

PROGRAMMA

80-GRAFIX

GUIDE DE REFERENCE DE L'UTILISATEUR (traduction automatique)

80-GRAFIX
80-GRAFIX

Développé par :
Ted Carter

Distribué exclusivement par :
PROGRAMMA International, Inc.
3400 Wilshire Boulevard
Los Angeles, CA 90010

COPYRIGHT (C) 1980

Tous droits réservés, aucune partie de ce manuel ou de l'accompagnement programmes d'ordinateur, peuvent être reproduits sous quelque forme que ce soit, ou par moyens électroniques ou mécaniques, y compris le stockage d'informations et systèmes d'extraction, sans le consentement écrit et la permission PROGRAM International, Inc.

GARANTIE LIMITÉE

PROGRAMMA n'a aucune responsabilité envers l'acheteur ou toute autre personne ou entité à l'égard de toute responsabilité, perte ou les dommages causés ou présumés être causés directement ou indirectement par ce produit, y compris, mais sans s'y limiter, toute interruption de service, perte d'exploitation et bénéfices anticipés ou dommages indirects résultant de l'utilisation ou du fonctionnement de ce produit. Ce produit sera réparé ou remplacé à notre discrétion dans les quatre-vingt-dix (90) jours suivant la date d'achat s'il s'avère défectueux dans la fabrication, l'étiquetage ou l'emballage, à l'exception de la réparation ou le remplacement la vente ou l'utilisation ultérieure de ce produit est sans garantie ni responsabilité.

PROGRAMMA et ses auteurs ne sont pas responsables des dommages causés à matériel, logiciels ou données en raison de ses produits autres que remplacement du dit produit. À l'exception d'un tel remplacement, le vente ou toute manipulation ultérieure de ces produits est sans garantie ou responsabilité même en cas de défaut, de dommage ou de perte par négligence ou autre faute.

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

80-GRAFIX

Table des matières

80-PROGICIEL GRAFIX	
Description.....	1
80-GRAFIX DOCUMENTATION	
Bienvenue.....	4
1.0 INSTALLATION.....	5
1.1 DÉPANNAGE DE VOTRE INSTALLATION.....	9
2.0 THÉORIE DE FONCTIONNEMENT.....	11
3.0 UTILISATION DU 80-GRAFIX.....	13
3.1 CONTRÔLE DU MODE.....	13
3.2 PROGRAMMATION DE LA RAM DE CARACTÈRES.....	14
Programme #1.....	16
Programme #2.....	16
3.3 TECHNIQUES DU LANGAGE D'ASSEMBLAGE.....	19
Lecture recommandée.....	20
Pilote de clavier pour 80-GRAFIX.....	21
3.4 PROGRAMMATION DE GRAPHIQUES ANIMÉS.....	23
3.5 AUTRES IDÉES DE PROGRAMMATION.....	24
3.6 QUELQUES RÉFLEXIONS DE FIN.....	26

Liste des illustrations

- Fig. 1 VUE D'ENSEMBLE - Installation
- Fig. 2 ENSET - Connexion à souder Étape 7
- Fig. 3 INSET - Connexion à souder Étape 10

80-GRAFIX

PROGICIEL 80-GRAfix

Face 1

Programmes BASIC : utilisez « CLOAD »

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

Face 2

1. LOCASE - SYSTEM <CR> *?LOCASE <CR> *?/ <CR>
2. LCDOS -- Transfert sur disque à l'aide de TAPEDISK, etc.
START=7000H / END=71FAH / TRA=7000H

Les programmes Face 1 sont des exemples de méthodes de programmation dans **BASIC**.

Les programmes de la face 2 sont des jeux de caractères en minuscules avec vidéo Pilotes ajoutés pour permettre une « vraie » utilisation du clavier de la machine à écrire (c'est-à-dire que les minuscules s'affichent pour les entrées normales, les majuscules (Avec la touche majuscule).

80-GRAFIX

DOCUMENTATION

Bienvenue dans le monde passionnant de 80-GRAFIX!

Votre TRS-80 (tm) se transformera en un nouveau système puissant, doté de capacités graphiques bien au-delà de vos rêves après avoir installé votre carte 80-GRAFIX. De nouveaux mondes d'aventures de programmation attendent vos doigts dansants !

Dans les pages suivantes, nous explorerons les différentes techniques utilisables pour programmer les nouveaux graphiques haute résolution de votre ordinateur et examinerons certaines théories de fonctionnement applicables à la carte 80-GRAFIX. Vous découvrirez ensuite quelques exemples de programmation concrète que vous pourrez utiliser dans vos propres travaux, notamment quelques approches du Basic Niveau II (et Disque), ainsi que des listes de sources assembleur pour les plus aventureux.

Le projet 80-GBAFIX™ est si nouveau et radicalement différent de tout ce qui était auparavant disponible pour le TRS-80™, que nous n'avons fait qu'effleurer ses possibilités. En tant que pionniers dans ce domaine, nous attendons vos découvertes.

Ralph Burris
for PROGRAMMA INTERNATIONAL
July, 1980

INSTALLATION AVANT DE PRENDRE VOTRE TOURNEVIS...

Comme vous le savez, si vous n'avez jamais ouvert votre clavier auparavant, cela annulera votre garantie limitée de 90 jours et pourrait contrarier votre centre de réparation en cas de problème. Il est donc judicieux d'évaluer vos compétences en soudure avant de commencer. Si vous ne vous sentez pas à la hauteur pour réaliser deux soudures, pensez à faire appel à une personne expérimentée, ou à confier cette tâche à un technicien en électronique ou à un atelier de réparation. Envie d'essayer ? C'est parti...

Commencez par dégager un espace de travail de la taille d'une table à cartes. (Il est conseillé d'utiliser une surface douce et non statique pour éviter de rayer les protections en plastique et le clavier.) Ensuite, rassemblez les outils suivants :

- 1 Tournevis cruciforme
- 1 paire de petites pinces coupantes/digues
- 1 paire de petites pinces à bec effilé
- 1 Fer à souder à pointe de 15 watts, 3/32"
- 1 tube de ciment Duco ou similaire (facultatif)
- 1 magnifying glass

Retirez toutes les prises, placez l'unité de clavier sur l'espace de travail et procédez comme suit :

1. Retournez le clavier : localisez et retirez les six (6) vis situées sous le clavier. (L'une d'elles est délicate : elle traverse la grille et peut être recouverte d'un autocollant ou de la pellicule de garantie.)
2. Remettez le clavier à l'endroit avec précaution et retirez le capot supérieur.
3. Soulevez la carte du clavier et laissez-la pivoter vers vous. ATTENTION à ne pas tordre ni casser le câble du connecteur en bas à gauche. Retirez les cinq entretoises blanches sous le clavier (et notez leur emplacement d'insertion).
4. Retirez l'ensemble du circuit imprimé et du clavier du boîtier inférieur, puis placez le composant de la carte PC face vers le haut, le clavier au-dessus.

5. Localisez le circuit intégré #Z-11. Ce circuit intégré peut être câblé ou bien sur support, selon la version de votre carte. Si le circuit intégré est sur support, soulevez-le délicatement (saisissez-le par les extrémités et tirez doucement). Si le circuit intégré est câblé, localisez la broche 16 et COUPEZ cette broche avec une pince, en veillant à ne couper aucune piste ni aucune autre broche du circuit intégré.
6. Pliez les câbles plats de la carte 80-GRAFIX de manière à ce qu'ils se replient sur le dessus des connecteurs de la carte, puis posez la carte sur les composants TRS-80, côté circuit vers le haut et les câbles plats sortant par le côté gauche.
7. Localisez le fil isolé dépassant de la carte 80-GRAFIX et le circuit intégré #Z-25 sur la carte mère du TRS-80. En prenant grand soin de ne pas surchauffer le circuit intégré, soudez le fil isolé à la broche 8 du #Z-25 (la broche la plus à gauche). Vérifiez cette connexion à la loupe pour vous assurer qu'aucun pont de soudure n'a été créé accidentellement, ce qui pourrait court-circuiter les pistes ou une broche à une autre, et que la connexion est solide.
8. Placez maintenant le connecteur DIP FL1, sur le CI #Z-47. Déposez une goutte de colle (ou de ruban adhésif double face) sur le CI et appuyez fermement le connecteur sur le CI, en vous assurant que toutes les broches sont en contact avec leurs broches respectives et qu'elles ne touchent aucune autre broche. Si de la colle est utilisée pour maintenir la connexion, assurez-vous que le connecteur affleure le haut du CI lorsqu'il est enfoncé. Il sera nécessaire de couper légèrement les extrémités des broches du connecteur si la connexion n'est pas affleurante.

Lors de tous nos tests, ce type de connexion s'est avéré fiable. Il est donc déconseillé de souder les connecteurs directement aux circuits intégrés, car cela les rend difficiles à retirer et risque d'endommager le circuit en cas de surchauffe. Bien entendu, si vous avez une bonne expérience de ce type de travail, vous préférerez peut-être souder des supports compacts aux circuits intégrés TRS-80 et y brancher les connecteurs. (La même procédure qu'à l'étape 8 est appliquée pour fixer les autres connecteurs de nappe.)

Placez le connecteur DIP FL2 sur le CI #Z-29 de manière à ce que les broches n° 1 et n° 18 ne soient pas connectées. Vous remarquerez que le CI #Z-29 est un CI à 18 broches, et que notre connecteur n'en comporte que 16. Il est donc décentré, de sorte que les deux broches du CI les plus proches sont déconnectées.

9. Le connecteur DIP FL3 (qui comporte deux broches manquantes (6 et 16) et un fil qui dépasse) est connecté au CI #Z11. (Si votre TRS-80 possède un support pour cette puce, insérez le connecteur dans le support.) Localisez la broche 11 du CI #Z-12 (broche centrale à gauche). En suivant les précautions mentionnées précédemment, soudez le fil du câble de connexion à la broche 11 et vérifiez votre travail pour déceler les erreurs habituelles.
10. Le connecteur DIP FL4 (câble le plus long) est maintenant fixé au CI #Z75, sur le côté opposé de la carte.

Comparez maintenant attentivement l'installation terminée avec l'illustration ci-jointe. Si tout est conforme, 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 desserrer les connecteurs lors du remontage du clavier.

Maintenant vient le plaisir (on l'espère !).

Branchez les câbles d'alimentation et vidéo, et démarrez ! Si tout semble normal à l'écran (par exemple, « Taille de la mémoire » s'affiche), nous sommes prêts ! En cas d'imprévu, coupez rapidement l'alimentation et passez à la section « Dépannage de votre installation ».

Vous pouvez maintenant tester 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 des caractères TRS-80, suivis des caractères graphiques standard (également deux fois). Lorsque vous appuyez sur la touche « ENTRÉE », le mode est réglé sur haute résolution et les caractères graphiques sont remplacés par des bandes verticales ou un ensemble de petits points clairs et foncés aléatoires. (Si ce n'est pas le cas, éteignez l'appareil et consultez la section « Dépannage ».)

Les nouveaux caractères à l'écran affichent le contenu des 64 caractères programmables. Ils ne sont pas identifiables car ils n'ont pas été programmés, mais ce test indique que la carte est installée et fonctionne correctement.

Si vous êtes opérationnel, allez à la section 2.0.

1) DÉPANNAGE DE VOTRE INSTALLATION

Votre carte 80-GRAFIX a été rodée et testée avant de quitter l'usine. Par conséquent, si vous rencontrez des problèmes de fonctionnement, il faut d'abord suspecter un problème d'installation.

Voici quelques problèmes possibles et leurs causes probables :

1. ÉCRAN TOTALEMENT NOIR À LA MISE SOUS TENSION

Cela peut être un problème grave. Tout d'abord, vérifiez que le moniteur est connecté et fonctionne correctement. Si le problème persiste, cela indique probablement que deux ou plusieurs broches du circuit imprimé TRS-80 sont en court-circuit. Un tel court-circuit accidentel pourrait endommager d'autres composants de votre ordinateur. Vérifiez soigneusement chaque connexion à la loupe et recherchez les ponts de soudure et les connecteurs pliés. L'utilisation d'un ohmmètre pour vérifier l'intégrité des circuits est également recommandée.

2. ÉCRAN GRIS SCINTILLANT (pas d'image)

Ce problème est probablement dû au connecteur du circuit intégré #Z-11. Assurez-vous que les broches appropriées sont bien fixées et fermement en contact.

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

Vérifiez la connexion au circuit intégré #Z-12 pour vous assurer qu'elle est correctement installée sur la broche appropriée et fermement fixée. Le problème suivant est probablement la connexion au circuit intégré #Z-11.

4. POINTS ÉTINCELANTS ALÉATOIRES SUR L'ÉCRAN

Suivez la connexion jusqu'au CI #Z-29 pour une installation correcte et un bon contact.

5. LES GRAPHIQUES DE L'ANCIEN TRS-80 S'AFFICHENT PENDANT LE TEST PROGRAMME

La connexion à un seul fil à la broche 8 du CI #Z-25 ou la connexion au CI #Z-75 est probablement incorrecte ou n'a pas établi un bon contact.

(Chaque fois que la carte 80-GRAFIX ne parvient pas à afficher correctement les caractères programmés, n'efface pas les caractères graphiques à l'écran en mode programmation ou affiche des graphiques normaux en mode haute résolution, ces connexions sont généralement les coupables.)

6. REPRODUCTION INCORRECTE DES CARACTÈRES PROGRAMMÉS

Si un programme de génération de caractères (tel que « CREATE », inclus avec la carte 80-GRAFIX) ne parvient pas à renvoyer les caractères créés, vérifiez la connexion de l'IC #Z-47 pour une installation correcte du contact.

En général, vérifiez soigneusement chaque connexion pour vous assurer qu'elle est au bon endroit, que les connecteurs sont bien en place et que les deux fils isolés sont correctement soudés.

Si vous ne parvenez toujours pas à identifier et à corriger le problème, écrivez à l'usine et décrivez-lui en détail tous les aspects du problème. Si vous identifiez et corrigez toutes les erreurs d'installation et que 80-GRAFIX ne fonctionne toujours pas correctement, il est possible que votre erreur ait endommagé la carte 80-GRAFIX, le TRS-80, ou les deux. Si tel est le cas, assurez-vous de décrire dans votre courrier toutes les opérations d'installation de la carte.

2) THÉORIE DE FONCTIONNEMENT

Si vous possédez un exemplaire du « Manuel de référence technique du micro-ordinateur TRS-80 » (Radio Shack Cat. N°26-2103), vous trouverez des descriptions détaillées des différents composants abordés dans cette section.

La section vidéo de votre TRS-80 se compose d'un bloc de 1 Ko de RAM vidéo, d'une ROM de caractères (MCM6670) qui génère les caractères alphanumériques et d'un multiplexeur (74153) qui permet de générer des graphiques bruts avec une résolution de 128 caractères horizontaux sur 48 caractères verticaux. Chaque caractère alphanumérique est stocké dans la ROM de caractères sous forme de motif de bits (1 ou 0) dans une matrice de 5 caractères horizontaux et 7 caractères verticaux. Les caractères graphiques sont créés différemment et produisent des « cellules » de 2 caractères horizontaux sur 6 caractères verticaux, mais utilisent les 12 lignes de balayage de chaque ligne. Le 80-GRAFIX fournit une mémoire graphique alternative qui offre au mode graphique une sélection de motifs de points similaires aux motifs alphanumériques, à la différence que chaque caractère est une grille de 6 caractères horizontaux sur 12 caractères verticaux. (Plus d'informations à ce sujet ultérieurement.)

Le TRS-80 mélange des caractères alphanumériques et graphiques à l'écran et sait quel type afficher en vérifiant l'état du bit de poids fort (MSB) dans chacun des 1024 octets de la RAM vidéo. Si le NSB est à l'état haut (activé), un caractère graphique est affiché. S'il est à l'état bas (désactivé), un caractère alphanumérique est affiché. Par ailleurs, la différence entre l'état activé et désactivé (0 ou 1 binaire) a pour effet d'ajouter ou de soustraire 128 à la valeur de chaque octet. Les valeurs graphiques vont de 128 à 191 en décimal (80 à BF Hex).

La mémoire vidéo du TRS-80 comprend 1024 emplacements (octets), situés aux adresses RAM 15360 à 16383 (3C00 à 3FFF Hex), et sont transmis au moniteur vidéo sous forme de 16 lignes de 64 ou 32 caractères. Ces emplacements mémoire sont accessibles en Basic avec les instructions PEEK, POKE et PRINT. Pour plus d'informations sur les fonctions ci-dessus et l'utilisation des valeurs CHR\$, consultez le Manuel Basic de niveau II. Consultez également la table des fonctions, le tableau des fonctions et les annexes des codes ASCII et graphiques pour plus de précisions sur l'adressage de l'écran.

Il existe une différence entre la RAM vidéo TRS-80 normale et le reste de la mémoire RAM : chaque octet ne comporte que 7 bits ! Le 8 - ème bit a été supprimé. Ainsi, les caractères de valeur inférieure à 32 qui sont insérés dans la RAM renvoient une valeur PEEK avec 64 bits ajoutés, et les valeurs supérieures à 191 se voient soustraire 128 bits lors de l'insertion PEEK. Si vous avez installé un kit de minuscules Radio Shack, le 8e bit est rétabli et la RAM vidéo se comporte comme les autres. (D'autres types de kits de minuscules font la même chose : certains disposent d'un commutateur de dérivation pour ce bit supplémentaire, d'autres le contrôlent par logiciel.) Cette configuration de mémoire n'a pas d'effet particulier sur l'utilisation de 80-GRAFIX, mais elle mérite d'être prise en compte pour les futures applications de programmation.

L'installation de la carte 80-GRAFIX permet d'étendre l'écran graphique à une résolution de 384 pixels à l'horizontale et 192 pixels à la verticale, ce qui est possible, comme mentionné précédemment, grâce à des cellules graphiques de 6 x 12.

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

1. Affichage normal : affiche les caractères graphiques TRS-SO standard.
2. Haute résolution : affiche le jeu de caractères programmé dans 80-GRAFIX.
3. Mode programmation : permet à 80-GRAFIX d'accepter les données de caractères programmables et de générer un jeu de caractères.

La carte 80-GRAFIX intègre un bloc mémoire de 1 Ko (octet) sur 6 bits, servant de RAM de caractères programmables (PCR). À partir de cette mémoire, la RAM vidéo appelle les cellules de caractères, comme il le fait avec la ROM de caractères MCM6670. La PCR est organisée en 64 groupes (appelés « Champs »), qui représentent le jeu de caractères programmables. Chaque champ PCR comporte 16 octets, dont les 12 premiers contiennent les données de chaque ligne du caractère (les 4 derniers octets de chaque champ sont ignorés). En mode « Programmation », le 80-GBAFIX utilise la RAM vidéo comme tampon de données de programmation. Dans ce mode, les données de caractères utilisateur sont transférées à l'écran, puis lues comme l'image de la PCR et copiées dans la mémoire PCR. Pour éviter que trop de données inutiles n'apparaissent à l'écran, tout caractère dont la valeur est égale ou supérieure à 128 est masqué.

Les caractères programmables 6 x 12 de la carte 80-GRAFIX permettent d'accéder à chaque point de chaque ligne de chaque caractère. Cela permet d'obtenir des graphiques continus, car chaque point de la matrice binaire correspond à un point sur l'écran vidéo. Lorsqu'une cellule graphique est visualisée sur l'écran, le NSB est situé à l'extrême gauche et le LSB à l'extrême droite de chaque ligne de balayage de la cellule.

3) UTILISATION DE LA CARTE 80-GRAFIX

3.1 CONTRÔLE DU MODE

Comme mentionné dans le chapitre précédent, La carte 80-GRAFIX dispose de trois modes de fonctionnement, sélectionnés en envoyant une valeur au port cassette (Port 255 décimal, FF hexadécimal). Les trois bits de poids fort, 5, 6 et 7, servent de bits de contrôle lorsque cette valeur est envoyée au port. Les instructions suivantes contrôlent le mode de fonctionnement en Basic :

1. OUT 255,32 - Configure le 80-GRAFIX pour afficher les blocs graphiques TRS-80 normaux. (Bit de contrôle 5). Dans ce mode, le TRS-80 est tel qu'il était avant l'installation du 80-GRAFIX. Il s'agit de l'état normal de mise sous tension de la carte. Ce mode peut être activé et désactivé à tout moment et n'aura aucun effet sur les caractères programmés sur la carte 80-GRAFIX.
2. OUT 255,160 - Règle le 80-GRAFIX en haute résolution et affiche le jeu de caractères programmé sur la carte. (Bits de contrôle 5 et 7). Dans ce mode, les graphiques normaux sont remplacés par des graphiques haute résolution. Les caractères alphanumériques restent inchangés.
3. OUT 255,96 - Règle la carte 80-GRAFIX en mode programmation. (Bits de contrôle 5 et 6). Tous les caractères POKE ou PRINT sur l'écran du TRS-80 sont lus par le 80-GRAFIX et stockés sous forme de séquence binaire dans la RAM de caractères programmables. Tous les caractères dont le bit de poids fort (MSB) est activé sont masqués à l'écran afin d'éviter l'affichage de données parasites pendant la programmation de la carte. Vous devez quitter ce mode dès que vous avez terminé la programmation du 80-GRAFIX.

3.2 PROGRAMMATION DE LA RAM DE CARACTÈRES

La RAM de caractères programmables est organisée en 64 cellules de 72 bits (dans un tableau de 6 bits horizontaux x 12 bits verticaux). Pour programmer le PCR, le 80-GRAFIX est placé en mode PROGRAMMATION (OUT 255,96) et les valeurs appropriées sont affichées à l'écran par blocs de 16 octets chacun, dont les 12 premiers sont lus par le PCR (les 4 derniers octets de chaque bloc sont ignorés). Chacun des blocs de données correspond à un caractère programmé et doit commencer au premier octet de l'écran (15360 décimal, 3C00 hexadécimal), chaque bloc successif commençant à un multiple de 16 à partir de la première position. Ainsi, le premier bloc de données occupera les positions 15360+0 à 15360+15, le bloc suivant commencera à 15360+16 et se terminera à 15360+27, et ainsi de suite pour les 64 caractères.

Le numéro (octet) placé en première position de chaque bloc correspondra au modèle binaire de la ligne supérieure de ce 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 mettre vos craintes de côté, car la valeur de chaque caractère créé dépend des bits activés ou désactivés. En fait, vous pouvez approfondir vos connaissances en logique binaire en utilisant 80-GRAFIX, car voici, mes amis, un appareil qui traduit ces termes abstraits en réalité instantanée. Mais je m'égare. Ce concept est très simple à comprendre. Pour chaque point que vous souhaitez allumer parmi les six positions horizontales, le bit correspondant est activé. Ainsi, si vous souhaitez, par exemple, allumer un point sur deux, le motif binaire ressemblerait à ceci :

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

Pour traduire cela en un nombre courant programmable dans 80-GRAFIX, additionnez simplement les valeurs ACTIVÉES (c'est-à-dire qu'il y a un 1 à cette position). De droite à gauche, la première position (LSB) a la valeur 1, et chaque position successive représente le DOUBLE de la valeur de la précédente. Ainsi, notre octet de 6 bits aux valeurs suivantes :

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

Ainsi, en prenant les valeurs de notre exemple ci-dessus, nous pouvons constater que ces bits sont activés :

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

En additionnant ces valeurs de position, notre nombre décimal est 42. Chacune des 12 lignes (valeurs) peut être facilement déterminée en notant le modèle binaire à représenter et en suivant les étapes ci-dessus.

Voici un exemple de caractère qui ressemblera à une flèche 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 par cette formule sont multipliées par 2 et additionnées à 128 décimales avant d'être affichées sur l'écran de programmation. Ceci est nécessaire pour supprimer l'affichage de « données parasites » lors de la programmation de la carte 80-Grafix. Si vous souhaitez observer les valeurs présentées à 80-Grafix à des fins de débogage, omettez l'ajout de 128 aux valeurs ; les caractères seront alors affichés, car seuls les caractères graphiques sont supprimés à l'écran.

Ce caractère est programmé et affiché par le programme 01, ci-dessous. En faisant varier les coordonnées X et Y, vous pouvez contrôler l'emplacement d'affichage du caractère à l'écran. Ce caractère peut être l'un des 64 caractères programmables en plaçant les 12 octets de données dans l'un des champs de l'écran de programmation.

Vous pouvez visualiser l'intégralité de votre jeu de caractères à tout moment en saisissant les instructions suivantes (en tant que programme ou au 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, définit 80-GRAFIX sur le nœud haute résolution et imprime successivement les 64 caractères graphiques à l'écran, précédés du numéro du caractère.
N'oubliez pas que la valeur ASCII réelle du caractère est 127+C (CHR\$(127+C)).

PROGRAM #1

```

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

```

The following is data for the Character Cell:

```

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 6 program 80-Grafix
55 READ D
60 POKE 15360+(C-1)*16+(K-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 n°1, vous êtes prêt à passer à des choses plus importantes et plus performantes !

Le programme suivant (programme 02), appelé « CREATE », est un programme de base qui vous permet de « dessiner » chaque caractère sur une grille agrandie et de visualiser son aspect en taille réelle. CREATE imprime également les instructions de données nécessaires à la programmation du caractère dans 80-GRAFIX. Une fois satisfait du caractère créé, copiez les instructions de données et intégrez-les dans les futurs programmes qui utiliseront ce jeu de caractères. (« 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

```

```

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 63,63,63,63,63,63,63,63,63,63,63,63,63,0,0,0,0
1340 DATA 63,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
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 +

```

```
      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 guide pour le développement futur de votre programme, vous constaterez que 80-GRAFIX est très simple à manipuler dans votre logiciel. Pour vous entraîner, pourquoi ne pas sortir votre jeu graphique préféré, trouver les valeurs CHR\$ qui dessinent les images et créer votre propre jeu pour les remplacer ? Incluez une sous-routine à la fin du programme qui place vos nouveaux personnages dans 80-GRAFIX et jouez avec VOTRE jeu de personnages !

Vous pouvez obtenir une résolution de 192 x 192 pour votre nouveau jeu de caractères en passant en mode 32 caractères. Comme vous le savez, ce mode double la largeur de chaque caractère et imprime un caractère sur deux à l'écran. L'instruction Basic PRINT CHR\$(23) fait passer le TRS-80 en mode 32 caractères, tandis que PRINT CHR\$(28) ou PRINT CHR\$(28) le ramène en mode 64 caractères. Le mode 32 caractères n'affecte pas la programmation des caractères. Cependant, lorsque vous utilisez POKE pour afficher le jeu de caractères, veillez à n'utiliser POKE que sur les emplacements pairs de l'écran, car les emplacements impairs ne sont pas affichés.

Pour plus d'informations sur la programmation, reportez-vous au manuel d'instructions du TRS-80 NIVEAU II pour l'utilisation des fonctions PRINTS, POKE et CHR\$, ainsi que du Memory Nap, du tableau des fonctions, des codes ASCII et GRAPHICS.

3.3 TECHNIQUES AVEC UN LANGAGE D'ASSEMBLAGE

La programmation de 80-GRAFIX en langage assembleur (machine) est tout aussi simple que les méthodes du langage de base décrites ci-dessus. À la fin de cette section, vous trouverez la liste complète d'une routine de pilote vidéo, dont le code objet est inclus sur la cassette de votre carte 80-GRAFIX.

Tout d'abord, configurez un tampon de données contenant les données de chaque caractère à programmer. Ce tampon est organisé de la même manière que dans un programme de base et comprend jusqu'à 64 champs de 16 octets chacun, dont les 12 premiers s'adressent à la carte 80-GRAFIX. (En fait, pour économiser de l'espace, vous pouvez déplacer des champs de 12 octets chacun sur l'écran de programmation, positionnés de manière à ce que le premier octet de chaque champ commence à une position multiple de 16, ajoutée 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 est définie, il est simple 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 déplace le tampon de 16 octets vers l'écran de programmation et réinitialise la carte pour un affichage haute résolution.

Le caractère programmé a une valeur ASCII de 80H (128 décimal). Une routine pour afficher ce caractère pourrait ressembler à ceci :

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

Cette routine a effacé l'écran et affiché le caractère ASCII numéro 128 (celui que nous avons programmé) dans le coin supérieur gauche de l'écran.

La routine du pilote vidéo qui suit intègre ces techniques pour vous fournir un jeu de caractères « minuscules ». Elle intègre également d'autres fonctionnalités intéressantes, telles que l'installation d'un nouveau pilote vidéo et une fonction d'autoprotection qui le place en haut de votre mémoire et informe le TRS-80 de sa présence, afin qu'il ne soit pas perturbé par d'autres opérations informatiques.

*** NOTE ***

Lorsque vous utilisez le pilote vidéo sur votre ordinateur, il affiche toutes les entrées en majuscules (Avec la touche majuscule) en majuscules, et toutes les entrées en minuscules (Sans la touche majuscule) en minuscules GRAPHIQUES fournies. Vous verrez des minuscules à l'écran lorsque vous utilisez ce pilote, mais n'oubliez pas que les caractères affichés ne sont PAS des minuscules au sens habituel du terme, mais des GRAPHIQUES. Cela signifie que si vous passez en mode graphique normal (OUT 255,32), vous verrez des ORDURES à l'écran. Cela signifie également que les minuscules ne sont pas compatibles avec des programmes tels que « The Electric Pencil », « Script » ou tout autre traitement de texte conçu pour les entrées en minuscules. Cependant, tous les programmes Basic que vous écrivez fonctionneront sur n'importe quelle machine, car les caractères saisis sont inchangés. Si vous exécutez le programme sur une machine non 80-GRAFIX ou sans pilote vidéo, il devrait afficher les caractères normalement, en majuscules.

Pour plus d'informations sur la programmation en langage assembleur, les livres 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 pilote précédent peut être utilisé pour tout jeu de caractères que vous souhaitez saisir. Par exemple, si vous souhaitez programmer en APL, vous pouvez créer le jeu de caractères APL et insérer les instructions de données comme indiqué ci-dessus. Le pilote affichera alors le jeu APL au lieu des minuscules (les entrées décalées restent inchangées, bien entendu).

3.4 PROGRAMMATION DE GRAPHIQUES EN MOUVEMENT

Pour qu'un objet semble se déplacer horizontalement à l'écran, il est nécessaire de programmer jusqu'à 12 caractères différents pour représenter le mouvement. Puisque chaque caractère ne peut se déplacer que de 6 points horizontaux (ou 12 points verticaux), vous devez utiliser six ensembles de deux cellules de caractères pour représenter chaque position du déplacement, le motif de points de chaque cellule étant décalé d'un point vers la droite (ou la gauche). Chaque paire de cellules est affichée au même emplacement à l'écran jusqu'à ce que les six paires soient affichées. Ensuite, la position suivante est utilisée de la même manière, jusqu'à ce que l'objet ait « déplacé » l'ensemble de l'écran. Des affichages très efficaces peuvent être créés de cette manière, même en utilisant Basic.

Le même schéma peut être utilisé pour déplacer des objets dans n'importe quelle direction à l'écran. Pour les objets en mouvement rapide, ou lorsqu'une résolution inférieure est requise, le déplacement d'un seul point n'est pas nécessaire. Vous pouvez déplacer le motif de points de deux positions ou plus dans chaque caractère, en utilisant moins de caractères à chaque déplacement.

Dans les programmes assembleur, le jeu initial de caractères peut être créé, et les segments d'« animation » peuvent être créés par « décalage de bits ». Par exemple, si l'objet à animer nécessite, disons, 12 octets pour être dessiné (comme une fusée), les 12 premiers caractères pourraient être le vaisseau de base. Votre programme assembleur peut alors être amené à lire chacune des 144 instructions de données initiales (12 par caractère) et à décaler arithmétiquement à gauche (SLA) ou à droite (SRA) chacun des octets de données, plaçant la « nouvelle valeur » dans les champs de données suivant le groupe initial. La même procédure peut être suivie pour chaque groupe de caractères jusqu'à ce que le nombre requis soit créé. Il est possible de le faire « à la volée », en générant un nouveau jeu de caractères entre chaque affichage, afin de faire circuler un grand nombre de caractères dans chaque jeu.

3.5 AUTRES IDÉES DE PROGRAMMATION

Outre les méthodes décrites dans la section précédente, le programmeur en langage assembleur trouvera une multitude de possibilités d'adressage avec 80-GRAFIX. Par exemple, grâce aux techniques de décalage de bits mentionnées ci-dessus, il est possible d'utiliser le bit de retenue pour programmer dynamiquement le caractère suivant d'une séquence de figures animées. Un jeu de caractères inversé peut être créé en lisant les données initiales et en complétant (CPL) chaque octet avant de le stocker à sa nouvelle position (ou à sa position précédente). 80-GRAFIX offre la possibilité d'explorer pleinement la logique binaire, car chaque cellule de caractère est une image fidèle des valeurs binaires introduites dans la carte.

Le programmeur BASIC peut employer des techniques similaires pour réorganiser les valeurs des données. Par exemple, supposons que vous ayez configuré un tableau contenant les valeurs binaires de chaque ligne de données, comme dans le programme « CREATE » ci-dessus. En utilisant l'instruction NOT, vous pourriez obtenir le même effet que CPL dans 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, la valeur du bit de chaque cellule est testée pour savoir si elle est activée (1) ou désactivée (0). Si le bit est activé (1), la fonction NOT de la ligne 30 renvoie la valeur -2, et l'ajout de 2 la ramène à zéro. Si le bit est désactivé (0), NOT renvoie la valeur -1, qui, ajoutée à 2, donne un. Ce tableau est traité comme dans « CREATE », et la nouvelle valeur est utilisée pour programmer 80-GRAFIX.

Une routine similaire pour produire un effet de décalage binaire à gauche pourrait prendre la forme suivante :

```
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 supérieure du tableau est abaissée d'une position (ou extraite de la position vers la droite dans chaque « ligne »).

Pour effectuer un décalage vers la droite, cette routine peut 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 une courte routine que vous pouvez exécuter pour illustrer 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 NEXT Y
50 FOR Y=6TO2 STEP-1:   'Shift right
55 V=(X-1)*6+Y
60 A(V)=A(V-1)
65 NEXT Y
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 NEXT Y:PRINT
95 NEXT X

```

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

3.6 QUELQUES RÉFLEXIONS DE FIN

Nous sommes sûrs que vous débordez d'idées et d'applications pour vos nouvelles fonctionnalités 80-GRAFIX, et nous sommes impatients de découvrir vos idées.

Voici quelques idées du concepteur de 80-GRAFIX, Ted Carter :

- Affichage de notes de musique
- Comparaison de synchronisation de circuits intégrés
- Art numérique
- Agencement de bâtiments
- Conception automobile
- Graphiques
- Caractères spéciaux (Algol, APL, grec, etc.)
- Soulignement de mots
- Inversion de caractères vidéo
- En minuscules
- ... et ainsi de suite.

Le reste est à vous !

FIN DE LA DOCUMENTATION DE 80-GRAFIX.

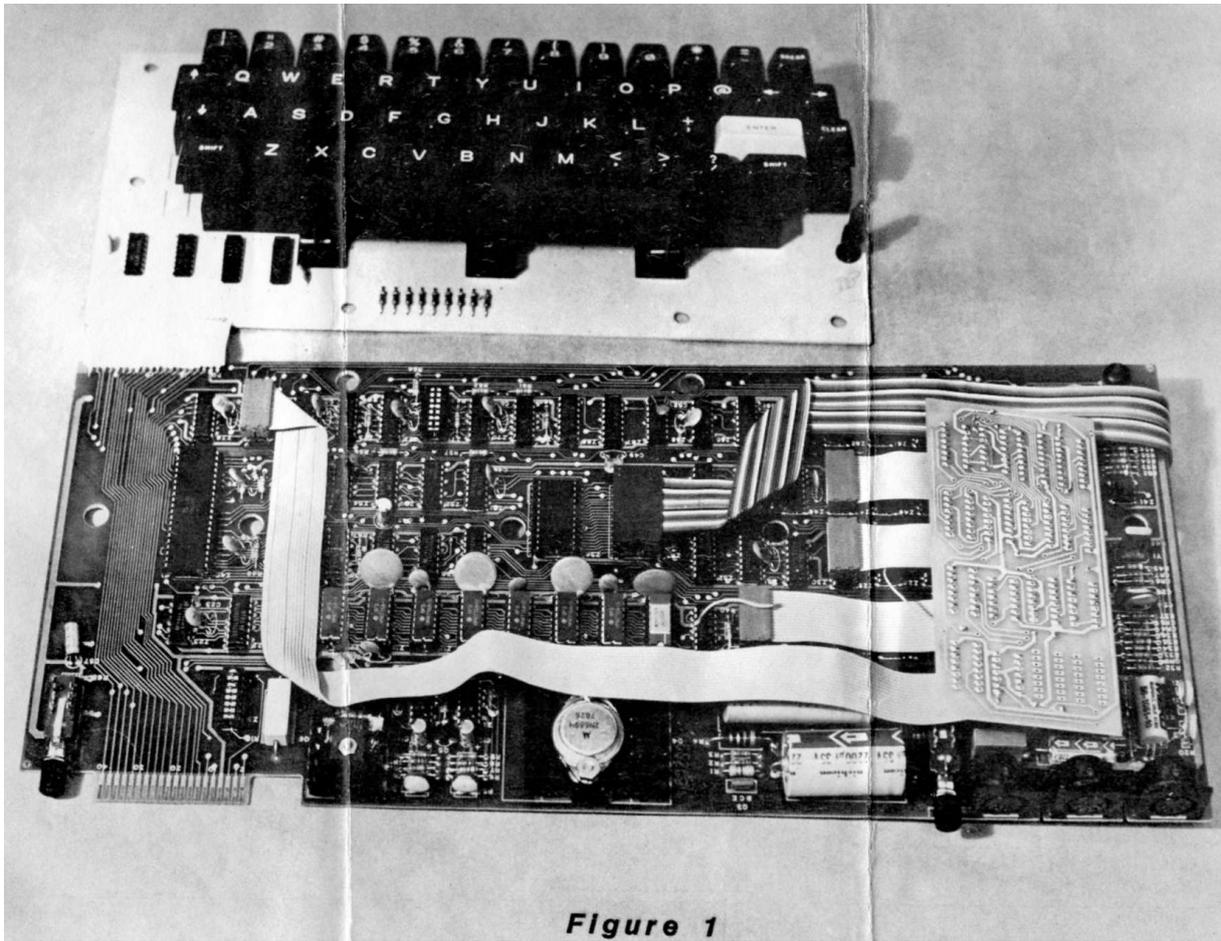


Figure 1

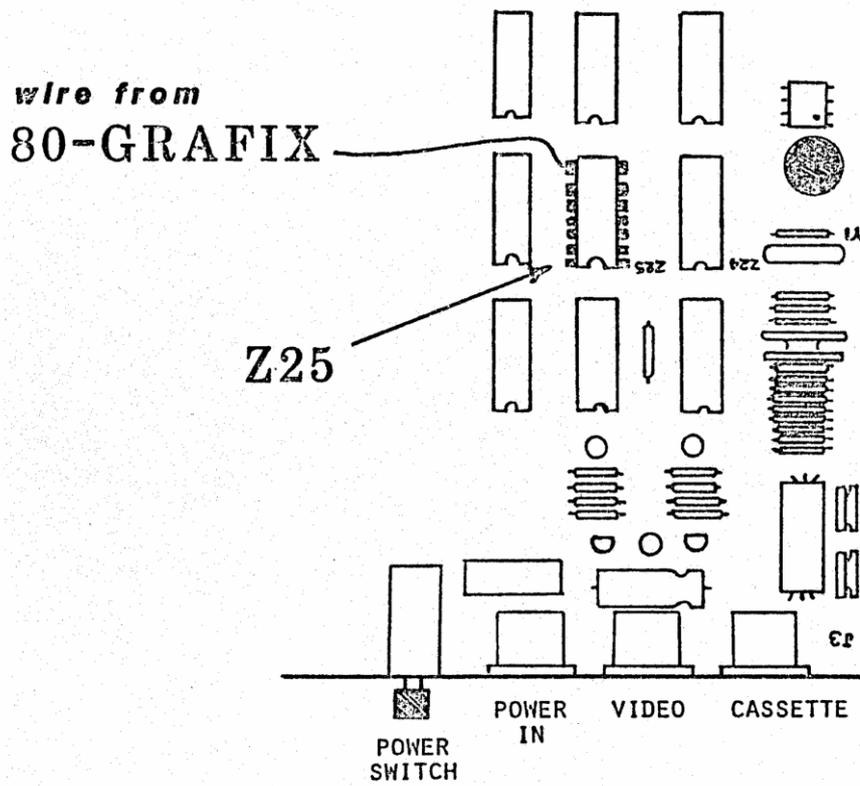
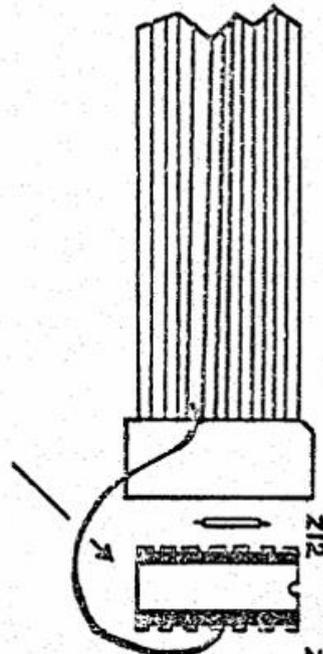
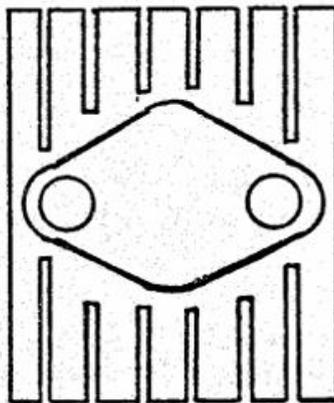
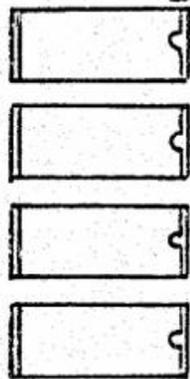


Figure 2

RAM

Z12



ribbon cable from
80-GRAFIX

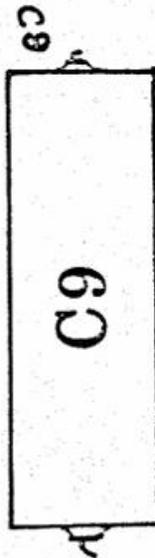


Figure 3