

Réalisez votre ordinateur individuel

UTILISATION DE LA CARTE

CPU 09

LE MONITEUR TAVBUG 09

A PRES avoir traité dans notre précédent numéro du câblage et de la mise en service de la carte CPU09, nous allons étudier aujourd'hui toutes les commandes dont dispose le moniteur TAVBUG09 ainsi que tous les sous-programmes auxquels vous pourrez faire appel pour vos propres réalisations.

Comme à l'accoutumée, nous allons commencer cet article par quelques informations suscitées directement par votre courrier et par l'évolution de notre développement de logiciel.

Informations diverses

Comme nous l'avons annoncé le mois dernier, un certain nombre de programmes sont déjà disponibles pour notre ordinateur et leur liste descriptive détaillée peut vous être envoyée sur simple demande adressée à la revue à l'attention de l'auteur en indiquant « informations 6809 », sous réserve que vous joigniez à celle-ci une enveloppe de format minimum 16 X 22 cm affranchie à 4 F (ou trois coupons réponse si vous résidez à l'étranger). Ne joignez aucune question à ces demandes, leur traitement n'étant pas fait par l'auteur, vos questions ne seraient pas prises en compte. Par ailleurs, en raison des pertes de temps que cela occasionne vu le volume du courrier reçu, les demandes ne se conformant pas aux indications ci-dessus (pas d'enveloppe, enveloppe trop petite ou pas de timbre) ne seront pas traitées.

Pour répondre à une question souvent formulée. Il est évident que les « gros » programmes tels que Editeur, Assembleur et Basic ne peuvent tourner avec le seul

K-octet de mémoire disponible sur la carte CPU09. Par contre, nous avons développé quelques petits programmes de jeux tels que : jeu du pendu, mastermind, jeu de marienbad, etc., qui s'accrochent fort bien d'une taille mémoire aussi réduite et qui vous permettront déjà de faire tourner votre carte CPU09 seule.

Si vous possédez des cartes RAM de l'ancien mini-ordinateur, il est évident que vous pouvez utiliser les programmes Editeur, Assembleur et Basic. Nous vous recommandons cependant d'attendre leur description dans la revue, prévue pour le numéro d'août (ou septembre au plus tard), car leurs possibilités sont plus importantes que celles de leurs « homologues » en 6800.

Au risque de nous répéter, les drives TANDON TM 100-1, - 2, - 4, que vous les ayez acquis pour l'ancien système ou pour celui-ci, conviennent ainsi que la carte IFD de l'ancien système.

Avant d'en finir avec ces informations, précisons qu'une petite erreur a pu être constatée par certains d'entre vous sur le circuit imprimé de la carte CPU09. Les résistances de rappel des lignes PA5 et PA6 du PIA ne

vont pas au + 5 V comme indiqué sur le schéma théorique mais à la masse sur les premières versions de circuit imprimé. Cela n'a strictement aucune incidence sur le fonctionnement de la carte et il est inutile de modifier quoi que ce soit à ce niveau.

Présentation de TAVBUG09

Avant de commencer cette présentation, il nous semble nécessaire de rappeler, pour le lecteur novice en micro-informatique, ce qu'est le moniteur d'un système afin qu'il ne soit pas surpris par ce qui va suivre. Un moniteur est un programme qui s'occupe de tout ce qui est « bas niveau » dans un système, c'est-à-dire gestion des circuits d'interface, gestion de la mémoire, etc.

Si vous réalisez l'ordinateur pour y développer vos propres programmes en langage machine, le moniteur vous sera d'une grande utilité puisqu'il est indispensable en phase de mise au point de programmes. Par contre, si vous considérez l'ordinateur comme une boîte noire destinée à faire du Basic, du Pascal ou de la gestion (par exemple), la connaissance détaillée des possibilités du moniteur vous importe peu puisqu'il vous suffit de savoir comment charger les programmes qui vous intéressent.

Enfin, comme pour notre précédent article, nous souhaitons

faire ici une présentation complète de TAVBUG09, aussi certaines commandes vont-elles vous paraître incompréhensibles ; n'y attachez pas trop d'importance pour l'instant. Abordez donc la lecture de ce qui suit avec l'état d'esprit correspondant à ce que vous voulez faire de votre ordinateur individuel et vous ne serez pas désorienté.

Une dernière remarque avant d'entrer dans le vif du sujet, la notion de moniteur est bien souvent inconnue sur certains mini-ordinateurs amateurs du commerce, car leurs fabricants ont délibérément choisi l'option « boîte noire » de la machine et vous présentent un produit destiné à faire du Basic, du Pascal ou autre. C'est une façon de voir qui est discutable car elle prive les utilisateurs de la possibilité de travailler, s'ils le désirent, en langage machine, alors que notre approche, tout en permettant le concept « boîte noire » évoqué ci-avant, permet aussi le travail en langage machine qui, lorsqu'on le possède bien, peut faire des merveilles.

Ce moniteur, que nous avons baptisé d'un nom semblable à celui de notre précédent système, dispose de possibilités analogues à celles de son homonyme, mais, étant donné qu'il occupe deux fois plus de place (4 K contre 2 K pour TAVBUG 6800) et qu'il est prévu pour un microprocesseur 6809, plus puissant que le 6800, vous devez bien penser que certaines fonctions nouvelles y ont été in-

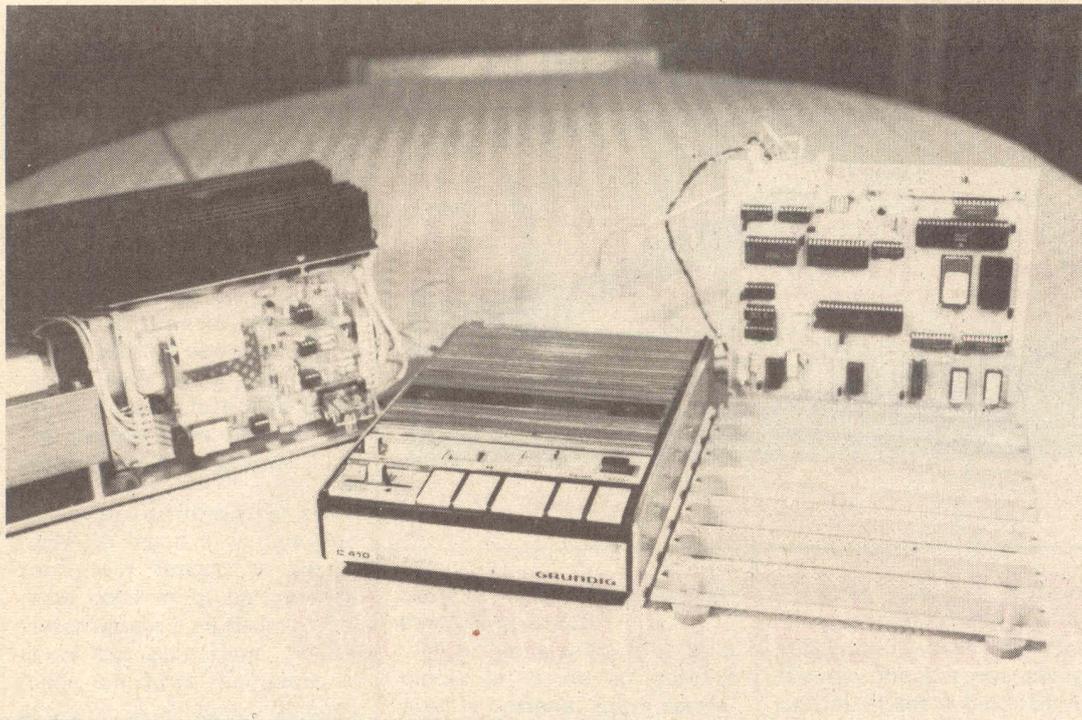


Photo 1. — Il suffit de peu de choses pour essayer TAVBUG 09.

corporées. De plus, la syntaxe de la frappe des commandes a été allégée et la présentation des informations fournies a été clarifiée.

Enfin, pour en finir avec ces généralités, il ne sera pas nécessaire de reprogrammer TAVBUG09 lorsque vous désirez utiliser le DOS, les fonctions nécessaires étant déjà prévues d'origine. De plus, et bien que cela ait été sous-entendu le mois dernier, TAVBUG09 est prévu d'origine pour se configurer automatiquement selon que vous travaillez avec un terminal quelconque ou la carte IVG de votre ancien système.

Sachez aussi que TAVBUG09 dispose de certaines protections, plus ou moins évidentes, destinées à éviter la prolifération de copies « pirates » et que l'auteur ne garantit le fonctionnement du système et des logiciels associés qu'avec les TAVBUG09 qu'il a lui-même programmé.

Pour assimiler au mieux le fonctionnement des diverses commandes décrites ci-après, nous vous conseillons de lire cet article en ayant votre système à portée de main, afin que vous puissiez essayer immédiatement les manipulations décrites. Nous vous rappelons qu'à la mise sous tension, si tout se passe bien, le circuit de RESET automatique initialise TAVBUG09 conformément à ce que vous avez signalé au moyen des mini-interrupteurs de

configuration (voir précédent numéro) et le message d'appel TAVBUG09 apparaît avec, sur la ligne suivante, le caractère d'attente de commande qui est le symbole « supérieur à » (>). Un tel caractère s'appelle un « PROMPT » dans le jargon des informaticiens, aussi utiliserons-nous cette dénomination par la suite, car elle est tout de même plus pratique que son équivalent français « caractère d'attente de commande ».

Syntaxe générale des commandes

Pour avoir le moins de choses possibles à retenir, la syntaxe des commandes a été normalisée et simplifiée au maximum possible non nuisible aux performances du moniteur. Toutes les commandes doivent être frappées en majuscules, la frappe en minuscule se traduisant par l'apparition du message ERREUR suivi à nouveau du PROMPT.

Les commandes normales répondent à la syntaxe suivante :

- Commande suivie d'un retour chariot.
- Commande Espace EXPRESSION suivie d'un RETOUR CHARIOT.
- Commande ESPACE EXPRESSION1 ESPACE EXPRESSION2 suivie d'un RETOUR CHARIOT.
- Commande ESPACE EXPRESSION1 ESPACE EXPRESSION2

ESPACE EXPRESSION3, suivie ou non d'un RETOUR CHARIOT.

Par ailleurs, il existe deux commandes dites rapides qui ne nécessitent ni argument ni retour chariot. Ce sont le « slash » (/) et le point (.) dont nous verrons le rôle en temps utile.

Les commandes utilisant des expressions sont séparées de celles-ci par un espace ou blanc, de même que les diverses expressions entre elles. Si une erreur est faite lors de la frappe d'une commande, il est possible de frapper CNTRL X pour annuler celle-ci, le message ERREUR apparaît alors suivi du PROMPT.

Le terme commande dans les exemples ci-dessus est constitué d'une lettre majuscule d'imprimerie selon la commande demandée. Les expressions peuvent revêtir plusieurs aspects selon le type de commande concerné. La forme la plus simple d'une expression est une valeur hexadécimale comportant de 1 à 4 chiffres utilisée pour définir une donnée ou une adresse. Ainsi, si nous voulons visualiser la mémoire d'adresses 1000, nous frapperons M 1000. Il est inutile de frapper les 0 non significatifs ainsi M 0010 aura le même effet que M 10. Une expression peut aussi être constituée par plusieurs valeurs numériques hexadécimales séparées par deux opérateurs arithmétiques qui sont l'addition et la soustraction, ainsi nous avons le droit de frap-

per : M 10 + 23 - 45 + 1000 - 48. Le moniteur calculera l'expression pour déterminer l'adresse réelle ainsi définie. Les valeurs numériques sont converties en interne en un mot de 16 bits et les résultats des calculs de ces expressions le sont modulo 10000, ce qui signifie que, si votre expression conduit à un résultat supérieur à FFFF qui est l'adresse mémoire maximum que peut manipuler un 6809, sa valeur sera réduite de façon à entrer dans la plage 0 à FFFF par soustractions successives de la valeur 10000. En d'autres termes, les valeurs « tournent en rond » autour de FFFF, la valeur suivant FFFF étant 0000 et ainsi de suite. Quoi qu'il en soit, cette possibilité, bien que prévue car elle ne compliquait pas le moniteur, est inutile puisque cela signifierait que l'expression que vous auriez spécifiée est en dehors de l'espace mémoire adressable, ce qui est inconcevable.

Il faut aussi noter que, lors de la frappe d'expression pour signifier des adresses par exemple, le fait de frapper plus de 4 chiffres est possible, TAVBUG09 considérera comme valides les 4 derniers chiffres frappés ; ainsi, M123456789 aura pour effet d'examiner la mémoire d'adresse 6789.

Les expressions peuvent aussi revêtir une troisième et dernière forme constituée par une lettre parmi trois ayant des significations particulières. Ces lettres sont M, P et W. M représente la valeur de la dernière mémoire examinée au moyen de la commande M. C'est-à-dire que, chaque fois que vous faites un examen mémoire au moyen de la commande M, la dernière valeur utilisée est conservée par le moniteur et vous pouvez la rappeler lorsque vous le désirez au sein d'une autre commande, quelles que soient les commandes effectuées entre temps. La lettre P représente la valeur courante du PC, ou Program Counter, ou compteur ordinal. Il est ainsi possible de savoir, lors de la mise au point d'un programme, où pointe le compteur ordinal sans avoir à se plonger dans un listing pour connaître la valeur réelle de l'adresse. W enfin est une lettre qui signifie la valeur de votre choix, définissable au moyen d'une commande de TAVBUG09.

C'est une commodité prévue pour éviter d'avoir à frapper souvent une adresse ou une expression qui revient souvent lors de la mise au point d'un programme.

Comme si cela ne suffisait pas, il existe encore une possibilité au niveau des expressions qui peuvent être utilisées. C'est la notion d'indirection. En effet, le microprocesseur 6809, comme nous l'avons vu lors de l'étude de ses modes d'adressage, peut utiliser l'adressage indirect, c'est-à-dire qu'il va chercher à l'adresse qu'on lui indique, non pas une donnée mais l'adresse où aller chercher la donnée. Pour vous éviter de fastidieuses acrobaties lors de la mise au point de programmes, TAVBUG09 peut interpréter les adresses en mode indirect, ce qui est matérialisé par la frappe après l'expression spécifiant l'adresse de caractère Américain « à commercial » (@). Ainsi si 1000 et 1001 contiennent 2537, le fait de frapper M 1000 @ fera apparaître le contenu de la mémoire d'adresse 2537 puisque le moniteur sera allé chercher en 1000 l'adresse (2537 dans cet exemple) où aller chercher la donnée.

Cette possibilité d'indirection ne fonctionne pas qu'avec les expressions numériques mais aussi avec les expressions contenant les opérateurs + et - ainsi que les lettres « spéciales » M, W et P.

La figure 1 vous présente quelques exemples d'expressions valides avec la valeur hexadécimale réelle qui leur correspond. De toute façon, comme nous l'avons conseillé ci-avant, la meilleure façon de faire est encore d'essayer. Vous pourrez

ainsi constater que l'interpréteur de commandes de TAVBUG09 peut digérer pas mal de choses.

Par ailleurs, lors des commandes générant des sorties de données qui peuvent durer longtemps (commande examen mémoire de plusieurs K par exemple), il est possible à tout instant de frapper CNTRL X pour terminer prématurément la commande en cours. D'autre part, toujours lors de ce type de commande, le fait de frapper n'importe quelle touche du clavier pendant la sortie des données suspend le déroulement de la commande aussi longtemps que vous le désirez, la reprise ayant lieu par la frappe de n'importe quelle touche du clavier. Cette possibilité a été prévue pour faciliter l'exploration d'une zone mémoire étendue de façon très rapide.

Nous aurons l'occasion de revenir sur toutes ces possibilités ci-après lorsque ce sera justifié au niveau des commandes pouvant y faire appel, commandes que nous allons maintenant étudier.

Commande A transfert mémoire

Cette commande permet de transférer sans altération le contenu d'une zone de mémoire quelconque dans une autre. La syntaxe est du type :

- A ADRESSE1 ADRESSE2 ADRESSE3 RETOUR CHARIOT, où ADRESSE1 représente l'adresse de départ de la zone de mémoire à transférer, ADRESSE2 l'adresse de fin de cette zone et ADRESSE3 l'adresse où vont

être placées les données transférées. Les remarques faites ci-avant s'appliquent, en particulier, il n'est pas nécessaire de frapper les 0 non significatifs. Par ailleurs, cette commande présente la particularité, si ADRESSE2 est inférieure à ADRESSE1, de considérer ADRESSE2 comme le nombre d'octets à transférer, ainsi par exemple :

- A 1000 2000 3000 transférera le contenu de la zone allant de 1000 à 2000 en 3000, par contre :

- A 1000 100 3000 transférera les 100 octets à partir de 1000 en 3000. Il est évidemment possible de spécifier les adresses en utilisant les possibilités décrites ci-avant pour les expressions et nous pouvons écrire :

- A W 100 P qui transférera les 16 octets (10 en hexadécimal) situés à partir de l'adresse spécifiée par W à l'adresse courante sur laquelle pointe le PC.

Commande B : points d'arrêt

Lors de la mise au point d'un programme, il est essentiel de pouvoir arrêter le déroulement de celui-ci en des points critiques

pour vérifier que le déroulement se fait bien comme prévu (ce qui n'est en général pas le cas pendant la phase de mise au point !). Cette possibilité s'appelle la mise en place de points d'arrêt. Lorsque le programme en cours d'essai arrive sur un tel point d'arrêt, il s'arrête et le contenu des registres du microprocesseur à cet instant est affiché, puis le contrôle du système est rendu au moniteur afin que vous puissiez décider de la suite des opérations.

Un moniteur de bonne qualité doit permettre automatiquement la mise en place de points d'arrêt en n'importe quel endroit d'un programme et leur effacement tout aussi automatique. TAVBUG09 permet cela et autorise jusqu'à 8 points d'arrêt simultanément dans un même programme, toute tentative d'en entrer plus se traduisant par l'affichage du message ERREUR et le refus de prendre en compte le point d'arrêt surnuméraire. La commande relative aux points d'arrêt peut revêtir plusieurs aspects suivant la fonction à accomplir :

- B EXPRESSION RETOUR CHARIOT place un point d'arrêt à l'adresse spécifiée par l'expres-

W = 1000
PC = 2000
M = 3000

1000 contient 80
1001 contient 00

Expression	Equivalent hexadécimal
1000	1000
W	1000
P - 2	1FFE
W + 100	1100
M + W - P	2000
1000 @	8000
M - P @	8000
W @	8000

Fig. 1. - Quelques exemples d'expressions valides.

```
>A 1000 2000 3000
>A W P+2 1000
>A M 10 2000
>A 2000 100 3000
```

Fig. 2. - Exemples d'utilisation de la commande A.

```
>B 1000
1000
>B 2000
1000 2000
>B 1200
1000 2000 1200
>B 1300
1000 2000 1200 1300
>B 1400
1000 2000 1200 1300 1400
>B 1500
1000 2000 1200 1300 1400 1500
>B 1600
1000 2000 1200 1300 1400 1500 1600
>B 1800
1000 2000 1200 1300 1400 1500 1600 1800
>B 1900
ERREUR
>B - 1800
1000 2000 1200 1300 1400 1500 1600
>B - 2000
1000 1200 1300 1400 1500 1600
>B -
>B
```

Fig. 3. - Exemples d'utilisation de la commande B.

sion, celle-ci bénéficiant de toutes les propriétés exposées ci-avant.
 - B - EXPRESSION RETOUR CHARIOT efface un point d'arrêt à l'adresse spécifiée par l'expression.

- B RETOUR CHARIOT fait imprimer la liste de tous les points d'arrêt en place à cet instant.

- B - RETOUR CHARIOT efface tous les points d'arrêt.

Par ailleurs, pour entrer plusieurs points d'arrêt successifs, il suffit de faire autant de fois que nécessaire la commande B EXPRESSION RETOUR CHARIOT. La figure 3 présente quelques exemples de frappe de cette commande.

Attention ! Vu le principe utilisé pour la mise en place des points d'arrêt, ceux-ci ne peuvent être placés que sur du programme contenu en RAM ; ce qui est normal puisque les points d'arrêt sont utilisés pour mettre au point des programmes et que cela se fait toujours en RAM (on ne programme pas des ROM avec un programme qui n'est pas encore au point !). Quoi qu'il en soit, si cette restriction vous pèse, sachez que nous avons tourné la difficulté en permettant de faire du pas à pas en ROM au moyen du timer programmable de la carte CPU09 et d'une commande décrite plus avant dans ces pages.

La commande C : appel d'un sous-programme utilisateur

Comme vous le savez, ou comme vous l'apprendrez en utilisant votre système ou en lisant nos articles d'initiation à la micro-informatique, un sous-programme ne peut être exécuté seul hors de son contexte puisqu'il doit toujours être appelé par un programme dit « principal » et qu'après son exécution, il retourne à ce programme principal. Lorsque l'on met au point un programme, il est intéressant de pouvoir essayer facilement les sous-programmes qui le composent ; c'est ce que permet la commande C qui peut lancer l'exécution d'un sous-programme se trouvant n'importe où en mémoire, la fin du sous-programme étant matérialisée par un retour sous le contrôle du moniteur avec impression du contenu des registres du 6809. La seule res-

triction à l'emploi de cette commande est que le sous-programme ainsi appelé soit vraiment un sous-programme, c'est-à-dire se termine par une instruction RTS (Return From Subroutine). La syntaxe est tout simplement :

- C RETOUR CHARIOT qui lance le sous-programme qui se trouve à la valeur courante du PC au moment de la frappe de la commande ou :

- C EXPRESSION RETOUR CHARIOT ou EXPRESSION spécifie l'adresse de début du sous-programme à exécuter.

La commande D : examen mémoire

Cette commande permet de visualiser sur le terminal le contenu d'une zone quelconque

de mémoire. La présentation a été optimisée pour faciliter la lecture, comme le montre l'exemple de la figure 4. Les adresses sont rappelées en début de ligne, tous les 128 octets, une ligne est imprimée pour rappeler la position des octets en horizontal ; de plus, sur la droite de cet affichage, l'équivalent ASCII de chaque octet est imprimé, les valeurs qui ne correspondent pas à un code ASCII imprimable étant matérialisées par un point. Cette disposition est optimisée pour tout terminal supportant des lignes de 80 caractères, telle que, par exemple, la carte IVG de notre ancien système. Le terminal vidéo décrit dans le Haut-Parleur de décembre permet d'exploiter cette commande mais avec un peu moins de confort puisque ses 64 caractères par

ligne l'empêchent de contenir complètement chaque ligne, l'information affichée se trouvant scindée en deux et répartie sur deux lignes consécutives. C'est là la seule limitation de ce terminal utilisé conjointement avec TAV-BUG09.

La syntaxe de cette commande est analogue à celle employée pour la commande A de transfert-mémoire, il faut en effet frapper :

- D EXPRESSION1 EXPRESSION2 RETOUR CHARIOT, où EXPRESSION1 est l'adresse de début de la zone à visualiser et EXPRESSION2 l'adresse de fin de la zone à visualiser. Si EXPRESSION2 est inférieure à EXPRESSION1, la valeur de EXPRESSION2 est prise comme le nombre d'octets à visualiser. Pour conserver à l'affichage un

```
>D 1000 1040

      0 1 2 3 4 5 6 7 8 9 A B C D E F
1000 FF 00 .....
1010 FF 00 .....
1020 FF 00 3B 20 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 ...; .....
1030 FF 00 .....
>D F050 15

      0 1 2 3 4 5 6 7 8 9 A B C D E F
F050 45 02 C4 FF E3 03 54 03 23 03 B9 03 4E 03 B6 03 E.....T.#...N...
F060 B3 05 9C FF 99 06 B5 09 16 09 59 01 A6 EB 04 00 .....Y.....
>D F100

      0 1 2 3 4 5 6 7 8 9 A B C D E F
F100 DB 9E F4 6F 04 6F 05 CC 01 A6 A7 01 B6 B2 A7 B4 ...o.o.....
>D F106

      0 1 2 3 4 5 6 7 8 9 A B C D E F
F100 DB 9E F4 6F 04 6F 05 CC 01 A6 A7 01 B6 B2 A7 B4 ...o.o.....
>D F100 F220

      0 1 2 3 4 5 6 7 8 9 A B C D E F
F100 DB 9E F4 6F 04 6F 05 CC 01 A6 A7 01 B6 B2 A7 B4 ...o.o.....
F110 E7 01 C6 12 E7 B4 0F 7A B6 EB 00 B4 60 10 8E FD .....z.....
F120 92 A1 A1 27 04 A1 A0 20 FB EC 3F 26 04 0A 7A EC ...?.....?&..z.
F130 39 ED 02 A6 8D 0C 67 A7 8D F9 C7 8E EB 00 E6 8D 9.....g.....
F140 01 C2 3A A6 84 A1 8D 0C 55 10 26 01 AF 8E EB 00 .....U.&.....
F150 E6 8D 01 B1 3A A6 84 A1 8D 0C 43 10 26 01 9D 6F .....C.&..o
F160 8D F9 9F 6D 61 26 06 30 8D FF 5D 3F 03 3F 06 17 ...ma&.O..$?.?..
F170 07 9E 2A 0C 50 D7 FA 5A 2B 06 A6 30 A7 B1 20 F7 ..*.P..Z+.O..
      0 1 2 3 4 5 6 7 8 9 A B C D E F
F180 AE 6A 9F 93 86 3E 3F 01 33 E4 DF 95 4F 5F DD 9B .j...>?.3...O..
F190 DD 8F DD 91 C6 02 34 07 17 05 DF 30 8D 06 D9 81 .....4.....O...
F1A0 2E 27 5A 30 8D 06 74 81 2F 27 52 81 20 23 14 34 ..'ZO..t./'R. #.4
F1B0 02 6C 5F 81 2F 27 4F 17 05 96 27 02 6A 5E 17 05 .1_/'^O...'.j^..
F1C0 B9 20 EB 80 0D A7 5D 9E C4 E6 80 2A 10 9E EC 5C .....5.....*...Q
F1D0 27 F7 10 DE 95 30 8D 01 7C 3F 02 20 90 5A E1 5F ?.....O..ù?. .Z.
F1E0 24 03 3A 20 E4 31 5D A6 5F 80 02 A7 5E 5A A6 80 $.: .15...^Z.
F1F0 A1 A2 26 EE 6A 5E 26 F5 3A EC 1E 30 8B 6D 5D 32 ..&.j^&:...O.m$2
      0 1 2 3 4 5 6 7 8 9 A B C D E F
F200 C4 AD 1E 16 FF 7A 6D 5E 2B C8 30 8B AE DC 9B 20 .....zm^+.O....
F210 EC FE 04 42 06 A5 04 43 05 A2 04 44 06 28 04 45 ...B...C...D.(E
>
```

Fig. 4. - Exemples d'utilisation de la commande D.

format cohérent, ce nombre est arrondi au multiple de 16 immédiatement supérieur, comme le confirment les exemples visibles figure 4. Enfin, pour ceux que la frappe des touches fatigue, une forme réduite de cette commande existe ; le fait de frapper D EXPRESSION1 RETOUR CHARIOT fait afficher les 16 octets (10 en hexadécimal) qui suivent l'adresse spécifiée par EXPRESSION1.

Comme pour la commande A, EXPRESSION1 et EXPRESSION2 peuvent utiliser toutes les possibilités déjà décrites (lettres, expressions calculées, indirection, etc.).

Le fait de frapper une touche quelconque durant l'exécution de la commande suspend l'affichage à la fin de la ligne en cours, la reprise ayant lieu lors de la frappe d'une touche tout aussi quelconque. De plus la commande peut être terminée prématurément par la frappe d'un CTRL X.

Commande E

Cette commande très particulière est destinée à simplifier un peu le travail des personnes désireuses d'écrire des programmes 6809 sans avoir d'assembleur à leur disposition. En effet, si le codage à la main des instructions 6809 est possible au moyen du manuel de programmation, certaines instructions sont délicates à coder car chaque bit du mot codé a une signification. C'est le cas en particulier pour les très nombreux modes d'adressage indexé dont dispose le 6809. Cette commande E permet donc de calculer automatiquement ce code. La syntaxe est la suivante :

— E EXPRESSION RETOUR CHARIOT, où EXPRESSION est une représentation valide sous la forme de la syntaxe assembleur 6809 d'un mode d'adressage indexé. Il est évident que vous ne connaissez pas cette syntaxe pour l'instant, mais, comme nous l'avons dit en début de cet article, nous présentons ici toutes les commandes de TAVBUG09 afin de faire un dossier complet sur lequel il n'y aura pas besoin de revenir. La figure 5 montre quelques exemples d'exécution de cette commande ; exemples que vous comprendrez sans peine lorsque nous vous aurons parlé programmation et assembleur.

Commande G : lancement d'un programme

Cette commande est classique sur tout moniteur, elle permet de lancer l'exécution d'un programme à une adresse spécifiée. La syntaxe en est simple :

— G EXPRESSION RETOUR CHARIOT, où EXPRESSION représente l'adresse de début du programme à exécuter. Il est également possible de frapper G seul suivi d'un retour chariot, l'exécution se fait alors à partir de la valeur courante du PC ce qui est très utile pour repartir d'un point d'arrêt par exemple.

Commande L : chargement d'une cassette en mémoire

Cette commande permet de charger en mémoire le contenu d'une cassette au format Kansas City évoqué dans notre précédent article. La syntaxe revête la forme suivante :

— L VALEUR RETOUR CHARIOT, où VALEUR est une valeur hexadécimale représentant un offset avec lequel la cassette peut être chargée en mémoire. En effet, le codage des informations sur la cassette contient de façon intrinsèque l'adresse de chargement en mémoire ; par contre, pour certaines applications, il peut être utile de charger un programme ailleurs qu'à l'adresse initialement prévue. Cela s'obtient en spécifiant derrière la commande L un offset (décalage en anglais). Cet offset n'est autre que la valeur qui sera

ajoutée à l'adresse de chargement en mémoire du programme. Ainsi, si vous avez une cassette devant se charger en 1000, le fait de spécifier un offset de 3000 fera charger la cassette en mémoire à partir de 1000 + 3000, soit 4000. Si vous désirez charger la cassette à son adresse normale, il faut quand même spécifier un offset qui, dans ce cas doit être nul. Comme indiqué ci-avant, les adresses tournent en rond autour de FFFF ; ainsi, si vous avez un programme qui se charge en E000 et que vous spécifiez un offset de 5000, votre programme se chargera en 3000 (E000 + 5000 - 10000).

L'utilisation de cette commande doit se faire avec certaines précautions ; en effet, il faut, avant de lancer l'exécution par la frappe du retour chariot, mettre le magnétophone à cassettes en lecture et s'assurer que la cassette se trouve bien sur une plage où est enregistré un programme, car, dans le cas contraire, le souffle de la bande magnétique peut induire en erreur le MODEM de la carte CPU09 et empêcher un chargement correct du programme qui suit. Si le chargement se passe bien, le contrôle est rendu au moniteur en fin de celui-ci et le PROMPT apparaît. Attention ! cette commande n'effectue aucun contrôle quant au chargement réel de la cassette en mémoire, en particulier si vous avez demandé un chargement à un emplacement où se trouve de la ROM ou à un endroit où il n'y a pas de mémoire, la commande L ne s'en apercevra pas. Il faut utiliser pour cela la commande V décrite plus avant dans cet article.

La commande M : examen / changement mémoire

Cette commande est une des plus utilisées lorsque l'on travaille en langage machine ; elle permet en effet d'examiner la mémoire octet par octet et de modifier celle-ci, également octet par octet. Du fait de cet emploi très fréquent, la commande M de TAVBUG09 dispose de nombreuses possibilités accroissant sa souplesse d'emploi. La syntaxe de base peut revêtir trois formes différentes :

— M EXPRESSION, où EXPRESSION possède toutes les propriétés exposées ci-avant et spécifie l'adresse à examiner.

— ADRESSE /, où ADRESSE est impérativement une valeur numérique hexadécimale ; les lettres P, W et M ne pouvant être utilisées.

— / (slash frappé seul) qui a pour effet d'examiner la mémoire dont l'adresse a été spécifiée en dernier, lors de l'exécution d'une précédente commande M.

Chacune de ces expressions a pour effet de faire imprimer l'octet qui se trouve à l'adresse spécifiée. Par contre, la suite du déroulement de la commande dépend de ce que vous allez frapper et vous avez le choix entre diverses possibilités :

— un RETOUR CHARIOT a pour effet de terminer la commande définitivement,

— un ESPACE fait passer à l'adresse immédiatement suivante, ne fait pas imprimer celle-ci mais uniquement son contenu, sans changer de ligne,

— une VIRGULE (,) fait passer aussi à l'adresse immédiatement suivante, mais ne fait imprimer ni l'adresse ni le contenu,

— un SAUT LIGNE fait passer à l'adresse immédiatement suivante et fait imprimer celle-ci ainsi que son contenu sur la ligne suivante,

— un ACCENT CIRCONFLEXE ou FLECHE VERS LE HAUT selon les claviers (↑) fait passer à l'adresse immédiatement précédente et fait imprimer celle-ci et son contenu sur la ligne suivante, *

— une EXPRESSION remplace le contenu visualisé par celui obtenu en calculant l'expression,

— un SLASH (/) fait imprimer l'adresse en cours d'examen et

```
>E ,Y
A4
>E ,U
C4
>E H, Y
20
>E ,X++
81
>E ,Y+
A0
>E ,--X
83
>E [HHHH,PCR]
9D
```

Fig. 5. — Exemples d'utilisation de la commande E.

```
>M 010+100-26+3
00- FF- 00- FF- 00-
00EC FF-
00ED 00-^
00EC FF-^
00EB 00-/
00EB 00-/
00EB 00-,,,,
00F0 FF-^'ABCDEF' FF-^
00F5 46-^
00F4 45-^
00F3 44-^
00F2 43-^
00F1 42-^
00F0 41-^
00EF 00-
```

Fig. 6. — Exemples d'utilisation de la commande M.

son contenu sur la ligne suivante, — un TEXTE compris entre deux apostrophes fait rentrer le code ASCII de chaque caractère à partir de l'adresse qui était en cours d'examen.

Comme vous pouvez le constater, cette commande M ne manque pas de possibilités.

De plus, si, pour une raison quelconque, la valeur que vous désirez placer en mémoire n'y « rentre » pas (parce que vous avez adressé de la ROM ou un endroit où il n'y avait pas de mémoire), un point d'interrogation apparaît et le contenu de la mémoire suivante est affiché.

La figure 7 montre quelques exemples d'utilisation de cette commande M que nous vous invitons à essayer sans retenue ; faites seulement attention à l'adresse de la mémoire RAM disponible sur la carte CPU09 qui va de EC00 à EF00, les 256 octets de EF00 à EFFF étant pris par le moniteur.

Commande N

Cette commande a plusieurs raisons d'être sauf si vous utilisez le terminal vidéo du numéro de décembre. En effet, elle permet de ralentir volontairement TAVBUG09 lors de la sortie d'informations sur le terminal. Pourquoi ? Tout simplement parce que certains terminaux sont trop rapides et d'autres sont trop lents.

En effet, pour fonctionner correctement, certains terminaux lents tels que de vieilles télétypes ou des terminaux Silent Texas nécessitent l'envoi, après chaque retour chariot, de caractères nuls qui ne seront pas imprimés mais qui laisseront au

chariot du terminal le temps de revenir en place. Cette façon de faire peut être étendue pour certains modèles précités à la nécessité d'envoyer un ou plusieurs caractères nuls entre deux caractères à imprimer. D'un autre côté, avec les terminaux ultra rapides comme la carte IVG, il est parfois fatigant de suivre un listing sur l'écran, cette commande permet alors d'introduire des caractères nuls qui n'apparaissent pas sur l'écran mais qui ont pour effet de ralentir l'affichage du texte sur celui-ci. Cette commande s'utilise de la façon suivante :

— N VALEUR RETOUR CHARIOT, où VALEUR est un nombre de 1 à 4 chiffres hexadécimaux, le ou les deux chiffres de poids fort indiquent le nombre de caractères nuls à envoyer entre deux caractères « normaux », et le ou les deux chiffres de poids faible indiquent le nombre de nuls à envoyer après chaque retour chariot. Ainsi, N 123 fera envoyer 1 nul entre deux caractères et 23 nuls sur un retour chariot, tandis que 4 ferait envoyer seulement 4 nuls sur un retour chariot. Pour faire fonctionner TAVBUG09 à la vitesse maximum, il suffit de frapper NO. Le terminal vidéo et la carte IVG peuvent fonctionner sans problème avec NO ; ils sont tous deux assez rapides.

Commande O : calculateur de déplacements

Comme pour la commande E, cette commande est destinée à aider les personnes qui veulent programmer en langage machine

sans avoir d'assembleur à leur disposition. Il faut en effet savoir qu'une des phases délicates de la réalisation de programmes « à la main », c'est-à-dire sans l'aide d'un assembleur, est le calcul des déplacements utilisés en adressage relatif, car cela requiert de faire des additions et surtout des soustractions en hexadécimal qui sont souvent sources d'erreurs. Cette commande calcule donc automatiquement le déplacement à utiliser en adressage relatif entre deux adresses spécifiées. La syntaxe en est la suivante :

— O EXPRESSION1 EXPRESSION2 RETOUR CHARIOT, où EXPRESSION1 est l'adresse où sera placé le déplacement ainsi calculé et où EXPRESSION2 est l'adresse où devra aboutir cet adressage relatif. Les deux expressions présentent les caractéristiques déjà maintes fois exposées. Des exemples d'utilisation sont donnés figure 7.

Commande P : enregistrement sur cassette

Cette commande est l'opposée de la commande L puisqu'elle permet la sauvegarde sur cassette d'une portion de mémoire de taille quelconque. Le format employé est le Kansas City (voir précédent article) et est, bien sûr, compatible de celui de la commande L. La syntaxe est la suivante :

— P EXPRESSION1 EXPRESSION2 RETOUR CHARIOT où EXPRESSION1 représente l'adresse du début de la zone à mettre sur cassette et où EX-

PRESSION2 représente l'adresse de fin de cette même zone.

Contrairement à la commande L, moins de précautions sont à prendre dans l'utilisation ; en effet, l'enregistrement commence par une vingtaine de secondes de données sans signification particulière et utilisées seulement à des fins de synchronisation du MODEM lors des opérations de lecture avec la commande L. Le magnétophone peut donc être mis en marche immédiatement après la frappe du retour chariot sans crainte de perdre des données. Lorsque la commande est terminée, la main est rendue au moniteur, ce qui est matérialisé par l'apparition du PROMPT. Une remarque pour les habitués de l'ancien mini-ordinateur : rien ne s'affiche sur l'écran pendant la commande P, c'est normal.

Commande R : examen/ modification des registres

Cette commande n'est utilisée que pendant la phase de mise au point de programmes, conjointement aux commandes d'exécution de programme en pas à pas et de mise en place de points d'arrêt. Son rôle consiste à empêcher le déroulement d'un programme en pas à pas à partir d'un certain point. Ainsi, par exemple, soit à mettre au point un programme dans lequel figurent des sous-programmes longs (sous-programmes de temporisation par exemple). Il est parfaitement inutile, si vous exécutez le programme en pas à pas, de faire

```
>O P+2 1000
3C3E
>O P+2 W
2B3E
>O P+2 M
2D2D
>O 1000 1010
0F 000E
```

```
>R
PC-D3BE A-4E B-00 X-CC9C Y-C847 U-C073 S-C07F CC-F4 DP-00
PC-      A-      B-45 X-      Y-AAAA
>R
PC-D3BE A-4E B-45 X-CC9C Y-AAAA U-C073 S-C07F CC-F4 DP-00
PC-
```

Fig. 8. — Examen et modification des registres avec la commande R.

Fig. 7. — Utilisation de la commande O.

```
>T 4
0P-393B PC-004D A-4E B-45 X-CC9C Y-AAAA U-C073 S-C081 CC-F4 DP-00
0P-00FF PC-004F A-4E B-45 X-CC9C Y-AAAA U-C073 S-C081 CC-F4 DP-00
0P-00FF PC-0051 A-4E B-45 X-CC9C Y-AAAA U-C073 S-C081 CC-F4 DP-00
0P-00FF PC-0053 A-4E B-45 X-CC9C Y-AAAA U-C073 S-C081 CC-F4 DP-00
>.0P-00FF PC-0055 A-4E B-45 X-CC9C Y-AAAA U-C073 S-C081 CC-F4 DP-00
>.0P-00FF PC-0057 A-4E B-45 X-CC9C Y-AAAA U-C073 S-C081 CC-F4 DP-00
```

Fig. 9. — Déroulement d'un programme en pas à pas avec la commande T.

```
>V 0000
ADR ME K7
1008 8C 8E
```

Fig. 10. — Mise en évidence d'une erreur lors d'une commande V.

dérouler à chaque fois ces sous-programmes longs, surtout une fois que vous êtes sûr de leur fonctionnement. La commande S permet donc d'inhiber automatiquement le pas à pas dès que le contenu du pointeur de pile (le registre S du 6809) descend au-dessous d'une valeur que vous avez spécifiée.

Cette commande sera revue lorsque nous vous apprendrons à programmer en langage machine avec quelques exemples.

Sa syntaxe est très simple :
 - S EXPRESSION RETOUR CHARIOT, où EXPRESSION représente la valeur en dessous de laquelle la descente du contenu de S fera arrêter le mode pas à pas.

Commande T : exécution en pas à pas

Cette commande est parmi les plus utilisées en phase de mise au point de programmes puisqu'elle permet d'exécuter un programme en pas à pas avec visualisation à chaque pas du contenu de tous les registres du 6809.

Elle peut être lancée dès le début d'un programme ou, plus généralement, à partir d'un point d'arrêt. Le format en est :

- T NOMBRE RETOUR CHARIOT où NOMBRE indique le nombre de pas de programmes à exécuter. NOMBRE peut être n'importe quelle valeur comprise entre 0000 et FFFF. De plus, la frappe d'un CNTRL X pendant le déroulement d'une commande T de longueur quelconque interrompt celle-ci prématurément (comme pour la commande D vue précédemment). Comme il est parfois utile de faire exécuter un programme en pas à pas instruction par instruction, et pour ne pas avoir à frapper T1 à chaque fois, il existe un mode particulier de cette commande qui est appelé en frappant un point (.). A chaque frappe d'un point une instruction, et une seule, du programme en cours est exécutée.

Par ailleurs, pour éviter des pertes de temps, le mode pas à pas est automatiquement inhibé deux instructions après être rentré dans un sous-programme quelconque de TAVBUG09 et est automatiquement réactivé deux instructions avant de sortir de TAVBUG09.

Commande V : vérification d'une cassette

Cette commande fonctionne exactement comme la commande L vue précédemment, mais, au lieu de charger la cassette en mémoire, elle compare celle-ci avec la mémoire. Il est évidemment possible de définir un offset, ce qui fait que la syntaxe est tout à fait analogue à celle du L :

- V VALEUR RETOUR CHARIOT, où VALEUR est un offset répondant aux mêmes contraintes que pour L.

Si la vérification se passe bien, le moniteur reprend le contrôle du système et le PROMPT apparaît ; en cas d'erreur, l'emplacement de la première erreur détectée est affiché avec l'adresse mémoire en cause, le contenu lu dans la mémoire et la valeur lue sur la cassette, comme indiqué figure 10 à titre d'exemple.

Commande W : définition de fenêtre

Nous avons déjà évoqué ci-dessus la possibilité de faire référence à une valeur au moyen de la lettre W (de l'anglais Window qui signifie fenêtre). Cette commande permet tout simplement de définir la valeur affectée à W sous la forme :

- W VALEUR RETOUR CHARIOT, où VALEUR est n'importe quel nombre hexadécimal compris entre 0 et FFFF.

Commande I : initialisation mémoire

Cette commande permet d'initialiser à une valeur quelconque n'importe quelle zone de mémoire. La syntaxe se rapproche de celle des commandes A et D puisqu'elle est la suivante :

- EXPRESSION1 EXPRESSION2 VALEUR, où EXPRESSION1 est l'adresse de début de la zone à initialiser, EXPRESSION2 l'adresse de fin et VALEUR la valeur à placer dans cette zone. EXPRESSION1 et 2 obéissent aux règles déjà vues, par contre, VALEUR doit être un mot de 8 bits en hexadécimal. Le retour chariot est automatique dès la

frappe de VALEUR. Comme pour A, si EXPRESSION2 est inférieure à EXPRESSION1, EXPRESSION2 est considérée comme le nombre d'octets à initialiser à partir de EXPRESSION1.

Commande U : changement de page mémoire

Ainsi que nous l'avons expliqué lors de la présentation du schéma de la carte CPU09, notre ordinateur peut adresser 256 K-octets de mémoire au moyen de deux lignes d'adresses supplémentaires issues du PIA. La commande U permet de changer de page mémoire, une page étant constituée d'un bloc de 64 K-octets. L'intérêt de cette commande est indéniable, en effet, les commandes précédentes faisant appel à de la mémoire agissent sur la mémoire de la page courante. Il est donc indispensable de disposer de la commande U pour spécifier sur quelle page l'on souhaite agir. La syntaxe a été réduite au minimum :

- U NOMBRE, où NOMBRE est égal à 0, 1, 2, 3 selon la page souhaitée. Le retour chariot est automatique.

Sachez aussi, qu'à l'initialisation, et pour des raisons que nous verrons au moment opportun, TAVBUG09 travaille sur la page 1. La carte IVG se trouvant en page 0 tandis que les mémoires et les circuits périphériques de la carte CPU09 sont simultanément dans toutes les pages.

Commandes X et Y : chargement et lancement du DOS

Ces deux commandes, que vous ne pouvez utiliser pour l'instant, servent à charger le DOS lors de la mise en service du système (commande X) ; ou à relancer le DOS (sans avoir à le charger à nouveau) lorsque pour une raison quelconque vous êtes passé sous le contrôle de TAVBUG09. La syntaxe est des plus simples :

- X ou Y RETOUR CHARIOT. L'action de Y est immédiate et fait apparaître instantanément le PROMPT du DOS. L'action de X est différée et charge le DOS après une phase d'attente desti-

née à permettre au circuit contrôleur des unités de disques souples de s'initialiser correctement.

Commande Q : activation d'une imprimante

Un terminal vidéo c'est très bien et c'est écologique, malheureusement, il est parfois indispensable de sortir du papier lorsque l'on souhaite garder des traces de ce que l'on fait. Le DOS dispose bien évidemment de programmes de commande d'imprimante très sophistiqués. Par contre, il est parfois utile d'employer une telle machine même sans DOS, lors d'une mise au point de programme délicate par exemple ou, plus couramment, si vous avez décidé de travailler seulement avec des cassettes. Les sous-programmes nécessaires au pilotage d'une imprimante ont donc été inclus dans TAVBUG09 et peuvent être activés à partir du BASIC ou à partir d'une commande frappée au clavier, la commande Q.

La syntaxe est simple :
 - Q RETOUR CHARIOT active l'imprimante si elle ne l'était pas déjà et la désactive dans le cas contraire. Une fois activée, l'imprimante recopie scrupuleusement tout ce qui se passe sur l'écran du terminal.

Attention ! les signaux échangés entre la carte CPU09 et l'imprimante étant des signaux de dialogue, le fait de faire une commande Q sans imprimante connectée ou, si celle-ci n'est pas sous tension bloque le système, il faut alors faire un RESET.

Commandes propres à la carte IVG

La carte IVG ayant des possibilités que n'a pas le terminal vidéo de décembre, nous avons prévu les sous-programmes de gestion de celles-ci dans TAVBUG09. Pour l'instant vous ne pouvez pas les activer directement à partir des commandes du clavier car ils ont été prévus pour être pilotés par le DOS ou par vos propres programmes. Sachez qu'il vous sera donc possible d'activer les modes vidéo normale, inversée, demi-teinte, cli-

gnotante, inversée demi-teinte, inversée clignotante, demi-teinte clignotante et enfin inversée demi-teinte clignotante. Pour ce faire, il suffira que votre programme (ou que le DOS) envoie au sous-programme de sortie de caractère TAVBUG09 la séquence adéquate conformément à ce qui est indiqué dans le tableau de la figure 11.

Les sous-programmes de TAVBUG09

L'intérêt d'un moniteur bien conçu ne réside pas seulement dans l'éventail de commandes dont il dispose puisque celles-ci ne servent généralement que pendant la phase de mise au point des programmes, mais aussi dans l'éventail des sous-programmes mis à la disposition des utilisateurs et aussi à leur souplesse d'utilisation.

L'accent a été mis sur ces sous-programmes dans TAVBUG09 et, de plus, tous les programmes de haut niveau qui tournent sur notre ordinateur individuel y font appel, que ce soit le Basic, le DOS ou le Pascal (par exemple) ; une bonne connaissance des plus classiques de ces sous-programmes vous permettra donc toutes les fantaisies. Par ailleurs, lors de l'écriture de vos propres programmes, vous pourrez vous faciliter le travail de façon importante grâce à ceux-ci.

L'appel à ces sous-programmes ne se fait pas, comme dans les moniteurs classiques, par une instruction de saut telle que JMP ou JSR. Nous avons en effet utilisé une technique employée en programmation de haut niveau faisant appel à l'instruction SWI du 6809 ; c'est-à-dire Software Interrupt ou Interruption Logi-

cielle. Cette façon de faire complique un peu le moniteur mais simplifie votre travail pour l'appel des sous-programmes vous intéressant alors...

Le mode d'utilisation est le suivant : soit un programme dans lequel vous avez, à un endroit donné, besoin de faire appel à un sous-programme de TAVBUG09 ; il vous suffit de placer un SWI à cet endroit, suivi par un numéro de code compris entre 0 et 12 et correspondant à la fonction à accomplir, TAVBUG09 fera le reste. Il faut noter que cette façon de procéder est rigoureusement identique au fait de mettre un JSR à une adresse quelconque et que, après avoir exécuté le sous-programme concerné, TAVBUG09 reviendra automatiquement à l'instruction suivant le SWI et son numéro de code comme cela est schématisé figure 12.

Par ailleurs, nous allons voir que les sous-programmes mis à votre disposition sont très agréables d'emploi car, vus de l'extérieur, ils utilisent un minimum de registre, vous évitant ainsi d'avoir à faire des sauvegardes de ceux-ci avant chaque appel. Nous allons détailler maintenant ces sous-programmes un par un, même si cela vous semble un peu prématuré, mais revoyez, si nécessaire ce que nous avons écrit au début de cette présentation de TAVBUG09, et vous apprécierez par la suite de trouver toutes ces informations en un seul et même numéro.

Sous-programme : entrée de caractère

Son numéro de code est 0, c'est-à-dire qu'il est appelé par la fréquence SWI 0. Le caractère frappé au clavier est fourni dans l'accumulateur A débarrassé de

son bit de parité. Le contrôle n'est pas rendu au programme appelant tant qu'un caractère valide n'a pas été frappé. Le caractère NULL (code ASCII 00) et le RUBOUT (code ASCII 7F) sont ignorés. Ce sous-programme appelle automatiquement le sous-programme de sortie de caractère si le contenu de la RAM ECHO EST NUL ; il ne l'appelle pas dans le cas contraire. De plus, la réception d'un retour chariot fait automatiquement sortir un saut ligne par le sous-programme de sortie de caractère. Hormis l'accum A, aucun registre n'est modifié.

Sous-programme de retour au moniteur

Son numéro de code est 8. Si ce sous-programme est appelé avec le contenu de l'accumulateur A nul, il y a réinitialisation des circuits d'entrée/sorties et impression du message TAVBUG09 ; si A n'est pas nul, les entrée/sorties ne sont pas réinitialisées et seul le PROMPT apparaît. Aucun registre n'est conservé par ce sous-programme puisque celui-ci équivaut à une réinitialisation du système.

Sous-programme de sortie de caractère

Numéro de code 1. L'accumulateur A doit contenir le code ASCII du caractère à sortir. Si le caractère n'est pas un saut ligne le sous-programme ne fait rien de particulier ; par contre, si c'est un saut ligne, le bit C du CCR est mis à 0 si la sortie s'est passée normalement et à 1 si un CNTRL X a été frappé durant cette sortie. De plus, les nom-

bres de caractères nuls définis par la commande N de TAVBUG09 sont respectés par ce sous-programme, de même que le fait de frapper une touche pendant une sortie de caractère.

Hormis le CC dans le cas d'un saut ligne, aucun registre n'est modifié par ce sous-programme.

Sous-programme de sortie de deux chiffres hexadécimaux

Numéro de code 4. Ce sous-programme convertit un mot de 8 bits pointé par le registre d'index X en deux chiffres hexadécimaux qui sont envoyés au terminal suivis d'un espace. Aucun registre n'est modifié.

Sous-programme de sortie de quatre chiffres hexadécimaux

Numéro de code 5. Ce sous-programme convertit un mot de 16 bits pointé par l'index X en quatre chiffres hexadécimaux qui sont envoyés au terminal suivis par un espace. Aucun registre n'est modifié.

Sous-programme de sortie d'un saut ligne retour chariot

Numéro de code 6. Ce sous-programme fait sortir sur la console un retour chariot suivi par un saut ligne. Comme pour le sous-programme de sortie de caractère, le fait de frapper un caractère pendant l'exécution de ce sous-programme suspend la sor-

Séquence à envoyer	Etat de la carte IVG
1B 30	vidéo normale
1B 31	vidéo normale clignotante
1B 32	vidéo inversée
1B 33	vidéo inversée clignotante
1B 34	vidéo demi-teinte
1B 35	vidéo demi-teinte clignotante
1B 36	vidéo demi-teinte inversée
1B 37	vidéo demi-teinte inversée clignotante

Fig. 11. - Séquences des caractères de contrôle de la carte IVG.

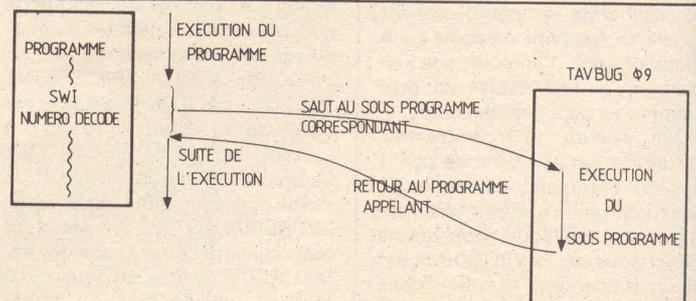


Fig. 12. - Principe de l'appel des sous-programmes par un SWI.

tie jusqu'à la frappe d'un nouveau caractère (comme pour les commandes de TAVBUG09) ; les valeurs définies par la commande N sont respectées et le bit C du CCR est mis à 0 si tout s'est bien passé et à 1 si un CNTRL X a été frappé pendant la sortie. Seul le CCR est éventuellement modifié.

Sous-programme de sortie d'une chaîne de caractères

Numéro de code 2. Ce sous-programme fait sortir sur le terminal la chaîne de caractères dont le premier est pointé par X

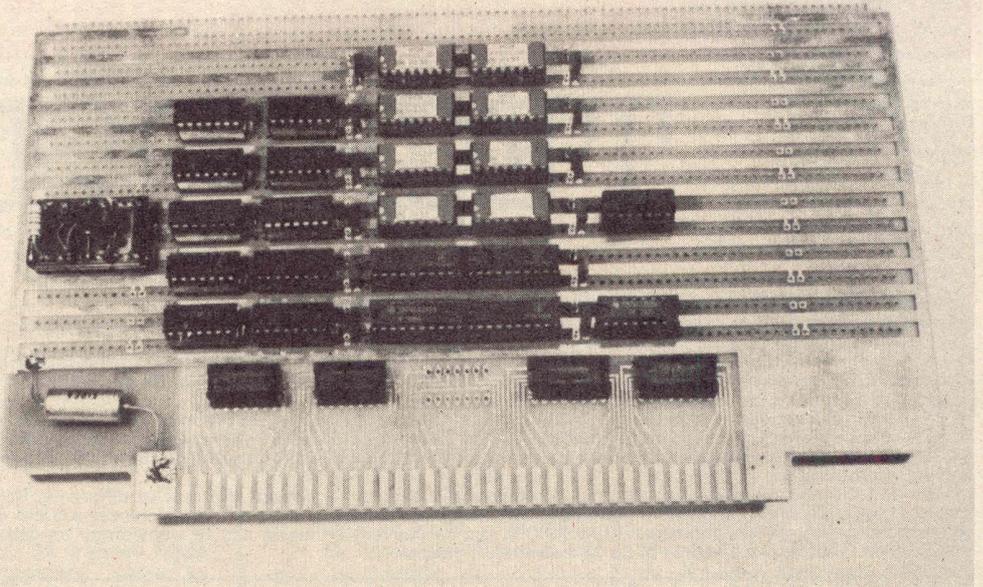


Photo 2. - Gros plan sur la version expérimentale de la carte RAM dynamique.

N° de code	Appellation	Fonction
0	INCH	entrée de caractère depuis le terminal
1	OUTCH	sortie de caractère sur le terminal
2	PDATA1	sortie d'une chaîne de caractères
3	PDATA	sortie d'un saut ligne-retour chariot suivi d'une chaîne de caractères
4	OUT2HS	sortie de 2 chiffres hexa et d'un espace
5	OUT4HS	sortie de 4 chiffres hexa et d'un espace
6	PCRLF	sortie d'un saut ligne-retour chariot
7	OUTS	sortie d'un espace
8	MON ITR	retour au moniteur

Fig. 13. - Tableau des sous-programmes standards de TAVBUG09.

Commande	Fonction
A	transfert mémoire
B	gestion des points d'arrêt
C	appel d'un sous-programme utilisateur
D	visualisation d'une zone de mémoire
E	codage des adressages indexés
G	lancement d'un programme utilisateur
I	initialisation mémoire
L	chargement d'une cassette en mémoire
M	examen/modification mémoire
N	adaptation au terminal
O	calcul des déplacements
P	enregistrement d'une cassette
Q	activation de l'imprimante
R	examen/modification des registres
S	inhibition automatique du pas à pas
T	fonctionnement en pas à pas
U	changement de page mémoire
V	vérification d'une cassette
X	chargement du DOS à froid
Y	lancement du DOS depuis TAVBG09
W	définition d'une fenêtre
/	examen mémoire rapide pas à pas rapide

Fig. 14. - Résumé des commandes de TAVBUG09.

et dont le dernier doit être impérativement un EOT (code ASCII 04). Le comportement de ce sous-programme est identique à celui de sortie du saut ligne retour chariot. X et éventuellement le CCR sont modifiés.

Sous-programme de sortie d'un retour chariot saut ligne suivi d'une chaîne de caractères

Numéro de code 3. Ce sous-programme correspond à l'enchaînement des deux sous-programmes précédents ; la sortie du retour chariot - saut ligne ayant lieu avant la chaîne de caractères. Comportement et contraintes identiques à ceux des sous-programmes précédents.

Sous-programme de sortie d'un espace

Numéro de code 7. Ce sous-programme fait sortir un espace sur le terminal ; il répond aux mêmes contraintes et critères que le sous-programme de sortie de caractère vu ci-avant.

Autres sous-programmes

Une vingtaine d'autres sous-programmes, moins classiques sont disponibles ; nous n'en par-

lons cependant pas maintenant car ils font appel à des connaissances matérielles du système qu'il nous est impossible d'exposer ici. De plus, ces sous-programmes sont surtout utilisés avec le DOS ou des cartes d'entrées/sorties spécialisées et seront donc logiquement présentés à ce moment-là.

Conclusion

Afin que cette présentation de TAVBUG09 soit profitable, nous pensons qu'il vous faut essayer quelques commandes parmi celles que vous pouvez exploiter compte tenu de vos connaissances actuelles du 6809 et du système. Nous vous rappelons ce que nous avons dit en guise d'avertissement au début de cette présentation, TAVBUG09 est le moniteur du système, c'est-à-dire qu'il gère le fonctionnement de l'ordinateur au niveau le plus bas. Si vous voulez utiliser votre ordinateur comme une boîte noire destinée à faire du Basic ou du Pascal ou de la gestion, vous pouvez, à la limite, ignorer TAVBUG09, les seules commandes que vous aurez à en connaître étant le X et le Y pour charger le DOS.

Le mois prochain nous franchirons un grand pas en avant avec la RAM dynamique. Nous décrirons la carte 256 K-octets mais nous expliquerons aussi comment utiliser la carte RAM dynamique de l'ancien système sur cet ordinateur.

à suivre.
C. TAVERNIER