

**un an après,
le second souffle...**

La comparaison entre l'homme et l'ordinateur peut faire l'objet de discussions interminables, certes, mais elle présente l'avantage indéniable de l'éloquence et de la clarté: le matériel (hardware) remplirait les fonctions de nos organes; la chair et le sang en somme. Et le logiciel (software), celles de l'intelligence "immatérielle". Arrivé à un certain âge, le corps n'évolue plus (ou peu!), mais l'intelligence, elle, continue d'évoluer (c'est du moins ce qu'il faut espérer!). Le Junior Computer a atteint ce stade. Le matériel est au point, et en mesure de recevoir un logiciel (toujours susceptible d'évoluer). Ceci dit, le matériel n'est pas figé pour

de ce dernier, chaque sous-ensemble est monté sur une carte à part. Celles-ci sont toutes de même dimension (100 x 160 mm pour les cartes au format européen), et sont enfichées sur une carte de bus qui les relie les unes aux autres par le jeu de pistes cuivrées reliant les différents connecteurs. Sans oublier la possibilité de relier les différentes parties par du câble en nappe. En principe la longueur de la carte de bus, c'est-à-dire le nombre des sous-ensembles qu'elle est capable de recevoir, est illimitée. Le système à carte de bus est universel, et extensible par étapes successives et indépendantes les unes des autres. L'autre

carte d'interface pour le Junior Computer

Comme nous l'avions annoncé, voici enfin la deuxième pièce maîtresse de l'édifice du Junior Computer. Il s'agit d'une carte d'interface qui, d'un seul coup, va faire passer votre micro-ordinateur à l'âge d'homme. D'adolescent solitaire et bouillonnant de virtualités, le voici devenu sociable, et en pleine possession de ses moyens. Il s'accommode désormais d'un terminal vidéo avec clavier ASCII associé (Elekterminal), d'une imprimante, d'extensions mémoire RAM et/ou EPROM, de circuits d'interface divers, et bien sûr, ce n'est pas là sa moindre qualité, d'une interface cassette. Cet article n'est qu'un condensé du contenu des quelques centaines de pages du livre 3 (qui paraîtra d'ici à la fin de l'année en deux volumes). Ne vous attendez donc pas ici à une étude exhaustive de cette carte. Nous allons tout de même en quelques pages passer en revue (et en détails) l'essentiel de ce nouveau circuit. Prenez le bus, et progressez...

autant. Rien n'empêche en effet d'autres extensions et développements; nous considérons toutefois qu'à l'heure actuelle, un certain équilibre est atteint. Oublions un instant les programmes en langage machine tels que nous les avons pratiqués jusqu'ici, qu'il s'agisse de programmes d'application ou du logiciel résident (moniteur, éditeur, etc.). Qu'en est-il alors des langages évolués? Le choix semble se porter sur le BASIC. Dès à présent, nous croyons pouvoir affirmer qu'un tel BASIC (par exemple 8K sur cassette, associés à une RAM de 16K) émanera d'un logiciel déjà existant, et non pas développé par Elektor. Quelles sont vos idées (par écrit de préférence!) sur ce sujet?

(Il y a dans nos cartons une carte RAM dynamique de 16K. Du point de vue de l'investissement, ce n'est pas un gouffre à économies, bien au contraire, puisque pour un prix égal, voire moindre, une telle carte remplace deux cartes 8K RAM statique).

Mais chaque chose en son temps... Revenons à notre propos du moment! Que faut-il pour étendre un micro-ordinateur tel que le Junior Computer? Comment cela doit-il marcher? Ou plutôt, comment cela peut-il marcher, puisqu'à priori les possibilités sont vastes?

Bus, carte et carte de bus

Faisons une distinction préliminaire: pour un usage amateur, un système d'ordinateur peut prendre essentiellement deux aspects: d'une part le système monocarte, appelé "single board" outre-Atlantique, et d'autre part le système à carte de bus. Dans le cas

de ce dernier, chaque sous-ensemble est monté sur une carte à part. Celles-ci sont toutes de même dimension (100 x 160 mm pour les cartes au format européen), et sont enfichées sur une carte de bus qui les relie les unes aux autres par le jeu de pistes cuivrées reliant les différents connecteurs. Sans oublier la possibilité de relier les différentes parties par du câble en nappe. En principe la longueur de la carte de bus, c'est-à-dire le nombre des sous-ensembles qu'elle est capable de recevoir, est illimitée. Le système à carte de bus est universel, et extensible par étapes successives et indépendantes les unes des autres. L'autre

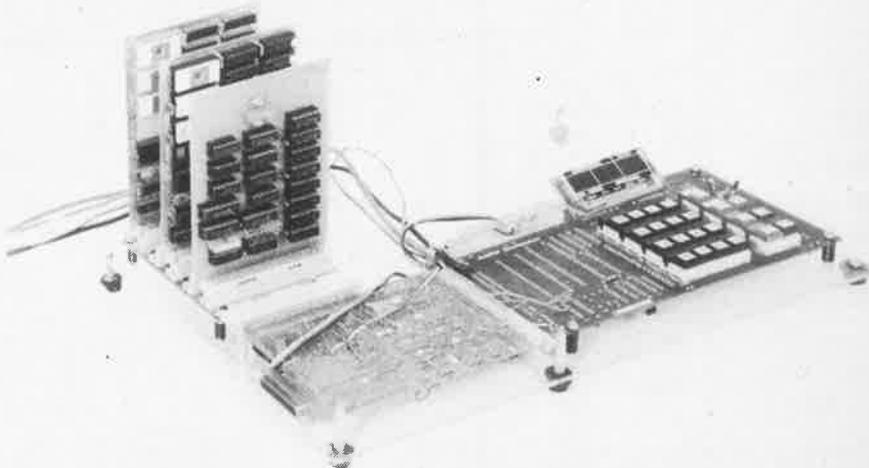
système (monocarte) se présente, comme son nom l'indique, de manière radicalement différente, puisque tous les sous-ensembles se trouvent réunis sur la même carte. Les possibilités du système sont alors étroitement liées à la taille de la carte, et à son contenu. Pour ce qui est des extensions, la carte unique apparaît souvent comme un boulet à traîner... Le principe "monocarte" est particulièrement bien adapté aux systèmes d'initiation, ainsi qu'aux systèmes de commande de processus relativement peu complexes (dans lesquels l'ordinateur est constamment en service). Dans sa version standard, le Junior Computer est un système monocarte puisqu'il répond aux définitions que nous venons de donner. Vous êtes plusieurs milliers à vous en servir sous cette forme... et vous avez bien raison!

Nous avons aussi eu des échos d'usages plus spécifiques, qui vont de la simple connexion de convertisseurs D/A ou A/D au pilotage d'unités de fabrication de semiconducteurs. Les possibilités de la version standard sont illimitées en fait, et cela n'étonnera personne. Et pourtant, il y a un connecteur d'extension!

S'il est là, c'est bien pour qu'on s'en serve. Plus précisément, pour permettre de réaliser toutes les extensions possibles et imaginables (lire: les plus indispensables d'abord): nous en avons énuméré certaines ci-dessus.

Mais maintenant que nous allons faire usage de ce connecteur d'extension, nous ne pouvons plus parler de système monocarte à propos du Junior

Computer. Cela signifie-t-il que nous basculons dans le monde des systèmes à carte de bus "de longueur illimitée"? Non. Et pourtant, le système étendu n'est plus un simple système monocarte. La carte d'extension contient tout ce qu'il faut pour relier le Junior Computer au monde extérieur par des périphériques tels que le terminal vidéo, l'imprimante, ou le lecteur de cassettes. Elle a exactement les mêmes dimensions que la carte principale et sera placée en dessous. Elle recevra à son tour une carte de bus qui servira de support aux cartes de mémoire que l'on voudra rajouter. Nous voici donc en face d'un sandwich garni d'une carte de bus! Nous l'appellerons donc "système double-carte avec bus". Mais en fait, tout cela est sans grande importance...



La carte d'interface

Le terme d'interface introduit les notions d'interconnexion, d'adaptation, de communication. Dans notre cas, il s'agit d'interconnecter le Junior Computer et le monde extérieur via une entrée/sortie spéciale, le lecteur de cassette via un circuit idoine, et les périphériques usuels via le circuit RS232; reste à le relier à la carte d'interface elle-même, ce qui nécessite aussi un circuit particulier.

Tout le circuit de la carte d'interface est reproduit par les schémas des figures 1 et 2. Il s'agit d'un circuit plus complexe que celui de la carte principale, mais il tient tout de même sur un circuit imprimé de dimensions réduites. Ce qui n'a pas été facile, vous pouvez nous croire! Nous allons examiner les différents sous-ensembles un à un.

Tampons

Sur la partie gauche, au bord de la figure 1, nous trouvons toutes les lignes provenant du connecteur d'entrée. C'est à travers ces lignes, ainsi que cinq autres provenant du connecteur de ports, que se fait l'échange d'informations avec la carte principale. Hormis les lignes EX et K1...K6 qui ne sont utilisées que pour la carte d'interface, nous retrouvons toutes ces lignes sur le connecteur de sortie, à l'extrême droite de la figure 1. Celui-ci pourra recevoir la carte de bus qui à son tour comporte des connecteurs pour les cartes de mémoire.

Les lignes d'adresses A et les lignes de données D sont rebaptisées respectivement AB et DB; le B signifiant "buffered" = tamponné. Pourquoi ces lignes doivent-elles être tamponnées? Il y a deux raisons à cela:

La première est d'éviter une trop grande charge qui résulterait de la connexion d'un grand nombre de circuits sur les mêmes lignes. Ceci concerne plus précisément les lignes d'adresses.

La deuxième raison concerne les lignes de données: il faut que celles-ci ne soient reliées qu'aux circuits momen-

tanément adressés. Les tampons des lignes d'adresses sont constitués par N1...N6. L'orientation des triangles qui servent de symboles pour ces tampons indique clairement le sens de circulation des informations depuis la carte principale vers la carte d'interface: l'entrée est à gauche, la sortie à droite. N17...N32 sont les tampons des lignes de données. Ceux-ci sont répartis en deux groupes: l'un, portant des numéros pairs avec l'entrée à droite et la sortie à gauche, et l'autre, portant des numéros impairs, avec l'entrée à gauche et la sortie à droite.

Il y a trois cas de figure possibles:

1) Les tampons portant des numéros pairs sont actifs; ceux qui portent des numéros impairs sont inactifs. C'est ce qui se produit lorsque les lignes READ et WRITE sont au niveau logique bas (0). Il y a flux de courant de la carte principale vers la carte d'extension, c'est-à-dire que se déroule une opération d'écriture dans des cellules de mémoire qui ne font pas partie de la version standard.

2) Les tampons impairs sont inactifs ("tri-state", c'est-à-dire que leur entrée et leur sortie présentent une très haute impédance) et les tampons pairs sont actifs. C'est ce qui se produit lorsque les lignes READ et WRITE sont au niveau logique haut (1). Il y a flux de courant depuis la carte d'interface vers la carte principale, c'est-à-dire que se déroule une opération de lecture dans des cellules qui ne font pas partie de la version standard.

3) Tous les tampons de N17...N32 sont inactifs lorsque la ligne WRITE est au niveau logique 1 et la ligne READ au niveau logique zéro. Aucun courant ne circule, ni dans un sens, ni dans l'autre. Il y a donc une opération de lecture ou d'écriture dans des cellules de la version standard du Junior Computer. Les

signaux READ et WRITE sont délivrés par la PROM IC 17. Nous y reviendrons plus tard.

NB: Il reste un cas de figure que nous n'avons pas envisagé: la ligne WRITE est au niveau logique zéro et la ligne READ au niveau logique 1, ce qui signifierait que tous les tampons N17...N32 sont actifs en même temps. Ceci ne se produit jamais.

Circuit d'entrée/sortie

Le VIA contenu dans IC1, du type 6522, c'est toute une histoire à lui seul. Il fera l'objet, comme le 6532 dans le livre 2, d'un chapitre entier, et pas des moindres, dans le livre 3. Ce circuit Versatile Interface Adapter offre encore plus de possibilités que le PIA que nous avons utilisé dans la version standard. La liaison entre le VIA et le monde extérieur se fera par le "connecteur VIA" que l'on trouve sur la figure 1 (il ne s'agira pas d'un véritable connecteur, d'où la présence des guillemets).

Le 6522 est adressé par les lignes AB0...AB3 et divers signaux provenant du bus de commande. La présence des huit lignes de données ne devrait étonner personne, pas plus que la présence de la ligne IRQ (émission d'une interruption, à la fin du décomptage, par le temporisateur).

Le 6522 possède comme le 6532 deux entrées "chip select": CS2 du 6522 est précisément relié au signal K6 du décodeur d'adresses IC6, comme l'est déjà CS2 du 6532. CS1 est relié à la sortie de N35, qui reçoit à son entrée les lignes K6 et AB9. Pour que le VIA soit adressé, CS2 doit être au niveau logique bas (0) et CS1 au niveau haut (1). N35 est une porte NOR dont la sortie est au niveau logique haut lorsque toutes ses entrées sont au niveau logique bas (AB9 et K6 doivent être à 0). Pour que

1

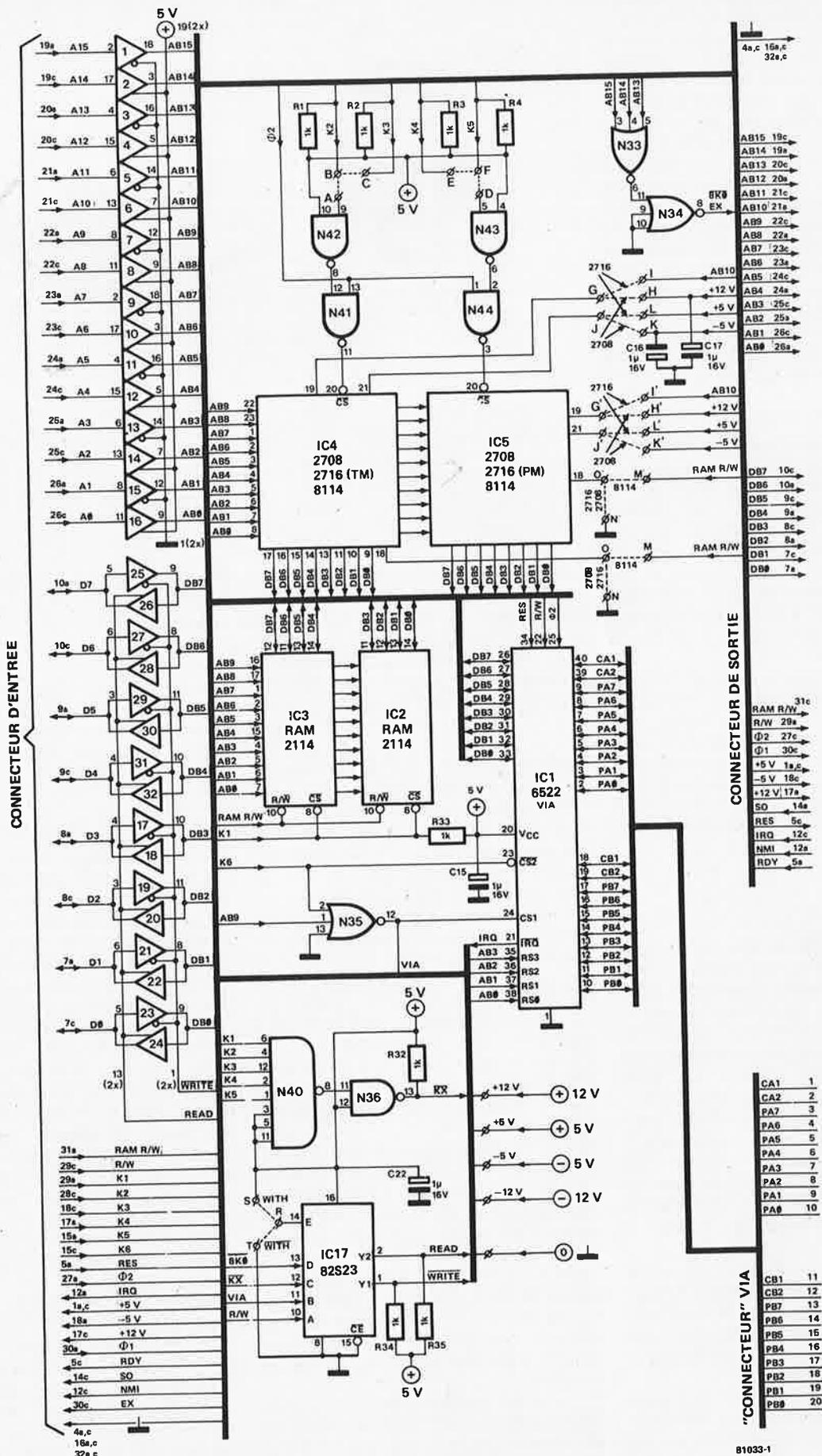


Figure 1. Voici le schéma de la carte d'extension. Cette partie du circuit comporte le décodage d'adresses, un nouveau circuit d'entrée/sortie, de la mémoire supplémentaire et bien sûr les tampons des bus de données et d'adresses.

le 6532 de la carte principale soit adressé, il suffit que A9 soit au niveau logique 1. En tenant compte du fait que A8 (AB8) n'est relié ni au 6522, ni au 6532, on déduit que ces deux unités seront adressées de la manière suivante: 1800 = 1900 et 18FF = 19FF (AB8 = X; AB9 = "0"; K6 = "1") pour le 6522, 1A00 = 1B00 à 1AFF = 1BFF (AB8 = X; AB9 = "1"; K6 = "1") pour le 6532. Omettons le double adressage résultant de AB8 = AB8 = X, et nous nous apercevons que le PIA et le VIA disposent de 256 adresses. Pour le PIA 6532, nous avons 19 cellules différentes, plus les 128 octets de mémoire vive. Pour le VIA 6522, il n'y a que 16 cellules différentes. Comme on le voit d'ailleurs sur la figure 1, les lignes AB4... AB7 ne sont pas reliées à ce circuit.

Espace mémoire

Il y a de la place sur la carte d'extension de la mémoire supplémentaire: cet espace disponible fait usage de la totalité du décodage d'adresses encore disponible sur la carte standard (8K0). Il s'agit de 5K maximum, sélectionnés par K1... K5.

Voyons d'abord la mémoire vive: en dehors des extensions possibles, on dispose sur la carte d'interface elle-même d'un K de RAM contenu dans IC2 et IC3 (voir figure 1) à raison de 1024 quartets par circuit de RAM. Pour sélectionner cet espace, il faut que K1 = CS soit au niveau logique bas (0). Ce qui nous donne les adresses suivantes pour le supplément de mémoire vive:

0400... 07FF.

Nous disposons donc de 4 pages 04... 07, faisant suite aux 4 pages 00... 03 de la version standard, ce qui permettra d'introduire de longs programmes d'un seul tenant.

Aussi bien pour IC4 que pour IC5,

on a le choix entre 1 K RAM (8114), 1K EPROM (2708) et 2K EPROM (2716). La sélection se fait à l'aide d'un ou deux signaux K2... K5 par circuit intégré, et du signal d'horloge Φ 2 (portes N41... N44); ce dernier servant à une bonne synchronisation de la lecture (et éventuellement) de l'écriture dans ces circuits.

L'adressage est le suivant:

K2 = "0"

→ adresses 0800... 0BFF → ½ ou 1 x IC4

K3 = "0"

→ adresses 0C00... 0FFF → ½ ou 1 x IC4

K4 = "0"

→ adresses 1000... 13FF → ½ ou 1 x IC5

K5 = "0"

→ adresses 1400... 17FF → ½ ou 1 x IC5

Dans le cas de l'utilisation d'1 K RAM (8114) ou d'1 K EPROM (2708), on dispose de deux possibilités de sélection par circuit. Pour 2K EPROM (2716), deux signaux K sont combinés: K2/K3 pour IC4 (2716) (adresses 0800... 0FFF); et K4/K5 pour IC5 (2716) (adresses 1000... 17FF).

Pour sélectionner ces adresses, on dispose des points A... F qui doivent être connectés de façon adéquate selon le choix retenu pour IC4 et IC5. Il en va de même pour les points G... O et G'... O'. La connexion de ces points dépend du choix des tensions d'alimentation d'une part, et de la nécessité d'une connexion de la ligne AB10 ou du signal RAM-R/W d'autre part.

Décodage d'adresses

Dans la version standard du Junior Computer, huit des 64 K adressables sont décodés; les lignes d'adresses A13, A14 et A15 ne sont pas du tout utilisées. Ce qui a les conséquences suivantes: les pages 0X, où X = 0... F, ne sont pas distinctes des pages 2X, 4X,

6X, 8X, AX, CX et EX. Il en va de même pour les pages 1X où X = 0... F, qui sont assimilées aux pages 3X, 5X, 7X, 9X, BX, DX et FX. Ceci est dû au fait que le point D du décodeur d'adresses IC6 de la version standard est relié à la masse en permanence, et se trouve donc au niveau logique 0 en toutes circonstances.

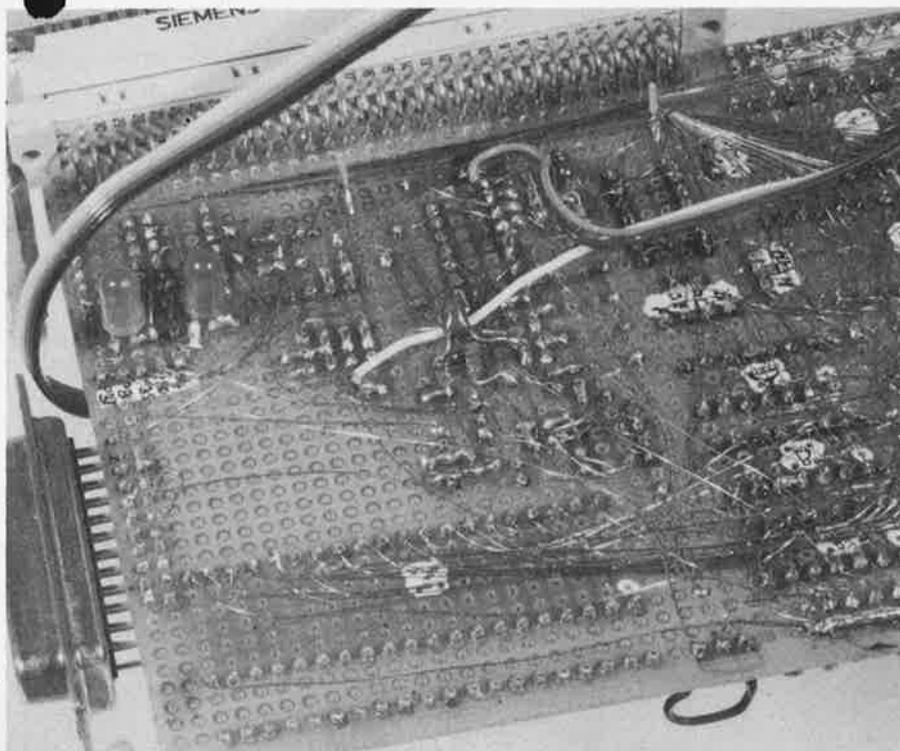
Si nous voulons étendre la mémoire au delà de 5K, il nous faut mettre en place un décodage d'adresses qui nous permette d'accéder à des adresses comprises entre 2000 et 2FFF.

Partons du fait que les 5K figurent sur la carte d'interface: il faut donc prévoir un décodage d'adresses pour la ou les carte(s) qui seraient reliées à la carte de bus. Si par contre, on se limite à l'espace mémoire disponible sur la carte principale et la carte d'interface, il n'est pas nécessaire d'étendre le décodage existant.

Voici comment modifier le décodage d'adresses de la carte principale: la première et la seule chose à faire est de retirer le strap qui relie le point D d'IC6 à la masse sur la carte principale du Junior Computer, et d'en implanter un autre qui relie ce même point D au point EX. Celui-ci est relié à la sortie de la porte NOR N34, qui elle-même est reliée à une autre porte NOR, N33; à son tour, celle-ci reçoit les lignes d'adresses A13, A14 et A15. Il suffit que l'une de ces lignes soit au niveau logique haut (1) pour que EX, donc D, soit au niveau logique haut aussi. Ce qui fait qu'à ce moment, aucune des sorties K0... K7 d'IC6 ne pourra passer au niveau logique bas. Lorsque les trois lignes d'adresses sont à zéro, le point D est à zéro aussi, et l'un des signaux K0... K7 sera actif. En d'autres termes, les cellules de la carte principale et celles de la carte d'extension ne peuvent désormais plus être adressées qu'aux pages 0X et 1X, où X = 0... F. Les pages 2X et au delà sont adressées via la carte de bus. Un décodage d'adresses supplémentaire sera nécessaire sur la ou les carte(s) reliée(s) à ce bus, afin de définir un espace mémoire compris entre 2000 et FFFF.

Le signal EX parvient aussi, sous le nom de 8K0, à l'une des lignes d'adresses de la PROM IC17. Celle-ci contient 32 octets (de 8 bits chacun); il n'est pas fait usage ici de la totalité ni des bits, ni des octets. Deux bits suffisent en effet pour définir les signaux WRITE = Y1 et READ = Y2. Nous avons vu auparavant que ceux-ci commandent les tampons des lignes de données N17 à N32. Pour adresser 32 octets, il suffit de cinq lignes d'adresses E... A que nous allons décrire.

Mais pourquoi une PROM? Pourquoi ne se contente-t-on pas du signal R/W tel qu'il est délivré par la carte principale? Il y a deux raisons à cela, dont voici la première: Supposons que nous lisions dans la RAM, ou dans l'EPROM, ou encore dans le PIA de la carte principale. Si les signaux READ et WRITE



de 8K est adressé (cellules de la carte principale et de la carte d'extension) et au niveau logique haut lorsque la mémoire via la carte de bus est adressée; 5) Le point E d'IC7 est relié soit au +5 V (strap RS), soit à la masse (strap RT). Ce point E doit être au niveau logique bas s'il n'y a pas de mémoire connectée à la carte de bus (d'où l'appellation WITH = AVEC, ce qui signifie WITHOUT = SANS). Il devra être au niveau logique haut s'il y a de la mémoire sur la carte de bus (WITH = AVEC). Mais pourquoi? Très simplement pour que vous gardiez la liberté de connecter ou pas une carte de mémoire via la carte de bus.

Le tableau 1 donne le contenu de la PROM IC17. Les bits Y3...Y8 sont nuls et ne jouent aucun rôle ici.

Il y a en principe 32 situations différentes possibles, qui se soldent chacune par l'une des trois combinaisons possibles des signaux READ et WRITE. En réalité, il y en a beaucoup moins. En dehors de la différenciation des phases d'écriture et de lecture, il reste 8 situations:

1) Écriture ou lecture dans la mémoire de la carte d'interface (IC2...IC5). Huit tampons sont en service selon que l'on écrit ou qu'on lit (adresses 00 et 01 de la PROM).

2) Lecture de l'EPROM ou lecture et écriture de la RAM ou du PIA de la carte principale. Les tampons sont tous bloqués (adresses 04 et 05 de la PROM).

3) Lecture ou écriture du VIA. Huit tampons sont en service selon que l'on écrit ou qu'on lit (adresses 06 et 07 de la PROM).

4) Lecture ou écriture dans l'espace mémoire relié à la carte de bus (adresses 0C et 0D de la PROM). Puisque, comme pour les cas 1...3, lorsque E est nul - WITH = WITHOUT - il n'y a pas de carte reliée à la carte de bus, tous les tampons doivent être inactifs, et les vecteurs NMI, RES et IRQ doivent être saisis dans l'EPROM standard.

5) Voir 1) (adresses 10 et 11 de la PROM)

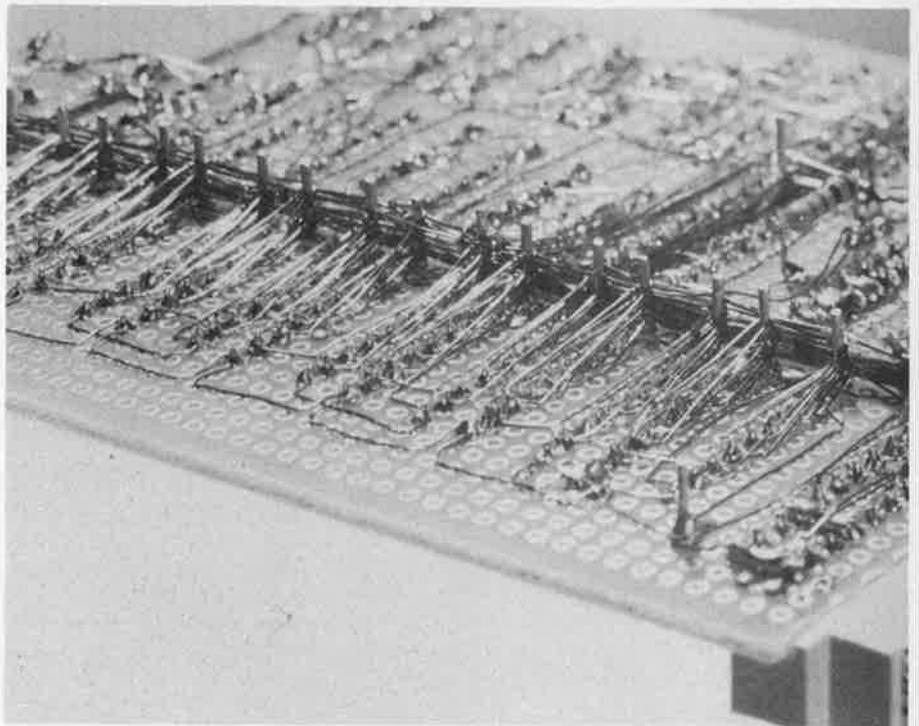
6) Voir 2) (adresses 14 et 15 de la PROM)

7) Voir 3) (adresses 16 et 17 de la PROM)

8) Voir 4) en partie (adresses 1C et 1D de la PROM).

Puisque, comme dans les cas 5), 6) et 7), lorsque E est au niveau logique haut (WITH), il y a une carte au moins sur la carte de bus. Il faut donc que les tampons correspondant au mode lecture ou écriture soient actifs. Les trois vecteurs sont maintenant extraits de la page FF. Il faut donc qu'à cette page il y ait de l'EPROM, qui spécifie les données convenables pour les vecteurs (adresses FFFA...FFFF).

Nous voyons que la moitié des trente-deux octets de la PROM suffirait, les 16 autres cas de figure ne se produisant jamais, puisque $D = 8K0$, $C = KX$ et $B = VIA$ ne sont par définition jamais au même niveau logique au même instant. Dans ces 16 situations là, Y1 et Y2 sont toujours à zéro, ce qui implique que les tampons sont en mode



écriture, totalement inoffensif. Voici pour la figure 1; reste la figure 2...

Interface cassette

L'essentiel du circuit de la figure 2 fait partie de l'interface cassette. Tout ce qu'il faut (hormis le logiciel, bien sûr) pour enregistrer des données sur cassette et pour les relire, s'y trouve. Pendant la lecture, c'est évidemment la bande qui émet les données, et pendant l'écriture, c'est elle qui les reçoit.

Ces données vont ou proviennent du CPU 6502 via la ligne de port PB7 du connecteur de ports. Pendant l'enregistrement de données, c'est-à-dire au cours du sous-programme DUMP/DUMPT (voir l'article consacré au logiciel), qui est une partie du "TAPE MANAGEMENT", PB7 est programmé comme une sortie, ainsi que PB5 et PB6. La ligne PB5 est au niveau logique zéro et la ligne PB6 au niveau un. Ce qui fait que l'entrée de N38 (broche 8) est à zéro; sa sortie, par conséquent, est à un (N38 a une sortie à collecteur ouvert qui présente une haute impédance lorsqu'elle est à un). Comme PB5 est à zéro, le transistor darlington PNP T3, relié à cette ligne par R15, est ouvert, et provoque l'allumage de la LED rouge D5 (OUTPUT ON) et le collage du relais Re2. Les contacts de J4 sont fermés. Si ce dernier est monté en série avec l'alimentation du moteur du magnétophone d'enregistrement (OUTPUT), celui-ci pourra donc être lancé par un logiciel adéquat (il faut qu'au préalable les touches d'enregistrement et de reproduction aient été enfoncées!).

Nous avons dit que pendant ce temps, PB6 est au niveau logique haut; par conséquent T2 est bloqué, la LED verte D4 n'est pas allumée et le relais Re1 n'est pas activé; les contacts de J3 sont

ouverts. La sortie de N39 nous conduit, via R20 et C14, vers P2, à l'aide duquel on ajuste le niveau d'enregistrement maximal. La sortie des données se fait via J2.

Pendant la lecture de données, c'est-à-dire au cours du sous-programme RDTAPE, PB7 est programmé comme entrée. PB5 et PB6 restent programmés comme sorties, mais la ligne PB5 passe au niveau logique 1 et la ligne PB6 au niveau zéro. Ce qui a pour conséquence que N38 transmet à PB7 le signal inversé provenant d'IC7. La porte N39 reste conductrice, ce qui signifie qu'il y a aussi un signal qui parvient sur J2; ceci n'a aucune importance en l'occurrence. La LED rouge D5 est éteinte, puisque PB5 est au niveau logique haut, et les contacts de J4 sont ouverts. Par contre PB6 est au niveau logique bas, donc T2 conduit, la LED verte D4 (INPUT ON) s'allume et le relais Re1 est activé; les contacts de J3 sont fermés (commande à distance du lecteur INPUT). Il n'y a pas d'obligation d'utiliser deux lecteurs distincts. La commande à distance perd son intérêt lorsqu'on n'utilise qu'un seul magnétophone, mais elle reste possible: il faut qu'en tout état de cause, les deux relais soient montés sur le circuit!

La commande des transistors T2 et T3 assure en toutes circonstances l'indication claire du mode dans lequel on se trouve: la LED verte est allumée pendant la lecture de données depuis le magnétophone dans l'ordinateur, et la rouge pendant l'enregistrement de données depuis l'ordinateur sur le magnétophone. Nous allons examiner maintenant la partie du circuit de la figure 2 qui se situe entre l'entrée J1 et l'entrée de N38 (broche 9), c'est-à-dire le circuit construit autour d'IC6 et IC7. Ceux d'entre vous qui ont pratiqué le KIM se diront peut-être: "tiens, j'ai déjà

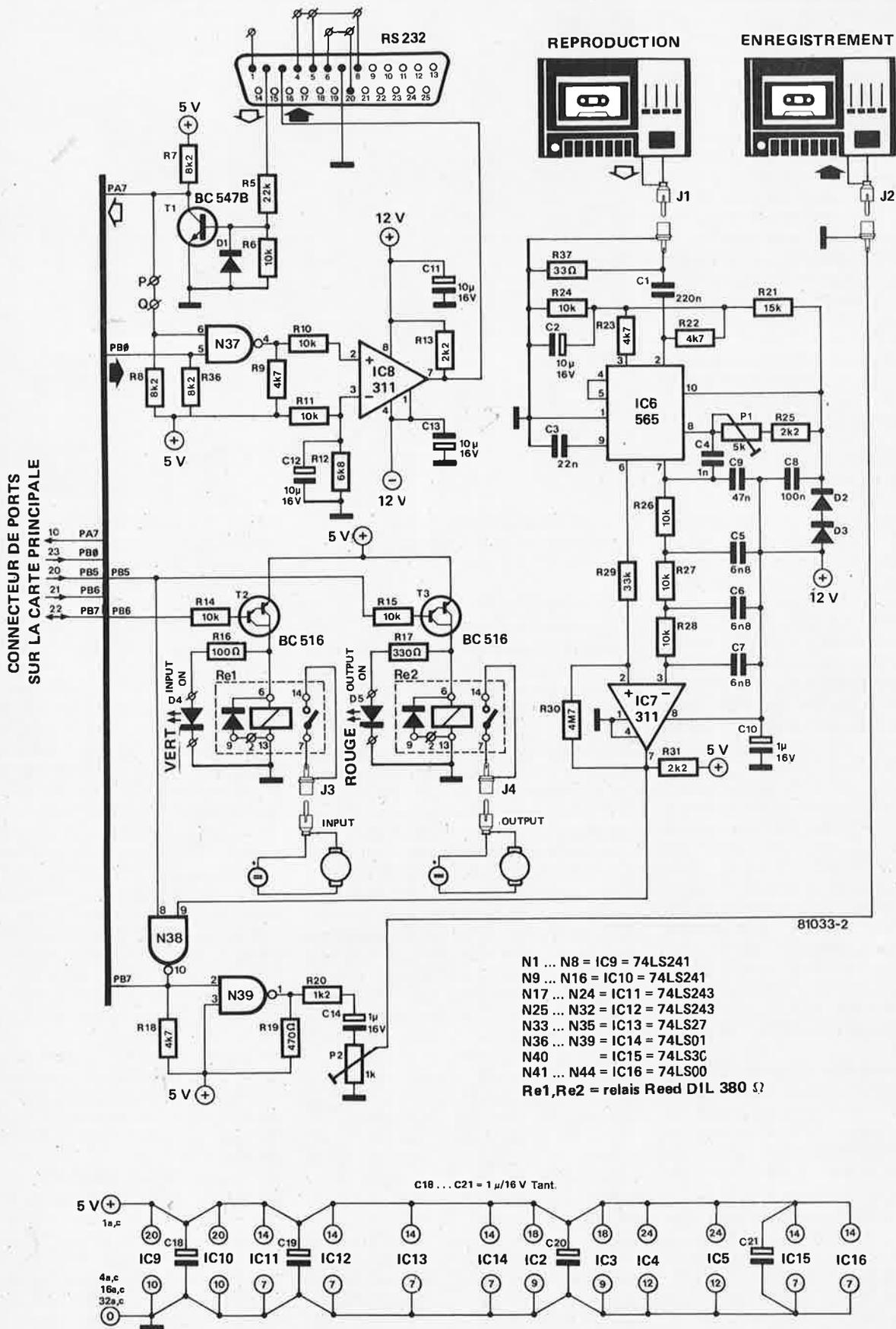


Figure 2. Voici le schéma du circuit de la carte d'interface. Il s'agit d'une part d'un circuit d'adaptation aux normes RS232 et d'autre part d'un circuit de communication avec deux lecteurs de cassettes.

vu ça quelque part". Et ils auront raison: nous sommes d'avis qu'il n'est pas nécessaire de réinventer la roue tous les matins... En termes plus clairs, il s'agit d'une partie du circuit du KIM, améliorée toutefois. On se référera également au circuit publié dans le numéro de vacances 1980.

Avant d'aborder le détail de ce circuit autour d'IC6 et IC7, il est bon de savoir que les données binaires sont fixées sur la bande sous la forme d'un signal audible, dont la forme d'onde est rectangulaire, d'une fréquence d'environ 3600 Hz, alternant avec un signal de même forme d'onde, mais dont la fréquence est de 2400 Hz. Les niveaux logiques sont obtenus à partir de la cadence à laquelle alternent ces fréquences et non pas directement, comme on pourrait le croire!

Le circuit construit autour d'IC6 et IC7 produit un "1" logique lorsqu'une fréquence de 3600 Hz est présente à l'entrée J1, et un "0" logique lorsque c'est une fréquence de 2400 Hz (ces niveaux logiques ne sont pas les bits du mot transmis). Le logiciel de RDTAPE distille à partir des deux fréquences une suite de bits parfaitement digestibles pour le micro-ordinateur. Le circuit d'IC6 avec les composants associés est ce que l'on appelle une PLL (Phase Locked Loop = boucle à verrouillage de phase). Il y aurait de quoi remplir Elektor pendant toute une année avec cette affaire-là! Ne nous en veuillez pas si nous y renonçons, et contentez-vous de quelques explications qui vont suivre.

Dans le cadre de la carte d'interface, le plus simple serait de considérer la PLL comme un "suiveur de fréquence". Exactement comme dans un montage à émetteur suiveur, la tension de sortie évolue proportionnellement à la tension d'entrée; nous avons ici un oscillateur



dont la fréquence de sortie suit celle de l'entrée, ceci dans les limites d'une certaine plage de fréquences, et au-delà d'un niveau minimum du signal d'entrée; un oscillateur qui produit une fréquence proportionnelle à une tension de commande (VCO). En l'absence de tout signal à l'entrée, la fréquence de sortie est d'environ 3000 Hz, à mi-chemin par conséquent entre 2400 et 3600 Hz. Lorsque la fréquence présente à l'entrée est de 3600 Hz, la fréquence du signal de sortie augmente de 600 Hz; lorsque la première est de 2400 Hz, la deuxième baisse de 600 Hz.

Pour que la fréquence baisse, il faut que la tension augmente à l'entrée, et inversement. La hauteur de la tension de

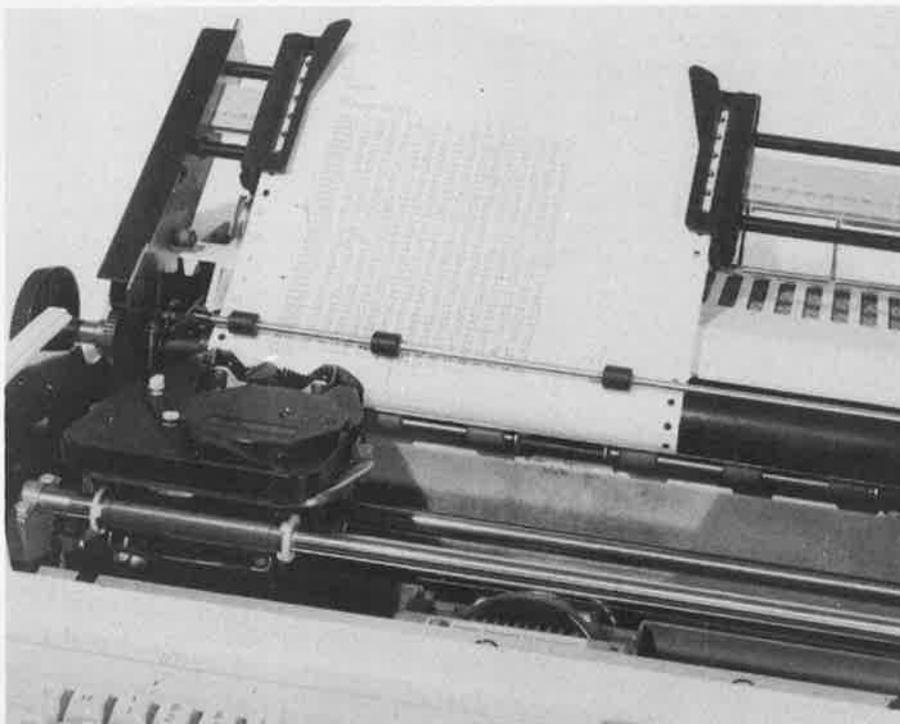
commande est donc liée à la fréquence présente à l'entrée. Ainsi, après comparaison avec une tension fixe, il est possible de distinguer les deux fréquences: "1" pour une fréquence, "0" pour l'autre, en accord avec le principe de FSK (Frequency Shift Keying = déplacement de fréquence).

La PLL est alimentée à partir du +12 V via le montage parallèle de D2 + D3 et C8. Ce circuit quelque peu remarquable assure à la PLL une assez grande immunité aux tensions parasites qui pourraient se produire sur la ligne d'alimentation et perturber la boucle.

Le "12 V protégé et nettoyé" (environ 11 V à cause des diodes) est également utilisé pour l'équilibrage des tensions d'entrée (broches 2 et 3 d'IC6) via R21...R24. Dans le circuit du KIM, il est fait usage pour cela d'une tension de 5 V qui est truffée de parasites qui parviennent allégrement à la PLL. Et pourtant, il y avait aussi du "12 V nettoyé".

De surcroît, il y a une autre différence: il n'est pas question ici d'une atténuation selon un facteur 10 du signal d'entrée avant d'atteindre la broche 2. Les améliorations que nous venons d'évoquer se font agréablement sentir dans la pratique: il faut déjà faire de grosses bêtises pour rater un chargement de données depuis la bande... on veillera au choix des bandes magnétiques, à l'azymutage des têtes ainsi qu'à leur propreté. Il nous est arrivé de nous apercevoir **après un chargement de données parfaitement réussi**, qu'il manquait la liaison de masse entre l'ordinateur et le magnétophone!!!

Des entrées 2 et 3, il n'est fait usage que de la première (commande asymétrique). La connexion avec J1 se fait via C1 dont la valeur est nettement plus faible que dans le circuit du KIM afin de filtrer



un maximum de fréquences parasites en deçà de 2400 Hz. La résistance R37 est nécessaire dans le cas où on se sert de la sortie haut-parleur ou écouteur du lecteur de cassettes.

C3, R25 et P1 influencent la fréquence du VCO en l'absence de signal à l'entrée, c'est-à-dire que leur valeur détermine cette fréquence. La méthode selon laquelle on réglera P1 détermine à son tour la qualité et la fiabilité de la lecture des données sur la bande. Dans le livre 3, nous donnerons divers moyens de procéder pour obtenir le réglage optimal. La broche 7 d'IC 6 est la sortie de la PLL. Celle-ci délivre la tension qui veille à ce que la fréquence du VCO se déplace avec le signal présent à l'entrée. Ce point est relié au + 12 V via C9. Ce condensateur, associé à une résistance de 3K6 intégrée dans le circuit, constitue ce qu'on appelle le filtre de boucle.

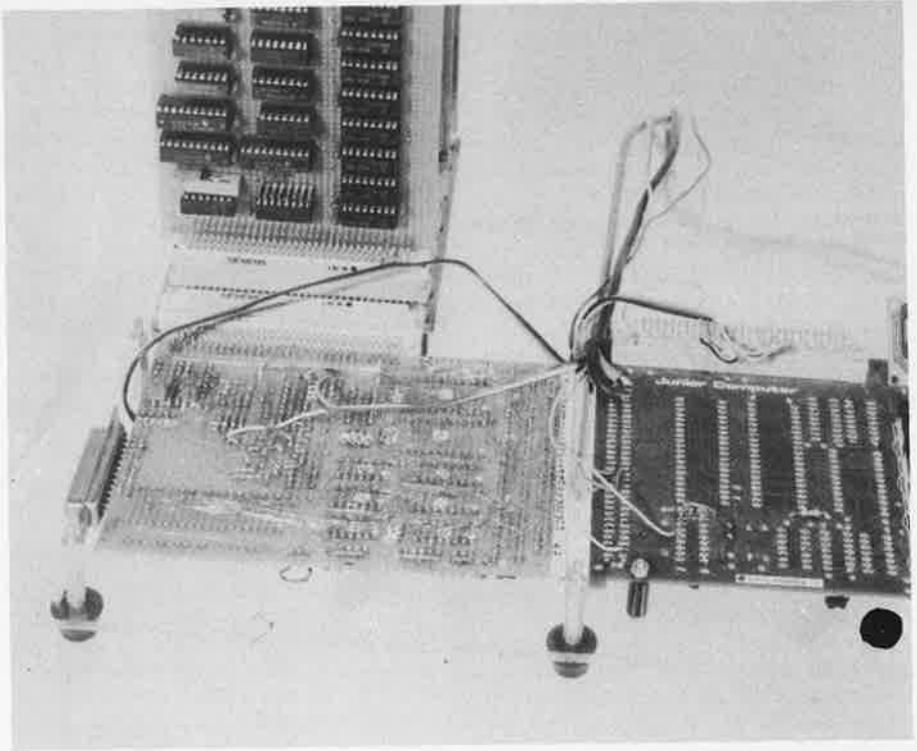
Celui-ci est important: il favorise la bonne réaction de la PLL au changement de fréquence à l'entrée. La sortie de la PLL parvient via le filtre R26... R28/C5... C7 à l'entrée inverseuse du comparateur IC7, dont l'entrée non-inverseuse reçoit la tension fixe délivrée par IC6 (broche 6) via R29.

La configuration de ce filtre est liée à la vitesse à laquelle les fréquences de 3,6 et 2,4 KHz alternent, c'est-à-dire la vitesse à laquelle les bits se succèdent sur la bande. Ce qu'on appelle le taux de transmission (baudrate in english), c'est-à-dire le nombre de bits reçus ou émis à la seconde: 800 (bauds) pour l'interface cassette du Junior Computer.

Nous avons vu auparavant que si la fréquence du VCO monte à 3600 Hz, la tension de sortie à la broche 7 diminue (dans une boucle fermée, cause et conséquence ne sont pas distinctes) et que si cette fréquence passe à 2400 Hz, la tension de sortie augmente. Ainsi, selon que la tension qui parvient à l'entrée inverseuse d'IC7 est plus basse ou plus élevée que la tension de référence présente à l'entrée non-inverseuse, nous obtiendrons à la sortie d'IC7 une tension "haute" (+ 5V via R31) lorsque la fréquence d'entrée est de 3600 Hz, et "basse" (potentiel de la masse) lorsque la fréquence d'entrée est de 2400 Hz. C'est exactement ce qu'il nous faut. La tension de sortie d'IC7 est inversée par N38 et appliquée à PB7.

NB: La PLL ne réagit pas extrêmement rapidement à un changement de la fréquence d'entrée. Ce qui a pour conséquence que la sortie d'IC7 ne bascule pas une seule fois de 1 à 0 ou de 0 à 1 lors du changement de fréquence à l'entrée. Elle bascule en effet un certain nombre de fois avant de se stabiliser: c'est ce que l'on appelle le rebond de la PLL, que l'on peut comparer au rebond mécanique des touches d'un clavier.

Le logiciel du sous-programme RDTAPE a été conçu de telle sorte qu'il s'accommode de ce petit caprice.



Connexion de périphériques via RS232C

Le petit circuit permettant de raccorder des périphériques est bien modeste (figure 2, en haut à gauche) par rapport aux énormes possibilités qu'il offre. Il s'agit d'un récepteur et d'un émetteur de données. La ligne de port PA7 est programmée comme entrée, et la ligne PB0 comme sortie. Le récepteur est constitué de T1 et des composants associés. Il n'est pas difficile de comprendre comment ce transistor inverse les niveaux qui lui parviennent via R5. Supposons que la broche 6 de N37 soit mise au potentiel + 5 V via R8, et oublions l'effet produit par le strap PQ. N37 inverse le signal qui lui parvient depuis PB0 et sa sortie commande le comparateur IC8 via R10. Celui-ci établit la comparaison avec la tension présente au point commun à R11/R12 et C12. Si la sortie de N37 est haute, la sortie d'IC8 est d'environ + 12 V. Là encore nous constatons qu'il y a inversion du signal de sortie de PB0, indépendamment de l'adaptation des niveaux logiques à des tensions particulières (+ et - 12 V).

L'entrée (via R5) et la sortie (IC8) sont reliées à un connecteur normalisé du type D à 25 broches, appelé aussi connecteur RS232. Il s'agit là d'une norme particulière à la transmission sérielle de données. Chaque octet est précédé d'un bit de départ qui annonce l'arrivée du mot. Les octets sont codés en format ASCII. La norme RS232 impose deux tensions relativement précises pour les niveaux logiques. Par après, on a rajouté la normalisation du connecteur lui-même.

Pour un niveau logique bas (0), la tension doit être comprise entre + 5 et + 15 V (version RS232C), et pour un niveau

logique haut (1) entre - 5 et - 15 V. Nous avons retenu des tensions d'environ + et - 12 V. On aura remarqué que pour un niveau logique haut, la tension est négative, et que pour un niveau logique bas, elle est positive. C'est ce que l'on appelle de la logique négative.

Mais comme pour notre interface les niveaux logiques sont inversés autant à l'émission qu'à la réception, l'ordinateur n'a que faire de ces subtilités.

Le connecteur D comporte un certain nombre de pontages internes, ceux-ci peuvent changer selon le brochage particulier du périphérique utilisé. A ce propos, qu'y a-t-il derrière un tel connecteur?

Nous pensons immédiatement à l'Elekterminal, un terminal vidéo avec clavier ASCII, mis au point dans la cadre du SC/MP, mais convenant parfaitement pour le Junior Computer. Le programme résident baptisé PRINTER MONITOR est basé sur l'utilisation d'un clavier ASCII pour l'introduction des données, et l'Elekterminal ou une imprimante convenable (et non, du moins pas pour l'instant, de l'imprimante à matrice de points publiée il y a quelques mois par Elektor) pour l'émission ou plutôt la visualisation des informations.

Voici pour la description du matériel de cette carte d'extension. Il resterait à parler de l'alimentation "revisitée" du Junior Computer standard, et de quelques détails à modifier sur la carte principale. Pour tout cela, ainsi que pour la description du logiciel et du montage de la carte, nous vous renvoyons aux deux articles qui font suite, et, bientôt... au livre 3 à paraître encore cette année, dans les meilleurs délais. ■