

On peut voir les choses sous un autre angle, et plutôt que de parler de touches nouvelles, on parle de fonctions nouvelles; il suffit de rajouter des inscriptions sur les touches existantes. Soit dessus, soit dessous, sur la tranche, où vous voudrez! Ceux d'entre vous qui pondent des programmes fleuves doivent commencer à s'impatienter: "Elle arrive, cette interface cassette". Oui, elle arrive, souple et séduisante, avec son mode d'emploi... A plus forte raison que maintenant, avec la carte d'extension, il est permis d'introduire des programmes depuis l'adresse 0200 jusqu'à l'adresse 07FF, d'un seul tenant; soit 1536 octets qu'il vaut mieux ne pas avoir à réintroduire avec les doigts sur le clavier. Autrement dit, le slogan de l'interface cassette pourrait être "1536 octets les doigts dans le nez!"

de la quincaillerie à la programmation

Deux kilos de logiciel à la carte

Pas de matériel sans logiciel, ça va de soi. Si nous élargissons le matériel (voir ailleurs dans ce numéro), il va bien falloir élargir aussi le logiciel. Voici donc deux acolytes que nous mettons à votre disposition, pour noter vos petits plats de bits à la sauce Elektor.

Tout d'abord un programme baptisé "TAPE MANAGEMENT", qui assure les opérations de lecture et d'écriture de données sur la bande magnétique.

L'autre, baptisé "PRINTER MONITOR", gère les contacts entre le Junior Computer et un Elekterminal ou une imprimante. Le premier reconnaît deux touches de fonctions nouvelles, et le second en reconnaît 10. Les unes se trouveront sur le clavier standard du Junior Computer (tape management), les autres sur le clavier du terminal ou de l'imprimante (printer monitor). Cet article n'est qu'une version abrégée du livre 3 qui ira lui au fond de ce nouveau logiciel d'envergure.

La cassette: une RAM magnétique

Comptons 150 à 200 F pour un lecteur de cassettes (mono) convenant à notre usage, et quelques cassettes à 8 F la pièce environ, et nous voici équipés. Sur une cassette C-60, il faut compter qu'il y aura 3 ou 4 minutes de pauses, d'amorce de début et de fin; il reste donc, par face, environ 25 minutes pour les données proprement dites. Pour une vitesse de transmission de 50 octets à la seconde (nous reviendrons là-dessus... moins vite), nous aurons donc de quoi emmagasiner quelques 25 x 60 x 50 octets, soit 73 kilooctets (1 kilo = 1024 octets). Ça va comme ça, ou je vous l'emballe... n'oubliez pas votre monnaie!

Il serait étonnant que vous trouviez moins cher; vous pouvez toujours chercher, on ne vous remboursera rien de toutes façons!

Comment les données figurent-elles sur la bande?

La figure 1 donne les indications nécessaires à la compréhension de la conformation des données, une fois qu'elles sont enregistrées sur la bande. Ces données peuvent constituer des programmes complets, bien sûr, mais elles peuvent aussi être dépareillées, sous forme de données isolées, de tableaux, textes, etc. En règle générale on parle d'un **bloc de données**.

Voyons comment est constituée la figure 1:

1) 255 caractères de synchronisation: ceux-ci sont là pour permettre de distinguer indubitablement le début effectif d'un bloc de données d'autres informations peu comestibles pour le

Junior Computer; n'essayez pas de lui faire avaler les premiers mots de bébé, la fanfare de votre fils ou votre frère, ou plus vraisemblablement, des commentaires du genre: "Ceci est ma première horloge... mon dernier programme d'alunissage..." Le Junior Computer n'est pas une autruche!

Remarque: Toute information enregistrée sur la bande magnétique l'est en format ASCII, par mots de 8 bits. Le dernier (ou premier) bit à gauche est réservé à des fonctions spéciales: dans notre cas il est à zéro. Les bits d'un octet en format ASCII sont stockés les uns derrière les autres, en série, ou à la queue leu leu si vous préférez. Le code ASCII d'un caractère de synchronisation est 16 en base hexadécimale.

2) le caractère de début "*": sa fonction est de signaler que la série des caractères de synchronisation est passée, et que les données proprement dites arrivent (code ASCII hexadécimal 2 A)
 3) le numéro de programme ID: celui-ci permet de différencier les programmes les uns des autres: "ID" pour identification. 254 numéros de programmes différents sont possibles, c'est à dire toutes les valeurs comprises entre 01 et FE. Les valeurs 00 et FF sont réservées à un usage particulier au cours de la phase de lecture des données sur la bande.

4) l'octet d'adresse droit SAL: il correspond à la première adresse de mémoire du programme ou du bloc de données à translater sur la bande.

5) l'octet d'adresse gauche SAH

6) le bloc de données lui-même: chaque octet de données est transcrit en 2 octets ASCII, à raison d'un par quartet. Il apparaît alors sur la bande sous la forme d'une suite de 16 bits. On commence par l'octet qui figure à l'adresse de départ (SAH, SAL) et on s'arrête après l'octet qui figure à l'adresse EAH, EAL moins une.

7) le caractère de fin de données "/" qui indique que toutes les données ont été transmises (ASCII: 2 F)

8) les octets de contrôle CHKL et CHKH: ce qui se passe ici est particulièrement intéressant, bien que (et peut-être justement parce que) parfois cela tourne mal. Il s'agit de vérifier après transmission d'un bloc de données que tout s'est déroulé sans faute. En cours de route, il s'est peut-être perdu l'un ou l'autre bit... La bande est particulièrement propice à la mutilation de données, et ce malgré la PLL améliorée (voir article précédent). Il est donc nécessaire de vérifier si lors de la lecture, l'un ou l'autre bit (ou les deux) n'ont pas été sauvagement mutilés par une perfide chasse-trape. Le meilleur moyen est de faire comme tout berger qui se respecte: *compter ses moutons*.

Au début d'une opération d'écriture d'un bloc de données, les emplacements CHKH/CHKL sont mis à zéro. Puis les octets sont additionnés les uns aux autres à partir de SAL. Ceci se passe avant la transcription en code ASCII. Le

1

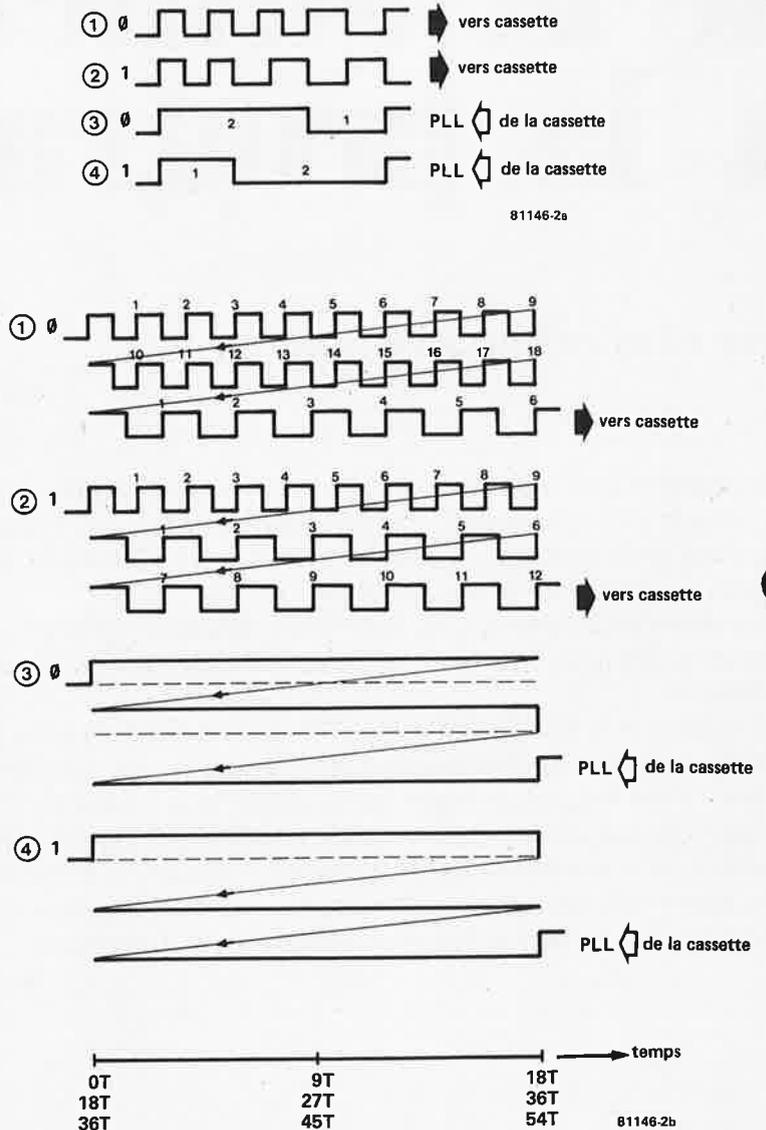


81146-1

Figure 1. Voici la configuration d'un bloc de données sur la bande: au milieu, les données proprement dites, précédées par 255 caractères de synchronisation, le caractère de début, le numéro d'identification, l'adresse de départ; après les données viennent le caractère de fin de données, les octets de contrôle, et les deux octets de fin d'émission.

2

$$T = \frac{1}{2 \cdot f} \text{ avec } f = 3600 \text{ Hz}$$



81146-2b

Figure 2. Les données sont émises sous forme de signaux de fréquence audible. Les quatre premiers trains d'impulsions appartiennent au Junior Computer (a) et les quatre derniers au KIM. Étonnante différence de longueur entre des signaux qui remplissent pourtant exactement la même fonction! On remarquera que nous avons dû "casser" les trains du KIM afin de les faire tenir sur notre schéma.

numéro de programme (ID) n'est pas pris en compte. Chaque fois que le contenu de CHKL atteint FF, le contenu de CHKH est incrémenté une fois; lorsqu'à son tour celui-ci arrive à FF, on recommence à zéro sans autre complication. Lorsqu'on lit les données sur la bande, on procède exactement de la même manière. Il suffit ensuite de comparer le résultat des deux opérations de comptage: la somme des octets lors de l'écriture doit être égale à la somme des octets lors de la lecture, et inversement. Si c'est le cas, il est permis d'espérer que la transmission s'est faite correctement. Dans le cas contraire, il est certain qu'il y a eu mutilation, et tout est à refaire. Nous pourrions entamer ici un nouveau chapitre et vous assommer de statistiques, théorie de l'information et autres subtilités. Mais à quoi bon, il y a plus urgent... Et nous avons encore besoin de vous! Sachez que la transcription en format ASCII que nous avons évoquée précédemment n'est pas vide de sens, et trouvera très bientôt sa justification.

Prenons l'exemple d'une banque: on s'y intéresse à l'argent des autres, c'est bien connu. On ne voudrait pas, par conséquent, qu'un petit malin se laisse enfermer un soir dans les locaux dont il ressortirait aux pâles heures du lendemain matin avec la caisse sous le bras. Comment faire pour écarter une telle éventualité? Il suffit de compter chaque jour, depuis l'heure d'ouverture jusqu'à celle de fermeture, le nombre des clients entrés et sortis de la banque, puis de comparer ces deux nombres. S'ils sont différents, il y a tout lieu de s'inquiéter; première éventualité: le système de comptage n'est pas fiable; deuxième éventualité: deux personnes ont passé exactement au même instant le seuil détecteur, qui n'en a donc enregistré qu'une (il faut changer de système, celui-ci est décidément trop primitif). Autre éventualité: un client a gagné le gros lot au LOTO, et a quitté la banque par une porte dérobée après un long entretien seul à seul avec le directeur. Bref, il se peut aussi que tout paraisse en ordre si toutes ces éventualités se présentent en même temps, et s'annulent l'une l'autre. Le ver est dans le fruit (et le voleur dans la banque). Revenons à CHKL et CHKH; à part le concours de circonstances fortuit et fâcheux que nous venons de décrire, la méthode de comptage des octets par addition est parfaitement étanche.

9) deux caractères de fin d'émission EOT: est-il besoin de le préciser, ces caractères marquent sans ambiguïté la fin de la transmission d'un bloc (ASCII: 04)

Écrire sur la bande: comment?

Nous l'avons déjà dit; les données sont transmises bit après bit. Voyons la figure 2. On y voit des trains d'impulsions, avec des wagons longs (fréquence plus élevée) et des wagons courts (fréquence moins élevée. Un bit de

niveau logique haut est fait de trois demies-périodes de 3600 Hz et quatre demies-périodes de 2400 Hz (② en figure 2a). Un zéro logique est fait de six demies-périodes de 3600 Hz et deux demies-périodes de 2400 Hz. On voit que la longueur totale du train reste la même, c'est à dire que sa durée est la même, qu'il s'agisse d'un niveau logique haut ou bas. (Soit $9T$, où T est égal à la durée d'une demie-période de 3600 Hz.) Autre particularité: dans les deux cas (1 ou 0) on commence le train par une fréquence élevée. Le rapport des durées pour les fréquences haute et basse est de deux sur un, ou un sur deux.

La figure 2b montre la manière dont les choses se déroulent dans le KIM. Nous avons été dérangés de répartir notre graphique sur plusieurs lignes, il faut donc suivre la flèche. Un bit de niveau logique haut est constitué ici de neuf périodes entières de 3700 Hz (nous arrondissons ici à 3600) plus douze périodes entières de 2400 Hz. Un bit de niveau logique bas est quant à lui fait de dix-huit périodes entières de 3600 Hz plus six périodes de 2400 Hz. Nous déduisons de ces chiffres que la durée d'un train d'impulsions, c'est à dire la durée d'un bit est presque six fois plus grande dans le KIM que dans le Junior Computer, avec tout ce que cela suppose comme conséquences sur la vitesse de lecture ou d'écriture des données "magnétiques".

Ceci dit, il ne faut pas penser de mal du KIM pour autant, car rien n'est irréversible. Nous voulons pour preuve le logiciel HYPERTAPE mis au point par J. Butterfield, qui permet d'augmenter sensiblement la vitesse de transmission, et amène la durée de bit du KIM égale à celle du Junior Computer. Le logiciel de DUMP/DUMPT (routine d'écriture) du Junior Computer diffère d'HYPERTAPE sur un certain nombre de points essentiels. Mais cela nous mènerait bien trop loin d'examiner ce détails maintenant; aussi vous renvoyons nous au livre 3 que nous préparons en ce moment; nous y consacrerons quelques paragraphes dans le deuxième volume. Nous avons dit que la durée de bit du logiciel pour cassette du Junior Computer est de 9 demies-périodes de 3600 Hz, soit $9 \times 139 = 1250 \mu s$. Soit 800 bits

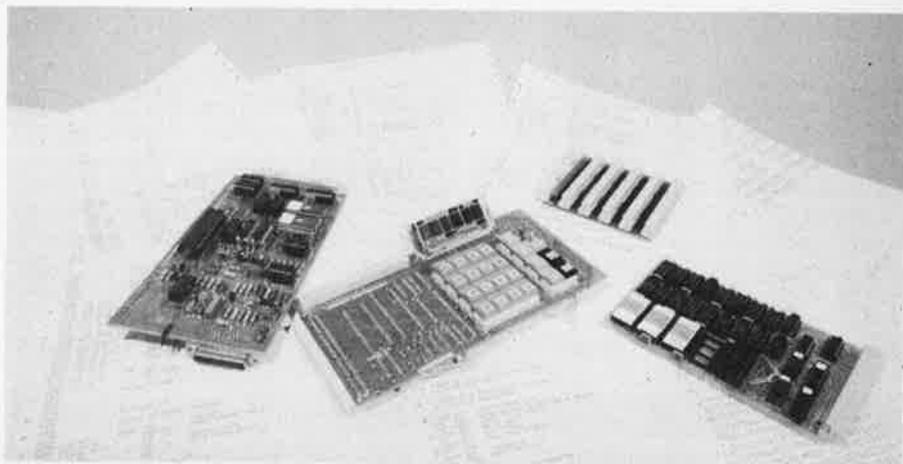
(donc caractères ASCII, et ainsi 50 octets de données) par seconde.

L'écriture des données sur la bande se fait avec la routine DUM/DUMPT que nous aborderons ultérieurement. Pendant cette opération d'écriture les six afficheurs sont éteints. Pour que l'opération puisse être entreprise, il faut indiquer:

- le numéro de programme ID (0...FE; 00 et FF ont une fonction particulière sur laquelle nous reviendrons)
- l'adresse de départ SA
- une adresse de fin EA

Lire sur la bande: comment?

Revenons à notre figure 2a. On y trouve aussi les signaux tels qu'ils peuvent se présenter à la sortie de la PLL. Celle-ci est mise en circuit lors de lectures de données sur la bande. Nous avons parlé de cela dans l'article sur le matériel, de même que le passage de ① à ③ et de ② à ④ dans la figure 2a. Nous voyons très bien au signal de sortie de la PLL comment se fait le rapport 2 : 1 et 1 : 2. Le logiciel pour la lecture distille à partir du signal de sortie de la PLL des bits de niveau logique haut ou bas selon les durées des niveaux hauts et bas en sortie de la PLL. Les durées absolues ne jouent aucun rôle, c'est le rapport entre les durées qui est déterminant: 'la durée du 3600 Hz (PLL haute) est elle notablement supérieure à la durée du 2400 Hz (PLL basse), ou l'inverse? Si c'est le cas (3600 Hz plus long que 2400 Hz), il s'agit d'un niveau logique 0, si non; c'est un 1. Comme les durées absolues n'ont pas d'importance, il est permis de relire avec le Junior Computer des durées écrites sur une bande avec le KIM, et ceci malgré la différence entre les durées de l'un et l'autre système. On voit bien en effet que les signaux ③ et ④ de la figure 2a sont du même type que les signaux ③ et ④ de la figure 2b. Par contre ils ne sont pas du même gabarit, puisque la longueur des seconds est presque six fois supérieure à celle des premiers. Un avantage qui concerne tout le monde (même ceux qui n'ont pas l'occasion de travailler avec le KIM): cette méthode des durées relatives fait que les variations dans la vitesse de lecture ou d'enregistrement des cassettes



(4,75 cm/s le plus souvent) n'ont strictement aucun effet sur la qualité de la transmission. Il n'y a donc aucune contrainte pour le choix des magnétophones et surtout leur échange, etc. Le pleurage est superbement ignoré, puisque le rapport 2 : 1 (niveau logique 0) et le rapport 1 : 2 (niveau logique 1) sont suffisamment différents pour ne pas être confondus.

NB: Au cours d'une opération de lecture de données, il n'est pas tenu compte des rebonds de la PLL (voir l'article consacré au circuit de la boucle).

La lecture se fait avec la routine RDTAPE, à laquelle on fait appel avec le programme TAPE MANAGEMENT. Il est commode de pouvoir suivre ce qui se passe (et ne se passe pas) au cours d'une opération de lecture: les deux afficheurs de droite du Junior Computer donnent les indications nécessaires (les quatre autres restent éteints).

Le dessin ① de la figure 3 apparaît lorsque:

a) la bande qui défile à ce moment devant la tête de lecture (magnéto en mode lecture) ne comporte rien de comestible (espace entre deux blocs de données, bande vierge, vagissements de bébé, etc.). Notez qu'alors l'affichage n'est pas stable, et on voit clignoter D5 et D6.

b) la bande comporte un bloc de données que l'on est en train de parcourir, mais dont on a raté le début, ou dont le numéro d'identification ne correspond pas à celui qui a été spécifié. D5 et D6 ne clignotent plus et les traits sont stables.

Le dessin ② de la figure 3 apparaît lorsque l'ordinateur est en phase de synchronisation. En d'autres termes, il est en train de lire les caractères de synchronisation qui précèdent un bloc de données. Il se peut que pour les premiers caractères, la lecture ne soit pas encore parfaite: le dessin ② est instable, mais devra se stabiliser très rapidement (1 seconde à peu près).

Il y a 255 caractères de synchronisation sur la bande. Ces 255 octets en format ASCII requièrent une durée de lecture d'environ 2,5 secondes. Le Junior Computer détecte sans coup férir le début d'un bloc de données dès qu'il a lu 10 caractères de synchronisation les uns après les autres sans interruption. On voit que la marge est grande, puisqu'il y a quelques 20 tentatives successives possibles jusqu'à épuisement des 255 caractères. Pour le KIM, il n'y a que 100 caractères de synchronisation et d'autant plus de chances que les choses tournent mal.

Le dessin ③ de la figure 3 apparaît lorsque le numéro d'identification spécifié par l'utilisateur a été trouvé et chargé dans la mémoire du Junior Computer.

Pour que la lecture des données sur la bande puisse commencer, donc avant le saut dans la routine RDTAPE, il faut qu'un numéro d'identification ait été spécifié. Sur labande peuvent figurer

3

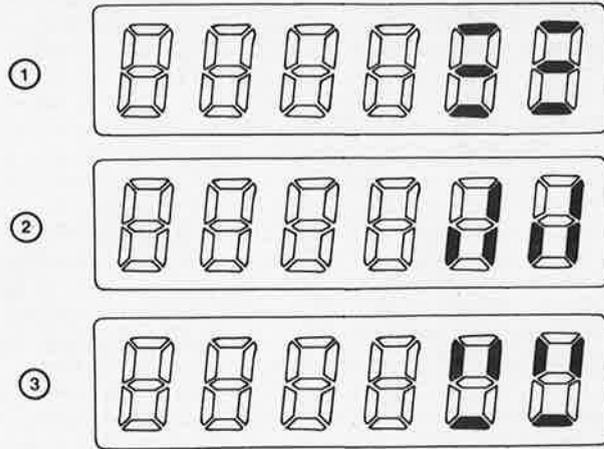


Figure 3. Voici ce que vous verrez apparaître au cours des différentes phases de lecture et d'écriture des données sur la bande.

254 blocs de données différents: il suffit de rentrer le numéro de l'un d'entre eux avant la lecture pour que l'ordinateur ait toutes les chances de le retrouver.

Il existe une autre manière de procéder: si l'on introduit 00 ou FF comme numéro d'identification, l'ordinateur chargera le premier bloc de données venu (en bonne et due forme). Dans le cas où l'on spécifie 00, le numéro du bloc de données sera ignoré, et le bloc sera rangé en mémoire à l'adresse SA figurant sur la bande. Si c'est FF que l'on spécifie, non seulement le numéro du bloc de données sera ignoré, mais aussi l'adresse de départ SA figurant sur la bande. Le bloc de données sera rangé en mémoire à une adresse spécifiée par l'utilisateur à ce moment-là.

Ceci permet de manipuler aisément les blocs de données. La seule contrainte est que lorsque FF ou 00 sont utilisés comme numéro d'identification, ce sera toujours le premier bloc de données qui se présente en bonne et due forme qui sera chargé en mémoire. Aussi est-il indispensable de connaître précisément le bloc de données à translaté (transférer, déplacer) de même que l'endroit où il se situe sur la bande. Pour cela, un magnétophone équipé d'un compteur sera le bienvenu.

Gestion de données

TAPE MANAGEMENT

Le programme TAPE MANAGEMENT (que nous appelleront TM désormais) pourrait aussi s'appeler "tape monitor". Il est là pour exécuter tous vos désirs en ce qui concerne l'écriture de données depuis le Junior Computer sur la bande (= lecture dans la mémoire du Junior Computer) et la lecture de données depuis la bande dans le Junior Computer (= écriture dans la mémoire du Junior Computer). Le programme réside dans une EPROM du type 2716. Il ne comporte pas 2048 octets, mais tout de même plus que 1024; il restera donc de la place pour quelques octets, ce qui

pourra nous être très utile plus tard. TM s'étend de 0800 à 0C7F. L'adresse de départ est 0810 et non 0800 comme on pourrait le penser! Dans certains cas on quitte TM via l'éditeur (voir figure 2), dans d'autres, après de bons et loyaux services, on le quitte par une pression sur la touche RST; on se retrouve alors dans le moniteur standard du Junior Computer. Dès après le lancement (AD 0810 GO) on voit apparaître le dessin ① de la figure 4 sur l'affichage. Si l'on actionne ensuite la touche PAR (= touche +), on voit apparaître le dessin ② de la figure 4. Et ainsi de suite pour les autres dessins; lorsqu'ils seront tous apparus une fois, une dernière pression sur PAR nous fait revenir au début (dessin ①).

Avez-vous deviné pourquoi "PAR"? Il s'agit tout simplement de l'abréviation de "paramètre", terme que l'on utilise pour les données qui servent à définir la taille du bloc de données et sa situation géographique. Les neuf dessins de la figure 4 correspondent aux neuf paramètres, dont il faut en spécifier un ou plusieurs (selon le choix que l'on aura fait parmi les quatre autres touches de fonction que reconnaît aussi TM). Ces paramètres sont:

- ID (numéro de programme ou de bloc de données)
- SAH
- SAL
- EAH
- EAL
- BEG(AD)H
- BEG(AD)L
- END(AD)H
- END(AD)L

Notez qu'ici on commence par spécifier l'octet de poids fort d'une adresse, et non pas par l'octet de poids faible comme d'habitude.

Une fois que le paramètre à spécifier est affiché, on actionne successivement deux touches numériques. Les deux valeurs (quartets) apparaissent de droite à gauche sur les deux afficheurs de droite, exactement comme en mode DA.

Après le lancement de TM, les neuf paramètres sont mis à zéro. C'est exactement ce qu'indique la figure 4. La touche PAR permet ainsi l'introduction de données qui sont nécessaires au Junior Computer pour mener à bien les opérations de lecture et/ou d'écriture. Pour l'utilisateur le procédé est commode puisqu'il "voit" de quoi il s'agit. Alors qu'avait le système plus primitif: AD 0 0 E 2 DA X X + Y Y, on ne sait pas toujours très bien où on en est: 00E2, c'est BEGADL ou BEGADH? ...

NB: Aux neuf premiers paramètres de la figure 4 correspondent neuf emplacements en page 00 ou 1A. Les emplacements 1A69... 1A7F ne doivent pas être surchargés au cours de la lecture de données sur la bande!

Que sont ces touches de fonction évoquées précédemment?

1) SAVE: un nouveau nom pour une nouvelle fonction assurée par l'ancienne touche AD. On se doute de quoi il s'agit: SAVE = SAUVER, SAUVE-GARDER des données contenues dans la mémoire du Junior Computer en les transférant sur la bande. Avant d'actionner SAVE, on enfonce les touches d'enregistrement et de reproduction (mode enregistrement) du magnétophone.

Encore avant, il faut indiquer un ID (ni 00 ni FF), et les paramètres SAH, SAL, EAH, EAL pour le bloc de données à enregistrer. **Attention!** EAH et EAL sont une adresse qui se situe un emplacement après la dernière adresse du bloc de données à enregistrer. Ainsi, pour une adresse de fin de bloc qui serait 03FF, on aura EAH = 04 et EAL = 00.

Lorsque la touche SAVE est actionnée, il est fait appel à la routine DUMP. La LED rouge D5 s'allume, mais les six afficheurs restent éteints. Lorsque le bloc de données a été enregistré, le Junior Computer s'annonce en affichant "ID XX" (dessin ① de la figure 4, avec XX à la place de 00, XX est ici le numéro du bloc de données qui vient d'être copié sur la bande.)

NB: Il peut s'avérer fort utile de noter (sur un bout de papier) l'ID, SA et EA, de même que la valeur affichée par le compte-tours du magnétophone.

2) GET: il s'agit d'une nouvelle fonction attribuée à la touche PC. Lorsque celle-ci est actionnée, le Junior Computer lit un certain bloc de données sur la bande et le copie en mémoire. Au préalable le lecteur de cassette aura été mis en mode reproduction. Et comme précédemment, il faut spécifier le numéro d'identification, c'est tout.

Sur la bande ne figurent que des numéros d'identification compris entre 01 et FE. Si avant d'actionner GET nous introduisons 00 comme numéro d'identification, le premier bloc convenable que rencontrera le Junior Computer sera copié en mémoire, sans qu'il soit tenu compte de l'ID figurant sur la bande. Par contre SAH et SAL du bloc de données sur la bande vont déterminer

l'endroit où le bloc sera déposé en mémoire.

Si avant d'actionner GET nous introduisons FF comme numéro d'identification, c'est aussi le premier bloc convenable rencontré par le Junior Computer, qui sera copié en mémoire, indépendamment de l'ID figurant sur la bande; mais il sera chargé aux adresses spécifiées au préalable par l'utilisateur. SAH et SAL figurant sur la bande sont ignorés.

Lorsque la touche GET est actionnée, il est fait appel à la routine RDTAPE. La LED verte D4 s'allume, et sur l'affichage du Junior Computer apparaît l'un des dessins de la figure 3. Une fois que le bloc est chargé en mémoire, le Junior Computer se manifeste avec "ID XX" (dessin ① de la figure 4, où XX est compris entre 00 et FF).

NB: lorsqu'on a chargé en mémoire un bloc de données avec FF pour ID, seuls les contenus d'ID et de SAH/SAL seront corrects. Qu'on ne s'attende pas à trouver l'adresse de fin de bloc dans EAH/EAL!

Il se passe encore autre chose dans les conditions que nous venons d'évoquer: l'adresse de départ (contenu de SAH/SAL) affichée sera l'adresse de fin du bloc de données qui vient d'être chargé. Ceci permet de juxtaposer des programmes édités sans qu'il y ait perte de place (supprimer les caractères EOF). Dans le même ordre d'idées, si l'on veut juxtaposer des blocs de données quelconques via ID = FF, il faut modifier (augmenter de 1) EAL et éventuellement aussi EAH, avant de charger le nouveau bloc de données. Ce type de chargement est évidemment nécessaire lorsqu'il s'agit de reconstituer un programme dont différentes sections sont disséminées sur la bande. Il s'agit alors de savoir exactement "où chercher quoi"!

3) EDIT: c'est là une nouvelle fonction attribuée à la touche DA. En fait, ce n'est pas une fonction tout à fait nouvelle, puisqu'actionner EDIT équivaut à actionner AD 1 C B 5 GO, c'est à dire un lancement à froid de l'éditeur. Auparavant on introduit BEGAD et ENDAD via la touche PAR (actionner PAR jusqu'à ce qu'apparaisse BEG(AD)H, introduire les données convenables, actionner PAR, introduire les données convenables). On sait que lors du lancement à froid de l'éditeur, on voit apparaître "77" sur les deux afficheurs de gauche. C'est également ce qui se produit lorsqu'on actionne EDIT. Mais pourquoi au fait cette nouvelle fonction? Etait-ce bien nécessaire? Oui. La réponse pourra être déduite de ce qui suit.

4) SEF: cette fonction est assurée par la touche GO. Elle assure une tâche particulière, juste avant le lancement de TM. SEF = Save Edited File, c'est à dire que l'on translate sur la bande un bloc de données qui n'est pas complètement édité, et par conséquent non assemblé. Partant du fait que la touche EDIT a été actionnée auparavant (lancement à froid

de l'éditeur), toutes les données du programme depuis BEGAD (qui a été introduit, ainsi qu'ENDAD, avant d'actionner EDIT) jusqu'à l'adresse vers laquelle pointe le pointeur variable CEND, sont copiées sur la bande sous forme d'un bloc de données unique via la routine DUMP.

Pour qu'il soit possible de translater un tel bloc de données via la touche SEF, il est indispensable que BEGAD et CEND soient définis, ce qu'ils ne peuvent être que lorsque le programme a été édité! Avant de copier sur la bande un programme édité via la touche SEF, il faut donc (le Junior Computer restant impérativement sous tensions: BEGAD et CEND sont en mémoire vive en page 00!) que l'on quitte l'éditeur via le moniteur pour sauter ensuite dans TM:

- RST 0 8 1 0 GO
- introduire l'ID
- et actionner SEF

L'écriture des données sur la bande se fait avec la routine DUMP. L'affichage reste éteint après que l'on a actionné SEF, par contre la LED rouge s'allume. Une fois que toutes les données ont été translattées, on voit apparaître la première instruction du programme édité que l'on vient de copier sur la bande. Ceci grâce au fait que TM assure après DUMP, un lancement à chaud de l'éditeur. Il aura fallu introduire au préalable un numéro de programme; quant à l'adresse de départ, elle est égale à BEGAD, et l'adresse de fin (la dernière adresse du bloc de données plus une) est égale à CEND.

Quels sont les avantages que nous procure SEF? Cette fonction nous permet de mettre au frais des programmes courts ou longs qui ne sont pas encore au point (non assemblés). Il suffit d'aller rechercher ces programmes via la touche GET, puis de relancer l'éditeur à chaud. Voyons à présent comment faire.

Le pointeur d'adresse de fin variable CEND désigne le premier emplacement libre à la fin du bloc de données en mémoire. Juste après le caractère EOF par conséquent. On peut se référer au chapitre 8 du livre 2 pour se remettre ces choses là en mémoire. Comme c'est le pointeur CEND qui donne EA lors d'une opération SEF, il est sûr que la dernière donnée du bloc à être lue est 77, le caractère EOF. L'utilisateur n'a pas à introduire BEGAD = SA et CEND = EA avant d'actionner SEF. Ceci est fait automatiquement par la machine, il faut par contre spécifier le numéro de programme ID, et de préférence noter ces renseignements - Pourquoi?

Parce qu'après la relecture du bloc de données:

RST ou AD 0 8 1 0 GO X Y (XY = numéro de programme)

il faut préparer le saut vers l'éditeur. Tout d'abord on actionne RST (retour de TM au moniteur). Puis on rend le contenu de BEGAD (L = 00E2, H = 00E3) égal au contenu de BEGAD =

SA, que l'on a noté lors de l'enregistrement du bloc sur la bande; on rend ensuite le contenu de CEND (L = 00E8, H = 00E9) égal au contenu de CEND = EA, que l'on aura aussi noté auparavant. Reste le contenu de CURAD qu'il faut rendre égal au contenu de BEGAD. Puis on peut procéder au lancement à chaud de l'éditeur:

AD 1 C C A GO.

Comme on a rendu CURAD égal à BEGAD, la première instruction apparaît sur l'affichage.

NB: Il ne faut **jamais** lancer l'éditeur à froid dans ce cas-là, par exemple en actionnant EDIT. La conséquence serait qu'on verrait apparaître un inopportun 77 à la place de la première instruction. Il ne vous reste alors plus qu'à pédaler courageusement dans la choucroute.

NB: Lorsque plusieurs blocs de données édités sont lus sur la bande via ID = FF, la valeur de CEND est l'adresse du dernier bloc de données lu.

Il est possible de ré-adresser un bloc de données (ou plusieurs) édités et copiés sur la bande, en le relisant avec FF pour ID. Il va de soit que dans ce cas, les paramètres CEND et CURAD doivent être corrigés en conséquence.

Le programme PRINTER MONITOR

Le papier ou un tube cathodique pour remplacer les afficheurs

Le programme PRINTER MONITOR (que nous appellerons PM désormais) s'avère être un grand consommateur de papier, lorsqu'il est fait usage d'une imprimante. Pour en parler, il nous en faudra pas mal aussi! Il est contenu dans une EPROM du type 2716, depuis l'adresse 1000 jusqu'à l'adresse 14F3. On voit qu'ici non plus l'EPROM n'est pas entièrement remplie; il reste donc de la place d'autres programmes résidents. PM connaît les touches de fonction standard suivantes: AD, DA + et GO. Mais cette fois, au lieu que les données se surchargent les unes les autres sur les six afficheurs du Junior Computer, nous allons disposer d'une visualisation synoptique de ce que nous avons fait. Dans le cas de l'utilisation d'une imprimante, la bande de papier peut se dérouler indéfiniment, et toutes les informations restent disponibles; ce qui n'est pas le cas avec un terminal vidéo dont la capacité de mémoire et d'affichage est par définition limitée.

Le lancement de PM se fait via le moniteur:

AD 1 0 0 0 GO

Lorsque l'on actionne la touche RUB de l'Elekterminal (ou pour d'autres appareils: CTRL, éventuellement DEL), le Junior Computer répond en affichant le mot "JUNIOR". On peut alors introduire une adresse de travail avec les touches 0...9 et A...F du clavier ASCII. Les zéros **non significatifs** sont facultatifs: "200" suffit pour "0200". L'adresse de travail apparaît avec le contenu de l'emplacement mémoire correspondant, une fois que la touche

4

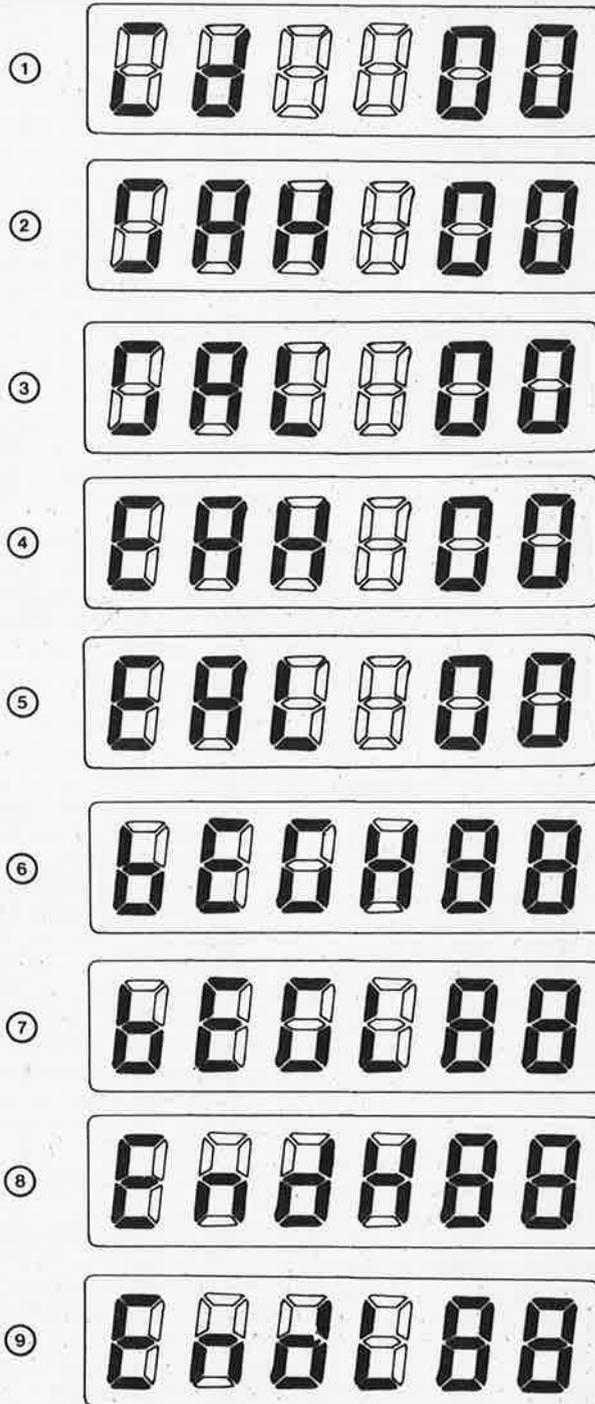


Figure 4. Les neuf paramètres nécessaires aux transferts de données de la mémoire vers la bande et réciproquement. On peut les faire apparaître un à un sur l'affichage au cours du programme TAPE MANAGEMENT, à l'aide de la touche PAR (= +).

SP (SPACE) a été actionnée. Si l'on désire introduire des données à cette adresse, on actionne successivement deux touches (0...F), puis la touche "." (point). La donnée est chargée dans la mémoire du Junior Computer qui fait alors apparaître l'adresse suivante et son contenu. Le processus peut se reproduire tel que nous venons de le décrire.

Touches de fonction

En dehors des fonctions auxiliaires telles que RUB, CR, etc, il y a dix touches de fonction:

1) la touche "-": celle-ci permet de faire apparaître l'adresse qui se situe juste avant l'adresse de travail; autre-

ment dit, cette dernière est décrémente.

2) la touche "+": celle-ci incrémente l'adresse de travail comme le fait la touche + du clavier standard du Junior Computer. La nouvelle adresse apparaît avec le contenu de l'emplacement mémoire correspondant.

3) la touche SPACE: l'adresse de travail spécifiée apparaît avec son contenu. Cette touche est comparable à tous points de vue à la touche AD du clavier standard du Junior Computer.

Les touches AD et DA du moniteur du Junior Computer sont actionnées **avant** l'introduction des données; les touches SPACE et "." le sont **après**.

5) la touche R: ("R" comme "run")

celle-ci est comparable à la touche GO; le programme est lancé à partir de la dernière adresse affichée (= adresse de travail).

6) la touche L: ("L" comme "list") le fait d'actionner cette touche provoque l'affichage du contenu de tous les registres internes du μP 6502, c'est à dire ACC, Y, X, PC, SP et P, l'un après l'autre. Le registre P est représenté par huit bits, sous chacun desquels se trouve la lettre qui permet de l'identifier: N, V, (espace), B, D, I, Z, C.

7) la touche P: ("P" comme "print") on actionnant cette touche on fait apparaître le contenu du PC (compteur ordinal) tel qu'il était dans le cas d'une exécution pas à pas juste avant que l'on quitte le programme (après exécution d'une instruction) vers PM. Aussi la touche P assure-t-elle comme le fait la touche PC du moniteur standard, la préparation de l'exécution de la prochaine instruction (actionner R). La programmation pas à pas n'est possible avec PM que si S24 est en position "ON" (la LED de la touche GO est alors allumée), et si le circuit d'IC 10 sur la carte principale du Junior Computer est correct (voir "chronique du Junior Computer, figure 1b, dans le numéro de MAI 1981).

8) la touche M: on voit apparaître le texte "HEXDUMP" lorsque l'on actionne cette touche. On introduit ensuite une première adresse (sans les zéros non-significatifs), on actionne la touche "." et enfin une deuxième adresse. Si l'on actionne alors la touche CR, on verra apparaître un vidage de la mémoire entre les deux adresses spécifiées. Au début de chaque rangée de 16 données figure une adresse qui est celle de la première donnée de la ligne.

Au dessus de chaque colonne de données figure un des chiffres de 0 à F, ce qui permet de retrouver l'adresse d'une certaine donnée, en recoupant la ligne avec son adresse et la colonne avec son numéro. La dernière ligne de l'hexdump n'est pas forcément complète, le nombre d'adresses n'étant pas toujours un multiple de 16.

9) la touche G: ("G" comme "GET" lorsque l'on actionne cette touche, puis le numéro de programme que l'on désire (ID), et enfin la touche CR, le bloc de données correspondant à l'ID spécifié est recherché par le Junior Computer sur la bande, puis copié en mémoire (à supposer que le lecteur de cassette est bien en mode lecture!). Lorsque l'opération de lecture est achevée, on voit apparaître le mot "READY". Tout s'est bien passé. Si le numéro de programme spécifié est 00, le bloc de données qui se présentera en premier de manière convenable, sera lu et copié en mémoire. Si l'ID est FF, on verra apparaître "SA"; il faudra alors introduire cette adresse, après quoi aura lieu l'opération de recherche du premier bloc convenable, et la copie dans la mémoire du Junior Computer à l'adresse spécifiée par l'utilisateur

5

(RST 1 \emptyset \emptyset \emptyset GO)

(CTRL+DEL=RUB)
JUNIOR

1A7E (SP)

1A7E 04 CF.

1A7F 00 14.

1A80 80 100 (SP)

0100 5D 18.

0101 1D A9.

0102 1C 13.

0103 3D 69.

0104 3C 08.

0105 3C +

0106 2C -

0107 AC -

0106 2C -

0105 00 -

0104 08 -

0103 69 -

0102 13 -

0101 A9 -

0100 18 L

ACC: C0

Y : CB

X : CA

PC : F33C

SP : 01FF

PR : 00000100

NV BDI ZC 100 (SP) (STEP:OFF)

0100 18 R

0107 AC L

ACC: 1B

Y : CB

X : CA

PC : 0107

SP : 01FF

PR : 00110100

NV BDI ZC F3 (SP)

00F3 1B 100 (SP) (STEP:ON)

0100 18 R

0101 A9 L

ACC: 1B

Y : CB

X : CA

PC : 0101

SP : 01FF

PR : 00100100

NV BDI ZC P

0101 A9 R

0103 69 L

ACC: 13

Y : CB

X : CA

PC : 0103

SP : 01FF

PR : 00100100

NV BDI ZC P

0103 69 R

0105 00 L

ACC: 1B

Y : CB

X : CA

PC : 0105

SP : 01FF

PR : 00100100

NV BDI ZC P

0105 00 R

0107 AC L

ACC: 1B

Y : CB

X : CA

PC : 0107

SP : 01FF

PR : 00110100

NV BDI ZC M (STEP:OFF!)

HEXDUMP: 100,105

0 1 2 3 4 5 6 7 8 9 A B C D E F

0100: 18 A9 13 69 08 00

JUNIOR

S11,100,105

READY

10) la touche S: celle-ci permet l'écriture d'un bloc de données sur la bande. Les choses se passent comme suit: actionnez la touche S; introduisez le numéro de programme que vous désirez; actionnez la touche "." et introduisez l'adresse de départ SA; actionnez encore une fois la touche "."; introduisez l'adresse de fin EA (une adresse après la dernière du bloc de données); mettez le lecteur en position enregistrement et lancez la bande: actionnez la touche CR. Une fois que tout aura été enregistré correctement, le Junior vous en avertira en affichant "READY". Voici donc l'aperçu qu'il nous était possible de donner dans le cadre de cet article, que nous allons terminer avec quelques petits détails qui peuvent vous être utiles.

Nous savons comment faire usage des touches numériques (ASCII 0...9 et A...F) ainsi que d'autres ("." ou SPACE) pour l'introduction d'une adresse de travail, ou la modification de données immédiatement après exécution de la touche de fonction, les tampons correspondants sont mis à zéro. On actionne alors la touche SPACE, et l'adresse de travail devient 0000; si l'on actionne "." à une certaine adresse de travail, le contenu de cette adresse sera mis à zéro.

Le programme PM spécifie automatiquement le vecteur NMIL; la règle est NMIL = CF (adresse 1A7A) et NMILH = 14 (adresse 1A7B). Ceci en rapport avec l'exécution pas à pas d'un programme. Si un programme se termine par l'instruction BRK, on peut sauter dans PM après déroulement du programme, à condition que le vecteur IRQ soit correctement positionné. La règle est IRQL = CF (adresse 1A7E) et IRQH = 14 (adresse 1A7F). L'introduction des données convenables pour le vecteur peut avoir lieu aussi bien avant le lancement de PM qu'après.

Après le lancement d'un programme (touche R, avec adresse de travail = adresse de départ) qui se termine par l'instruction BRK, le Junior Computer se manifeste à nouveau avec l'adresse et son contenu, qui se trouve deux adresses après celle à laquelle se trouve l'instruction BRK (à supposer que le vecteur IRQ désigne 14 CF). En mode pas à pas, le Junior Computer se manifeste en affichant l'adresse à laquelle se trouve le code opération de l'instruction suivante.