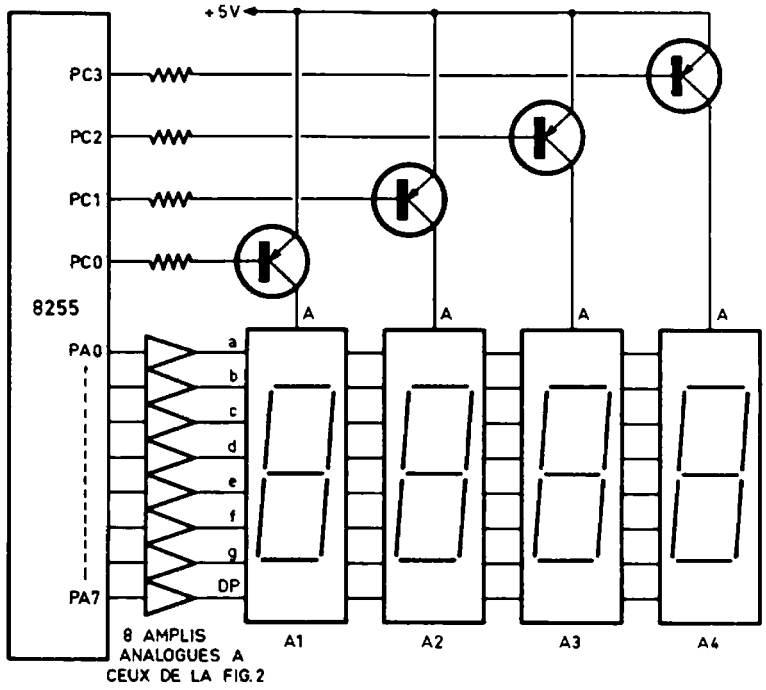


Fig. 4. – Réalisation d'un affichage multiplexé avec un 8255.

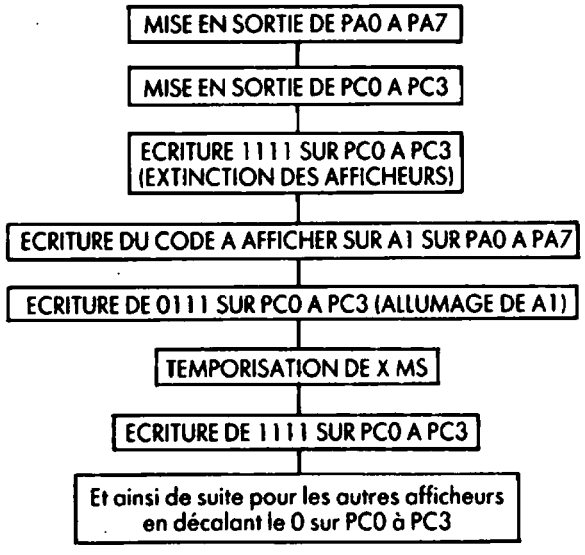


Fig. 5. – Organigramme de commande du circuit d'affichage de la figure 4.

Aucune circuiterie spéciale n'est nécessaire, car tout le travail est fait par le logiciel dont un organigramme vous est proposé figure 5. Nous allons le commenter rapidement, et vous pourrez constater qu'il suit exactement le principe de fonctionnement du montage traditionnel décrit ci-avant.

Le programme commence par placer PA0 à PA7 en sorties ainsi que PC0 à PC3. Il place ensuite sur PA0 à PA7 le code du symbole à afficher sur A1. Lorsque c'est fait, l'écriture de 0111 sur PC0 à PC3 permet à l'afficheur A_1 d'être alimenté et, donc, d'afficher quelque chose. Après une phase de temporisation, 1111 est envoyé sur PC0 à PC3, et le code du symbole à afficher sur A2 peut alors être placé sur PA0 à PA7. La combinaison 1011 est ensuite envoyée sur PC0 à PC3 pour alimenter A2.

Ce processus se répète ensuite indéfiniment, balayant ainsi à tour de rôle les quatre afficheurs. Bien sûr, pour que cela fonctionne correctement, il faut que la commutation soit suffisamment rapide, comme nous l'avons expliqué ci-avant, ce qui exclut de pouvoir programmer le 8255 à partir d'un langage évolué (Basic ou Pascal par exemple) ; il faut impérativement écrire en assembleur le morceau de programme correspondant à l'organigramme de la figure 5. Quel que soit le microprocesseur utilisé, ce n'est cependant ni long ni difficile, car les opérations réalisées ne sont que des suites d'écriture dans le 8255.

Par rapport à un affichage classique tel celui présenté au deuxième paragraphe, l'affichage multiplexé est très avantageux vis-à-vis du nombre de lignes utilisées sur le circuit d'interface parallèle ; en effet, alors que l'on ne peut piloter que trois chiffres avec un 8255 par la méthode classique (3 ports de 8 bits), on peut théoriquement commander 16 chiffres par la méthode multiplexée (1 port de 8 bits pour les codes des symboles et 2 ports de 8 bits pour commander les alimentations des afficheurs). Mais – car il y a un mais – un problème auquel vous n'avez peut-être pas pensé risque très vite de se poser...

OU IL EST QUESTION DE SIMULTANEITE

Tout ce que nous venons d'exposer quant à l'affichage multiplexé est parfaitement correct mais pèche néanmoins sur un point. Nous avons dit en ef-

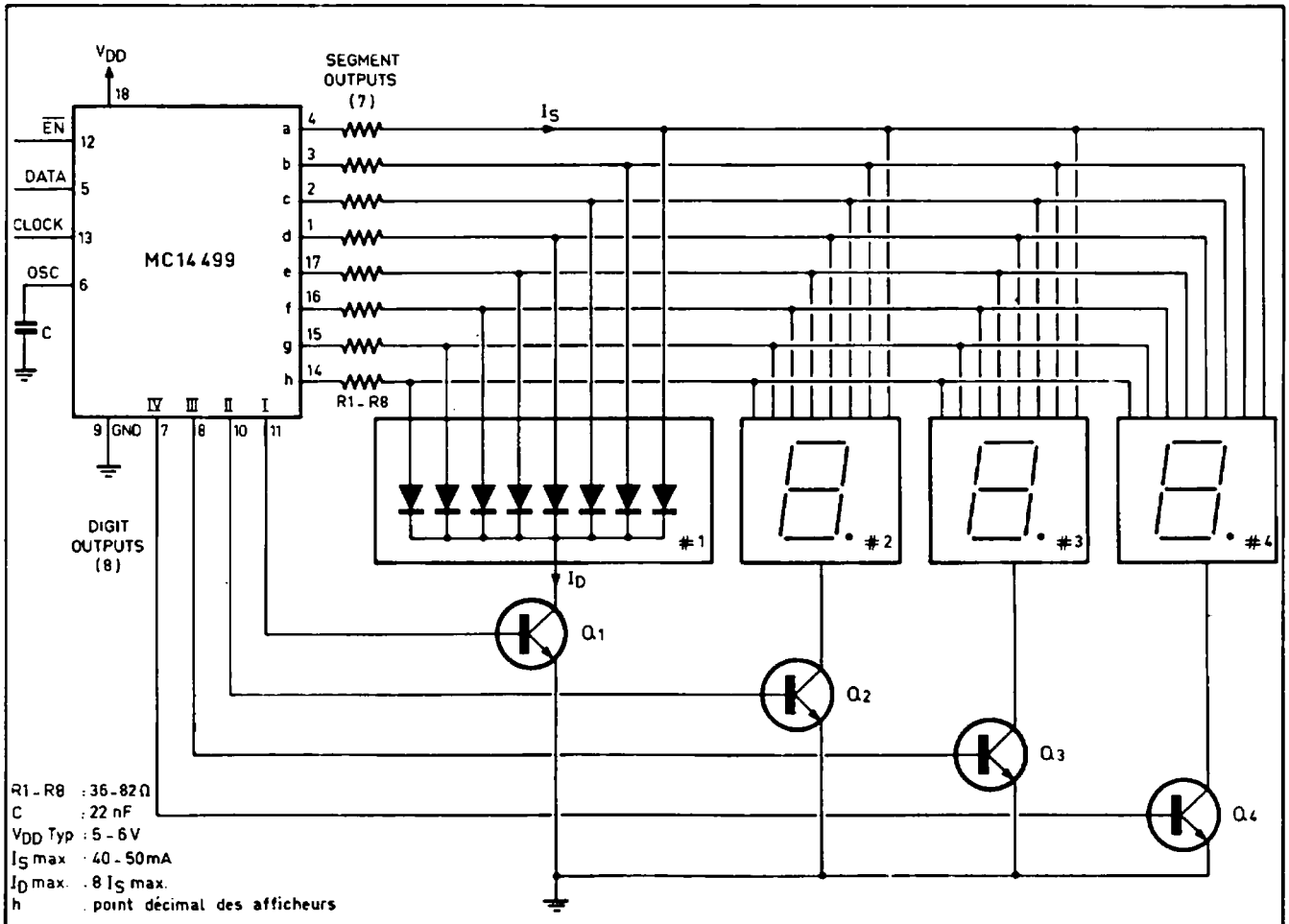


Fig. 6. – Schéma d'utilisation typique du MC14499.

fet, lors des commentaires de l'organigramme de la figure 5, qu'il fallait que notre programme tourne en permanence pour obtenir un affichage correct. C'est très bien, mais le microprocesseur qui commande le 8255 ne peut plus faire autre chose si tel est le cas, puisqu'il passe son temps à fournir des données à ce circuit ; il devient donc inutilisable pour l'application prévue, et nous sommes bien loin du but recherché.

Pour se sortir de ce mauvais pas, il faut écrire le programme général de l'application de façon à ce que le microprocesseur travaille pendant les phases de temporisations visibles sur l'organigramme de la figure 5 ; en effet, pendant ces phases, le 8255 se borne à maintenir à des niveaux constants l'état de ses sorties et le microprocesseur peut donc faire tout autre chose. Bien sûr, cela complique un peu le travail, mais c'est tout à fait possible, et cela fonctionne très bien sous réserve que le travail à effectuer par ce dernier ne soit ni trop long ni trop complexe.

Une autre solution consiste à écrire le programme d'affichage comme un programme d'interruption et à utiliser une horloge ou un timer qui, toutes les X ms, va venir déclencher une interruption sur le microprocesseur. Ce dernier commandera alors l'afficheur périodiquement toutes les X ms et, pour peu que X soit assez faible, l'utilisateur humain placé devant les afficheurs n'y verra que du feu.

Dans certains cas, lorsque le microprocesseur a beaucoup de tâches à accomplir ou que les interruptions sont utilisées pour autre chose, notre affichage multiplexé logiciel n'est plus utilisable et il faut faire appel à d'autres techniques. Nous allons voir ci-après une des plus répandues, surtout en raison de l'avènement des afficheurs à cristaux liquides.

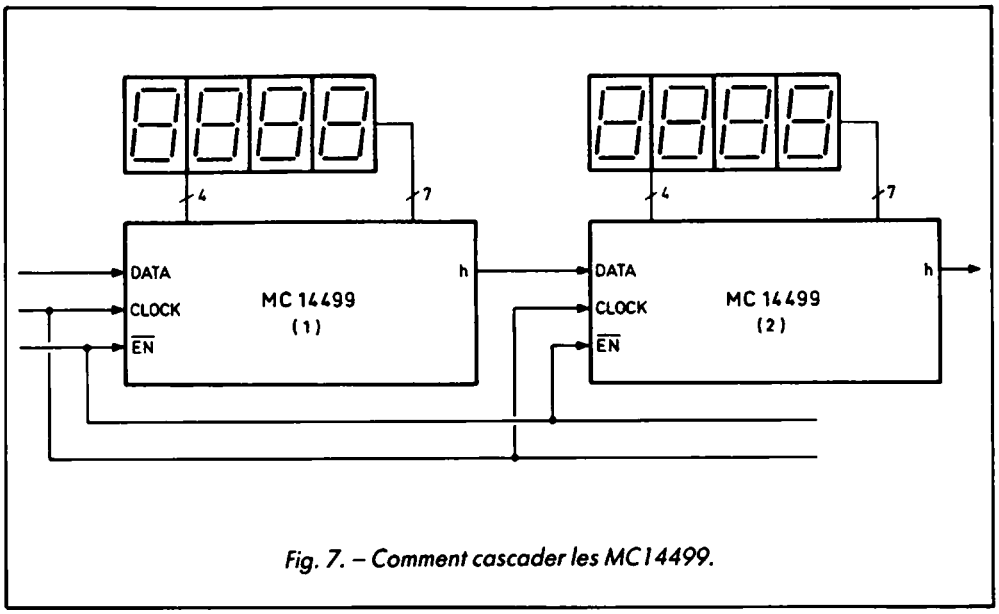


Fig. 7. – Comment cascader les MC14499.

DES INTERFACES POUR INTERFACE...

Les circuits dont nous allons parler maintenant peuvent, théoriquement, être utilisés directement sur le bus de tout microprocesseur, mais, vu la logique nécessaire pour ce faire et compte tenu du faible prix des circuits d'interfaces parallèles du type 8255 par exemple, ils sont généralement connectés « derrière » ces derniers, d'où le titre donné à ce paragraphe.

Comme nous l'avons expliqué ci-avant, ces circuits servent à soulager le logiciel de commande des afficheurs lorsque le microprocesseur est trop occupé. Ce sont donc des circuits « intelligents », capables de maintenir tous seuls un affichage déterminé une fois qu'on leur a fourni les données nécessaires.

Nous avons choisi comme exemple le MC14499 de Motorola, prévu pour piloter des afficheurs à LED, mais sachez qu'il existe des circuits en tous points analogues pour le pilotage d'afficheurs à cristaux liquides (le MC145000, toujours chez Motorola, par exemple). Si vous êtes un fidèle lecteur du *Haut-Parleur*, vous reconnaîtrez dans les schémas que nous allons présenter maintenant des analogies avec nos récents articles consacrés à une centrale de contrôle domestique où nous préconisions l'utilisation de ces circuits.

Comme le montre la figure 6, le MC14499 permet de commander 4 afficheurs par la technique du multiplexage mais en ne consommant que trois fils de notre port de sortie ; fils qui sont reliés aux pattes EN barre, DATA et CLOCK du MC14499. Le principe de ce circuit est le suivant. On le charge avec les données à afficher en respectant un chronogramme particulier sur DATA et CLOCK. Ensuite, il est inutile de s'en occuper ; il procède seul au multiplexage des afficheurs sur lesquels on peut lire en permanence les données fournies au 14499. Le microprocesseur associé au 8255 peut donc faire tout autre chose. Il n'a à s'occuper de l'affichage que lorsque les données changent, et, dans ce cas, son rôle se limite à fournir les nouvelles données, *via* le 8255.

Pour accroître encore la souplesse d'emploi des MC14499, le fabricant a prévu que l'on puisse les monter en cascade, comme schématisé figure 7. Il devient alors possible de commander un nombre quelconque d'afficheurs avec toujours trois lignes de sorties d'un port parallèle.

Afin de ne pas déborder du cadre de cette série, nous en resterons là quant à ces circuits. Si vous désirez en savoir plus à leur sujet, en particulier si vous voulez connaître leurs chronogrammes de fonctionnement, nous vous renvoyons à l'article précité publié dans notre numéro de septembre 1987, page 142 et suivantes.

CONCLUSION

Nous avons gardé le meilleur pour la fin avec la gestion des liaisons parallèles Centronics utilisées sur l'immense majorité des imprimantes que l'on rencontre en micro-informatique. L'ampleur du sujet nécessite un article entier que vous découvrirez donc le mois prochain.

(à suivre)

C. TAVERNIER