

Réalisez votre ordinateur individuel

CARTE CGC 09

ET SON LOGICIEL

Chose promise, chose due ; nous vous avons dit que cet article serait consacré à la description de la réalisation de la carte alphanumérique et graphique couleur basse résolution et à son logiciel, et c'est le cas, comme vous avez pu vous en rendre compte si vous avez déjà rapidement parcouru du regard ces quelques pages. Les listings vous ont peut-être impressionnés ? Il n'y a vraiment pas de quoi, car ils sont largement commentés dans les lignes qui suivent. Mais commençons par le commencement avec...

Réalisation de la carte

Muni du circuit imprimé, du schéma théorique publié dans notre précédent numéro et de la figure 1 qui représente le plan d'implantation des composants, vous allez pouvoir entreprendre le montage qui, s'il n'est pas plus compliqué que pour les autres cartes, demande un peu de soin vu le nombre important de composants passifs.

Mettez en place, en premier lieu, les supports de circuits intégrés et le connecteur vingt contacts de haut de carte si vous l'utilisez, ce qui n'a rien d'obligatoire, comme expliqué ci-après. Vous allez ensuite devoir câbler les résistances. Elles sont assez nombreuses et, bien que leur mise en place ne prête pas à confusion, respectez bien le plan d'implantation de la figure 1. En cas de doute sur un trou, contrôlez votre geste avec un ohmmètre et le schéma théorique : nous avons en effet vu plusieurs

cartes en panne à cause d'un composant dont une des pattes était mise dans un trou voisin du bon. Prenez l'habitude d'orienter toutes les résistances dans le même sens, cela facilite une rapide vérification des valeurs.

Si vous utilisez des résistances 1/2 W et que vous avez du mal à les placer côte à côte en certains endroits, utilisez une disposition analogue au schéma de la figure 2 ; c'est à la fois propre et pratique.

Lorsque ces composants sont câblés, mettez en place les condensateurs de découplage de 22 nF (C sur la figure 1), puis les chimiques, en respectant bien le sens. Le 22 μ F situé à gauche de T₁ a bien le pôle plus à la masse ; ce n'est pas une erreur de dessin !

Les potentiomètres seront montés ensuite ainsi que les mini-interrupteurs S₁ et S₂. Si vous comptez mettre des straps soudés pour ces derniers, laissez des emplacements vides pour l'instant ; vous déciderez de la mise en

place des straps plus tard, lors des essais. Le quartz est monté à plat sur la carte où il est peut-être maintenu par un fil nu soudé dans les deux trous qui se trouvent de part et d'autre de l'emplacement de son boîtier. Ce dernier peut être plaqué sans hésiter sur le circuit imprimé, la surface cuivrée qui se trouve en dessous étant la masse.

Montez ensuite les transistors en faisant attention au sens, encore qu'une confusion soit assez difficile à faire, mais aussi à la polarité de ces derniers ; en effet : T₁ et T₅ sont des NPN alors que T₂, T₃ et T₄ sont des PNP ; une erreur ne serait pas fatale, mais empêcherait la carte de fonctionner.

Lorsque tous les composants à souder sont en place, que les soudures sont vérifiées et qu'un contrôle à l'ohmmètre vous a assuré de l'absence de courts-circuits entre pistes voisines, vous pouvez placer les circuits intégrés sur leurs supports respectifs en faisant, bien sûr, attention au sens, et vous pouvez vous préparer pour la mise à feu avec le câblage de la prise péritélévision.

La prise péritélévision

Ainsi que nous l'avons expliqué dans notre précédent numéro, les signaux délivrés par notre carte doivent être appliqués à une telle prise si vous souhaitez obtenir une image couleur sur votre récep-

teur TV. Nous avons suivi en cela l'exemple de tous les fabricants de micro-ordinateurs, ce qui n'a pas été bien difficile puisque tout le monde se heurte au même écueil : la fabrication d'un codeur SECAM.

Le choix de la prise péritélévision ne présente que des avantages : image bien meilleure que si l'on passe par un codeur et l'entrée antenne, carte beaucoup plus simple, absence totale de réglage, etc. Son seul défaut est que cette prise n'est obligatoire que depuis fin 1980 et que, de ce fait, tous les récepteurs couleurs n'en sont pas équipés ; nous allons donc voir comment traiter ce problème.

Cinq cas peuvent se présenter et nous allons les analyser successivement :

— Vous disposez d'un moniteur noir et blanc, et vous souhaitez utiliser la carte avec celui-ci, bénéficiant ainsi de huit niveaux de gris au lieu des huit couleurs annoncées. Vous n'aurez aucun problème de connexion puisqu'une sortie vidéo composite existe sur notre carte. Il vous suffira de fermer S₂ et de connecter l'entrée de votre moniteur sur « composite-synchro » repérée figure 1, ou figure 5 si vous passez par le connecteur vingt contacts.

— Vous disposez d'un moniteur couleur, auquel cas vous ne devriez pas avoir de problème de connexion, sous réserve que celui-ci dispose bien de trois entrées vidéo séparées pour chacune des composan-

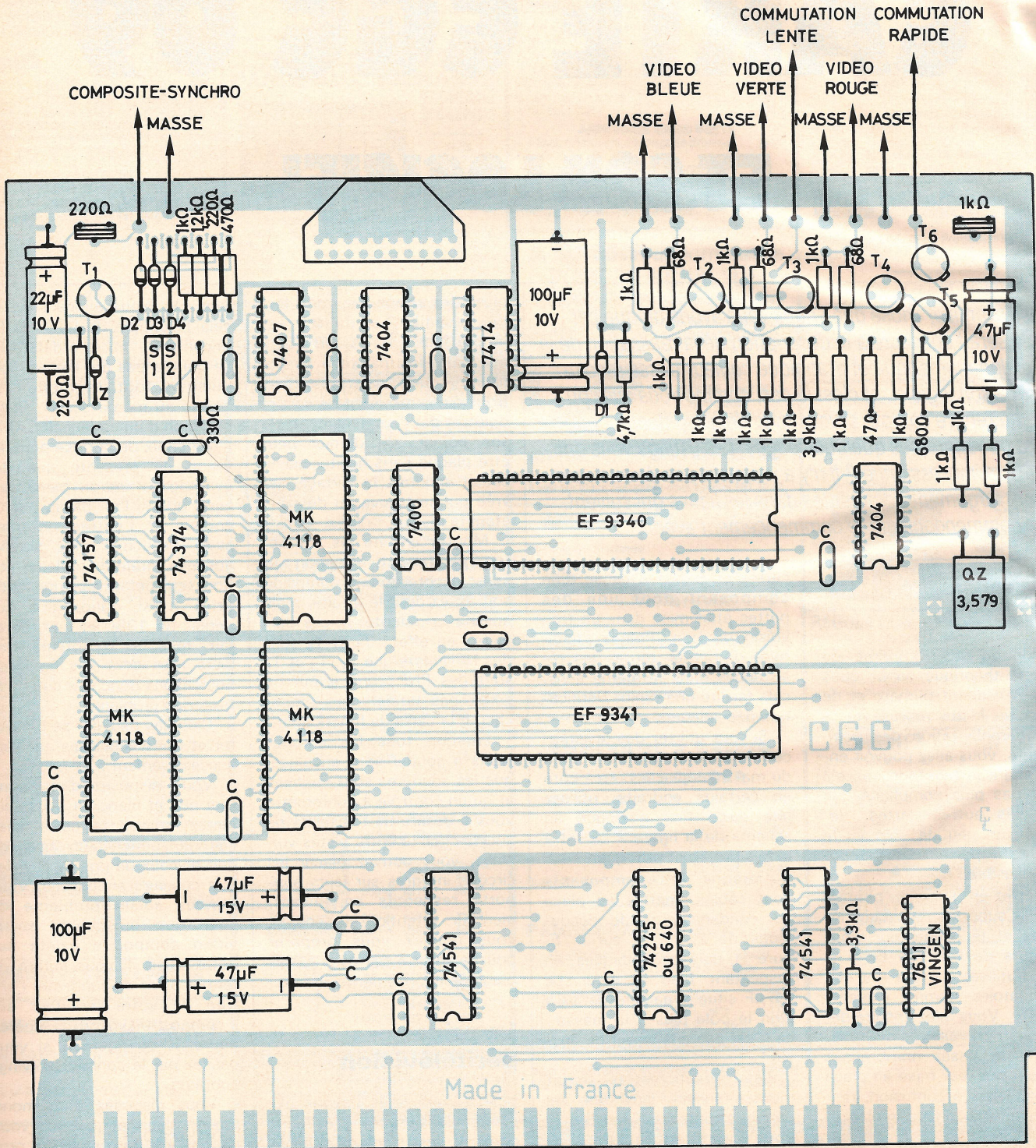
tes rouge, verte et bleue. Si tel est le cas, les entrées des trois composantes de votre moniteur seront reliées aux sorties correspondantes de la carte et l'entrée de synchronisation de

votre moniteur sera reliée à la sortie « composite-synchro » de la carte. L'interrupteur S₂ sera, cette fois-ci, ouvert.

— Vous disposez d'un récepteur TV noir et blanc non muni

d'une prise péritelvision. Le problème est un peu plus compliqué ; il va en effet falloir que vous vous procuriez un modulateur UHF noir et blanc, disponible chez la majorité des re-

vendeurs de composants, et que vous modifiez celui-ci par le signal « composite-synchro ». Le modulateur sera relié à l'entrée antenne UHF de votre récepteur qui sera ac-



C = 22nF DECOUPLAGE

Fig. 1. — Implantation des composants.

cordé aux e
c'est géné
fréquence
les modula
merce). L
dulateur s
système, l
ces appare
Comme da
teur noir e
S₂ sera fer
— Vous d
teur TV co
prise périt
peuvent a
vous voule
blanc ou v
en couleur
vailler en
êtes rame
et devez
venons d
voulez tra
ne vous r
qui passe
codeur S
UHF, don
tuellement
le marché
que CGV
Match. U
pose d'un
ritélévisio
allez racc
de notre
après ; et
côté d'un
SECAM c
connecté
classique.
néanmoins
puisque
que ce r
500 F. Pr
ponible ch
revendeur
informati
blème, de
revendeur
tional Vid
Alexand
Strasbourg
cas du m
avant, S₂
pouvoir tr
— Enfin, c
posez d'u
leur muni
sion. Dan
cas précé
le boîtier
dulateur U
câble mur
vision mât
La figu
brochage
lévision v

cordé aux environs du canal 36 (c'est généralement sur cette fréquence que sont pré-réglés les modulateurs UHF du commerce). L'alimentation du modulateur sera prélevée sur le système, la consommation de ces appareils étant très faible. Comme dans le cas du moniteur noir et blanc vu ci-avant, S₂ sera fermé.

— Vous disposez d'un récepteur TV couleur non muni de la prise péritélévision. Deux cas peuvent alors se présenter : vous voulez travailler en noir et blanc ou vous voulez travailler en couleur. Si vous voulez travailler en noir et blanc, vous êtes ramené au cas précédent et devez faire ce que nous venons d'expliquer. Si vous voulez travailler en couleur, il ne vous reste qu'une solution qui passe par l'acquisition d'un codeur SECAM-modulateur UHF, dont un modèle est actuellement commercialisé sur le marché français sous la marque CGV par la société Vidéo Match. Un tel ensemble dispose d'un côté d'une prise péritélévision dans laquelle vous allez raccorder le cordon issu de notre carte et décrit ci-après ; et il dispose de l'autre côté d'une sortie UHF codée SECAM qui peut donc être connectée à tout récepteur TV classique. Cette solution est néanmoins assez coûteuse puisque nous croyons savoir que ce module vaut environ 600 F. Précisons qu'il est disponible chez de très nombreux revendeurs de matériel micro-informatique ; en cas de problème, demandez la liste des revendeurs au distributeur national Vidéo Match, 8-10, rue Alexandre-Dumas, 67200 Strasbourg. Comme dans le cas du moniteur couleur vu ci-avant, S₂ sera ouvert afin de pouvoir travailler en couleur.

— Enfin, dernier cas, vous disposez d'un récepteur TV couleur muni de la prise péritélévision. Dans ce cas, et dans le cas précédent si vous utilisez le boîtier codeur SECAM-modulateur UHF, il faut réaliser un câble muni d'une prise péritélévision mâle.

La figure 4 vous indique le brochage d'une embase péritélévision vue côté câblage ou,

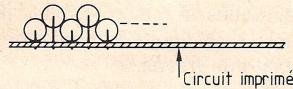


Fig. 2. — Comment monter proprement des résistances trop grosses.

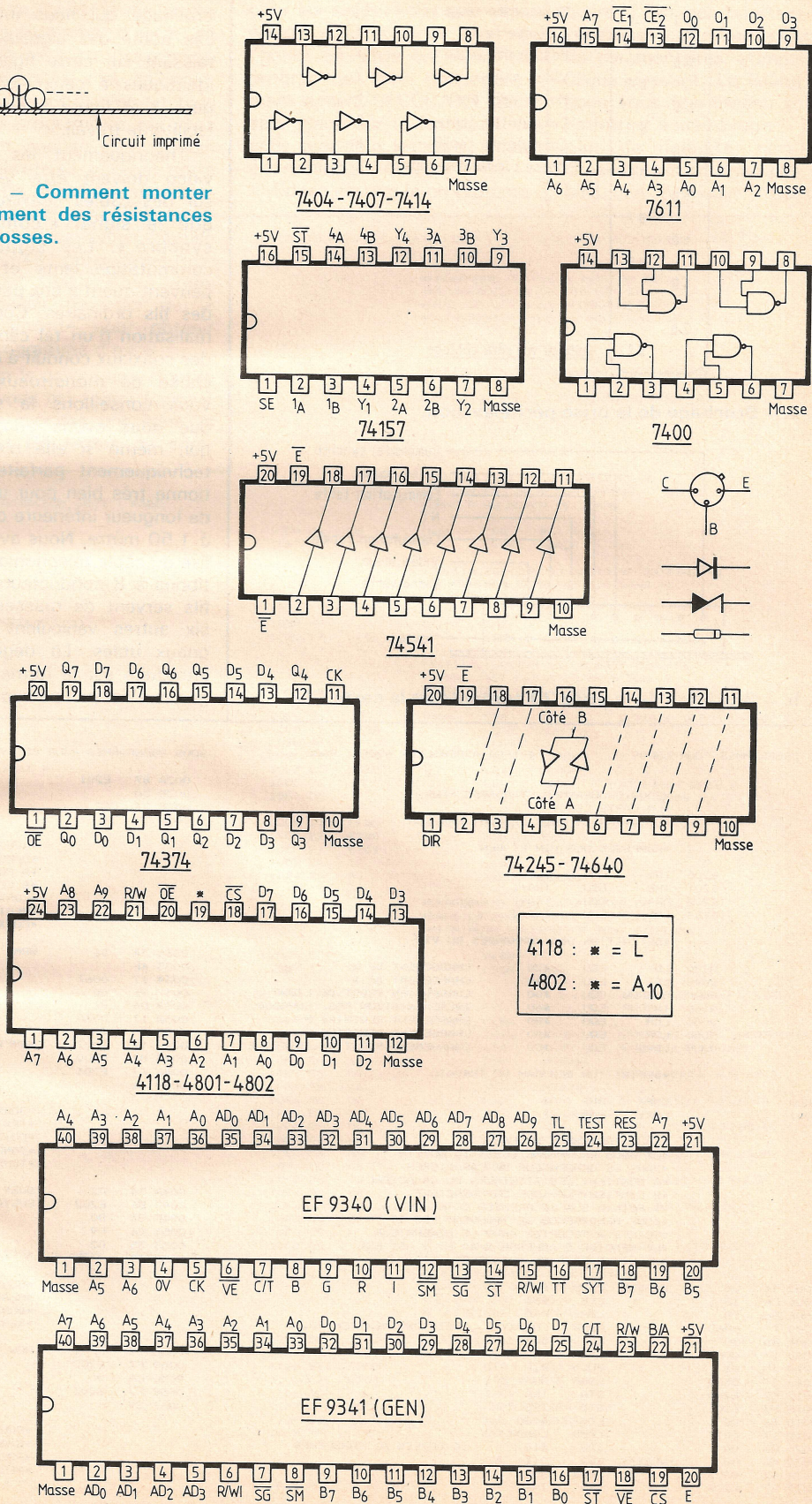


Fig. 3. — Brochages des semiconducteurs.

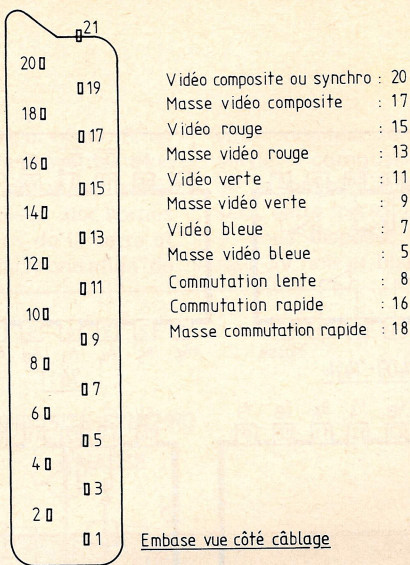


Fig. 4. — Brochage de la prise péritélévision.

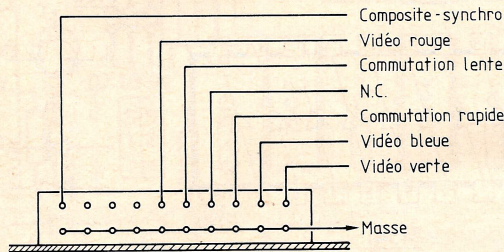


Fig. 5. — Brochage de la prise 20 contacts de la carte CGC.

plus exactement, la partie du brochage qui nous intéresse. Les noms des signaux apparaissant sur cette figure sont identiques à ceux visibles figure 1 et figure 5 pour vous faciliter le travail.

Théoriquement, les signaux vidéo doivent être véhiculés sur des câbles coaxiaux ainsi que le signal « composite-synchro ». Les signaux de commutation lente et rapide peuvent quant à eux passer sur des fils ordinaires. Comme la réalisation d'un tel câble avec des coaxiaux conduit à quelque chose de monstrueux, nous vous conseillons la solution que nous avons adoptée et qui, même si elle n'est pas techniquement parfaite, fonctionne très bien pour un câble de longueur inférieure ou égale à 1,50 mètre. Nous avons utilisé du câble téléphonique multibrins à 8 conducteurs ; deux fils servent de masses et les six autres véhiculent les signaux utiles. La dégradation introduite sur la forme des signaux n'est pas perceptible sur

un récepteur TV couramment ainsi réalisé discret et souple.

Ce câble peut être directement sur les plots de cordement de la carte ou peut être relié au tout vingt contacts dont le brochage est indiqué sur la figure 5. Dans le premier cas, le signal sortira du boîtier par un câble à deux fils alors que, dans le second cas, il sera nécessaire de prévoir un connecteur « DIN » à la face arrière. Sur le boîtier, un dec préconisé au chapitre 2 de cette réalisation, nous avons prévu quatre trous pour les fiches BNC, un pour chaque couleur fondamentale pour les signaux de commutation lente et rapide n'étaient pas envisagés lors de la conception du boîtier. Nous vous conseillons donc de munir un boîtier pour prise DIN d'une carte à 7 broches (cela existe dans la gamme standard) qui permettra de très bien pour la commutation.

SOUS PROGRAMMES POUR CGC09

4-1-84 ASSEMBLEUR 6809 PAGE 1

SOUS PROGRAMMES POUR CGC09

4-1-84 ASSEMBLEUR 6809

*SDUS PROGRAMMES STANDARDS POUR
*LA CARTE CGC09
*C. TAVERNIER POUR LE HAUT PARLEUR

*ADRESSES DES VIN ET GEN

E200 TRA EQU #E200
E201 TRB EQU TRA+1
E202 CRA EQU TRA+2
E203 CRB EQU TRA+3

*DEFINITION DES COMMANDES DU VIN

0020 LDY EQU #20 CHARGEMENT DE Y
0040 LDX EQU #40 CHARGEMENT DE X
0000 BEGROW EQU #00 CURSEUR EN DEBUT DE LIGNE
0060 INCCUR EQU #60 INCREMENTATION POS. CURSEUR
0080 LDM EQU #80 CHARGEMENT REGISTRE M
00A0 LDR EQU #A0 CHARGEMENT REGISTRE R
00C0 LDY0 EQU #C0 CHARGEMENT POSITION INITIALE

*DEFINITION DES RAM DE TRAVAIL

0000 CURX RMB 1
0001 CURY RMB 1

*SOUS PROGRAMME WRGENM

*CE SOUS PROGRAMME ECRIT 10 OCTETS
*DANS LE GENERATEUR DE CARACTERES
*A CONTIENT LES ATTRIBUTS DU CARACTERE
*B CONTIENT LE CODE DU CARACTERE
*X POINTE SUR LE PREMIER OCTET DE LA
*ZONE MEMOIRE OU SE TROUVENT LES 10
*OCTETS A RECOPIER DANS LE GENERATEUR
*LA MEMOIRE D'ADRESSE X=40 ET Y =31 EST
*UTILISEE POUR CE TRANSFERT

0002 34 06 WRGENM PSHS A,B
0004 C6 1F LDB #31
0006 86 2B LDA #40
>0008 17 006D LBSR LOADXY POSITIONNEMENT EN 40,31
000B 86 40 LDA #40
000D 17 00BE LBSR LOADM
0010 35 06 PULS A,B
>0012 17 0035 LBSR BUSY
0015 B7 E200 STA TRA
0018 F7 E201 STB TRB
001B 86 80 LDA ##80
>001D 17 007E LBSR LOADM
0020 C6 0A LDB #10
0022 A6 80 LDA 0,X+
>0024 17 0023 LBSR BUSY
0027 B7 E200 STA TRA

002A B7 E201 STA TRB
002D 5A DECB
002E 26 F2 BNE WRGENO
0030 39 RTS

*SOUS PROGRAMME D'ECRITURE D'UN CARACTERE

*DANS LA PAGE VISUALISEE
*A CONTIENT LES ATTRIBUTS DU CARACTERE
*B CONTIENT LE CODE DU CARACTERE
*CURX ET CURY L'ADRESSE DE LA POSITION
*REGISTRE MODIFIE AUCUN
*CURX ET CURY NON MODIFIES

0031 34 06 WRMP PSHS A,B
0033 4F CLR A
>0034 17 0067 LBSR LOADM
0037 96 00 LDA CURX
0039 D6 01 LDB CURY
>003B 17 003A LBSR LOADXY
003E 35 06 PULS A,B
>0040 17 0007 WRMP1 LBSR BUSY
0043 B7 E200 STA TRA
0046 F7 E201 STB TRB
0049 39 RTS

*SOUS PROGRAMME BUSY
*TESTE LE BIT 7 DU CRA
*DOIT ETRE APPELE AVANT TOUT TRANSFERT
*DANS CRA CRB OU TRA TRB
*TOUS LES REGISTRES SONT SAUVEGARDES

004A 34 02 BUSY PSHS A
004C B6 E202 BUSY0 LDA CRA
004F B4 80 ANDA ##80
0051 26 F9 BNE BUSY0
0053 35 02 PULS A
0055 39 RTS

*SOUS PROGRAMME ROW
*PLACE LE CURSEUR AU DEBUT DE LA LIGNE
*DONT LE NUMERO EST CONTENU DANS B
*REGISTRE MODIFIE : B

>0056 17 FFF1 ROW LBSR BUSY
0059 F7 E202 STB CRA
005C C6 00 LDB #BEGROW
005E F7 E203 STB CRB
0061 39 RTS

*SOUS PROGRAMME LOADX
*CHARGE LE REGISTRE X DU CURSEUR
*AVEC LE CONTENU DE A
*REGISTRE MODIFIE : A

Premiers essais

Comme toute carte utilisant un circuit d'interface évolué, et encore plus ici où la carte comporte son propre processeur (l'EF 9340), il est impossible de faire le moindre essai sans logiciel. Nous allons donc vous donner le listing d'un petit programme d'essai et nous commenterons ensuite celui-ci lors de la description de la programmation des circuits EF 9340 et EF 9341. Avant de vous précipiter sur votre éditeur de textes, positionnez les potentiomètres de la carte à mi-course, ouvrez S₂ si vous travaillez en couleur et ouvrez S₁ pour l'instant.

Pour ne pas faire du travail en double, nous avons conçu notre programme d'essai en deux parties : un programme principal appelé « jeu de caractères carte CGC » et un bloc de sous-programmes qui vous serviront ensuite dans toutes les applications utilisant cette carte. Ce bloc de sous-programmes est rangé dans un fichier que nous avons baptisé SPCGC.TXT pour Sous-Programmes CGC. Vous allez donc devoir procéder de la façon suivante pour obtenir votre programme d'essai :

— Editez le programme dont le listing vous est fourni figure 6 sans en modifier une ligne, si ce n'est les textes de com-

mentaires qui ne sont là que pour les explications que nous allons vous donner tout à l'heure. Le fichier contenant ce programme sera obligatoirement appelé SPCGC.TXT et il sera placé sur le lecteur 1 si vous avez deux lecteurs ; sinon il sera bien sûr en 0.
— Editez ensuite le programme

dont le listing vous est indiqué figure 7 en ne modifiant aucune ligne autre que les lignes de commentaires, comme expliqué ci-avant. Le listing que nous vous fournissons est un listing d'assemblage ; pour plus de lisibilité, certaines choses n'y sont donc pas visibles. En particulier, sous la

SOUS PROGRAMMES POUR CGC09 4-1-84 ASSEMBLEUR 6809 PAGE 3

```
0062 8D E6 LOADX BSR BUSY
0064 B7 E202 STA CRA
0067 86 40 LDA #LDR
0069 B7 E203 STA CRB
006C 39 RTS
```

*SOUS PROGRAMME LOADY
*CHARGE LE REGISTRE Y DU CURSEUR
*AVEC LE CONTENU DE A
*REGISTRE MODIFIE : A

```
006D 8D DB LOADY BSR BUSY
006F B7 E202 STA CRA
0072 86 20 LDA #LDY
0074 B7 E203 STA CRB
0077 39 RTS
```

*SOUS PROGRAMME LOADXY
*POSITIONNE LE CURSEUR EN X ET Y
*X EST CONTENU DANS A
*Y EST CONTENU DANS B
*REGISTRE MODIFIE : A

```
0078 8D EB LOADXY BSR LOADX
007A 1F 984D TBA
007D 8D EE BSR LOADY
007F 39 RTS
```

*SOUS PROGRAMME LOADYO
*CHARGE LE REGISTRE YO
*PAR LE CONTENU DE A
*REGISTRE MODIFIE : AUCUN

```
0080 34 02 LOADYO PSHS A
0082 8D C6 BSR BUSY
0084 B7 E202 STA CRA
0087 86 C0 LDA #LDYO
0089 B7 E203 STA CRB
008C 35 02 PULS A
008E 39 RTS
```

*SOUS PROGRAMME LOADR
*CHARGE LE REGISTRE R
*AVEC LE CONTENU DE A
*REGISTRE MODIFIE : AUCUN

```
008F 34 02 LOADR PSHS A
0091 8D B7 BSR BUSY
0093 B7 E202 STA CRA
0096 86 40 LDA #LDR
0098 B7 E203 STA CRB
009B 35 02 PULS A
009D 39 RTS
```

SOUS PROGRAMMES POUR CGC09 4-1-84 ASSEMBLEUR 6809 PAGE 4

*SOUS PROGRAMME LOADM
*CHARGE LE REGISTRE M
*AVEC LE CONTENU DE A
*REGISTRE MODIFIE : AUCUN

```
009E 34 02 LOADM PSHS A
00A0 17 FFA7 LBSR BUSY
00A3 B7 E202 STA CRA
00A6 86 80 LDA #LDM
00AB B7 E203 STA CRB
00AB 35 02 PULS A
00AD 39 RTS
```

END

0 ERREUR(S) DETECTEE(S)

JEU DE CARACTERES CARTE CGC 4-1-84 ASSEMBLEUR 6809 PAGE 1

```
*PROGRAMME DE VISUALISATION DU JEU
*DE CARACTERES DU CIRCUIT GEN
*UTILISE LES SOUS PROGRAMMES SPCGC
*C.TAVERNIER POUR LE HAUT PARLEUR

1000 ORG $1000

*DEFINITION DES CONSTANTES ET DES RAM
*APPEL DE SPCGC.TXT AU MOYEN D'UN LIB
OPT LIS

*DEBUT DU PROGRAMME

*INITIALISATION DU REGISTRE R
*LE CONTENU DE A EST CHARGE
*DANS R AU MOYEN DE LOADR
10B0 86 CD INIR LDA #Z11001101
10B2 17 FFDC LBSR LOADR

*INITIALISATION DU REGISTRE ORIGINE YO
*LE CONTENU DE A EST MIS DANS YO
*AU MOYEN DE LOADYO
10B5 4F INIYO CLRA
10B6 17 FFC9 LBSR LOADYO

*INITIALISATION DE LA PAGE MEMOIRE
*VISUALISEE LES ATTRIBUS DES
*CARACTERES SONT DANS B , LE CODE DANS A
10B9 86 1F INIM LDA #S1 CURSEUR DEBUT LIGNE SERVICE
10BB 17 FF9A LBSR ROW
10BE 4F CLRA
10BF 17 FFDE LBSR LOADM
10C2 8E 03EB LDX #1000
10C5 86 88 LDA #S8 ATTRIBUT DANS A
10C7 C6 7F LDB #S7F CODE DANS B

10C9 17 FF80 BOUCLE LBSR BUSY
10CC B7 E200 STA TRA
10CF F7 E201 STB TRB
10D2 30 1F LEAX -1,X
10D4 26 F3 BNE BOUCLE

*CHARGEMENT DES CARACTERES DU GEN
*DANS LA PAGE VISUALISEE , LES
*ATTRIBUS SUCCESSIFS SONT DANS A
10D6 5F CLRB
10D7 17 FF7E LBSR ROW
```

JEU DE CARACTERES CARTE CGC 4-1-84 ASSEMBLEUR 6809 PAGE 2

```
10DA 4F CLRA
10DB 17 FFC2 LBSR LOADM ECRITURE AVEC INCREMENTATION
10DE 86 0F LDA #S0F ALPHA BLANC/NOIR
10E0 8D 1A BSR JEUCAR
10E2 86 09 LDA #S09 ALPHA ROUGE SUR NOIR
10E4 8D 16 BSR JEUCAR
10E6 86 0A LDA #S0A ALPHA VERT SUR NOIR
10E8 8D 12 BSR JEUCAR
10EA 86 0C LDA #S0C ALPHA BLEU SUR NOIR
10EC 8D 0E BSR JEUCAR
10EE 86 4B LDA #S4B ALPHA NOIR SUR JAUNE
10F0 8D 0A BSR JEUCAR
10F2 86 9B LDA #S9B GRA JAUNE SUR ROUGE
10F4 8D 06 BSR JEUCAR
10F6 86 CA LDA #S1CA GRA VERT SUR BLEU
10F8 8D 02 BSR JEUCAR
10FA 3F SWI
10FB 08 FCB 8 RETOUR A TAVBUG09
```

*SOUS PROGRAMME D'AFFICHAGE
*DU JEU DE CARACTERES
*LE CODE EST DANS B
*L'ATTRIBUT EST DANS A

```
10FC 5F JEUCAR CLRB
10FD 17 FFA7 JEUCAR LBSR BUSY
1100 B7 E200 STA TRA
1103 F7 E201 STB TRB
1104 5C INCB
1107 C1 80 CMPB #S80
1109 26 F2 BNE JEUCAR
110B 39 RTS
```

END

0 ERREUR(S) DETECTEE(S)

Fig. 7. — Listing du programme jeu de caractères carte CGC.

ligne APPEL DE SPCGC, il faut placer un LIB 1.SPCGC.TXT pour appeler les sous-programmes que vous avez édités ci-avant (revoyez la directive LIB de l'assembleur si nécessaire). Si votre fichier SPCGC.TXT est sur le lecteur 0, le LIB devient évidemment LIB 0.SPCGC.TXT. Ne faites pas attention à la directive OPT LIS qui est inutile pour l'instant.

— Assemblez alors ce programme avec les options L pour en avoir un listing complet et B pour créer un fichier binaire. Le listing que vous obtiendrez sera identique à celui de la figure 7 mais, en plus, vous disposerez, entre la ligne APPEL... etc., et la ligne DEBUT DU PROGRAMME, du listing de la figure 6. C'est normal puisque c'est le résultat de la directive LIB. Nous verrons plus tard comment faire disparaître cette partie qui, sinon, se retrouve identique à elle-même sur tous les listings, ce qui est inutile lorsqu'on l'a vue une fois.

Vous pouvez alors mettre en place la carte CGC dans le système, charger le DOS puis le programme que vous venez d'assembler et constater que, tant que vous ne lancez pas le programme, l'écran associé à cette carte est quelconque. Généralement l'on y voit une image non stabilisée horizontalement ne contenant que très-peu de caractères ; c'est normal tant que le VIN EF 9340 n'est pas initialisé.

Lancez alors votre programme par un G à l'adresse 10B0 ; une image doit apparaître et vous présenter :

- le jeu de caractères alpha-numériques complet en blanc sur fond noir ;
- le même jeu en rouge sur fond noir ;
- le même jeu en bleu sur fond noir ;
- le même jeu en vidéo inversée noire sur fond jaune ;
- le jeu graphique complet en jaune sur fond rouge ;

— le jeu graphique complet en vert sur fond bleu ;

Il se peut fort bien que, même si votre carte est en état de marche, rien ne soit visible sur l'écran ; en effet, il faut régler les potentiomètres de synchro et de couleurs, et la position correcte de ce dernier est assez critique. Commencez par tourner le potentiomètre de couleurs (le 1 kΩ) pour voir apparaître les caractères. Le point le meilleur se situe juste après l'apparition des caractères dans toutes les couleurs et avant que ceux-ci ne commencent à baver par excès de brillance. Il est conseillé d'agir simultanément sur ce réglage et sur ceux de contraste et de luminosité du récepteur TV pour obtenir la meilleure image possible. Vous devez arriver à une qualité irréprochable. Il faut ensuite stabiliser l'image au moyen du potentiomètre de niveau de synchronisation de 220 Ω. Cette stabilisation doit être atteinte pour une très large plage de rotation de ce

potentiomètre, compte tenu du schéma utilisé à son niveau. Le curseur sera placé au milieu de cette plage, encore que sa position exacte n'ait pas une grande importance.

Si vous travaillez en noir et blanc, le réglage exact du potentiomètre de 1 kΩ détermine les divers niveaux de gris, mais le réglage de synchronisation intervient aussi car, dans ce cas, il ne règle plus le niveau de synchro seul mais également celui de la vidéo composite. Si vous trouvez que ces niveaux ne sont pas assez échelonnés, vous pouvez retoucher les résistances de 1,2 kΩ, 470 Ω et 220 Ω placées entre les sorties des 7407 de sommation des signaux vidéo et le + 5 V. Ces résistances sont dans un rapport correspondant approximativement à l'équation colorimétrique qui indique comment doser les trois couleurs rouge, verte et bleue pour faire une image noir et blanc, mais ce rapport n'est pas impératif et vous pouvez donc ajuster ces valeurs en fonction de vos goûts et, surtout, des possibilités de votre téléviseur noir et blanc.

En cas de non fonctionnement, plusieurs cas sont à distinguer :

— Si aucune image n'est visible à l'écran, quelle que soit la position des potentiomètres et que le logiciel ait « tourné » ou non, il faut penser à un problème au niveau de l'horloge à quartz ou au niveau de la cir-

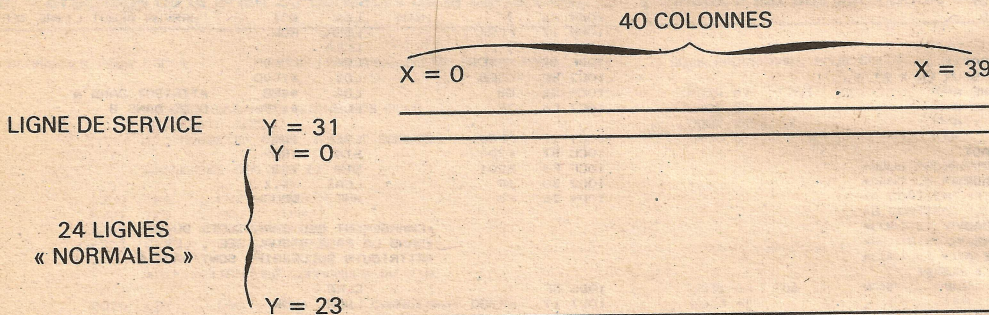


Fig. 8. — Organisation de l'écran sous contrôle de la carte CGC 09

CRB							CRA							NOM	FONCTION		
B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀		
0	0	0									K	K	K	K	K	BEGIN ROW	0 → X K → Y
0	0	1									K	K	K	K	K	LOADY	K → Y
0	1	0								K	K	K	K	K	K	LOAD X	K → X
0	1	1														INCC	C + 1 → C C = Curseur
1	0	0						K	K	K	K	K	K	K	K	LOAD M	K → M
1	0	1						K	K	K	K	K	K	K	K	LOAD R	K → R
1	1	0								K	K	K	K	K	K	LOAD Y0	K → Y0

Fig. 9. — Codes des commandes comprises par le VIN

cuiterie raccordée sur les sorties R, G, B, TT, TL et I de l'EF 9340.

— Si une image est visible après lancement du logiciel mais que son contenu est incohérent, il faut penser à un défaut au niveau des amplis de bus de données, à un défaut au niveau des RAM 4118 ou à un défaut au niveau des interconnexions entre EF 9340 et EF 9341.

Le schéma employé étant très sûr, un défaut de fonction-

nement ne peut, de toute façon, venir que d'une erreur de câblage ou d'un composant défectueux. En cas de problème rebelle, nous croyons savoir que certaines sociétés (Facim, Centrale d'achats informatique) peuvent procéder au dépannage ou à la mise au point de vos cartes sous certaines conditions (conditions à demander directement à ces sociétés et non à l'auteur de ces lignes qui n'a rien à voir avec ces services).

Les registres des EF 9340 et EF 9341

Une fois que vous vous êtes assuré du fonctionnement de votre carte au moyen du programme de la figure 7, il vous faut passer à la phase la plus intéressante (ou la plus désagréable, selon les goûts !) de l'utilisation de celle-ci, à savoir : l'écriture de vos logiciels. Pour vous faciliter au maxi-

mum cette tâche, nous allons expliciter, dans les lignes qui suivent, les fonctions des registres internes des EF 9340 et EF 9341. Nous commenterons ensuite les sous-programmes que nous vous avons fait éditer en suivant la figure 6 et qui reviendront dans toutes les applications, puis nous verrons un exemple de programme utilisant ceux-ci et faisant appel aux possibilités offertes par le générateur de caractères en RAM que nous n'avons pas encore utilisé.

Avant de nous plonger dans les registres, examinons la figure 8 qui représente l'organisation des informations sur l'écran et la définition de certains paramètres utilisés par la suite. La carte CGC peut fournir une image de 25 lignes de 40 caractères ; ces 25 lignes se subdivisent en deux blocs inégaux, 24 lignes « normales » et une ligne dite de service que l'on peut faire afficher ou non. Les lignes sont numérotées au moyen d'un registre interne appelé Y dont le contenu évolue de 00 à 23 pour les 24 lignes normales et qui doit contenir 31 (décimal) pour pointer sur la ligne de service.

Les colonnes des caractères sont repérées par le registre interne X dont le contenu peut évoluer de 00 à 39.

Ces deux registres internes X et Y sont en réalité les registres des coordonnées du curseur ; curseur qui va nous permettre de pointer les positions où l'on désire écrire ou dessiner. Ce curseur peut être visible ou invisible, comme expliqué lors de la description du registre R.

Indépendamment de cela, il existe un autre registre interne YO, visible figure 10, qui a pour nom registre origine. Les bits b_0 à b_4 de ce registre permettent de définir quelle sera la première ligne visualisée sur l'écran, étant entendu que la ligne de service de numéro 31 est toujours présente en haut de l'écran, qu'elle soit visible ou non. La figure 11 donne un exemple d'utilisation de ce registre YO. Le bit b_5 de ce registre permet aussi de sélectionner la fonction zoom lorsqu'il est positionné à 1. Dans ce cas, les douze premiè-

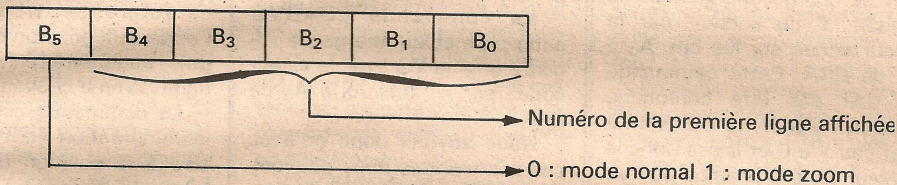
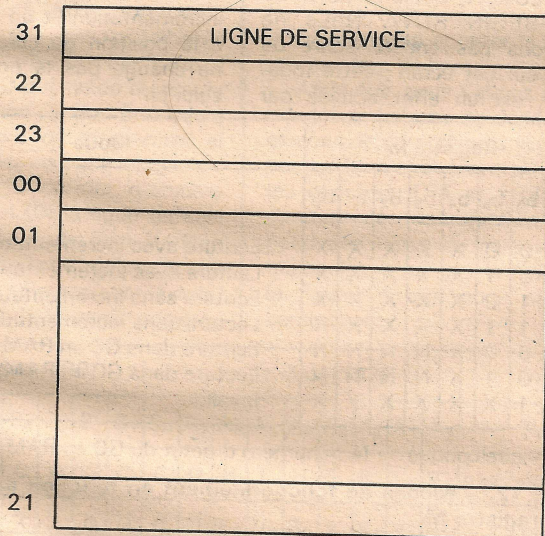
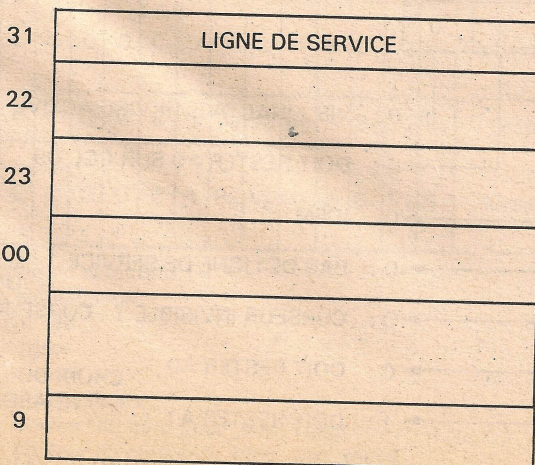


Fig. 10. — Fonction des bits du registre Y_0



YO = 22

PAS DE ZOOM



YO = 22

ZOOM

Fig. 11. — Disposition des lignes sur l'écran en fonction du contenu de YO.

res lignes de l'écran sont représentées en double hauteur. La figure 11 à nouveau donne un exemple d'utilisation de ce registre origine YO en mode zoom.

Deux autres registres internes existent ; ce sont M et R. Nous allons voir leurs fonctions dans un instant, mais il nous faut au préalable voir comment l'on accède à ces divers registres.

Ainsi que nous l'avons dit le mois dernier, et tel que cela apparaissait sur la figure 3 de notre précédent article, les échanges d'informations entre la carte et le bus du système se font au moyen d'un double registre 16 bits qui s'appelle TRA TRB pour les échanges de données et CRA CRB pour les échanges de commandes ou d'informations d'état. L'ensemble TRA TRB peut être lu et écrit par le 6809, alors que l'ensemble CRA CRB peut être écrit, mais seul le bit 7 de CRA peut être lu. Vu le contenu de la PROM VINGEN et la connexion des lignes A₀ et A₁ sur le circuit GEN EF 9341, les divers registres sont aux adresses suivantes : TRA en E200, TRB en E201, CRA en E202 et CRB en E203.

Ces registres étant des registres 16 bits et les informations exploitées par les circuits VIN et GEN étant des mots de 16 bits, l'écriture et la lecture des données ne doit pas s'y faire n'importe comment. Pour écrire dans TRA TRB ou CRA CRB, il faut toujours écrire d'abord dans TRA (CRA) puis dans TRB (CRB). C'est en effet l'écriture dans TRB (CRB) qui déclenche la procédure de prise en compte du mot de 16 bits. Si vous faisiez dans l'autre sens, le mot de 16 bits pris en compte serait exact pour sa partie B (celle écrite dans TRB ou CRB), mais la partie A serait quelconque.

Ces registres sont vos seules interfaces avec les circuits VIN et GEN et les accès aux registres évoqués ci-avant : R, M, YO, X, Y se font par leur intermédiaire selon une procédure que nous allons maintenant étudier.

La figure 9 présente la liste des commandes comprises par le circuit processeur VIN (EF 9340) avec l'affectation

des bits des registres CRA et CRB.

Ces commandes sont au nombre de sept :

- La commande BEGIN ROW permet de positionner le curseur au début de la ligne désirée. Le numéro de ligne est codé en binaire sur les bits A₀ à A₄ de CRA, comme indiqué figure 9. L'exécution de cette commande charge les registres X avec 0 (début de ligne) et Y avec la valeur codée sur les bits A₀ à A₄ de CRA.
- La commande LOADY permet de charger directement le registre Y du curseur par la valeur codée sur les bits A₀ à A₄ de CRA. Cette commande ne doit pas être confondue avec la précédente car elle ne met pas X à 0 et laisse donc le curseur dans la colonne où il se trouvait : c'est l'équivalent d'un saut ligne avant ou arrière sans retour chariot.
- La commande LOADX permet de charger directement le registre X du curseur par la valeur codée sur les bits A₀ à A₅ de CRA. Cette commande ne modifie pas le contenu de Y, et le curseur reste donc sur la ligne où il se trouvait.
- La commande INCC permet d'incrémenter de une unité la position du curseur sans que vous ayez à vous soucier de quoi que ce soit. Si, lors de cette incrémentation, X arrive à 39, il passera automatiquement à 00 lors de l'incrémentation suivante et Y sera alors augmenté de 1. Lorsque Y arrive à 23, l'incrémentation suivante le fait passer à 00. Si,

par contre, Y était sur la ligne de service de numéro 31, le fait que X passe de 39 à 00 fait passer Y de 31 à 00.

- La commande LOADM permet de charger le registre interne M dont nous n'avons pas encore parlé. Cette commande place la valeur codée sur A₀ à A₇ du CRA dans le registre M.
- La commande LOADR permet de charger le registre interne R dont nous n'avons pas encore parlé non plus. Cette commande place la valeur codée sur A₀ à A₇ dans le registre R.
- La commande LOADYO enfin permet de charger le registre origine YO avec la valeur codée sur les bits A₀ à A₅ de CRA.

Nous voyons donc qu'avec ces commandes nous pouvons positionner un curseur n'importe où sur l'écran au moyen de LOADX, LOADY, LOADYO et BEGIN ROW. Nous ne savons pas encore écrire ou lire sur cet écran ; cette fonction est en effet activée par

l'écriture d'un code particulier dans le registre M dont nous allons parler maintenant.

- La figure 12 présente les différents contenus autorisés pour le registre M avec les fonctions que cela active :
- La fonction écriture avec incrémentation permet d'écrire un caractère sur l'écran à la position du curseur, puis augmente celui-ci d'une unité en suivant les règles que nous avons exposées pour la commande INCC vue ci-avant.
 - La fonction lecture avec incrémentation permet de lire un caractère se trouvant sur l'écran à la position du curseur, puis augmente celui-ci d'une unité comme ci-avant.
 - La commande écriture sans incrémentation écrit un caractère à la position du curseur, mais ne change pas la position de celui-ci.
 - La commande lecture sans incrémentation lit un caractère à la position du curseur mais ne change pas la position de celui-ci.

Registre M								Mode
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	
0	0	0	X	X	X	X	X	Ecriture avec incrémentation
0	0	1	X	X	X	X	X	Lecture avec incrémentation
0	1	0	X	X	X	X	X	Ecriture sans incrémentation
0	1	1	X	X	X	X	X	Lecture sans incrémentation
1	0	0	X	N	N	N	N	Ecriture dans GC en RAM
1	0	1	X	N	N	N	N	Lecture dans GC en RAM
1	1	X	X	X	X	X	X	Invalide

X = quelconque N = numéro d'octet du GC en RAM

Fig. 12. - Modes de fonctionnement en fonction du contenu du registre M.

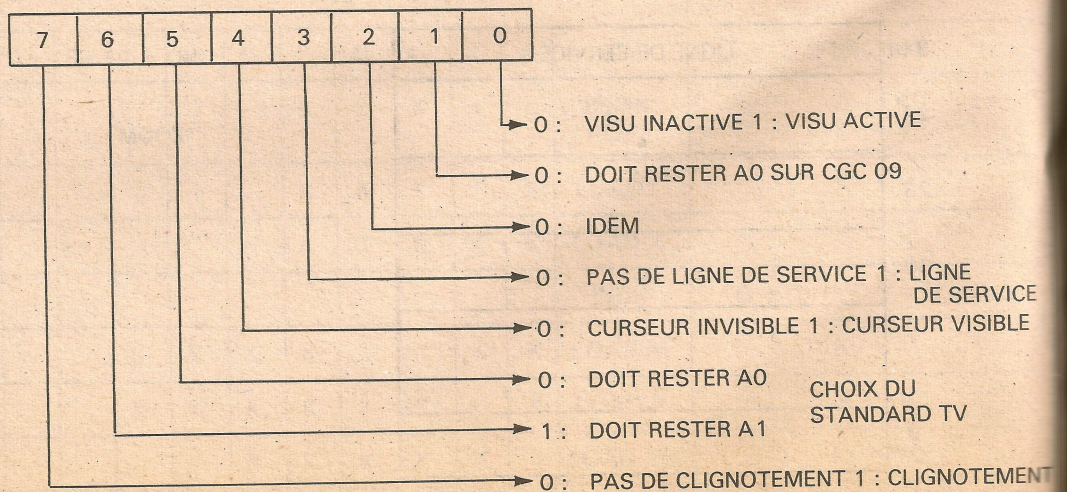


Fig. 13. - Fonctions des bits du registre R.

— La commande écriture dans le GC en RAM permet de définir un caractère dans le générateur de caractères en mémoire vive. Les bits b_0 à b_3 du registre M contiennent le numéro de la ligne de la matrice de définition du caractère (ces notions sont explicitées plus avant dans cet article).

— La commande lecture dans le GC en RAM permet de relire un caractère défini dans le générateur en RAM selon une procédure analogue à celle de l'écriture.

Certaines des notions exposées ci-avant vous semblent certainement obscures ; rassurez-vous, elles s'éclairciront lors de la présentation des exemples de programmes.

Il nous reste encore le registre R à étudier ; la définition de ses bits est indiquée figure 13 et peut presque se passer de commentaire. Seuls certains bits ont à être manipulés compte tenu des possibilités de notre carte. Ces bits sont :

— Le bit 0 qui valide ou non la visualisation ; il doit être mis à 1 en début de programme si vous voulez voir une image.

— Le bit 3 permet d'afficher (1) ou non (0) la ligne de service en haut de l'écran. Que cette ligne soit présente ou non, son emplacement existe, et la taille de la partie occupée

par les 24 lignes « normales » n'est donc pas modifiée.

— Le bit 4 autorise (1) ou non (0) la matérialisation du curseur ; cela permet donc d'utiliser la carte comme un terminal classique (curseur visible) ou comme une vraie visu alphanumérique et graphique (curseur invisible).

— Le bit 7 enfin autorise (1) ou non (0) le clignotement. Lorsque le clignotement est autorisé ainsi que la matérialisation du curseur, celle-ci se fait par un passage du caractère où se trouve le curseur de vidéo normale à vidéo inversée à la vitesse de clignotement (0,5 Hz) alors que, si le clignotement est interdit, le curseur est matérialisé par une inversion permanente de vidéo à l'endroit où il se trouve.

Ces définitions étant vues, il ne nous reste plus qu'à vous expliquer comment sont définis les codes des caractères pour pouvoir ensuite aborder la programmation de la carte.

Ainsi que nous l'avons déjà dit, les transferts de données, et donc de caractères à visualiser se font exclusivement par les registres TRA et TRB. Le contenu de ces registres dépend des types de caractères manipulés, ainsi que le montre la figure 14 que nous allons commenter.

Quatre types de caractères peuvent être manipulés par les circuits VIN et GEN : les caractères alphanumériques contenus dans le générateur interne, les caractères graphiques contenus dans le générateur interne, les caractères alphanumériques définis par vos soins et contenus dans le générateur de caractères en RAM et enfin les caractères graphiques définis par vos soins et contenus dans ce même générateur. Quel que soit son type, chaque caractère se voit associer un ou plusieurs attributs définissant sa taille, sa couleur, la couleur du fond, etc.

En règle générale, les codes des caractères sont placés dans le registre TRB et les attributs dans le registre TRA, comme schématisé dans le tableau de la figure 14. Ainsi, pour afficher un A majuscule sur l'écran, de code ASCII 41, nous écrirons 41 dans TRB et une combinaison de bits correspondant aux attributs que nous souhaitons donner à A dans TRA.

Pour les caractères alphanumériques, les attributs peuvent être au nombre de sept :

— Le bit A_6 définit si le caractère est en vidéo normale (0) ou inversée (1).

— Le bit A_5 définit si le caractè-

re est de largeur normale (0) ou est en double largeur (1).

— Le bit A_4 définit si le caractère est de hauteur normale (0) ou est en double hauteur (1).

— Le bit A_3 définit si le caractère clignote (0) ou est stable (1).

— Le bit A_2 définit si la couleur du caractère ne contient pas la composante bleue (0) ou la contient (1).

— Le bit A_1 définit de même la contenance de la composante verte.

— Le bit A_0 définit de même la contenance de la composante rouge.

Ainsi, pour écrire un A majuscule stable de taille normale, en vidéo normale et en blanc, nous écrirons 41 dans TRB (code ASCII de A) et 0F dans TRA (les quatre bits A_3 à A_0 à 1 puisque le blanc est la somme des trois composantes de base).

Pour ce qui est des caractères définis dans le générateur interne du circuit GEN EF 9341, les codes à employer dans TRB sont ceux indiqués sur les figures 4 et 5 de notre précédent article. En ce qui concerne les caractères alphanumériques « normaux », c'est-à-dire ceux auxquels l'informatique vous a habitués, le code ASCII est respecté, ce qui évite toute recherche ; seuls les caractères spéciaux tels que les minuscules accentuées, certains symboles et certaines « lettres » telles le oe collés ne suivent pas le code ASCII (pour la simple raison que leur définition selon ce code n'existe pas) et occupent les codes généralement affectés aux caractères de contrôle.

Pour ce qui est des caractères alphanumériques et graphiques que vous définissez vous-mêmes dans le générateur de caractères en RAM, les codes à utiliser sont ceux que vous décidez d'affecter aux caractères, la seule contrainte à respecter se situant au niveau des bits B_5 , B_6 et B_7 de TRB qui doivent être conformes à ce qui est indiqué figure 14.

En ce qui concerne les attributs des caractères graphiques, leur principe d'action est le même que pour les caractères alphanumériques, mais la fonction des bits d'attributs est différente ; en effet, il est

Code (TRB)										Attribut (TRA)							Type	Remarques
B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0			
0	C	C	C	C	C	C	C	0	N	L	H	S	B_1	V_1	R_1	Alphanumérique	jeu alpha du GEN	
1	0	0	X	X	0	0	0	0	B_0	V_0	R_0	X	B_1	V_1	R_1	Delete		
1	0	1							N	L	H	S	B_1	V_1	R_1	Alphanumérique	jeu en RAM	
1	1	0	C	C	C	C	C	0	N	L	H	S	B_1	V_1	R_1	Alphanumérique	jeu en RAM	
1	1	1							N	L	H	S	B_1	V_1	R_1	Alphanumérique	jeu en RAM	
0	C	C	C	C	C	C	C	1	B_0	V_0	R_0	S	B_1	V_1	R_1	Graphique	Jeu graphique du GEN	
1	0	1							B_0	V_0	R_0	S	B_1	V_1	R_1	Graphique	Jeu en RAM	
1	1	0	C	C	C	C	C	1	B_0	V_0	R_0	S	B_1	V_1	R_1	Graphique	Jeu en RAM	
1	1	1							B_0	V_0	R_0	S	B_1	V_1	R_1	Graphique	Jeu en RAM	

C : bit de code du caractère

X : bit sans signification

N : vidéo inversée

L : double largeur

H : double hauteur

S : vidéo non clignotante

B_0 , V_0 , R_0 : couleur du fond (bleu, vert, rouge)

B_1 , V_1 , R_1 : couleur du caractère (bleu, vert, rouge)

Fig. 14. — Codes et attributs des divers types de caractères.

possible, dans ce cas, de définir pour chaque caractère une couleur de caractère et une couleur de fond. La couleur du caractère est définie, comme pour les caractères alphanumériques, par les bits A₀, A₁ et A₂ alors que la couleur du fond est définie par A₄, A₅ et A₆. Le bit A₃, quant à lui, conserve la fonction de sélection entre caractère fixe ou clignotant.

Voilà ! Nous en avons fini avec cette description des registres internes. Nous sommes bien conscient du fait que c'était un peu pénible à suivre, bien que nous ayons allégé et simplifié certaines explications de la fiche technique Thomson-Efcis d'où ces informations sont extraites. Cet exposé était cependant nécessaire car, sans lui, il est impossible de programmer les circuits VIN et GEN. Pour la petite histoire, précisons que cela n'est rien à côté de la description des registres et fonctions du processeur graphique haute résolution qui équipe la carte du même nom (avis aux amateurs !).

Les sous-programmes standards

Compte tenu de la place déjà occupée par cet article, nous n'allons décrire aujourd'hui que les sous-programmes ne faisant pas appel au générateur de caractères en RAM, car ceux-ci nécessitent un long développement pour bien les maîtriser.

Ces sous-programmes sont ceux qui apparaissent sur le listing de la figure 6. Ce listing étant celui du bloc de sous-programmes, il dispose au début d'un certain nombre de définitions de constantes, d'adresses et de mémoires de travail utilisées par les sous-programmes. On y retrouve les adresses des registres, du GEN EF 9341, et aussi les codes des commandes présentées ci-avant qui évitent, à chaque appel de l'une d'elles, de devoir compter les bits sur ses doigts.

Nous laisserons de côté le sous-programme WRGENM, qui sera abondamment com-

menté le mois prochain avec plusieurs exemples d'utilisation du générateur de caractères en RAM, pour nous intéresser tout d'abord aux sous-programmes LOADR, LOADM, LOADX, LOADY et LOADYO. Ces cinq sous-programmes fonctionnent de la même façon et permettent de charger le registre dont ils portent le nom avec la valeur contenue dans l'accumulateur A lors de l'appel du sous-programme. Hormis pour LOADX et LOADY, aucun registre n'est modifié par l'exécution des sous-programmes. Si vous souhaitez qu'il en soit de même pour LOADX et LOADY, il suffit d'ajouter en début de sous-programme un PSHS A et, en fin de sous-programme, juste avant le RTS, un PULS A.

Le fonctionnement de ces sous-programmes est très simple : l'on commence par faire appel au sous-programme BUSY que nous allons voir dans un instant et qui est chargé d'indiquer si les 9340 et 9341 sont prêts ou non ; on place ensuite le contenu de A dans le registre CRA puisque nous avons vu, en figure 9, que c'est via ce registre que s'effectue le passage de paramètres à destination de R, M, X, Y et YO ; l'on écrit ensuite dans le registre CRB le code correspondant à la commande à exécuter. C'est simple et logique.

Le sous-programme BUSY, quant à lui, fait appel à une fonction du registre CRA dont nous n'avons pas parlé en détail : l'indication d'état des circuits VIN et GEN. En effet, le bit 7 du CRA permet de savoir si ces circuits sont libres ou sont en train de travailler : lorsque b₇ de CRA est à 0, les circuits sont libres et peuvent accepter une commande ; lorsque b₇ est à 1, les circuits sont en train d'exécuter la commande précédente et ne peuvent donc en accepter une autre. Le sous-programme BUSY lit donc ce bit et attend qu'il soit à 0 pour effectuer un RTS, revenant ainsi au programme qui l'a appelé. Ce sous-programme doit impérativement être appelé avant toute exécution de commande, sinon vous courez le risque de voir votre commande ignorée

ou, ce qui est pire, mal interprétée.

Trois autres sous-programmes vous sont proposés sur ce listing :

— Le sous-programme ROW permet de placer le curseur au début de la ligne dont le numéro est spécifié dans l'accumulateur B. Après avoir attendu la fin d'une éventuelle commande grâce à BUSY, il place le contenu de B dans le CRA et exécute une commande... BEGROW, bien sûr, comme indiqué dans le tableau de la figure 9.

— Le sous-programme LOADXY positionne le curseur sur la ligne dont le numéro est contenu dans B et sur la colonne dont le numéro est contenu dans A. Son principe est fort simple, surtout du fait de l'existence des sous-programmes LOADX et LOADY. Le numéro de colonne étant contenu dans A, il suffit d'utiliser LOADX puis de transférer B dans A pour pouvoir ensuite appeler à nouveau LOADY. Ce sous-programme n'est pas réellement utile vu sa simplicité, mais nous l'avons tout de même inclus pour montrer que, si l'on écrit suffisamment de petits sous-programmes élémentaires, l'accomplissement de toute fonction plus complexe se résume à des appels à ceux-ci assortis d'opérations très simples.

— Le dernier sous-programme nous restant à voir est un des plus utiles puisqu'il permet d'écrire un caractère, dont le code est contenu dans B et les attributs dans A, à la position choisie dont le numéro de ligne est contenu dans la RAM CURY et le numéro de colonne dans la RAM CURX. Qui plus est, ce sous-programme ne modifie ni CURX, ni CURY, ni aucun registre du 6809. Son fonctionnement est fort simple : l'on commence par sauvegarder A et B sur la pile, on met ensuite A à 0, puis l'on exécute une commande LOADM au moyen du sous-programme adéquat. Si vous regardez la figure 12 vous constaterez que cela place les circuits 9340 et 9341 en mode écriture avec incrémentation du curseur ; après avoir chargé A et B par CURX et CURY respectivement, on exécute

ensuite un LOADXY qui a pour effet d'amener le curseur à la position où l'on veut écrire. Il ne reste plus alors qu'à récupérer A et B de sur la pile, c'est-à-dire de les recharger avec le code et les attributs du caractère à afficher, d'attendre la fin d'exécution de la commande précédente grâce à BUSY pour pouvoir écrire A et B dans TRA et TRB, ce qui assurera leur transfert sur l'écran à la position désirée.

Ces explications étant données, il est facile de comprendre le fonctionnement du petit programme d'essai de la carte dont nous vous avons présenté le listing en figure 7.

Ce programme commence par une initialisation du registre R, réalisée au moyen du sous-programme LOADR. Les bits sont positionnés conformément à la figure 13 et aux fonctions désirées.

Le registre YO est initialisé ensuite au moyen du sous-programme LOADYO, la valeur placée dans YO étant nulle pour que la première ligne soit en haut de l'écran, en dessous de la ligne de service qui, rappelons-le, occupe toujours la première position.

Les quelques lignes suivantes initialisent toute la mémoire d'écran avec le caractère DEL (code ASCII 7F) et les attributs permettant de réaliser un écran noir (aucune couleur activée). Cette initialisation est indispensable, car le contenu des RAM est quelconque à la mise sous tension. La taille de la page visualisée étant de 25 X 40, soit 1 000 caractères, l'initialisation ne porte que sur ces mille là, les 24 restants n'ayant pas besoin d'initialisation puisqu'ils ne sont jamais visibles. Remarquez le numéro de ligne initial positionné à 31 qui permet d'initialiser aussi la ligne de service avec passage automatique aux lignes suivantes, puisque nous vous avons expliqué que le compteur de lignes Y rebouclait de 31 en 00.

La partie suivante permet ensuite de charger successivement les divers jeux de caractères avec des attributs différents dans la page visualisée.

On fait pour cela appel au sous-programme LOADM pour placer les circuits en mode

écriture avec incrémentation automatique du curseur puis, à son rôle, un attribut est stocké dans l'accumulateur A et l'appel au sous-programme JCAR fait afficher les uns à la suite des autres (en raison du mode incrémentation automatique) tous les caractères.

Le sous-programme JEUR est très simple ; il attend que les circuits soient prêts par l'appel à BUSY, puis écrit l'attribut dans TRA et le contenu de B dans TRB. Comme le contenu de B évolue de 00 à 7F inclus, l'on passe en revue tous les codes des caractères contenus dans le jeu de caractères.

Nous en resterons là pour aujourd'hui avec ces explications logicielles ; il nous reste, en effet, beaucoup à dire concernant le générateur de caractères en RAM, et nous y consacrerons une partie de notre prochain article.

A titre d'exercice dont la solution vous sera fournie dans notre prochain numéro, essayez d'écrire un sous-programme qui lit un caractère affiché sur l'écran et place son code dans B et ses attributs

dans A. La position du caractère est contenue dans deux RAM CURX et CURY. Inspirez-vous pour cela du sous-programme WRMP dont vous serez très proche.

A propos du logiciel Graphix

Nous vous avons présenté, dans notre précédent numéro, ce logiciel qui permet de faire du graphique sur la carte IVG09 ; logiciel réalisé par la Centrale d'Achats Informatique et que nous avons beaucoup apprécié.

Le couple logiciel Graphix et carte IVG ou IVG09 peut cependant être amélioré très sérieusement au niveau du comportement de la carte lors des déplacements d'objets sur celle-ci. Telle que la carte est conçue, ces déplacements s'accompagnent d'effets très désagréables pendant tout le mouvement ; il suffit, pour y remédier, d'enlever une des TMS 4044 de la carte et de court-circuiter son entrée avec sa sortie. La figure 15 précise la mémoire à enlever et les

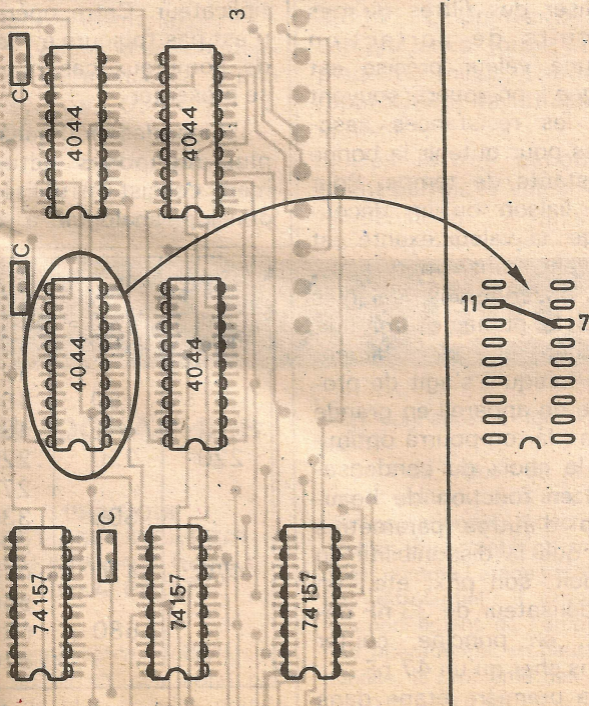


Fig. 15. — Amélioration du comportement de la carte IVG09 avec le logiciel Graphix.