

Ein professionelles System zum Preis eines Home-Computers

Der EC-68 ist ein kleines, aber sehr leistungsfähiges Computersystem, das auch professionellen Ansprüchen gerecht wird. Gemessen an der verfügbaren Anwender-Software und den damit verbundenen unzähligen Einsatzmöglichkeiten bietet dieses System ein enorm günstiges Preis/Leistungsverhältnis. Zwei leistungsfähige Betriebssysteme können auf dem EC-68 eingesetzt werden: Von Beginn an Flex und bei ausreichendem Interesse bald auch das Multitasking-Betriebssystem OS-9 (siehe Kasten auf der folgenden Seite).

Wie bereits der Titel verrät, arbeitet der EC-68 mit dem Betriebssystem Flex. Da Flex vielen Lesern (noch) unbekannt sein dürfte, werden wir im Software-Teil dieses und der folgenden Hefte auf dieses interessante System eingehen: In einer Artikelserie wird Flex und die mit Flex arbeitende Anwender-Software ausführlich vorgestellt. Schon jetzt sei verraten, daß das Software-Angebot recht reichhaltig ist. Viele bekannte Programme werden auch für Flex angeboten. Das Betriebssystem OS-9 muß noch an den EC-68 angepaßt werden, dies ist jedoch ein reines Software-Problem und hat mit der hier beschriebenen Hardware nichts zu tun. Ein größeres Problem bereiten uns der Preis und die Lizenzbedingungen. Darauf gehen wir in einem Kasten auf der folgenden Seite genauer ein.

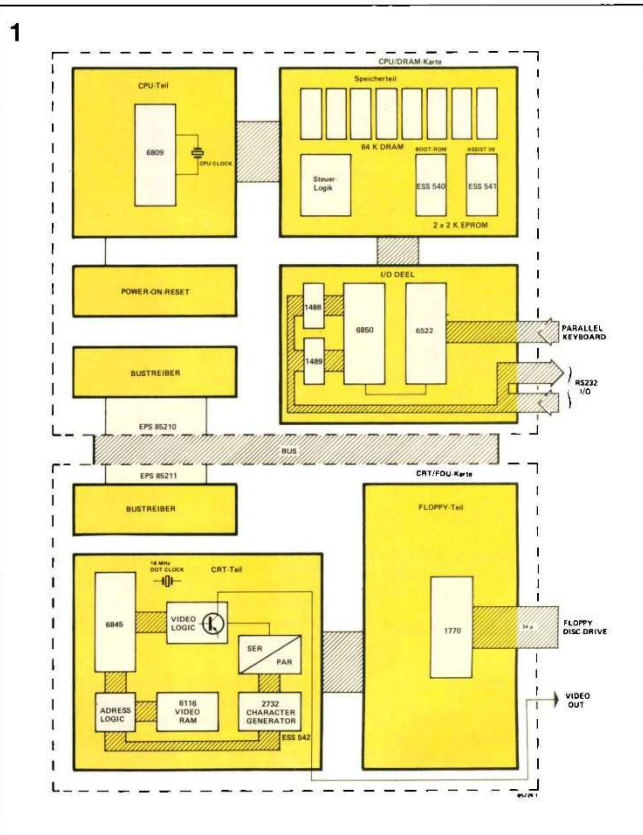
Systemhardware

Zur Hardware jedes Home- und Personal-Computers gehören bestimmte Funktionsgruppen, ohne die das System einfach unvollständig wäre. Beim Flex-Computer EC-68 ist das nicht anders: Er besteht aus dem CPU-Teil, dem Arbeitsspeicher, einem Video-Interface sowie einem Floppy-Disk-Interface. Eine einfache Rechnung läßt vermuten, daß die System-Hardware wahrscheinlich auf vier Euro-Karten verteilt ist. Die Rechnung ist falsch! Alle genannten Funktionen konnten auf nur zwei Euro-Karten untergebracht werden. Die erste Karte beherbergt den CPU- und Speicher-Teil, die zweite das Bildschirm- und das Floppy-Interface.

Bild 1 zeigt das Blockscha der Systemhardware; gleichzeitig gibt dieses Bild bereits einen Überblick über die verwendeten Chips.

EC-68 — der Elektor-Flex-Computer

Entwurf: Guido de Cuyper
Text: Paul Panter



Natürlich besteht ein Computer nicht nur aus zwei Europakarten mit der Elektronik. Beim EC-68 werden alle peripheren Komponenten über Industrie-Standard-schnittstellen angeschlossen. Man benötigt:

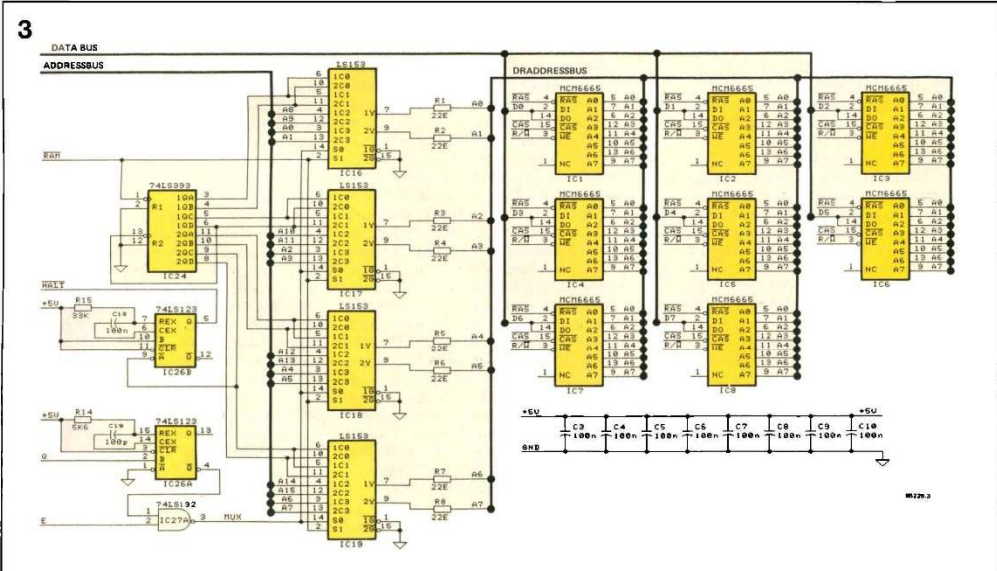
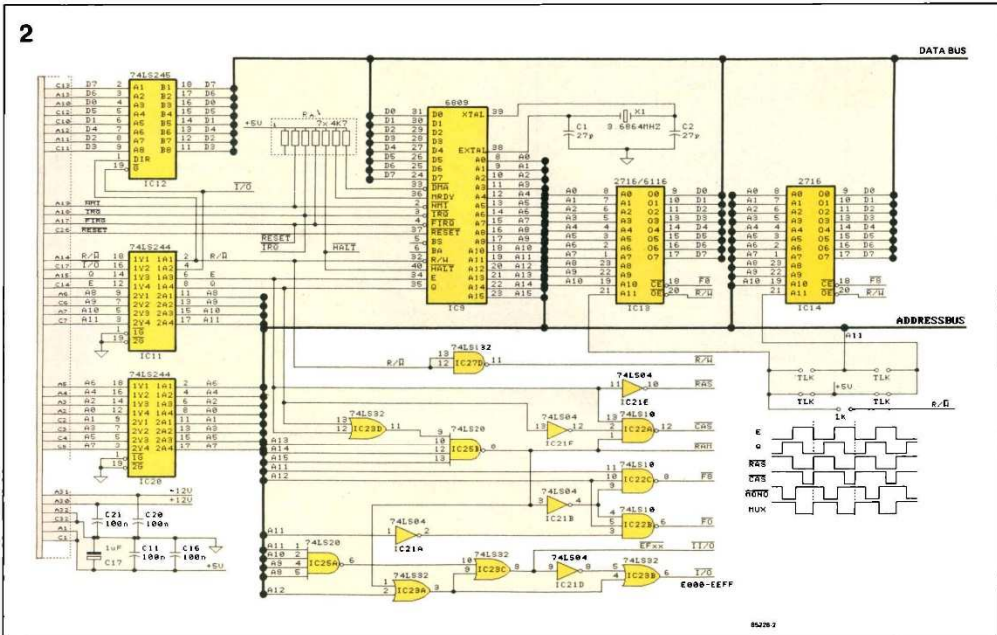
- ◆ Eine Parallel-Tastatur (ASCII).
- ◆ Einen Monitor mit 75-Ohm-Eingang, minimale Bandbreite 16 MHz.
- ◆ Minimal ein Floppy-Laufwerk 40 oder 80 Track mit "Shugart-Bus". Wichtige Einschränkung: Es muß ein doppelseitiges Laufwerk sein, da die Systemdiskette doppelseitig beschrieben ist.
- ◆ Ein Netzteil 5 V / 3 A, +12 V / 2 A (für Floppy-Laufwerke), +12 V / 250 mA und -12 V / 250 mA (für die RS-232-Schnittstelle). Ein starkes Netzteil für spätere Erweiterungen kann natürlich nicht schaden.

Bild 1. Dieses Blockschaubild gibt eine Übersicht über die Funktionsblöcke des EC-68 und ihre Verteilung auf den beiden Euro-Karten.

Bild 2...4. Drei computergezeichnete Schaltbilder sind nötig, um die Schaltung der CPU/DRAM-Platine wiederzugeben. Man kann sich leicht denken, daß es auf der Platine etwas enger zugeht als auf normalen EPS-Platinen.

- ◆ Eine Busplatine, zum Beispiel EPS 83102 (OmniBus), mit Konnektoren.

Insgesamt kann man sagen: Da es sich um einen Eurokarten-Computer handelt, gelten für das gesamte "Drumherum" die



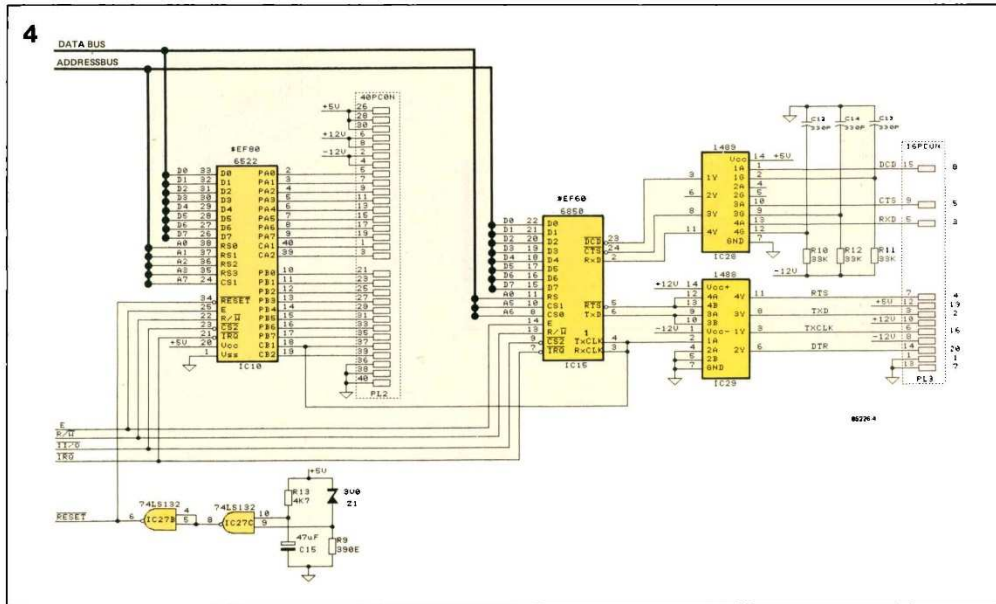
gleichen Hinweise, wie wir sie für den EC-65 im Artikel "Das Drumherum" im ersten Computer-Sonderheft gegeben haben. Das gilt natürlich auch für das dort beschriebene Gehäuse, in das wir auch unseren EC-68 wieder eingebaut haben. Einzige Ausnahme und bereits erwähnt: Die Laufwerke müssen doppelseitig sein.

CPU-Teil
Die Bilder 2 bis 4 zeigen die Schaltung

der CPU/DRAM Karte. Bild 2 zeigt den CPU-Teil.

Der Prozessor gilt mit Recht als das Herz jedes Mikrocomputer-Systems. Beim Flex-Computer EC-68 schlägt als Herz ein 6809, denn Flex wurde für den 6800 und den 6809 geschrieben; die Version für den 6809 erhielt dabei den Namen "Flex-9". Die Firmware (Assist-09 und Bootrom) ist in zwei EPROMs des Typs 2716

(IC13 und IC14 in Bild 2) resident gespeichert. Als System-Frequenz wurde 4 MHz gewählt, genauer gesagt: 3.6864 MHz, denn von dieser Frequenz lassen sich bequem die Baudraten für den ACIA ableiten (siehe unten). Um mit der doppelten Frequenz von 8 MHz arbeiten zu können, müssen nicht nur der Quarz X1 mit C1 und C2 angepaßt werden, sondern es werden außerdem die B-Typen der 68XX-Reihe sowie schnellere dynamische



RAMs benötigt. Ferner muß auch noch das Timing der RAMs geändert werden. Das alles ist zwar durchaus machbar, wir raten jedoch davon ab, wenn Meßgeräte und Erfahrungen zum Anpassen des Timing fehlen.

Wie schon beschrieben, ist die komplette Hardware auf zwei Euro-Karten untergebracht. Da das Platinenlayout busorientiert gestaltet wurde, stehen späteren Hardware-Erweiterungen keine Hindernisse im Weg. Der Bus (Adreß-, Daten und Kontrollbus) wird von den Treibern IC11, IC12 und IC20 gepuffert, sie bilden die "Nahtstelle" zwischen den beiden Karten. Einer Erklärung bedürfen in Bild 2 noch die Drahtbrücken und die Gatterschaltungen. Mit den Brücken kann man die Schaltung von IC13 an das CMOS-RAM 6116 anpassen, wenn die Karte einem anderen als dem hier beabsichtigten Zweck dienen soll (rein hardwaremäßig betrachtet kann man mit den beiden Platinen auch ein Terminal aufbauen). Beim Elektor-Flex-Computer ist IC13 stets ein EPROM vom Typ 2716, in Bild 2 sind die Brücken dementsprechend gezeichnet. Über den Inhalt der beiden EPROMs (IC13 und IC14) folgt Näheres bei der Beschreibung der Firmware. Bleiben noch die Gatter im unterem Mittelteil von Bild 2: Sie erzeugen verschiedene Chip-Select-Signale sowie die RAS- und CAS-Signale für die dynamischen RAMs.

Arbeitsspeicher und I/O

Im Elektor-Flex-Computer werden 8 dynamische RAMs des Typs 4164 als Arbeitsspeicher eingesetzt, Bild 3 zeigt die Schaltung. Der eigentliche Speicher besteht aus IC1...IC8 (Kapazität insgesamt 64 KByte), während die übrigen ICs Steuerungsaufgaben haben. Ein Timing-Diagramm der Steuersignale ist rechts unten in Bild 2 zu finden.

Daneben hat auf der ersten Platine noch die Hardware ihren Platz, die die Verbindungen mit der Außenwelt herstellt. Aus Bild 4 geht hervor, daß es sich dabei um zwei I/O-Chips (IC10 und IC15) und zwei ICs zur Pegelanpassung (IC28 und IC29) handelt. Die Hardware ist so ausgelegt, daß Daten sowohl parallel als auch seriell gesendet und empfangen werden können. Den parallelen Datenaustausch, z.B. mit dem Keyboard und einem Drucker mit Centronics-Anschluß, übernimmt der VIA 6522 (IC10). Der ACIA 6850 bildet zusammen mit der Pegelanpassung (IC28/IC29) eine serielle RS-232-Schnittstelle. Die Kommandos und anderen Eingaben des Benutzers nimmt der Computer über ein Keyboard mit parallelem Ausgang entgegen. Die dafür vorgesehene Schnittstelle ist einer der beiden Ports des 6522 (Port A von IC10 in Bild 4), während man den anderen als Centronics-Ausgang verwenden kann. Darauf werden wir in ec 4 genau eingehen. Die serielle RS-232-Schnittstelle kann beliebig verwendet werden, zum Beispiel für die Datenübertragung mit einem Modem. Auf der Systemdiskette ist auch ein Modemprogramm enthalten (MODEM.CMD).

Betrachtet man die Takteingänge des 6850 (Pin 3 und Pin 4 von IC15), so erhält man die Erklärung für den "merkwürdigen" Systemtakt-Quarz. Der VIA-Port CBI (Pin 18 von IC10) wird von der Firmware als Takteiler programmiert, der den Systemtakt auf dem ACIA-Takt herunterteilt. Dazu muß der Systemtakt natürlich ein ganzzahliges Vielfaches der gängigen ACIA-Frequenzen sein. Da der 6809 die Quarzfrequenz intern durch 4 teilt und der Systemtakt möglichst dicht bei 4 MHz liegen sollte, ergab sich aus diesen beiden Bedingungen die Quarzfrequenz von 3.6864 MHz.

Taktgenerator und -teiler für den ACIA konnten durch diese genial-einfache Idee eingespart werden. Und das spart nicht nur Geld. Durch diese Maßnahme und 5 Drahtbrücken, die auf der Platinenrückseite "von Hand" zu legen sind (wir kommen gleich darauf zu sprechen), ist es erst möglich geworden, soviel Hardware auf einer Eurokarte unterzubringen.

Bild 5 zeigt die Belegung des 40-poligen Steckers mit den parallelen Ports für Keyboard und Drucker. Bild 6 enthält die entsprechende Belegung des 16-poligen Steckers auf der Platine für die serielle Schnittstelle, die Bildunterschrift enthält die Belegung der 25-poligen D-Sub-Buchse der Standard-RS-232-Schnittstelle. Die Beschaltung der Bus-Konnektoren, die die Verbindungen zwischen den beiden Karten und eventuellen späteren Erweiterungen herstellen, geht aus Bild 7 hervor. Schließlich ist noch der Power-On-Reset mit den Gattern IC27B und IC27C (Bild 4) zu erwähnen, der den Computer beim Einschalten der Netzspannung startet und einen Taster für den System-Reset überflüssig macht.

Der Bau der CPU-Karte ist nicht schwierig, sofern man bei der Bestückung sorgfältig arbeitet (sprich: lötet). Es wird angesichts der recht hohen Packungsdichte auf den beiden Platinen niemand verwundern, daß das Layout kein typisches Elektor-Layout sein konnte: Zum Beispiel für die großen 8-eckigen Lötungen bei den ICs war hier einfach kein Platz. Bei der Montage der Bauteile (Bild 8 zeigt den Bestückungsplan) dürfen die fünf Leitungen nicht vergessen werden. Es sind stets die Punkte miteinander zu verbinden, die durch den gleichen Buchstaben gekennzeichnet sind: A mit A, B mit B usw. Bei IC13 lötet man die Verbindungen

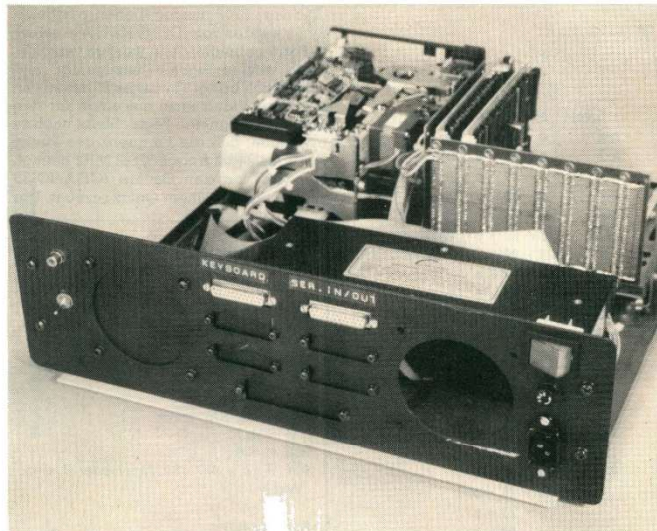
dungsleitungen direkt an die Anschlüsse des EPROM-Sockels. Die zweite Reihe Anschlußpunkte befindet sich zwischen IC 20 und IC 21. Hier muß man scharf aufpassen: Hält man die Platine so, daß man auf die Bestückungsseite sieht und der Buskonnektor nach links zeigt, sind die unteren fünf Lötungen die Anschlußpunkte A... E. Der Lötunkt direkt darüber ist lediglich eine Durchkontaktierung und darf nicht benutzt werden! Es empfiehlt sich, die ICs wie in der Stückliste angegeben in IC-Fassungen zu stecken. Die EPROMs müssen natürlich programmiert sein, bevor sie auf die Karte kommen (siehe "Firmware" und "Bezugsquellen").

Am besten legt man die Verbindungsleitungen auf der Lötseite (Rückseite) der Platine. Dieser kleine Kompromiß war leider nötig, um alles auf einer Eurokarte unterzubringen.

Video- und Floppy-Interface.

Der Schaltungsaufwand, der notwendig ist, um Zeichen auf einem Bildschirm sichtbar zu machen, beschränkt sich leider nicht allein auf den CRTC (Cathode Ray Controller). Ein Blick auf die Bilder 9 und 10 macht deutlich, daß außer dem CRTC (hier der bekannte 6845) noch erheblich mehr vorhanden ist.

Das Schirmbild setzt sich zusammen aus 24 Reihen mit jeweils 80 Zeichen. Insgesamt sind dies 24 * 80 = 1920 Zeichen, so daß für den Bildwiederholungspeicher ein



Hardware

2-K-RAM benötigt wird. Dafür bietet sich das CMOS-RAM-6116 an, in der Schaltung (Bild 9) ist es mit IC 20 bezeichnet. In den Speicher gelangen die Zeichen in Form von ASCII-Codes über IC1 und IC10, die Adressierung der Speicherplätze erfolgt

über IC17, IC18 und IC19. Mit dem Ablegen der ASCII-Zeichencodes im Bildwiederholungspeicher ist es noch nicht getan, sondern die Zeichen müssen auch noch in für das Auge lesbare Punktmuster (Charakter) umgesetzt werden. Diese Aufgabe hat der Zeichengenerator IC21 (ein 2732), wobei IC12 (74LS377) als Zwischenpuffer dient. Der ASCII-Code des darzustellenden Zeichens wird quasi als Adresse der Speicherplätze im 2732 benutzt, unter der das Punktmuster dieses Zeichens abgelegt ist.

5	CA-1	o - 1	2 - o	- 12 volt	Aus
	CA-2	o - 3	4 - o	+ 12 volt	
	PA-0	o - 5	6 - o	+ 12 volt	
	PA-1	o - 7	8 - o		
	PA-2	o - 9	10 - o		
	PA-3	o - 11	12 - o		
	PA-4	o - 13	14 - o		
	PA-5	o - 15	16 - o		
	PA-6	o - 17	18 - o		
	PA-7	o - 19	20 - o		
	FB-0	o - 21	22 - o		
	PB-1	o - 23	24 - o		
	PB-2	o - 25	26 - o	+ 5 volt	
	PB-3	o - 27	28 - o	+ 5 volt	
	PB-4	o - 29	30 - o	+ 5 volt	
	PB-5	o - 31	32 - o		
	PB-6	o - 33	34 - o		
	PB-7	o - 35	36 - o		
	CB-1	o - 37	38 - o	Gnd	
	CB-2	o - 39	40 - o	Gnd	

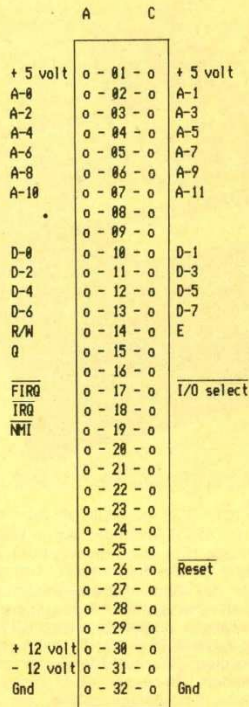
Bild 5. Die Parallelschnittstelle. Der parallele Port PA ist der Tastatureingang. An PA-0 liegt Bit 0, an PA-1 liegt Bit 1 usw. Das Strobe-Signal (Daten werden mit der Vorderflanke der Strobe-Impulse übernommen) wird an CA-1 gelegt. Über CB-1 erhält der ACIA (6850) sein Taktsignal, diese Leitung ist deshalb mit Pin 3 und 4 des 6850 verbunden. Port PB kann bei Bedarf als Parallelausgang (Centronics-Schnittstelle) für einen Drucker benutzt werden.

Bild 6. Die serielle Schnittstelle. Das Bild zeigt die Pinbelegung von Konnektor 2 auf der CPU-Karte. Eine normgerechte RS-232-C-Schnittstelle entsteht, wenn die Signale zu einer (im Gehäuse eingebauten) 25-poligen Sub-D-Buchse geführt werden. Die Signale sind dort an folgende Pins zu legen:

6	GND	o - 1	2 - o	
	TxD	o - 3	4 - o	
	RxD	o - 5	6 - o	TxCk
	RTS	o - 7	8 - o	- 12 volt
	CTS	o - 9	10 - o	+ 12 volt
		o - 11	12 - o	+ 5 volt
	GND	o - 13	14 - o	DTR
	DCD	o - 15	16 - o	

- 01 = Gnd (ground)
- 02 = TxD (transmit data)
- 03 = RxD (receive data)
- 04 = RTS (ready to send)
- 05 = CTS (clear to send)
- 06 = frei
- 07 = Gnd (ground)
- 08 = DCD (data carrier detect)
- 09 = frei
- bis
- 15 = frei
- 16 = TxCk (transmit clock)
- 17 = frei
- 18 = frei
- 19 = +5 V
- 20 = DTR (data terminal ready)
- 21 = frei
- bis
- 25 = frei

7



Die vom 2732 ausgegebenen parallelen Bytes werden von IC22 (74LS165) in serielle Form gebracht, dann werden verschiedene weitere Signale hinzugefügt, und schließlich bringt Transistor T1 das so entstandene Videosignal auf einen für den Monitor geeigneten Pegel. Bleibt noch zu erwähnen, daß die Frequenz des Punkttahtsignals (dot frequency) 16 MHz beträgt, sie wird von den Gattern IC13A/IC13B zusammen mit einem Quarz erzeugt. Der

CRTC IC2 (der 6845) hat bei der Generierung des Videosignals die Funktion der Leitzentrale: Er koordiniert den zeitlichen Ablauf des gesamten Prozesses und erzeugt außerdem die Synchronisationssignale (horizontal und vertikal) für den Monitor. Seinerseits wird der CRTC von der CPU (6809) gesteuert, denn er ist kein "intelligenter" Videokontroller und kann ohne CPU nicht arbeiten. Die CPU adressiert ihn wie jeden anderen peripheren

**EC68: Stückliste
CPU/DRAM-Karte**

Widerstände (alle 1/8 W):

- R1...R8 = 22 Ω
- R9 = 390 Ω
- R10...R12, R15 = 33k
- R13 = 4k7
- R14 = 5k6
- Ra1 = 7 x 4k7 (SIL-Array oder 7 einz. Wid.)

Kondensatoren:

- C1, C2 = 27 p ker.
- C3...C11, C16, C18, C20, C21 = 100 n ker.
- C12...C14 = 330 p ker.
- C15 = 47 µ / 16 V Tantal
- C17 = 1 µ / 16 V Tantal
- C19 = 100 p ker.

Halbleiter:

- Z1 = 3V0 / 400 mW Z-Diode
- IC1...IC8 = 4164 (200 ns f. Grundversion)
- IC9 = 6809
- IC10 = 6522
- IC11, IC20 = 74LS244

- IC12 = 74LS245
- IC13 = 2716 (ESS 540)
- IC14 = 2716 (ESS 541)
- IC15 = 6850
- IC16...IC19 = 74LS153
- IC21 = 74LS04
- IC22 = 74LS10
- IC23 = 74LS32
- IC24 = 74LS393
- IC25 = 74LS20
- IC26 = 74LS123
- IC27 = 74LS132
- IC28 = 1489 (75189)
- IC29 = 1488 (75188)

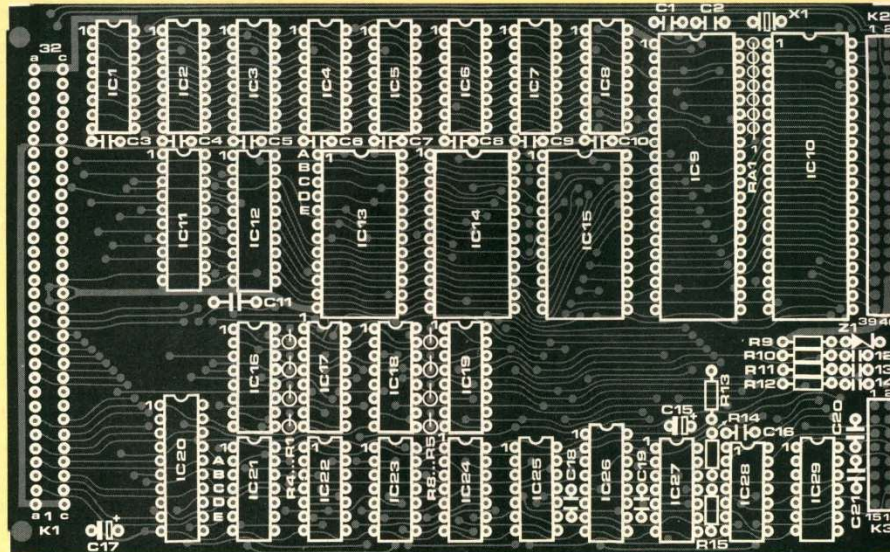
außerdem:

- X1 = Quarz 3,6864 MHz (kleine Bauform)
- K1 = Busstecker nach DIN 41612, a und c best.
- K2 = Doppelp. Pfostenfeldstiftleiste, 40-pol., mit passendem Stecker für Flachbandkabel
- K3 = wie K2, aber 16-pol.

Hinweis: K2, K3 z.B. Fabrikat Molex, Lieferant Rätev

- K2: 5342-40-GS1 & 5320-40-BGS1
- K3: 5342-16-GS1 & 5320-16-BGS1
- Platine 85210

8



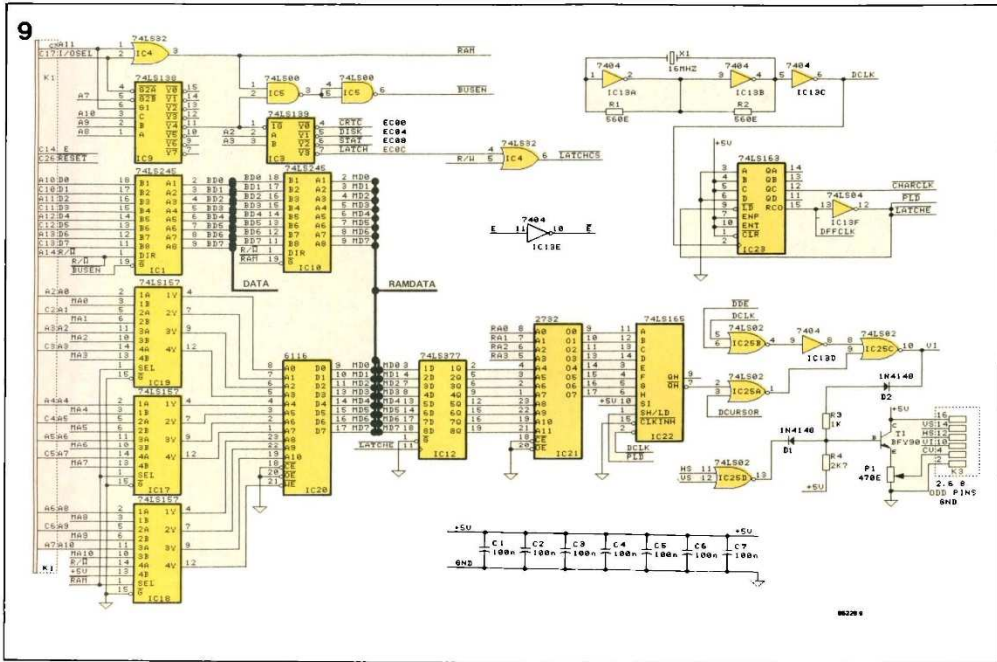


Bild 7. Belegung der Busstecker beim Elektor-Flex-Computer. Es ist noch so manche Busleitung frei für spätere Erweiterungen. Wichtiger Hinweis: die Busbelegung des EC-68 entspricht nicht dem vielen Lesern bekannten Elektor-Bus!

Bild 8. Der Bestückungsplan der CPU/DRAM-Karte bestätigt die bei Bild 2...4 getroffene Aussage: Insgesamt 29 ICs sind auf dieser Karte untergebracht, darunter 5 "Hinkelsteine".

Bild 9 und 10. Ebenfalls computergezeichnet wird in diesen beiden Schaltbildern der "Inhalt" der FDC/CRT-Platine wiedergegeben. Sie ist ein kleines bisschen weniger dicht bestückt als die CPU/DRAM-Platine: Der Floppy-Controller ist hoch integriert und erfordert nur eine geringfügige externe Beschaltung.

Tabelle 2. Diese Stückliste enthält alle für den Aufbau der CPU/DRAM-Karte benötigten Bauteile.

Chip (VIA und ACIA); hier läuft die Adressierung über IC9 und IC3, links oben in Bild 9.

Anders als beim Video-Teil ist für die Steuerung eines oder mehrerer Disketten-Laufwerke außer dem Floppy-Disk-Controller (FDC, IC17 in Bild 10) nur noch wenig zusätzlicher Aufwand nötig. Fast alle Funktionen sind auf dem hochintegrierten (und deshalb nicht ganz billigen) FDC-Chip vereint, einem 1770 von Western Digital. Trimpotentiometer, die man noch in älteren FDC-Schaltungen findet, fehlen hier völlig, denn auch der Datenseparator ist auf dem Chip untergebracht.

Auf Einzelheiten des FDC-Innenlebens soll hier nicht eingegangen werden, es sei lediglich festgestellt, das IC17 Daten in seriellen Format auf die Diskette schreibt und von ihr liest. IC8, IC9 und IC16 puffern die Steuersignale und verhindern gleichzeitig Beschädigungen des 1770, falls versehentlich der Stecker des Laufwerks falsch angeschlossen wird oder die Ausgänge aus anderem Grund kurzgeschlossen werden. Die Gefahr besteht deshalb, weil jede zweite Ader des Verbindungskabel an Masse liegt; eine gegenseitige Beeinflussung der Signale über die Kabelkapazitäten wird dadurch weitgehend verhindert. Aus dem gleichen Grund sollte auch das Kabel möglichst kurz sein.

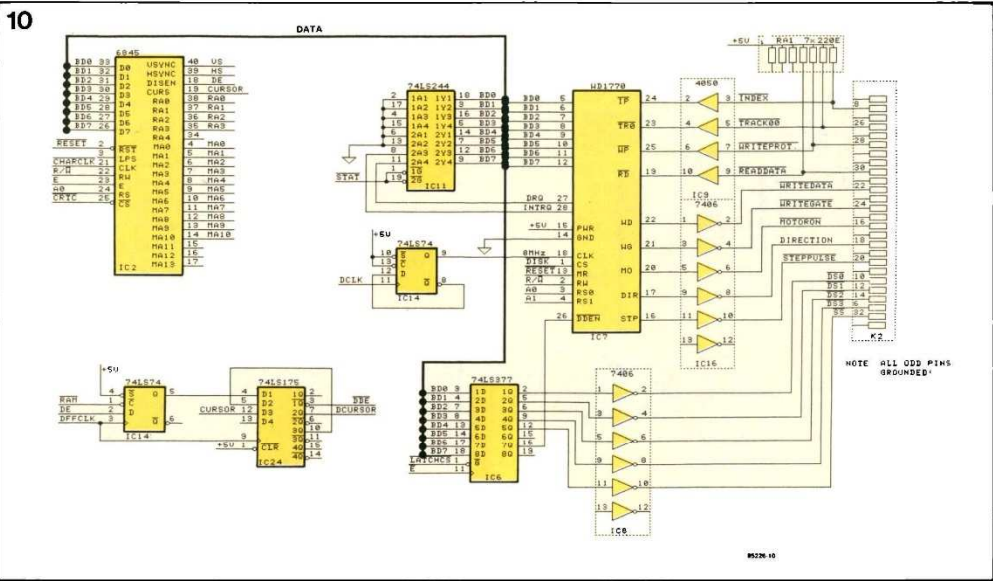
Wenn alle Adern des Bandkabels korrekt im Steckverbinder eingeklemmt wurden (liegt Ader 1 tatsächlich an Pin 1?), kann allerdings kaum noch etwas schiefgehen. Die in der Stückliste angegebenen Verbindungen lassen sich nämlich nicht seitenverkehrt zusammenstecken. Welche Signale auf den einzelnen Konnektor-Pins liegen, geht aus Bild 11 hervor. Außerdem läßt sich dort den Signalbezeichnungen entnehmen, das der 1770 maximal vier Laufwerke steuern kann: die Signale DS1...DS3 sind die dafür notwendigen Drive-Select-Signale. In Bild 12 sind schließlich die Signale angegeben, die auf dem Video-Konnektor liegen. Von hier führt man ein Koaxkabel zum Monitor.

Nach der Montage der Bauelemente auf der zweiten Karte (Bestückungsplan in Bild 13, Bauteilliste siehe Tabelle 2) ist der Aufbau der Hardware abgeschlossen.

Firmware

Ohne Software ist ein Computersystem nicht einmal eine halbe Sache, sondern es ist quasi tot. Beim Flex-Computer EC-68 lassen sich zwei Software-Kategorien unterscheiden: Zum einen die Routinen, die in den EPROMs auf der Platine gespeichert sind, zum anderen die auf Disketten abgelegten Programme. Kennzeichnend für das Betriebssystem Flex ist die Eigenschaft, daß fast die gesamte Software auf Disketten steht. Anders als bei Home-Computern üblich wurde die Zahl der EPROMs auf das unumgängliche Mindestmaß beschränkt. Der Vorteil ist eine hohe FLEXibilität, denn die Disketten-Software läßt sich leicht ändern und anpassen. Ferner wird von den EPROMs nur ein kleiner Bereich des verfügbaren Adressbereiches der CPU belegt. Auf der anderen Seite müssen diese Vorteile durch eine etwas niedrigere Arbeitsgeschwindigkeit erkauft werden. Die Zugriffszeit auf die Daten liegt bei Disketten nun einmal deutlich niedriger als bei EPROMs.

An dieser Stelle soll nur der Inhalt der drei EPROMs betrachtet werden, die auf der Flex-Diskette stehende Software ist erst später Gegenstand unserer Betrachtungen. Auf der CPU-Platine befinden sich zwei EPROMs, und auf der CRT/FDC-Platine ist ein weiteres EPROM untergebracht. Das zuletzt genannte EPROM enthält kein Programm, sondern den Zeichensatz (Character-Generator) des Video-Interfaces. Für den Fall, daß hier Änderungen erwünscht sind, steht auf der Flex-Diskette ein Text-File mit dem Namen CGTXT. Diese File gibt Auskunft über den Inhalt des Zeichengenera-



tors, es kann mit dem Kommando LIST CGTXT auf den Bildschirm geholt werden. Übrig bleiben die beiden EPROMs auf der CPU-Platine, in denen die Programme "Assist-09" und "Bootrom" stecken.

Erstes Sprungziel des Prozessors nach dem Power-On-Reset ist Assist-09, ein Monitorprogramm, das ursprünglich von Motorola stammt. Inzwischen ist Assist-09 jedoch frei für die Benutzung durch jedermann. Der Entwickler des EC-68 nahm dieses Monitorprogramm jedoch nur als Grundlage; das in IC14 gespeicherte Programm ist nicht das Original, sondern eine an den Flex-Computer EC-68 angepasste Version! Die Startadresse wird vom Reset-Vektor bestimmt, der im Prozessor-Chip festgelegt ("festverdrahtet") ist, er hat die Adresse \$FFFE und führt den Prozessor zur Adresse \$F800.

Nach dem Start erscheint auf dem Bildschirm der Text "ASSIST 09" gefolgt vom Monitor-Prompt ">".

Assist-09 wartet nun auf die Eingabe eines Kommandos. Durch Tippen des Buchstabens "F" (für "FLEX") und eines Carriage Return wird Assist-09 veranlaßt, mit Unterstützung von "Bootrom" das Betriebsprogramm von der Diskette in den Arbeitsspeicher zu laden. Ferner gehören zu Assist-09 verschiedene weitere Kommandos, mit denen sich unter anderem die Inhalte der diversen Register sichtbar machen und modifizieren lassen. So setzt zum Beispiel ein "R" die Prozessor-Register-Inhalte auf den Bildschirm. Gibt man den Buchstaben "D" gefolgt von zwei Hexadressen ein, kann man den Inhalt des betreffenden Speicherplatzes betrachten und verändern. Die Kommandos, die sämtlich auf Maschinsprache-Ebene liegen, kön-

11

Gnd	0 - 1	2 - 0	
Gnd	0 - 3	4 - 0	
Gnd	0 - 5	6 - 0	DS3
Gnd	0 - 7	8 - 0	Index
Gnd	0 - 9	10 - 0	DS0
Gnd	0 - 11	12 - 0	DS1
Gnd	0 - 13	14 - 0	DS2
Gnd	0 - 15	16 - 0	Motor On
Gnd	0 - 17	18 - 0	Direction
Gnd	0 - 19	20 - 0	Steuimpulse
Gnd	0 - 21	22 - 0	Write Data
Gnd	0 - 23	24 - 0	Write Gate
Gnd	0 - 25	26 - 0	Track 00
Gnd	0 - 27	28 - 0	Write Protect
Gnd	0 - 29	30 - 0	Read Data
Gnd	0 - 31	32 - 0	SS
Gnd	0 - 33	34 - 0	

nen auch von den Flex-Dienstprogrammen ("Utilities") aufgerufen werden, die auf der Diskette stehen.

Unter dem Namen "Bootrom" sind die für die Kommunikation notwendigen maschinenspezifischen (hardwareabhängigen) Routinen im zweiten EPROM zusammengefaßt. Es handelt sich dabei hauptsächlich um die Video-, Floppy- und I/O-Treiber, in denen maschinenspezifische Adresse vorkommen. Die wichtigsten Adressen des Flex-Systems EC-68 lassen sich bereits den Schaltbilder entnehmen. So werden zum Beispiel die vom Keyboard kommenden Daten über den PLA-Port mit der Adresse \$EF80 eingelesen, während die serielle Schnittstelle (der ACIA) die Adresse \$EF60 und die Floppy-Schnittstelle (der FDC) die Adres-

12

Gnd	0 - 1	2 - 0	Gnd
Gnd	0 - 3	4 - 0	CV
Gnd	0 - 5	6 - 0	Gnd
Gnd	0 - 7	8 - 0	Gnd
Gnd	0 - 9	10 - 0	VI
Gnd	0 - 11	12 - 0	HS
Gnd	0 - 13	14 - 0	VS
Gnd	0 - 15	16 - 0	+ 5 volt Aus

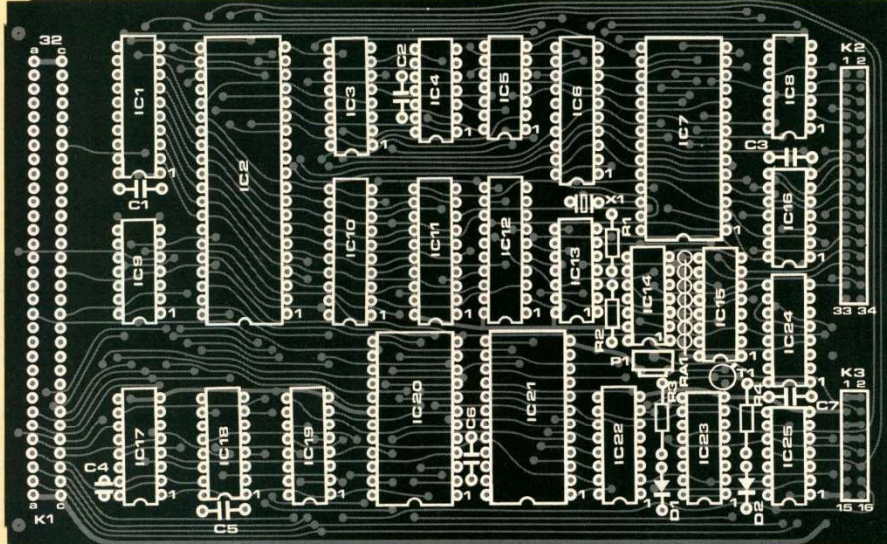
Bild 11. Die Floppy-Schnittstelle entspricht dem verbreiteten Shugart-Standard. Dadurch ist es sehr einfach, gängige Floppy-Laufwerke mit dem EC-68 zu verbinden. Wichtiger Hinweis: Es muß sich um doppel-seitige Laufwerke handeln (40 oder 80 Tracks).

Bild 12. Der Konnektor für den Monitor. An sich benötigt man nur das Composite-Video-Signal und eine Masseleitung, diese beiden führt man über ein Koax-Kabel (75 Ohm) oder über ein nicht zu langes Stück abgeschirmtes flexibles Kupferkabel an den Monitoreingang. Auf dem Konnektor kann man aber auch das invertierte Video-Signal und die beiden Synchronsignale abgreifen. Letztere sind zu Meßzwecken auf den Konnektor geführt.

Bild 13. Der Bestückungsplan der DISK/CRT-Platine. Mit "nur" 25 ICs ist sie "etwas lockerer" bestückt als die CPU/DRAM-Karte.

Tabelle 1. Aus dieser Tabelle gehen die wichtigsten Systemadressen und die Speicheraufteilung des EC-68 hervor.

Tabelle 3. Die Stückliste der FDC/CRT-Karte.



Stückliste CRT/FDC-Karte

Widerstände (alle 1/8 W):

- R1, R2 = 560 Ω
- R3 = 1k
- R4 = 2k7
- P1 = 470 (500) Ω
- Ra1 = 7 x 220 Ω (SIL-Array oder 7 einz. Wid.)

Kondensatoren:

- C1...C7 = 100 n (ker.)

Halbleiter:

- D1, D2 = 1N4148
- T1 = BFY90
- IC1, IC10 = 74LS245

- IC2 = 6845
- IC3 = 74LS139
- IC4 = 74LS32
- IC5 = 74LS00
- IC6, IC12 = 74LS377
- IC7 = 1770 (Western Digital, Lieferant z.B. Ratev)
- IC8, IC16 = 74LS06
- IC9 = 74LS138
- IC11 = 74LS244
- IC13 = 74LS04
- IC14 = 74LS74
- IC15 = 4050
- IC17...IC19 = 74LS157
- IC20 = 6116
- IC21 = 2732 (ESS 542)
- IC22 = 74LS165
- IC23 = 74LS163

- IC24 = 74LS175
- IC25 = 74LS02

außerdem:

- X1 = Quarz 16 MHz (kleine Bauform)
- K1 = Busstecker nach DIN 41612, a & c best.
- K2 = Doppelp. Pfostenfeldstiftleiste, 34-pol., mit passendem Stecker für Flachbandkabel
- K3 = wie K2, 16-pol.
- Hinweis: K2, K3 z.B. Fabrikat Molex, Lieferant Ratev:
- K2: 5342-34-GS1 & 5320-34-BGS1
- K3: 5342-16-GS1 & 5320-16-BGS1
- Floppyseitiges Gegenstück f. K2: 8173-34a
- Platine EPS 85211

Tabelle 1.

:F000...FFFF:Assist-09 (EPROM);
:F000...F7FF:Bootrom (EPROM);
:E800...EFFF:I/O;
:ED00...EFFF:User-I/O;
:EF00...EFFF:ACIA, VIA;
:EF80:VIA (parallele I/O);
:EF60:ACIA (serielle I/O);
:EC00...ECFF:DISK, VIDEO;
:EC07:Floppy Disc Controller (Datenregister);

:EC06:Floppy Disc Controller (Sektorregister);
:EC05:Floppy Disc Controller (Trackregister);
:EC04:Floppy Disc Controller (Status- und Kommandoregister);
:EC00:CRTC;
:E800...EBFF:User-I/O;
:E000...E7FF:Videomemory;
:C000...DFFF:FLEX;
:DE00...DFFF:Disk Drivers;
:D400...DDFF:FMS (File Management System);

:C000...D3FF:DOS;
:C980...CBFF:System Files Area;
:C840...C97F:System FCB (File Control Block);
:C700...C83F:Scheduler & Printer Spooler (nicht implementiert);
:C100...C6FF:Utility Command Area;
:C080...C0FF:Input Buffer;
C000...C07F:Stack Area (SP wird auf C07F initialisiert);
:C000:Flex-Betriebssystem-Ladeadresse;
:0000...BFFF:User-RAM;

se \$EC0x haben.

Die verschiedene Treiber sind ausführlich durch eine Reihe von Textfiles auf der Diskette dokumentiert, so daß sich eine nähere Darstellung an dieser Stelle

erübrigt. Um die beim Systemstart ablaufenden Vorgänge auf "Bitebene" nachzuvollziehen (falls man daran Interesse hat), kann man diese Files auf den Bildschirm holen oder ausdrucken lassen.

Eine globale Übersicht über die Adressen-Organisation des EC-68, beginnend mit der höchsten Adresse, gibt Tabelle 1.

Bezugsquellen

Nachfolgend nennen wir — ohne Anspruch auf Vollständigkeit — einige Bezugsquellen. Die genannten Firmen informieren Sie nach Möglichkeit auch über Flex-Anwenderclubs in der Nähe ihres Wohnorts.

♦ Die Platinen und EPROMs können vom Elektor-Platinen-Service (EPS) beziehungsweise Elektor-Software-Service (ESS) oder vom Fachhandel bezogen werden.

CPU/DRAM-Karte: EPS-Nr. 85210
CRT/FDC-Karte: EPS-Nr. 85211

- ESS 540: 1 x 2716 EPROM, Programmname: Bootrom

- ESS 541: 1 x 2716 EPROM, Programmname: Assist-09 (nicht gleich mit dem Programm gleichen Namens von Motorola aus dem "Motorola-Datenbuch 6809"!)

- ESS 542: 1 x 2732 EPROM: Zeichengenerator EC-68

♦ Die Boot-Diskette mit dem Betriebssystem Flex und verschiedenen Informations-Files ist erhältlich bei:

EDV-Systeme & Peripherie
Dipl.-Ing. Werner Schorstein
Pankratiusstr. 11
6100 Darmstadt
Tel. (06151) 74350

♦ Die Firma Zacher Computer GmbH bietet für das Betriebssystem Flex eine Reihe von Software-Paketen an. Außerdem kann man dort das "68 Journal" beziehen. Die Anschrift lautet:

Zacher Computer GmbH
Im Schwarzenstein 34
5521 Irrel
Tel. (06525) 299

♦ Der Floppy-Disc-Controller WDI770 (Hersteller: Western Digital) und die Molex-Konnektoren werden vertrieben von:

RATEV
Gothaerstr. 15
4030 Ratingen
Tel. (02102) 42051/52

Weitere Angebote und Informationen finden Sie im Anzeigenteil dieses Heftes.

OS-9

Die Lizenzrechte an dem Betriebssystem OS-9 besitzt die US-Firma Microware. Diese Firma versucht sich gegen Raubkopierer zu schützen, indem sie ihren Lizenznehmern (Distributoren) relativ strenge Regeln auferlegt. OS-9 darf nur maschinengebunden verkauft werden. Dies bedeutet für den EC-68: Der Elektor-Verlag als Platinenhersteller muß von Microware eine Unterlizenz kaufen und darf dann die Systemdisketten für den EC-68 weitergeben, die mit einem von Microware gelieferten Label versehen und mit mehreren Original-Handbüchern von Microware zusammen ausgeliefert werden.

Diese Lizenz kostet etliche zehntausend Mark, zu denen noch die Personalkosten für die Diskettenherstellung und den Vertrieb hinzukämen, und solche Kleinigkeiten wie die Disketten selber, Versandkosten und ähnliches. Nach unserer Grobkalkulation würde OS-9 auf diesem Wege den einzelnen Leser zwischen 700,- und 900,- DM kosten, wobei die Anzahl der (vor)bestellten Systeme den Preis erheblich beeinflussen kann. Das ist immerhin noch mehrere hundert Mark weniger, als OS-9 für den 68000 kostet (dort liegen die Preise zwischen 1200,- und 1400,- DM, je nach Rechner).

Der Elektor-Verlag hat sich daher zu folgendem Vorgehen entschlossen:

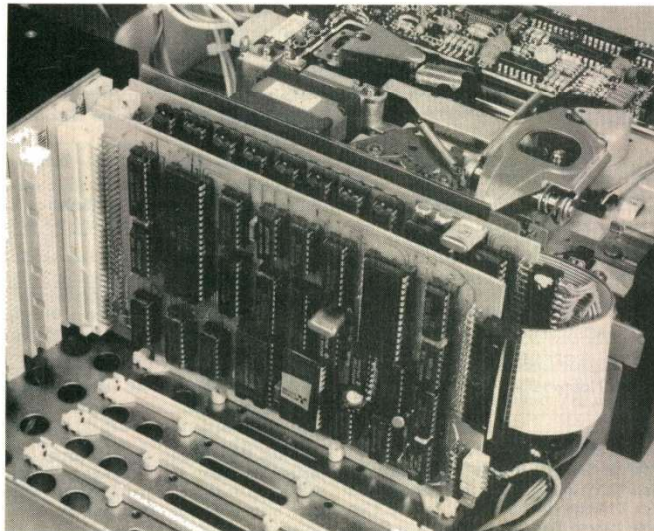
Jeder ausgelieferten Platine (auch den über den Fachhandel ausgelieferten) wird eine Kaufbescheinigung beigelegt, die mit dem Verlagsstempel

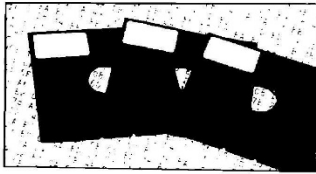
versehen ist. Diese kann jeder Leser an den Verlag einschicken, der Interesse an OS-9 hat (Kopien sind ungültig, nur die einzeln abgestempelten Originale gelten). Auf diesem Blatt muß deutlich (Maschine oder Blockschrift) Name und Anschrift des Käufers eingetragen werden. Außerdem kann man auf diesem Blatt OS-9 fest vorbestellen oder sein noch unverbindliches Kaufinteresse bekunden.

Wenn eine ausreichende Anzahl fester Vorbestellungen und Kaufinteresse-Bekundungen vorliegen, wird der Elektor-Verlag eine Lizenz vom deutschen Microware-Generalvertreter erwerben und OS-9 an den EC-68 anpassen.

Außerdem werden wir in Kürze ein Informationsblatt zu OS-9 und der unter OS-9 laufenden Software erstellen und Interessenten gegen einen fankierten Rückumschlag zuschicken, es kann unter dem Stichwort "OS-9 Info" beim Verlag angefordert werden. Wer die Platinen jetzt noch nicht kaufen will, kann auf einem Coupon, der dem Info-Blatt beigelegt ist, OS-9 fest vorbestellen oder sein Kaufinteresse bekunden (die Platinen müssen dann allerdings später vor der Auslieferung der Systemdiskette gekauft werden).

Wir schätzen, daß auf diesem Wege bei ausreichendem Interesse OS-9 bis Mitte 1986 auf dem EC-68 laufen kann (Hinweis dazu: Die Anpassung ist nach Auskunft des deutschen Microware-Generalvertreters für ihn sehr einfach und kann auch in kurzer Zeit vorgenommen werden.)





In diesem Heft stellen wir einen 6809-Computer auf zwei Euro-Karten vor: Den EC-68. Dazu gehört natürlich auch Software. Wir haben das in den USA verbreitete und bekannte Betriebssystem Flex-9 ausgewählt. Zumindest für den Anfang, später soll noch das Echtzeit-Betriebssystem OS-9 hinzukommen. Das Software-Angebot für Flex-9 ist sehr groß. Es kann jedoch im Einzelfall vorkommen, daß man direkt in den USA bestellen muß, wenn man etwas ganz spezielles haben will. Im Normalfall ist das nicht nötig: Das allermeiste kann man direkt von deutschen Firmen beziehen.

Das Betriebssystem Flex ist ein Disk-Operating-System für CPUs der 68xx-Serie (zum Beispiel 6800, 6802, 6809). Dieses Betriebssystem ist in Deutschland weitgehend unbekannt, zumindest was den Bereich der Hobby-Computeristen angeht. Im kommerziellen Bereich hat es durchaus auch in Deutschland eine Rolle gespielt, wird jetzt allerdings zunehmend von dem leistungsfähigeren Nachfolge-Betriebssystem UNI-FLEX abgelöst. Den Preis für UNI-FLEX wird allerdings kaum ein Hobbyist bezahlen können, und auch der Etat von kleinen Firmen dürfte da häufig überfordert sein. Dafür ist UNI-FLEX nach Aussagen von zwei deutschen Vertriebsfirmen, die wir besucht haben, noch leistungsfähiger als UNIX - und natürlich ein Multi-User-System. Doch zurück zu Flex: Flex-9 (für den 6809) dürfte das leistungsfähigste 6809-Betriebssystem sein, das überhaupt erhältlich ist, und ganz sicher ist es CP/M in vielen Punkten überlegen. Wobei es nur eine Einschränkung an dieser Aussage gibt: Für Echtzeitanwendungen wird man OS-9 vorziehen.

Wer sich intensiv mit dem Betriebssystem Flex befassen will, dem sei die Anwen-derzeitschrift "68 Micro Journal" empfohlen (eine deutsche Bezugsquelle nennen wir zusammen mit anderen Bezugsquellen am Artikelende der Hardwarebe-
schreibung weiter vorne im Heft). Leser, denen "Computerchinesisch auf Grund-lage der englischen Sprache" nichts Fremdes ist, finden hier eine Fülle von Softwareinformationen und Hardware-tips. Und wer sich nicht scheut, auch einmal eine Diskette in den USA zu bestellen, wird auch ausgefallene Programme für seinen 6809-Flex-Computer bestellen können. Die am Ende der Hardware-Beschreibung genannten Bezugsquellen, von denen eine auch eine an unseren EC-68 angepaßte Flex-Systemdiskette

Das Betriebssystem FLEX

Paul Panter

```
1
Der vom Benutzer eingegebene Text ist unterstrichen.
+++ LIST STARTUP.TXT
0.VERIFY IS ON
0.HECHO 0C
0.HECHO 1B 2E 33
0.HECHO 0D 0A
0.DATE
0.HECHO 0D 0A
0.ECHO Good day FLEX software at your command
0.ECHO ### E L E K T O R .6809 Flex computer ###
0.ECHO My system information is:
0.ECHO [1] Memory capacity is 64 KBytes
0.ECHO [2] Microprocessor is 6809
0.ECHO [3] Disc format according to FD 1771
0.ECHO [4] If drive is 40 tracks and we have double side
0.ECHO      and double density with 18 sectors per track.
0.ECHO      Then we have a disc capacity of 1404 sectors
0.ECHO      of 256 bytes each. Thus we have 376 KBytes of
0.ECHO      disc storage.
0.HECHO 0D 0A
0.HECHO 0D 0A
0.ASN W=0 S=0
0.TTYSET WD=80 DP=25 PS=Y
0.TTYSET TB=5C EJ=00 ES=20 BE=08
0.LIST 0.START1.TXT
+++
```

anbietet, werden aber sicher für die meis-ten Nachbauer ein zumindest in der Anfangsphase ausreichendes Softwarean-gebot sicherstellen. Wir wollen hier gleich die Anmerkung einflechten: Hat man einmal eine lauffähige Systemdisket-te, kann man alle andere Flex-Software ohne weitere Probleme verwenden.

Neben der US-Firma Technical Systems Consultants (TSC), die die Rechte an dem von uns verwendeten Betriebssystem besitzt, liefern auch verschiedene andere Softwarehäuser Anwenderprogramme für dieses Betriebssystem. Die ganze Fülle der Flex-Software werden wir sicher nicht

Bild 1. So kann beispielsweise der Inhalt einer STARTUP.TXT-File aussehen.

in ec vorstellen können, wir müßten sonst ein reines Flex-Heft machen. Aber für die folgenden ec-Ausgaben sind bereits Arti-kel zum Teil fertig, zum Teil in Vorberei-tung, in denen Flex-Software besprochen wird. Im Anschluß an diesen Artikel fin-den Sie eine Beschreibung wichtiger Flex-Utilities. Ein Artikel über die Flex-Diagnostics befand sich auf der Reser-ve-liste für dieses Heft, wird aber aus Platz-gründen in ec 4 erscheinen.

SYSTEMSOFTWARE

Alle zum Systemstart notwendigen Files gehören zum Software-Paket auf der Systemdiskette. An dieser Stelle sollen zunächst nur die wichtigsten besprochen werden.

Wir beginnen mit Flex, auf der Diskette "FLEX.SYS" genannt. Um dem Anwender die Arbeit zu erleichtern, ist mit dem eigentlichem Betriebssystem (DOS) eine Systemfile verbunden, die diverse Fehlermeldungen ausgibt (Filename: ERRORS.SYS).

Nach Einlesen des DOS sucht Flex eine Kommandofile (Kennzeichen: .CMD), die EXEC (von "execute") heißt: EXECCMD. FLEX führt nämlich die Kommandozeile aus, die in einer Textfile mit Namen STARTUPTXT steht. In den meisten Fällen ist dies: EXEC @STARTTXT. Eine File "STARTTXT" muß deshalb ebenfalls auf der Diskette vorhanden sein. Diese Textfile wird vom Anwender selbst mit Hilfe

eines Textverarbeiters geschrieben. Sie enthält verschiedene Kommandozeilen, die sequentiell ausgeführt werden. EXEC @STARTUPTXT bedeutet folglich: Führe nacheinander die Kommandos aus, die in Textfile STARTUPTXT zusammengefaßt sind. Bild 1 zeigt ein typisches Beispiel für den Inhalt der STARTUP-File, während Bild 2 das Flußdiagramm für den Ablauf der Prozedur wiedergibt. Die beiden Bilder machen deutlich, welche Funktion STARTUPTXT innerhalb des Gesamtgeschehens hat. Aus Bild 1 geht ferner hervor, daß für jedes Kommando eine neue Zeile benötigt wird. Die einzelnen Kommandozeilen haben folgende Bedeutung: Am Anfang von STARTUPTXT steht VERIFY ON. Die Kommandofile VERIFY.CMD überprüft das Beschreiben der Diskette, indem sie die zuvor dorthin geschriebene File noch einmal liest und mit dem Original vergleicht. Diese sicherlich sehr nützliche Kontrolle kostet allerdings Zeit; ein Grund, weshalb die Zeile in der STARTUP-File von Anwenderprogrammen häufig fehlt.

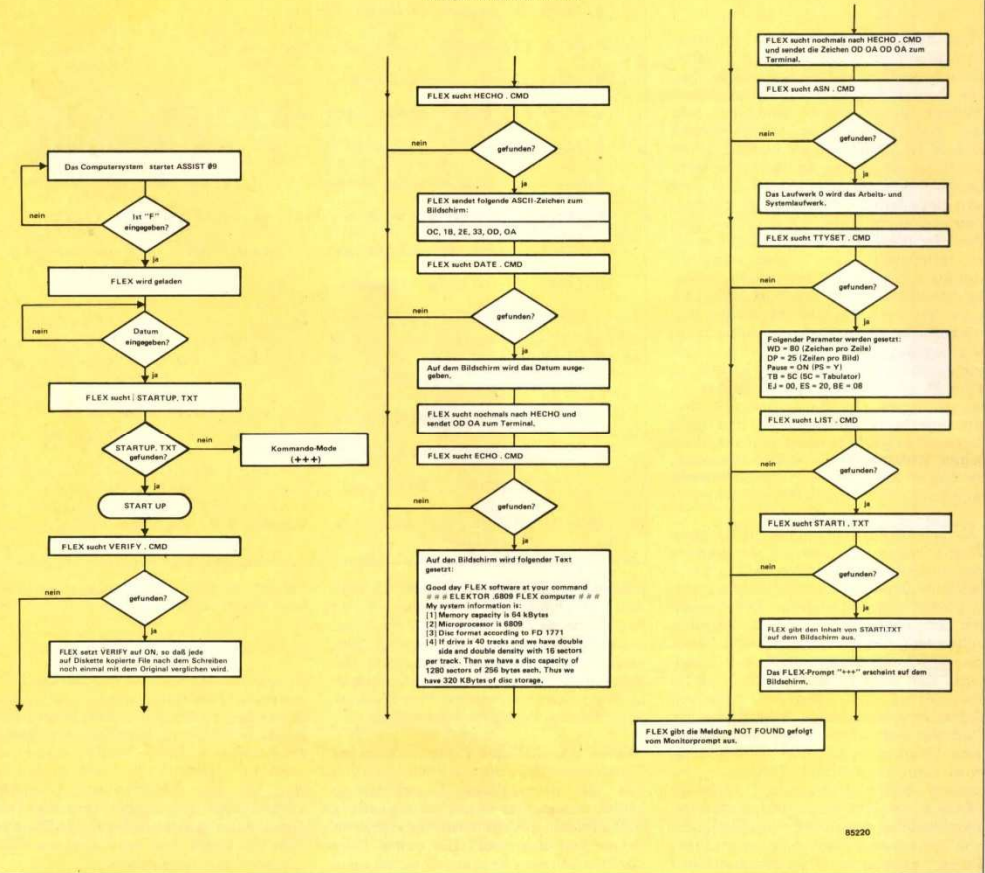
HECHOCMD ist eine File, die hexadezimale Zeichenketten zum Terminal sendet. So bedeutet beispielsweise HECHO 0C: "Sende das ASCII-Zeichen 0C zum Terminal". Hierdurch wird der Bildschirm gelöscht (0C = Clear Screen). Um eine neue Zeile zu beginnen, muß das Kommando folglich HECHO 0D 0A lauten (0D = Carriage Return, 0A = Line Feed).

DATE.CMD gibt das beim Systemstart eingegebene Datum auf dem Bildschirm aus. Mit dem gleichen Kommando läßt sich auch das zuvor eingegebene Datum ändern. Dieses Kommando ist jedoch stark von der Hardware des Computers abhängig. In vielen Fällen muß deshalb die Datei DATE.TXT modifiziert und neu assembliert werden.

ECHOCMD ist mit HECHOCMD eng verwandt: ECHO setzt alphanumerische Zeichen auf den Bildschirm, so daß hiermit unter anderem Systeminformationen und -meldungen ausgegeben werden können. Die Anzahl der ECHO-Zeilen ist

2

Flußdiagramm für den Kaltstart von FLEX



```

3
+++ newdisk
ARE YOU SHURE? Y
SCRATCH DISK IN DRIVE 0? Y
80 TRACKS (N=40 TRACKS)? Y
DOUBLE SIDED DISK? Y
DOUBLE DENSITY DISK? Y
16 SECTORS (N=16 SECTORS)? Y
VOLUME NAME? ELEKTOR
VOLUME NUMBER? 1

FORMATTING COMPLETE
TOTAL SECTORS = 2844
+++

```

beliebig, sie müssen jedoch stets mit **ECHO** beginnen.

ASNCMD (von "assign") bestimmt, welches Diskettenlaufwerk als Arbeitsspeicher (W) dient, und welches das Systemlaufwerk ist. Ist nur ein Laufwerk vorhanden, so muß das Kommando **ASNCMD W=0 S=0** lauten, in diesem Fall ist das Laufwerk 0 sowohl das Arbeits- als auch das Systemlaufwerk.

TTYSETCMD hat eine wichtige Funktion, denn mit diesem Kommando werden verschiedene Terminal-Parameter gesetzt. Dazu gehören unter anderem: Die Anzahl der Zeilen pro Bildschirmseite (DP=25; 25 Zeilen/Bild) und die Zuordnung eines ASCII-Zeichens zur TAB-Funktion (hier: 5C).

LISTCMD beschließt das Beispiel in Bild 1. Es gibt die angegebene File auf dem Bildschirm aus: **LIST START1.TXT** setzt den Inhalt des Textfile auf den Schirm, das den Namen **START1.TXT** trägt. Da man von der **START1.TXT** File beispielsweise zu einem Textverarbeiter springen kann (dazu genügt die Zeile **SCR DEMOTXT**), läßt sich **START1.TXT** für Hinweise oder Erläuterungen verwenden. Diese können sich auf den Programmstart mit **FLEX** beziehen und gegebenenfalls die wichtigsten Kommandos beschreiben. Auf einer anderen Diskette, die zum Entwickeln und Testen von **BASIC**-Programmen dient, kann die File **START2.TXT** dem Benutzer erklären, daß nach dem Laden von z.B. **Extended Basic (XBASIC)** sofort die Eingabe von **BASIC**-Programmzeilen möglich ist.

Selbstverständlich müssen die in **START1.TXT** genannten Kommandofiles auf der Diskette vorhanden sein; anderenfalls reagiert **FLEX** mit "NOT FOUND".

Nicht weniger wichtig als **FLEX.SYS** ist die File **LINKCMD**, die für die Verbindung ("linken") von **FLEX** mit dem Umlader (bootstrap loader) notwendig ist, die Syntax lautet hierzu: **LINK FLEX <CR>**.

```

4
+++ DIR
DIRECTORY OF DRIVE NUMBER 0
DISK: ELEKTOR #1   CREATED: 28-MAY-85

FILE#  NAME  TYPE  BEGIN  END  SIZE  DATE  PRT
-----
1  FLEX  .SYS  01-01  01-17  23  28-MAY-85
2  ERRORS .SYS  01-18  01-18  4  28-MAY-85
3  EXEC  .CMD  01-1C  01-1C  1  28-MAY-85
4  STARTUP .TXT  01-1D  01-1D  1  28-MAY-85
5  VERIFY .CMD  01-1E  01-1E  1  28-MAY-85
6  ECHO  .CMD  01-1F  01-1F  1  28-MAY-85
7  HECHO .CMD  01-20  01-20  1  28-MAY-85
8  DATE  .CMD  01-21  01-22  2  28-MAY-85
9  ASN  .CMD  01-23  01-23  1  28-MAY-85
10 TTYSET .CMD  01-24  02-01  2  28-MAY-85
11 LIST  .CMD  02-02  02-04  3  28-MAY-85
12 LINK  .CMD  02-05  02-05  1  28-MAY-85
13 P     .CMD  02-06  02-06  1  28-MAY-85
14 PRINT .SYS  02-07  02-07  1  28-MAY-85
15 CAT  .CMD  02-08  02-0E  7  28-MAY-85
16 DIR  .CMD  02-0F  02-13  5  28-MAY-85
17 APPEND .CMD  02-14  02-16  3  28-MAY-85
18 BUILD .CMD  02-17  02-17  1  28-MAY-85
19 COPY  .CMD  02-18  02-1C  5  28-MAY-85
20 DELETE .CMD  02-1D  02-1E  2  28-MAY-85
21 NEWDISC .CMD  02-1F  03-01  7  28-MAY-85
22 O     .CMD  03-02  03-03  2  28-MAY-85
23 RENAME .CMD  03-04  03-04  1  28-MAY-85

FILES=23, SECTORS=76, LARGEST=23, FREE=1328
+++

```

Zur Abrundung des Überblicks über das Betriebssystem hier noch eine kurze Betrachtung der Files, denen man beim Umgang mit **FLEX** am häufigsten begegnet: **PCMD** und **PRINTSYS** werden verwendet, wenn Informationen nicht auf dem Bildschirm, sondern über den seriellen Drucker ausgegeben werden sollen. Anstelle von **LIST Filename.Filetype** steht in diesem Fall einfach **P LIST Filename.Filetype**. Es ist auch möglich, einen Drucker mit Centronics-Interface an den **EC-68** anzuschließen. Wie, darauf werden wir in **ec 4** eingehen.

CATCMD und **DIRCMD** setzen den Katalog bzw. das Inhaltsverzeichnis einer Diskette auf den Bildschirm. Ebenso gibt **P**

Bild 2. Dieses Flußdiagramm zeigt den Ablauf einer Startup-Prozedur. Ist **Flex** geladen, dann sucht es nach der Datei **STARTUP.TXT**. In dieser Datei stehen alle Kommandos, die der Computer benötigt, um sich für den Anwender zu konfigurieren. Die Kommandos werden vom Computer so gelesen, als kämen sie von der Tastatur.

Bild 3. Sequenz für das Formatieren einer **80-Spur-ds-dd**-Diskette. Vor Beginn der Prozedur muß die zu formatierende Diskette in Laufwerk 0 eingelegt sein. Ein bereits formatiertes Exemplar wird erneut formatiert; dabei gehen alle früheren Informationen verloren!

Bild 4. Ein typisches Beispiel für das Inhaltsverzeichnis (Directory) einer Diskette, wie es durch das **DIR**-Kommando aufgerufen wird.

DIR 1 das Inhaltsverzeichnis der in Laufwerk 1 befindlichen Diskette auf dem Drucker aus. DIR liefert stets umfangreichere Information als CAT.

APPENDCMD verbindet beliebige Files miteinander. Beispiel: APPEND,Filename1,Filetyp,Filename2,Filetyp... usw. ... neuer Filename,neuer Filetyp.

BUILDCMD ist ein ziemlich einfaches Programm, das zum Erstellen von Texten dient (Syntax: BUILD,Filename). Den Filetype braucht man hier nicht anzugeben, denn zum Filenamen wird automatisch txt hinzugefügt. BUILD wird hauptsächlich dazu verwendet, die File STARTUP.TXT (siehe oben) zu erstellen.

COPYCMD dupliziert Files von einer Diskette auf eine andere.

DELETECMD schafft Platz auf der Diskette, indem es Files löscht. Vor der Ausführung stellt DELETE stets die Frage "ARE YOU SURE?". Die Eingabe des Buchstaben Y (für "YES") löst das Löschen der Spur- und Sektor-Informationen im Inhaltsverzeichnis (oder Katalog) aus. Die eigentlichen Informationen bleiben dabei erhalten; es werden lediglich die betreffenden Sektoren zum Überschreiben freigegeben.

NEWDISCCMD ist in der Regel das allererste, praktisch erprobte Kommando, denn mit ihm werden neue Disketten for-

matiert. Die File stellt eine Reihe von Fragen an den Benutzer, die sich auf die Anzahl der Spuren, einfache oder doppelte Schreibdichte und auf den Diskettennamen und die Diskettennummer beziehen. Nach dem Formatieren überprüft NEWDISC alle Sektoren auf einwandfreie Beschaffenheit und schlägt Alarm mit der Meldung "BAD SECTORS", falls irgendetwas nicht in Ordnung ist. Ein Beispiel für eine NEWDISC-Sequenz ist in Bild 3 zu finden.

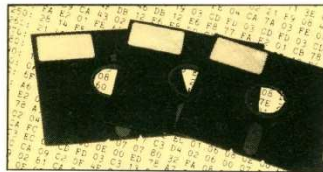
OCMD schreibt die aus einer anderen Kommandozeile resultierenden Informationen (z.B. ein Inhaltsverzeichnis) nicht auf den Bildschirm, sondern auf eine Diskette. Das Kommando lautet in diesem Fall: O,DIR. Auf der Diskette steht das Inhaltsverzeichnis dann unter DIR.OUT. O...(von "Output") erzeugt also stets eine Output-File.

RENAMECMD beschließt die Reihe. Mit diesem Kommando können die Namen und Typkennzeichen der Files geändert werden. So kann z.B. DEMOTXT ein mit dem Textverarbeiter geschriebenes BASIC-Demonstrations-Programm sein. XBASIC kann jedoch eine TXT-File weder laden noch starten. Mit dem Kommando RENAME DEMOTXT DEMO.BAS wird in diesem Fall aus der TXT-Datei eine BAS(Basic Source Code)-Datei.

Alle bisher erwähnten Files (und noch verschiedene andere) gehören zum

Standard-Software-Paket von FLEX. Eine ausführliche Beschreibung findet man im Handbuch "The Flex Disc Operating System", das ebenso wie der "Advanced Programmers Guide" im Lieferumfang enthalten ist. Am Schluß dieses ersten Teils der Beschreibung von FLEX geben wir das Inhaltsverzeichnis der Diskette wieder, auf der die besprochenen Files gespeichert sind. Zuvor jedoch noch eine Anmerkung: FLEX kann mit bis zu vier Diskettenlaufwerken gleichzeitig arbeiten. Wenn nur ein Laufwerk existiert, ist sowohl das System- als auch das Arbeitslaufwerk das Laufwerk 0 (vgl. Erklärung zur ASN-File). Sind mehrere Laufwerke vorhanden, so muß die Laufwerknummer im Anschluß an den Namen der Kommandofile eingegeben werden. Zum Beispiel erzeugt DIR 2 das Inhaltsverzeichnis der in Laufwerk 2 befindlichen Diskette. Kommandos nimmt FLEX übrigens stets nach Erscheinen des Systemprompt "+++ " entgegen.

Bild 4 ist ein typisches Beispiel für ein auf dem Bildschirm ausgegebene Inhaltsverzeichnis. Es gehört zu einer mit 40 Spuren und doppelter Schreibdichte doppelseitig beschriebenen Diskette (1404 Sektoren). Der vom Benutzer eingegebene Text ist unterstrichen.



FLEX-Utilities

Paul Panter

Technical Systems Consultants bietet zur Ergänzung und Unterstützung des Disketten-Betriebssystems einen recht breit gefächerten Satz von Dienstprogrammen (Utilities) an. Dazu gehören auch einige kleinere Hilfsprogramme, die für den ernsthaften FLEX-Anwender eigentlich unverzichtbar sind. Im Rahmen dieses Artikels kann leider nur eine Auswahl aus dem sechsteiligen Dienstprogramm-Set vorgestellt werden, so daß wir uns auf die interessantesten Utilities beschränken müssen.

In der Dokumentation, die zu der Systemdiskette geliefert wird, sind die enthaltenen Utilities selbstverständlich vollständig erläutert.

Bevor wir damit beginnen, hier zuerst die Vorgehensweise: Zu jeder File werden nachfolgend die Syntax, die Default-Werte (Erklärung folgt), ein Beispiel sowie das Ergebnis angegeben. Die Syntax schreibt bekanntlich vor, mit welchen Worten und Zeichen dem Computer ein Kommando mitzuteilen ist. Default-Werte stehen stellvertretend für Werte oder Texte, die vom

Benutzer nicht eingegeben wurden. Der Computer setzt dann von sich aus beispielsweise das Textfile-Kennzeichen .TXT ein. Wenn nachfolgend eine Textfile zur Erläuterung herangezogen wird, so handelt es sich dabei um einen beliebigen Text. Als Textfile-Name wurde der Name "PARAGR1.TXT" gewählt.

Doch nun zur Sache:

FIND.CMD

FINDCMD sucht Textfiles nach einer vorgegebenden Zeichenketten ab und gibt die Zeilen, in der die Zeichenketten gefunden wurde, auf dem Bildschirm oder Drucker aus. Das Kommando, das eigentlich typisch für einen Textverarbeiter ist (dort ist es ebenfalls vorhanden), macht in vielen Fällen den Sprung zum Textverarbeiter überflüssig. Es kann immer dann verwendet werden, wenn in einer Textfile (oder bestimmten anderen Files) ein Buchstabe, ein Wort, eine Zahl oder auch ein ganzer Satz gesucht wer-

den soll. Das Laden von FINDCMD kostet weniger Zeit als das Laden eines Textverarbeiters von durchschnittlicher Länge.

◆ Syntax: FIND,Filename,gesuchte Zeichenkette

◆ Defaults: Filetype-Kennzeichen ist TXT

◆ Beispiel:

```
+++ FIND,PARAGR1,Diskette
```

◆ Ergebnis: Die Ausgabe enthält alle Textzeilen der File PARAGR1, in denen das Wort "Diskette" vorkommt. Zusätzlich ist jeweils die Zeilennummer angegeben.

◆ Ausgabe:

```
+++ FIND,PARAGR1,Diskette
(Zeilenummer)1. Zeile, in der "Diskette"
steht)
(Zeilenummer)2. Zeile, in der "Diskette"
steht) (usw.)
.
.
TOTAL STRING OCCURENCE IS XX
```

```

1 +++ DUMP,SCRHELP.SYS

03 09
03 0A 00 01 43 6F 6D 6D 61 6E 64 73 20 69 6E 20 ____Commands in
74 68 65 20 63 6F 6D 6D 61 6E 64 20 6D 6F 64 65 the command mode
3A 0D 0D 50 41 47 45 3D 4E 09 08 53 65 74 20 6E :__PAGE=N__Set n
65 77 20 70 61 67 65 6C 65 6E 67 74 68 2E 0D 4C ew padelenth,__L
49 4E 45 3D 4E 09 08 53 65 74 20 6E 65 77 20 6C INE=N__Set new l
69 6E 65 6C 65 6E 67 74 68 2E 0D 53 54 41 54 55 inelenth,__STATU
53 3D 54 52 55 45 09 06 45 6E 61 62 6C 65 20 73 S=TRUE__Enable s
74 61 74 75 73 20 6C 69 6E 65 20 69 6E 20 74 65 tatus line in te
78 74 20 6D 6F 64 65 2E 0D 53 54 41 54 55 53 3D xt mode,__STATUS=
46 41 4C 53 45 09 05 44 69 73 61 62 6C 65 20 73 FALSE__Disable s
74 61 74 75 73 20 6C 69 6E 65 2E 0D 49 4E 53 45 tatus line,__INSE
52 54 3D 54 52 55 45 09 06 45 6E 61 62 6C 65 20 RT=TRUE__Enable
61 75 74 6F 20 66 69 65 6C 64 69 6E 67 2E 3D 49 auto fieldno,__I
4E 53 45 52 54 3D 46 41 4C 53 45 09 05 44 69 73 NSERT=FALSE__Dis
61 62 6C 65 20 61 75 74 6F 20 66 69 65 6C 64 69 able auto fieldi
6E 67 2E 0D 46 49 45 4C 44 3D 4E 09 0A 44 65 66 no,__FIELD=N__Def

03 0A
03 0B 00 02 69 6E 65 20 66 69 65 6C 64 20 63 68 ____line field ch
61 72 61 63 74 65 72 20 61 73 20 64 65 63 69 6D aracter as decim
61 6C 20 6E 75 6D 62 65 72 2E 0D 46 49 45 4C 44 al number,__FIELD
3D 27 43 09 09 44 65 66 69 6E 6F 00 46 69 65 6C ='C__Definr
64 20 19 61 72 61 63 74 6F 73 20 41 d charac
53 00 63 68 61 72 0D 4D 45 SCII r
08 44 6F 4F 6D NT="

```

Bild 1. Dies Beispiel für die Ausgabe von DUMP.CMD spricht für sich. Die beiden ersten Bytes eines Sektors enthalten stets die Adresse des folgenden. Bei Files, die nur einen Sektor enthalten, steht dort 00 00. Die File SCRHELPSYS ist übrigens eine Utility, die zu einem Textverarbeitungsprogramm gehört.

Wie das Beispiel zeigt, wird nach den Zeilen, in denen die gesuchte Zeichenkette enthalten ist, auch noch die Anzahl der Fundstellen ausgegeben.

WORDS

WORDSCMD ist ein weiteres nützliches Programm. In einem Flex-Textverarbeiter haben wir ein Äquivalent bisher noch nicht gefunden. WORDS zählt die Wörter und Zeilen von Texten, was bei vorgegebener maximaler oder minimaler Anzahl der Wörter oder Zeilen (zum Beispiel für den Umbruch) sehr hilfreich sein kann.

- ◆ Syntax: WORDS,Filename
- ◆ Defaults: Filetyp-Kennzeichen ist TXT
- ◆ Beispiel:


```
+++ WORDS,PARAGR1
```

◆ Ergebnis: Ausgegeben wird die Anzahl der Wörter und Zeilen, die zum Textfile PARAGR1 gehören. Als Wortgrenzen werden die Leerzeichen zwischen den Wörtern und das Zeichen "neue

Zeile / Cursor nach rechts" erkannt.

- ◆ Ausgabe:


```
+++ WORDS,PARAGR1
TOTAL WORD COUNT IS 1662
TOTAL LINE COUNT IS 144
```

TYPOS

TYPOSCMD ist eine abgewandelte Version von Words. Das Kommando stellt nicht nur die Anzahl der Wörter fest, sondern gibt zusätzlich eine Liste aus, die die Häufigkeit aller im Text vorkommenden Wörter enthält. Hiermit kann man beispielsweise die korrekte Schreibweise eines Wortes kontrollieren oder sicherstellen, daß ein bestimmtes Wort im Text nicht zu oft vorkommt. Da natürlich bestimmte Worte in jedem Text häufig vorkommen (zum Beispiel "der", "die", "das", ...), gibt man eine bestimmte maximale Häufigkeit ein. Tut man dies nicht, nimmt das System automatisch "3" als maximale Häufigkeit an.

- ◆ Syntax: TYPOS, Filename,maximale Häufigkeit
- ◆ Defaults: Filetyp-Kennzeichen ist TXT; die maximale Häufigkeit ist 3 (bei fehlender Eingabe werden alle Wörter ausgegeben, die im Text dreimal oder weniger oft vorkommen).
- ◆ Beispiel:


```
+++ TYPOS,PARAGR1,5
```

◆ Ergebnis: Eine Liste, die alle in der Textfile PARAGR1 fünf, vier, drei, zwei, und einmal vorkommenden Wörter enthält (jeweils zuerst die Anzahl und dann das betreffende Wort). Die Liste wird zunächst nach Häufigkeit und dann nach dem Alphabet sortiert.

◆ Ausgabe: Da die Liste ziemlich lang ist, soll an dieser Stelle auf ihre Wiedergabe verzichtet werden.

SPLIT

SPLITCMD erweist sich bei der Bearbeitung von langen Texten als nützlich. Solche Texte sind oft unübersichtlich, hier kann eine Untergliederung in mehrere Abschnitte die Arbeit erleichtern. SPLITCMD teilt eine Textfile in zwei neue Textfiles auf, wobei die Trennstelle eine Zeilennummer ist.

- ◆ Syntax: SPLIT, Name des Ursprungfile, neuer Name der ersten neuen File, neuer Name der zweiten neuen File, Zeilennummer der gewünschten Trennstelle
- ◆ Defaults: Filetyp-Kennzeichen ist TXT
- ◆ Beispiel:


```
+++ SPLIT,PARAGR1,TEIL1,TEIL2,100
```

◆ Ergebnis: Aus dem Textfile PARAGR1 werden zwei neue Textfiles abgeleitet, von denen die erste "TEIL1" heißt und die Zeilen 1 bis 99 enthält, während die zweite den Namen "TEIL2" trägt und die Zeilen 100 bis zum Ende enthält. Die ursprüngli-

che File PARAGR1.TXT bleibt bei dieser Operation erhalten.

LOW-UP, UP-LOW

LOW-UPCMD und UP-LOWCMD sind Routinen, die kleine Buchstaben in große bzw. umgekehrt große in kleine Buchstaben umwandeln. Eine Umwandlung von Klein- in Großbuchstaben kann zum Beispiel notwendig sein, wenn ein BASIC-Programm für einen Interpreter geschrieben wurde, der beides "versteht". Soll dieses Programm mit einem anderen Interpreter laufen, bei dem nur Großbuchstaben zulässig sind, so müssen die kleinen Buchstaben durch große ersetzt werden.

◆ Syntax: LOW-UP, Name der Ursprungsfile mit Groß- und Kleinbuchstaben, neuer Name der neuen File mit ausschließlich Großbuchstaben.

◆ Defaults: Filetyp-Kennzeichen ist TXT

◆ Beispiel:

```
+++ LOW-UP,KLEIN,GROSS
```

◆ Ergebnis: Aus der Textfile KLEIN, die Klein- und Großbuchstaben enthält, wird die Textfile GROSS erzeugt; in dieser sind nur noch große Buchstaben vorhanden.

Beim Kommando UP-LOW verläuft die Operation analog hierzu in umgekehrter Richtung.

DUMP

DUMPCMD gibt den Inhalt eines Diskettensektors (oder mehrerer Sektoren) auf dem Bildschirm oder Drucker aus. Zuerst erscheinen die Spur- und die Sektoradresse, anschließend werden in den nächsten Zeilen jeweils 16 Byte sowohl in hexadezimaler Schreibweise als auch in der ASCII-"Übersetzung" ausgegeben. Das Kommando kann sehr nützlich sein, wenn eine Diskette beschädigt wurde und sich ihr Inhalt durch einen Sicherheitsausdruck rekonstruieren läßt.

◆ Syntax: DUMP,Filename.Filetyp

◆ Defaults: Filetyp-Kennzeichen ist BIN (binary)

◆ Beispiel:

```
+++ DUMP,SCRHELP.SYS
```

◆ Ergebnis: Auf dem Bildschirm erscheint die Startadresse der File SCRHELP.SYS (Spur- und Sektornummer in hexadezimaler Schreibweise). Es folgen 16 Zeilen mit jeweils 16 Byte und den ihnen entsprechenden ASCII-Zeichen (jeder Sektor enthält 256 Byte). In diesem Format werden alle zur File gehörenden Sektoren ausgegeben.

◆ Ausgabe: Siehe Bild 1.

CHECK

CHECKCMD vergleicht den Inhalt von zwei Files miteinander. Nach dem Vergleich wird abhängig vom Ergebnis eine

der beiden folgenden Meldungen ausgegeben: THE FILES CHECKED DO NOT MATCH oder THE FILES CHECKED ARE IDENTICAL. Das Kommando ist unter anderem nützlich, wenn man den Verdacht hat, daß zwei Files mit gleichem Inhalt, aber unterschiedlichen Namen existieren.

◆ Syntax: CHECK,Filename1,Filename2

◆ Defaults: Filetyp-Kennzeichen ist TXT

◆ Beispiel:

```
+++ CHECK,
PARAGR1.TXT,PARAGR1.BAK
```

◆ Ergebnis: Der Inhalt des Textfile PARAGR1 wird mit dem Inhalt der zugehörigen Backup-File verglichen.

◆ Ausgabe:

```
+++
CHECK,PARAGR1.TXT,PARAGR1.BAK
THE FILES CHECKED ARE IDENTICAL
```

MAP

MAPCMD ist ein Programm, das die Lade- und Arbeitsadressen von Kommandofiles feststellt. Es gibt darüber Auskunft, an welcher Stelle im Speicher die "gemapte" Kommandofile abgelegt und bei welcher Adresse sie ausgeführt wird (diese Adresse nennt man auch "Transfer-Adresse"). MAP wird häufig zusammen mit dem Kommando SAVE gebraucht, das den Inhalt der Speicherblöcke unter einem frei wählbaren Namen auf die Diskette schreibt.

◆ Syntax: MAP,Filename

◆ Defaults: Filetyp-Kennzeichen ist BIN

◆ Beispiel:

```
+++ MAP,MAPCMD
```

◆ Ergebnis: Auf dem Bildschirm erscheinen die Anfangs- und Endadressen der Speicherblöcke, in die MAPCMD eingelesen wird, gefolgt von der Transfer-Adresse.

◆ Ausgabe:

```
+++ MAP,MAPCMD
C100-C126
C100-C1E8
C100
```

FREE

FREECMD gibt den noch verfügbaren Platz auf der Diskette an. Er wird durch die Anzahl der Sektoren sowie in KBytes ausgedrückt.

◆ Syntax: FREE X

(X = Laufwerksnummer)

◆ Defaults: Laufwerk 0

◆ Beispiel: FREE

◆ Ergebnis: Auf dem Bildschirm wird die Meldung ausgegeben:

```
SECTORS REMAINING = X
APPROXIMATE KILOBYTES = YY
```

◆ Ausgabe (Beispiel):

```
+++ FREE
SECTORS REMAINING = 7
APPROXIMATE KILOBYTES = 17
```

TEST

TESTCMD kann viel Ärger und Verdruß beim Gebrauch von Disketten minderer Qualität ersparen. Alle Sektoren werden auf einwandfreie Beschaffenheit getestet, fehlerhafte Sektoren werden durch die Meldung BAD SECTOR und ihre hexadezimale Adresse angezeigt.

◆ Syntax: TEST,Laufwerksnummer

◆ Default: Nummer des mit ASN definierten Arbeitslaufwerks (siehe Artikel: "Das Betriebssystem FLEX" in diesem Heft).

◆ Beispiel: TEST,0

◆ Ergebnis: Die Diskette in Laufwerk 0 wird getestet; eine eventuelle Fehlermeldung lautet BAD SECTOR AT ... Eine nicht existierende Laufwerksnummer hat die Fehlermeldung ILLEGAL DRIVE NUMBER zur Folge.

◆ Ausgabe:

```
+++ TEST,0
BAD SECTOR AT NN-MM
BAD SECTOR AT XX-YY
usw.
```

```
TEST COMPLETED
```

oder:

```
+++ TEST,0
TEST COMPLETED
```

Natürlich wünscht man sich die zweite Meldung, sie bedeutet: Die Diskette ist völlig einwandfrei.

RPT

RPTCMD erleichtert die Arbeit, wenn ein Programm mehrfach wiederholt ausgeführt werden soll. Normalerweise tippt der Benutzer die betreffende Kommandozeile jedesmal neu ein, was zwar möglich, aber nicht sehr effizient ist. RPT (von repeat) macht dies überflüssig, indem es die Kommandozeile so oft wie angegeben wiederholt.

◆ Syntax: RPT,Anzahl der Wiederholungen, zu wiederholende Kommandozeile

◆ Defaults: entfallen

◆ Beispiel: RPT,3,P LIST PARAGR1.TXT

◆ Ergebnis: Der Inhalt des Textfile PARAGR1 wird dreimal ausgedruckt.

PDEL

PDELCMD ist die Abkürzung für "prompting delete". Während beim normalen DELETE-Kommando der Name der zu löschenden File stehen muß, verhält es sich hier anders: Zuerst werden auf dem Bildschirm die Filenamen ausgegeben, wobei die Reihenfolge dem Disketten-Inhaltsverzeichnis entspricht. Hinter jedem Namen steht ein Fragezeichen, das mit Y (yes), N (no) oder mit Carriage Return beantwortet werden kann.

Carriage Return bricht die Ausführung von PDEL ab; das System kehrt dann in den Kommando-Zustand (+++) zurück. Wichtig: PDEL fragt vor der Ausführung nicht "ARE YOU SHURE" (wie DELETE.COM), sondern wird nach der Eingabe von "Y" sofort aktiv! Man kann die Filenamen selektieren, indem man beispielsweise .TXT an das Kommando anfügt. Die Folge ist, daß auf dem Bildschirm nur die Namen von Textfiles erscheinen. Ebenso lassen sich alle mit einem bestimmten Buchstaben oder einer bestimmten Buchstabenkombination beginnenden Filenamen ausfiltern.

◆ Syntax: PDEL,Laufwerk-Nummer,(Auswahlliste)

◆ Defaults: entfallen

◆ Beispiel: PDEL,0,TXT

◆ Ergebnis: Die Namen der Textfiles, die auf der Diskette in Laufwerk 0 stehen, erscheinen hintereinander auf dem Bildschirm. Nach jedem Namen muß entweder Y, N oder Carriage Return gedrückt werden. Bei Y oder N folgt der nächste Filename, bei Carriage Return und dem

letzten Filenamen wird wieder der Kommando-Zustand erreicht.

MEMDUMP

MEMDUMPCMD erzeugt die blockweise Ausgabe des Speicherinhalts als hexadezimale Liste (Hexdump) auf dem Bildschirm. Nach dem ersten Block (jeder Block umfaßt 256 Byte) stehen drei Kommandos zur Wahl: F (forward), B (backward) und Carriage Return. Mit F oder f erscheint der nächste 256-Byte-Block, und B oder b holt den vorherigen Block auf den Bildschirm zurück. Carriage Return dient auch hier zum Abbruch; das System ist dann wieder im Kommando-Zustand.

◆ Syntax: MEMDUMP,hexadezimale Startadresse

◆ Defaults: Fehlt die Eingabe der Startadresse, so nimmt das Programm die Startadresse 0000 an.

◆ Beispiel: MEMDUMP,F000

◆ Ergebnis: Auf dem Bildschirm wird der Inhalt des Speicherbereichs F000...F0FF als Hexdump ausgegeben.

MEMTEST

MEMTESTCMD testet, wie der Name bereits sagt, den Speicher auf einwandfreie Funktion, was natürlich selten schaden kann.

◆ Syntax: MEMTEST,Startadresse,Endadresse

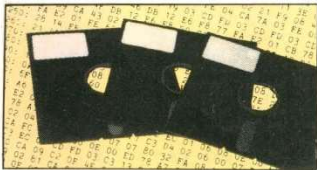
◆ Defaults: werden nicht eingesetzt

◆ Beispiel: MEMTEST,0,01FF

◆ Ergebnis: In den Speicherbereich 0000...01FF werden Zufallszahlen geschrieben, danach werden diese gelesen und mit dem Original verglichen. Die Prozedur wird eine Zeitlang wiederholt, der Test eines 4-K-Blocks dauert ca. 60 Minuten. Natürlich darf nicht der Bereich getestet werden, in dem das Programm MEMTEST selbst steht. Um das zu verhindern, kann man den fraglichen Bereich leicht mit dem Programm MAP lokalisieren. Am Ende eines erfolgreichen Testzyklus erscheint eine "!" auf dem Bildschirm. Abgebrochen wird MEMTEST auf eine etwas rigorose Weise: Durch Drücken des Reset-Tasters.

Wie schon am Anfang erwähnt, konnten wir im Rahmen dieses Artikels nur eine relativ kleine Auswahl aus dem reichhaltigen FLEX-Dienstprogramm-Paket herausgreifen. Den Schluß des Kapitels soll eine alphabetische Liste der beschriebenen Utility-Programme bilden:

```
CHECK
DUMP
FIND
FREE
LOW-UP
MAP
MEMDUMP
MEMTEST
PDEL
RPT
SPLIT
TEST
UP-LOW
WORDS
```



Weiter gehts, möchte man hier sagen: Wie in ec 2 versprochen, folgen hier der zweite und der dritte Teil der Assembler-Source der Steuersoftware zu unserer Grafikkarte. Inzwischen ist in den Elektorheften die Grafikkarte in ihrer Grundversion hardwaremäßig komplett vorgestellt. Im Januarheft geht es mit der Softwarebeschreibung weiter, und im Februarheft wird der Farbausbau beschrieben.

Hier nun also die Teile 2 ("Grafik") und 3 ("Circle") der Steuersoftware für 6502-Computer. Die EC-65-Besitzer können die Maschinenkode-File auf einer Diskette zusammen mit einigen Demonstrationsprogrammen über den Softwareverteiler oder die Fa. Rieger bekommen. Beim

EC-65 liegt die Steuersoftware im Bereich \$C000...CFFF. Sie läßt sich sehr leicht in eine Systemdiskette einbauen:

Bei einer 80Track-Diskette ist noch Platz genug dafür, bei einer 40Track-Systemdiskette muß ein 2 Tracks langes Utility-Programm - halt eines, daß man nicht so häufig benötigt -, entfernt werden (das macht man natürlich mit einer Kopie der Systemdiskette, nicht mit dem Original). Mit den DOS-Befehlen "CALL" und "SAVE" werden die beiden Tracks dann von der Verteilerdiskette geladen und auf der Grafik-Systemdiskette auf den entsprechenden Spuren abgespeichert. Dann wird in das Programm BEXEC* an passender Stelle eine zusätzliche Kommandozei-

le "eingebaut", etwa so:
1905 DISK!"CA C000=60,1!"DISK!"CA
C800=61,1!"DISK!"GO C000"

Achtung! Dies ist ein Beispiel, die Zeilennummer und die Tracknummern können je nach der BECEX*-Version und den für die Grafiksoftware verwendeten Tracks anders sein! Nach dieser Einfügung wird die Grafiksoftware automatisch beim Booten mit geladen, man kann also sofort die Grafik durch BASIC-Befehlszeilen - auch im BASIC-Immediate-Mode - ansprechen.

Da wir gerade von Demonstrationsprogrammen sprachen: Auf dem Innenumschlag finden Sie wieder Farbbilder einiger dieser Programme. Und sie sind

Grafiksoftware

Paul Lavigne
Denis Meyer