PROGRAMMA

80-GRAFIX

GUIDE DE RÉFÉRENCE DE L'UTILISATEUR

80-GRAFIX

Développé par :
Ted Carter

Distribué exclusivement par:
PROGRAMMA International, Inc.
3400 Wil sh ire Boulevard
Los Angeles, CA 90010

COPYRIGHT (C) 1980

Tous droits réservés, aucune partie de ce manuel ou des programmes informatiques associés ne peuvent être reproduits sous quelque forme que ce soit, ni par tout moyen Electronic ou mechanical, y compris les systèmes de stockage et de récupération d'informations, sans le consentement et la permission de PROGRAMMA Internat Inc.

GARANTIE LIMITÉE

PROGRAMMA n'a aucune responsabilité envers l'acheteur ou toute autre personne ou entité en ce qui concerne une quelconque responsabilité, Perte ou dommage causé ou prétendument causé directement ou indirectement par ce produit, y compris, mais sans s'y limiter, toute interruption de service, perte d'activité et bénéfices anticipés, ou dommages conséquents résultant de l'utilisation ou de l'exploitation de ce produit. Ce produit sera réparé ou remplacé par notre option dans les quatre-vingt-dix (90) jours suivant la date d'achat s'il est jugé défectueux dans la fabrication, l'étiquetage ou l'emballage, mais sauf pour cette réparation ou un remplacement, la vente ou l'utilisation ultérieure de ce produit est sans garantie ni responsabilité.

PROGRAMMA et ses auteurs ne sauraient être tenus responsables des dommages causés au matériel, aux logiciels ou aux données par l’utilisation de leurs produits. Hormis le remplacement du produit concerné, la vente ou toute manipulation ultérieure de ces produits s’effectue sans aucune garantie ni responsabilité, y compris en cas de défaut, de dommage ou de perte résultant d’une négligence ou de toute autre faute.

REMARQUE : L'utilisation ou l'installation de 80-GRAFX peut annuler la garantie de votre système informatique TRS—80.

Table des matières

LOGICIEL 80-GRAFX
Description 1

DOCUMENTATION 80-GRAFIX
Bienvenue 2

INSTALLATION 1.0 3

1.1 DÉPANNAGE DE VOTRE INSTALLATION..... 7

2.0 THÉORIE DU FONCTIONNEMENT 9

3.0 UTILISATION DU 80-GRAFIX..... 11

3.1 CONTRÔLE DE MODE..... 11

3.2 PROGRAMMATION DE LA RAM DE CARACTÈRES 12
Programme #1..... ….....14
Programme #2.....14

3.3 TECHNIQUES EN LANGAGE ASSEMBLEUR 17
Lectures recommandées 18
Pilote de clavier pour 80-GRAFIX..... 19

3.4 PROGRAMMATION DE GRAPHIQUES MOBILES 21

3.5 AUTRES IDÉES DE PROGRAMMATION 22

3.6 QUELQUES RÉFLEXIONS POUR CONCLURE.....24

Liste des illustrations

Fig. 1 VUE D'ENSEMBLE - Installation

Fig. 2 ENCART – Connexion par soudure, étape 7

Fig. 3 ENCART – Connexion par soudure, étape 10

80-GRAFIX SOFTWARE PACKAGE

CÔTÉ 1

BASIC programs : use "CLOAD"

1. DEMO
2. CREATE
3. LOCASE/BAS
4. TIE FIGHTER

CÔTÉ 2

```
1.  LOCASE -- SYSTEM <CR>  *?LOCASE <CR>  *?/ <CR>
2.  LCDOS  -- Transfer to disk using TAPEDISK, etc.
              START=7000H / END=71FAH / TRA=7000H
```

Les programmes Side I constituent des exemples de méthodes de programmation en BASIC.

Les programmes Side 2 sont des jeux de caractères en minuscules avec des pilotes video ajoutés pour permettre un véritable fonctionnement de clavier de machine à écrire (c'est-à-dire que les minuscules sont affichées pour les entrées normales, les majuscules pour les lettres hachées).

80-GRAFIX DOCUMENTATION

Bienvenue dans le monde passionnant de 80-GRAFIX!

Une fois la carte 80-GRAFIX installée, votre TRS-80™ se transformera en un système inédit et puissant, doté de capacités graphiques qui dépasseront vos rêves les plus fous. De nouveaux horizons en matière de programmation s'ouvrent à vous ; à vos doigts, prêts à parcourir le clavier pour de nouvelles aventures informatiques !

Dans les pages qui suivent, nous explorerons les diverses techniques pouvant être utilisées pour programmer les nouvelles capacités graphiques haute résolution de votre ordinateur, et nous examinerons certains principes de fonctionnement propres à la carte. Par la suite, vous découvrirez des exemples concrets de programmation que vous pourrez intégrer à vos propres projets ; cela inclut des approches en langage BASIC (versions Level II et Disk), ainsi que des listings en langage assembleur pour les plus audacieux.

Le projet 80-GRAFIX™ est si novateur et radicalement différent de tout ce qui existait jusqu'alors pour le TRS-80™ que nous n'en avons fait qu'effleurer les possibilités ; nous attendons avec impatience vos découvertes, en tant que pionnier dans ce domaine !

Ralph Burris
for PROGRAMMA INTERNATIONAL
July, 1980

INSTALLATION

1.0 INSTALLATION

AVANT DE SAISIR VOTRE TOURNEVIS...

Comme vous le savez, si vous n'avez jamais ouvert votre clavier auparavant, sachez qu'une telle opération annulera votre garantie limitée de 90 jours. De plus, en cas de problème, le centre de réparation pourrait ne pas apprécier votre intervention ; il est donc judicieux d'évaluer vos compétences en soudure avant de vous lancer. Si vous ne vous sentez pas capable d'effectuer deux soudures, n'hésitez pas à demander de l'aide à une personne expérimentée, ou confiez cette tâche à un technicien en électronique ou à un atelier de réparation. Vous souhaitez tenter l'expérience ? C'est parti...

Tout d'abord, dégagez un espace de travail d'une taille comparable à celle d'une table de jeu. (Il est judicieux d'utiliser une surface souple et antistatique afin d'éviter de rayer les capots en plastique et le clavier.) Ensuite, rassemblez les outils suivants :

1 tournevis cruciforme
1 pince coupante
1 petite pince à bec effilé
1 fer à souder de 15 watts, avec panne de 3/32" (environ 2,4 mm)
1 tube de colle « Duco » ou equivalent (Optionnel)
1 loupe

Retirez toutes les fiches, positionnez la surface du clavier et suivez ces étapes :

1. Retournez le clavier ; repérez et retirez les six (6) vis situées sur la face inférieure. (L'une d'elles est un peu délicate : elle est accessible à travers la grille et peut être dissimulée par un autocollant ou par la substance utilisée pour sceller la garantie.)

2. Retournez délicatement le clavier et retirez le capot supérieur.

3. Soulevez la carte de circuit imprimé du clavier et laissez-la pivoter vers vous. FAITES ATTENTION à ne pas tordre ni casser le câble du connecteur en bas à gauche. Retirez les cinq entretoises blanches sous le (et notez où le clavier était inséré).

4. Retirez l'ensemble du circuit imprimé et du clavier du boîtier et placez la face des composants de la carte PC vers le haut, avec le clavier au-dessus

INSTALLATION

5. Repérez le circuit intégré (CI) n° Z-11. Selon la version de votre carte, ce CI peut être câblé directement ou monté sur un support. S'il est monté sur un support, retirez-le délicatement (saisissez-le par les extrémités et tirez doucement). S'il est câblé directement, repérez la broche 16 et court-circuitez-la avec une pince coupante, en veillant à ne pas endommager les pistes ou autres broches du CI.

6. Pliez les nappes de la carte 80-GRAFIX de manière à ce qu'elles se rabattent au-dessus des connecteurs présents sur la carte, puis posez celle-ci sur les composants du TRS-80, côté circuit vers le haut, en veillant à ce que les nappes ressortent sur le côté gauche.

7. Repérez le fil unique qui dépasse de la carte 80-CRAFIX, ainsi que le circuit intégré (CI) n° Z-25 sur la carte TRS-80 PC. En veillant tout particulièrement à ne pas surchauffer le circuit intégré, soudez ce fil à la broche 8 du composant Z-25 (la broche située tout à fait à gauche). Vérifiez cette soudure à l'aide d'une loupe afin de vous assurer qu'aucun « pont de soudure » ne s'est formé accidentellement ce qui risquerait de provoquer un court-circuit entre des pistes ou entre deux broches et vérifiez que la connexion est bien solide.

8. Positionnez maintenant le connecteur à nappe le plus éloigné de vous au-dessus du circuit intégré (CI). Déposez une goutte de colle (ou utilisez du ruban adhésif double face) sur le dessus du CI, puis pressez le connecteur contre celui-ci. Assurez-vous que toutes les broches du connecteur sont en contact avec les broches correspondantes du CI et qu'elles n'entrent en contact avec aucune autre broche. Si vous utilisez de la colle pour maintenir l'assemblage, veillez à ce que le connecteur soit parfaitement aligné (à fleur) avec le dessus du CI lors de la mise en place. Il peut être nécessaire de retoucher légèrement les extrémités des broches du connecteur si l'alignement n'est pas parfait.

   Lors de tous nos tests, ce type de connexion s'est avéré fiable. Il est donc déconseillé de souder les connecteurs directement sur les circuits intégrés, car cela rendrait leur retrait difficile et risquerait d'endommager le CI par surchauffe. Bien entendu, si vous possédez une grande expérience de ce genre de travaux, vous pouvez préférer utiliser des supports (socles) pour les circuits intégrés du TRS-80. Dans ce cas, soudez des supports à profil bas, puis insérez les connecteurs dans ces supports (en suivant la procédure décrite à l'étape n°8 pour les connecteurs à nappe).

INSTALLATION

9. Placez le composant suivant sur le circuit intégré (CI) Z-29, de manière à ce que certaines broches ne soient pas connectées. Notez que le CI Z-29 comporte 18 broches, alors que notre connecteur n'en possède que 16 ; il est donc positionné de manière décentrée, de sorte que les deux broches du CI les plus proches de vous restent déconnectées.

10. Le connecteur suivant, situé plus loin (et qui présente deux broches manquantes ainsi qu'un fil qui en dépasse), est relié au circuit intégré IC 11. (Si votre TRS-80 est équipé d'un support pour ce circuit, insérez le connecteur dans ce support.) Repérez la broche n° 11 du circuit intégré IC Z-12 (il s'agit de la broche centrale sur le côté gauche). En faisant preuve de la prudence recommandée précédemment, soudez le fil provenant du câble du connecteur à la broche n° 11, puis vérifiez votre travail afin de vous assurer qu'aucune erreur courante n'a été commise.

11. À présent, le dernier connecteur (le plus long) est fixé au circuit intégré N° Z-75, situé de l'autre côté de la carte.

Vous devez maintenant comparer attentivement votre installation terminée avec l'illustration fournie. Si tout est correct, vous pouvez remonter votre ordinateur. La carte 80-GRAFIX s'insère dans le compartiment droit du clavier et peut être fixée à l'aide des pastilles adhésives fournies. Veillez à ne pas débrancher les connecteurs lors du remontage du clavier.

Place maintenant à la partie amusante (espérons-le !).

Branchez les câbles d'alimentation et vidéo, puis mettez l'appareil sous tension. Si tout semble normal à l'écran (par exemple, si la mention « Memory Size » s'affiche), alors tout est prêt ! Si un événement imprévu survient, coupez RAPIDEMENT l'alimentation et consultez la section « Dépannage de votre installation ».

Vous pouvez désormais découvrir 80-GRAFIX en saisissant et en exécutant ce programme;

```
10 CLS: OUT 255, 32      :set to normal screen
20 FOR X = 0 TO 255      :poke rom character set
30 POKE X+15616,X        :  on screen
40 NEXT
50 INPUT"HIT ENTER FOR HI-RES! READY";A$
60 OUT 255, 160          :set to hi-res mode
```

Ce programme affichera deux fois l'ensemble complet des caractères du TRS-80, suivi de l'ensemble standard des caractères graphiques (également répété deux fois). Lorsque vous appuierez sur la touche « ENTER », le mode passera en haute résolution ; les caractères graphiques devraient alors être remplacés par des bandes verticales ou par une multitude de minuscules points clairs et sombres disposés de manière aléatoire. (Si cela ne se produit pas, éteignez l'appareil et consultez la section « Dépannage ».)

INSTALLATION

Les nouveaux caractères qui s'affichent à l'écran correspondent au contenu des 64 caractères programmables. Ils ne sont pas identifiables en tant que tels, car ils n'ont pas encore été programmés ; toutefois, ce test démontre que la carte est bien installée et qu'elle fonctionne correctement.

Si tout est opérationnel, passez à la section 2.

DÉPANNAGE

1.1 DÉPANNAGE DE VOTRE INSTALLATION

Votre carte 80-GRAFIX a fait l'objet d'un rodage et de tests approfondis avant de quitter l'usine. Par conséquent, si vous rencontrez des problèmes de fonctionnement, vous devriez d'abord envisager un éventuel problème d'installation.

Voici quelques-uns des problèmes possibles ainsi que leurs causes probables :

1. Écran totalement sombre au démarrage

Cela peut constituer un problème sérieux. Vérifiez d'abord que le moniteur est correctement branché et qu'il fonctionne normalement. Si le problème persiste, cela indique probablement que deux broches (ou plus) sur la carte de circuit imprimé TRS-80 sont entrées en contact, provoquant un court-circuit. Un tel court-circuit accidentel pourrait endommager d'autres composants de votre ordinateur. Examinez attentivement chaque connexion - idéalement à l'aide d'une loupe - pour détecter d'éventuels ponts de soudure ou des broches tordues. L'utilisation d'un ohmmètre pour vérifier l'intégrité des circuits serait également une bonne initiative.

2. ÉCRAN GRIS SCINTILLANT (aucune image)

   Ce problème est probablement lié au connecteur du circuit intégré N° Z-11. Assurez-vous que les broches appropriées sont bien raccordées et qu'elles sont en contact ferme.

3. POINTS CLIGNOTANTS ALÉATOIRES PENDANT L'EXÉCUTION DU PROGRAMME DE TEST

   Vérifiez la connexion au circuit intégré (CI) N° Z-12 pour vous assurer qu'elle est correcte : il est installé sur la broche appropriée et est solidement fixé. La cause la plus probable suivante se situe au niveau de la connexion au circuit intégré (CI) N° Z-II.

4. POINTS SCINTILLANTS ALEATOIRES A L'ECRAN

   Vérifiez la connexion au circuit intégré (CI) N° Z-29 afin de garantir une installation correcte et un bon contact.

5. DES GRAPHIQUES TRS-80 D'AUTREFOIS S'AFFICHENT PENDANT LE PROGRAMME DE TEST

La connexion par fil unique à la broche 8 du circuit intégré #Z-25 est probablement incorrecte ou n'établit pas un bon contact.

DÉPANNAGE

(Si la carte 80-GRAFIX ne parvient pas à afficher correctement les caractères programmés, à effacer les caractères graphiques à l'écran en mode programmation, ou à afficher des graphismes normaux en mode haute résolution, ces connexions sont généralement en cause.)

6. REPRODUCTION INCORRECTE DES CARACTÈRES PROGRAMMÉS

Si un programme de création de caractères (tel que « CREATE », fourni avec la carte 80-GPAFIX) ne renvoie pas les caractères créés, vérifiez la connexion au circuit intégré (CI) N° Z-47 ainsi que sa bonne installation.

En règle générale, vous devez vérifier attentivement chaque connexion pour vous assurer qu'elle se trouve au bon emplacement, que les connecteurs sont bien en place et que les deux fils isolés sont correctement soudés.

Si vous ne parvenez pas à identifier et à corriger le problème, écrivez au fabricant en fournissant une description détaillée de tous les aspects du problème. Si, après avoir identifié et corrigé toutes les erreurs d'installation, le module 80-GRAFIX ne fonctionne toujours pas correctement, il est possible qu'une erreur de votre part ait endommagé le 80-GRAFIX, le TRS-80, ou les deux. Dans ce cas, veillez à décrire, dans votre courrier, toutes les étapes que vous avez suivies lors de l'installation du module.

PRINCIPE DE FONCTIONNEMENT

2.0 PRINCIPE DE FONCTIONNEMENT

Si vous avez un exemplaire du manuel de référence technique TRS-80 (Radio Shack Cat. 26-2103), vous y trouverez des descriptions détaillées des différents composants que nous aborderons dans cette section.

La section vidéo de votre TRS-80 se compose d'un bloc de 1 Ko de mémoire vive (RAM) vidéo, d'une mémoire morte (ROM) de caractères (MCM6670) qui génère les caractères alphanumériques, et d'un multiplexeur (74153) permettant de créer des graphismes rudimentaires avec une résolution de 129 points horizontalement sur 48 verticalement. Chaque caractère alphanumérique est stocké dans la ROM de caractères sous forme d'une configuration de bits (0 ou 1) au sein d'une matrice de 5 points horizontalement sur 7 verticalement. Les caractères graphiques sont créés différemment : ils forment des « cellules » de 2 points horizontalement sur 6 verticalement, tout en exploitant l'ensemble des 12 lignes de balayage pour chaque rangée. 80-GRAFIX propose une mémoire graphique alternative qui offre, en mode graphique, un choix de configurations de points comparable à celui des caractères alphanumériques, à la différence que chaque caractère est ici constitué d'une grille de 6 points horizontalement sur 12 verticalement. (Plus de détails à ce sujet ultérieurement.)

Le TRS-80 combine des caractères alphanumériques et des caractères graphiques à l'écran ; il détermine quel type afficher en examinant l'état du bit de poids fort (MSB, pour *Most Significant Bit*) dans chacun des 1024 octets de la mémoire vidéo (RAM vidéo). Si le MSB est à l'état haut (activé), un caractère graphique est affiché. S'il est à l'état bas (désactivé), il s'agit d'un caractère alphanumérique. Par ailleurs, la différence entre ces deux états (0 ou 1 en binaire) équivaut à ajouter ou à soustraire 128 à la valeur de chaque octet. Les valeurs correspondant aux caractères graphiques s'échelonnent de 128 à 191 en décimal (de 80 à BF en hexadécimal).

La mémoire vidéo du TRS-80 se compose de 1024 emplacements (octets), situés aux adresses RAM 15360 à 16383 (3C00 à 3FFF en hexadécimal) ; ces données sont affichées sur le moniteur vidéo sous la forme de 16 lignes contenant chacune 64 ou 32 caractères. Ces emplacements mémoire sont accessibles depuis le langage BASIC via les instructions PEEK, POKE et PRINT. Pour plus de détails sur ces fonctions ainsi que sur l'utilisation des valeurs CHR$, veuillez consulter le manuel « Level II BASIC ». Reportez-vous également aux annexes -notamment la carte mémoire (« Memory Map ») et les tableaux des codes ASCII et graphiques - pour obtenir des précisions sur l'adressage de l'écran.

Il existe une différence entre la mémoire vidéo (Video RAM) standard du TRS-80 et le reste de la mémoire vive : chaque octet ne comporte que 7 bits ! Le 8eme bit a été supprimé ; par conséquent, si l'on écrit (via l'instruction POKE) une valeur inférieure à 32 dans cette mémoire, une lecture (via l'instruction PEEK) renverra cette valeur augmentée de 64. À l'inverse, pour les valeurs supérieures à 191, une lecture (PEEK) renverra la valeur diminuée de 128.Si vous avez fait installer un kit « minuscules » (lower case kit) de Radio Shack, le 8eme bit est rétabli et la mémoire vidéo se comporte alors comme le reste de la mémoire vive. (D'autres types de kits « minuscules » offrent des fonctionnalités similaires ; certains disposent d'un commutateur pour activer ou désactiver ce bit supplémentaire, tandis que d'autres permettent de le gérer par logiciel.) Cette organisation mémoire n'a pas d'incidence particulière sur l'utilisation de 80-GRAFIX, mais il est utile d'en tenir compte pour vos futurs développements logiciels.

PRINCIPE DE FONCTIONNEMENT

L'installation de la carte 80-GPAFIX étend l'écran graphique à une résolution de 384 horizontale et 192 verticale, ce qui est accompli, comme mentionné précédemment, en fournissant des cellules graphiques de 6 par 12.

Le 80-GRAFIX propose également un verrouillage de sélection de mode contrôlé par logiciel, accessible via le port cassette. Le 80-GRAFIX dispose de trois modes de fonctionnement :

1. Affichage normal - affiche les caractères graphiques TRS-80 standard.

2. Haute résolution – affiche le jeu de caractères programmé dans le 80-GRAFIX.

3. Mode programmation – permet au 80-GRAFIX d’accepter des données de caractères programmables et de générer un jeu de caractères.

Le 80-GRAFIX intègre un bloc de mémoire de 1 Ko (octets) sur 6 bits, qui fait office de mémoire vive pour caractères programmables (PCR - *Programmable Character RAM*). La mémoire vidéo (VRAM) y puise les cellules de caractères, tout comme elle le ferait à partir de la mémoire morte (ROM) de caractères MCM6670. La PCR est organisée en 64 unités - appelons-les « champs » (*FIELDS*) - qui constituent l'ensemble des caractères programmables. Chaque champ de la PCR se compose de 16 octets, dont les 12 premiers contiennent les données correspondant à chaque ligne du caractère (les 4 derniers octets de chaque champ sont ignorés). Lorsqu'il est placé en « mode programmation », le 80-GRAFIX utilise la mémoire vidéo comme tampon de données. Dans ce mode, les données des caractères définies par l'utilisateur sont transférées vers l'écran ; celui-ci est alors interprété comme une représentation de la PCR, dont le contenu est ensuite copié dans la mémoire PCR. Afin d'éviter l'affichage d'éléments parasites (« parasites » ou *garbage*) à l'écran, tout caractère dont la valeur est égale ou supérieure à 128 est masqué (rendu invisible).

Les caractères programmables 6x12 de 80-GRAFIX permettent d'accéder à chaque point de chaque ligne de chaque caractère. Cela autorise la création de graphismes continus, car chaque point de la matrice binaire correspond à un point sur l'écran vidéo. Lorsqu'une cellule graphique est affichée sur l'écran, le bit de poids fort (MSB) se situe à l'extrême gauche, et le bit de poids faible (LSB) à l'extrême droite, sur chaque ligne de balayage de la cellule.

3.0 UTILISATION DU 80-GRAFIX

3.1 CONTRÔLE DU MODE

Comme indiqué dans le chapitre précédent, 80-GRAFIX propose trois modes de fonctionnement, que l'on sélectionne en envoyant une valeur au port cassette (port 255 en décimal, soit $FF en hexadécimal). Les trois bits de poids fort - les bits 5, 6 et 7 - servent de bits de contrôle lorsque cette valeur est transmise au port. En langage BASIC, les instructions suivantes permettent de définir le mode de fonctionnement :

1. OUT 255,32 – Configure 80-GRAFIX pour afficher les blocs graphiques standard du TRS-80 (bit de contrôle 5). Dans ce mode, le TRS-80 se comporte exactement comme avant l'installation de 80-GRAFIX. Il s'agit de l'état par défaut de la carte au démarrage. Ce mode peut être activé ou désactivé à tout moment, sans aucune incidence sur les caractères programmés éventuellement stockés dans la carte 80-GRAFIX.

2. OUT 255,160 – Configure le 80-CRAFIX en mode graphique haute résolution et affiche le jeu de caractères que vous avez programmé dans la carte. (Dans ce mode, les bits graphiques 5 et 7 habituels sont remplacés par des données haute résolution). Les caractères alphanumériques restent inchangés.

3. OUT 255, 96 – place le 80-GRAFIX en mode programmation (bits de contrôle 5 et 6). Tous les caractères insérés via les commandes POKE ou PRINT sur l'écran du TRS-80 sont lus par le 80-GRAFIX et stockés sous forme de configuration de bits dans la mémoire vive (RAM) des caractères programmables. Les caractères dont le bit de poids fort (MSB) est activé sont masqués à l'écran, afin d'éviter l'affichage de données incohérentes (« parasites ») pendant la programmation du module. Il est conseillé de quitter ce mode dès que la programmation du 80-GRAFIX est terminée.

3.2 PROGRAMMATION DE LA RAM DE CARACTÈRES

La mémoire de caractères programmable (PCR) est organisée en 64 cellules de 72 bits (disposées en une matrice de 6 colonnes sur 12 lignes). Pour programmer la PCR, le module 80-GRAFIX doit être placé en mode « PROGRAMMATION » (via la commande `OUT 255, 96`) ; les valeurs correspondantes sont ensuite placées à l'écran par blocs de 16 octets, dont seuls les 12 premiers sont lus par la PCR (les 4 derniers octets de chaque bloc sont ignorés). Chaque bloc de données correspond à un caractère programmé et doit débuter à l'adresse correspondant au premier octet de l'écran (15360 en décimal, 3C00 en hexadécimal) ; chaque bloc suivant doit commencer à une adresse située à un multiple de 16 octets de la position initiale. Ainsi, le premier bloc de données occupe les positions 15360+0 à 15360+15, le bloc suivant commence à 15360+16 et se termine à 15360+27, et ainsi de suite pour l'ensemble des 64 caractères.

L'octet placé en première position de chaque bloc correspond à la configuration binaire de la ligne supérieure du caractère ; le deuxième octet correspond à la deuxième ligne, et ainsi de suite pour chacune des 12 lignes de chaque caractère.

Si la logique binaire vous a déjà effrayé, il est temps de reléguer ces craintes aux oubliettes, car les valeurs correspondant à chaque caractère que vous créez dépendent de l'état - activé ou désactivé - des différents bits. En réalité, vous pouvez approfondir votre compréhension de la logique binaire grâce à 80-GRAFIX : il s'agit, mes amis, d'un véritable outil qui transforme ces concepts abstraits en une réalité immédiate ! Mais je m'égare. Appréhender ce concept est, en fait, très simple : pour chaque point que vous souhaitez allumer parmi les six positions horizontales, le bit correspondant est activé. Ainsi, si vous désirez, par exemple, allumer un point sur deux, la configuration des bits se présentera comme suit :

(MSB) 1 0 1 0 1 0 (LSB)

Pour traduire cela en un nombre courant pouvant être programmé dans le 80-GRAFIX, il suffit d'additionner les valeurs correspondant aux positions activées (c'est-à-dire celles qui présentent un « 1 »). En partant de la droite vers la gauche, la première position (le bit de poids faible, ou LSB) a une valeur de 1, et chaque position suivante représente le double de la valeur de la précédente. Ainsi, notre octet de 6 bits possède les valeurs suivantes :

(MSB) 32 16 08 04 02 01 (LSB)

Ainsi, en reprenant les valeurs de notre exemple ci-dessus, on peut constater que ces bits sont activés :

(MSB) 32 xx 08 xx 02 xx (LSB)

En additionnant ces valeurs de position, on obtient le nombre 42 en base décimale. Chacune des 12 lignes (valeurs) peut être facilement déterminée en notant la séquence de bits que l'on souhaite représenter, puis en suivant les étapes ci-dessus.

PROGRAMMATION DE LA RAM DE CARACTÈRES

Voici un exemple de caractère qui ressemble à une flèche vers la gauche :

Line	Character	Binary	Dec	Hex
1	`  *`	0 0 0 0 0 1	1	1
2	` *`	0 0 0 0 1 0	2	2
3	`*`	0 0 0 1 0 0	4	4
4	`*`	0 0 1 0 0 0	8	8
5	`*`	0 1 0 0 0 0	16	10
6	`******`	1 1 1 1 1 1	63	3F
7	`******`	1 1 1 1 1 1	63	3F
8	`*`	0 1 0 0 0 0	16	10
9	`*`	0 0 1 0 0 0	8	8
10	`*`	0 0 0 1 0 0	4	4
11	`*`	0 0 0 0 1 0	2	2
12	`*`	0 0 0 0 0 1	1	1

Les valeurs obtenues grâce à cette formule sont multipliées par 2, puis augmentées de 128 (en décimal) avant d'être intégrées à l'écran de programmation. Cette opération est nécessaire pour éviter l'affichage de caractères parasites « garbage » lors de la programmation de la carte 80-Grafix. Si vous souhaitez observer les valeurs transmises à la carte 80-GRAFIX par exemple à des fins de débogage, ne leur ajoutez pas 128 ; les caractères s'afficheront alors à l'écran, car seuls les caractères « graphiques » sont masqués.

Ce caractère est programmé et affiché par le programme N° 1, ci-dessous. En faisant varier les coordonnées X et Y, vous pouvez contrôler l'emplacement où le caractère s'affiche à l'écran. Il est possible de définir n'importe lequel des 64 caractères programmables en plaçant les 12 octets de données dans l'un des champs de la zone de programmation.

Vous pouvez consulter l’intégralité de votre jeu de caractères à tout moment en saisissant les instructions suivantes (sous forme de programme ou depuis le niveau de commande, sur une seule ligne) :

```
CLS
OUT 255,160
FOR C=1 TO 64
PRINT@(C-1)*8,C;CHR$(128+(C-1));
NEXT C
```

Cette routine efface l'écran, active le mode haute résolution « 80-GRAFIX » et affiche successivement à l'écran les 64 caractères graphiques, précédés de leur numéro. N'oubliez pas que la valeur ASCII réelle de chaque caractère est 127+C (CHR$ (127+C)).

PROGRAMMATION DE LA RAM DE CARACTÈRES

PROGRAM #1

```
10 C=1            :REM C=Character No.
15 X=1            :REM X=1 to 64 (horizontal)
20 Y=8            :REM Y=1 to 16 (vertical)
25                 REM X=Y=1 is upper-left corner
```

Voici les données relatives à la cellule de personnage :

```
30 DATA 1,2,4,8,16,63,63,8,4,2,1
35 POKE 16553,222        :REM correct data read
40 RESTORE:CLS
45 OUT 255,96    :REM Set board to Programming Mode
50 FOR Z=1TO12   :REM Get data & program 80-Grafix
55 READ D
60 POKE 15360+(C-1)*16+(Z-1),D*2+128
65 NEXT Z
70 OUT 255,160   :REM switch to Hi-Res Mode
                 :REM Display programmed chr at X,Y
75 POKE 15360+(X-1)+(Y-1)*16,128+(C-1)
80 END
```

Maintenant que vous avez saisi et exécuté le programme, vous voilà prêt à passer à des projets plus ambitieux et plus intéressants !

Le programme suivant (programme N° 2), baptisé « CREATE », est un utilitaire élémentaire qui vous permet de « dessiner » chaque caractère sur une grille agrandie, tout en visualisant son rendu à taille réelle. « CREATE » génère également les instructions de données (« DATA statements ») nécessaires pour intégrer ce caractère dans 80-GRAFIX. Une fois que le résultat vous convient, notez ces instructions de données et intégrez-les dans vos futurs programmes utilisant ce jeu de caractères. (Le programme « CREATE » est fourni sur cassette avec la carte 80-GRAFIX.)

PROGRAM #2

```
1000 '    "CREATE"
1010 '     (c)1980 by Ted Carter
1020 '     (With comments by Ralph Burris)
1030 '
1040 '    This program allows creation of Hi-Resolution
1050 '    Graphics characters when used in conjunction with
1060 '    80-GRAFIX.  It allows the user to design individual
1070 '    Programmable characters, and prints out a list of
          the DATA statements for each character.
1080 '
1090 '
1100 '    The techniques employed in this program may be used
1110 '    in other programs to reproduce the characters
          generated, using the data values shown as the
1120 '    finished character.
1130 '
1140 DIM A(6,12):     'Stores status of each grid block
                       (on/off)
1150 POKE 16553,222:  'Correct ROM Data read error
1160 FOR X=1 TO 6:    'Loop through grid
```

PROGRAMMATION DE LA RAM DE CARACTÈRES

```
1170    FOR Y=1 TO 12
1180      A(X,Y)=0:     'Set up empty grid values
1190    NEXT
1200 NEXT
1210 OUT 255,96: CLS:      'Set to Programming Mode
1220 FOR X=15360 TO 16383: 'Loop through screen
1230    POKE X,128:         'And set up blank chrs
1240 NEXT
1250 RESTORE: FOR X=1 TO 76:  'Loop through data statements
1260 READ D:                  'Get character value
1270 POKE 15375+X,D*2+128:    'Convert to graphic value
1280 NEXT:                    'and put on programming screen
1285 '
1290 '*** Programmable character data - draws grid
1295 '
1300 DATA 63,32,32,32,32,32,32,32,32,32,32,32,0,0,0,0
1310 DATA 63,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0
1320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1330 DATA 63,63,63,63,63,63,63,63,63,63,63,63,0,0,0,0
1340 DATA 63,0,0,0,0,0,0,0,0,0,0,0
1345 '
1350 ' The following lines draw the Programming Grid
1355 '
1360 OUT 255,160: 'Set to Hi-Resolution mode
1370 CLS:PRINT"      1 2 3 4 5 6":
1380 FOR Y=1 TO 12:IF Y<10 THENPRINT"  ";Y;" ";ELSE
     PRINT " ";Y;" ";
1390 FOR X=1 TO 6:IF A(X,Y)=0
     THENPRINTCHR$(129);CHR$(130);:GOTO 1410
1400 PRINTCHR$(132);CHR$(132);
1410 NEXT:PRINT:NEXT:PRINT"      ";
1420 FOR X=1 TO 12:PRINT CHR$(133);:NEXT:
     PRINT@418,CHR$(131);
1425 '
1430 ' The following allows input of grid location to change
1435 '
1440 PRINT@896,"ENTER X,Y OF DOT TO CHANGE        ";
1450 PRINT@922," ";:INPUTA,B
1460 IF A<1 OR A>6 OR B<1 OR B>12 THEN 1440
1465 '
1470 ' The following tests screen grid position for ON or OFF
1480 ' then changes value to opposite condition.
1485 '
1490 R=15360+4+2*A+64*B: IF PEEK(R)=132 THEN 1510
1500 POKE R,132: POKE R+1,132: A(A,B)=1: GOTO 1520
1510 POKE R,129: POKE R+1,130: A(A,B)=0
1520 PRINT@960,"CHARACTER DATA: ";
1525 '
1530 ' This section sets 80-GRAFIX to the "Programming Mode"
1535 '
1540 OUT 255,96: FOR X=1 TO 12
1545 '
1550 ' The following line reads values in A(X,Y) and converts
1560 ' to proper value for each of the 12 bytes in each
1570 ' programmable character.
1575 '
1580 R=(A(1,X)*32 + A(2,X)*16 + A(3,X)*8 + A(4,X)*4 +
```

PROGRAMMATION DE LA RAM DE CARACTÈRES

```
     A(5,X)*2 + A(6,X))*2 + 128
1590 POKE 15407+X,R:     'Load value into Programming screen
1595 '
1600 ' The following prints out data value
1605 '
1610 PRINT USING "###";R;: IF X<12 THEN PRINT ",";
1620 NEXT: PRINT@48,"";: OUT 255,160:PRINT"                ";
1630 OUT 255,160:       'Set to Hi-Resolution display mode
1640 PRINT"              ";
1650 GOTO 1440:   'Zero programming grid & get next character
```

En utilisant « CREATE » comme ligne directrice pour le développement de vos futurs programmes, vous constaterez que 80-GRAFIX est très simple à intégrer dans vos logiciels. Pour vous entraîner, pourquoi ne pas reprendre votre jeu graphique préféré, identifier les valeurs CHR$ qui servent à dessiner les images, puis créer votre propre jeu de caractères pour les remplacer ? Ajoutez une sous-routine à la fin du programme qui charge vos nouveaux caractères dans 80-GRAFIX, et jouez au jeu avec VOTRE propre jeu de caractères !

Vous pouvez obtenir une résolution de 192 par 192 pour votre nouvel ensemble de caractères en passant au mode « 32 caractères ». Comme vous le savez, ce mode double la largeur de chaque caractère et n'affiche qu'un caractère sur deux à l'écran. L'instruction BASIC `PRINT CHR$(23)` active le mode « 32 caractères » sur le TRS-80, tandis que `CLS` ou `PRINT CHR$(28)` permet de revenir au mode « 64 caractères ». Le mode « 32 caractères » n'a aucune incidence sur la programmation des caractères. Toutefois, lorsque vous utilisez `POKE` pour afficher l'ensemble des caractères, veillez à n'effectuer des opérations `POKE` qu'aux emplacements d'écran correspondant à des adresses paires, car les emplacements impairs ne sont pas affichés.

Pour plus d'informations sur la programmation, consultez le manuel d'utilisation du TRS-80 LEVEL II concernant l'emploi des fonctions PRINT@, POKE et CHR$, ainsi que la carte mémoire, le tableau des fonctions et les codes ASCII et graphiques.

3.3 TECHNIQUES EN LANGAGE ASSEMBLEUR

La programmation du 80-GRAFIX en langage assembleur (langage machine) est tout aussi simple que l'utilisation des méthodes en langage BASIC évoquées précédemment. À la fin de cette section, vous trouverez le listing complet d'une routine de pilotage vidéo (pilote vidéo) ; le code objet correspondant est fourni sur la cassette accompagnant votre carte 80-GRAFIX.

Tout d'abord, configurez un tampon de données (« Data Buffer ») contenant les informations relatives à chaque caractère que vous souhaitez programmer. Ce tampon est structuré de la même manière que dans un programme en langage BASIC ; il se compose d'un maximum de 64 « champs » de 16 octets chacun, dont les 12 premiers sont destinés à la carte 80-GRAFIX. (Notez que, si vous souhaitez économiser de l'espace, il est possible de placer des champs de 12 octets chacun sur l'écran de programmation, en veillant à ce que le premier octet de chaque champ commence à une adresse correspondant à un multiple de 16, ajouté à l'adresse de début de l'écran.)

Par exemple :

```
1000 BUFFER:  EQU     $        ;The following data
1010          DEFB    081H     ;is the same as that
1020          DEFB    082H     ;used for the "arrow"
1030          DEFB    084H     ;in part 3, above.
1040          DEFB    088H
1050          DEFB    090H
1060          DEFB    0BFH
1070          DEFB    0BFH
1080          DEFB    090H
1090          DEFB    088H
1100          DEFB    084H
1110          DEFB    082H
1120          DEFB    081H
1130          DEFB    080H     ;Following is "dummy"
1140          DEFB    080H     ;Data for filler.
1150          DEFB    080H
1160          DEFB    080H
1170; End of text buffer
```

Maintenant que la zone tampon a été définie, il est facile de « programmer » 80-GRAFIX avec ces données, en utilisant la technique suivante :

```
2000 START: LD   DE,03C00H      ;Point to screen
2010        LD   HL,BUFFER      ;Point to text
2020        LD   BC,16          ;Length of buffer
2040        LD   A,96           ;Program mode
2050        OUT  (255),A        ;Set Board
2060      - LDIR                ;Move data to screen
2070        LD   A,160          ;Set to HiRes Mode
2080        OUT  (255),A
2090        CALL 01C9H          ;ROM Clear screen
2100        JP   402DH          ;Return to DOS
```

La routine ci-dessus transfère le tampon de 16 octets vers l'écran de programmation et réinitialise la carte pour un affichage en haute résolution. Le

caractère programmé possède une valeur ASCII de 80H (128 en décimal). Une routine permettant d'afficher ce caractère pourrait se présenter comme suit :

```
3000 DSPLY: CALL 01C9H          ;Clear screen
3010        LD   A,80H          ;Character no.
3020        LD   (3C00H),A      ;Put chr on screen
3040        RET                 ;Return to caller
```

Cette routine effaçait l'écran et y plaçait le caractère ASCII numéro 128 (celui que nous avions programmé) dans le coin supérieur gauche.

La routine de gestion vidéo qui suit intègre ces techniques pour vous offrir un jeu de caractères en minuscules. Elle comprend également d'autres éléments intéressants, tels que l'installation d'un nouveau pilote vidéo, ainsi qu'un mécanisme d'autoprotection qui place la routine dans la zone supérieure de la mémoire et en informe le TRS-80, évitant ainsi qu'elle ne soit « écrasée » par d'autres opérations informatiques.

*** NOTE ***

Lorsque vous utilisez le pilote vidéo sur votre machine, toutes les saisies en majuscules (effectuées avec la touche Maj) s'affichent sous forme de caractères majuscules, tandis que toutes les saisies en minuscules (sans la touche Maj) s'affichent selon le jeu de caractères minuscules « GRAPHICS » fourni. Lorsque vous utilisez ce pilote, des minuscules s'affichent à l'écran ; toutefois, gardez à l'esprit qu'il ne s'agit pas de minuscules au sens habituel, mais d'éléments graphiques. Par conséquent, si vous basculez vers le mode graphique standard (via la commande `OUT 255, 32`), l'affichage deviendra illisible (caractères incohérents). Cela signifie également que cet ensemble de minuscules n'est pas compatible avec des logiciels tels que « The Electric Pencil », « Scripsit » ou tout autre traitement de texte conçu pour la saisie en minuscules. En revanche, les programmes en BASIC que vous rédigez resteront fonctionnels sur n'importe quelle machine, car les caractères saisis demeurent inchangés. Si vous exécutez un tel programme sur une machine dépourvue de la fonctionnalité « 80-GRAFIX » — ou sans ce pilote vidéo — les caractères s'afficheront normalement, sous forme de majuscules.

Pour plus d'informations sur la programmation en langage assembleur, les ouvrages suivants sont recommandés :

TRS-80 Assembly Language Programming
by Wm Barden
RS Catalog # 62-2006

Z-80 Microcomputer Handbook
by Wm Barden
Howard W. Sams & Co.

Z-80 Assembly Language Programming Manual
from Zilog

```
;       *** Keyboard driver for 80-GRAFIX ***
SCREEN    EQU   3C00H
          ORG   7000H
START     LD    A,96      ;Program Characters mode
          OUT   (255),A
          LD    HL,TXT    ;Letter text buffer
          LD    DE,SCREEN
          LD    BC,BOTTOM-TXT
          LDIR            ;Move data to screen
; Following section displays character set & jumps
; to DOS READY.
SHOW      LD    A,160
          OUT   (255),A     ;Set to Display mode
          CALL  01C9H       ;ROM Call to clear screen
          LD    A,128
          LD    DE,SCREEN
NEXT      LD    (DE),A
          INC   A
          INC   DE
          CP    '*'         ;Compare for end of text
          JR    NZ,NEXT
          LD    BC,LAST-ENTRY
          LD    HL,(4049H)  ;Use 40B1H for LV2
          SBC   HL,BC       ;compute length of driver
          PUSH  HL
          DEC   HL
          LD    (4049H),HL  ;Protect driver (40B1H=LV2)
          POP   DE
          LD    HL,ENTRY
          LD    (401EH),HL  ;Install new Video Driver
          LDIR              ;and move it to new loc.
          JP    402DH       ;Ret to DOS (06CCH =LV2)
TXT       EQU   $
; User text buffer area
; Fill with data statements to create chr set
;
; NOTE: Data in groups of 16 bytes - up to 64
;       blocks may be included.
BOTTOM    DEFB  '*'         ;End of text marker
; Video driver - converts upper to lower
;                & lower to grafix
ENTRY     LD    L,(IX+03)     ;Same as ROM routine
          LD    H,(IX+04)
          JP    C,049AH
          LD    A,(IX+05H)
          OR    A
          JR    Z,AGN
          LD    (HL),A
AGN       LD    A,C
          CP    80H           ;Compare for grafix value
          JP    NC,04A6H      ;Display graphic chr
          CP    20H           ;Compare for control chr
          JP    C,0506H       ;Do control code
          CP    41H           ;Compare for numeric chr
          JP    C,047DH       ;Do normal display
          CP    60H           ;Compare for lower case
          JR    NC,FIX        ;Go if lower case
```

```
        ADD     A,3FH           ;Bump to graphics value
        JP      047DH           ;Back to display routine
FIX     SUB     20H             ;Make lower case into upper
LAST    JP      047DH           ;Back to display routine
        END     START           ;Auto start
```

Le programme pilote précédent peut être utilisé pour n'importe quel jeu de caractères que vous souhaitez définir. Par exemple, si vous souhaitiez programmer en APL, vous pourriez créer le jeu de caractères APL et insérer les instructions de données comme indiqué ci-dessus. Le programme pilote affichera alors le jeu de caractères APL au lieu des minuscules (les caractères obtenus avec la touche « Maj » restent, bien entendu, inchangés).

3.4 PROGRAMMATION D'ANIMATIONS GRAPHIQUES

Pour donner l'impression qu'un objet se déplace sur un plan horizontal à travers l'écran, il est nécessaire de programmer jusqu'à 12 caractères différents afin de représenter ce mouvement. Comme chaque caractère ne peut être déplacé que de 6 points horizontalement (ou 12 points verticalement), il faut utiliser six ensembles composés chacun de deux cellules de caractères pour représenter chaque étape du déplacement ; la configuration des points dans chaque cellule est décalée d'un point vers la droite (ou vers la gauche). Chaque paire de cellules est affichée au même emplacement à l'écran jusqu'à ce que les six paires aient été présentées ; on passe ensuite à l'emplacement suivant, en procédant de la même manière, jusqu'à ce que l'objet ait « traversé » tout l'écran. Cette méthode permet d'obtenir des affichages très convaincants, même en utilisant le langage BASIC pour gérer l'affichage.

Le même procédé peut être utilisé pour déplacer des objets dans n'importe quelle direction sur l'écran. Pour les objets se déplaçant rapidement, ou lorsqu’une résolution moindre suffit, il n'est pas nécessaire de procéder par déplacements point par point ; vous pouvez déplacer la configuration de points de deux positions ou plus au sein de chaque caractère, ce qui permet d'utiliser moins de caractères pour chaque déplacement.

Dans les programmes en langage assembleur, il est possible de créer un ensemble initial de caractères, puis de générer des séquences d'« animation » par le biais de décalages de bits (« bit shifting »). Par exemple, si l'objet que vous souhaitez animer nécessite, disons, 12 octets pour être représenté (comme un vaisseau spatial), les 12 premiers caractères peuvent constituer la forme de base de ce vaisseau. Votre programme en assembleur peut alors lire chacune des 144 données initiales (à raison de 12 par caractère) et appliquer une opération de décalage arithmétique vers la gauche (« Shift Left Arithmetic » – SLA) ou vers la droite (« Shift Right Arithmetic » – SRA) sur chacun de ces octets, en plaçant les nouvelles valeurs obtenues dans les emplacements mémoire situés à la suite du groupe initial. Cette procédure peut être répétée pour chaque groupe de caractères, jusqu'à ce que le nombre souhaité soit atteint. Il est même possible d'effectuer cette opération « à la volée » — en générant un nouvel ensemble de caractères entre chaque affichage — afin de permettre à un grand nombre d'éléments de se déplacer simultanément à l'écran.

3.5 AUTRES IDÉES DE PROGRAMMATION

Outre les méthodes abordées dans la section précédente, le programmeur en langage assembleur dispose d'un large éventail de possibilités pour interagir avec le 80-GRAFIX. Par exemple, en utilisant les techniques de décalage de bits mentionnées plus haut, il est possible d'exploiter le bit de retenue (« carry bit ») pour programmer dynamiquement le caractère suivant dans une séquence d'animations. On peut également créer un jeu de caractères inversés en lisant les données initiales et en effectuant une opération de complément (instruction CPL) sur chaque octet, avant de le stocker à son nouvel emplacement (ou à sa position précédente). Le 80-GRAFIX permet d'explorer la logique binaire dans toute sa profondeur, car chaque cellule de caractère constitue une représentation fidèle des valeurs binaires transmises à la carte.

Le programmeur en langage BASIC peut employer des techniques similaires pour réorganiser les valeurs des données. Supposons, par exemple, que vous ayez défini un tableau destiné à contenir les valeurs des bits pour chaque ligne de données, comme dans le programme « CREATE » présenté plus haut. En utilisant l'instruction NOT, il est possible d'obtenir un résultat identique à celui de l'opération CPL de l'exemple précédent, de la manière suivante :

```
10 ' A(X) array stores a "1" or "0" for bit value
20 FOR X=1TO72
30 A(X)=2+NOT A(X)
40 NEXT X
```

Dans cette routine, on vérifie si la valeur du « bit » de chaque cellule est activée (1) ou désactivée (0). Si le bit est activé (1), la fonction NOT à la ligne 30 renvoie la valeur -2 ; en y ajoutant 2, on obtient zéro. Si le bit est désactivé (0), la fonction NOT renvoie la valeur -1 ; en y ajoutant 2, on obtient un. Ce tableau est traité de la même manière que dans « CREATE », et la nouvelle valeur obtenue est utilisée pour programmer 80-GRAFIX.

Une routine similaire, destinée à produire un effet de décalage de bits vers la gauche, pourrait prendre cette forme :

```
10 FOR X=1TO 12:          'Twelve rows
20   FOR Y=1TO6:          'Six columns
30      V=(X-1)*6+Y:      'Actual array position
40      A(V)=A(V+1):      'Shift the value left
50   NEXT Y:              'Do the row
60   A(V)=0:              'Zero right position in row
70 NEXT X
```

Comme vous pouvez le constater, la valeur de chaque position située plus haut dans le tableau est décalée d’un cran vers le bas (ou récupérée depuis la position située à droite dans chaque « ligne »).

AUTRES IDÉES DE PROGRAMMATION

Pour effectuer un décalage vers la droite, cette routine pourrait être utilisée :

```
10 FOR X=1TO12
20   FOR Y=6TO2 STEP-1
30     V=(X-1)*6+Y
40     A(V)=A(V-1)
50   NEXT Y
60   A(V)=0
70 NEXT X
```

Voici un court programme que vous pouvez exécuter et qui illustrera ces idées :

```
10 DEFINT X-Z:                  'Use integers
15 DIM A(73):                   'Dimension array
20 FOR X=1TO72:                 'Set up random bit pattern
25 A(X)=(RND(2))-1:             'As "1" or "0"
30 NEXT X
35 FOR X=1TO12:FOR Y=1TO6:      'Show bit values
40 PRINT A((X-1)*6+Y);
45 NEXTY
50 FOR Y=6TO2 STEP-1:           'Shift right
55 V=(X-1)*6+Y
60 A(V)=A(V-1)
65 NEXTY
70 A(V-1)=0:                    'Zero left row position
75 FOR Y=1TO6:                  'Print the new values
80 V=(X-1)*6+Y
85 PRINT A(V);
90 NEXTY:PRINT
95 NEXT X
```

Cette routine indique la valeur de chaque ligne et colonne, décale la valeur vers la droite et affiche les nouvelles valeurs pour la ligne.

QUELQUES RÉFLEXIONS POUR CONCLURE

3.6 QUELQUES RÉFLEXIONS POUR CONCLURE

À ce stade, nous sommes certains que vous avez la tête qui tourne, remplie d'idées et d'applications pour les nouvelles capacités de votre 80-GRAFIX, et franchement, nous avons hâte de voir quelles idées vous allez concevoir !

Voici quelques-unes des idées imaginées par Ted Carter, le concepteur de 80-GRAFIX :

Afficher les notes de musique
Comparaisons de synchronisation IC
Art informatique
Dispositions des bâtiments
Conception automobile
Graphiques
Caractères spéciaux (Algol, APL, grec, etc.)
Souligner les mots
Personnages vidéo inversés
Caractères minuscules
... et ainsi de suite.

Le reste dépend de vous !

FIN DE LA DOCUMENTATION 80-GRAFIX.

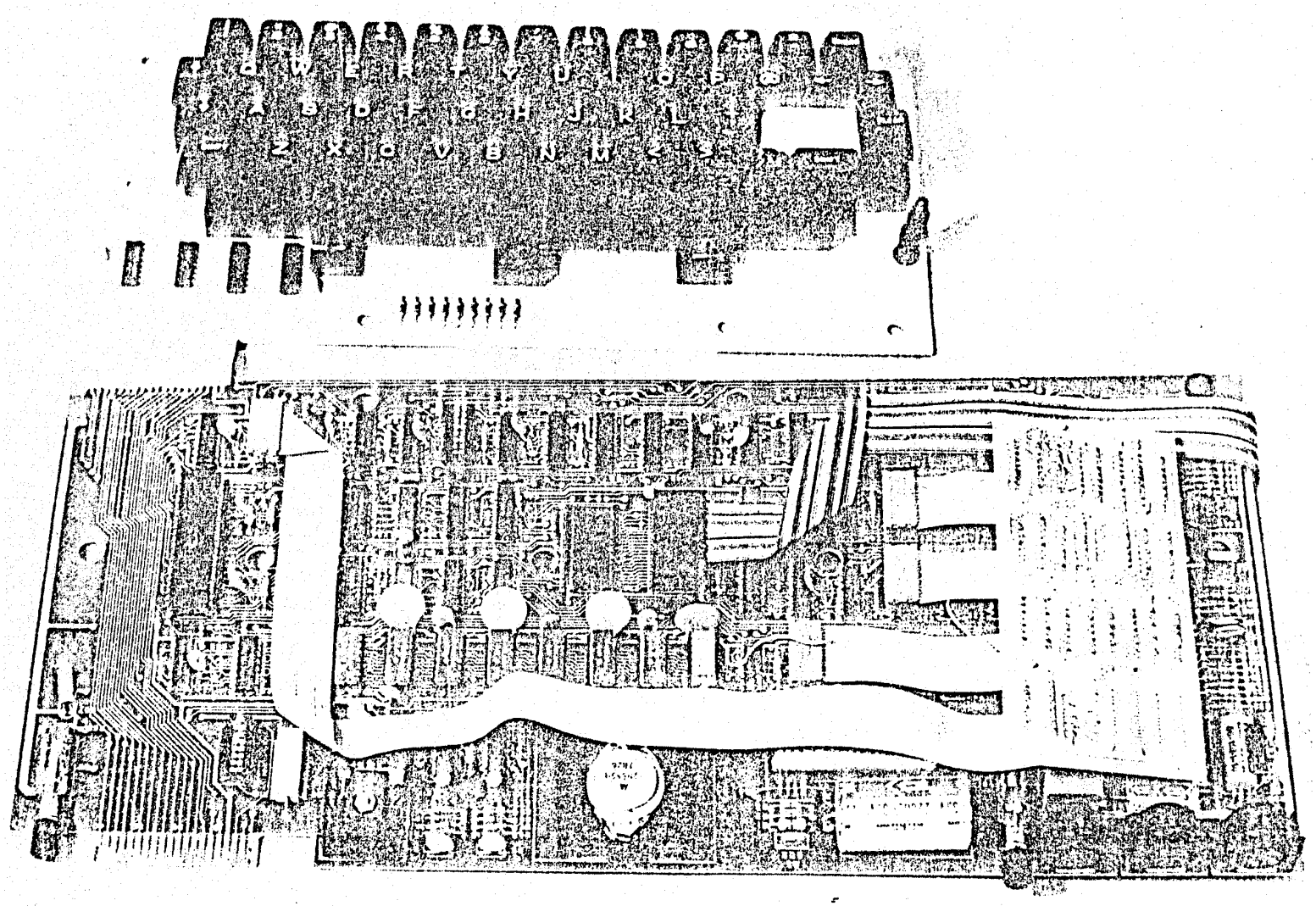


Figure 1

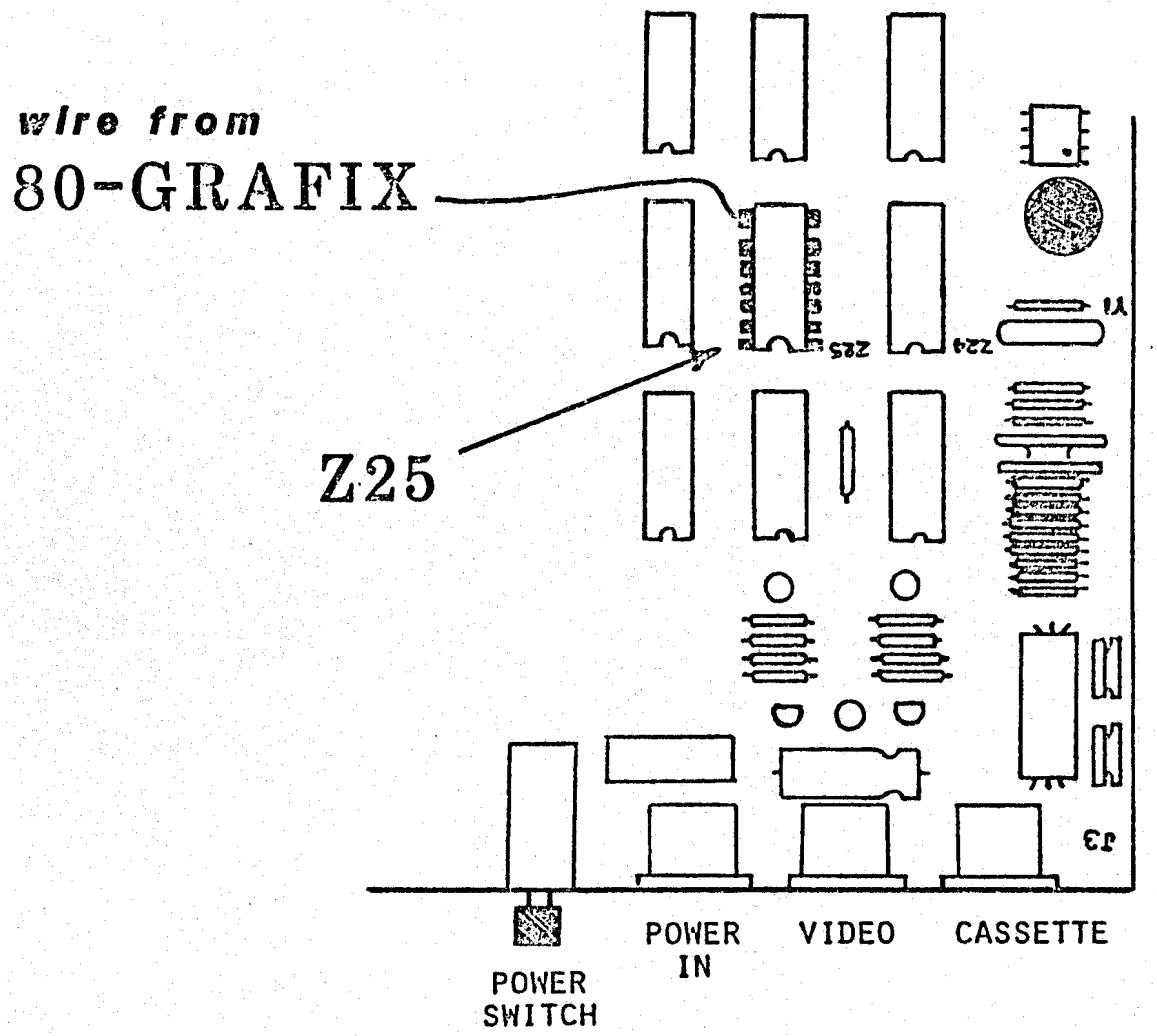

wire from
80-GRAFIX
Z25
Z25
Z24
Y1
J3
POWER
SWITCH
POWER
IN
VIDEO
CASSETTE

Figure 2

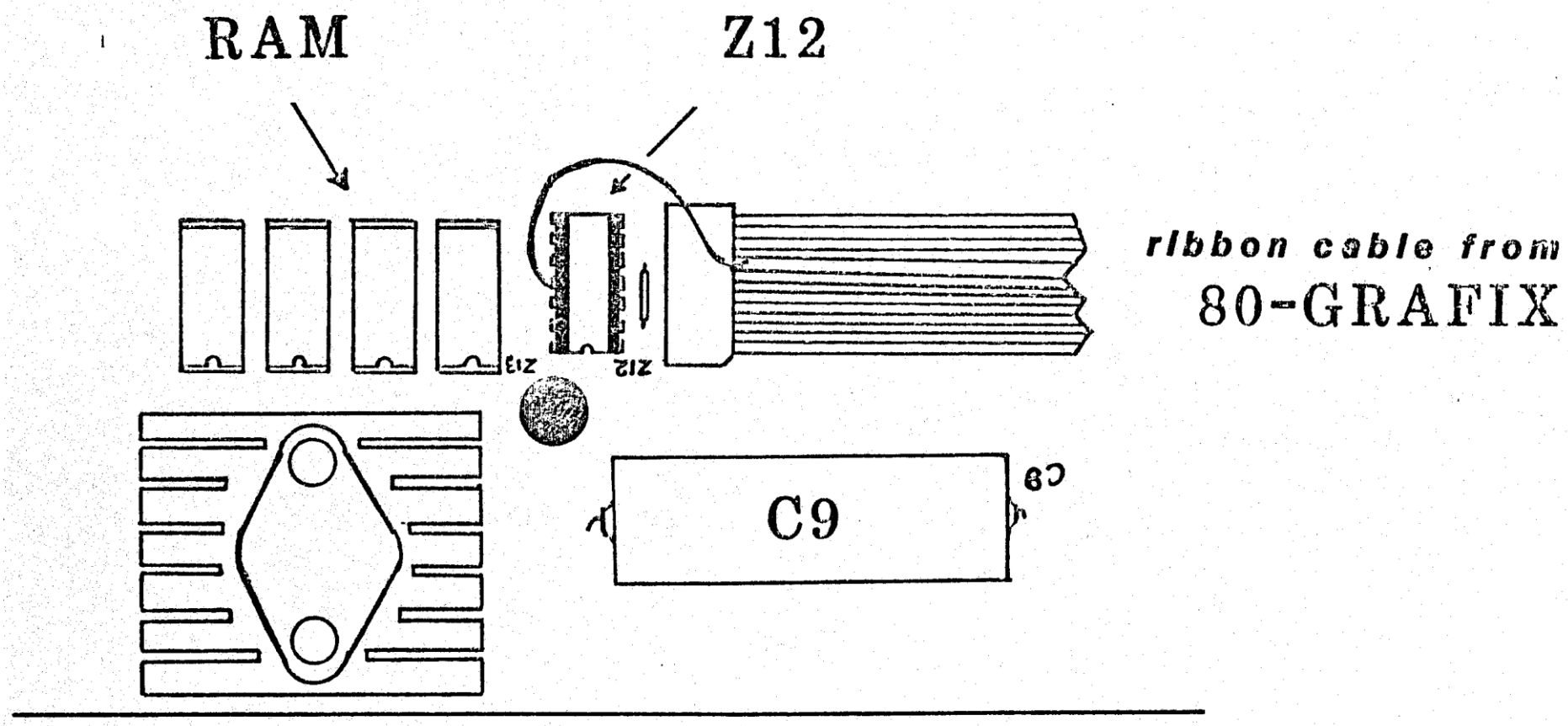


Figure 3