

ATARI

ST COMPUTER

Die Fachzeitschrift für den ATARI-ST Anwender.



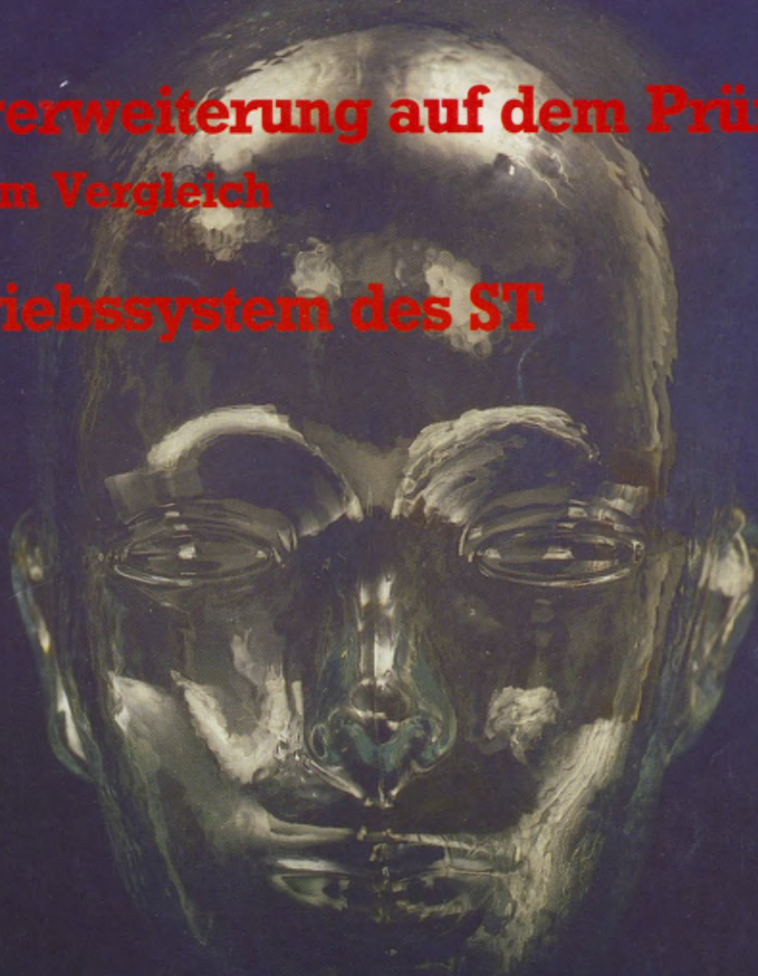
April '86
ÖS 53,—/Sfr. 6,—
DM 6,—
NR. 4



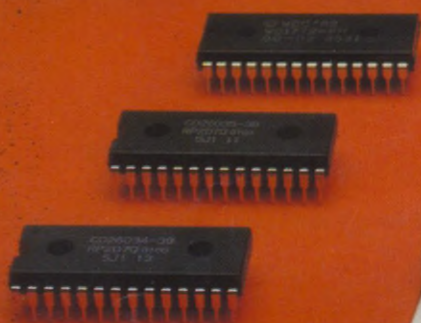
Speichererweiterung auf dem Prüfstand

3 Systeme im Vergleich

Das Betriebssystem des ST



- * **Computerlexikon**
- * **Druckeranpassung
für 1st Word**



**Listings
zum
Abtippen**

SIE IST A!

DIE MASSGESCHNEIDERTE SOFTWARE FÜR IHREN ATARI* 520 ST+

DataExpert

Das Expertensystem. Künstliche Intelligenz in der Datenverwaltung.

Anwendungsbereiche:
Geschäfts-, Hobby-, Privatbereich.
Einsatzmöglichkeiten:
unerschöpflich.

SCS FIBU

SCS FIBU ermöglicht professionelle Finanzbuchhaltung mit Ihren Atari. Sehr einfache Bedienung, auch durch den Computerlaien, und Übersichtlichkeit zeichnen dieses Programm aus.

LAGER

LAGER verwaltet neben Ihren Artikeln auch Ihre Lieferanten. Artikellisten, Lieferantenlisten, Lagerbestand überprüfen, Inventurprotokoll sind bei diesem Programm selbstverständlich. Sehr einfach anwendbar. Ideale Erweiterung zu FAKT.

**DataExpert DM 398, -- DECIDE DM 398, -- SCS FIBU DM 498, --
FAKT DM 248, -- LAGER DM 248, -- ADRESS DM 148, --**

Unverbindliche Preisempfehlungen incl. Mwst.

UNSERE PRODUKTE ERHALTEN SIE BEI FÜHRENDEN FACHHÄNDLERN (siehe Händlerverzeichnis rechts) oder direkt bei uns.

**Direktbestellungen richten Sie bitte an:
SCS SOFTWARE, Stefan Seucan, POSTFACH 2444,
8600 BAMBERG, TEL.: 09542/8348**

Lieferung erfolgt per Rechnung. ATARI ist ein eingetragenes Warenzeichen.

Für DataExpert und DECIDE:
DISTRIBUTOREN IN ENGLAND UND IN DEN USA GESUCHT.

DECIDE

Entscheidungen treffen durch künstliche Intelligenz. DECIDE betrachtet alle Faktoren objektiv und trifft immer die richtige Auswahl. Das Programm hilft Ihnen nicht nur Geld sparen.

FAKT

Die Fakturierung, die auch Ihnen ein unproblematisches Arbeiten ermöglicht. Zwei Adressen pro Kunde, mehrere Preise pro Produkt, individuelle Preisänderungen etc. machen dieses Programm sehr flexibel und somit universell einsetzbar. FAKT ist selbstständig lauffähig, kann aber mit ADRESS und LAGER zu einem Komplettsystem ausgebaut werden.

ADRESS

ADRESS ersetzt Ihre Karteikarten und verschönt Sie vor lästiger und zeitaufwendiger Auswert-Sortierarbeiten. Mit ADRESS sind Listen und Etiketten blitzschnell erstellt.

SCS Software-Produkte erhalten Sie bei:

1000 Berlin 31, ALPHA COMPUTERS E. MUTH GmbH, Kurlfürstendamm 121A, Tel. 030/891092
1000 Berlin 10, COMPUTARE ELEKTRONIK VERTRIEB, Behmstraße 3, Tel. 030/2139021
1000 Berlin 30, RUNOW BÜROTECHNIK, Kienhöfstraße 26, Tel. 030/2611128
1000 Berlin 12, DIGITAL COMPUTER, Knebeckstraße 76, Tel. 030/8927791
2000 Hamburg 20, BIT COMPUTERSHOP GMBH, Osterstraße 173, Tel. 040/494400
2000 Hamburg 71, CREATIME MICROCOMPUTER, Bramfelder Chaussee 300, Tel. 040/6416473
2000 Hamburg 76, C&I GmbH, Wandsbücker Chaussee 59, Tel. 040/2512418/7
2000 Hamburg 54, BASTIAN HARMS KG, Münsterstraße 9, Tel. 040/56601-1
2000 Nordstedt, SELLHORN GMBH, Ullburger Straße 2, Tel. 040/5273047
2000 Pinneberg, BPO GES. F. PLANUNG - DRG GMBH, Dingshede 34, Tel. 04101/26071/2
2120 Lüneburg, SIENKNECHT BÜROKOMMUNIKATION, Heilgengeststraße 20, Tel. 04131/46122
2160 Stade, ERNST BERGAU GMBH, Neue Straße 5, Tel. 04141/2364-84
2210 Itzehoe, DER COMPUTERLADEN, Holtenauer 12, Tel. 04821/3390
2300 Braunschweig, COMPUTER STUDIO GMBH, Rehmstraße 49-50, Tel. 0531/332277-79
2300 Kiel 1, MCC MICRO COMPUTER CHRIST, Drecksploß 7, Tel. 0431/567042
2370 Rendsburg, FK BÜROTECHNIK, Thormannplatz 20/22, Tel. 04331/7321
2390 Flensburg, ELEKTRONIC-COMPUTER-LADEN OHG, Nordstr. 94-96, Tel. 0461/28191/28193
2400 Lübeck, JESSEN UND LENZ, Wahnstraße 38, Tel. 0451/705030
2800 Bremen, BRINKMANN, Oberstraße 52/59, Tel. 0421/3089
2800 Bremen, PS-DATA, Dovenströmweg 41, Tel. 0421/17057
2850 Bremerhaven, KURT NEUMANN PERSONALCOMPUTER, Georgstraße 71, Tel. 04171/302129
2900 Oldenburg, CARL WOLTE GMBH & CO. KG, Heilgengeststraße 6, Tel. 0441/404589
2950 Leer, S & F DATENTECHNIK GMBH & CO. KG, Augustenstraße 3, Tel. 0491/4589
3000 Hannover 1, TRENDDATE COMPUTER GMBH, Am Maistal 18-22, Tel. 0511/16605-0
3000 Hannover 1, DATALOG, Cahlenberg Straße 28, Tel. 0511/326499
3000 Hannover 1, COM-DATA, Am Schiffgraben 19, Tel. 0511/326336
3040 SOLTAA, F & T COMPUTERVERTRIEB, Am Hornberg 1, Tel. 05191/16522
3100 Celle/OT Westercelle, L. HAUPT JR., Gerhard-Kamm-Straße 2, Tel. 05141/93045-47
3170 Gifhorn, COMPUTER HAUS GIFHORN, Braunschweiger Straße 50, Tel. 05371/54499
3250 Hameln, WITTE BÜROTECHNIK, Bückersstraße 22, Tel. 05151/7595
3700 Paderborn, GETS, FÜR ELEKTRONISCHE TE., in Schilders 15, Tel. 0531/332277-79
3500 Kassel, HERMANN FISCHER GMBH, Rudolf-Schneider-Straße 5, 9-13, Tel. 0561/10000-0
3550 Marburg/Lahn, LWM COMPUTER SERVICE, Biegenstraße 43, Tel. 06421/62236
4000 Düsseldorf, DATA BECKER, Merowingerstraße 30, Tel. 0211/310010
4000 Düsseldorf, HOCO EDV ANLAGEN GMBH, Flugstraße 47, Tel. 0211/776270
4050 Mönchengladbach, COMPUTER COMMERCE, Mendenburgerstraße 249, Tel. 02161/19764
4100 Duisburg, COM-DATENSYSYSTEME, Rathausstraße 10, Tel. 0203/559269
4150 Krefeld, SVI GMBH, Carl-Wilhelm-Straße 27, Tel. 02151/64295
4200 Borken, HI-TRONIC, Neutor 3, Tel. 02961/63336
4350 Recklinghausen, MICRO-DATA HINZE, Herne Straße 24, Tel. 02361/12926
4400 Münster, BASIS COMPUTER SYSTEME GMBH, Damlingsweg 39, Tel. 0251/71995/9
4422 Ahaus, ORGANISATIONS- UND COMPUTERERATINGS GMBH, Weidstr. 3, Tel. 02561/5021
4500 Osnabrück, HEINICKE ELEKTRONIK, Kommandantenstraße 120, Tel. 0541/92789
4600 Dortmund 1, CC COMPUTERSTUDIO GMBH, Elsayehstr. 5, Tel. 0231/529184
4600 Dortmund, CITY ELEKTRONIK, Güntherstraße 75, Tel. 0231/529033
4630 Bochum, BU-DATA COMPUTER, Querenberger Höhe 209, Tel. 0234/701022
4650 Gelsenkirchen, BEATE VOLLRAATH COMPUTER, Georgstraße 4, Tel. 0209/208261
4780 Paderborn, GETS, FÜR ELEKTRONISCHE TE., in Schilders 15, Tel. 0531/332277-79
4800 Bielefeld 1, CSF COMPUTER & SOFTWARE GMBH, Heegerstr. 106-109, Tel. 0521/61663
4830 Gütersloh, COMPUTER STORE, Schulstraße 9, Tel. 05241/12090
5000 Köln, BÜROMASCHINEN BRAUN AM RUDELPLATZ GMBH, Richard-Wagner-Str. 39, Tel. 0221/219171
5010 Bergheim, COMPUTERSTUDIO HOLSCHER, Zeppenstraße 7, Tel. 02271/62086
5060 Bergisch Gladbach, COMPUTER CENTER, Buchholstraße 1, Tel. 02202/35053
5240 Betzdorf/Sieg, BYTE ME COMPUTERSYSTEME, Wilhelmstraße 25, Tel. 02241/23537
5450 Neuwied, STAADT COMPUTERTECHNIK, Mittelstraße 92, Tel. 02631/29647
5451 Straßburg, Dr. AUMANN GMBH, Schulstraße 12, Tel. 02634/4091
5500 Trier, BÜROCENTER LEHR GMBH, Güntersstraße 92, Tel. 0651/25044
5540 Prüm, ATC COMPUTERSYSTEME, Kalvarienbergstraße 34, Tel. 06551/3493
5630 Remscheid, COM-SOFT, Schindlerstraße 12, Tel. 02191/2033-34
5880 Lüdenscheid, HAMPE ELEKTRONIK GMBH, Hochstraße 1, Tel. 02351/25359
6000 Frankfurt/Main, SCHMITT COMPUTER SYSTEME, Große Friedberger Straße 30, Tel. 069/294065
6000 Frankfurt/Main, MÜLLER & NEMCEK GMBH, Kaiserstraße 44, Tel. 069/232544
6100 Darmstadt-Eberstadt, HEIM OHG BÜROTECHNIK, Heideburger Landstraße 184, Tel. 06151/55375
6330 Wetzlar/Lahn, KA-WC COMPUTERERATINGS, Bahnhofstraße 14, Tel. 06441/410
6400 Neuheim, COMPUTER PROFESSIONAL GMBH, Hauptstraße 92, Tel. 06032/2098/9
6500 Mainz, ELPHOTEC COMPUTER SYSTEME GMBH, Schlegelgasse 7, Tel. 0631/231947-9
6520 Worms, ORION COMPUTER SYSTEME, Friedrichstraße 22, Tel. 06241/6757-8
6700 Ludwigshafen, MKV GMBH, Im Bismarck Zentrum/Bismarckstraße 71, Tel. 0621/525495/525596
6750 Kaiserslautern, GOTHOLD BÜROCENTER, Im Alstadparkhaus, Tel. 0631/63074
6800 Mannheim, GÄUCH & STURM, Casselestraße 74-76, Tel. 0621/350040
6900 Heidelberg, JACOM COMPUTERTECHNIK GMBH, Mönchhofstraße 3, Tel. 06221/410514
7000 Stuttgart 50, BNT COMPUTERF. SEIBEL & CO. OHG, Markstraße 49, Tel. 0711/559383
7030 Böblingen, MCA COMPUTER CENTER, Sindelinger Allee 1, Tel. 07031/223619
7080 Aalen, BÖHMER ELECTRONIC, Wilhelm-Zapl-Straße 9, Tel. 07361/62686
7100 Heilbronn, FRITZ SEEL GMBH & CO. KG, Am Wolhaus 8, Tel. 07131/69401-03
7100 Heilbronn, WALKER & CO., Monchsstraße 98, Tel. 07131/60049
7170 Schwäbisch Hall, BÜRO-ORGANISATION LINDE GMBH, Heimbacher Gasse 5, Tel. 0791/6701-7319
7450 Hohenheim, SRE, Schloßplatz 3, Tel. 07141/14507
7480 Sigmaringen, SOFT & EASY GMBH, Rapp Gasse 1, Tel. 0751/12493
7500 Karlsruhe, MKV GMBH, Ruppertstraße 20, Tel. 0721/373071
7500 Karlsruhe, PAPERHAUS ERHARDT GMBH & CO. KG, Am Ludwigplatz, Tel. 0721/23925
7520 Bruchsal, HELMUT JOSEF VIDEO CENTER, Weimer-von-Siemens-Str. 47A, Tel. 07821/10309/92
7530 Pforzheim, DM-TECHNIK METAKOLO GMBH, Durracher Straße 39, Tel. 07231/13939
7600 Offenburg, LEONHARDT-ELECTRONIC, In der Jeuch 3/Gew.-Geb. Waltersw., Tel. 0781/57974
7700 Singen, UDO MEIER, Am Posthalterwäldchen 8, Tel. 07731/44211
7707 Eggen, HARTMUT SCHABER COMPUTER ANLAGEN, Industriestraße 4A, Tel. 07733/9401
7900 Um/Donau, COMPUTER STUDIO CLAUS WECKER, Haltenhof 19/1, Tel. 0731/68076-77
7920 Heidenheim, EXPERT DORASZELSKI, Wilhelmstraße 45-49, Tel. 07321/45021-23
9000 München 40, COMP-CARRY COMP. SYST. V. GMBH, Ingolstädter Str. 20-24, Tel. 089/3587091
9000 München 82, MSD MARKETING UND SERVICE GMBH, Adelmanstraße 5, Tel. 089/4300333
9000 München 2, SCHULZ COMPUTER, Schillerstraße 22, Tel. 089/587330/39
9120 Weidheim, VIKTOR KLEMENT, Admiral-Höfer-Straße 1, Tel. 0939/4500
9170 Bad Tölz, ELEKTRONIK CENTER BAD TÖLZ, Wachterstraße 3, Tel. 09041/41565
9200 Rosenheim, SZEREDY COMPUTING, Theodor-Giet-Straße 3, Tel. 09931/69021
9300 Landshut, BÜRO-DALLMER Altsiedl 69, Tel. 0971/21062/21064
9400 Regensburg, C-SOFT GMBH, Holzfallerstraße 4, Tel. 0941/93996
9480 Cham, A & P SHOP, Auf der Schanze 4, Tel. 09971/9727
9500 Nürnberg, HIB, Aubere Bayernstraße 72, Tel. 0911/515939
9500 Bamberg, BÜROZENTRUM A-R KUTZ, Am Kranen 12A, Tel. 0951/27909
9570 Hof, COMPUTER CENTER BURGER, Leimitzer Straße 11-13, Tel. 09281/44075
9700 Würzburg, HALLER GMBH, Büttner Straße 29, Tel. 0931/16705
9720 Schweinfurt, UHLENHUTH GMBH, Albrecht-Dürer-Platz 2, Tel. 09721/652154
9752 Münchberg, COMPUTOUT, Im Kahlhof Zentrum, Tel. 09029/6520
9800 Ansbach, RUDOLPH BUSCH GMBH, Usterstraße 9-10, Tel. 0991/2144
9900 Augsburg, ADDOL & SCHMOLL COMPUTER STUDIO, Horststraße 6, Tel. 0921/529533
9940 Memmingen, EDV-SCHWEITZER KG, Babinger Straße 34, Tel. 09331/12220
9960 Kempten, STAHLIN, Gerbersstraße 32/34, Tel. 0931/28001

und in den Filialen der Kaufhäuser Karstadt und Kaufhof



Nicht nur ein Basic Rechner

Liebe ST Leser!

Basic ist zweifellos einer der verbreitetsten Programmiersprachen, zumindest bei Home- und Personal-Computern. Dies liegt nicht zuletzt daran, daß Basic wirklich leicht zu erlernen ist und bei den meisten Rechnern bereits fest im ROM eingebaut ist oder auf einem Datenträger mitgeliefert wird. Basic hat aber auch seine Schattenseiten, die ich hier nicht aufzählen will, man denke nur an die schlechte Lesbarkeit langer Basic-Listings (Strukturierung). Jeder ST-Besitzer kennt wohl mittlerweile einige Fehler seines Basic-Interpreters. Auch wenn diese Fehler nun (nach der CeBit) durch einige neue Basic Interpreter bzw. Compiler verschwinden, wollen wir Ihnen andere wichtige Programmiersprachen nicht vorenthalten, die den 68000er Prozessor besser ausnutzen. Als maschinennahe Hochsprache bietet sich besonders 'C' an und im mathematisch-naturwissenschaftlichen Bereich Pascal. Für den Pascal Anwender haben wir in diesem Heft ein sehr nützliches Programm abgedruckt, daß das Arbeiten mit dem ST Pascal Compiler wesentlich vereinfacht und beschleunigt. Aber auch 'C' haben wir in dieser Ausgabe nicht vergessen, bevor wir im nächsten Heft einen ausführlichen Kurs in dieser Sprache beginnen. Wem dies nicht ausreicht, der kann sich die Sprache Forth in unserem Public Domain Service kostenlos kopieren lassen. Was dabei zu beachten ist, steht am Ende des Heftes. Wir werden Sie natürlich auch über alle anderen Programmiersprachen informieren, denn wir meinen, der ST ist zu schade um ihn nur in Basic zu programmieren.

Allgemeines

Editorial 1

Impressum 80

Computer Lexikon 72

Software

1st Word
komfortable Druckeranpassung 51

ST-Betriebssystem 25

USCD p-System 20

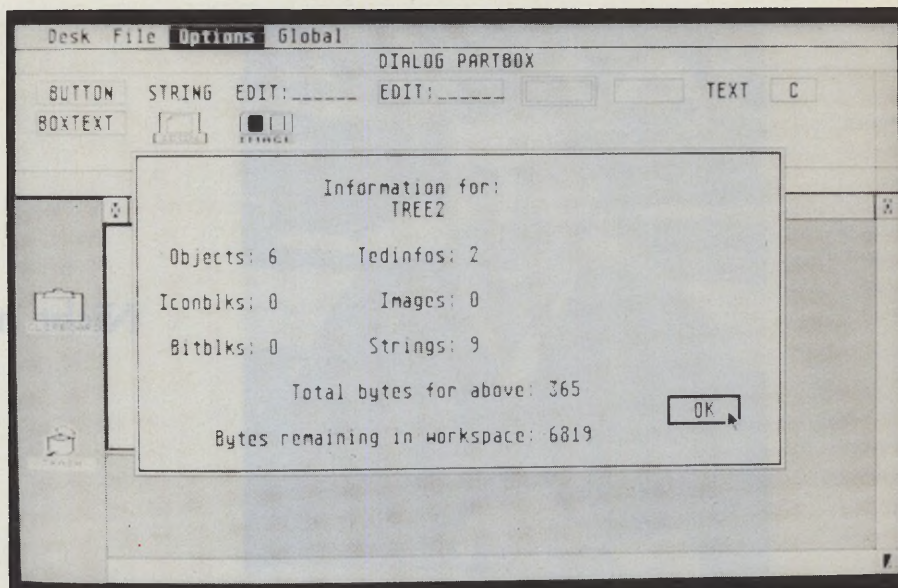
C
– Esperanto der Computersprache 17

RCS
– Resource Construction Set 69

Deskaccessoires contra Multitasking 56

CP/M

Datenbanksysteme & dBase II 30



Hardware

TEAC Laufwerk FD-55 F 45

Speichererweiterungen auf dem
Prüfstand 46

Tips zum Monitor 49

Nachtrag 3 1/2" Floppy 50

Listing

82-Track Formatierprogramm 4

Menüsteuerung für ST-Pascal 8

Mauseditor 13

Vokabeltrainer 66

Kurse

Einführung in GEM (Teil 3) 39

Einführung in die Programmier-
sprache PASCAL (Teil 3) 33

Aktuelles

Leserbriefe 76

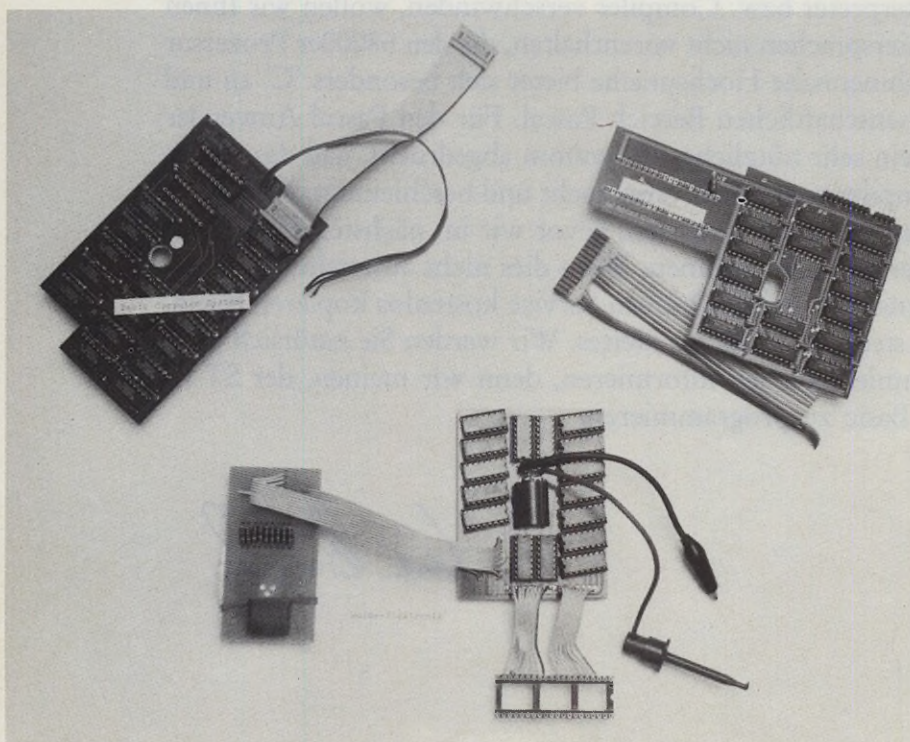
Kleinanzeigen 78

Public Domain 79

Vorschau 80

Einkaufsführer 58

News & Infos 3



Jack Tramiel schlägt schon wieder zu

Gerade vor einem Jahr sorgte ATARI auf der Hannover-Messe für großes Aufsehen, als der 520ST zu einem ungewöhnlich niedrigen Preis vorgestellt wurde. Nach der erfolgreichen Einführung der beiden ST Modelle, gibt es nun, unter dem gleichen bekannten Motto „Power Without The Price“ ein weiteres ST Modell. Mit der Einführung des 1040 ST/F zur CEBIT am 12.3.86 werden außerdem die Preise für die anderen Modelle stark gesenkt. Alle drei Rechner sind voll kompatibel. Eine saubere Marktpolitik, die bei einigen anderen Computerherstellern nicht selbstverständlich ist. Von der Aufmachung ähnelt der 1040 ST/F stark den beiden bekannten Modellen 520 ST+ und 260 ST, jedoch ist auf der rechten Seite ein Diskettenlaufwerk mit 720 Kilobyte eingebaut. Weiterhin ist ein leistungsstarkes Schaltnetzteil für Rechner und Floppy im Gehäuse integriert. Damit entfällt endlich der lästige Kabelsalat. Technisch gesehen entspricht der 1040 ST/F einem 520 ST+ mit einer Floppy SF 314. Die Leistung dieses neuen kompakten Rechners muß allerdings bezahlt werden; So kostet der 1040 ST/F mit Monochromen-Monitor DM 3 298,- incl. MwSt. und ist ab sofort lieferbar.

Die Preise der bekannten Modelle ändern sich folgendermaßen:

520 ST+ mit der Floppy SF 314 und Monochromen-Monitor einschließlich der ROMs (die das TOS enthalten) DM 2 898,- incl. MwSt. Der Preis für den 260 ST wird auf DM 998,- incl. MwSt. reduziert.



Neues TOS

Ab sofort kann man sich bei jedem ATARI Vertragshändler eine neue TOS-Diskette kopieren. Diese Version ist die ROM-TOS Version vom 6.2.86.

Matrixdrucker von ATARI

Ab April wird es den Matrixdrucker SMM 804 zum Preis von DM 698,- incl. MwSt. von ATARI geben.

Der ST wird IBM kompatibel

Eine weitere sensationelle Neuheit ist der MS-DOS Emulator von ATARI. Um Geschwindigkeitsprobleme zu vermeiden, wurde auf eine Hardware Lösung zurückgegriffen. So entstand auf einer separaten Platine in einem eigenen Gehäuse ein quasi eigenständiger Rechner, der an den ST angeschlossen wird. Die Emulator-Platine beinhaltet einen eigenen Prozessor, der mit 8 MHz getaktet wird, 512 Kilobyte RAM-Speicher, sowie ROMs, in denen das BIOS enthalten ist. Außerdem ist ein freier Sockel für einen Arithmetik-Coprozessor vorhanden. Im Gehäuse ist neben der Platine und dem Netzteil noch Platz für ein 5 1/4-Zoll Laufwerk vorhanden. Mit diesem Emulator sollen fast alle IBM kompatiblen Programme auf dem ATARI ST lauffähig sein. Der Preis stand leider noch nicht fest, soll aber die DM 1 000,- Grenze nicht überschreiten. Die Einführung des Emulators soll noch im zweiten Quartal dieses Jahres stattfinden.

5 1/4-Zoll Festplatte von ATARI

Nach der CEBIT wird es eine 20 MB Festplatte zum Preis von ca. DM 2 000,- geben.

Speicheroszilloskop für den ST

Jetzt lassen sich auch einmalige oder aperiodische Mess-Vorgänge mit einem ATARI ST aufnehmen und anzeigen. Die Software ist GEM unterstützt und erlaubt das Einlesen von bis zu 20 000 Werten in der Sekunde. Die Messgenauigkeit umfaßt 8-Bit. Das Speicheroszilloskop kostet zusammen mit der Software DM 498,- und ist bei PRINT-Technik in München zu bekommen.

Terminal Programm von KUMA

Die englische Firma KUMA präsentiert K-COMM, ein neues voll GEM unterstütztes Terminalprogramm. Trotz leistungsfähiger Merkmale, läßt sich das Programm leicht bedienen. Das Programm wurde Anfang März offiziell in London zu einem Preis von 50 Pfund (ca. DM 180,-) vorgestellt.

CP/M

Die Firma Markt & Technik bietet ab sofort das Welt-bekannteste Textverarbeitungs Programm **WORDSTAR** für den ATARI ST an. Das Programm läuft mit dem CP/M Emulator der zum Nulltarif beim Händler oder bei uns zu beziehen ist.

SM Software

Das Programm SM-Manager: Text der Münchner Firma SM Software ist nun in einer verbesserten Version zu erhalten. Die wohl bekannten Mankos die das Programm hatte, sind aufgehoben. Die alte Version kann gegen Einsendung der Diskette und einer Update-Gebühr von etwa DM 39,- bei SM umgetauscht werden.

Platinenservice

Ab sofort können Sie die beiden Leiterplatten der Märzangabe über den Heim-Verlag beziehen. Die Platinen sind bestückungsfertig ausgesägt, gebohrt und verzinkt.

Floppy-Stecker-Platine (FSP) ST 001ub DM 8,80 incl. MwSt
Treiber- und Netzteilplatine ST 002ub : DM 19,80 incl. MwSt

Der Versand erfolgt nur per Nachnahme plus DM 3,- für Porto und Verpackung.

Formatieren auf 408 bzw. 828 kByte

FATDISK ist ein Formatierprogramm, daß die Speicherkapazität der beiden ATARI Floppies voll ausnutzt. Nach Formatieren mit diesem Programm steht der folgende Speicherplatz zur Verfügung.

SM 354 408 576 Bytes
SM 314 828 416 Bytes

Dies erreicht man durch zusätzliches Formatieren der Tracks 80–82, die normalerweise nicht benutzt werden. Der Zugriff erfolgt problemlos, sowohl bei Einzel- als auch bei Diskcopy. Die Disketten können deshalb ohne Schwierigkeiten mit dem normalen GEM-Disk-Copy (Laufwerk A nach B) kopiert werden.

Die Balken, die beim Kopieren einer ganzen Diskette die gelesenen bzw. die geschriebenen Tracks anzeigen, wandern dabei über das eingerahmte Feld hinaus. Das hier abgedruckte Listing entspricht dem Formatierprogramm auf der zweiten Public-Domain Diskette, die bei der Redaktion zu beziehen ist. Somit ist es auch für 'Nicht C Programmierer' möglich die Kapazität der Floppy ein wenig zu erhöhen.

Detlef & Andreas Hoepfner

FATDISK

DISK-Information

Diskstationskennung: A:
Disketten-Name: _____
Anzahl Ordner: ___0___
Anzahl Dateien: ___0___
Belegte Bytes: ___0___
Freie Bytes: ___408576___

OK

DISK-Information

Diskstationskennung: A:
Disketten-Name: _____
Anzahl Ordner: ___0___
Anzahl Dateien: ___0___
Belegte Bytes: ___0___
Freie Bytes: ___828416___

OK

```
/* Formattierprogramm ca.400Kbyte einseitig oder 800Kbyte doppelseitig */
```

```
extern long xbios();  
extern long gemdos();
```

```
/* ein paar Macros braucht man schon..... */
```

```
#define format(a,b,c,d,e,f,g,h,i) xbios(0x0a,a,b,c,d,e,f,g,h,i)  
#define WHICH 9  
#define getchar() gemdos(0x08)  
#define writes(a,b,c) xbios(0x09,a,01,0,c,b,0,1)  
#define makeboot(a,b) xbios(0x12,a,b,2 + (s1 ? 1 : 0),0)
```

```
#define cur_off() printf("\033f")  
#define cur_on() printf("\033e")  
#define SERIAL 0x123456781
```

```
static char *message[] = {  
/* 0 */ "Zu formatierende Diskette in Laufwerk A: legen\0",  
/* 1 */ "Formatierfehler kann diese Diskette nicht formatieren !\0",  
/* 2 */ "Moechten Sie noch eine Diskette formatieren ?\0",  
/* 3 */ "unused\0",  
/* 4 */ "Diskette formatiert\0",  
/* 5 */ "Formatiere Diskette...\0",  
/* 6 */ "unused\0",  
/* 7 */ "unused\0"  
}
```

```

static char *line[] = {
    "\033E          D I S K F O R M A T T E R 410Kb\n\r\0",
    "===== \n\r\n\r\n\r\0",
    "          \275 1985 by BestSoft Paderborn\n\r\n\r\0",
    "          written 1985\n\r\0",
    "          by\n\r\n\r\0",
    "          Detlef H. & Andreas H.\n\r\n\r\0"
};

long do_format(buffer,sides)    /* Formatiererroutine */
long *buffer,sides;
{
    char track;

    cur_off();                /* Cursor aus */
    for(track = 0; track < 83; track++)
    {
        if( 0 == format(buffer,0l,0,10,track,0,1,0x87654321,0l) &&
            (sides ? 0 == format(buffer,0l,0,10,track,1,1,0x87654321,0l) : 1))
            printf("\rFormatiere Track: %d",track);
        else {
            printf("\n\rKann track %d nicht formatieren !",track);
            cur_on();
            return(2);
        }
    }
    cur_on();                /* Cursor an */
    return(0);
}

long copyright()              /* Bildschirmtop anzeigen */
{
    int x;
    cur_off();
    for(x = 0; x < 6; printf("%s",line[x++]));
    cur_on();
    return(SERIAL);
}

main()                        /* Hauptprogramm anfang */
{
    char c,serien_nr[6],*sbuff;
    int count,track,sector;
    long buffn[2048],ser_number,s1,*buff;
    buff = buffn;
    do
    {
        copyright();
        sbuff = buff;

        do
            printf("\n\r(E)inseitig oder (Z)weiseitig ?");
        while((c = getchar()) != 'E' && c != 'e' && c != 'Z' && c != 'z');

        (s1=(c=='Z'?!c=='z')) ? printf("\rZweiseitig") : printf("\rEinseitig");
        printf("\n\r");

        printf("\n\r%s",message[0]);    /* Diskette einlegen */
        c = getchar();
        printf("\n\r%s\n\r",message[5]); /* Formatiere..... */
        if(!do_format(buff,s1))
        {
            for(count = 0; count < 512; *(sbuff + count++) = 0); /* Bootsektor */
            makeboot(sbuff,0l); /* schreiben */
            *(sbuff + 0x13) = (820 + (s1 ? 820 : 0)) & 0xff;
            *(sbuff + 0x14) = ((820 + (s1 ? 820 : 0)) >> 8) & 0xff;
            *(sbuff + 0x18) = 10;
            writes(sbuff,0,1);
            for(count = 0; count < 512; *(sbuff + count++) = 0);
            *(sbuff) = 0xf7;
            *(sbuff + 1) = 0xff;
            *(sbuff + 2) = *(sbuff + 1);
            writes(sbuff,0,7);
            writes(sbuff,0,2);
            sprintf(sbuff,"BestSoftfmt\010");
            writes(sbuff,1,3);
            printf("\n\r%s\n\r",message[4]);
        } else printf("\n\r%s",message[1]);
        printf("\n\r\n\r%s",message[2]);
    }
    while(!((c=getchar) == 'n' !! c == 'N')); /* Nochmal ? (J/N) */
}

```

GRAFIK HOCH 3



Grafik und Sound auf dem ATARI ST. Ein Traum wird wahr! Grafikgrundlagen, Animationsgrafik, Funktionsdiagramme, 2-D/3-D Grafik, CAD, Soundgrundlagen und das MIDI-Interface sind nur einige Schwerpunkte dieses Buches. Alle Beispiele sind gründlich erklärt und mit vielen Beispielprogrammen verdeutlicht. Werden Sie zum Bildschirmkünstler und Computerdirigenten.
ATARI ST Grafik & Sound
249 Seiten, DM 49,-



Mit diesem Buch wird Ihnen die Erstellung von 3D-Grafiken in Maschinensprache leicht gemacht. Von einer Einführung in Assembler über die nötige Theorie bis zur Grafikanimation in atemberaubender Geschwindigkeit reicht das Spektrum dieses Buches. Außerdem enthält es spezielle Grafikroutinen, die schneller sind als alles bisher dagewesene. Da wird Echtzeitanimation erst möglich.
3D-Grafikprogrammierung, zum ATARI ST
ca. 300 Seiten, DM 59,-



Damit Sie die hervorragenden Fähigkeiten Ihres Rechners richtig ausnutzen können, brauchen Sie auch die entsprechende Software. Zeichenprogramme wie GEM-DRAW, DEGAS oder NEOCHROME sprechen für sich. Dieses Buch beinhaltet nicht nur ausführliche und leicht verständliche Beispiele, sondern liefert auch wertvolle Tips und Tricks beim Umgang mit diesen Programmen.
ATARI ST Grafikanwendung
ca. 200 Seiten, DM 29,-

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (02 11) 31 00 10

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,- Versandkosten

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name _____

Straße _____

PLZ/Ort _____

Name und Adresse bitte deutlich schreiben.

Superbücher zum ATARI ST



Der neue ATARI ist eine Supermaschine! Aber nur der richtige Einstieg garantiert den professionellen Umgang damit. Deshalb sollte dies Ihr erstes Buch sein. Eine Einführung in Handhabung, Einsatz und Programmierung des ST: die Tastatur, die Maus, der Editor, der erste Befehl, das erste Programm, der Anschluß der Geräte u.v.m. Dieses Buch ist ein Muß für jeden Einsteiger!
ATARI ST für Einsteiger, 262 Seiten, DM 29,-



Das Informationspaket zum ATARI ST mit ausführlicher Hardwarebeschreibung, detaillierter Erläuterung der Schnittstellen: V.24, Expansion-Interface, Aufbau und Funktionsweise der Maus, MIDI-Interface, Aufbau von Grafiken, BIOS, GEM, wichtige Systemadressen und was man damit machen kann. Unentbehrlich fürs professionelle Arbeiten mit dem ATARI ST.
ATARI ST INTERN, 464 Seiten, DM 69,-



Sie haben den Einstieg auf dem ATARI ST geschafft? Dann werden Sie mit diesem Buch zum Profi. Aus dem Inhalt: Datenfluß- und Programmablaufpläne, Grafik- und Soundprogrammierung, Sortierverfahren, Dateiverwaltung und viele nützliche Tips. Mit einer Befehlsübersicht incl. der nicht bekannten Befehle!
Das große BASIC-Buch zum ATARI ST, 268 Seiten, DM 39,-



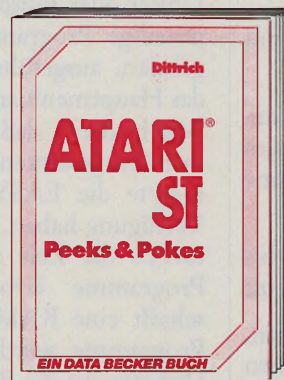
LOGO ist keineswegs nur eine Sprache für Kinder, sondern eröffnet viele interessante Bereiche wie z.B.: Rechnen mit LOGO, Grafikprogrammierung, Wörter- und Listenverarbeitung, Prozeduren, Rekursionen, Sortierroutinen, Maskengenerator, Datenstrukturen und Künstliche Intelligenz. Mit LOGO können Sie schwierige und komplexe Probleme oft leichter lösen als mit anderen Programmiersprachen!
Das LOGO-Buch zum ATARI ST, ca. 300 Seiten, DM 49,-



Sie können BASIC und wollen „C“ lernen? Mit diesem Buch kein Problem! Die elementaren Grundelemente wie Bildschirmoperationen, Variablen, Zeiger, arithmetische Ausdrücke und Kontrollstrukturen werden als Einführung benutzt, um weiterführende Sprachelemente wie Datenfelder, Strukturen und Funktionen zu erklären. So können Sie die Stärken von „C“ schnell für eigene Programme ausnutzen!
Von BASIC zu C mit dem ATARI ST, 297 Seiten, DM 39,-



Eine riesige Fundgrube faszinierender Tips & Tricks um ihren ATARI ST voll auszunutzen! Benutzung des ATARI-BASIC, Programmierung einer RAM-Disk, Druckerspooles und Farbharcopies für Drucker und Plotter sind nur einige der umfangreichen Beispiele, die von DATA BECKER Spezialisten für Sie erstellt wurden. Ein fantastisches Buch zu einem fantastischen Rechner!
ATARI ST Tips & Tricks, 256 Seiten, DM 49,-



Schlagen Sie dem Betriebssystem Ihres ATARI ST ein Schnippchen. Wie? Mit PEEKS & POKES natürlich! Dieses Buch erklärt Ihnen den Umgang damit. Mit vielen wichtigen POKES und ihren Anwendungsmöglichkeiten. Dabei wird der Aufbau Ihres ST's prima erklärt: Betriebssystem, Interpreter, Pointer und Stacks sind nur einige Stichworte dazu.
PEEKs & POKES zum ATARI ST, 194 Seiten, DM 29,-



Ein Buch für jeden, der unter GEM Programme erstellen will! Arbeiten mit der Maus, Icons, Virtual Device Interface, Application Environment System und Graphics Device Operating System. Ein besonderer Schwerpunkt liegt im Einbinden von GEM-Routinen in C und 68000-Assembler und der Programmierung in diesen Sprachen. GEM – das Betriebssystem der Zukunft!
Das große GEM-Buch zum ATARI ST, 459 Seiten, DM 49,-



Den ATARI ST voll ausnutzen können Sie nur in Maschinensprache! Zahlensysteme, Bitmanipulation, der 68000 im ATARI ST, Registerverwendung, Struktur des Befehlsatzes, Programmstrukturen, Rekursion, Stacks, Prozeduren, Grundlagen der Assemblerprogrammierung Schritt für Schritt, Verwendung von Systemroutinen und Tips zum Einbinden von Assembler-routinen in Hochsprachen. Eine hervorragend geschriebene Einführung!
ATARI ST Maschinensprache, 250 Seiten, DM 39,-



Kein 68000-Programmierer sollte auf dieses Handbuch verzichten. Sie finden detailliertes Sachwissen zur Technik und Programmierung: Entwicklung des 68000, Aufbau, Signal- und Busbeschreibung, Peripheriebausteine, Befehlssatz, Programmbeispiele, Vergleich mit anderen 16-Bit-Prozessoren u.v.m. Ein Buch für echte Computertrecks!
Das Prozessorbuch zum 68000, 516 Seiten, DM 59,-

Wichtig: Die DATA WELT 4/86 bringt den ST SOFTWARE-FÜHRER

Alle Programme zum ATARI ST – unter diesem Motto steht ein ausführlicher Sonderteil der DATA WELT 4/86. Über 20 Seiten randvoll mit Kurzttests, Tips & Tricks und Erfahrungsberichten. Außerdem wie in jeder DATA WELT viele weitere ST-Artikel und großer ST-Zubehörmarkt.

DATA WELT 4/86
ab 17. März am Kiosk

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

☐ per Nachnahme ☐ zzgl. DM 5,- Versandkosten
☐ Verrechnungsscheck liegt bei
Name und Adresse
bitte deutlich
schreiben

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (02 11) 31 00 10

EASY PAS

Das Utility zu CCD-Pascal Schnell & komfortabel

Leider läßt GEMDOS PASCAL im Bezug auf die Benutzerschnittstelle einiges zu wünschen übrig. Wer TURBO PASCAL kennt, weiß was ich meine. Der normale Ablauf der Programmerstellung läuft bei GEMDOS PASCAL folgendermaßen ab:

1. Editieren des Quellcodes mit EDIT.TTP
2. Compilieren (BATCH.TTP anklicken, Name des Batchfiles zum Compilieren und Programmname eingeben)
3. Vorgang 1. und 2. wiederholen, bis keine Fehlermeldung mehr erscheint
4. Linken (BATCH.TTP anklicken, Name des Batchfiles zum Linken und Programmname eingeben)
5. Das fertige Programm starten und hoffen, daß es läuft. Wenn nicht, alles von 1. bis 5. wiederholen.

Man kann sich das Leben auch etwas einfacher machen und ein Batchfile schreiben, das nach dem Compilieren direkt linkt. Das Problem dabei ist, wenn der Compiler einen Fehler findet, wird kein Objektfile angelegt und das Programm RELMOD.PRГ erstellt ein File, das den Platz, der noch auf der Diskette zur Verfügung steht, voll ausfüllt. Das unangenehmste dabei ist die Zeit, die dafür benötigt wird. Dieses Problem erübrigt sich für die glücklichen Besitzer des neuen Linkers 'FASTLINK'. Dieses etwa 3 K-Byte lange Programm ersetzt LINK68 und RELMOD und weiß auch, daß ein nichtvorhandenes Objektfile nicht in ein Programm umgewandelt werden kann. Nebenbei macht er, was die Geschwindigkeit angeht, seinem Namen alle Ehre.

Trotzdem ersetzt er keine Menüsteuerung und muß immer noch über den Batchprozessor aufgerufen werden.

Der CHAIN Befehl brachte mich auf die Idee, die Sache selbst in die Hand

zu nehmen. Das Ergebnis ist eine Benutzeroberfläche, deren Ähnlichkeit mit anderen nicht zufällig ist.

EASYPAS arbeitet mit dem Prinzip des Chaining. Wählt der Benutzer eine Funktion aus, die entweder Compiler, Linker oder Editor benötigt, wird das jeweilige Programm in den Speicher geladen, ausgeführt und dann wieder das Hauptmenü angezeigt. Der Vorteil besteht darin, daß die Programme immer den gesamten Speicher (bis auf 10 K-Byte die EASYPAS benötigt) zur Verfügung haben. Der Nachteil ist natürlich die Zeit die zum Laden der Programme erforderlich ist. Hier schafft eine RAMDISK Abhilfe. Die Programme werden einmal von der Diskette auf die Ramdisk kopiert und dann beim Aufruf nur noch im Speicher bewegt. Soviel zur Arbeitsweise des Programms.

Bei der Programmerstellung mit EASYPAS sieht eine Sitzung wie folgt aus:

1. EASYPAS starten
2. 'W' drücken und Name des Quellcodefiles angeben
3. 'E' drücken und Quelltext editieren
4. 'C' drücken, um zu compilieren
5. Punkt 2. bis 4. wiederholen bis keine Fehler mehr gemeldet werden
6. 'L' drücken, um zu linken
7. 'X' drücken, um das fertige Programm zu testen ggf. 3. bis 7. wiederholen.

Sollte man einmal aus Versehen nicht die Originaldiskette eingelegt haben und startet den Compiler, erscheint eine Copyrightmeldung und man muß neu booten. Nachdem mir das zum dritten Mal passiert war implementierte ich die Funktion 'org_disk', die nur dann TRUE liefert, wenn die Originaldisk im Laufwerk ist. Im Hauptprogramm wird bei org_disk = FALSE der Be-

nutzer aufgefordert die Originaldiskette einzulegen und eine Taste zu drücken.

Im Untermenü 'OPTIONS' kann der Benutzer die Compileroptionen DEBUG und LIST auswählen und angeben ob beim Compilieren die Compilerliste auf dem Drucker ausgegeben werden soll (funktioniert nicht bei Version 1.0 des PASCAL Compilers).

Es besteht weiterhin die Möglichkeit zu wählen, ob das Directory bei jedem Einsprung in das Hauptmenü angezeigt werden soll oder nicht.

Implementierungshinweise:

Um das Programm flexibel zu halten, wird beim Linken immer noch der Batchprozessor eingesetzt. Das File 'LINK.BAT' enthält den Aufruf des Linkers und ggf. den Aufruf von RELMOD. Da EASYPAS als TOS Anwendung läuft können Programme die als GEM Anwendungen arbeiten zwar compiliert und gelinkt werden, aber die Funktion Execute darf nicht benutzt werden.

Folgende Files werden von EASYPAS benötigt und müssen auf dem Laufwerk verfügbar sein von dem EASYPAS geladen wurde.

PASCAL.PRГ Der eigentliche Compiler
EDIT .TTP Der Editor
LINK .BAT Die Linkbefehle
BATCH .TTP Der Batchprozessor

Zusätzlich wird noch ein Linker benötigt, der entweder aus den Programmen LINK68 und RELMOD, oder nur aus FASTLINK besteht. Die Library-Files, die beim Linken verwendet werden, müssen natürlich auch verfügbar sein. Auf welchem Laufwerk der Linker und alle Files, die in LINK.BAT angesprochen werden, sind, bleibt dem Benutzer freigestellt.

Jürgen Leonhard

Listing of EASY.PAS

```

{$S10} { Maximal 10 KByte fuer Programm und Daten }
PROGRAM easy;

CONST edit    = 'EDIT.TTP ';
      compile = 'PASCAL.PRG ';
      link    = 'BATCH.TTP LINK ';
      ptr     = '>PRN:';

TYPE name     = STRING[80];

VAR  workfile, prgname, parastr, chstr : name;
      response                : CHAR;
      poption, debug, list, adir  : BOOLEAN;

PROCEDURE clrscr;

BEGIN
    WRITE(CHR(27),'E'); { Loescht den Bildschirm (siehe TURBO) }
END; { clrscr }

PROCEDURE gotoxy(x,y : INTEGER);

BEGIN
    WRITE(CHR(27),'Y',CHR(32+y),CHR(32+x)); { Corsor Positionieren }
END; { gotoxy }

PROCEDURE terminate;
VAR key : CHAR;

BEGIN
    gotoxy (1,24);
    WRITE ('Press any key...');
    READ  (key);
END; { terminate }

PROCEDURE dir;
TYPE buffer = PACKED ARRAY [1..45] OF CHAR;
      ch_6   = PACKED ARRAY [1..6]  OF CHAR;
      word   = PACKED ARRAY [1..2]  OF CHAR;

VAR  i,j,k : INTEGER;
      buff44 : buffer;
      fname  : ch_6;
      attrib : word;

PROCEDURE setdta (VAR buf : buffer);GEMDOS($1a);
FUNCTION sfirst (VAR filnam : ch_6; attr : word):INTEGER;GEMDOS($4e);
FUNCTION snext : INTEGER ;GEMDOS($4f);

BEGIN
    gotoxy(8,23);
    WRITE(' ');
    setdta(buff44);
    fname := '*.PAS '; { Hier kann natuerlich auch *.* stehen }
    fname[6] := chr(0); { auf jeden Fall aber $0 am Ende !!!!! }
    attrib[1] := chr(0);
    attrib[2] := chr(0);
    FOR i := 31 TO 44 DO buff44[i] := ' '; { Buffer loeschen }
    j := sfirst(fname,attrib);
    gotoxy (8,14);
    FOR i := 31 TO 43 DO WRITE(buff44[i]);
    WRITE(' ');
    k := 2;
    REPEAT
        FOR i := 31 TO 44 DO buff44[i] := ' ';
        j := snext;
        IF j=0 THEN
            FOR i := 31 TO 43 DO WRITE(buff44[i]);
            k := k + 1;
            IF k > 4 THEN
                BEGIN

```

Listing

```

        k := 1;
        WRITELN;
        WRITE(' ');
        END
    ELSE
        WRITE(' ');
    until j <> 0;
end; { dir }

FUNCTION org_disk:BOOLEAN;

TYPE long      = PACKED ARRAY [1..4] OF CHAR;
    block      = PACKED ARRAY [1..512] OF CHAR;
    buff       = PACKED ARRAY [1..512] OF CHAR;

VAR dummy      : long;
    a,b,c,d,e,res : INTEGER;
    daten       : block;

FUNCTION Floprd (VAR buffer : block; dumm : long; drive,sector,track,side,
    count : integer):INTEGER;xbios(8);

begin
a := 80; { Track 80 }
b := 1;  { Sector 1 }
c := 0;  { Drive A: }
d := 1;  { 1 Sector }
e := 0;  { Side 0 }
res := Floprd (daten,dummy,c,b,a,e,d);
IF res = 0 THEN
    org_disk := TRUE { Original war in Drive A: }
ELSE
    BEGIN
        c := 1; { testen ob die Original Disk in Drive B ist }
        res := Floprd(daten,dummy,c,b,a,e,d);
        IF res = 0 THEN org_disk := TRUE { Original war in B: }
            ELSE org_disk := FALSE; { Kein Original }
        END;
    END;{ org_disk }

PROCEDURE set_options;
VAR opt : CHAR;

BEGIN
REPEAT
    REPEAT
        clrscr;
        WRITELN('Optionflags setzen');
        WRITE (CHR(10),' Printer = ');
        IF poption THEN WRITE(' ON')
            ELSE WRITE('OFF');
        WRITE (' DEBUG = ');
        IF debug THEN WRITE(' ON')
            ELSE WRITE('OFF');
        WRITE (' LIST = ');
        IF list THEN WRITE(' ON')
            ELSE WRITE('OFF');
        gotoxy(8,6 ); WRITE('[P]rinter [L]ist ');
        gotoxy(8,8 ); WRITE('[D]ebug [Q]uit');

        gotoxy(8,23); WRITE('COMMAND >');
        READ(opt);
        IF ORD(opt) > 96 then opt := CHR(ORD(opt)-32);
    UNTIL opt IN ['P','L','D','Q'];
    IF opt <> 'Q' THEN
        CASE opt OF
            'P' : poption := NOT (poption);
            'D' : debug := NOT (debug);
            'L' : list := NOT (list);
        END; { CASE }
    END;

```

```

UNTIL opt = 'Q';
END; { set_options }

PROCEDURE linker;
BEGIN
  clrscr;

  chstr := CONCAT (link,workfile);
  chstr [LENGTH(chstr)+1] :=CHR(0);
  CHAIN(chstr);
  terminate;
END;{ linker }

PROCEDURE klein_gross (VAR zeile : name);
VAR i : INTEGER;

BEGIN
  FOR i := 1 TO LENGTH (zeile) DO
    IF ((ORD(zeile[i]) > 96) AND (ORD(zeile[i]) < 123))
    THEN zeile[i] := CHR(ORD(zeile[i])-32);
  zeile [LENGTH(zeile)+1] := CHR (0);
END; { klein_gross }

PROCEDURE delext (VAR fname : name);
VAR p, l : INTEGER;

BEGIN
  { Loeschen der Fileextension falls vorhanden }
  p := POS ('.',fname);
  IF p > 0 THEN
    BEGIN
      l := LENGTH(fname) - p + 1;
      DELETE (fname,p,l);
    END;
  END; { delext }

PROCEDURE hauptmenu (VAR response : CHAR);
BEGIN
  REPEAT
    clrscr;
    WRITE (' EASYPAS U 1.7 ',CHR(189),' by J. J. Leonhard 1986');
    gotoxy(8,3 ); WRITE('[W]orkfile [E]dit ');
    gotoxy(8,5 ); WRITE('[C]ompile [Q]uit');
    gotoxy(8,7 ); WRITE('[L]ink [D]irectory');
    gotoxy(8,9 ); WRITE('[X]ecute [O]ptions ');
    gotoxy(8,12); WRITE('Workfile = ',workfile,'.PAS');
    IF adir THEN dir;
    gotoxy(8,23); WRITE('COMMAND >');
    READ(response);
    IF response = CHR(27) THEN response := 'q';
    IF ORD(response) > 96 then response := CHR(ORD(response)-32);
  UNTIL response IN ['W','E','C','L','Q','X','O','D'];
END; { hauptmenu }

BEGIN { main }
workfile := 'NONAME'; { Initialisierung }
poption := FALSE;
list := TRUE; { der Flags mit }

debug := FALSE;
adir := TRUE; { Default-Werten }
repeat
  hauptmenu(response);
  CASE response OF
    'E' : BEGIN
      chstr := CONCAT(edit,workfile,'.PAS');
      chstr [LENGTH(chstr)+1] :=CHR(0);
      CHAIN (chstr);
      END;

```

Listing

```

'C' : BEGIN
  IF NOT org_disk
  THEN
    BEGIN
      clrscr;
      WRITE(CHR(7),'Bitte Original PASCAL Diskette in ');
      WRITELN('Laufwerk A: oder B: einlegen');
      terminate;
    END;
  IF list
  THEN chstr := CONCAT(compile,workfile,' /LIST ');
  ELSE chstr := CONCAT(compile,workfile);
  IF debug
  THEN chstr := CONCAT(chstr,'/DEBUG ');
  IF poption THEN chstr := CONCAT(chstr,ptr);
  chstr [LENGTH(chstr)+1] :=CHR(0);
  clrscr;
  WRITELN('SOURCE-FILE = ',workfile,'.PAS      CONTROL & S to stop');
  WRITELN;
  CHAIN(chstr);
  terminate;
  END;

'W' : BEGIN
  REPEAT
    gotoxy (1,12);
    WRITE (CHR(27),'K');
    gotoxy (8,12);
    WRITE ('New Workfile = ');
    READLN (workfile);
    deletx (workfile);
    klein_gross(workfile);
    IF LENGTH(workfile) > 8
    THEN
      BEGIN
        gotoxy(8,14);
        WRITE ('Bitte maximal 8 Buchstaben eingeben !!!');
        WRITE (CHR(7),CHR(7),CHR(7));

        gotoxy(1,12);
        WRITE (CHR(27),'K');
        END;
      UNTIL LENGTH(workfile) <= 8;
    END;

  'L' : BEGIN
    linker;
    END;

  'X' : BEGIN
    clrscr;
    chstr := CONCAT (workfile,'.PRG');
    chstr [LENGTH(chstr)+1] :=CHR(0);
    CHAIN(chstr);
    terminate;
    END;

  'D' : adir := NOT (adir);

  'O' : set_options;

  END; ( CASE )
  UNTIL response = 'Q';
  END. ( main )

```

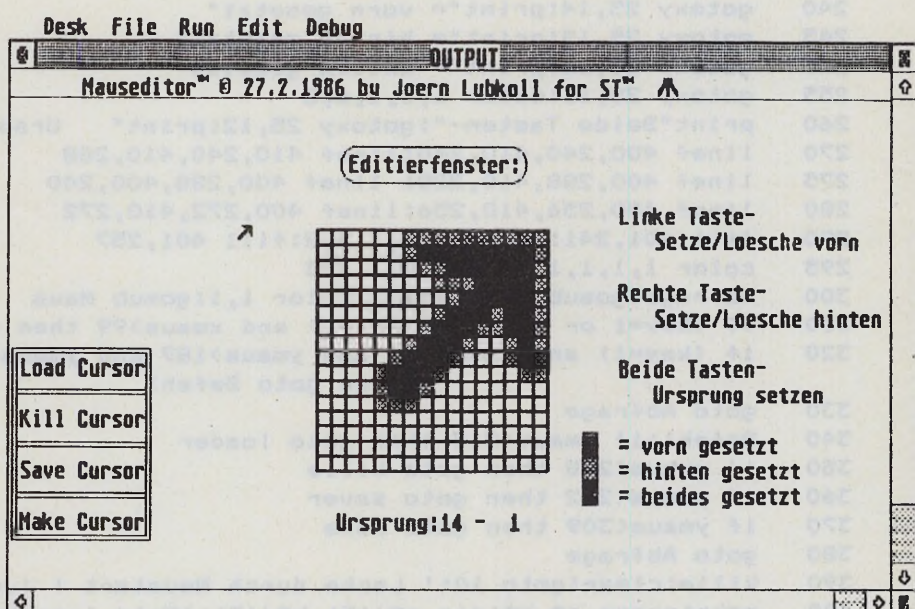
Mauseditor

- Editieren mit der Maus
- Speichermöglichkeit der Daten
- Erzeugen eines Basic-Laders

Dieses Programm zeigt einige praktische Anwendungen der GEM-Routinen in Basic. Es dient zum Editieren des Mauszeigers auf dem Bildschirm. Dabei sind alle Funktionen des Programms mit der Maus ansteuerbar. Das Programm kann das fertig erstellte Maussymbol als Datei abspeichern, oder ein direkt lauffähiges Basic Unterprogramm erzeugen, welches die Datas und die dazugehörige Definitionsroutine (siehe ST 3/86) enthält.

Das Programm wird folgendermaßen bedient:

Die einzelnen Punkte innerhalb des 16 × 16 Rasters werden per Maustaste gesetzt oder gelöscht. Dabei setzt die linke Taste die Punkte der eigentlichen Mausform. Die rechte Taste setzt das Hintergrundmuster, welches für die Überlagerung des Zeigers mit dem Hintergrund verantwortlich ist. Dieses sollte immer etwas größer sein als das Maussymbol, da der Mauszeiger bei dunklem Hintergrund verschwindet.



Wenn man beide Maustasten hintereinander drückt, so setzt das Programm sowohl Vorder- als auch Hintergrundbit. Die Position innerhalb des Maussym-

bols, die den Koordinaten der Maus entsprechen, werden durch gleichzeitigen Druck beider Tasten an der gewünschten Stelle festgelegt und abgespeichert.

List of \C.BAS

```

10 ' Editor fuer Mausbilder... Benutzt Routine aus ST 3.86
20 ' (C) 1986 by Joern Lubkoll...
30 '-----
40 ' Initialisiere Programm und Variablen
50 gosub hidemaus:option base 0:dim a*(15),b*(15)
55 for i=0 to 15:a*(i)=""
60 b*(i)="" :next i:on ERROR goto fehler:dim a(15,15),b(15,15)
70 for a=0 to 15:for b=0 to 15:a(a,b)=0:b(a,b)=0:next b,a
80 fullw 2:clearw 2:color 1,1,1,1,1:mausr=0
85 gosub mausform
90 ' Erstellen der Bildschirmmaske
100 gotoxy 3,0:print"Mauseditor"chr$(191);" ";
105 print chr$(189)" 27.2.1986 by ";
110 print"Joern Lubkoll for ST"chr$(191);" ";chr$(14);chr$(15)
120 linef 0,18,616,18:gotoxy 13,3:print" Editierfenster:"
130 xpos1=29*8:ypos1=3*16+2:xpos2=45*8:ypos2=4*16+10:fuell=0
140 gosub rechteckrund:gotoxy 0,11:" Load Cursor "
145 "??" Kill Cursor "
150 print:print" Save Cursor ":"??" Make Cursor "

```

```

160 linef 2,185,99,185:linef 99,185,99,312
165 linef 99,312,2,312:linef 2,312,2,185
170 linef 5,188,96,188:linef 96,188,96,309
175 linef 96,309,5,309:linef 5,309,5,188
180 linef 5,212,96,212:linef 96,212,5,212
185 linef 96,247,5,247:linef 96,250,5,250
190 linef 5,281,96,281:linef 96,284,5,284
200 for x=215 to 375 step 10:linef x,105,x,265
205 linef x+1,105,x+1,265:next x
210 for y=105 to 265 step 10:linef 215,y,375,y
215 linef 215,y+1,375,y+1:next y
220 gotoxy 25,5:print"Linke Taste-"
225 gotoxy 26,6:print" Setze/Loesche vorn"
230 gotoxy 25,8:print"Rechte Taste-"
235 gotoxy 26,9:print" Setze/Loesche hinten"
240 gotoxy 25,14:print"= vorn gesetzt"
245 gotoxy 25,15:print"= hinten gesetzt"
250 gotoxy 25,16:print"= beides gesetzt"
255 gotoxy 25,11:color 1,1,1,6,2
260 print"Beide Tasten-":gotoxy 25,12:print" Ursprung setzen"
270 linef 400,240,410,240:linef 410,240,410,288
275 linef 400,288,410,288: linef 400,288,400,240
280 linef 400,256,410,256:linef 400,272,410,272
290 fill 401,241:color 1,1,1,3,2:fill 401,257
295 color 1,1,1,1,1:fill 401,273
300 Abfrage:gosub showmaus: color 1,1:gosub Maus
310 if (key=1 or key=2 or key=3) and xmaus>99 then goto modify
320 if (key=1) and xmaus<96 and ymaus>187 and ymaus<310
    then goto Befehl
330 goto Abfrage
340 Befehl:if ymaus<213 then goto loader
350 if ymaus<248 then goto kille
360 if ymaus<282 then goto saver
370 if ymaus<309 then goto make
380 goto Abfrage
390 kille:clear:goto 10:' Lsche durch Neustart ! (vorsicht)
400 make:erase a$:b$:dim a$(15),b$(15):'Feld loeschen + setzen
410 for b=0 to 15:a$(b)="" :b$(b)="" :for a=0 to 15
420 if a(a,b) =0 then a$(b)=a$(b)+"-" else a$(b)=a$(b)+"*"
430 if b(a,b) =0 then b$(b)=b$(b)+"-" else b$(b)=b$(b)+"*"
440 next a,b:gosub Editor:goto Abfrage
450 goto Abfrage
460 modify: xp=(xmaus-215)/10:yp=(ymaus-105)/10
465 if xp<0 or xp>=16 then goto Abfrage
470 if yp < 0 or yp >= 16 then Abfrage ' Cursor im Feld ?
480 xp=int(xp):yp=int(yp)
490 if key=1 then gosub vorn:' Invertiert einen Punkt vorn.
500 if key=2 then gosub hinten:' " " " hinten."
510 if key=3 then my=xp:mx=yp:gotoxy 13,18:
    gosub ursprung: goto Abfrage 'Mitte
520 gosub Darstellung:goto Abfrage
530 Darstellung: a=xp:b=yp:gosub hidemaus:color 0,0,0,0,0
540 Darst2: for i=2 to 9
545 linef 217+a*10,105+b*10+i,217+a*10+7,105+b*10+i:next i
550 if a(a,b)=1 and b(a,b)=0 then color 1,1,1,6,2:
    fill a*10+217,b*10+107
560 if a(a,b)=1 and b(a,b)=1 then color 1,1,1,1,1:
    fill a*10+217,b*10+107
570 if a(a,b)=0 and b(a,b)=1 then color 1,1,1,3,2:
    fill a*10+217,b*10+107
580 gosub showmaus
590 return
600 vorn: a(xp,yp)=a(xp,yp) xor 1:return
610 hinten: b(xp,yp)=b(xp,yp) xor 1:return
620 mitte:mx=xp:my=yp:goto Abfrage
630 Ursprung:gotoxy 13,17:print spc(10):gotoxy 13,17:?" Ursprung:"

```

```

640 print USING"##"my;
650 print" - ":print using"##"mx;color 1,1,1,1
660 return
670 saver: gotoxy 13,18:print"_____."
675 gotoxy 13,18:line input na#
680 open "0",#1,na#
690 for a=0 to 15:for b=0 to 15:x#=mki*(a(a,b)):print #1,x#
700 x#=mki*(b(a,b)):print #1,x#:next b,a:print #1,mx,my
710 aa: close 1:gotoxy 13,18:print spc(30):goto abfrage
720 loader:gotoxy 13,18:print"_____."
725 gotoxy 13,18:line input na#
730 if len(na#)<13 then open "I",#1,na#:x#="" else loader
740 for a=0 to 15:for b=0 to 15
745 input#1,x#:a(a,b)=cvi(x#):input#1,x#
750 b(a,b)=cvi(x#):color 0,0,0,0,0
755 gosub darst2:next b,a:input#1,mx,my
760 color 1,1,1,1,1:gosub Ursprung:goto aa
770 ' Fehlerbehandlungung, pipstion und lischen des Namens.
780 fehler: out 2,7:close 1:out 2,7:gotoxy 10,18
785 print spc(30):color 1,1:goto Abfrage
790 ' Verschiedene Grafikroutinen aus ST 2,3. 86
795 ' -----
800 MAUS: POKE CONTRL,124:VDISYS
810 XMAUS=PEEK(PTSOUT)-1:YMAUS=PEEK(PTSOUT+2)-38
815 KEY=PEEK(INTOUT):RETURN
820 SHOWMAUS: POKE CONTRL,122:POKE INTIN,0:VDISYS
830 RETURN
840 HIDEAUS: POKE CONTRL,123:POKE INTIN,0:VDISYS
850 RETURN
860 EDITOR:gotoxy 13,18:Line Input "Programm erstellen ?";a#
862 gotoxy 13,18:print spc(30)
865 if a#="j" then flag=1:gotoxy 13,18:
      line Input"Name:";na# else bb
867 open "0",#1,na#:li=32000:gotoxy 13,18
868 print spc(30):gosub Routine
869 bb: for a=0 to 15
870 MUSTERVOR=0:MUSTERHINTER=0
880 FOR B=15 TO 0 STEP-1
890 IF MID*(A*(A),B+1,1)="*" THEN BIT=1 ELSE BIT=0
900 MUSTERVOR=MUSTERVOR+2^(15-B)*BIT
910 IF MID*(B*(A),B+1,1)="*" THEN BIT=1 ELSE BIT=0
920 MUSTERHINTER=MUSTERHINTER+2^(15-B)*BIT
930 NEXT
940 POKE INTIN+A*2+42,MUSTERVOR
950 POKE INTIN+A*2+10,MUSTERHINTER
953 li=li+1
955 If flag=1 then gosub output
960 NEXT
965 if flag=1 then print#1,"32100 data "mx","my
970 close 1:GOTO MAUSNEU
980 MAUSDATA: FOR A=0 TO 15:READ MUSTERVOR,MUSTERHINTER
990 POKE INTIN+A*2+42,MUSTERVOR:POKE INTIN+A*2+10,MUSTERVOR:NEXT I
1000 MAUSNEU: POKE CONTRL,111:POKE CONTRL+6,37:POKE INTIN,my
1010 poke INTIN+2,mx: POKE INTIN+4,1:POKE INTIN+6,0:POKE INTIN+8,1
1015 VDISYS
1020 OUT 2,7:RETURN
1030 MAUSFORM: IF MAUSNR>7 THEN RETURN
1040 ADD#=#0B:GINTIN=PEEK(ADD#+8):ADDRIN=PEEK(ADD#+16)
1050 POKE GINTIN,MAUSNR: POKE ADDRIN,0:GEMSYS(78)
1060 RETURN
1070 RECHTECKRUND: POKE CONTRL,11:POKE CONTRL+2,2:POKE CONTRL+6,0
1080 IF FUELL=1 THEN POKE CONTRL+10,9 ELSE POKE CONTRL+10,8
1090 POKE PTSIN,XPOS1+1:POKE PTSIN+2,YPOS1+38:POKE PTSIN+4,XPOS2+1
1100 POKE PTSIN+6,YPOS2+38:VDISYS
1110 RETURN

```

```

1120 ' Ende des Programmes
1130 output: print #1, str$(fix(li));" data ";mustervor;",";musterhinter
1140 return
1150 Routine: print #1,"31900 MAUSDATA: restore 32001:for a=0 to 15: ";
1160 print#1,"read mustervor,musterhinter"
1170 print#1,"31910 poke intin+a#2+42,mustervor"
1175 print#1,"31915 poke intin+a#2+10,musterhinter"
1180 print#1,"31920 next:poke contrl,111:poke contrl+6,37:read mx,my "
1190 print#1,"31930 poke intin,mx:poke intin+2,my"
1195 print#1,"31935 poke intin+4,0:poke intin+6,0"
1200 print#1,"31940 poke intin+8,1:vdisys"
1210 print#1,"31950 return"
1220 return

```

```

31900 MAUSDATA: restore 32001:for a=0 to 15:read mustervor,musterhinter
31910 poke intin+a#2+42,mustervor
31915 poke intin+a#2+10,musterhinter
31920 next:poke contrl,111:poke contrl+6,37
31925 read mx,my
31930 poke intin,mx:poke intin+2,my
31935 poke intin+4,0:poke intin+6,0
31940 poke intin+8,1:vdisys
31945 return
31950 '-----
32001 data 0 , 1023
32002 data 510 , 1023
32003 data 126 , 511
32004 data 30 , 255
32005 data 62 , 127
32006 data 118 , 255
32007 data 230 , 511
32008 data 450 , 999
32009 data 898 , 1991
32010 data 1792 , 3971
32011 data 1536 , 3840
32012 data 0 , 3584
32013 data 0 , 0
32014 data 0 , 0
32015 data 0 , 0
32016 data 0 , 0
32100 data 1 , 14

```

Der erzeugte Basiclader

Der Geheimtip fuer den ST.....copyStar V1.0

- * Kopiert geschuetzte Programme ! (wirklich alle !!)
- * Kopiert 'normale' Disketten **wahnsinnig** schnell (in 34 Sekunden !)
- * Konvertiert Disketten in ein schnelleres Spezial-Format.
Das bringt doppelte Geschwindigkeit bei allen Floppy-Arbeiten ohne zusaetzliche Hardware.
- * **Update Service.** Bei Erscheinen einer neuen copyStar-Version werden Sie automatisch informiert
- * Benutzerfuehrung wahlweise in deutsch, englisch, spanisch oder franzoesisch.
- * Ausfuehrliches Handbuch - natuerlich in deutscher Sprache.
- * All das zum Superpreis von DM 169,-
- * **WO ? NATUERLICH BEI IHREM ATARI-HAENDLER!**
oder direkt bei



DATALOGIC

Ihr ATARI-Spezialist in Hannover
Calenbergerstr. 26 · 3000 Hannover 1

Tel. 05 11 - 32 64 89

C Esperanto der Computersprache ?

Immer mehr ernsthafte Programmierer wählen die Sprache „C“ zur Entwicklung von Software. Der Grund liegt mit Sicherheit in den starken Leistungsmerkmalen dieser Sprache. Gerade für ST-Besitzer ist diese Sprache von großer Bedeutung, weil das Betriebssystem zu einem großen Teil in „C“ geschrieben ist.

Wir möchten mit diesem Artikel einen Überblick über „C“ geben. Was macht eine Programmiersprache beliebter als die andere? Welches sind die Kriterien, unter denen die Programmierer eine Sprache auswählen? Um eine einigermaßen objektive Antwort zu erhalten, sollte man möglichst vermeiden, engagierte Programmierer einer bestimmten Sprache zu fragen. Mit Sicherheit sind fast alle Probleme der Informatik auf verschiedene Arten zu lösen und somit auch durch verschiedene Sprachen, ohne Rücksicht auf Faktoren wie Zeit, Aufwand, Länge usw. Allerdings ist bei der Entwicklung von professioneller Software nicht nur die Endlösung von Bedeutung, sondern auch Form und Aufwand spielen eine wichtige Rolle.

Da man in der Computerwelt mit der Tatsache leben muß, daß es verschiedene Computer-Typen gibt, die sich nicht vertragen, ist es von großer Bedeutung, eine gemeinsame Sprache zu finden, die eine Art Brücke bildet.

C und ein wenig Geschichte für die Historiker

Die Sprache „C“ wurde 1972 von Denis Ritchie am Bell Laboratories entwickelt. Grund dafür war das bis zu diesem Moment in Assembler PDP 11 geschriebene Betriebssystem UNIX umzuschreiben, um es auch für andere Maschinen lauffähig zu machen. Ken Thompson, der Schöpfer dieses Betriebssystems, entwarf zu diesem

Zweck eine Sprache mit dem Namen „B“, welches die Basis von der heutigen „C“ war. So wurde im Jahre 1974 ein in „C“ geschriebenes UNIX in eine PDP 11 Anlage installiert. Ohne Zweifel, und das ist ungemein wichtig, sind die Entwicklung von „C“ und die Entwicklung des Betriebssystems UNIX untrennbar.

Später, im Jahre 1978, schreibt Denis Ritchie zusammen mit Brian Kernighan ein Buch über „C“, das als Standardwerk zu der Definition dieser Sprache gilt. Dieser Standard (an den sich die verschiedenen Compiler-Entwickler im wesentlichen halten) hat dazu geführt, daß die verschiedenen Versionen einheitlicher bleiben als bei anderen Sprachen, wie z. B. bei PASCAL.

Die Philosophie einer „universalen“ Sprache

Bei der Entwicklung von „C“ wurde besonders darauf geachtet, eine Sprache zu schaffen, die so unabhängig wie möglich von jeder Hardware-Architektur bleibt. So gehören zur tatsächlichen Sprache keine Anweisungen für Ein- und Ausgabe, Dateizugriff und Speicherverwaltung. Alle diese Funktionen werden in externe, sogenannte C-Libraries, implementiert. Damit erreicht man eine fast absolute Portabilität von Programmen aus verschiedenen Rechnern. Die eigentlichen Anweisungen, Operanden und Operatoren beschränken sich bei „C“ auf ein paar Dutzend. Damit vereinfacht sich das Erlernen der Sprache und der Compiler selbst kann sehr kompakt sein. Eine Darstellung von „C“-Anweisungen sehen Sie auf Bild 1. Bei der C-Biblio-

SCHLÜSSELWORTE

auto	double	if	static
break	else	int	struct
case	entry	long	switch
char	extern	register	typedef
continue	float	return	union
do	goto	short	while

OPERATOREN

+	Addieren	-	
★	multiplizieren	/	dividieren
%	modulo	>	größer als
<	kleiner als	>=	größer als oder gleich
<=	kleiner als oder gleich	=	gleich
!=	ungleich	()	funktions-klammer
()	array-klammern	->	strukturelement
.	strukturname	!	negation
~	komplement	++	inkrement
--	dekrement	&	unt (bitweise)
=	xor		oder (bitweise)
&&	logisches und		logisches oder
?:	bedingter operator	=	zuweisung
+=	+zuweisung	-=	-zuweisung
<<	shiftlinks	>>	shiftrechts

Bild 1: Schlüsselworte und Operatoren

thek verhält sich dies etwas anders.

Einerseits tendiert man dazu, zu jedem Compiler eine Standard-Library mitzuliefern. Andererseits berücksichtigt man die einzelnen Eigenschaften von eigenen Betriebssystemen und versucht, den Compiler so auszurüsten, daß das Betriebssystem optimal ausgenutzt werden kann. Gerade diese beliebige Anzahl von Routinen macht aus „C“ eine besonders leistungsfähige Sprache.

Bild 2 zeigt einige Standardroutinen.

Bei den ST-Rechnern gibt es einige Compiler, die schon eine Sammlung von Routinen anbieten, und die die Nutzung von fertigen Prozeduren, bei GEM wie auch bei TOS, ermöglichen. Bild 3 veranschaulicht einige dieser Spezial-Routinen.

Die C-Programmstruktur

Ein C-Programm besteht in der Regel aus einer Sammlung von Funktionen, die voneinander unabhängig sind. Eine dieser Funktionen muß den Namen „Main“ tragen, die andere kann jeden beliebigen Namen tragen. Die „main ()“-Funktion signalisiert die Stelle, an der ein Programm beginnt und darf logischerweise innerhalb eines Programms nur einmal erscheinen. Die Funktionen werden durch zwei runde Klammern angedeutet. So ist `plot ()` nicht gleich `plot`. Die Klammern dienen außerdem dazu, die formalen Parameter zu umfassen. Wenn keine Parameter existieren, dürfen die Klammern leer bleiben.

Jede Funktion kann sowohl andere Funktionen wie auch sich selbst (Rekursion) aufrufen.

Sehen wir uns ein Beispiel eines kleinen C Programms an:

```
main ()
{
    printf ("ST-COMPUTER MAGAZIN\n");
}
```

Das Programm enthält nur die Funktion `main ()`, die keinen Parameter besitzt, und hat die Ausgabe „ST-COMPUTER MAGAZIN“ auf den Bildschirm zu bringen. Die Anweisung „`printf`“ gehört normalerweise zu der Standard E/A-Funktion; sie ist eine der umfangreichsten. Solch ein Pro-

STANDARD BIBLIOTHEKSFUNKTIONEN

<code>printf</code>	<code>puts</code>	<code>scanf</code>
<code>rewind</code>	<code>getchar</code>	<code>fread</code>
<code>fflush</code>	<code>atoi</code>	<code>abs</code>
<code>gamma</code>	<code>floor</code>	<code>ceil</code>
<code>sqrt</code>	<code>frexp</code>	<code>modf</code>
<code>hcreate</code>	<code>sleep</code>	<code>system</code>

Bild 2: Einige Standardroutinen

Nicht Standard Bibliotheksfunktionen

<code>Cconin</code>	<code>Pterm</code>	<code>Dosound</code>
<code>Fdup</code>	<code>Getshift</code>	<code>Initmous</code>
<code>Setscreen</code>	<code>Vslide</code>	<code>Hslide</code>
<code>Upparrow</code>	<code>Dnarrow</code>	<code>Root</code>
<code>Small</code>	<code>Tochexit</code>	<code>Edstart</code>

Bild 3: Einige Nicht Standard Routinen

gramm wird textmäßig nur ein paar Bytes groß sein und nach der Compilierung einige Kbyte.

Wieso dieses immense Wachstum?

Es liegt an den verschiedenen Routinen, die zu „`printf`“ gehören und die in das eigentliche Programm eingebunden werden müssen. Eine Funktion beginnt mit der geschweiften Klammer und endet, wo sie geschlossen wird.

So braucht in der Regel ein in „C“ geschriebenes Programm keine Return-Anweisung (sie ist aber trotzdem vorhanden), um auf eine Funktion zurückzukehren.

Quo vadis, C?

Ein Programm läuft in der Regel nicht geradlinig. Es werden bestimmte Entscheidungen getroffen, und je nach Ergebnis werden dieser oder jener Zweig verfolgt. Dazu besitzt „C“ gewichtige Anweisungen, mit denen der Fluß eines Programms gesteuert werden kann. Die typischen Konstruktionen, die schon durch andere strukturierte Programmiersprachen (z. B. PASCAL) bekannt sind, stehen auch bei „C“ zur Verfügung. Eine genaue Beschreibung aller Programmsteuerungen würde den

Rahmen dieses Artikels sprengen, deswegen beschränken wir uns auf einige wenige davon.

Die „if“-Anweisung wird am häufigsten benutzt, und trotz ihrer Knappheit ist sie sehr effizient.

Eine typische „if“-Anweisung sieht folgendermaßen aus:

if (logischer Ausdruck) Anweisung

Der logische Ausdruck wird auf „wahr“ geprüft. Ist das der Fall, wird die Anweisung ausgeführt. Ist es nicht der Fall, wird die Anweisung übersprungen.

Nehmen wir ein Beispiel:

```
if (zw == 1) pu++;
printf("falsch\n");
```

Zuerst wird geprüft, ob „zw“ gleich „1“ ist. Falls das stimmt, wird „pu“ inkrementiert. Wenn „zw“ ungleich „1“ ist, wird „`printf`“ ausgeführt.

Unser Beispiel zeigt die einfachste Form der Benutzung der „if“-Anweisung. Die Möglichkeiten sind jedoch größer. Wenn wir nach dem abgefragten logischen Ausdruck eine Reihe von Anweisungen mit geschweiften Klammern umschließen, können wir

ein Bündel von Anweisungen auf einen Schlag ausführen.

Betrachten wir das nächste Beispiel.

```
if (x > y) {
    pu ++;
    z = 2;
    t = z + x;
```

```
h = t + 3;
```

Wenn $x > y$ wahr ist, werden alle Anweisungen, die in der Klammer stehen, ausgeführt. Ist dies nicht der Fall, wird weiter mit $h = t + 3$ fortgefahren. Die „if“-Struktur hat eine Variante, die noch vielseitiger ist als die von uns oben erwähnte. „if“ kann in Verbindung mit „else“ gebracht werden, und die gesamte Struktur sieht folgendermaßen aus

```
if (logischer Ausdruck) Anweisung 1
else Anweisung 2
```

Wenn der logische Ausdruck wahr ist, wird die Anweisung 1 ausgeführt, wenn nicht, dann Anweisung 2.

Wie schon erwähnt, besitzt „C“ noch einige Anweisungen zur bedingten Ausführung von Programmen, sowie zur Programmierung von Schleifen.

Aus Bild 4 können Sie einige dieser Strukturen entnehmen.

Fast so schnell wie Assembler

Man behauptet, daß in „C“ geschriebene Programme fast genauso schnell wie in „Assembler“ sind, mit der zusätzlichen Möglichkeit alle Elemente strukturierter Programme zur Verfügung zu haben. Zur Entstehung solcher Gerüchte trägt sogar der Schöpfer dieser Sprache bei. In seinem Buch („The C programming language“, geschrieben zusammen mit Kernigham) beschreibt er, daß diese Sprache die Möglichkeiten heutiger Computer widerspiegelt, und daß es keinen Grund zur Programmierung in Maschinsprache mehr gebe. Natürlich reicht diese Aussage nicht, um überhaupt anzunehmen, daß „C“ doch so schnell wie „Assembler“ sein könnte. Es gibt aber Elemente, die diese Behauptung bekräftigen. Die Datentypen von „C“ sind sehr „maschinenherkömmlich“ (Integer, Longs, Char, etc.). Prozesso-

KONTROLLSTRUKTUREN BEI 'C'

```
if (Ausdruck)
    Anweisung 1
else Anweisung 2

while (Ausdruck)
    Anweisung

for (Ausdruck 1; Ausdruck 2; Ausdruck 3)
    Anweisung

do
    Anweisung
while (Ausdruck)

goto
    Label

switch (Ausdruck)
    case .....
    case.....
```

Bild 4: Kontrollstrukturen

renoperationen lassen sich durch „C“-Operatoren sehr leicht nachbilden. Durch die Bildung von Zeigern lassen sich Adressen direkt ansprechen bzw. Inhalte manipulieren.

Es wäre aber trotzdem ein Unsinn, zu glauben, daß ein „C“-Programm genauso schnell sei wie ein gutes, in „Assembler“ geschriebenes Programm.

Eine Alternative, sich „Assembler“ zu nähern, bietet aber keine andere hohe Sprache.

Fazit

„C“ ist eine gewichtige Programmiersprache, die man lernen sollte. Für Personal-Computer, unter die auch die ST-Reihe von Atari fällt, sind schon mehrere Compiler auf dem Markt vorhanden. Reichlich Literatur hilft zum Erlernen dieser Sprache, die man aber nicht nach einem Wochenende beherrschen kann. Vor allem ST-Besitzer sollten diese Sprache erlernen, denn ein Hauch von „C“ steckt ganz tief in der Maschine.

Zum Teil vorhandene Literatur für 'C'

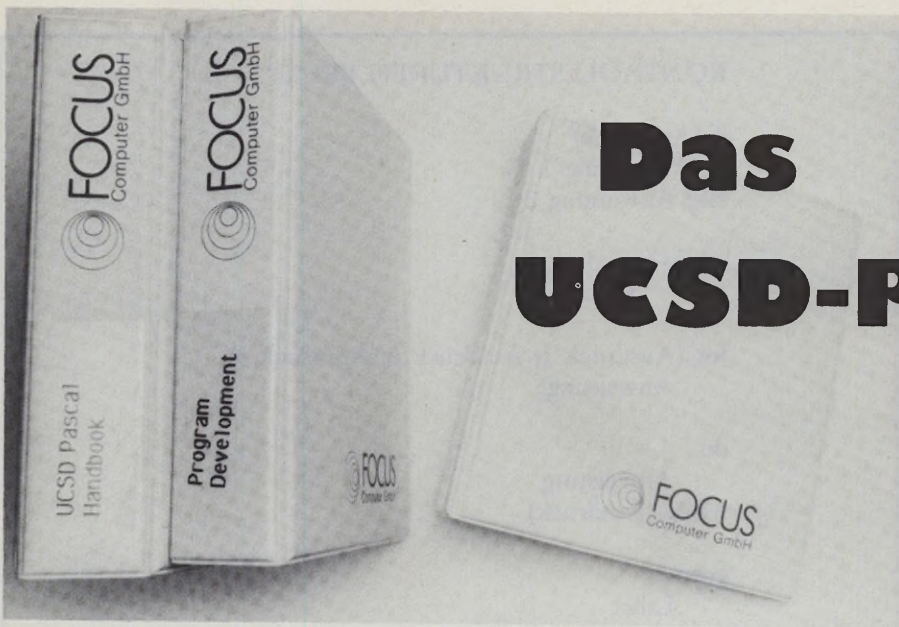
Programmieren in C
Kernighan und Ritchie
Hanser Verlag
DM 48,00

Die C-Programmbibliothek
Purdum
Markt & Technik Verlag
DM 60,00

Erfolgreich Programmieren mit C
Illik
Sybex Verlag
DM 58,00

Von BASIC zu C mit dem Atari ST
Hartwig
Data Becker Verlag
DM 39,00

Alle Preisangaben sind ohne Gewähr



Das UCSD-P-System

Historie

Als Professor Niklaus Wirth gegen Ende der 60er Jahre die Programmiersprache Pascal entwarf, arbeitete man noch mit Lochkarten und Magnetbändern. Die neue Sprache, die auf den Erfahrungen mit PL/1 und Algol aufbaute, entsprach also den Erfordernissen der damaligen Hardware mit allen darausfolgenden nachteiligen Konsequenzen. Dies beeinträchtigte jedoch bis heute die Verbreitung von Pascal keineswegs. Pascal ist ein exzellentes Programmierlehrmittel und gleichzeitig dafür prädestiniert, hochkomplexe Probleme strukturiert und einfach zu programmieren.

In der folgenden Zeit wurde immer wieder versucht, die Beschränkungen der Urversion durch alle möglichen Erweiterungen zu mildern. Es wurden einige „Pseudo“-Standards festgelegt, mit der Hoffnung auf Einheitlichkeit. Solche Versuche waren ja leider schon bei der Programmiersprache Basic gescheitert.

Bis Mitte der 70er Jahre an der University of California at San Diego das UCSD-Pascal entwickelt wurde.

Wirth's Pascal wurde dabei den höheren Anforderungen der Software-Entwicklung und vor allem der moderneren Hardware angepaßt.

Dieses neue Pascal entstand auf einer – extra dafür entwickelten – Programmierumgebung: Dem UCSD-P-Betriebssystem.

Das P-System als Alternative zu TOS

Die Entwickler des P-Systems hatten sich folgende drei Ziele gesetzt:

- Die Benutzeroberfläche muß für den Anfänger leicht verständlich und für den Experten akzeptabel sein.
- Das System muß in einen Computer mit minimal 64K Bytes plus Diskettenstation implementierbar sein.
- Das System muß absolut Hardware-unabhängig und somit portabel sein.

Das endgültige P-System entsprach diesen – sicher nicht einfach zu bewältigenden – Punkten in jeder Hinsicht.

Dies begründete auch den sich schon bald einstellenden Erfolg des UCSD-Pascal-P-Systems: Jeder kann es benutzen. Es ist völlig egal, ob man vor einem Mainframe oder einem Homecomputer sitzt, das P-System ist universell.

Es ist bis heute auf über 150 verschiedenen Rechnern mit den unterschiedlichsten Prozessoren von VAX-Systemen über den 68000, der 80XX-Familie der 65XX-Serie und einigen anderen bis hin zum guten alten Z80 verfügbar.

Es gibt acht verschiedene Nativ-Code-Generatoren für das P-System. Damit lassen sich acht verschiedene CPU's auf jeden beliebigen Rechner simulieren. Somit ist der Programmierer in

der Lage, Assemblerprogramme für jeden beliebigen Computer zu entwickeln. Außer dem UCSD-Pascal-Compiler gibt es noch einen Basic- und einen Fortran-77-Compiler, wobei man aus jeder Sprache Routinen aus jeder beliebigen anderen Sprache

sowie fremde Betriebssystemroutinen aufrufen und ausführen kann.

Unter UCSD-Pascal ist es auch Multitasking möglich, wobei die verschiedenen Prozesse über Semaphoren synchronisiert werden können. Chaining von Programmen sowie virtuelle und dynamische Speicherplatzverwaltung werden unterstützt.

Die Erzeugung von Subdirectories ist möglich.

Auch Turtlegrafik wird unterstützt, wobei die erzeugten Bilder auf jeden anderen Rechner übertragbar sind, da das Turtlegrafik-Unit mit eigenem, rechnerunabhängigen Grafikmodus arbeitet.

So funktioniert das P-System

Das „P“ bedeutet Pseudo. Dieses Pseudosystem läuft ausschließlich auf einem Pseudocomputer, der P-Maschine, die auch nur eine Pseudosprache, den P-Code, versteht.

Da es keinen Computer gibt, der diesen P-Code versteht, muß er eben simuliert werden. Dies geschieht durch den P-Maschinen-Emulator, dem PME.

Die P-System-Compiler übersetzen ausschließlich in den P-Code und sind somit auf jedem Rechner, der den PME besitzt, ohne jede Änderung ablauffähig.

Um das P-System auf einen neuen Rechner anzupassen, ist es nur nötig, den PME und das BIOS des P-Systems zu ändern.

Da der PME den P-Code erst interpretieren muß, ist er um einiges langsamer als reiner Maschinencode.

Trotzdem ist die Zeitkomplexität bei den meisten Programmen bei den P-Code besser als beim Maschinencode, da sie als P-Code bis zu fünfmal kürzer sind.

Dies ergibt zusätzlich den Vorteil, einiges an Speicherplatz sparen zu können, was in der 8-Bit-Ära eine große Bedeutung hatte und heute, bei weiter fallenden Preisen für Peripheriespeicher und RAM's, die gleichzeitig immer größere Kapazitäten besitzen (bestes Beispiel dafür ist die ST-Serie), nur noch ein hübscher Nebeneffekt ist.

Alles Menügesteuert

Nach dem Booten des P-Systems erscheint die Versionsnummer des Bios und die Copyright-Angaben, danach wird das Betriebssystem-Menü (auch Kommando-Menü oder -Ebene genannt) sichtbar. Von dort aus kann man die verschiedenen P-System-Komponenten auswählen oder auch P-Code-Programme ablaufen lassen.

Am Ende einer jeden Befehlsausführung kehrt man automatisch in die Kommando-Ebene zurück.

Das P-System ist vollständig menügesteuert und baumstrukturiert, was bedeutet, daß jeder Menüpunkt der Kommandoebene ein eigenes Untermenü besitzt, das wiederum ein Untermenü besitzen kann. Ein Komman-

do-Menüpunkt kann bis zu drei Untermenü-Ebenen tief gestaffelt sein.

Alle Menüpunkte werden durch das Betätigen von nur einer einzelnen Taste aktiviert.

Diesem hierarchischen Aufbau ist es hauptsächlich zu verdanken, daß auch ein Anfänger nach einer relativ kurzen Eingewöhnungszeit in der Lage ist, dieses komplexe Betriebssystem angemessen zu benutzen.

Bei Bedienungs- oder Systemfehler meldet sich das System mit einer entsprechenden Fehlermeldung, die auch die Möglichkeit zur Fehlerbehebung angibt.

Falls eine Behebung des Fehlers nicht möglich ist, so muß in manchen Fällen das komplette System neu gebootet werden. Dies ist bei einem Atari ST, dessen TOS nicht Rom-resident ist, noch ziemlich ärgerlich, da nicht nur das P-System, sondern auch das TOS neu gebootet werden muß.

Die Kommando-Ebene

Das Kommando-Menü zeigt alle wichtigen Betriebspunkte des Systems an:

Es ist ein Editor vorhanden, um Quellprogramme einzugeben oder einfach Texte niederzuschreiben. (Es sind geringe Textverarbeitungsfähigkeiten vorhanden.)

Ein Filer übernimmt alle Aufgaben, die sich bei der Verwaltung und Pflege von Dateien ergeben.

Der Compiler übernimmt die Übersetzung des Programmtextes (in den Sprachen Pascal, Fortran oder Basic) in den P-Code.

Mit dem Linker können P-Code-Programm-Module zu einem neuen Programm zusammengefügt werden. (Wobei es gleichgültig ist, ob es sich um einen ursprünglichen Pascal-, 8086-, Fortran- oder 68000-Code handelt.)

Es ist auch ein Assembler-Menüpunkt vorhanden, der den Assembler-Code eines der acht verschiedenen CPU-Familien mit dem entsprechenden Nativ-Code-Generator in den P-Code übersetzt. Ein symbolischer Debugger übernimmt das Austesten des compilierten P-Codes.

Zur Protokollierung der Tätigkeiten bei der Arbeit mit dem P-System ist

Die Preisrevolution auf dem Computermarkt.



ATARI
Besuchen Sie uns!

MÜNZENLOHER GMBH
Fordern Sie Prospekte an!
Tölzer Straße 5
D-8150 Holzkirchen
Tel. (08024) 1814

wir machen Spitzentechnologie preiswert

Ein Hochleistungscomputer f. d. Klein-, Mittelbetrieb und für alle Handwerker.
Ihr Atari-Partner im Süden v. München!
Wir haben die neuen Atari-Geräte auf Lager - Kommen Sie zum Testen!

FARBMONITORE
Thomson CM 36382 AR mit Anschlußkabel 1.398,-
Sony KX-14CP1 1.498,-
Orion 1280 1.099,-

DRUCKER
Panasonic-Drucker KX-P-1080 m. Schönschrift, 100 Zeichen/sec 998,-
Panasonic-Drucker KX-P-1091 m. Schönschrift, 120 Zeichen/sec 1.198,-
Panasonic-Drucker KX-P-1092 m. Schönschrift, 180 Zeichen/sec 1.398,-
Juki-Farbd. 5520, 180 Z./sec. 1.795,-
Drucker-kabel 69,-

ZUBEHÖR
Staubschutzhülle, Atari 520ST 39,-
Staubschutzhülle, Floppy 354/314 29,-
Staubschutzhülle, Monitor SM 124 49,-
PC-Gehäuse lackiert 198,-

PROGRAMMIERSPRACHEN
GST-C Compiler 355,-
LATTICE C Comp. v. Metacomco 595,-
GST-Assembler 179,-
Makro-Assembler v. Metacomco 395,-
UCSD PASCAL - Comp. u. Linker 249,-
PASCAL Comp. v. Metacomco 475,-
Hippo-C 288,-
dBase komp. Datenbank 398,-
VIP-Professional - Datenbank, Kalkulation, Graphik 748,-
Geschäfts- u. Anwenderprogramme
ST-Managerpaket: Kunden, Lager, Rechnung 598,-

ST-Manager-Lohn - Lohnabrechn. 279,-
ST-Text - vielseitige Textverarb. 239,-
ST-Address: 2600 Adressen/Disk 49,-
ST-Address/Notiz 79,-
ST-Literaturverzeichnis 49,-
ST-Rundschreiben (Serienbriefe) 79,-
Superprogramme für GRAFIK
Easy-Draw, Zeichn., Konstruieren, Vertexten i. versch. Schriften u. Gr. 448,-
Degas: Texte (HI-RES-Grafik) 159,-
Typesetter ST, 23 Schriften in 12 Größen 198,-
UTILITIES
Calculator (Taschenrechner) 59,-
Bildschirm-Editor 98,-
IIC-Diskmon - Diskettenmonitor 89,-
S-Term - Kommunik.-Programm-Paket, Gem 199,-
Utility-Disk - viele nützliche Hilfsprogramme 248,-
VT 100 Terminal Emulator 248,-
Spread - Kalkulations-Programm, dt., mit Gem 237,-
Hippo Simple - Dateiverwaltung 149,-
Hippo Ram Disk 99,-
SPIELE
Hippo Back Gammon - Grafikspiel 99,-
Kings Quest II (Sierra) 145,-
Mudpits (Mich Tron) 69,90
A Mind Forever Voyaging 172,-
Starcross 169,-
Deadline 89,-
Hitch Hiker's 110,-
Planetfall 89,-
Seastelker 89,-
Bratcas 132,-
Zkui-West 129,-
Borrowed Time 89,-
Hacker 89,-
Mindshadow 89,-



Computer KÖNIGSTEIN



Funkfernsehprogramm

ATTY Test in ST 3-86.....98,-DM

Wettersatellitenanlage..2.998,-DM

Oh! Drucker NLQ.....998,-DM

260 ST.....1.298,-DM

260 ST mit 1 MB.....1.548,-DM

Aufrüstung auf 1 MB.....298,-DM

3,5" Markendisketten 10 Stck.

einfache Dichte.....79,-DM

2 Side-00 135 TPI...99,-DM

Angebot 30er Diskkasten..19,-DM

Okimate 20 ein Farbdrucker der sogar

Overheadfolien Druckt !888,-DM

New Star NL 10.....1.148,-DM

Drucker-kabel.....79,-DM

Monitorstecker.....15,-DM

12" Druckerpapier 1000 Blatt28,-DM

die aktuellsten Preise im Mailboxservice: 06174 / 5355

KFC Wiesenstr. 18 6240 KÖNIGSTEIN Tel. 06174/3033 BTX #921069

ein Monitor vorhanden, der die Möglichkeit bietet, alle Aktionen in einem speziellen File festzuhalten. Damit lassen sich vor allem Bedienungsfehler zu einem späteren Zeitpunkt exakt reproduzieren und werden leicht erkennbar.

Das System kann auch neu initialisiert werden, was von großer Hilfe ist, wenn man sich in irgendwelchen Fehlern verstrickt hat. Nach dem Initialisieren reagiert das System, als wäre nie ein Fehler aufgetreten.

Mit dem Halt-Menüpunkt wird das P-System gestoppt, und man kehrt zum Desktop zurück.

Das Execute-Kommando führt ein kompiliertes Programm aus, der Run-Befehl kompiliert ein angegebenes Workfile und läßt es danach ablaufen.

Mit User Restart ist es möglich, ein Programm mehrmals hintereinander auszuführen.

Mit dem Set-Kommando ist es möglich, alle wichtigen Parameter für die Benutzung des Systems zu bestimmen. So zum Beispiel:

- die Zeit und das Datum.
- das Workfile, mit dem gearbeitet werden soll.
- der Name des gewünschten Compilers, Editors oder Assemblers.
- sowie der Name der benutzten Userlibrary, also der Programm-bibliothek, die zur Erstellung des Programmes benötigt wird.

Der Filer

Der Filer übernimmt das komplette File-Management im P-System. Er bewältigt alle Aufgaben, die sich bei der Handhabung mit Dateien ergeben, wie etwa das Erzeugen, Kopieren oder Umbenennen von einzelnen Files.

Falls es einmal nötig ist, mehr Files auf einem externen Speicher unterzubringen, als die Directory verwalten kann, (beim UCSD-P-System sind es maximal 77 Files) so bietet der Filer die Möglichkeit, Subdirectories zu verwalten.

Subdirectories sind an dem Extender '.SVOL' erkennbar.

Der Filer kennt folgende vordeklarierte Extender:

'.TEXT' kennzeichnet ein normales Textfile.

'.BACK' bedeutet das gleiche, nur ist es für Datensicherungszwecke erstellt.

Ein File mit dem Extender '.CODE' beinhaltet direkt ausführbaren Objektcode.

'.INFO': Informationen für den Debugger.

Eine Grafik wird mit '.FOTO' gekennzeichnet und an '.BAD' wird ein File erkennbar, das einen beschädigten Bereich auf der Diskette belegt.

Nun die einzelnen Funktionen, über die der Filer verfügt:

'Bad Blocks': Diese Funktion ermittelt alle fehlerhaften Blöcke auf einer Diskette.

Mit 'Change' kann man einen File- oder Volume-Namen ändern.

Es sind zwei Möglichkeiten vorhanden, die Directory zu listen.

Entweder mit 'List Directory', das alle Files mit ihrer Länge zeigt, oder mit 'Extended Directory List', das zusätzlich Zeit und Datum der letzten Änderung, die Adresse des ersten benutzten Blockes des jeweiligen Files und die Anzahl der belegten Bytes des letzten Blockes sichtbar macht.

Nach dem Namen der gewünschten Volume-Directory kann, durch Komma getrennt, ein anderes Volume angegeben werden. Das Listing wird dann auf dieses Volume geleitet. Dieses Volume kann eine andere Datei in der die Directory abgespeichert wird, oder ein Drucker sein, der im P-System auch ein Volume darstellt.

Der Filer kann in zwei verschiedenen Modi arbeiten:

Der Swap- und der Flip-Mode. Zwischen den beiden kann mit dem Menüpunkt 'Flip Swap/Lock' hin- und hergewechselt werden.

Der Swap-Mode lädt die gerade benötigten Segmente des Filers erst bei Bedarf von Diskette. Dies ist bei geringen Hauptspeicherkapazitäten interessant.

Der Lock-Mode lädt den Filer bei Aufruf komplett in den Speicher.

'Krunch' formatiert den Speicherplatz eines Speichervolumes neu. Er fügt alle belegten Blöcke des Volumes und alle unbenutzten Speicherbereiche zu jeweils einem Block aneinander.

'Make' erzeugt Dummy-Files auf dem benutzten Speichermedium. Es wird Speicherplatz für zukünftige Dateien reserviert.

Mit 'On/Off-Line' können neue Volumes nachgeladen oder auch für den Benutzer gesperrt werden.

'Prefix' ändert den geltenden Volumenamen für den Systemzugriff.

Mit 'Remove' kann das jeweils angegebene File gelöscht werden.

Der Befehl 'Transfer' kopiert Files an ein gegebenes Ziel.

So wird mit der Befehlssequenz „Transfer #9:Testfile.text,★\$“ die Textdatei 'Testfile' von Diskstation A (besitzt im UCSD-P-System die Volume-Nummer 9) ohne Dateinamenänderung (gekennzeichnet durch \$) auf die Systemdiskette, die mit dem Joker ★ bestimmt wird, kopiert. Weitere Joker, die der Filer akzeptiert, sind '=' und '?', die gleichbedeutend sind und für alle Filenamen des Volumes stehen.

Das Kommando 'Volumes' listet alle aktivierten Volumes samt ihren Nummern. Dabei werden Speichervolumes (zum Beispiel Diskstationen oder Harddisk) mit einem '#' gekennzeichnet und die jeweilige Speicherkapazität durch Angabe der verfügbaren Blöcke ersichtlich. Ein Block hat hier die Größe von 512 Byte.

Falls die 'Bad Blocks'-Funktion ein positives Ergebnis geliefert hat, können mit dem Befehl 'Examine' zusätzliche Informationen über die Beschädigungen beschafft werden.

Das Kommando 'Zero' schließlich initialisiert die Directory des angegebenen Speichervolumes. Eine Diskette kann danach nur vom P-System aus benutzt werden. Ein Lesen von TOS aus ist nun nicht mehr möglich.

Der Editor

Zum P-System sind zur Zeit zwei verschiedene Editoren verfügbar: Der „Advanced“ und der „normale“ System-Editor. Letzterer gehört zu dem Lieferumfang des P-Systems.

Bevor ein Editor gestartet werden kann, muß ein Workfile bestimmt werden. Dieses Workfile kann mit der 'Set'-Option festgelegt werden. In die-

ser hat man die Wahl zwischen 'Environment' und 'Marker'. Mit 'Marker' kann man im zu editierenden Text Markierungen bestimmen.

Mit 'Set Environment' hat man die Möglichkeit, alle Parameter einzustellen, die zur Textverarbeitung nötig sind.

Es existieren alle bei besseren Editoren üblichen Befehle wie 'Insert', 'Delete', 'Copy', 'Verify', usw.

Der Editor kann auf vier Wegen verlassen werden:

Verlassen des Editors mit und ohne Erneuern des Workfiles sowie Rücksprung in den Editor mit den gleichen Optionen.

Auf weitere Möglichkeiten des Editors sowie auf seltener benötigte Fähigkeiten des UCSD-P-Systems werden wir an dieser Stelle verzichten.

Das P-System sowie die mitgelieferten (hervorragenden) Utilities und die Turtlegraphic in allen Einzelheiten zu untersuchen, würde den Rahmen dieses Beitrags bei weitem sprengen. Wir werden an anderer Stelle sicherlich weitere interessante Ergänzungen dazu publizieren können.

Doch nun wollen wir uns noch mit einem der zu dem P-System gehörenden Compiler beschäftigen: Dem UCSD-Pascal.

Der UCSD-Pascal-Compiler

Das UCSD-Pascal baut weitestgehend auf dem von Wirth und Jensen definierten Ur-Pascal auf (auch 'Standard'-Pascal genannt). Es weichte vor allem in der erweiterten Stringbehandlung, der Möglichkeit zur Arbeit mit Units sowie der Handhabung von Random-Access-Dateien von der ursprünglichen Version ab.

Man hat mit UCSD-Pascal auch die Möglichkeit, in den Quelltext Compileroptionen einzufügen, um die Compilation nach speziellen Wünschen zu beeinflussen.

Die Optionen werden dem Compiler durch Begrenzer '(★' und '★)' sowie durch vorangestelltes '\$' kenntlich gemacht und durch ein nachgestelltes '+' oder '-' an- oder abgeschaltet.

Zum Beispiel: (★\$L+, I-, Q-★)

Hierbei wird durch L+ das Sourcelisting auf ein vorbestimmtes File geleitet, mit I- wird der I/O-Check

unterbunden, und Q- verhindert eine Compilation ohne Compilerinformationen.

Ein eingefügtes P im Programmtext verursacht etwa einen Seitenvorschub.

Im UCSD-Pascal können 'Long Integers' definiert werden, deren Länge durch den Benutzer festgelegt werden kann.

Ergänzend zu den arithmetischen Standardfunktionen in Pascal existieren die Funktionen LOG, die den dekadischen Logarithmus der gegebenen Zahl berechnet, sowie PWROFTEN, die die Zehnerpotenz der Zahl ermittelt.

Eine Einschränkung gegenüber Standard-Pascal besteht darin, daß Funktionen oder Prozeduren nicht als formale Parameter übergeben werden können.

UCSD-Pascal verfügt über drei spezielle interaktive Dateien: 'Input' als übliche Eingabeoperation, 'Keyboard' als Möglichkeit zur Eingabe ohne deren Anzeige auf dem Bildschirm und 'Output' als definierbare Ausgabeform.

Es ist außerdem möglich, typenfreie Dateien zu erzeugen.

Qualitätsbücher aus dem Heim-Verlag



Das Standardwerk für alle ATARI ST-Besitzer. Auf über 300 Seiten eine klare und verständliche Einführung in die Programmiersprache BASIC, elementare BASIC-Kommandos, Diskettenhandhabung und vieles, was zur perfekten Beherrschung des ATARI ST gehört.

Ein Spitzenbuch mit über 80 Übungs- und Anwenderprogrammen wie z. B.:

Sortierprogramme / Textverarbeitung / Umgang mit sequentiellen- u. Random-Dateien / Fakturierprogramm / Programmieretechniken an ausgewählten Beispielen u. v. a. m.

Zum Buch gibt es die Programmdiskette mit sämtlichen Beispiel-Programmen.

Buch: 49,- DM
Progr.-Diskette: 58,- DM
Erscheint: Februar 1986



Das Buch für den richtigen Einstieg mit dem ATARI ST. Leicht verständlich wird der Lernende in den Lernstoff eingeführt.

Einige der Themen:

die Hardware des ATARI ST

- Überblick über die Systemkomponenten
- Aufstellung des Computers
- Wartung
- Das „Innenleben“

die Software des ATARI ST

- wie arbeite ich mit GEM (das Desktop / Maus / Icons etc.)
- die Programmiersprachen BASIC / LOGO
- Kopieren von Files und Disketten, Löschen und Formatieren u. v. a. m.

Buch: 49,- DM
Progr.-Diskette: 58,- DM
Erscheint: März 1986



Der ideale Einstieg in die Programmierung mit LOGO. Ein Buch für alle, die LOGO schnell verstehen und perfekt erlernen möchten. Ein Buch wie es sein muß, leicht zu verstehen und interessant geschrieben.

Einige der Themen:

- Programme und Prozeduren
- Die Turtlegraphic
- Variablen- und Listenverarbeitung
- Viele ausführlich erklärte Beispielprogramme: z. B.
 - Arbeiten mit relativen und sequentiellen Files
 - Anwendungen aus Mathematik und Physik
 - Hobby- und Spielprogramme u. v. a. m.

Zum Buch gibt es die Programmdiskette mit sämtlichen Beispielprogrammen.

Buch: 49,- DM
Progr.-Diskette: 58,- DM
Erscheint: Februar 1986

• unverb. Preisempfehlung



Anhand von über 100 Programmbeispielen lernen Sie das Programmieren in BASIC. Von einfachen, aber grundlegenden Beispielen bis zur ausgereiften Anwendung findet der ATARI ST-Besitzer Beispiel-Programme, die den perfekten Einstieg in die Programmiersprache BASIC leicht und interessant machen.

Alle Beispiele werden so dargestellt, daß das Verständnis für die Programmstruktur gefördert wird, aber auch die Details der ST-BASIC-Version deutlich werden. Die Programme sind strukturiert aufgebaut und gut dokumentiert.

Ein hervorragendes Buch, das Ihnen richtiges Programmieren von Anfang an vermittelt.

Zum Buch gibt es die Programmdiskette mit sämtlichen Beispielprogrammen.

Buch: 49,- DM
Progr.-Diskette: 58,- DM
Erscheint: März 1986

Die Arbeit mit Units ermöglicht den modularen Aufbau von komplexen Programmen. Ein Unit besteht aus dem Programmkopf, der dessen Bezeichnung enthalten muß. Nun wird ein Programmteil 'Interface' von dem Compiler erwartet. Dieser enthält alle Deklarationen, die von dem rufenden Programm und dem Unit benötigt werden.

Darauf folgt der 'Implementation'-Teil. Er beinhaltet alle lokalen Variablen und Deklarationen des Units.

Letztendlich folgt der eigentliche Programmkörper.

Mit Hilfe dieser Units ist es einfach, sich eine Hilfsprogrammbibliothek anzulegen.

Mit UCSD-Pascal hat man auch die Möglichkeit, sein Programm in Segmente aufteilen zu können. 'Segments' ermöglichen die Overlay-Abarbeitung des Programmcodes. Diese Fähigkeit wird auf dem ST ja in den meisten Fällen ungenutzt bleiben.

Desweiteren unterstützt UCSD-Pascal besonders die Arbeit mit Strings, wofür der ergänzende Datentyp 'String' eingeführt wurde, der eine vordefinierte Länge von 80 besitzt. Man kann diese Standardlänge beliebig ändern, darf jedoch das Maximum von 255 nicht überschreiten.

Nur noch einige Möglichkeiten zur Stringmanipulation:

'Fillchar' etwa ermöglicht es, einen definierten Speicherplatz mit einem bestimmten Zeichen zu füllen.

'Moveleft' und 'Moveright' verschieben Zeichenketten in die bestimmte Richtung um eine angegebene Länge.

Mit 'Scan' kann ein String nach einen Teilstring untersucht werden.

Weiterhin sind die Befehle 'Copy', 'Insert' und 'Delete' vorhanden, deren Funktion wohl klar ist.

Man kann weiterhin die Länge eines Strings und die Position eines Substrings ermitteln.

Das P-System auf dem ST

Um das P-System auf dem Atari ST zu starten, muß man dem 'PSYSTEM.TTP' das 'PSYSTEM.VOL' als Parameter übergeben. Nach einer kurzen

Wartezeit erscheint das Kommandomenu auf dem Monitor.

Es ist auch möglich, durch Umbenennen des 'PSYSTEM.TTP' auf einer bootfähigen TOS-Diskette das P-System automatisch zu starten.

Weiterhin wird es auf dem ST erlaubt, eine Ramdisk zu benutzen, was aber nur in dem Falle des 520+ oder einer auferüsteten Maschine sinnvoll ist. Die Ramdisk kann maximal über 1000 Blocks (512 KByte) verfügen.

Diese Ramdisk zu vergrößern, ist durch das Entfernen von GEM und AES möglich. Dieser Trick verschafft einen zusätzlichen Platz von 320 Blocks (160 KByte).

Jedoch ist es nach der Entfernung von GEM nicht mehr möglich, das P-System nach einem Halt neu zu starten.

Der UCSD-Pascal-Compiler wird in Deutschland von Focus vertrieben. Er kostet zusammen mit dem P-System 890,- DM. Dies ist für Atari ST-Verhältnisse ein ganz schöner Brocken. Man erhält dafür jedoch ein komplettes, ausgereiftes Betriebssystem samt Pascal-Compiler und recht guter Dokumentation.

Diese besteht aus drei Ordnern:

Dem 'UCSD Pascal Handbook', dem 'Operating System' und dem 'Program Development'.

Dies sind drei ordentliche, zu jeder

Frage Antworten gebende, Bücher, die jedoch ausschließlich in Englisch gehalten sind. Dies erschwert außerordentlich den Einstieg in dieses komplexe Betriebssystem, vor allem für in dieser Art von Literatur unbedarften Anwender.

Dies und der breite Rahmen der Anwendungsmöglichkeiten, die zu Anfang überhaupt nicht überblickt werden können, stellen für den Anfänger eine recht große Hürde dar.

Vor allem wurde bei der Benutzung des P-Systems auf dem Atari ST bald klar, daß wirklich nicht alles klar ist.

Das Betriebssystem benahm sich einige Male recht egozentrisch, wobei es nicht ersichtlich wurde, ob nun einfach ein Bedienungsfehler oder ein Fehler im System schuld an allem war.

Des öfteren konnte ein Segment nicht gefunden bzw. gelesen werden, wodurch die Weiterarbeit an dieser Stelle illusorisch wurde.

Auch ist es von Vorteil, mit zwei Diskettenlaufwerken zu arbeiten, da ansonsten recht oft gewechselt werden muß, was eine zügige Arbeit verhindert. Zusätzlich zu dem UCSD-Pascal-Compiler wird von Focus unter dem P-System ein Fortran 77-Compiler (990,- DM), ein Basic-Compiler (790,- DM) und das 'Advanced Development Tool Kit' (490,- DM) angeboten.

(JM)

Jörg Mainusch

Durchbruch in der Halbleiterentwicklung

Nach Jahren des Hinterherlaufens in der Halbleiterherstellung ist in Deutschland endlich ein großer Wurf geglückt: Die Firma N. u. T. Zlos stellt am 1.4.86 ihr sensationelles WOM (Write Only Memory) vor. Dieses neue Bauelement soll es in 2 Versionen geben, als EWOM (durch UV-Licht löschar) und als phlegmatisches WOM mit einem Vergesszyklus von typisch 10 ms. Lukrative Lizenzverhandlungen mit ausländischen Herstellern sind aus unerfindlichen Gründen vorerst gescheitert.

Das ST-Betriebssystem

Mittlerweile ist schon bekannt, daß das Betriebssystem des ST-Computer mit CP/M 68k nicht viel gemeinsam hat, sondern eher mit MS-DOS zu vergleichen ist. Ohne Zweifel ist das TOS eines der wichtigsten Betriebssysteme für den 68000er-Rechner. Ob Rechner anderer Hersteller in Zukunft auf dieses System aufbauen werden, hängt nicht zuletzt von dem Erfolg der ST-Rechner ab.

TOS steht für Tramiel Operating System, nach dem Chef von Atari genannt, und ist von Atari selbst entwickelt worden. In dem Dschungel von Betriebssystemen ist das TOS ein typischer Vertreter von Einplatzsystemen ohne eingebaute Multitaskingfähigkeiten. Das bedeutet, daß der Ablauf von mehreren Programmen vom Betriebssystem nicht unterstützt wird. Aber bevor wir mit den Einzelheiten des TOS anfangen, werden wir erst einmal kurz allgemein über Betriebssysteme eingehen.

Ein Betriebssystem hat die Aufgabe, verschiedene Geräte eines Rechnersystems so zu organisieren, daß der Anwender von der Ansteuerung der Hardware völlig losgelöst ist. So stellt das Betriebssystem ein organisiertes Gebilde dar und erspart dem Anwender, sich mit den hochtechnischen Problemen seines Rechners auseinanderzusetzen. Die Zusammenarbeit zwischen Rechner und Peripheriegeräten, die Verwaltung des Speichers sowie die Organisation von Dateien usw., sind alles Aufgaben, die vollständig vom Betriebssystem erledigt werden. Beinahe alle Programme, wenn es sich um Textverarbeitungs-, Tabellenkalkulations- oder Buchhaltungs-Programme handelt, benötigen eine Eingabe über Tastatur und müssen die Daten zur weiteren Verarbeitung auf einem externen Massenspeicher sichern. Genauso ist die Darstellung solcher Programme über ein Sichtgerät

und/oder einen Drucker notwendig. Das Betriebssystem bietet den eigentlichen Benutzerprogrammen fertige Routinen zur Bewältigung solcher Aufgaben. So kommt der Anwender solcher Programme mit dem Betriebssystem überhaupt nicht in Berührung.

In der Regel sind moderne Betriebssysteme für Microcomputer in einen sogenannten hardware-abhängigen und einen hardware-unabhängigen Teil aufgeteilt. Das hat den Vorteil, daß, wenn das Betriebssystem auf einem anderen Rechner implementiert werden soll, nur der hardware-abhängige Teil geändert werden muß, um eine Anpassung zu erreichen.

Das ST-TOS

Bei den Atari ST Rechnern wurde auch von dem oben genannten Konzept Gebrauch gemacht. So ist das GEMDOS der hardware-unabhängige und das BIOS und das XBIOS der hardware-abhängige Teil.

Die GEMDOS Funktionen

\$00	TERM
\$01	CONIN
\$02	CONOUT
\$03	AUXIN
\$04	AUXOUT
\$05	PRINTOUT
\$06	RAWCONIO
\$07	CONIN WITHOUT ECHO
\$08	CONIN WITHOUT ECHO
\$09	PLINE
\$0A	READLINE
\$0B	CONSTAT
\$0E	SETDRV
\$10	CONOUT STAT
\$11	PRTOU STAT
\$12	AUXIN STAT
\$13	AUXOUT STAT
\$19	CURRENT DISK
\$1A	SET DTA
\$20	SUPER
\$2A	GET DATE
\$2B	SET DATE
\$2C	GET TIME
\$2D	SET TIME
\$2F	GET DTA

hardware-abhängige Teil. Das GEMDOS enthält wichtige Routinen zur Tastatureingabe, Ausgabe auf dem Bildschirm und Ansteuerung anderer vorhandener Schnittstellen. Das BIOS stellt, wie schon gesagt, den hardware-abhängigen Teil des TOS dar und bildet eine 'Schnittstelle' zwischen GEMDOS und Rechner. Alle Funktionen, die mit der Ein/Ausgabe zu tun haben, werden vom BIOS erledigt. Dies XBIOS ist eine Erweiterung des eigentlichen BIOS und für die Atari ST Rechner spezifisch.

Der Aufruf mittels TRAP-Befehl

Die Funktionen des TOS sind für den Programmierer einfach zu erreichen. Somit ist es möglich die Benutzeroberfläche GEM zu umgehen und Programme zu schreiben, die nur unter TOS arbeiten. Der Aufruf solcher Funktionen wird über den TRAP-Befehl durchgeführt. Dieser Befehl des

\$30 GEMDOS NUMBER
\$31 KEEP PROCESS
\$36 GET DISK FREE SPACE
\$39 MKDIR

\$3A RMDIR
\$3B CHDIR
\$3C CREATE
\$3D OPEN
\$3E CLOSE
\$3F READ
\$40 WRITE
\$41 UNLIK
\$42 LSEEK
\$43 CHANGE MODE
\$45 DUP
\$46 FORCE
\$47 GETDIR
\$48 MALLOC
\$49 MFREE
\$4A SETBLOCK
\$4B EXEC
\$4C TERM
\$4E .SFIRST
\$4F SNEXT
\$56 RENAME
\$57 GSDFTOF

68000 Prozessors ist ein 'Software-interrupt' und wird, solange kein Fehler vorliegt, sofort ausgeführt. Bei dem TRAP-Befehl wird zuerst das Statusregister sowie der aktuelle Inhalt des Programmcouters in den Supervisor-Stack gerettet. Anschließend wird die durch den TRAP-Vektor definierte Adresse als neuer Programmcouter geladen und der Prozessor fährt an dieser Adresse mit der Verarbeitung fort.

Bei dem TOS werden erst die Parameter (falls vorhanden), dann die gewünschte Funktionsnummer der gewählten Routine und danach die entsprechende TRAP-Nummer auf dem Stack abgelegt.

Das GEMDOS

Wie schon oben erwähnt wurde, bildet das GEMDOS den hardware-unabhängigen Teil des TOS und ist für die Funktionen der Tastatureingabe, Bildschirmdarstellung sowie das File-Handling zuständig. Die Funktionen des GEMDOS werden durch die TRAP-Nummer 1 aufgerufen. Der Aufruf sieht folgendermaßen aus: Erst werden die Parameter auf den Stack (sp) abgelegt, dann die Funktionsnummer, und anschließend wird die Funktion durch den Befehl TRAP #1 ausgeführt. Nach der Funktionsausführung muß eine sogenannte Korrektur des Stacks vorgenommen werden.

Die Funktionen des GEMDOS

Das GEMDOS hat insgesamt 51 Funktionen, und diese alle in einer Ausgabe zu beschreiben, würde zu viel Platz beanspruchen. Deswegen wird die Beschreibung der Funktionen auf mehrere Ausgaben verteilt. Wir werden uns bemühen, bei den meisten Funktionen ein kleines Beispiel anzugeben, wenn wir dies für sinnvoll halten.

\$00 TERM

Bei dem Aufruf dieser Funktion wird das momentan laufende Programm beendet und kehrt an dasjenige Programm zurück, von dem aus es gestartet wurde. Ein solches Programm, das vom Desktop gestartet wurde und nach Aufruf der Term-Funktion zu-

rück in den Desktop kehrt, ist zum Beispiel ein sogenanntes Applikations-Programm.

```
CLR.W -(SP)
TRAP #1
```

\$01 CONIN

Durch CONIN wird ein Zeichen über die Tastatur geholt. Das Programm wartet, bis ein Zeichen zur Verfügung steht. Das geholte Zeichen steht im Datenregister D0, wobei das untere Byte des Low Words den ASCII-Code und das untere Byte den High Words den sogenannten Scan-Code der gedrückten Tasten enthalten.

```
CONIN
MOVE.W #1, -(SP) ;Funktionsnummer
TRAP #1 ;GEMDOS Aufrufen
ADDQ.L #2, SP ;Stack Korrektur
CLR -(SP)
TRAP #1
END.
```

Inhalt von Register D0 nach Betätigung von BACKSPACE:
D0 = 000E 0008

\$02 CONOUT

CONOUT erlaubt, Zeichen auf dem Bildschirm darzustellen. Der ASCII-Code des darzustellenden Zeichens wird auf dem Stack abgelegt und dann die Funktion aufgerufen.

```
CONOUT
Loop: MOVE.B #9, D1 ;Anfangswert für Schleife
MOVE.W #67, -(SP) ;Zeichen 'C'
MOVE.W #2, -(SP) ;Funktionsnummer
TRAP #1 ;GEMDOS Aufrufen
ADDQ.L #4, SP ;Stack Korrektur
SUBQ #1, D1 ;Dekrementiert D1
CMP #0, D1 ;Ist D1 gleich 0?
BNE Loop
CLR -(SP)
TRAP #1
END.
```

\$03 AUXIN

Durch Auxiliary Input liest der Rechner ein Zeichen über die RS 232 Schnittstelle ein. Die Funktion ist beendet, wenn ein vollständiges Zeichen empfangen wurde. Auch hier steht das Zeichen anschließend in dem unteren Byte des Low Words des Datenregisters D0.

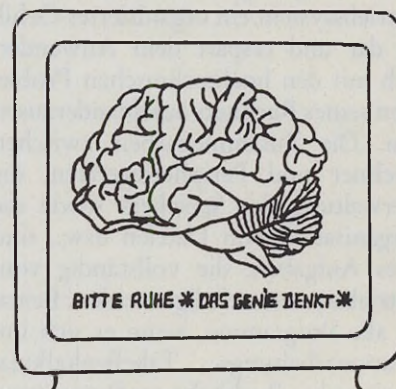
\$04 AUXOUT

AUXOUT bildet das Gegenstück von AUXIN, und dadurch wird ein Zeichen über die RS 232 Schnittstelle gesendet. Man sollte die oberen Byte vor Sendung eines Zeichens löschen.

\$05 PRINTER OUTPUT

Diese Funktion gibt Daten auf einen Drucker über die Centronics-Schnittstelle aus. Bei einer gelungenen Ausgabe (angeschlossener und eingeschalteter Drucker steht im Register D0 der Wert -1. Falls nach ungefähr 30 Sekunden keine Bestätigung erfolgt, daß das Zeichen angekommen ist (OFF LINE), so enthält das Register D0 eine Null.

```
PRINT OUT
MOVE.W #65, -(SP) ;Zeichen das gedruckt wird
MOVE.W #5, -(SP) ;Funktionsnummer
TRAP #1 ;GEMDOS Aufrufen
ADDQ.L #4, SP ;Stack Korrektur
CLR -(SP)
TRAP #1
END.
```



\$06 RAWCONIO

RAWCONIO ist eine Funktion, die das Übernehmen von Zeichen von der Tastatur und seine Darstellung auf dem Bildschirm ermöglicht. Um die Funktion überhaupt zu starten, wird der Stack zuerst mit dem Wert \$FF geladen und dann die Funktion aufgerufen. Die Eingabe erfolgt wie bei der schon besprochenen CONIN-Funktion. Scan- sowie ASCII-Code werden ins Register D0 übergeben. Ein anderer Wert als \$FF wird als Zeichen interpretiert und auf dem Bildschirm dargestellt.

RAWCONIO

```
Start: MOVE.W # $FF, -(SP) ;Prüft Tastatur
      MOVE.W # 6, -(SP) ;Funktionsnummer
      TRAP #1 ;GEMDOS Aufrufen
      ADDQ.L #4, SP ;Stack Korrektur
      TST.W D0 ;Zeichen eingetroffen?
      BEQ Start ;Wenn nicht 'Start'
      CMP.B #13, D0 ;Wenn RETURN
           ;Eingabe beendet
      BEQ End
      MOVE.B D0, -(SP) ;Inhalt von D0 in Stack
      MOVE.W #6, -(SP) ;RAWCONIO
      TRAP #1 ;GEMDOS
      ADDQ.L #4, SP ;Stack Korrektur
      BRA Start
End: CLR -(SP)
     TRAP #1
END.
```

\$07 DIRCONIN

Diese Funktion ist der Funktion CONIN sehr ähnlich. Der Unterschied liegt nur darin, daß das eingelesene Zeichen nicht auf dem Bildschirm dargestellt wird. Hier ein Beispiel, das auf die Eingabe von RETURN wartet:

WARTE:

```
MOVE.W #7, - (SP)
TRAP #1
ADDQ.L #2, SP
CMP #13, D0
BNE WARTE

CLR - (SP)
TRAP #1

END
```

\$08 CONIN WITHOUT ECHO

Diese Funktion unterscheidet sich nicht von der oben beschriebenen \$07-Funktion. Die einzige Erklärung ist, daß diese Funktion auch bei MS-DOS vorhanden ist.

\$09 PLINE

Wir haben schon oben bei der Funktion CONOUT die einfache Form, ein Zeichen auf den Bildschirm auszugeben, besprochen. PRINT LINE bietet eine sehr komfortable Art, Strings auszugeben. Betrachten wir das nächste Beispiel:

PRINT LINE

```
DC.B "***** ST Computer *****"
DC.B 13,10,13,10,13,10,0

MOVE.L #Tab, -(SP)
MOVE.W #9, -(SP) ;Funktionsnummer
TRAP #1 ;GEMDOS Aufrufen
ADDQ.L #6, SP ;Stack Korrektur

CLR -(SP)
TRAP #1

END.
```

GEM™ - Software

Profi - Software für den ATARI ST

C-adress/C-text

DM 285.00

C-auftrag/C-text

DM 570.00

C-video / C-text

DM 998.00

C-vertrieb / C-text

DM 998.00

C-EPROM

DM 387.00

C-lotto

DM 87.50

C-diskeditor

DM 87.50

Adressenverwaltung

Textverarbeitung

Auftragsabwicklung

Lagerverwaltung

Videothekenverwaltung

BAR-CODE-Leser

Kundenverwaltung, Termine

Aufträge, Rundschreiben

EPROM-Programmiergerät

Software

Das Spiel um Millionen

Systemtips

Disketteneditor

Ihr ATARI-Händler

ATARI-Hardware

STAR-Drucker

Olivetti M24, Tandon AT

Programme :

Lattice C-Compiler DM 380.00

H & D Base DM 348.00

dBASE II komp. Datenbank

VIP DM 798.00

Spread-ST, deutsch DM 228.00

Assembler (Metac.) DM 198.00

Brataccas, Spiel DM 118.00

und viele andere

Preise inkl. MWSt. Weitere Programme in Entwicklung. Fragen Sie an !

C-soft

GmbH
Programmentwicklung & Hardware

8400 Regensburg

Tel. 0941/83986

\$0A RLINE

READ LINE bildet eine sehr einfache und bequeme Eingabe von Zeichen über die Tastatur. Die Zeichen werden unmittelbar auf dem Bildschirm dargestellt. Die Zeichen werden in einen 'Eingabe-Puffer' übergeben, dessen Adresse zuerst als Parameter in Stack abgelegt sein muß. Das erste Byte enthält die maximale Länge des Puffers. In das zweite Byte wird die Anzahl der eingegebenen Zeichen nach Beendigung der Eingabe geschrieben. Control Zeichen wie Cointrol-H (BACKSPACE) oder Control-C (beendet Programm) werden erkannt. Nicht aber die Escape-Sequenzen.

\$0B CONSTAT

Die über die Tastatur eingegebenen Zeichen werden zunächst in einen vom Betriebssystem reservierten Puffer untergebracht. Der Puffer ist 64 Bytes groß. Somit kann eine Prüfung der Tastatur stattfinden. Ein Wert von \$FFFF in Register D0 signalisiert, daß Zeichen vorhanden sind. Ansonsten wird eine Null in das Register geschrieben.

\$0E SETDRV

Mit SET DRIVE kann man das aktuelle Laufwerk mit einer bestimmten Nummer anmelden. Bei einer Null als Parameter wird das Laufwerk A und bei einer Eins das Laufwerk B zugewiesen.

SET DRIVE

```
MOVE.W #1, -(SP) ;Funktionsparameter
MOVE.W #$E, -(SP) ;Funktionsnummer
TRAP #1 ;GEMDOS Aufrufen
ADDQ.L #4, SP ;Stack Korrektur
CLR -(SP)
TRAP #1
END.
```

Inhalt von Register D0 :
D0 = 0000 000B

\$10 CONOUT STAT

Diese Funktion prüft den Console-Status. Ist nach dem Aufruf von CONOUT STAT ein Wert von \$FFFF in Register D0 enthalten, kann ein Zeichen auf dem Bildschirm dargestellt werden. Ist aber eine Null im Register, so ist auf dem Sichtgerät keine Ausgabe möglich.

CON STAT

```
MOVE.W #$10, -(SP) ;Funktionsnummer
TRAP #1 ;GEMDOS Aufrufen
ADDQ.L #2, SP ;Stack Korrektur
CLR -(SP)
TRAP #1
END.
```

Inhalt von Register D0:
D0 = FFFF FFFF

\$11 PRTOUT STAT

PRINTER OUT STAT liefert den Status eines angeschlossenen Druckers. Ist der Drucker betriebsbereit (eingeschaltet, ON LINE), wird in Register D0 \$FFFF geschrieben. Ist jedoch der Drucker nicht empfangsbereit, wird ein Wert Null zurückgegeben.

PRTOUT STAT

```
MOVE.W #$11, -(SP) ;Funktionsnummer
TRAP #1 ;GEMDOS Aufrufen
ADDQ.L #2, SP ;Stack Korrektur
CLR -(SP)
TRAP #1
```

END.

Inhalt von Register D0, bei einem angeschlossenen Drucker: D0 = FFFF FFFF

\$12 AUXIN STAT

Nach dem Aufruf dieser Funktion wird signalisiert, ob ein Zeichen über die RS 232 Schnittstelle zur Verfügung steht. Wie bei den anderen Funktionen ist auch hier der entsprechende Wert, ob ein Zeichen vorhanden ist, in Register D0 enthalten.

3 1/2" + 5 1/4"-Floppy-Disk für ATARI 520/260

720 KB — Einzelstation als Zweitlaufwerk, anschlußfertig	(1 x 3 1/2")	548,— DM
dito.	(1 x 5 1/4")	568,— DM
720 KB — Einzelstation als Erstlaufwerk, wie oben, aber mit eingeb. Netzteil		
	Aufpreis	50,— DM
1,4 MB — Doppelstation, anschlußfertig, mit Netzteil	(2 x 3 1/2")	948,— DM
dito.	(2 x 5 1/4")	988,— DM
1,4 MB — Doppelstation, anschlußfertig mit Netzteil	(1 x 5 1/4" + 1 x 3 1/2")	968,— DM
ACHTUNG: 5 1/4"-Laufwerke auch 40/80 Spuren umschaltbar erhältlich		
(z.B. für Datentransfer IBM/ATARI), Aufpreis pro Laufwerk		105,— DM

Computer + Software Ulrich Schroeter
Scheider Str. 12 · 5630 Remscheid 1 · ☎ 02191/21034

\$13 AUXOUT STAT

AUXOUT STAT bildet das Gegenstück der AUXIN-Funktion. Dadurch wird ermittelt, ob die RS 232 sendebereit ist oder nicht. Ein Wert gleich \$FFFF in Register D0 bedeutet, daß die Schnittstelle in der Lage ist, ein Zeichen zu senden.

\$19 CURRENT DISK

Diese Funktion ermittelt die Nummer des aktuellen Laufwerks. Nach dem Aufruf erhält man in Register D0 die Nummer des aktiven Laufwerks zurück. Wie bei SET DRIVE entspricht eine Null Laufwerk A und eine Eins Laufwerk B.

CURRENT DISK

```
MOVE.W #$19,-(SP)
TRAP    #1
ADDQ.L  #2,SP
CLR     -(SP)
TRAP    #1
END.
```

Inhalt von Register D0 wenn Laufwerk 'A' angemeldet ist:
D0 = 0000 000A

Fortsetzung folgt

PRINT & TECHNIK VIDEO DIGITIZER

C 64/128	DM 398,-
ATARI 520 ST	DM 598,-
ATARI 520 PRO	DM 898,-
IBM-PC comp.	DM 598,-
IBM-PC comp.PRO	DM 898,-
APPLE 2	DM 498,-
ATARI 800/130	DM 498,-
SCHNEIDER	in Vorbereitung



Der VIDEO-DIGITIZER und eine komfortable Software erlauben ein VIDEO-Signal einer KAMERA oder eines RECORDERS in 4 sec. in den Speicher Ihres Computers (256 x 256) in 16 Graustufen einzulesen. Die professionelle (PRO 512 x 256) Version ist eine weiterentwickelte, verbesserte Version für die Industrie. Die Bilder lassen sich ablegen, mit Malprogrammen weiterverarbeiten und auf vielen Druckersystemen ausdrucken. Mustererkennung und Archivierung sind neue Gebiete.

Computerperipherien

8000 MÜNCHEN 40 · NIKOLAISTR. 2 · TEL. 089 / 36 81 97
KATALOG DM 3,- Täglicher Nachnahmeversand

SOFTWARE für Ihren ATARI ST

- ☐ Programmiersprachen
- ☐ Utilities
- ☐ Unterhaltung
- ☐ GRAFIK

Fordern Sie unseren kostenlosen Prospekt an!



Kersten & Partner
- Datensysteme -

Wildbox Hermannstr. 83
D-5100 Aachen
West Germany
Tel. (0241) 17 006/8

CompWare CompWare

Robert Bunsen Str. 8, 6084 Gernsheim Tel. 0 62 58/5 16 16
Ernst Ludwig Str. 7, 6840 Lampertheim Tel. 0 62 06/5 48 88

CompWare CompWare

	10 St.	50 St.	100 St.	> 100 St.		
3 1/2" Fuji MF 1 DD	7,90	7,20	6,90	auf Anfrage	Orion Farbmonitor CCM 14 mit Kabel an Atari 520/260	775,-
3 1/2" Disketten MF 2 DD	9,90	9,50	9,20	auf Anfrage	Panasonic Drucker KX-P1091, 120 z/sec, diverse Schriftarten	1 075,-
					Panasonic Drucker KX-P1092, 180 z/sec, diverse Schriftarten	1 375,-

Händleranfragen erwünscht

>>>> Wir tun alles damit Sie nicht bei der Konkurrenz kaufen <<<<



Datenbanksysteme

- **Aufbau und Funktionsweise**
- **als Beispiel: dBase II**

Der eine oder andere PC-Besitzer wird sich sicher wundern, daß in einer Zeitschrift, die sich mit einem derartigen neuen Rechner wie dem ATARI ST beschäftigt, über ein schon relativ altes Softwarepaket wie dBase II (III) berichtet wird. Der Artikel soll auch nur jene ansprechen, die mit dem ATARI ST zum erstenmal einen PC erworben haben bzw. von einem Homecomputer aufgestiegen sind.

Nun aber zu dBase. Bei dBase II (III) handelt es sich um ein „relationales Datenbanksystem“. Was ist das? Zur Beantwortung dieser Frage soll zunächst ein Überblick über die drei wichtigsten Datenbankmodelle gegeben werden.

1) Die hierarchische Datenbank

Eine hierarchische Datenbank besteht aus Tabellen, die in einer vorgegebenen Reihenfolge durchgesehen werden müssen, um eine gewünschte Information zu finden. Abbildung 1 zeigt eine hierarchische Datenbank und die Schritte, die zur Erlangung einer Information vorgenommen werden müssen.

2) Die netzwerkartige Datenbank

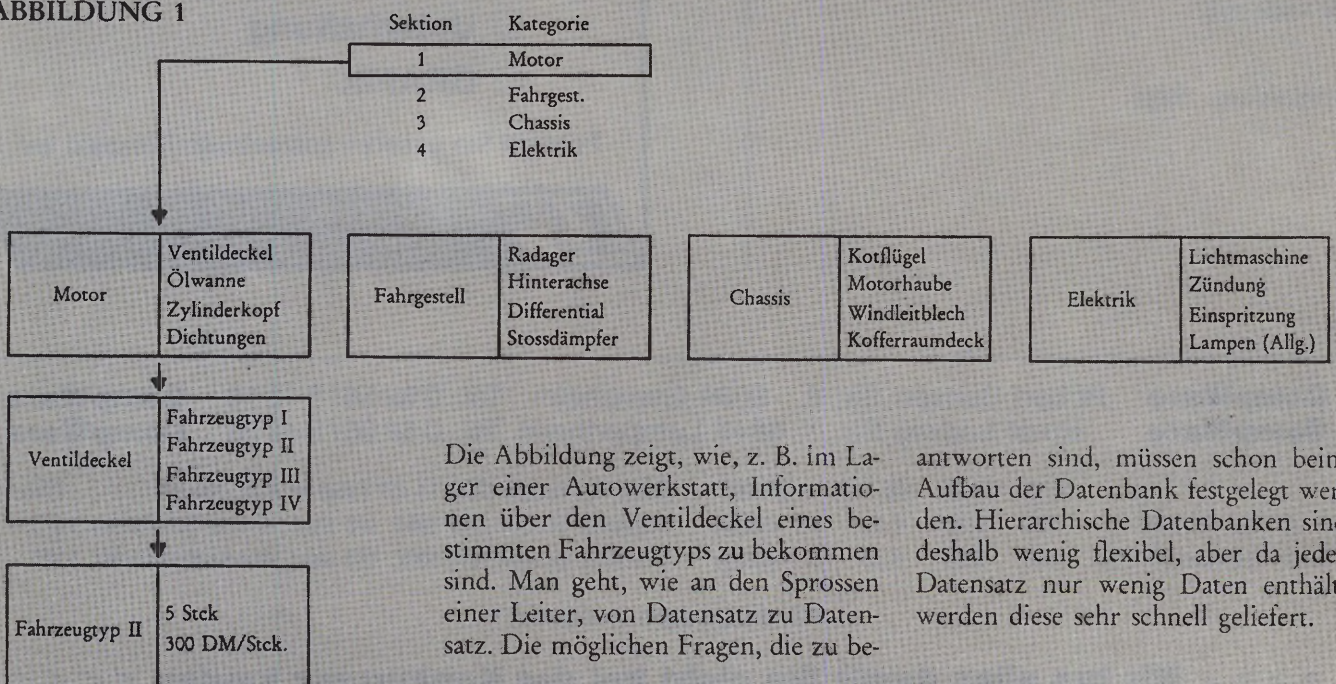
Bei dieser Datenbank handelt es sich vom prinzipiellen Aufbau her um ein ähnliches Modell wie die hierarchische Datenbank, nur daß hier auch noch einzelne Verbindungen zwischen den Datensätzen bestehen können, dies macht die Datenbank etwas flexibler,

da zur Erlangung einer Information nicht mehr die ganze Hierarchie durchlaufen werden muß.

3) Die relationale Datenbank

Dieses Modell unterscheidet sich von den ersten beiden grundlegend. Die Hierarchie der Felder ist nämlich aufgehoben. So lassen sich alle Felder als Schlüssel zum Auffinden von Informationen benutzen. Ein Datensatz wird nicht mehr als eine Menge diskreter Elemente angesehen, von denen eines als Hauptfeld herausgehoben ist, sondern als eine Reihe in einer zweidimensionalen Tabelle, wobei dann jedes Feld zu einer Spalte der Tabelle wird.

ABBILDUNG 1



antworten sind, müssen schon beim Aufbau der Datenbank festgelegt werden. Hierarchische Datenbanken sind deshalb wenig flexibel, aber da jeder Datensatz nur wenig Daten enthält, werden diese sehr schnell geliefert.

ABBILDUNG 2

Ersatzteil	Sektion	Kategorie	Typ	ANZ	Preis/Stck.
Ventildeckel	1	Motor	II	5	300
Kotflügel	3	Chassis	II	2	190
Windleitblech	3	Chassis	II	1	187
Hinterachse	2	Fahrgestell	II	1	970
Zylinderkopf	1	Motor	II	4	197
Differential	2	Fahrgestell	II	0	298
Motorhaube	3	Chassis	II	3	350
Lichtmaschine	4	Elektrik	II	9	160
Lampen (vorn)	4	Elektrik	II	24	58
Zündung	4	Elektrik	II	4	170
Dichtungen	1	Motor	II	88	9,78
Ölwanne	1	Motor	II	1	260
Stossdämpfer	2	Fahrgestell	II	40	150
Radlager	2	Fahrgestell	II	32	40,50
Kofferraumdeckel	3	Chassis	II	1	198
Einspritzung	4	Elektrik	II	2	1024
Ventildeckel	1	Motor	III	3	270

Ventildeckel	1	Motor	II	5	300
--------------	---	-------	----	---	-----

Mit einem Schritt kann hier die Information über den Ventildeckel des Fahrzeugtyps II erhalten werden. Aber es ist auch möglich eine Übersicht über die Ventildeckel aller Fahrzeugtypen zu erhalten

Ventildeckel	1	Motor	I	1	410
Ventildeckel	1	Motor	II	5	300
Ventildeckel	1	Motor	II	3	268
Ventildeckel	1	Motor	IV	4	191

Wie das Beispiel zeigt, sind relationale Datenbanken sehr viel flexibler als hierarchische oder netzwerkartige Datenbanken, allerdings kann hier das Problem auftreten, daß es bei sehr großen Datenmengen relativ lange dauern kann, bis die gewünschten Datensätze gefunden sind.

Nach diesem Exkurs in die Theorie der Datenbanken nun endlich zurück zu dBase II.

Zunächst etwas über die Einsatzmöglichkeiten. Es ist unangebracht, ein begrenztes Gebiet hervorzuheben, etwa den Finanz- oder Personalbereich. Mit der gleichen Berechtigung könnte man das Materialwesen, den Sport oder die Bibliotheksverwaltung nennen. Die Anwendungsbereiche von dBase sind sehr weit gesteckt. Sie decken praktisch alle Gebiete der Datenverwaltung auf dem Personal-Computer.

Es sollte der Phantasie des jeweiligen Anwenders überlassen bleiben, weitere Einsatzmöglichkeiten einer Datenbank zu erkennen und die entsprechenden Daten sinnvoll aufzubereiten. Um die Einsatzmöglichkeiten besser beurteilen

zu können, müssen natürlich die technischen Daten von dBase bekannt sein. (siehe Abb. 3)

Die Grenzen sollen allerdings nicht allzu euphorisch aufgenommen werden, denn wie eingangs erwähnt, kann es bei „großen“ Dateien zu erheblichen Zeitproblemen bei der Bearbeitung einer Datei kommen. So benötigt z. B. dBase II zum Sortieren einer Datei mit 500 Datensätzen ca. 270 sec und zum Finden eines Datensatzes ca. 11 sec, bei dBase III sind diese Zeiten auf ca. 44 sec bzw. ca. 1 sec reduziert. Diese Zeiten können ebenfalls durch das Anlegen von Indexschlüsseln reduziert werden. Es ist deshalb empfehlenswert, nicht zu große Dateien zu verwenden oder gegebenenfalls eine Datei zu teilen. So könnten, z. B. bei der Kundenverwaltung, Adressen und Kontostände in getrennten Dateien abgelegt werden. Eine Verbindung beider Dateien ist in der Regel problemlos (dBase bietet z. B. den Befehl „Join“, näheres ist im Handbuch oder der recht ansehnlichen Literatur zu finden).

Ein weiteres Problem stellt auch die Kapazität der vorhandenen Massenspeicher dar, man stelle sich eine Datei mit 65 535 Datensätzen und 1000 Zeichen Datensatz vor. Man brauchte für eine derartige Datei einen Speicher von ca. 66 MByte Kapazität, hiermit wäre wohl jeder PC weit überfordert. Für das Arbeiten mit Floppy-Disc ist es sicher nicht sinnvoll, Dateien die größer als 100 KByte sind, zu benutzen (Zugriffszeit). Beim Arbeiten mit Winchesterplatte lassen sich auch größere Dateien verwenden. Wobei allerdings bedacht werden sollte, daß jede

ABBILDUNG 3

KENN DATEN UND GRENZEN dBase II

Anzahl Felder/Datensatz	32 max
Anzahl Zeichen/Datensatz	1000 max
Anzahl Datensätze/Datei	65535 max
Anzahl Zeichen/Zeichenkette	254 max
Rechengenauigkeit	10 Stellen
größter Wert	1.8×10^{63} ca
kleinster Wert	1.0×10^{-63} ca

Länge einer Programmdatei unbegrenzt (der integrierte Programmeditor kann „nur“ 5 000 Programmzeilen gleichzeitig bearbeiten)

Datei „gepflegt“ werden muß, hier kann es bei großen Dateien schon zu Problemen kommen.

Nachdem nun ständig das Wort „Datei“ gebraucht wurde, soll endlich erklärt werden, wodurch eine Datei bzw. deren Struktur bestimmt wird. Sie wird nur vom Anwender bestimmt.

Abb. 4 zeigt zwei Beispiele, wie die Struktur einer Datei aussehen könnte.

Die Abbildung zeigt außerdem zwei Beispiele für den Einsatz von dBase sowie die Trennung von zu großen Dateien. Ohne auf die vielfältigen Befehle von dBase einzugehen (näheres auch hier im Handbuch), soll noch die für

die rationelle Arbeit mit dBase vorhandene Programmiersprache erwähnt werden.

Diejenigen Benutzer von dBase, für die Basic und Programmiersprache identisch ist, werden sicher ihre Anfangsprobleme haben, denn die Programmiersprache zwingt zu absolut strukturiertem Programmieren.

Es existiert z. B. nicht der GOTO-Befehl, alle „Sprünge“ Verzweigungen werden über DO-WHILE-Schleifen, Block-IF-Anweisungen bzw. DO-CASE-Schalter realisiert werden. Wer schon mit Pascal oder Fortran 77 gearbeitet hat, wird mit diesen Anweisungen keine Probleme haben. Mit den speziell zur Dateimanipulation vorhandenen Befehle ist man normalerweise vertraut, nachdem man das Handbuch durchgearbeitet hat. Im allgemeinen wurden die Befehle dann alle schon in direktem Dialog getestet. Die bereits manuell eingegebenen Befehle können unverändert in Programmen verwendet werden.

Natürlich gibt es auch in der dBase-Programmiersprache Befehle, die noch unbekannt sind, allerdings sollte deren Erlernen keine Probleme bereiten. Bemerkenswert ist auch, daß „jedes“ dBase Programm als Unterprogramm eines anderen dienen kann, da es keine speziellen Unterprogramme gibt. Wer nach dem Grund fragt, warum ein speziell zur Datenverwaltung erstelltes Programm noch eine Programmiersprache benötigt – der Grund ist schnell genannt. Zwar können alle Befehle direkt im Dialog verwendet werden, aber bei komplizierten Manipulationen, die öfters gebraucht werden, ist dies mit erheblichem Zeitaufwand verbunden. Somit bietet das Schreiben von Programmroutinen eine einfache Möglichkeit zur Zeitersparnis, und außerdem: wer möchte schon mehrmals dieselbe Routine neu schreiben?

ABBILDUNG 4

```
. LIST STRUCTURE
STRUCTURE FOR FILE:  ADRESSEN.DBF
NUMBER OF RECORDS:  00011
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD  NAME      TYPE  WIDTH  DEC
001  NAME      C    010
002  VORNAME   C    010
003  STRASSE   C    015
004  ORT       C    016
005  TELEFON   C    011
★★TOTAL★★          00063
```

Die Abbildung zeigt – nach dem Befehl LIST STRUCTURE – auf dem Bildschirm erscheinende Strukturen oder die Maske einer Datei. Von besonderem Interesse sind die vorletzten sechs Zeilen;

FLD $\hat{=}$ Feldnummer,

NAME $\hat{=}$ Name eines Feldes (beliebig wählbar),

TYPE $\hat{=}$ Type eines Feldes,

c Inhalt alphanum. Zeichen

n Inhalt des Feldes nur numerische Zeichen,

L Inhalt nur logische Variable T (true), F(false),

WIDTH $\hat{=}$ Länge des Feldes.

DEL $\hat{=}$ Anzahl der Dezimalstellen (nur bei num. Feldern)

```
NAME      TYP  LEN  DEC
FIELD 01 :KUNUMMER   C    003  000 :
FIELD 02 :ORT        C    004  000 :
FIELD 03 :lrech      c    005  000 :
FIELD 03 :datum      c    008  000 :
FIELD 04 :umsatz     n    010  002 :
FIELD 05 :bezahlt    n    010  002 :
FIELD 06 :offen      n    010  002 :
FIELD 07 :
```

Dieses Bild zeigt die Struktur einer Datei, deren Struktur verändert wurde. LEN entspricht WIDTH.

Computer preiswert Computer preiswert Computer preiswert Computer preiswert Computer

Neu **Farbmonitor** für 260/520 St+ Schneider 640x200 **nur 699,- DM**

Panasonic Drucker KX-P 1080 NLQ **nur 699,- DM**

Markendisketten 2 DD 135 TPI 10 Stück 50 Stück 100 Stück

8,90 8,50 7,90

Mit Umtauschgarantie!

Händleranfragen
erwünscht!

Lühr's Computerladen · 2245 Tellingstedt · Hauptstraße 1 · ab 14.00 Uhr · (0 48 38) 679

Einführung in die Programmiersprache Pascal

Teil 3



Nachdem im letzten Teil die FOR-NEXT-Schleife behandelt wurde, folgen nun zwei weitere Steuerbefehle. Für diese Schleifenbefehle ist charakteristisch, daß die Anzahl der Durchläufe nicht festgelegt ist, sondern von einer Endbedingung abhängt.

1. REPEAT-Anweisung

Die Anweisung hat die Form:

REPEAT

Anweisung(en)

UNTIL *Bedingung*

Dabei wird der Teil mit der Anweisung bzw. den Anweisungen so lange durchlaufen, bis die (End-)Bedingung erfüllt ist.

Der Anweisungsteil muß nicht mit BEGIN...END umschlossen sein, weil die äußere Form die Anweisungen klar umschließt.

Dadurch, daß bei der REPEAT-Schleife die Abbruchbedingung am Ende steht, wird die Schleife **mindestens einmal durchlaufen**. Erst dann wird geprüft, ob die Schleifenbedingung noch zutrifft. Diese Art der Schleifensteuerung nennt man deshalb auch **nichtabweisende** Schleife. Ihr Struktogramm ist von dem der abweisenden Schleife verschieden (siehe Bild 1).

Diese Eigenheit der REPEAT-Anweisung ist nicht immer erwünscht und muß deshalb besonders beachtet werden.

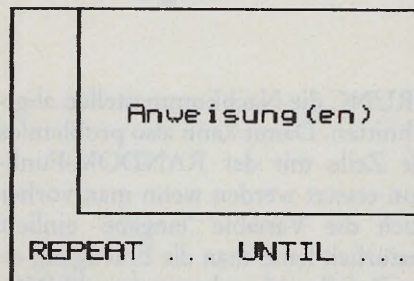
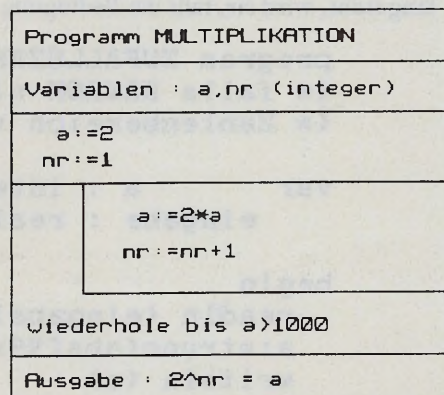


Bild 1: nichtabweisende Schleife

Das Programm 'Multiplikation' (Listing 1) zeigt eine Anwendung der REPEAT-Schleife. Die Schleife wird so oft durchlaufen, bis die Bedingung $a > 1000$ erfüllt ist. Danach wird der Endwert der Variablen a und die Anzahl der Schleifendurchläufe (nr) ausgegeben.



Struktogramm 1

```
program MULTIPLIKATION;
  (* mit Repeat/Until *)

  var a, nr : integer;

  begin

    a := 2;
    nr := 1;

    repeat
      a := a * 2;
      nr := nr + 1;
      writeln (a);
    until a > 1000;

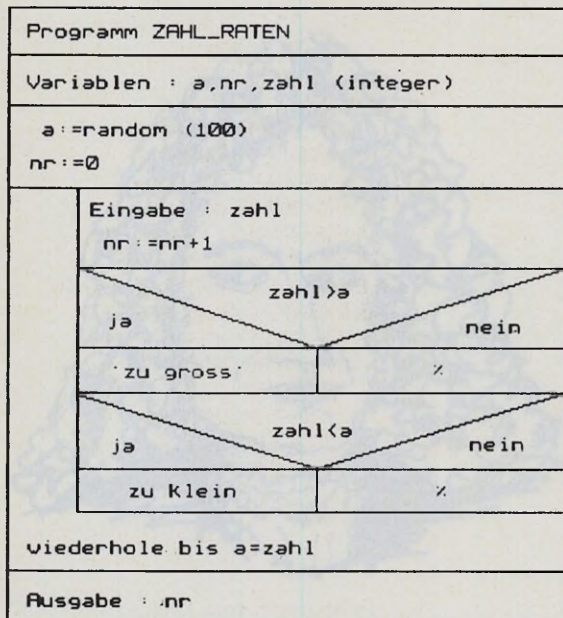
    writeln ('2 ^ ', nr, ' = ', a)

  end.
```

Listing 1

Das Programm 'Zahl Raten' (Listing 2) ist ein kleines Ratespiel. Dabei wird zuerst vom Programm eine 'Zufallszahl' erzeugt. Dies geschieht mit der Funktion **RANDOM** bzw. **RANDOM()**. **RANDOM** generierte eine Zufallszahl zwischen Null und Eins (ausschließlich!), **RANDOM (X)** hingegen eine Zahl zwischen Null und X (ausschließlich!).

Mit **RANDOM (100)** wird also eine Zahl im Bereich von 0 bis 99 erzeugt. Das Programm beinhaltet eine **REPEAT-Schleife**, die solange durchlaufen wird, bis die eingegebene Zahl (zahl) der Zufallszahl (a) entspricht. Dann wird die Anzahl der benötigten Züge ausgegeben.



Struktogramm 2

```

program ZAHL_RATEN;

var a,nr,zahl : integer;

begin
  writeln ('***** ZAHLENRATEN *****');
  writeln ('(Zahlenbereich von 0 bis 99)');
  a := random(100);
  nr:=0;

  repeat
    write ('Deine Zahl : ');
    read (zahl);
    nr:=nr+1;
    if zahl>a then writeln (' ... zu gross');
    if zahl<a then writeln (' ... zu klein');
  until a=zahl;

  writeln;
  writeln ('gratuliere, die Zahl ist gefunden !');
  writeln ('--> benoetigte Zuege : ',nr)
end.
  
```

Listing 2

Bei diesem Programm wird am Anfang $nr=0$ gesetzt. Dies ist bei Pascal, im Gegensatz zu den meisten Basic-Versionen, notwendig, weil die Variablen beim Programmstart nicht automatisch auf Null gesetzt werden. Da einige Pascal-Versionen die Funktion **RANDOM()** nicht implementiert haben, muß man sie simulieren. Dazu gibt es viele Möglichkeiten. Das Programm 'Zufallszahl' (Listing 3) erzeugt eine Zufallszahl.

Dabei werden die Funktionen **ABS(x)** und **TRUNC(x)** verwendet. Die Funktion **ABS** gibt den Absolutwert des Arguments (x) aus.

$ABS(-4) = 4$
 $ABS(-9.12) = 9.12$
 $ABS(4) = 4$

Die Funktion **TRUNC** gibt den Ganzzahlenanteil des Arguments (x) aus, sie 'schneidet' also die Nachkommastellen ab. Sie ist somit dem Befehl **ROUND** ähnlich, doch wird bei dieser Funktion gerundet.

$TRUNC(10.6) = 10$ $ROUND(10.6) = 11$
 $TRUNC(-5.5) = -5$ $ROUND(-5.5) = -5$
 $TRUNC(-5.9) = -5$ $ROUND(-5.9) = -6$

Das Programm errechnet nach Eingabe einer Zahl deren Sinuswert. Da dieser im Bereich Null bis Eins liegen kann, wird der Wert mit 99 multipliziert (obere Bereichsgrenze). Dann wird mit **ABS** ein eventuell auftretendes Minuszeichen beseitigt und mit

TRUNC die Nachkommastellen abgeschnitten. Damit kann also problemlos die Zeile mit der **RANDOM**-Funktion ersetzt werden wenn man vorher noch die Variable 'eingabe' einliest (natürlich kann man die Erzeugung einer Zufallszahl auch mit einer **FUNCTION** bzw. einer **PROCEDURE** realisieren, doch dazu später mehr!).

2. WHILE-Schleife

Diese Schleifenfunktion hat die Form:

WHILE Ausdruck DO
 Anweisung(en)

Falls mehrere Anweisungen innerhalb des Schleifenrumpfes stehen sollen, so müssen diese mit **BEGIN...END** umschlossen werden. Da bei der **WHILE**-Schleife die Bedingung schon am Anfang steht, wird sie, falls die Bedingung

schon beim ersten Durchlauf nicht erfüllt ist, auch nicht durchlaufen. Sie ist deshalb, wie auch die **FOR-TO-DO**-Anweisung, eine **abweisende** Schleife.

Das Programm 'RHOMBUS' (Listing 4) enthält eine **WHILE**-Schleife und darin verschachtelt noch zwei aufeinanderfolgende Schleifen des gleichen Typs. In jeder dieser Schleifen sind wiederum je zwei **FOR**-Schleifen enthalten. Das Struktogramm zeigt deutlich diese Anordnung.

Wenn Sie das Programm gestartet haben, entsteht auf dem Bildschirm ein Rhombus. Statt '★' können Sie auch den ASCII-Code dieses Zeichens mit dem Befehl **CHR(x)** eingeben. Der Wert für x müßte dann 42 lauten.

$CHR(65) = A$
 $CHR(66) = B$ usw.

```

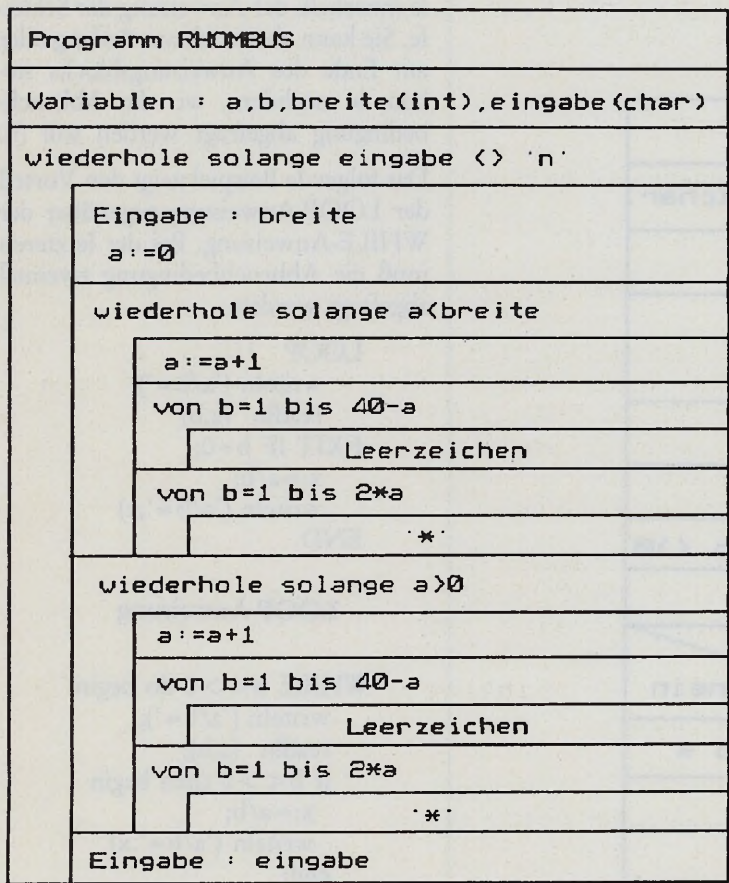
program ZUFALLSZAHl;
(* falls RANDOM nicht vorhanden *)
(* Zahlenbereich von 0 bis 99 *)

var
  a : integer;
  eingabe : real;

begin
  readln (eingabe);
  a:=trunc(abs(99*sin(eingabe)));
  writeln (a)
end.
  
```

Listing 3

Listing 4



Struktogramm 4

Probelauf von Listing 4

$$x \bmod y$$
$$\begin{array}{l} 13 \text{ MOD } 2 = 1 \\ 9 \text{ MOD } 3 = 0 \\ 17 \text{ MOD } 9 = 8 \end{array}$$

Mit der innersten Schleife wird in unserem Beispiel der Faktor ermittelt, durch den die Variable `a` teilbar ist. Die Faktoren sind dabei immer Primzahlen. Die Schleifenvariable `a` wird nun durch den gefundenen Faktor geteilt. Wenn $a=b$ war, dann ist `a` nun Eins und damit die Primfaktorzerlegung beendet. Um `a` durch `b` zu teilen, wurde die **DIV**-Funktion verwendet. Dies ist hier natürlich nicht notwendig, denn diese Funktion schneidet nach dem Teilen einen eventuell auftretenden Nachkommateil ab. Ein

```
program PRIMFAKTORZERLEGUNG;
```

```
var a,b,zahl : integer;
    antwort : char;
```

```
begin
```

```
    while antwort<>'n' do begin
```

```
        writeln ('***** PRIMFAKTORZERLEGUNG *****');
        writeln ('bitte eine natuerliche Zahl eingeben !');
        writeln;
        read  (zahl);
        write (' = ');
```

```
        a:=zahl;
```

```
        while a>1 do begin
```

```
            b:=2;
```

```
            while a mod b <> 0 do
```

```
                b:=b+1;
```

```
                if a=b then writeln (b,' <--- primzahl(en)')
```

```
                    else write (b,' * ');
```

```
                a:=a div b
```

```
            end;
```

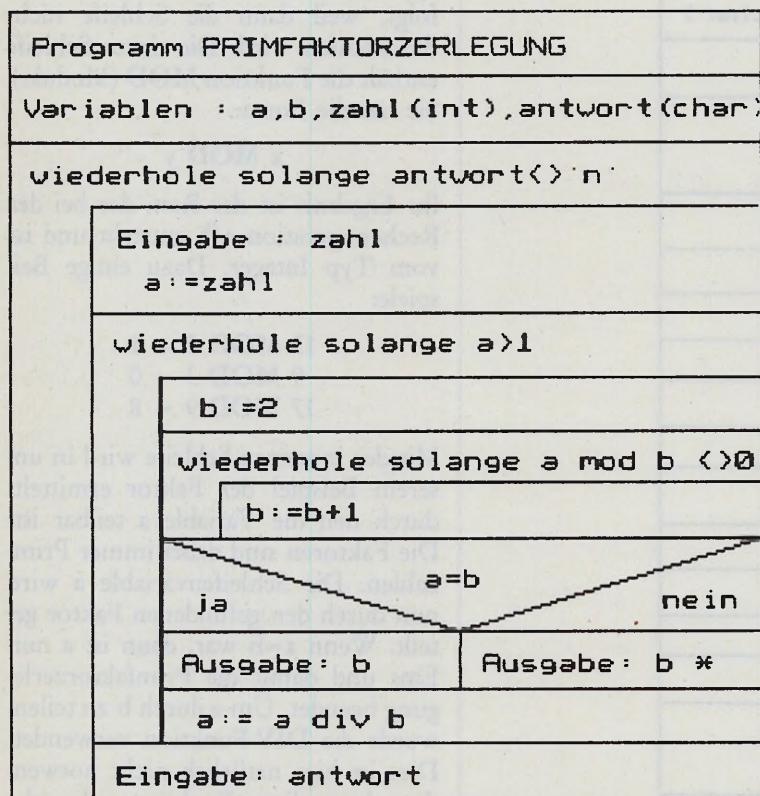
```
        writeln;
```

```
        writeln ('---> Weiter mit < RETURN > !   (n=ENDE)');
        readln (antwort)
```

```
    end;
```

```
end.
```

Listing 5



Struktogramm 5

Nachkommateil kann jedoch bei unserem Programm nicht vorkommen, weil b immer ein Teiler von a ist (siehe MOD). Man kann also in diesem Fall statt $a \div b$ auch a/b einsetzen.

Neben den drei in Standard-Pascal vorhandenen Kontrollstrukturen zur Schleifensteuerung gibt es in manchen Pascal-Versionen (z. B. ST-Pascal von ATARI) noch die LOOP-Anweisung. Sie hat die Form:

LOOP

Anweisung(en)

EXIT IF Ausdruck ;

Anweisung(en)

END

Die Bedingung für den Abbruch (EXIT IF) steht dabei an einer beliebigen Stelle innerhalb der Anweisung der Schleife. Sie kann aber auch am Anfang oder am Ende des Anweisungsblocks stehen, je nachdem, wo die Abbruchbedingung abgefragt werden soll (!).

Das folgende Beispiel zeigt den Vorteil der LOOP-Anweisung gegenüber der WHILE-Anweisung. Bei der letzteren muß die Abbruchbedingung zweimal abgefragt werden.

LOOP

writeln ('a/b=');

readln (a,b)

EXIT IF b=0;

x:=a/b;

writeln ('a/b=',x)

END

LOOP-Anweisung

WHILE b<>0 do begin

writeln ('a/b=');

readln (a,b);

if b<>0 then begin

x:=a/b;

writeln ('a/b=',x)

end;

end

WHILE-Anweisung

```

program HORNER;
  (Berechnung von Funktionswert und Ableitung)
  (an der Stelle x)

var a   : array [0..20] of integer;
    c   : array [0..20] of real;
    n,k,i,x,f : integer;

begin
  writeln ('HORNER-SCHEMA');
  writeln ('Funktionswert und Ableitung an der Stelle x');
  writeln ('Funktionsgrad und x eingeben');
  readln (n,x);
  for k:=n downto 0 do begin          (Werte einlesen)
    write ('a[' ,k, ']= ');
    readln (a[k]);
    c[k] := a[k]
  end;
  for i:=1 to n do                    (HORNER-Algorithmus)
    for k:=n-1 downto i-1 do
      c[k] := c[k+1]*x+c[k];
    writeln;
    write ('f(x) = ');
  for k:=n downto 0 do begin          (Funktion ausgeben)
    if a[k]<>0 then begin
      if (a[k]>0) and (k<>n) then write('+');
      if k=0 then write (a[0])
      else if k=1 then write (a[1], 'x')
      else write (a[k], 'x^',k);
      write (' ')
    end
  end;
  writeln;
  for i:=0 to n do begin              (Funktion & Ableitung)
    if i=0 then f:=1
    else f:=f*i;
    c[i]:=c[i]*f;
    writeln ('f',i, '(',x,') = ',c[i]:8:2)
  end
end.

```

Listing 6

Running HORNER-SCHEMA

Funktionswert und Ableitung an der Stelle x
Funktionsgrad und x eingeben

```

6 3
a[6]= 2
a[5]= -6
a[4]= 8
a[3]= -4
a[2]= 5
a[1]= 8
a[0]= 9

```

$f(x) = 2x^6 - 6x^5 - 4x^3 + 5x^2 + 9$

```

f0(3) = 54.00
f1(3) = 408.00
f2(3) = 1558.00
f3(3) = 3216.00
f4(3) = 4320.00
f5(3) = 3608.00
f6(3) = 1440.00

```

>

Bildschirmdarstellung von Listing 6

Das letzte Programm (Listing 6) behandelt ein Problem aus der Mathematik. Es handelt sich um die Berechnung des Funktionswerts und der Ableitung an einem bestimmten Punkt x. Dazu wird das – sicherlich bekannte – HORNER-Schema verwendet

Wie auch schon bei den letzten Programmen sind hier **Kommentarzeilen** enthalten. Der Text kann an einer beliebigen Stelle im Text stehen. Er wird entweder mit (★...★) oder mit {...} umschlossen und benötigt kein Semikolon am Ende.

Die Faktoren werden in ein Feld (ARRAY) eingelesen. Dieses Feld muß bei der Variablenvereinbarung definiert werden. Die Form der Vereinbarung ist:

ARRAY Indexbereich OF Elementtyp

Beispiel:

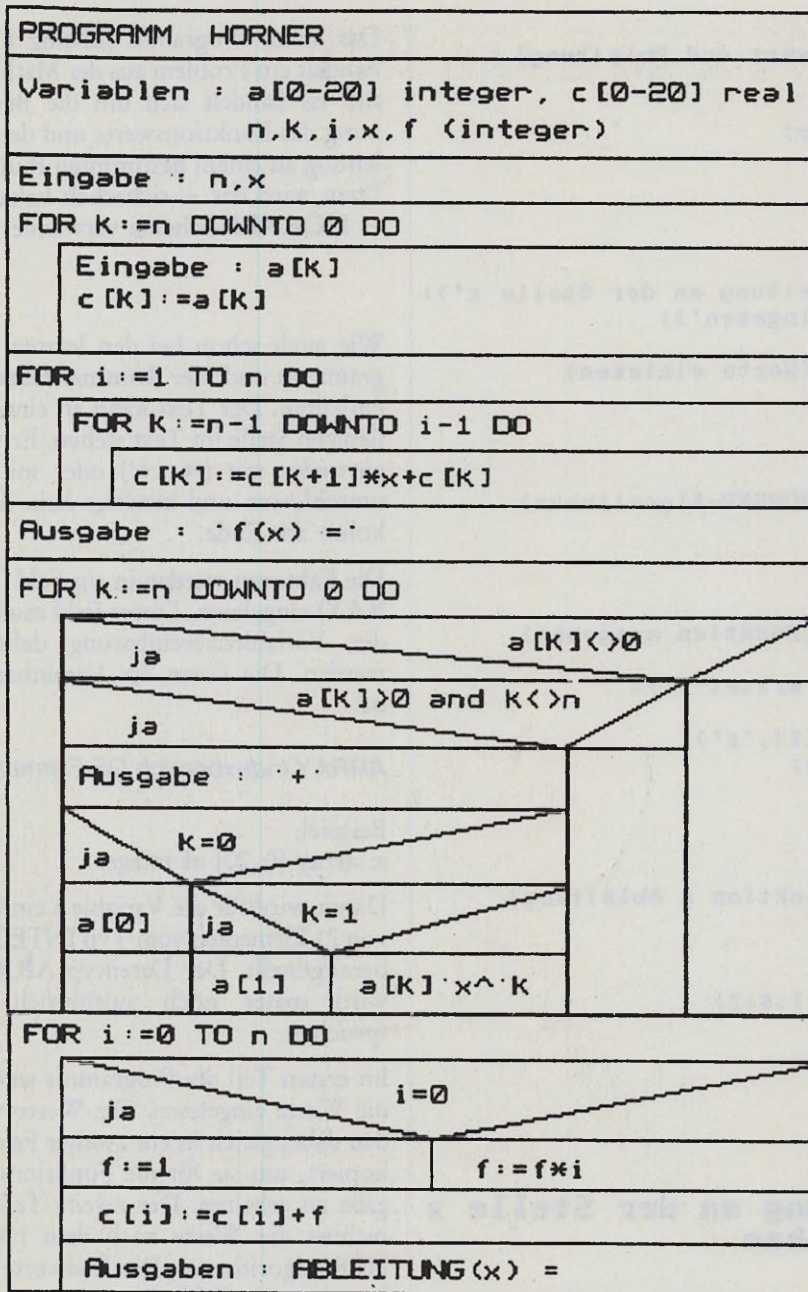
a: array [0..20] of integer

Damit wird für die Variable a ein Feld von 21 Elementen vom Typ INTEGER bereitgestellt. Der Datentyp ARRAY wird später noch ausführlich besprochen.

Im ersten Teil des Programms werden die Werte eingelesen. Die Werte werden dabei gleich in ein zweites Feld c[] kopiert, um sie für die Funktionsausgabe zu erhalten. Der zweite Teil berechnet die Werte nach dem HORNER-Algorithmus. Die Endwerte stehen dann im Feld c[].

Im anschließenden Teil wird die Funktion in einer übersichtlichen Form ausgegeben (siehe Bild2). Dabei werden die Nullterme gestrichen, und bei Bedarf wird ein Additionszeichen eingesetzt.

Der letzte Teil gibt die Werte der Ableitungen an der Stelle x aus. Dazu müssen die im Feld c[] befindlichen Werte noch mit der entsprechenden Fakultät multipliziert werden. Die Ausgabe der Werte erfolgt formatiert.



Struktogramm 6

Hinweis: Sie sollten unbedingt alle Beispiele dieses Kurses eingeben und dann damit experimentieren. Probieren Sie alle Werte aus, die Ihnen einfallen, und suchen Sie Schwächen des

Programms. Außerdem können Sie das Programm nach Ihren Wünschen ändern und erweitern. Wenn Sie Kritik oder Anregungen zu diesem Kurs haben, dann schreiben Sie mir.

Im nächsten Teil geht es weiter mit **Prozeduren und Funktionen**. Diese Strukturen unterstützen die übersichtliche Gestaltung von Programmen. Sie ermöglichen außerdem das Programmieren von Modulen und somit eine schrittweise Verfeinerung des zu lösenden Problems.

Ein Blick in die Geschichte

Wer war Pascal?



Sir Francis Pascal in Jugendjahren

Sir Francis Pascal, ein englischer Seeräuber, befaßte sich meistens mit der Berechnung nautischer Meilen und dem Zählen von erbeuteten Reichtümern. Mit seinem Schiff, der 'Algorithmika', durchsegelte er alle 17 Meere. Sein Unwesen trieb er im berühmten Pascal'schen Dreieck, das auch heute noch den Fachleuten Rätsel aufgibt. Durch ihn wurde die bekannte Sage geboren - 'Verschollen in Pascal-Dreieck'. Sein größter Gegner war die Armiga, die königliche Seeflotte, welche ihm große Schwierigkeiten bereitete, da sie mehrere Kampfhandlungen gleichzeitig tätigte (Multifighting). Aber Sir Francis Pascal, ein erfahrener Stratege, hatte immer ein Wundermittel aus dem Tisch zu ziehen, seine Desk-Accessoires. Sein Unwesen fand am Galgen ein Ende. Er wurde am 1.4.1636 mit einer Leerschleife um den Hals aufgehängt.



PHILGERMA GmbH, Ungererstr. 42, 8000 München 40, 089 / 395551 Info anfordern

Qualitätssoftware von METACOMCO jeweils mit Editor, Linker und Handbuch

C-Compiler LATTICE-C Fließkomma Arithmetik, Makros 380,- DM

MCC Pascal Compiler voller ISO 7185-Standard 340,- DM

MACRO-Assembler 198,- DM

FORTRAN 77-Compiler Prospero z. B. doppelte Genauigkeit 64 Bit 490,- DM

K-SPREAD Tabellenkalkulation in deutsch, 256 SP*8192 Reihen 198,- DM

RAM Disk + Print Spooler Talent 78,- DM

BRATACCAS Grafik Adventure MISSION MOUSE Grafik Adventure ZKUL & WEST Text Adventure 118,- DM 78,- DM 78,- DM

Diskettenlaufwerk 3'5 720 KB 698,- DM

Doppelauflaufwerk 3'5 2" 720 KB 1098,- DM

Diese CUMANA-Disklaufwerke sind auch in 5 1/4 Zoll erhältlich

Kabel ST an Drucker 46,- DM

Kabel ST an RS 232 69,- DM

Akustikkoppl. DATAPHONS21d + Kab. 340,- DM

10 Disketten 3'5 2DD Wabash 98,- DM

Bewegungseindrücke

AES verfügt über Routinen, die den Eindruck vermitteln, es bewege sich ein Rechteck von einem Punkt zum anderen oder wachse aus dem Bildschirm auf seine volle Größe heraus. Anwendungen finden diese Routinen beispielsweise beim Aufbau eines Fensters (z. B. Directory anzeigen) oder beim Einblenden einer Meldung (z. B.

Desk-Info).

Auch in Basic sind diese Routinen ansprechbar und helfen so, manches Programm grafisch interessanter zu gestalten.

Im AES existieren drei verschiedene Routinen.

- Bewegung eines sich vergrößernden Rechteckes (GROW_BOX)
- Bewegung eines sich verkleinernden Rechteckes (SHRINK_BOX)
- Bewegung eines gleichgroßen Rechteckes (MOVE_BOX)

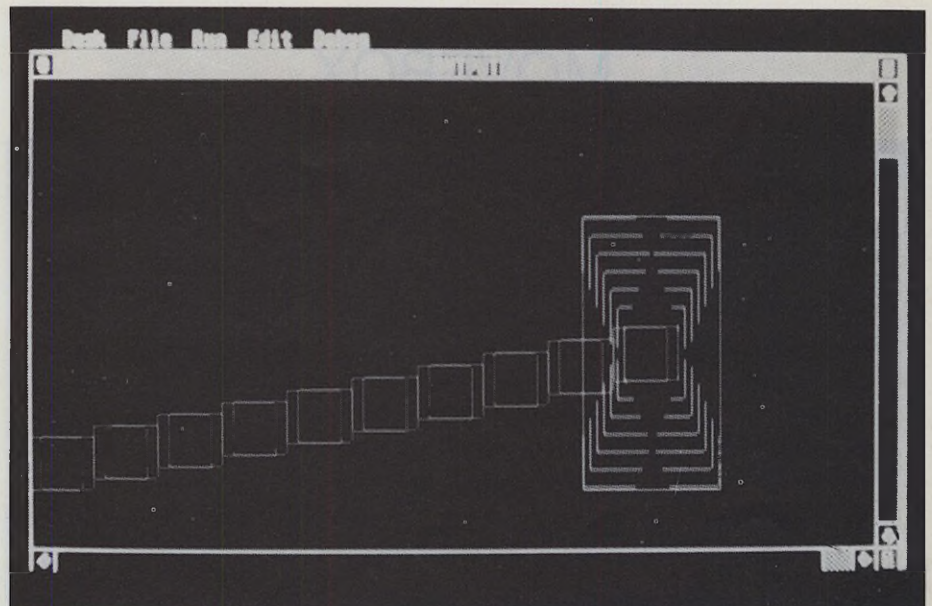
GROW BOX & SHRINK BOX

Mit dieser Routine kann der Bewegungsablauf eines Rechteckes abgebildet werden, das sich von einem Punkt des Bildschirms auf einen anderen Punkt zubewegt, um sich dort auf eine vorgewählte Größe auszudehnen. Die Routine benötigt als Parameter den oberen linken Eckpunkt des Startrechteckes und dessen Ausdehnung, ebenso die Werte für das Zielrechteck, wobei diese sich unterscheiden können. Das Rechteck beginnt nun am Startpunkt (*XPOS1; YPOS1*) mit der Anfangsgröße (*BREIT1; HOCH1*) seine Bewegung bis zum Zielpunkt. Anschließend vergrößert es sich auf die Endmaße (*BREIT2; HOCH2*). Diese Routine bildet ausschließlich den Bewegungsablauf ab; das entstehende Rechteck muß durch eine andere Routine gezeichnet werden. Dies könnte mit vier LINEF-Befehlen oder besser mit den dafür vorgesehenen VDI-Rechteckroutinen erreicht werden (siehe ST 2/86).

Der Aufruf von 'GROWBOX' erfolgt durch:

```
XALT = .. : YALT = ...
XNEU = .. : YNEU = ...
BREIT1 = .. : HOCH1 = ...
BREIT2 = .. : HOCH2 = ...
Gosub GROWBOX
```

Mit Hilfe dieser und der folgenden Routine werden sehr schöne grafische Effekte erzielt, beispielsweise zum Einblenden von Informationen oder eines Auswahlmenüs. Damit die Hin-



```
5      ' merge "INIT.bas"
6      ' merge "GROWBOX.bas"
7      ' merge "RECHTECK.bas"
9
10     gosub init
20     fullw 2:clearw 2
30     xalt=0 :yalt =300 :breit1=40 :hoch1=40
40     xneu =400:yneu =100:breit2=100:hoch2=200
50     start:
60     xpos1=xneu :ypos1=yneu :xpos2=xneu +breit2
70     ypos2=yneu +hoch2
80     color 1,1,0,0,0
90     gosub growbox :gosub rechteck
100    for a= 0 to 1000 : next
110    color 1,0
120    gosub rechteck :gosub schrumpfbbox
130    for a= 0 to 1000 : next
140    goto start
```

Listing 7: wachsendes/schrumpfendes Rechteck

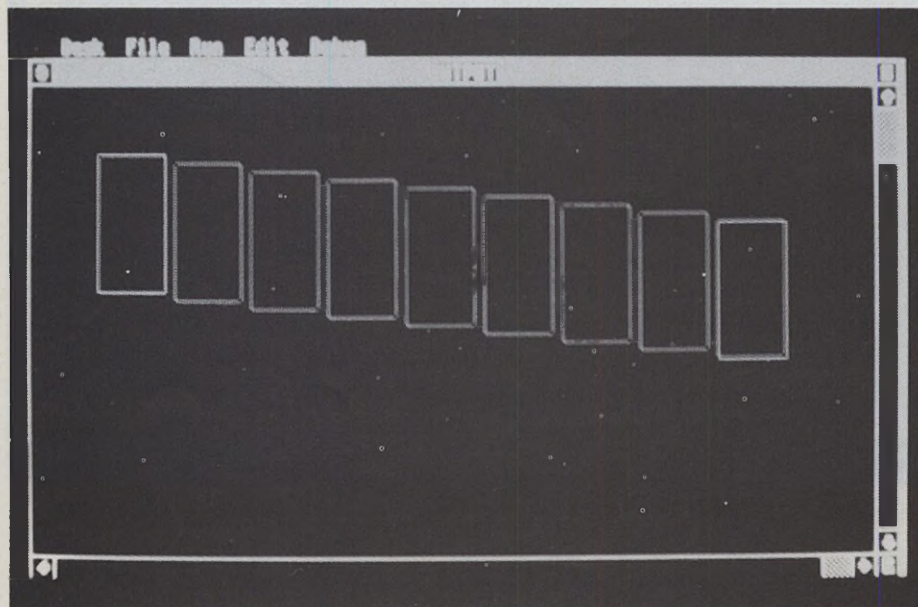
```

62100 SCHRUMPFBOX:
62102 ' --> XALT; YALT; BREIT1; HOCH1
62104 ' --> XNEU; YNEU; BREIT2; HOCH2
62106 schrumpf% = 1
62108 '
62110 GROWBOX:
62112 ' --> siehe Schrumpfbox
62114 '
62116 poke control+2,8
62118 poke control+4,1
62120 poke gintin, xalt
62122 poke gintin+2, yalt
62124 poke gintin+4, breit1
62126 poke gintin+6, hoch1
62128 poke gintin+8, xneu +1
62130 poke gintin+10, yneu +38
62132 poke gintin+12, breit2
62134 poke gintin+14, hoch2
62136 gemsys 73 + schrumpf%
62138 schrumpf% = 0
62140 return
62142 '-----

```

Routine GROWBOX

MOVEBOX



```

62150 MOVEBOX:
62152 ' --> BREIT; HOCH; XALT; YALT
62154 ' --> XNEU; YNEU
62156 poke control+2,6
62158 poke control+4,1
62160 poke gintin,breit
62162 poke gintin+2,hoch
62164 poke gintin+4,xalt +1
62166 poke gintin+6,yalt +38
62168 poke gintin+8,xneu +1
62170 poke gintin+10,yneu +38
62172 gemsys 72
62174 return
62176 '-----

```

Routine MOVEBOX

tergrundinformation nicht verloren geht, sollte man hierbei mit dem 'XOR-Grafikmodus' arbeiten (siehe ST-Computer 2/86).

SHRINX BOX ist die Umkehrfunktion der GROWBOX-Routine. Sie wird mit den gleichen Parametern angesprochen wie die 'GROWBOX' Routine. Der Bewegungsablauf ist genau entgegengesetzt, das Rechteck verkleinert sich und bewegt sich anschließend zum Zielpunkt auf den ehemaligen Startpunkt zurück, wo es verschwindet. Zum Einsatz kommt diese Routine beispielsweise beim Schließen eines Fensters (Löschen Directoryfenster). Falls die Parameter nicht verändert wurden, erfolgt der Aufruf durch:

Gosub SCHRUMPFBOX

andernfalls müssen die acht Parameter neu definiert werden (siehe GROWBOX).

MOVEBOX

Die letzte Routine dieser Gruppe ist 'MOVEBOX'. Sie bewegt ein Rechteck mit gleichbleibender Größe von einem Punkt zu einem anderen. Wie die vorherigen Routinen sorgt auch diese nur für den Bewegungsablauf. Die Rechtecke an sich bzw. der Inhalt dieses Bereiches müssen extra gezeichnet werden.

Der Aufruf dieses Unterprogrammes erfolgt durch:

```

XALT = .. : YALT = .. :
XNEU = .. : YNEU = .. :
BREIT = .. : HOCH = .. :
Gosub MOVEBOX

```

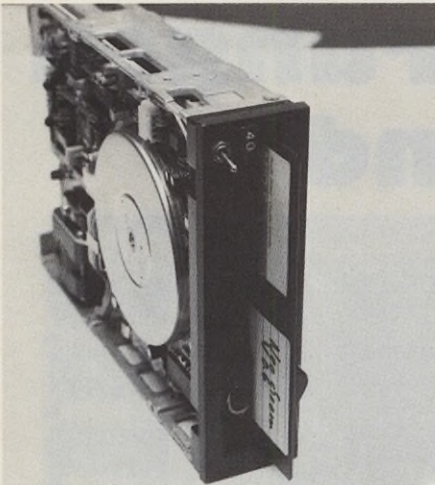
Fortsetzung folgt (HS)

```

5 'merge "INIT.bas"
6 'merge "MOVEBOX.bas"
7 'merge "RECHTECK.bas"
9 '
10 gosub init
20 fullw 2 : clearw 2
30 xalt =50 :yalt =50:breit =50:hoch =100
40 xneu =500:yneu =100
50 start:
60 gosub movebox
70 color 1,1,1,0,0
80 xpos1=xneu :ypos1=yneu :xpos2=xneu +breit
90 ypos2=yneu +hoch : gosub rechteck
100 swap xalt,xneu : swap yalt,yneu
110 for a=0 to 1000: next
120 color 1,0,0 : gosub rechteck
130 goto start
140 end

```

Listing 8: bewegtes Rechteck



Angeregt durch die viele Leserpost wollen wir hier eine Erweiterung zu unserem Artikel „5 1/4-Zoll Floppy am ST“ der Januarausgabe bringen.

Das TEAC Laufwerk FD-55F erfreut sich großer Beliebtheit, nicht zuletzt, weil es ein zuverlässiges Laufwerk ist, sondern auch, weil es normalerweise bei Apple Computern Verwendung findet und zudem recht preiswert ist. Der eigentliche Anschluß dieses Laufwerks kann genauso erfolgen wie in der Januarausgabe beschrieben wurde, oder noch einfacher über ein 34-poliges Flachbandkabel in Verbindung mit unserer Floppy-Stecker-Platine (FSP) der Märzangabe. Dazu muß an der FSP eine Kante etwas abgefeilt bzw. mit einem Seitenschneider abgetrennt werden. Jetzt paßt unsere Platine in die Buchse der ATARI Floppy, an der sonst eine zweite Floppy angeschlossen wird (I/O Connecotr; Out). Von der Platine aus kann nun über eine gewöhnliche 1:1 Quetschverbindung (34-polig) das TEAC Laufwerk angeschlossen werden. Dabei ist zu beachten, daß bei dieser Verbindung der Jumper für das Drive Select Signal im Fremdlaufwerk immer auf Drive Select 0 (DS0) gesetzt werden muß. Beim TEAC FD-55F sind folgende Jumper zu setzen: DSO, HS, UR und SM. Außerdem sind die in einem Widerstands-Array zusammengefaßten Pull-up Widerstände aus der Fassung zu nehmen. Ein Widerstand (R12), der sich neben dem Array befindet (siehe Bild 2 Pfeil), ist unbedingt von 330 Ohm auf 2 200 Ohm zu erhöhen, nur so ist eine einwandfreie Funktion gewährleistet. Alles Gesagte ist auch auf den Bildern zu erkennen.

Anschluß eines TEAC Laufwerks FD-55 F

Zum Schluß noch ein Tip: um das Laufwerk von 80 auf 40 Tracks umschalten zu können, ist ein Widerstand von 10 Ohm, über einen Schalter, für R15 einzubauen. Nach jedem Umschalten der Spure ist die Stromversorgung der Floppy einmal ein/aus zu schalten, um diese neu zu initialisieren. Das Netzteil der ATARI Floppies

ist übrigens stark genug, um auch noch ein TEAC Laufwerk mit den benötigten Spannungen (+5 und +12 Volt) zu versorgen. Wenn Sie Erfahrungen mit anderen Laufwerken gemacht haben, dann dürfen Sie uns das gerne mitteilen, wir freuen uns sehr darüber.

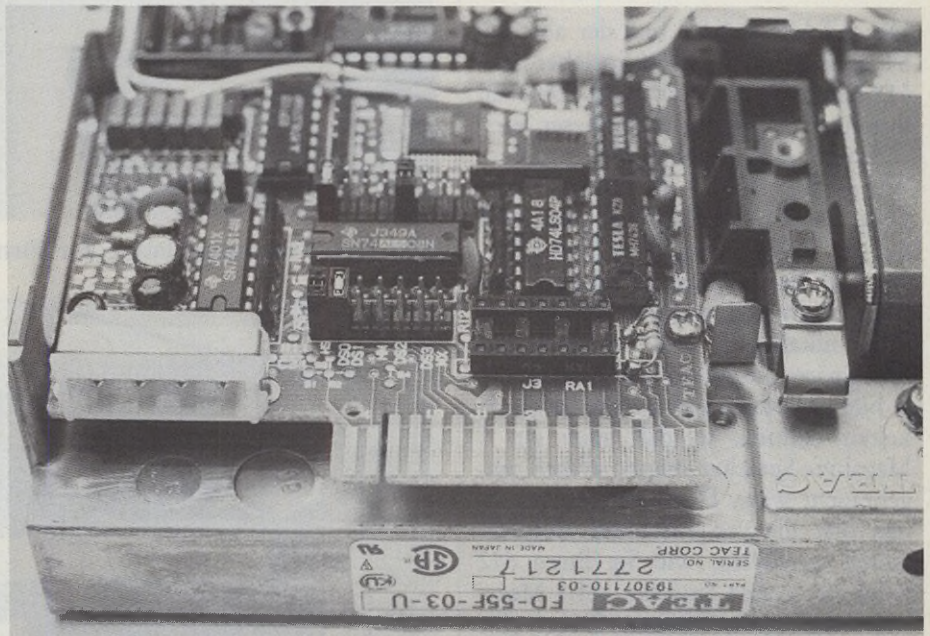


Bild 1: TEAC FD-55 F

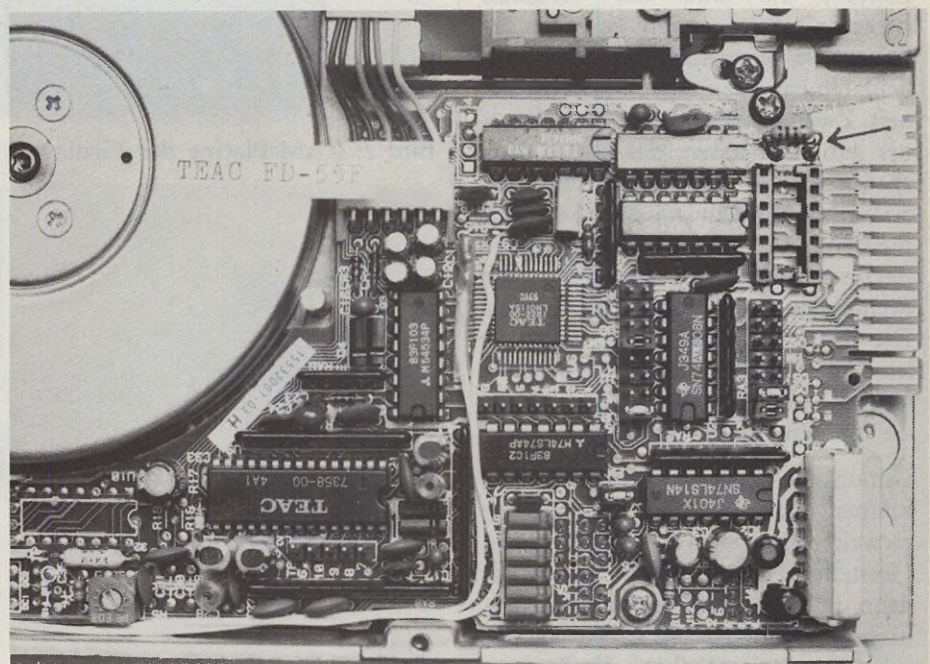


Bild 2: TEAC FD-55 F

Speichererweiterungen auf dem Prüfstand

Drei Systeme werden ausführlich getestet und miteinander verglichen.

Der ATARI 260 ST und der alte 520 ST stellen einen RAM-Speicher von 512 Kilobyte zur Verfügung. Obwohl dies nicht wenig ist, kann dieser Speicherraum manchmal zu klein werden. Wir denken dabei z. B. an eine RAM-Disk. Denn beim Kopieren oder auch bei der Verwendung eines Compilers, wie z. B. den ST Pascal Compiler von CCD, stellt eine solche RAM-Disk ein sehr nützliches und zeitsparendes 'imaginäres Laufwerk' dar. Man könnte hier natürlich noch andere Möglichkeiten nennen, so daß auch ein 1 Megabyte-Speicher relativ schnell aufgebraucht wäre. Doch wir wollen Ihnen hier die verschiedenen Möglichkeiten vorstellen, wie Sie Ihren 512 Kilobyte großen Speicher auf 1 Megabyte verdoppeln können. Dabei gibt es drei verschiedene Wege, dies zu tun.

Erstens, Sie sind ein versierter Elektronik-Bastler, dann ist es am billigsten, Sie verwenden unsere ausführliche Umbauanleitung aus dem Januarheft. Zweitens können Sie Ihren Rechner aufrüsten lassen, das kostet dann aber zwischen DM 300,- und DM 500,-. Außerdem müssen Sie dann meistens ein paar Tage lang Ihren Rechner entbehren, abgesehen von eventuellen Porto- bzw. Fahrtkosten. Die dritte Möglichkeit ist diese: Sie lesen die folgenden Seiten und bestellen sich eine RAM-Erweiterungsplatine, die Sie dann selbst einbauen müssen.

Um uns ein objektives Bild machen zu können, haben wir uns drei Speichererweiterungskarten verschiedener Hersteller besorgt. Diese haben wir dann in drei verschiedene Rechner, nach der mitgelieferten Anleitung, eingebaut und anschließend alle Rechner einem gründlichen Test unterzogen.

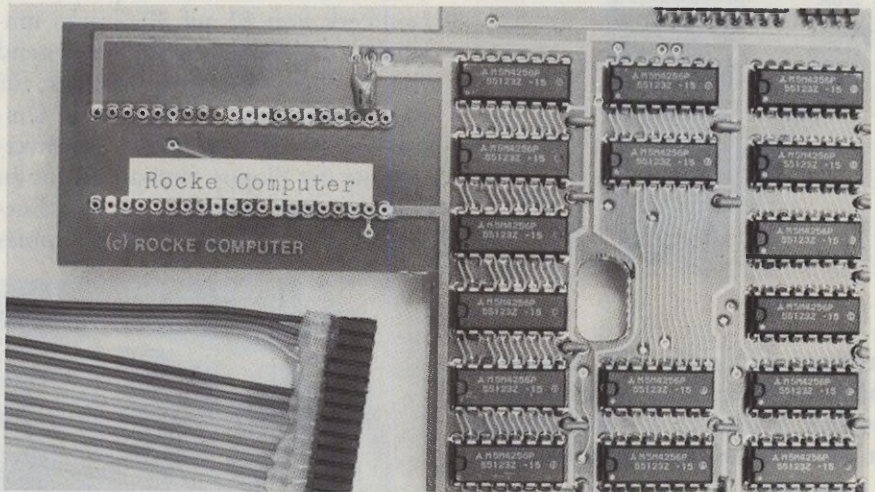


Bild 1: Erweiterungsplatine der Firma R. Roche Computer

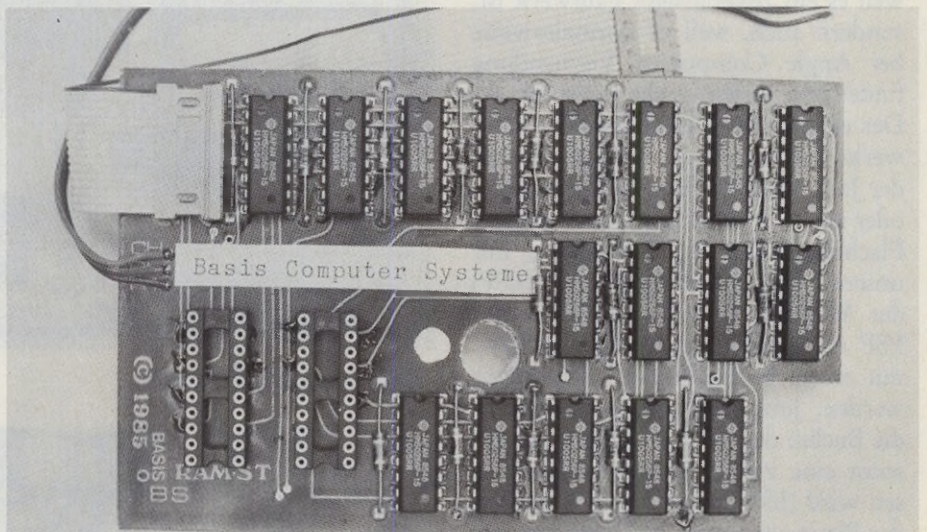


Bild 2: RAM-Platine der Firma Basic Computer

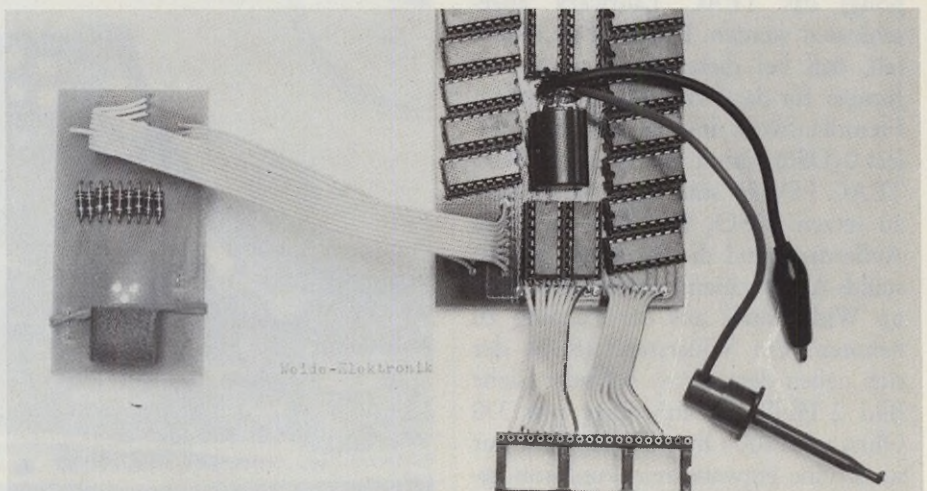


Bild 3: RAM-Platine der Firma Weide Elektronik

Bevor wir die einzelnen Systeme beschreiben, dürfen wir schon verraten, daß alle Erweiterungen auf Anhieb zuverlässig funktionierten, jedoch wird für den Einbau zum Teil erhebliches technisches Fingerspitzengefühl vorausgesetzt. Der Preis für diese Art der RAM-Erweiterung schwankt je nach System von DM 275,- bis DM 349,- incl. MwSt.

Als erstes möchten wir nun das System der Firma Rolf Rocke Computer, Austraße 1, 5090 Leverkusen 3, beschreiben. Die RAM-Erweiterung dieser Firma besteht aus einer Platine mit 16 RAM-Chips und einem Sockel (Adapter) für den Videoshifter. Die Platine ist komplett aufgebaut und funktionsgetestet. Ein für den Einbau notwendiges Kabel sowie 7 Seiten Einbauanleitung (DIN A4) werden mitgeliefert. Für den Einbau ist es erforderlich, die Abschirmung des Videoshifters auszulöten, um an der linken Stirnseite des Rahmens eine Aussparung für die Platine auszusägen bzw. mit einer Blechschere auszuschnitten (siehe Bild 4 Pfeile). Da dieser Blechrahmen an 8 Stellen mit der Rechnerplatine verlötet ist, kann das Auslöten der Abschirmung dem ungeübten Bastler Schwierigkeiten bereiten. Die Platine wird dann in den Sockel des Videoshifters eingesetzt und dieser wiederum auf die Erweiterungsplatine. Sodann sind noch 9 Leitungen auf der Bestückungsseite in die Nähe der MMU (U15) und 4 weitere von unten an die MMU zu löten. Beim Zusammenbau des Rechners können alle Abschirmbleche wieder montiert werden. Für das Aufrüsten benötigten wir (eine Person) ca. 60 Minuten. Diese Erweiterung kostet DM 349,- und ist somit das teuerste der von uns getesteten Systeme. Auf Wunsch führt die Firma den Einbau in ihrer eigenen Werkstatt durch. Aufpreis DM 45,- bei freier Anlieferung zuzüglich Versandkosten. Ferner wird noch eine andere Version dieser Platine angeboten, bei der alle Chips mit Sockeln versehen sind. Diese Sonderausführung der Erweiterungskarte ist besonders für Test- und Prüfzwecke geeignet. Da beim ST 520+ die Fehlerquote, bedingt durch fehlerhafte Lötstellen der aufgesetzten 2. RAM-Bank, relativ hoch ist, können defekte RAM-Chips oder schlechte Lötstellen durch probeweises, gezieltes

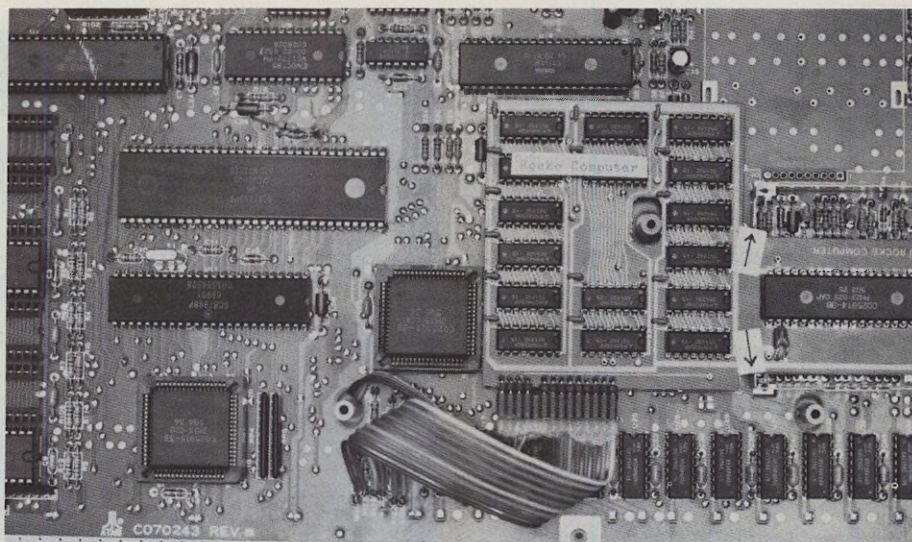


Bild 4: Eingebaute RAM Platine von R. Rocke Computer

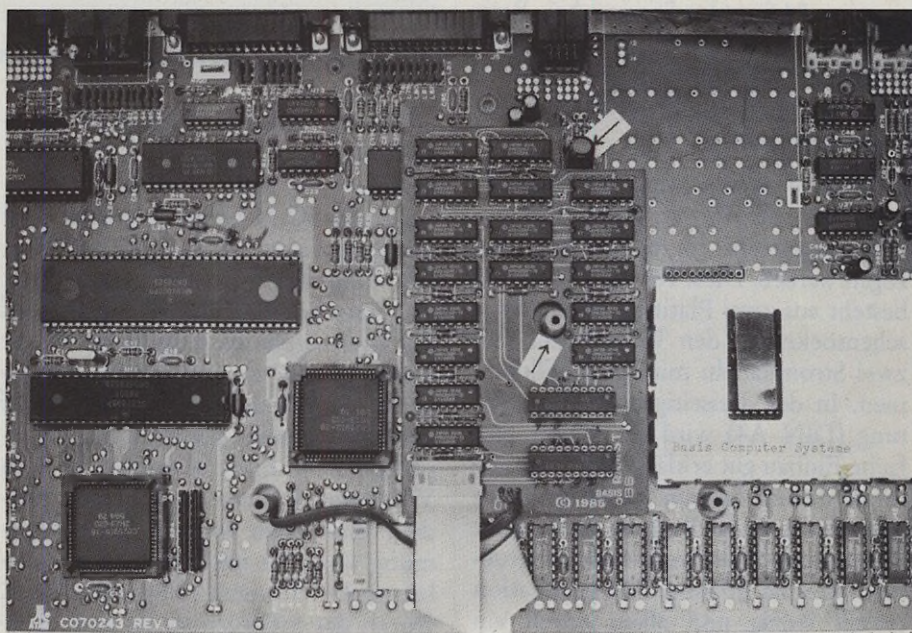


Bild 5: Eingebaute RAM Platine von Basis Computer

Austauschen leicht lokalisiert werden.

Eine weitere RAM-Erweiterung für den ST 260 bzw. ST 520 wird von Basis Computer Systeme GmbH, Eulerstraße 15, 4400 Münster, geliefert. Auch hier wird eine komplett mit 17 RAM-Chips bestückte und geprüfte Platine geliefert, jedoch unterscheidet sich der Einbau erheblich von den anderen Systemen. Für den Einbau werden technisches Verständnis und Kenntnisse des ATARI-Systems vorausgesetzt, so wundert man sich auch nicht über die eine Seite (DIN A4) umfassende Einbauanleitung. Es wird darin z. B. nicht gezeigt, wo sich an der MMU die Pins 18, 21 und 22 befinden,

an die je eine Leitung angelötet werden muß. Aber für solche Zwecke gibt es ja die Januarausgabe der ST. Zum Einbau der Platine müssen die beiden Treiber-IC's (74LS244 U26 und U27) sowie ein RAM-Chip (U16) aus der Rechnerplatine ausgelötet werden. Die beiden ausgelöteten Treiber-IC's werden dann in je eine Fassung auf der Erweiterungsplatine gesteckt. Die RAM-Platine selbst wird dann, über 40 lange Sockelstifte auf der Rechnerplatine, an die Stelle der Treiber-IC's, festgelötet. Dabei stellten wir fest, daß die Platine etwas zu groß ist und ein Kondensator (siehe Bild 5 Pfeil) ein wenig versetzt werden mußte. Ferner ist das Loch in der Mitte der Platine, durch das die

Gehäusehälften verschraubt werden, zu klein (siehe Pfeil). Es mußte vor der Montage der Platine vergrößert (aufgefeilt) werden. An die Stelle des RAMs wird ein Stecker mit angequetschtem Flachbandkabel eingelötet. Die Erweiterungsplatine ist somit fest in den Rechner eingebaut, was bei einer eventuellen Reparatur von Bauteilen, die sich unter der Platine befinden, sehr störend sein kann. Für den Einbau benötigte eine Person ca. 150 Minuten. Diese relativ lange Zeit ist vor allem auf das Auslöten von drei IC's zurückzuführen, was ohnehin nur mit entsprechendem Werkzeug möglich ist. Die Erweiterung kostet für diejenigen, die mit einem Lötkolben gut umgehen können, DM 302,-. Wenn Ihr Rechner im 24-Stunden-Service bei Basis Computer umgerüstet werden soll, kostet die Erweiterung DM 346,-. Die Firma gewährleistet eine Garantie von einem Jahr.

Das dritte System, das wir testeten, kann von der Firma Weide Elektronik, Regerstraße 34, 4010 Hilden, bezogen werden. Diese RAM-Erweiterung besteht aus zwei Platinen, einem Zwischensockel für den Videoshifter und zwei Stromkabeln mit Krokodilklemmen. In der dreiseitigen Einbauanleitung (DIN A4) wird der relativ einfache Einbau gut erklärt. Das Besondere an diesem System ist, es ist nicht notwendig, an der Rechnerplatine oder irgendwelchen elektronischen Baugruppen löten zu müssen. Die Installation erfolgt einzig und allein durch ein Stecksystem. Es ist allerdings möglich, daß das Abschirmblech der gesamten Rechnerplatine an zwei Stellen miteinander verlötet ist. Diese Verbindung

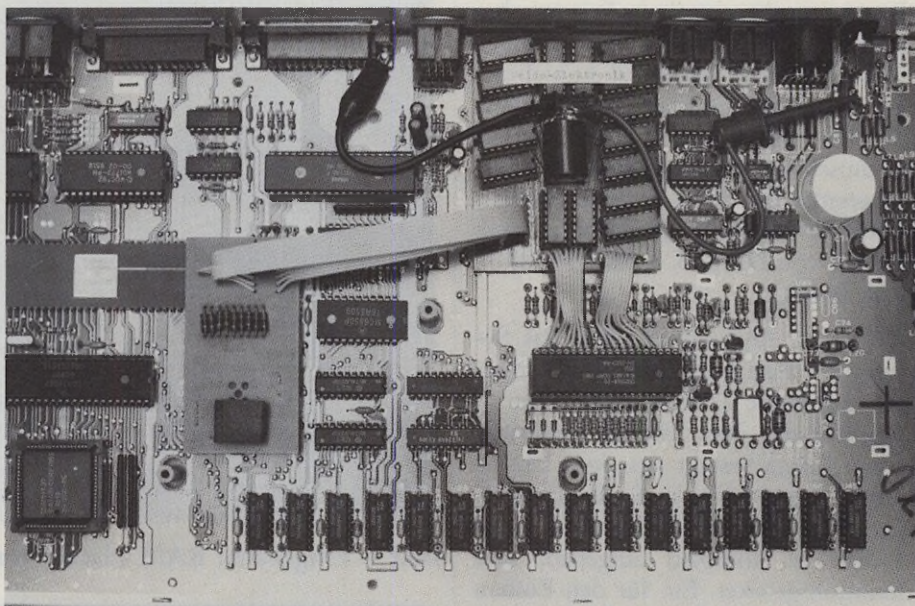


Bild 6: Eingebaute RAM-Erweiterung von Weide Elektronik

muß entfernt werden, damit der Rechner ausgebaut werden kann. Bemerkenswert ist, daß bei dieser Erweiterung alle RAM-Chips gesockelt sind und das bei einem Preis von DM 275,-. Die komplett bestückte Platine mit den Speicher-Chips wird einfach auf der Rechnerplatine aufgeklebt, an der früher einmal der HF-Modulator geplant war. Die andere Platine wird auf den MMU-Sockel gesteckt. Die Zuleitungen zum Zwischensockel für den Videochip, können auch verlegt werden, ohne einen Spalt in die Abschirmung sägen zu müssen. Für den Einbau sind 45 Minuten völlig ausreichend. Durch das Stecksystem ist eine eventuell nötige Reparatur des Computers problemlos möglich. Die Redaktion würde allerdings empfehlen, die Krokodilklemmen der beiden Stromkabel

durch Lötungen zu ersetzen (siehe Bild), um einen sicheren und störungsfreien Kontakt zu bieten. Bei Schwierigkeiten bietet die Firma einen telefonischen Kundendienst an. Dieses System arbeitete jedoch, wie alle anderen auch, sofort einwandfrei. Es scheint uns die einfachste, schnellste und preiswerteste Möglichkeit, seinen Rechner auf 1 Megabyte Speicherkapazität aufzurüsten.

(UB)

CompWare CompWare

Robert Bunsen Str. 8, 6084 Gernsheim Tel. 0 62 58 / 5 16 16
Ernst Ludwig Str. 7, 6840 Lampertheim Tel. 0 62 06 / 5 48 88

CompWare CompWare

	10 St.	50 St.	100 St.	> 100 St.		
3 1/2" Fuji MF 1 DD	7,90	7,20	6,90	auf Anfrage	Orion Farbmonitor CCM 14 mit Kabel an Atari 520/260	775,-
3 1/2" Disketten MF 2 DD	9,90	9,50	9,20	auf Anfrage	Panasonic Drucker KX-P1091, 120 z/sec, diverse Schriftarten	1 075,-
					Panasonic Drucker KX-P1092, 180 z/sec, diverse Schriftarten	1 375,-

Händleranfragen erwünscht

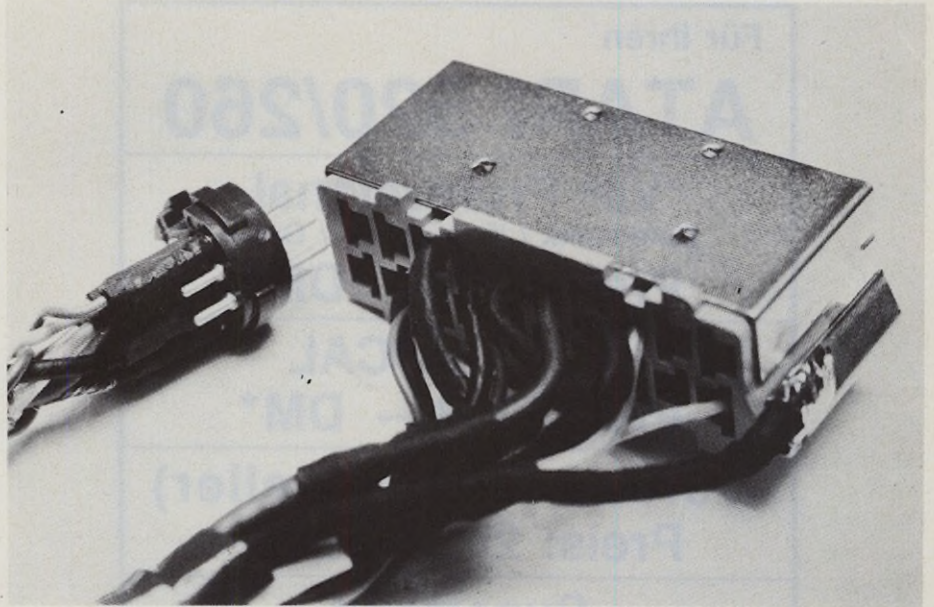
>>>> **Wir tun alles damit Sie nicht bei der Konkurrenz kaufen** <<<<

Tips zum Monitor

SCART-Kabel selber gebaut

Wer einen Fernseher mit der sogenannten SCART- oder auch EURO-Buchse besitzt, kann diesen unter gewissen Voraussetzungen als Farbmonitor für den ST verwenden. Für die Besitzer eines ST 260 oder ST 520+ ist dazu nur ein entsprechendes Kabel (siehe Tabelle) notwendig. Wer jedoch den 'alten' ST 520 besitzt, muß sich noch seine Monitorbuchse anpassen (lassen). Da bei diesem Modell das Composite Synchron Signal (Pin 2) fehlt und außerdem der Pin 8 der Monitorbuchse mit Masse (0 Volt), anstatt über einen Schutzwiderstand von 1200 Ohm mit +12 Volt verbunden ist, muß der Rechner mit der kleinen Schaltung aus dem Januarheft (Seite 6) erweitert werden. Aus diesem Grund läuft der Farbmonitor „Thomson for ATARI“ auch **nicht** mit dem ST 520 zusammen. Zur Herstellung des SCART-Kabels benötigen Sie einen 13 poligen Monitorstecker für den Rechner, einen SCART-Stecker und mindestens ein 7 adriges abgeschirmtes Kabel, das Sie nach folgender Tabelle verdrahten müssen.

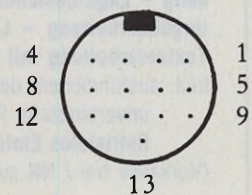
Optimal ist ein Verbindungskabel, wenn das Ton Signal über eine eigene Abschirmung verfügt, und diese dann an Pin 4 des SCART-Steckers gelötet wird. Ebenso sollten die drei Farbsignale je eine eigene Abschirmung besitzen, welche dann entsprechend an die Pins 9,13 und 5 des Scart-Steckers zu



löten ist. Für die drei Farbsignale sollte dann aber sogenanntes 75-Ohm-Kabel verwendet werden. Schließlich ist die Abschirmung **aller** Leitungen nur mit dem Steckergehäuse des SCART-Steckers und am anderen Ende mit dem Metallgehäuse des Monitorsteckers zu verbinden. Ein solches, aus technischer Sicht optimales Kabel ist auf dem Bild zu sehen. Wer aber kein meterlanges Monitorkabel benötigt, der kann sich die Abschirmung der einzelnen Farbsignale und somit das schwer erhältliche 75-Ohm-Kabel sparen. Einen Monitorstecker für den ST gibt es mittlerweile bei einigen ATARI-Fachhändlern für ca. DM 20,- zu kaufen. Wem dies zu teuer ist der muß sich einen, wie im Januarheft beschrieben wurde, selber bauen. Zum Schluß noch ein Wort zu Ihrem Farbfernseher; besitzen Sie ein sehr modernes Gerät oder eines mit digitaler Bild-

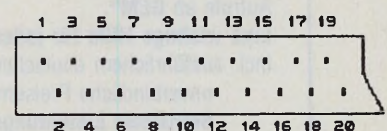
blenkung, kann es mit dem Bildfang Probleme geben. Dies liegt an der Tatsache, daß der Rechner normalerweise einen Synchronimpuls von 60 Hz liefert und die Empfänger einen von 50 Hz erwarten. Läßt sich Ihr Bild mit dem Bildfangregler nicht anhalten, so müssen Sie ein kleines Programm haben, daß den Rechner auf 50 Hz umschaltet. Ein solches Programm bieten wir in unserem Public Domain Service auf Diskette 2 an.

Monitorbuchse



von hinten (Loetseite)

SCART-Stecker



von hinten (Loetseite)

SCART-Stecker Fernseher	Monitorstecker ST	Signal
6	1	Ton
20	2	Composite Synchron
11	6	Grün
15	7	Rot
8	8	Schaltspannung +12 V
7	10	Blau
16	12	Vertikal Synchron
4,5,9,13	13	Masse

Für Ihren

ATARI 520/260

VIP Professional
Datenbank / Kalkulation / Grafik
Preis: 748,- DM*

GEM-PASCAL
Preis: 248,- DM*

Wörterbuch (Speller)
Preis: 248,- DM*

Superfond
Preis: 248,- DM*

Bei Ihrem Händler oder
Lieferung frei Haus durch uns
* unverbindlich empfohlener
Verkaufspreis inkl. MwSt.

Computer-Technik Kiechbusch
Am Seeufer 22 5412 Ransbach
Telefon 02623-1618

ST-QUALITÄTSSOFTWARE AUS BAYERN

● BS-HANDEL ●

Integriertes Software für den klein- und mittelständischen Handel. Benutzerführung unter GEM® – einheitliche Datenführung für alle Programmteile – aufwärtskompatibel für zukünftige Erweiterungen. Interessenten-, Kunden-, Lieferanten- und Personaldatenerfassung – Lagerbestandsverwaltung – Angebotserstellung – Auftragsbearbeitung – Lieferschein – Rechnung – Mahnung – Textverarbeitung mit Serienbrief – u.v.m.

Incl. ausführlichem deutschen Handbuch!

unverbindliche Preisempfehlung: DM 949,00 inkl. MwSt.

Befristetes Einführungsangebot: DM 899,00 inkl. MwSt.

(Vorkasse frei / NN zugl. P & V)

● BS – C-TOOL für GEM® ●

Nützliche Funktionen für den C-Programmierer im Umgang mit den hervorragenden Eigenschaften von GEM®, wie Pulldown-Menüs, Formularen, usw. Aus vielen komplizierten Funktionsaufrufen, um derlei Dinge zu aktivieren, werden wenige, einfach überschaubare Aufrufe an GEM®.

Eine wichtige Hilfe für jeden C-Programmierer!

Incl. ausführlichem deutschen Handbuch!

unverbindliche Preisempfehlung: DM 129,50 inkl. MwSt.

Befristetes Einführungsangebot: DM 112,00 inkl. MwSt.

(Vorkasse frei / NN zugl. P & V)

Bavaria-soft



Nachtrag zu unserem Bericht

„Zwei 3 1/2-Zoll Fremdlaufwerke am ST“
der Märzausgabe

Um unnötigen Leseranfragen zuvorzukommen, teilen wir Ihnen mit, daß das Bestückungsbild auf Seite 7 der Treiber- und Netzteilplatine einen kleinen Schönheitsfehler aufweist. Bei genauem Hinsehen ist auf dem Foto für IC3 ein 7406 zu erkennen, richtig ist jedoch, wie auch im Schaltbild und in der Stückliste angegeben, ein 7407.

Inzwischen hatten wir die Möglichkeit zwei Slimline Laufwerke von TEAC des Typs FD-135 F anzuschließen und auszuprobieren. Dabei zeigte sich, daß die Geschwindigkeit des Steppmotors dieser Laufwerke zu langsam ist. Sie beträgt bei diesem Modell 6 Millisekunden, der Floppycontroller im Rechner wird jedoch auf 3 Millisekunden initialisiert. Durch ändern einer Systemvariablen in der Adresse 1088 (\$440) ist es möglich, den Floppycontroller (WD 1772) mit einer anderen Steppgeschwindigkeit zu programmieren. Ändert man den an dieser Adresse stehenden Wert von 3 auf 6 um, so ist auch das FD-135 F Laufwerk am ATARI ST lauffähig. Es ist jedoch fraglich, wie man diese Speicherzelle ändern kann, wenn man nur zwei Fremdlaufwerke dieses Typs zur Verfügung hat, da das TOS nicht einmal geladen werden kann. Wir können daher dieses Laufwerk für den ST (noch) nicht empfehlen, es soll aber noch in diesem Jahr ein verbessertes Modell diesen Typs geben, das auch eine Steppgeschwindigkeit von 3 Millisekunden besitzt.

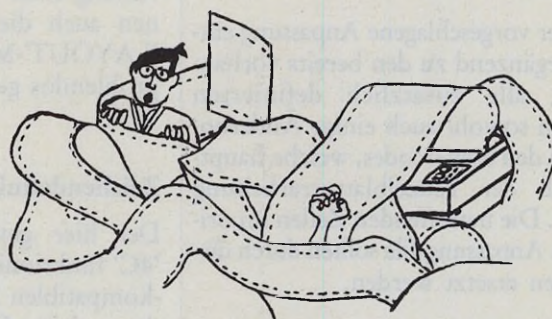
Weiterhin problemlos laufen bei uns die beiden Laufwerke F-354C von CHINON und das FD-35 F von TEAC.

Wir danken Uli Eickmann für seine hilfreiche Unterstützung.



Ist Word komfortable

- Einzelblattverarbeitung
- komprimierte/breite Schrift
- variabler Zeilenabstand
- Drucken nicht druckbarer Zeichen



Von allen bisher vorhandenen Textsystemen bietet 1st Word die beste Druckeranpassung. Mit ihr kann das Textprogramm an jeden Drucker optimal angepaßt werden, d. h. jedes Zeichen und jeder Steuercode aus dem ATARI ST Zeichensatz kann auf einem beliebigen grafikfähigen Drucker ausgegeben werden. Wie man solche Zeichen auf dem Drucker, der diese eigentlich nicht beherrscht, ausgibt, wird im folgenden erläutert.

Die Leistungsfähigkeit eines jeden Textsystems hängt nicht zuletzt auch vom Zusammenarbeiten Rechner-Drucker ab. Was nützt beispielsweise Fettschrift oder Subscript auf dem Bildschirm, wenn diese auf dem Drucker nicht erscheinen, oder wenn die deutschen Sonderzeichen, obwohl der Drucker sie beherrscht, nicht korrekt ausgegeben werden.

Falls solch ein Phänomen auftritt, liegt die Ursache nicht beim Drucker, sondern bei dem Textprogramm, das nur einige, oft die meistverkauften, Druckertypen berücksichtigt. Auch gerade bei einem neuen Rechner wird so manches Programm auf die Schnelle umgeschrieben und auf die Fähigkeiten des speziellen Rechners nicht genügend Wert gelegt.

Ein Textprogramm, das mit diesen Anpassungsschwierigkeiten nicht behaftet ist, ist 1st Word von dem englischen Softwarehaus GST. Jeder Code, der vom Rechner geschickt wird, kann im Druckeranpassungsprogramm (INSTALL) durch einen anderen, druckerverständlichen ersetzt werden. Dies betrifft die Steuercodes (z. B.: Schriftart) genauso wie die internationalen Sonderzeichen (z. B.: ß u ä ê []).

Auf der 1st Word-Diskette sind eine Reihe von fertig vordefinierten Druckeranpassungsfiles vorgesehen (★.HEX).

Damit ist ein problemloses Arbeiten mit fast allen Druckern möglich, und es steht dem sofortigen Ausdruck nichts im Wege. Falls es dennoch zu Problemen kommt oder falls eine ganz besondere Druckeranpassung erwünscht ist, ist es notwendig das Druckerhandbuch und das 'INSTALL' Programm zu Rate zu ziehen. Am besten nimmt man das bestfunktionierende Anpassungsfile zur Grundlage. Darin ist angegeben, an welcher Stelle die betreffenden Steuercodes eingetragen werden. Auch solche Drucker, die von Anfang an einwandfrei arbeiten, können durch die Druckeranpassung noch an Leistung hinzugewinnen. Bei Druckern, die eine Schriftart überhaupt nicht kennen, kann diese durch eine andere ersetzt werden.

Hierzu ein praktisches Beispiel:

Lightschrift wird von den wenigsten Druckern beherrscht. An deren Stelle kann beispielsweise Proportionalschrift treten.

Dazu ändert man die Zeilen

★ E	★ Draft light on
und	
★ F	★ Draft light off
in	
OE, 1B, 70, 01	★ Draft Proportionalschrift
und	
OF, 1B, 70, 00	★ Draft Proportionalschrift

Die erste Zahl (hier OE bzw. OF) steht für den Code, der aus dem Textfile an das Druckerprogramm geschickt wird. Dieser Code wird, falls angegeben, im Druckerprogramm durch die darauffolgenden Bytes ersetzt. Der Stern gilt als Kommentarzeichen, d. h. alles was hinter ihm steht, wird ignoriert.

Anpassung des gesamten ST Zeichensatzes

1st Word beherrscht alle Zeichen des ST-Zeichensatzes. Sie können auf dem Bildschirm abgebildet werden, doch beim Ausdrucken kommt es zu Problemen, da ein gewöhnlicher Drucker diese nicht kennt. Schade, mag so mancher Benutzer sagen, der gerne die griechischen und die mathematischen Zeichen, sowie die Integralzeichen, benutzen würde, der Drucker damit jedoch nichts anzufangen weiß. Gerade aber mit diesen Zeichen sowie der Fähigkeit die verschiedenen Schriftarten auf dem Bildschirm abzubilden, ist der ATARI ST ein ganzes Stück komfortabler als so manches altbewährte Textsystem. Also warum diese Fähigkeiten vernachlässigen? Mit Hilfe der sehr komfortablen Druckeranpassung können auch solche Zeichen gedruckt werden. Die Lösung liegt im Bit-Map Modus (Einzelnadel-Ansteuerung). Darin wird ein Zeichen frei definiert, d. h. die Werte gemäß Bild 1 an den Drucker

Die hier vorgeschlagene Anpassung enthält, ergänzend zu den bereits vorhandenen, alle zusätzlich definierten Zeichen sowohl auch einige Änderungen bei den Steuercodes, welche hauptsächlich der Einzelblattverarbeitung dienen. Die betreffenden Zeilen im originalen Anpassungsfile sollten durch diese neuen ersetzt werden.



In Version 1.00 arbeitete das Ausdrucken auf Einzelblätter nicht einwandfrei.

In der neueren Version (1.04), die uns zum Test zur Verfügung stand, wurde auch sie berücksichtigt. Die Entscheidung: Endlos / Einzelblatt wird im sechsten Konfigurationsparameter festgelegt (siehe Listing 1).

Wählt man 'Einzelblatt', so stoppt der Rechner nach jedem Blatt den Ausdruck und wartet, bis ein neues Blatt eingelegt wird. Damit das Papier auch bis zur letzten Zeile bedruckt wird, sollte man den Papierendsensor ausschalten. Das ist entweder durch die DIP-Schalter im Drucker oder durch einen Software-Steuercode möglich. Eine geeignete Stelle in der 1st Word Druckeranpassung bietet beispielsweise die vertikale Initialisierung. Ihr fügt man den Steuercode '1B,38' an. Diese Initialisierung schickt der Drucker am

Zeichendefinition

Der hier gewählte Bit-Map Modus '4C' funktioniert bei allen Epson- und -kompatiblen Druckern korrekt. Falls dennoch ein Drucker diesen Steuercode nicht versteht, muß dieser durch den druckerspezifischen Steuercode ersetzt werden.

Im Bit-Map Modus werden 12 Bytes übergeben, die auf dem Drucker das betreffende Zeichen erzeugen. Diese Anzahl von Punkten entsprechen hier gerade der Breite eines Zeichens. Dies sollte auch beibehalten werden, da sonst eine horizontale Verschiebung des restlichen Textes dieser Zeile auftritt. Natürlich können auch breitere Zeichen definiert werden. Dies ist lediglich durch die Zeilenlänge des Text-

programmes begrenzt (160 Zeichen), da sämtliche Bytes eines Zeichens in einer Zeile stehen müssen. Somit kann man sich ein eigenes Briefkopfsymbol definieren, das im Brief oder im 'LAYOUT' eingebaut werden kann. Ein Beispiel eines solchen Zeichens zeigt Listing. Hier wird die digitale NEUN benutzt. Diese erhält man im Layout-Menü durch die Tasten 'Controll + 9', im normalen Text aus dem Zeichenauswahlfenster.

Im Gegensatz zu den übrigen Zeichen gehören diese nicht der Umschaltung DRAFT/NLQ, so daß man sich auf eine Schriftart festlegen muß. Die Zeichendefinition in unserer Anpassung wurde der Epson Draft Schrift angepaßt (siehe Bild). Natürlich können auch Zeichen einer anderen Schriftart definiert werden. Dabei sollte man aber auf das spezielle Schriftbild des Druckers eingehen. Listing 3 enthält hierzu ein einfaches Basicprogramm das die für die Anpassung benötigten Werte errechnet.

```

100 ' program Zeichendefinition_fuer_Drucker;
110 laenge = 12: dim p%(laenge)
120 '
130 a$(7)= "-*****---"
140 a$(6)= "-*-----*---"
150 a$(5)= "--*-----"
160 a$(4)= "---*-----"
170 a$(3)= "---*-----"
180 a$(2)= "-*-----*---"
190 a$(1)= "-*****---"
200 a$(0)= "-----"
210 '
220 for x=1 to laenge: p%(x)=0
230 for y=0 to 7
240 p%(x)=p%(x)+abs(asc(mid$(a$(y),x,1))=asc("*")) *2^y
250 next: next
260 ' -----
270 lprint "__", 1B, 4C, 0C, 00, " ";
280 for x= 1 to laenge-1
290 lprint hex$(p%(x)) + ",": next:
300 lprint hex$(p%(laenge)) + " * ";
310 ' -----
320 lprint chr$(27)+chr$(&h4C)+chr$(laenge)+chr$(0);
330 for x=1 to laenge : out 0, p%(x) :next: lprint
340 end

```

Man lädt ein passendes Anpassungsfile (★.HEX) aus dem Printer-Ordner. Dazu ändert man im Lademenü durch Anklicken der Zeile unterhalb 'Disk-Verzeichnis' '★.DOC' in '★.HEX'.

Anschließend klickt man den linken Verschiebepfeil des Auswahlfeldes an. Daraufhin erscheinen alle Druckeranpassungsdateien.

- Hex-File laden
- Dabei ist darauf zu achten, daß man sich nicht im WP-Mode befindet. – Die zu ändernden Zeilen überschreibt man nun mit den neuen Informationen und speichert sie unter einem anderen Namen (z. B.: Druck_1.HEX) ab.
- Nun wird das Textprogramm verlassen und 'INSTALL.Prg' geladen. Dieses Programm fragt nach dem neuen Anpassungsfile und erzeugt das File '1st_Print.DOT'.
- Dieses File muß nun aus dem Printer-Ordner zu dem Hauptprogramm (1st_Word.Prg) kopiert werden.
- Falls dies fehlerfrei geschehen ist, kann der Probeausdruck erfolgen.

Weitere Steuercodes

Um alle Fähigkeiten des Druckers ausreizen zu können, kann man noch weitere Steuercodes definieren. Dazu wurden die digitalen Zahlen 'mißbraucht' (Listing 2). Auf diese Art können auch andere Schriftarten (komprimiert, breit, Elite oder Proportionschrift) zusätzlich benutzt werden. Eine weitere Möglichkeit ist die Einbindung von Steuerbefehlen zum Verändern des Zeilenabstandes. Ein jeder Drucker verfügt über mehrere feste Zeilenabstände, dazu gehören 1/6", 1/8", 7/72" sowie ein variabler Zeilenabstand in 1/256" Schritten. Wer gerne das Blatt dichter beschreiben möchte, kann dies durch Angabe des betreffenden Steuercodes dem Drucker mitteilen. Beachten Sie hierbei die Zeilenanzahl im LAYOUT-Menü. Der 7/72" Zeilenabstand eignet sich besonders für den Ausdruck von Tabellen. Die Zeichen werden hierbei nahtlos untereinander zusammengefügt. Am Beispiel des Interzeichen ist dies gut zu erkennen.

```
*****
*
*   Matrix Printer Driver Configuration Table
*
*****
* Name of printer
* =====
EPSON u.a.

* Miscellaneous configurable variables
* =====
*
* 1: printer type, 0=dot matrix, 1=daisy wheel
* Note if printer type is 0 the following 4 variables are never used.
* 2: unit width of one character
* 3: unit height of one line
* 4: Approximate middle of carriage after formfeed
* 5: Carriage shift for bold overstrike
* 6: Einzelblatt = 1 ; Endlos = 0

0, 0, 0, 0, 0, 0 * Endlos
* 0, 0, 0, 0, 0, 1 * Einzelblatt

* Printer characteristics
* =====

20, 1B, 38 * Papierendsensor aus

* Translation Table
* =====

1, 1B, 4C, C, 0, 0, 10, 30, 0, 5E, 0, 82, 0, 5E, 0, 30, 10 * 0
2, 1B, 4C, C, 0, 0, 10, 18, 0, F4, 0, 82, 0, F4, 0, 18, 10 * 1
3, 1B, 4C, C, 0, 0, 38, 0, 28, 0, 28, 44, 0, 28, 0, 10, 0 * 2
4, 1B, 4C, C, 0, 0, 10, 0, 28, 0, 44, 28, 0, 28, 0, 38, 0 * 3
5, 1B, 4C, C, 0, 0, 0, 7C, 38, 92, C6, EE, C6, 92, 38, 7C, 0 * 4
6, 1B, 4C, C, 0, 0, 0, FE, FE, FC, F8, F2, E6, CE, 9E, 0, 0 * 5
7, 1B, 4C, C, 0, 0, 0, EE, C6, D6, 92, 38, 92, D6, C6, EE, 0 * 6
8, 1B, 4C, C, 0, 0, 18, 0, C, 0, 6, 0, C, 0, 30, 0, C0 * 7

E, 1B, 4C, C, 0, 0, 0, 3, 0, 7, 0, E, 0, F8, 0, 0, FF * 8
F, 1B, 4C, C, 0, 0, 0, FF, 0, 0, F8, 0, E, 0, 7, 0, 3, 0, 0 * 9

10, 1B, 4C, C, 0, 0, 0, 6C, 0, EE, 0, 82, 0, 82, EE, 0, 6C * A
11, 1B, 4C, C, 0, 0, 0, 0, 0, 0, 0, 0, 0, EE, 0, EE * B
12, 1B, 4C, C, 0, 0, 0, C, 0, 9E, 0, 9E, 0, 92, F2, 0, 60 * C
13, 1B, 4C, C, 0, 0, 0, 0, 92, 0, 92, 0, 92, 0, FE, 0, 6C * D
14, 1B, 4C, C, 0, 0, 0, E0, 0, F0, 0, 10, 0, 10, FE, 0, EE * E
15, 1B, 4C, C, 0, 0, 0, 60, 0, F2, 0, 92, 0, 92, 1E, 0, C * F
16, 1B, 4C, C, 0, 0, 0, 6C, 0, FE, 0, 92, 0, 92, 9E, 0, C * 10
17, 1B, 4C, C, 0, 0, 0, 0, 80, 0, 80, 0, 80, 0, EE, 0, 6E * 11

18, 1B, 4C, C, 0, 0, 0, 6C, 0, FE, 0, 92, 0, 92, FE, 0, 6C * 12
19, 1B, 4C, C, 0, 0, 0, 60, 0, F2, 0, 92, 0, 92, FE, 0, 6C * 13

7C, 1B, 4C, C, 0, 0, 0, 0, 0, 0, 0, 0, FF, 0, 0, 0, 0 * 14
7F, 1B, 4C, C, 0, 0, 2, 6, 1E, 1A, 7A, 62, 62, 1A, 6, 2, 0 * 15

9B, 1B, 4C, C, 0, 0, 0, 18, 24, 0, 24, C3, 24, 0, 24, 0, 0 * 16
9F, 1B, 4C, C, 0, 0, 0, 9, 0, 9, 36, 48, 0, 48, 0, 40, 0 * 17

A9, 1B, 4C, C, 0, 0, 0, 0, 0, 1F, 0, 10, 0, 10, 0, 10, 0 * 18
AA, 1B, 4C, C, 0, 0, 0, 10, 0, 10, 0, 10, 0, 1F, 0, 0, 0 * 19
AB, 1B, 4C, C, 0, 0, 0, E3, 6, C, 18, 30, 65, C9, B, 5, 0 * 1A
AC, 1B, 4C, C, 0, 0, 0, E3, 6, C, 18, 30, 66, CA, 12, 1F, 0 * 1B

AE, 1B, 4C, C, 0, 0, 0, 10, 28, 44, 82, 10, 28, 44, 82, 0, 0 * 1C
AF, 1B, 4C, C, 0, 0, 0, 82, 44, 28, 10, 82, 44, 28, 10, 0, 0 * 1D

B4, 1B, 4C, C, 0, 0, 0, 1C, 22, 0, 22, 1C, 20, A, 20, 1A, 0 * 1E
B5, 1B, 4C, C, 0, 0, 0, 7C, 82, 0, 82, FE, 0, 92, 0, 92, 0 * 1F
B6, 1B, 4C, C, 0, 0, 0, 6, 8, 14, 80, 54, 0, 14, 8, 6, 0 * 20
B7, 1B, 4C, C, 0, 0, 0, 46, 88, 14, 80, 54, 0, 54, 88, 6, 0 * 21
B8, 1B, 4C, C, 0, 0, 0, 4C, 92, 0, 92, 40, 12, 40, 92, C, 0 * 22

BB, 1B, 4C, C, 0, 0, 0, 0, 20, 0, 7C, 0, 20, 0, 0, 0, 0 * 23
BC, 1B, 4C, C, 0, 0, 0, 60, 0, F0, 0, 94, 0, FF, 0, 4, 0 * 24
BD, 1B, 4C, C, 0, 0, 0, 7E, 0, 99, 0, A5, 24, 81, 0, 7E, 0 * 25
BE, 1B, 4C, C, 0, 0, 0, 7E, 0, BD, 0, A9, 14, 81, 0, 7E, 0 * 26
BF, 1B, 4C, C, 0, 0, 0, 80, 0, F0, 0, 80, F0, 0, 40, 0, F0, 0 * 27
C0, 1B, 4C, C, 0, 0, 0, 24, 0, BC, 0, 5, 0, 1, 2, BC, 0 * 28
C1, 1B, 4C, C, 0, 0, 0, 84, 0, FC, 0, 85, 0, 1, 2, FC, 0 * 29
```

C2,	1B,	4C,	C,	0,	0,	0,	40,	4,	28,	30,	0,	18,	28,	40,	4,	0	*	×	
C3,	1B,	4C,	C,	0,	0,	0,	44,	0,	44,	0,	44,	0,	7C,	0,	4,	0	*	×	
C4,	1B,	4C,	C,	0,	0,	0,	0,	0,	0,	4,	40,	8,	40,	10,	7C,	0	*	×	
C5,	1B,	4C,	C,	0,	0,	0,	40,	0,	40,	0,	40,	0,	7C,	0,	40,	0	*	×	
C6,	1B,	4C,	C,	0,	0,	0,	5C,	0,	40,	0,	40,	0,	40,	0,	3C,	0	*	×	
C7,	1B,	4C,	C,	0,	0,	0,	0,	0,	0,	0,	40,	0,	7C,	0,	0,	0	*	×	
C8,	1B,	4C,	C,	0,	0,	0,	0,	0,	40,	0,	40,	0,	60,	1C,	40,	0	*	×	
C9,	1B,	4C,	C,	0,	0,	0,	0,	0,	40,	3C,	40,	0,	40,	0,	7C,	0	*	×	
CA,	1B,	4C,	C,	0,	0,	0,	7C,	0,	4,	0,	4,	0,	44,	0,	7C,	0	*	×	
CB,	1B,	4C,	C,	0,	0,	0,	0,	0,	40,	0,	40,	0,	70,	0,	0,	0	*	×	
CC,	1B,	4C,	C,	0,	0,	0,	0,	0,	44,	0,	44,	0,	44,	0,	38,	0	*	×	
CD,	1B,	4C,	C,	0,	0,	0,	C0,	0,	40,	0,	40,	0,	44,	8,	70,	0	*	×	
CE,	1B,	4C,	C,	0,	0,	0,	5C,	0,	20,	0,	20,	0,	44,	0,	3C,	0	*	×	
CF,	1B,	4C,	C,	0,	0,	0,	0,	0,	4,	0,	44,	0,	7C,	0,	0,	0	*	×	
D0,	1B,	4C,	C,	0,	0,	0,	0,	0,	78,	0,	44,	0,	44,	0,	78,	0	*	×	
D1,	1B,	4C,	C,	0,	0,	0,	4,	0,	7C,	0,	4,	0,	4,	0,	7C,	0	*	×	
D2,	1B,	4C,	C,	0,	0,	0,	74,	0,	54,	0,	44,	0,	44,	0,	7C,	0	*	×	
D3,	1B,	4C,	C,	0,	0,	0,	64,	0,	24,	0,	14,	0,	C,	0,	64,	0	*	×	
D4,	1B,	4C,	C,	0,	0,	0,	0,	0,	5E,	0,	40,	0,	78,	0,	0,	0	*	×	
D5,	1B,	4C,	C,	0,	0,	0,	0,	0,	40,	0,	40,	0,	40,	0,	3C,	0	*	×	
D6,	1B,	4C,	C,	0,	0,	0,	7C,	0,	14,	0,	64,	0,	4,	0,	7C,	0	*	×	
D7,	1B,	4C,	C,	0,	0,	0,	4,	0,	7C,	0,	40,	0,	7C,	0,	0,	0	*	×	
D8,	1B,	4C,	C,	0,	0,	0,	0,	0,	0,	0,	40,	0,	7E,	0,	0,	0	*	×	
D9,	1B,	4C,	C,	0,	0,	0,	0,	0,	40,	0,	40,	0,	7E,	0,	0,	0	*	×	
DA,	1B,	4C,	C,	0,	0,	0,	40,	0,	7C,	0,	44,	0,	44,	0,	7C,	0	*	×	
DB,	1B,	4C,	C,	0,	0,	0,	0,	0,	70,	0,	50,	0,	40,	0,	7E,	0	*	×	
DC,	1B,	4C,	C,	0,	0,	0,	0,	0,	60,	0,	10,	0,	7E,	0,	0,	0	*	×	
DE,	1B,	4C,	C,	0,	0,	0,	6,	0,	C,	0,	18,	0,	C,	0,	6,	0	*	×	
DF,	1B,	4C,	C,	0,	0,	1C,	22,	0,	22,	14,	8,	14,	22,	0,	22,	1C	*	×	
E0,	1B,	4C,	C,	0,	0,	0,	1C,	22,	0,	22,	14,	8,	14,	22,	0,	0	*	×	
E2,	1B,	4C,	C,	0,	0,	0,	41,	0,	7F,	0,	41,	0,	60,	0,	70,	0	*	×	
E3,	1B,	4C,	C,	0,	0,	0,	20,	0,	3E,	0,	20,	0,	3E,	0,	20,	0	*	×	
E4,	1B,	4C,	C,	0,	0,	0,	82,	0,	C6,	0,	AA,	0,	92,	0,	C6,	0	*	×	
E5,	1B,	4C,	C,	0,	0,	0,	1C,	0,	22,	0,	62,	0,	9C,	0,	40,	0	*	×	
E6,	1B,	4C,	C,	0,	0,	0,	1,	0,	3E,	0,	2,	0,	3E,	0,	2,	0	*	×	
E7,	1B,	4C,	C,	0,	0,	0,	20,	0,	3E,	0,	20,	0,	20,	0,	0,	0	*	×	
E8,	1B,	4C,	C,	0,	0,	0,	0,	18,	A5,	0,	E7,	0,	A5,	18,	0,	0	*	×	
E9,	1B,	4C,	C,	0,	0,	0,	0,	18,	24,	52,	0,	52,	24,	18,	0,	0	*	×	
EA,	1B,	4C,	C,	0,	0,	0,	12,	28,	46,	0,	40,	0,	46,	28,	12,	0	*	×	
EB,	1B,	4C,	C,	0,	0,	0,	18,	4,	22,	40,	22,	90,	C,	40,	0,	0	*	×	
EC,	1B,	4C,	C,	0,	0,	0,	18,	1,	24,	1,	7E,	80,	24,	80,	18,	0	*	×	
ED,	1B,	4C,	C,	0,	0,	0,	1C,	0,	22,	0,	7F,	0,	22,	0,	1C,	0	*	×	
EE,	1B,	4C,	C,	0,	0,	0,	7C,	0,	92,	0,	92,	0,	92,	0,	92,	0	*	×	
EF,	1B,	4C,	C,	0,	0,	0,	7E,	0,	80,	0,	80,	0,	80,	0,	7E,	0	*	×	
F0,	4C,	C,	0,	0,	0,	54,	0,	54,	0,	54,	0,	54,	0,	54,	0	0	*	×	
F4,	1B,	4C,	C,	0,	0,	0,	0,	0,	0,	0,	3F,	40,	80,	0,	60,	0	*	×	
F5,	1B,	4C,	C,	0,	0,	0,	6,	0,	1,	2,	FC,	0,	0,	0,	0,	0	*	×	
F6,	1B,	4C,	C,	0,	0,	0,	10,	0,	10,	44,	10,	44,	10,	0,	10,	0	*	×	
F7,	1B,	4C,	C,	0,	0,	0,	24,	48,	0,	48,	24,	12,	0,	12,	24,	0	*	×	
FA,	1B,	4C,	C,	0,	0,	0,	0,	0,	0,	0,	1C,	0,	1C,	0,	0,	0	*	×	
FB,	1B,	4C,	C,	0,	0,	8,	0,	4,	0,	2,	C,	30,	0,	20,	0,	20,	0	*	×
FC,	1B,	4C,	C,	0,	0,	0,	0,	8,	80,	0,	80,	0,	78,	0,	0,	0	*	×	
FD,	1B,	4C,	C,	0,	0,	0,	98,	0,	A8,	0,	48,	0,	0,	0,	0,	0	*	×	
FE,	1B,	4C,	C,	0,	0,	0,	A8,	0,	A8,	0,	50,	0,	0,	0,	0,	0	*	×	
FF,	1B,	4C,	C,	0,	0,	40,	0,	40,	0,	40,	0,	40,	0,	40,	0,	40,	0	*	×

Steuercodes

1	komprimierte Schrift	ein
2	breite Schrift	ein
3	Doppeldruck	ein
4	Doppeldruck	aus
5	...	
6	Zeilenabstand 1/6"	(normal)
7	Zeilenabstand 7/75"	(Grafik)
8	Zeilenabstand 1/8"	
9	...	
0	breit/komprimiert	aus

Verschiedene Zeichensätze

Da der ATARI ST optional mit verschiedenen Zeichensätzen betrieben werden kann, stehen diese dann auch 1st Word zur Verfügung. Interessant sind Tabellensymbole oder andere Grafikzeichen anstelle der hebräischen Zeichen, da diese wohl am wenigsten benutzt werden. Damit würde der optische Eindruck der Texte weiter verbessert und die grafischen Fähigkeiten des Rechners mehr in Anspruch genommen. Doch vorerst viel Erfolg bei der Druckeranpassung. Zum Schluß zwei Tips:

Falls öfters Änderungen vorgenommen werden, ist die Verwendung einer RAM-Disk sehr zu empfehlen, da damit der gesamte Erstellungsvorgang (Text laden/speichern, Install, Umladen, Ausdruck) wesentlich schneller geht.

Wenn man unsicher ist, ob ein Code überhaupt genutzt wird, sollte man zusätzlich einen Pfeifton (07) im HEX-File anfügen.

Diskettenservice

Diskettenservice

Sämtliche, in der ST-Computer veröffentlichten Programme, können Sie auch auf Diskette bestellen.

Januar/Februar DM 28,-
März/April DM 28,-

Adresse:

Heim-Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon (0 61 51) 5 53 75

S-TERM Plus ist da!!!

Universelles Terminal-Programm für Modem, Mail-Box und Datenübertragung mit anderen Rechnern. Abspeichern auf Disk möglich, Belegung der Funktionstasten mit beliebigen langen Texten, Umlautwandlung und vielen Extras.
Jetzt unter GEM

Incl. ausführlichem deutschen Handbuch

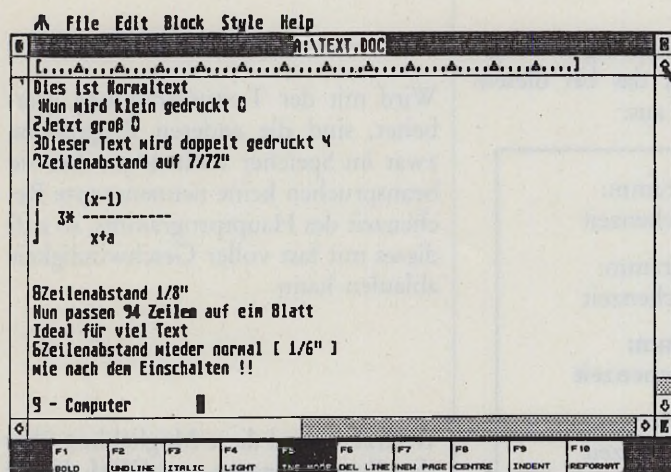
DM 199,— (Vorkasse frei/NN zugl. P + V)

ING.-Büro ZOSCHKE

Berliner Straße 3
2306 Schönberg/Holstein
Telefon 0 43 44/61 66

10, 12, 14 * klein/ groß aus	LCD 0
11, Of * komprimierter Druck	LCD 1
12, OE * doppelt-breiter Druck	LCD 2
13, 1B, 47 * Doppeldruck	LCD 3
14, 1b, 48 * Doppeldruck aus	LCD 4
* 15,	LCD 5
16, 1B, 32 * Zeilenabstand 1/6 (normal)	LCD 6
17, 1b, 31 * Zeilenabstand 7/72 (grafik)	LCD 7
18, 1B, 30 * Zeilenabstand 1/8 (	LCD 8
19, 1B, 4c, 24, 0, 0, 0, 0, 7, 7, 7, 7, 7, 3f, 7e, fe, fc, e0, e0, e0, e0, e0, 0, 0, 0, 0, c0, c0, c0, c0, c0, c0, ff, ff, ff, ff, ff, c0, c0, c0, c0, c0, c0 * ST (LCD9)	

Listing 1: Installfile



Original:

Dies ist Normaltext
 Nun wird klein gedruckt
Jetzt groß
 Dieser Text wird doppelt gedruckt
 Zeilenabstand auf 7/72"

$$\int 3x \frac{(x-1)}{x+a}$$

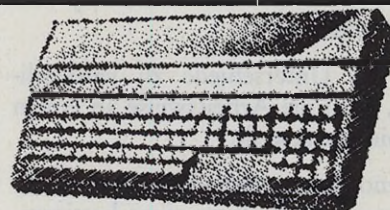
Zeilenabstand 1/8"
 Nun passen 94 Zeilen auf ein Blatt
 Ideal für viel Text
 Zeilenabstand wieder normal [1/6"]
 wie nach dem Einschalten !!

ST - Computer

Ausdruck:

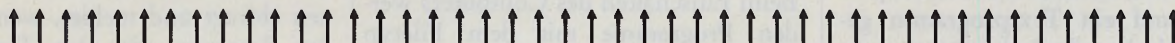
SOFTWARE FÜR DIE ATARI ST-COMPUTER

520ST



Professionelle Datenbank von H&D (unterstützt die speziellen Eigenschaften des ST) 298,- DM
 Ultraschneller Kopierprogramm für geschützte Software (z.B. VIP Professional) 149,- DM
 Profi-Textverarbeitung (automatisches Inhaltsverzeichnis, Fußnotenverwaltung, ...) 449,- DM

→ LATICE C-Compiler (Floating Point, Macros und Runtime Licence) 329,- DM ←
 → 4xForth Programmiersystem 439,- DM ←
 → Professionelle GAD und Satzsoftware auf Anfrage ←



Der Super Hit: 3.5" Disk 10 Stück 69,- DM

Steckbare Erweiterung für den Atari 268ST auf 1 MByte 349,- DM

Spiele auf Anfrage (Hacker, Universe II, Hex II, Montezumas Revenge)

Informationen erhalten Sie bei:

Hendrik Haase Computersysteme, Wiedfeldtstr. 77, D-4300 Essen 1, Tel.: 02 01 - 42 25 75
 ! Händleranfragen erwünscht !

Deskaccessories contra Multitasking

Seitdem es Computer mit leistungsfähigen Prozessoren gibt, ist es möglich geworden, mehrere Programme gleichzeitig ablaufen zu lassen, so daß z. B. im Hintergrund eine Uhr läuft, oder daß ein Spoolprogramm Texte auf dem Drucker ausdruckt, während man mit einem anderen Programm arbeitet. Das Verfahren bei dem quasi parallel mehrere Programme bearbeitet werden, nennt man Multitasking. Die verschiedenen Varianten des Multitasking sollen an zwei unterschiedlichen Konzepten dargestellt werden.

ATARI GEM im Vergleich mit AMIGADOS, dem Betriebssystem des AMIGA.

Um die Unterschiede zu verdeutlichen, wird die Arbeitsweise an einem Beispiel dargestellt.

Angenommen, man arbeitet an seinem Rechner und muß zu einer bestimmten Uhrzeit, vielleicht 15 Uhr, noch etwas wichtiges erledigen. Als erstes startet man ein Programm, das um 15 Uhr auf sich aufmerksam machen soll und dann anfängt zu klingeln. In der Zeit sollen noch ein Brief geschrieben und einige Bilder in den Text eingefügt werden, so daß ein Zeichenprogramm und ein Textprogramm geladen und abwechselnd benutzt werden. Zu irgendeinem bestimmten Zeitpunkt wird der Text ausgedruckt und dafür ein Spoolprogramm aufgerufen, das den Text vorläufig speichert und dann nebenbei an den Drucker weiterleitet. Zu diesem Zeitpunkt wird das Betriebssystem betrachtet. Ein einfaches Multitaskingsystem (AMIGA) teilt jedem Programm eine bestimmte Zeit zu, in der es jeweils abgearbeitet wird. Vorausgesetzt, daß die Zeitein-

teilung einigermaßen intelligent vorgenommen wird, sieht das bei diesem Beispiel ungefähr so aus:

Weckerprogramm: ca. 15 % Rechenzeit Zeichenprogramm: ca. 35 % Rechenzeit Textprogramm: ca. 35 % Rechenzeit Druckerspooler: ca. 15 % Rechenzeit
--

Das bedeutet, die beiden Hauptprogramme laufen in nur einem Drittel der gesamten Rechenzeit, und das kann bei rechenintensiven Routinen schon durch eine deutliche Verzögerung bemerkbar werden. Außerdem sind die 15 % Zeit für den Wecker eine Verschwendung, da das Programm bestimmt einige Male pro Sekunde aufgerufen wird, um dann festzustellen, daß es immer noch nicht Zeit zum Klingeln ist.

Bei GEM wird das Multitasking auf eine etwas elegantere Art gelöst. Voraussetzung ist hierfür, daß die Programme mit der GEM-Menüleiste arbeiten und Deskaccessories zugelassen werden. Beim Einschalten des Computers werden Programme mit dem Filetyp „ACC“ automatisch mit in den Rechner geladen und bleiben resident im Speicher (normalerweise!). Wenn man dann in einem Programm den Menüpunkt „Desktop Info“ anwählt, kann anschließend das jeweils gewünschte Accessory gestartet werden.

Eigentlich ist diese Möglichkeit dafür vorgesehen, kleinere Hilfsprogramme – wie Taschenrechner oder eine Uhr – bei Bedarf einzublenden. Bei einem

Hauptspeicher von 1 Mbyte und mehr kann es aber auch sinnvoll sein, etwas längere Programme als Deskaccessories zu benutzen. Angenommen, es sind Uhr (Wecker), Druckerspooler und Zeichenprogramm als DA's vorhanden und man benutzt ein Textverarbeitungsprogramm wie 1st Word, dann ist es viel eleganter, hier mit den verschiedenen Programmen zu arbeiten als bei einem einfachen Multitaskingsystem.

Wird mit der Textverarbeitung gearbeitet, sind die anderen Programme zwar im Speicher vorhanden, aber sie beanspruchen keine nennenswerte Rechenzeit des Hauptprogramms, so daß dieses mit fast voller Geschwindigkeit ablaufen kann.

Realisiert wird diese Möglichkeit über den Event-Manager von GEM. Unter einem EVENT (Ereignis, besser: Aktion) ist hierbei die Bewegung der Maus, das Drücken einer Taste oder des Mausknopfs, das Anwählen eines Menüpunktes oder das Verändern eines Fensters anzusehen. Ferner kann auch das Erreichen eines vorgegebenen Zeitpunkts dazu führen, daß der Event-Manager in Aktion tritt (Timer-Event). Deshalb sollten GEM-Programme ihre üblichen Eingaben auch über den „EVENT_MULTI“ erhalten, weil erst dadurch ermöglicht wird, daß der Event-Manager auswählen kann, welches Programm oder welche Aktion als nächstes zu bearbeiten ist. Mit „EVENT_MULTI“ wird dabei die Routine bezeichnet, die fast alle Aktionen abfragt und meldet, wenn durch den Benutzer oder den Timer eine Eingabe erfolgt bzw. ein bestimmter Zeitraum verstrichen ist.

In dem genannten Beispiel sieht das Multitasking folgendermaßen aus:

1. Der Druckerspooler wird geladen und hat vorerst nichts weiter zu tun als im EVENT_MULTI zu warten, bis er angewählt wird, um etwas zu Drucken.

2. Der Wecker wird geladen, die gewünschte Zeit eingestellt, und über EVENT__MULTI wird einmal pro Sekunde überprüft, ob die gewünschte Zeit schon erreicht ist.
3. Das Zeichenprogramm wird ebenfalls nur geladen und wartet geduldig, bis der EVENT__MULTI die Anwahl des Programms meldet.
4. Das Textprogramm wird geladen und gestartet, läuft mit nahezu voller Geschwindigkeit ab und empfängt seine Eingaben über den EVENT__MULTI.

Die Zeiteinteilung bei Textverarbeitung sieht hierbei ungefähr so aus:

Weckerprogramm: ca. 3 % Rechenzeit Zeichenprogramm: ca. 2 % Rechenzeit Textprogramm: ca. 93 % Rechenzeit Druckerspooher: ca. 2 % Rechenzeit

Wird nun zu einem beliebigen Zeitpunkt das Zeichenprogramm aufgerufen, zeigt sich die eigentliche Flexibilität von GEM, da dann fast die gesamte Zeit des Textprogramms an das Zeichenprogramm abgegeben werden kann und somit wieder nahezu die komplette Rechenzeit für das ausgewählte Programm zur Verfügung steht, ohne daß im Hintergrund unnötige Routinen ablaufen.

Als Resümee bleibt festzuhalten: wenn auf einem Microcomputer mehrere Arbeiten nebeneinander erledigt werden sollen, ist es wichtig, daß dabei der Vorteil eines schnellen Computers nicht dadurch zunichte gemacht wird, daß die Programme nur mit deutlich reduzierter Geschwindigkeit ablaufen, sondern ein intelligenter 'Aktionsmanager' für eine schnelle Programmverwaltung sorgt.

Oliver Joppich

Atari macht Spitzentechnologie preiswert.

WIR MACHEN SPITZENSOFTWARE PREISWERT !

.....

QUIWI

Unser erstes Programm für den Atari ST und das erste Computerspiel für die ganze Familie! Vorbei sind die einsamen Stunden am Monitor - jetzt können alle mitspielen, jung und alt.

- Bis zu 15 Mitspieler
- Original deutsche Fragen mit Umlauten (keine Übersetzung)
- Bund 4000 Fragen aus 6 Wissensgebieten
- Spielerisch dazulernen
- Einfache Bedienung mit der Maus
- Mit schöner (Farb-) Grafik und Musik
- Jederzeit erweiterbar durch Ergänzungsdisketten
- Für Farb- und S/W-Monitor geeignet

3,5"-Diskette für Atari

260 ST, 520 ST, 520 ST+ nur: 69, - DM

.....

PROGRAMMIERER GESUCHT !

Wir suchen Programmierer, die (Assembler-) Programme vom Commodore 64 für den Atari ST umschreiben oder neue Programme für Atari ST entwickeln. Wenn Sie Interesse haben, so rufen Sie uns an oder schreiben Sie uns (Ansprechpartner: Herr Schäfer).

.....

Alle Preise verstehen sich als unverbindliche Preisempfehlung inkl. MwSt. zzgl. 5,- DM Porto & Verpackung. Sie erhalten KINGSOFT-Programme im Computer-Fachhandel, in den Fachabteilungen der Kauf- und Warenhäuser oder direkt von uns.

KINGSOFT

SPITZEN-SOFTWARE
Made in Germany

F. Schäfer, Schnackebusch 4, 5106 Roetgen, Tel. 0 24 08 / 51 19

1000 Berlin

Computare oHG

- ☐ Keithstraße 18, 1000 Berlin 30
- ☐ Behaimstr. 3, 1000 Berlin 10

☎ (030) 2 13 90 21

Telex: 186 346 vom d



alpha computers g.m.b.h.

u. a. alphas, atari, commodore, dai, epson, sord mit pips, nec hard-/software nach maß — servicetechnik

Kurfürstendamm 121a, 1000 Berlin 31 (Halensee)
Telefon 030/8911082

☎ 030-69081 Ihre Tür zur Zukunft:
karstadt-computer-center
hardware-software-problemlösungen
1000 Berlin 61 Hermannplatz

Digital-Computer

Knesebeckstr. 76 · 1000 Berlin 12
Telefon
030-8827791



Vertragshändler

UNION ZEISS

Kurfürstendamm 57 · 1000 Berlin 15
Telefon 32 30 61

2000 Norderstedt



2080 Pinneberg

BPO

Ges. f. Beratung, Planung + Org. GmbH

Dingstätte 34
2080 Pinneberg
Telefon 04101-26071/72

2120 Lüneburg

Sienknecht

Bürokommunikation
Beratung - Verkauf - Werkstatt

Heiligengeiststr. 20, 2120 Lüneburg
Tel. 04131/46122, Btx 402422
Mo.-Fr. 9⁰⁰-18⁰⁰ und Sa. 9⁰⁰-13⁰⁰

2160 Stade



Büromaschinen · EDV-Systeme

Neue Straße 5, 2160 Stade

Telefon: (04141) 23 64 + 23 84

2210 Itzehoe

Der Computerladen

Inhaber: Ulrich Böbel Mathias Kentrup

Holzcamp 12 · 2210 Itzehoe
Telefon 04821/3390

2300 Kiel



Die Welt der Computer
Dreiecksplatz Nr. 7

2300 Kiel 1 · ☎ 0431/567042

2390 Flensburg

E C L

elektronik computer laden ohg

Norderstr. 94-96 · 2390 Flensburg
Telefon (0461) 28181/28193

2800 Bremen

PS-DATA

Doventorsteinweg 41
2800 Bremen
Telefon 0421-170577

2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Kurt Neumann

Georgstraße 71
2850 Bremerhaven
Tel. 0471/302129

HARDWARE · SOFTWARE · PAPIERWARE

2940 Wilhelmshaven

Radio Tiemann GmbH & Co. KG

2940 Wilhelmshaven
Telefon 04421/26145

2950 Leer



- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

Augustenstraße 3 · 2950 Leer
Telefon 0491-4589

3000 Hannover

COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon 0511-326736



DATALOGIC COMPUTERSYSTEME

ATARI ST-BERATUNG
COMPUTER SERVICE
HARDWARE VERKAUF
SOFTWARE
CALENBERGER STR. 26
3000 HANNOVER 1
TEL. 0511-326489

3040 Soltau

F & T Computervertrieb

Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 05191/16522

3100 Celle

Ludwig Haupt jr. Büro-Einkaufs-Zentrum

Gerhard-Kamm-Straße 2
Ruf 8 30 45, Postfach 140
3100 Celle

3550 Marburg

L W M COMPUTER SERVICE

Biegenstraße 43
3550 Marburg/Lahn
☎ 0 64 21 - 6 22 36

4350 Recklinghausen

COMPUTER CENTRALE

Douastr. 1 · 4350 Recklinghausen
Telefon 0 23 61 - 4 57 08

3170 Gifhorn

COMPUTER-HAUS GIFHORN

Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

4000 Düsseldorf

BERNSHAUS G m b H Bürotechnik – Bürobedarf

Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

4400 Münster

BASIS COMPUTER SYSTEME GMBH

Daimlerweg 39 · 4400 Münster
Telefon 02 51 / 71 99 75 - 9

3300 Braunschweig

COMPUTER STUDIO BRAUNSCHWEIG

Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77/78

H O C O EDV ANLAGEN GMBH

Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

4420 Coesfeld

BENSE KG



Computer-
Systeme
Software-
Entwicklung
☎ 0 25 41 / 52 31
Dieselstraße 10-12
D-4420 Coesfeld 1

COMPUTER SHOP

Am EKZ
Kupferpassage
Ausgang Gartenhof
☎ 0 25 41 / 7 23 59
Suringstraße 7
D-4420 Coesfeld 1

3320 Salzgitter

FRICKE

Beratung und Vertrieb
für Computer und Bildschirmtext
Berliner Str. 54 · 3320 Salzgitter 1
Telefon (0 53 41) 4 40 91-2

4050 Mönchengladbach

computer commerce

Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 187 64

4422 Ahaus

ATARI · Apricot · Epson · Fujitsu · Molecular
Computer-Systeme + Software

OCB

Wallstraße 3 · 4422 Ahaus
Telefon 0 25 61 / 50 21

4500 Osnabrück

Heinicke-Electronic

Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41 - 8 27 99

Wir liefern Micro-Computer seit 1978

3400 Göttingen

Büroeinrichtungs-Zentrum Wiederholdt

3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51/34031

4190 Kleve

Computer

Feldmann + Luft

4190 Kleve-Kellen Emmericher Str. 223
Telefon 0 28 21 / 9 10 38 · Telex 811 797

4200 Oberhausen

KAMP

Büro- und Computersysteme

Vestische Straße 89/91
4200 Oberhausen 12 (Osterfeld)
Fernruf-(02 08) 89 00 86
Fernschreiber 8 56 578

4600 Dortmund



Atari, Genie, Schneider, Tandy, Brother, Star, Memorex,
BASF, Verbatim
cc Computer Studio GmbH
Software-Hardware-Beratung
Service-Eilversand

Ihre Ansprechpartner: Elisabethstraße 5
v. Schablinski 4600 Dortmund 1
Jan P. Schneider T. 0231/528184 · Tx 822631 cccsd

3500 Kassel

Hermann Fischer GmbH autorisierter ATARI-Fachhändler

Rudolf-Schwander-Str. 5, 9 + 13
3500 Kassel
Tel. (05 61) 70 00 00

City Elektronik

Güntherstraße 75
4600 Dortmund
Telefon 02 31/57 22 84

Einkaufsführer

4600 Dortmund

Bürostudio BOLZ

Brauhausstraße 4 · 4600 Dortmund
Telefon 02 31-52 77 13-16

5010 Bergheim

Computerstudio HÖLSCHER

EDV-Beratung · Organisation
Programmierung · Home/Personal-Computer
Software · Zubehör · Fachliteratur
Zeppelinstr. 7 · 5010 Bergheim
Telefon 0 22 71-6 20 96

5457 Straßenhaus

DR. AUMANN GMBH Computer-Systeme

Schulstr. 12
5457 Straßenhaus
Telefon 0 26 34-40 81/2

Ihre Tür zur Zukunft:
**KARSTADT
computer-center**
hardware · software · problemlösungen
Dortmund, Kampstraße 1, Telefon (0231) 54391

5060 Bergisch-Gladbach

Computer Center

Buchholzstraße 1
5060 Bergisch-Gladbach
Telefon 0 22 02-3 50 53

5500 Trier

bürocenter LEHR

Güterstr. 82 - 5500 Trier
☎ 06 51-2 50 44
Fordern Sie unsere Zubehör-Liste an.

4790 Paderborn



GESELLSCHAFT FÜR ELEKTRONISCHE
TELEKOMMUNIKATION

IM SCHILDERN 15 TEL. (05251) 2 60 41
4790 PADERBORN BTX ★51051#

5090 Leverkusen

Rolf Rocke

Computer-Fachgeschäft
Austraße 1
5090 Leverkusen 3
Telefon 0 21 71/26 24

5540 Prüm

ATC COMPUTER J. ZABELL

Kalvarienbergstr. 34
5540 PRÜM
- Tel.: 0 65 51-34 83 -

4800 Bielefeld

MICROTEC

Ges. für Microcomputer-Vertrieb mbH

Paul-Schwarze-Str. 5
4800 Bielefeld 14

5200 Siegburg

Computer Center

Luisenstraße 26
5200 Siegburg
Telefon 0 22 41/6 68 54

5600 Wuppertal

Jung am Wall

Wall 31—33
5600 Wuppertal 1
Telefon 02 02/45 03 30

hardware
software
organisation
service **CSF**
CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106—108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

5240 Betzdorf

"BYTE ME" COMPUTERSYSTEME

Wilhelmstraße 7
D 5240 BETZDORF (SIEG)
Telefon (0 27 41) 2 35 37 u. 2 31 07

5630 Remscheid

C O M S O F T

Scheiderstr. 12 · 5630 Remscheid
Telefon (0 21 91) 2 10 33-34

5000 Köln

BÜRO MASCHINEN
braun
AM RUDOLFFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
RUF: 02 21/21 91 71

5400 Koblenz

SCHMITT COMPUTERSYSTEME

Casinostraße 40
5400 Koblenz
☎ 02 61-3 65 28

5800 Hagen

COMPUTER TECHNIK ERKELENZ und KLUG

ATARI-Apple Vertragshändler
Commodore Service-Stützpunkthändler
Hochstr. 96 · 5800 Hagen 1
Telefon 0 23 31/18 13 99

5900 Siegen

HeesComputer
Vertriebs GmbH
 Hardware · Software · Schulung
 Siegen · Weidenauer Str. 72 · ☎ 02 71/7 34 95

6000 Frankfurt

Müller & Nemecek
 Kaiserstraße 44
 6000 Frankfurt/M.
 Tel. 0 69-23 25 44

WAIZENEGGER
Büroeinrichtungen
 Kaiserstraße 41
 6000 Frankfurt/M.
 ☎ 0 69 / 23 92 31

SCHMITT
COMPUTERSYSTEME
 Grosse Friedbergerstr. 30
 6000 Frankfurt
 ☎ 0 69-28 40 65

6100 Darmstadt

Vertragshändler für
 IBM Personal-Computer

**PROFI
 COMPUTER
 STUDIO**

KARSTADT Aktiengesellschaft
 Elisabethenstr. 15
 6100 Darmstadt
 Luisencenter
 Tel. 0 61 51-10 94 20

Heim
Büro- und Computermarkt
 Heidelberger Landstraße 194
 6100 Darmstadt-Eberstadt
 ☎ 0 61 51/5 53 75

SCHMITT
COMPUTERSYSTEME
 Mühlstraße 76
 6100 Darmstadt
 ☎ 0 61 51-2 45 74

6200 Wiesbaden

SCHMITT
COMPUTERSYSTEME
 Rheinstraße 41
 6200 Wiesbaden
 ☎ 0 61 21-30 73 30

6240 Königstein

KFC
COMPUTERSYSTEME
 Wiesenstraße 18
 6240 Königstein
 Tel. 0 61 74-30 33
 Mail-Box 0 61 74-53 55

6300 Gießen

Schneider ATARI Commodore

BAUMS
 BÜRO · ORGANISATION
 Bahnhofstr. 26 · 6300 Gießen
 Telefon: 06 41 / 7 10 96

☎ 06 41/70 04-318 Ihre Tür zur Zukunft:

**karstadt-
 computer-center**
 hardware-software-problemlösungen
 6300 Gießen Seltersweg 64

6330 Wetzlar

F Fachmarkt
 für
 Computer u.
 Unterhaltungs-
 electronic in Wetzlar.
 Einkaufszentrum Bahnhofstraße. Tel. (0 64 41) 4 85 66

6350 Bad Nauheim

Computer
 Professional GmbH
 Hauptstr. 92 · 6350 Bad Nauheim
 ☎ 0 60 32/20 88/9

6400 Fulda

Schneider ATARI Commodore

WEINRICH
 BÜRO · ORGANISATION
 Ronsbachstraße 32 · 6400 Fulda
 Telefon: 06 61 / 4 92-0

6457 Maintal

Landolt-Computer
 Beratung · Service · Verkauf · Leasing
 Wingertstr. 112
 6457 Maintal/Dörnigheim
 Telefon 0 61 81-4 52 93

6500 Mainz

:ELPHOTEC
 Computer Systeme
 Ihr Atari Systemhändler
 mit eigenem Service-Center
 Schießgartenstraße 7
 6500 Mainz
 Telefon 0 61 31-23 19 47

SCHMITT
COMPUTERSYSTEME
 Karmeliterplatz 4
 6500 Mainz
 ☎ 0 61 31-23 42 23

6520 Worms

ORION
 Computersysteme GmbH
 Friedrichstraße 22
 6 5 2 0 W O R M S
 Tel. 0 62 41 / 67 57-67 58

6600 Saarbrücken

w.n. pfeiffer
 Büromaschinen KG
computer-shop

Einkaufsführer

6700 Ludwigshafen

MKV Computermarkt

Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

6720 Speyer

MKV Computermarkt

Gilgenstraße 4
6720 Speyer
Telefon 0 62 32 - 7 72 16

6730 Neustadt

Felten & Meier Computersysteme

Exterstr. 4 · 6730 Neustadt
Tel. 0 63 21 / 8 89 94

6750 Kaiserslautern

C.O.S. COMPUTER ORG. GmbH

Karl-Marx-Straße 8
6750 Kaiserslautern
Telefon (06 31) 6 50 61 - 62



6800 Mannheim

Computer Center

Am Hauptbahnhof GmbH
L 14, 16-17
6800 Mannheim
Tel. 06 21 / 2 09 83-4

GAUCH+STURM

Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912

6900 Heidelberg

Heidelberger Computer-Center

Bahnhofstraße 1
6900 Heidelberg
Telefon 0 62 21 / 2 71 32

JACOM COMPUTERWELT

Hardware · Software
Schulung · Service
Mönchhofstraße 3 · 6900 Heidelberg
Telefon 0 62 21 / 41 05 14-550

7000 Stuttgart

BNT Computerfachhandel Seibel & Co. oHG

Der Kleine mit der **großen Leistung**
Markstraße 48 · 7000 Stuttgart 50
Telefon 07 11 / 55 83 83

7022 L.-Echterdingen

Autorisierter ATARI-
System-Fachhändler
für **520 ST 130 XE**



Michael Matrai
Bernhauser Str 8
7022 L. Echterdingen
☎ (07 11) 79 70 49



7030 Böblingen

MCA Computer-Center

Sindelfinger Allee 1
7030 Böblingen
Tel. 0 70 31/22 36 18

7100 Heilbronn

Unser Wissen ist Ihr Vorteil

Walliser & Co.

Mönchseestraße 99
7100 Heilbronn
Telefon 07131/60048

Computer-Welt

Seel's

Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 03

7150 Backnang

Computer-Fans finden bei uns alles von:



7410 Reutlingen

Computer-Shop Werner Brock

Federnfeestr. 17 · 7410 Reutlingen

7450 Hechingen

SRE

Gesellschaft für Datenverarbeitung mbH

Computer · Drucker
Zubehör · Fachliteratur

Schloßplatz 3 · 7450 Hechingen
Telefon 0 74 71 / 1 45 07

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT

HARDWARE · SOFTWARE · LITERATUR SCHEURER

ATARI COMMODORE CUMANA DATA-BECKER
MULTITECH RITEMAN SCHNEIDER THOMSON

7475 Meßstetten 1 · Hauptstraße 10 · 0 74 31 / 6 12 80

7480 Sigmaringen

SOFT&EASY
COMPUTER GMBH
Rapp-Gassle
7480 Sigmaringen
Tel. 075 71/12483

7750 Konstanz

computertechnik
rösler
D-7750 Konstanz
Zasiusstr. 35 · ☎ 075 31/2 18 32

8000 München

Ludwig
COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STR. 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

7500 Karlsruhe

MKV Computermarkt
Rüppurer Straße 2d
7500 Karlsruhe
Telefon 0721-373071

7800 Freiburg

CDS
EDV-Service GmbH
Windausstraße 2
7800 Freiburg

MSG
Marketing u. Service
Autorisiertes Atari-Service Center
Adelmannstr. 5
8000 München 82
Telefon 089/4300333

papierhaus erhardt
Am Ludwigsplatz · 7500 Karlsruhe
Tel. 0721-23925

computer aktuell
Südbadens
kompetenter
Computer-Partner.
Kaiser-Joseph-Str. 232
7800 Freiburg, Tel.: 0761/2180225

8032 Gräfelfing

ProCE
COMPUTER SYSTEME
SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8545464, 851043

7600 Offenburg

FRANK LEONHARDT
ELECTRONIC
Ihr Fachgeschäft für Microcomputer · HiFi · Funk
In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 07751/3094

8070 Ingolstadt

DREYER GMBH
Elektrotechnik
Manchinger Straße 125
8070 Ingolstadt
Tel. 0841/6590

7700 Singen

U. MEIER
Groß- und Einzelhandel
7700 Singen-Htwl.
Postfach 447

7900 Ulm

HARD AND SOFT
COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Herrenkellergasse 16 · 7900 Ulm/Donau
Telefon 0731/2699

8150 Holzkirchen

ATARI
Besuchen
Sie uns!
Fordern Sie
unseren Soft-
ware-Katalog
(520ST) an!
M
MÜNZENLOHER GMBH
Tölzer Straße 5
D-8150 Holzkirchen
Telefon: (08024) 1814

7730 VS-Schwenningen

BÜS BRAUCH & SAUTER
COMPUTER TECHNIK
Villinger Straße 85
7730 VS-Schwenningen
Telefon 07720/38071-72

COMPUTERSTUDIO
Büro & Datentechnik
Claus Wecker
Hafenbad 18/1 + Frauenstr. 28
7900 Ulm/Do.
Telefon (0731) 680761

8170 Bad Tölz

Elektronik Center
Bad Tölz
Wachterstraße 3
8170 Bad Tölz
Telefon 08041/41565

Einkaufsführer

8200 Rosenheim

ZEALOW COMPUTING

Theodor-Gietl-Str. 3 · 8200 Rosenheim · Tel.: 08031/68021

8263 Burghausen

JASKULSKI
Hard- u. Software

Mautnerstr. - 8263 Burghausen
Telefon 08677 / 63320

8300 Landshut

BÜRO-DALLMER

Altstadt 69
8300 Landshut
Telefon 08711 / 21062-64

8400 Regensburg

C-SOFT GMBH
Programmentwicklung & Hardware
Holzfällerstraße 4
8400 Regensburg
Telefon 0941 / 83986

**Zimmermann
elektroland**

8400 Regensburg Dr.-Gessler-Str. 8
☎ 0941/95085

8390 Passau Meraner-Str. 5
☎ 0851/51016

8490 Cham

A&P-SHOP®
INH. A. STEUER

Auf der Schanze 4 · 8490 Cham/Opf. · Telefon (09971) 9723

Ihr Fachgeschäft für Microcomputer und Schreibwaren

Zoom - Fotokopieren

8500 Nürnberg

HIB-GMBH Computerladen
Außere Bayreuther Str. 72
Postfach 21 01 25
8500 Nürnberg 21
Tel: 0911 / 515 939
Telex: 17 - 911 8253 hib
Teletex: 911 8253 HIB



8520 Erlangen

BÜRO 2000 HAAS

Dresdener Str. 5 · Friedrichstr. 9
8520 Erlangen · Telefon 1201-0

Computerservice Decker

Meisenweg 29 - 8520 Erlangen
Telefon 09131 / 42076

8590 Marktredwitz

**STEINWALD
ELEKTRONIK GMBH**
Am Steingrund 1 · 8590 Marktredwitz
Telefon 09231 / 62018

8600 Bamberg

**BÜRO- ZENTRUM
A+R KUTZ**

Bamberg · Tel. 0951 / 27808-09

8670 Hof

COMPUTER-CENTER-BURGER
Spezialist für Personal- und Home Computer, Programme
Zubehör, Beratung, Service
8670 Hof · Leimitzer Straße 11 · Telefon 05281/40075 Abt. Computer

Commodore Atari Corona Alter allvaldi Schneider Tandy Drucker Software Bücher

8700 Würzburg

HALLER GMBH
Fachgeschäft für Mikrocomputer
Büttnerstraße 29
8700 Würzburg
Tel. 0931/16705

8700 Würzburg

**SCHÖLL
computer center**

Dominikanerplatz 5
8700 Würzburg
☎ 0931 / 59354

Systemhändler für Unterfranken
Beratung Verkauf Kundendienst

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 09721 / 652154

8752 Mömbris

COMPTIQUE
DIE EXPERTEN FÜR MICROCOMPUTER

Im Kahital Zentrum · 8752 Mömbris
Telefon: (06029) 6520 oder 1410

ATARI 520 ST APRICOT IBM Komp Festplatten Hardware Software Schulung

8850 Donauwörth

**ORG
technik**

☎ 0906 - 6099
Altes Sträßle 28
8850 Donauwörth
Ihr Büroustatter

8900 Augsburg

Vertragshändler für IBM Personal-Computer

**PROFI
COMPUTER
STUDIO**

KARSTADT Aktiengesellschaft
Bürgermeister-Fischer-Str. 6-10
8900 Augsburg
Telefon (0821) 3153-416

Adolf & Schmoll Computer Studio

Hörbrotstr. 6 · 8900 Augsburg
Telefon (0821) 528533

Wir sind außerdem autorisierte Service-Fachwerkstatt für:

Schneider Atari
Commodore

8901 Augsburg-Vogelsang

VIDEO + COMPUTER HANDELS GMBH

Steppacher Straße 8A
8901 Augsburg-Vogelsang
Telefon 08 21 / 48 20 76

A-8010 Graz

Zupan GmbH

Mandellstr. 23
8010 Graz
Telefon 03 16-70 28 40
78 39 23

CH-8021 Zürich

Senn Computer AG

Militärstrasse 14 · Postfach
CH-8021 Zürich
Tel. 01 / 241 73 73 · Telex 814 193 seco

8910 Landsberg

Szeredy Elektronik GmbH

Computervertrieb

Sandauerstr. 253
8910 Landsberg · ☎ 08191 / 395 00

Schweiz CH-1700 Fribourg

Softy Hard's Computershop

Die ATARI ST Spezialisten

Rue de Lausanne 28
CH-1700 Fribourg
Tel. 0041(0) 37 22 26 28

CH-9000 St. Gallen

VIDEO - COMPUTER - CENTER GÄCHTER AG

Webergasse 22
9000 St. Gallen
Telefon 071 / 22 60 05



8940 Memmingen

EDV-Organisation Hard- + Software Manfred Schweizer KG

Benninger Str. 34, Tel. 083 31 / 1 22 20

8940 Memmingen

CH-2503 Biel

URWA Electronic

Ihr ATARI ST Spezialist in
der Schweiz.
032 / 25 45 53

Lindenweg 24, 2503 Biel

8246 Langwiesen

ZIMELEC

Elektronik-Shop
8246 Langwiesen
☎ 053 55 22 4

Montag-Freitag von
10-20 Uhr geöffnet
Samstag von
10-16 Uhr.

Österreich A-1020 Wien

W + H. Computerhandel Ges m.b.H.

Förstergasse 6/3/2 · 1020 Wien
Tel. 02 22-3 38 45 92

CH-4625 Oberbuchsitzen

STECTRONIC M. Steck

Electronic-Computer-Shop

Hauptstr. 104/137
CH-4625 OBERBUCHSITZEN
Tel. 062 / 63 17 27 + 63 10 27

CH-9400 Rorschach

PAUS-Electronic

Ihr Computer-Fachhandel

Industriestr. 30

CH-9400 Rorschach ☎ 071-41 18 85

CH-9450 Altstetten

PAUS-Electronic

Ihr Computer-Fachhandel

Ringgasse 27 · CH-9450 Altstetten

A-5023 Salzburg

Jedermann Computer

Computerprodukte Ges.m.b.H.

5023 Salzburg, Warwitzstraße 8,
Telefon 06 62 / 7 38 38

CH-8021 Zürich



Das Warenhaus der neuen Ideen

Ihr Computer-Fachhandel an der
Bahnhofstrasse 75 · Zürich

Luxemburg

Ihr Spezialist + Service für

Computer

Commodore
Schneider
Atari

7, av. Viktor Hugo · Luxembourg · Tel.: 20148



ACHTUNG NEUE PREISE !



Computer KÖNIGSTEIN



260 ST.....998,-DM

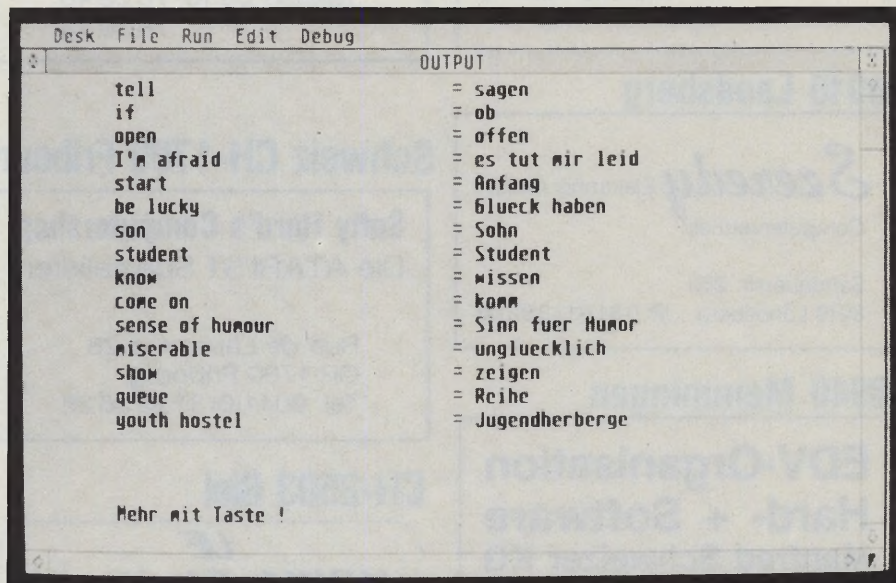
260 ST mit 1 MB.....1.248,-DM

KFC Wiesenstraße 18 6240 KÖNIGSTEIN Tel. 06174/3033 BTX*921069.

Vokabeln lernen mit Spaß

Keine Sprache ist normalerweise ohne das Pauken von Vokabeln erlernbar. Die Wege und Mittel den nötigen Sprachschatz zu erlernen, reichen dabei von Abfragen durch andere Personen bis zu Karteikästen, aus denen wahllos die Vokabelkärtchen gegriffen werden. Das Programm „Vokabeltest“ bietet nun für alle ST-Besitzer eine Möglichkeit eine weitaus luxuriösere Möglichkeit des Lernens und Abfragens von Vokabeln. Mit ihm kann man die Vokabeln in lektionenweise in Direktzugriffsdateien ablegen und somit zum Beispiel nach den Kapiteln eines ganz bestimmten Lehrbuchs vorgehen. In unserem Beispiel verwendeten wir die englische Sprache. Das Programm ist aber natürlich auch für jede andere Sprache, die über den entsprechenden Zeichensatz verfügt, verwendbar. In diesem Fall empfiehlt es sich, das „englisch“ im Listing in die Bezeichnung der betreffenden Sprache zu ersetzen.

Ein weiterer Leckerbissen an diesem Programm ist, daß beim Abfragen die Möglichkeit besteht, die Ausgabe der Vokabeln statt auf dem Bildschirm auf einen Drucker auszugeben. So hat man gleichzeitig eine Kontrolle über



sich und kann später nachkontrollieren welche Vokabeln einem besonders große Schwierigkeiten bereitet haben. Ferner besteht die Option falsch abgespeicherte Vokabeln nachträglich zu korrigieren und wieder abzuspeichern. Am Ende jeder Lektion werden dann die Anzahl der richtigen bzw. falschen Antworten, die der ausgelassenen Vo-

kabeln, der doppelten Eingaben und die Prozentzahl der richtigen Antworten ausgegeben.

Zum Schluß bleibt noch zu sagen, daß das Programm voll menügesteuert ist und somit keiner weiteren Anleitung bedarf. Viel Spaß beim Lernen!!!

(Hennecke & Glitsch)

```

5      fullw 2 :dim ev$(300):dim dv$(300):goto 180
10     VOKABELEINGABE:
20     clearw 2
30     gotoxy 0,0:print "                ***** Vokabeleingabe *****"
40     gotoxy 0,3:print"Ende mit '*'":print "
50     print "Deutsch :
60     print "Englisch:"
70     for i= anzahl+1 to 200-anzahl
80     gotoxy 0,4:print i;". Vokabel"
90     gotoxy 4,5:print"
100    gotoxy 4,6:print"
110    gotoxy 4,5:input dv$(i)
120    if left$(dv$(i),1)="/" then anzahl=i-1:goto 160
130    gotoxy 4,6:input ev$(i)
140    next i
150    print"ende des speichers"
160    return
170    '-----
180    clearw 2:gotoxy 10,0:print"***** Men *****"
190    print:print"<1> Vokabeleingabe      <2> Vokabelausgabe
200    print:print"<3> Vokabeln drucken    <4> Vokabeln berichtigen

```

```

210 print:print"<5> Vokabeln speichern    <6> Vokabeln laden
220 print:print"<7> Datei loeschen      <8> Vokabeln abfragen
230 print:print"                        <9> Ende
240 print:input"Ihre Wahl :","wahl
250 if wahl=1 then gosub VOKABELEINGABE
260 if wahl=2 then gosub VOKABELAUSGABE
270 if wahl=3 then gosub VOKABELAUSDRUCK
280 if wahl=4 then gosub VOKBERICHTIGEN
290 if wahl=5 then gosub VOKSPEICHERN
300 if wahl=6 then gosub VOKLADEN
310 if wahl=7 then gosub DATEILOESCHEN
320 if wahl=8 then gosub VOKABFRAGE
330 if wahl=9 then gosub ENDE
340 goto 180
350 '-----
360 VOKABELAUSGABE:
370 clearw 2:gotoxy 5,0:print "***** Vokabelausgabe *****"
380 if anzahl/2<>int(anzahl/2)then ev$(anzahl+1)="":dv$(anzahl+1)="
390 for i = 1 to anzahl step 2
400 print i;". ";dv$(i);"-";ev$(i),i+1;". ";dv$(i+1);"-";ev$(i+1)
410 if i= 31+k then k=k+31:input"<MEHR VOKABELN>","ant$
420 next i
430 input" < Return > ","ant$
440 k=0:return
450 '-----
460 VOKABELAUSDRUCK:
470 clearw 2:gotoxy 5,0:print "***** Vokabelausdruck *****"
480 input"berschrift:","txt$
490 lprint" ",txt$
500 for i = 1 to anzahl step 2
510 lprint i;". ";dv$(i);"-";ev$(i),i+1;". ";dv$(i+1);"-";ev$(i+1)
520 next i
530 return
540 '-----
550 VOKSPEICHERN:
560 clearw 2:gotoxy 3,0:print" ***** Vokabeln speichern *****"
570 input"Dateienname (*=Ende):","dat$
580 if dat$="*" then 710
590 on error goto 700
600 open"0",#1,dat$
610 on error goto 700
620 print#1,anzahl
630 for i= 1 to anzahl
640 print #1,dv$(i)
650 print #1,ev$(i)
660 on error goto 700
670 next i
680 close #1
690 goto 710
700 print "Fehler !!!!!":for i= 1 to 1000:next:goto 560
710 return
720 '-----
730 VOKLADEN:
740 clearw 2:gotoxy 3,0:print"***** Vokabeln laden *****"
750 print "Alle bis jetzt eingegebenen vokabeln werden geloescht"
760 input"O.K. (j/n)","ant$
770 if ant$ <>"j" then 180
780 for i= 1 to anzahl
790 dv$(i)=" ":ev$(i)= " ":next i
800 input"gesuchter Dateienname:","dat$
810 on error goto 920
820 open"I",#1,dat$
830 on error goto 920
840 input#1,anzahl
850 for i= 1 to anzahl
860 on error goto 920
870 input #1,dv$(i)
880 input #1,ev$(i)
890 next i
900 close #1
910 goto 930
920 print"Fehler !!!!!":for i= 1 to 1000:next:goto 740
930 return
940 '-----
950 Ende:

```

Listing

```

960 clearw 2:gotoxy 7,0:print"***** Ende ****"
970 input"Sind Sie sicher? :",ant$
980 if ant$<>"j" then return
990 end
1000 '-----
1010 VOKBERICHTIGEN:
1020 clearw 2:gotoxy 5,0:print"**** Vokabeln berichtigen ****"
1030 gotoxy 5,3:input"Vokabelnummer?:",vokn
1040 gotoxy 10,5:print"Deutsch :";dv$(vokn)
1050 gotoxy 10,7:print"Englisch:";ev$(vokn)
1060 gotoxy 0,5:input"D.K.(j/n):",ant$
1070 if ant$<>"n" then 1090
1080 gotoxy 10,6:input"Neue Vok:";dv$(vokn)
1090 gotoxy 0,8:input"D.K.(j/n):",ant$
1100 if ant$<>"n" then 1120
1110 gotoxy 10,8:input"Neue Vok:";ev$(vokn)
1120 gotoxy 5,10:input"Noch eine Vokabel berichtigen?:",ant$
1130 if ant$<>"j" then return
1140 goto 1020
1150 '-----
1160 DATEILOESCHEN:
1170 clearw 2:gotoxy 5,0:print"***** Datei loeschen *****"
1180 input"Zu loeschender Dateiname:",datn$
1190 input "Sind Sie sicher? :",ant$
1200 if ant$<>"j" then 1250
1210 print "Datei - '";datn$;"' wird geloescht"
1220 on error goto 1310
1230 kill datn$
1240 goto 180
1250 print"<1> Menue "
1260 print"<2> neuer Versuch"
1270 input"Ihre Wahl:",wahl
1280 if ant=1 then return
1290 if ant= 2 then 1170
1300 goto 1270
1310 print"Fehler !!!!!!!":goto 1170
1320 '-----
1330 VOKABFRAGE:
1340 clearw 2:gotoxy 2,0:print"***** Vokabelabfrage *****"
1350 print"<1> Deutsch - English"
1360 print"<2> English - Deutsch"
1370 print"<3> Ende"
1380 input"Ihre Wahl:",wahl
1390 f=0:r=0:deu=0:eng=0
1400 if wahl =1 then deu = 1:goto 1440
1410 if wahl =2 then eng = 1:goto 1440
1420 if wahl=3 then return
1430 goto 1330
1440 gotoxy 5,5:print"Ende mit '*' "
1450 for i= 1 to anzahl
1460 gotoxy 0,6:print i;". Vokabel"
1470 gotoxy 4,7:print" "
1480 gotoxy 4,8:print" "
1490 if deu = 1 then vok$=dv$(i):vok2$=ev$(i)
1500 if eng = 1 then vok$=ev$(i):vok2$=dv$(i)
1510 gotoxy 4,7:print vok$
1520 gotoxy 4,8:input af$
1530 if af$="*" then 1350
1540 gotoxy 0,10:print" "
1550 if af$<>vok2$ then f=f+1:goto 1580
1560 gotoxy 4,10:print"Richtig!":r=r+1
1570 goto 1590
1580 gotoxy 1,10:print"Falsch! Es haette:";vok2$;" heien muessen!!"
1590 gotoxy 4,11:print"Bisher sind ";int((r*100)/anzahl);"% Richtig! "
1600 gotoxy 4,12:print" und ";int(((f*100)/anzahl);"% Falsch!"
1610 next i
1620 clearw 2:gotoxy 0,0:print"***** Auswertung *****"
1630 gotoxy 4,10:print"Sie haben ";int((r*100)/anzahl);"% Richtig gemacht"
1640 gotoxy 4,11:print" und ";int(((f*100)/anzahl);"% Falsch gemacht "
1650 input"Nochmal:",ant$
1660 if ant$<>"j" then return
1670 goto 1330
1680 '-----

```

Das Resource Construction Set

Wem sind Sie noch nicht begegnet? Diese Dateien mit der Extension **.RSC**. Das RSC steht für eine sogenannte Resource-Datei. In ihr sind unter anderem die Daten für die Pull-Down-Menüs und die Dialog- und Meldefenster enthalten. Um nun diese Resource-Dateien möglichst einfach zu erstellen, wurde das Resource Construction Set geschaffen. Dieses im ATARI-Entwicklungspaket enthaltene Programm bietet die Möglichkeit auf luxuriöse Weise eine Resource-Datei „zusammenzustellen“. Zusammenstellen deswegen, weil man mittels der Maus auf ganz einfache Art und Weise seine eigenen Menüs und Fenster in allen Größen und Mustern zur weiteren Verarbeitung zusammenstellen kann.

Natürlich kann man mit dem Resource Construction Set auch bereits bestehende Resource-Dateien verändern. Ein Anwendungsgebiet dafür wäre zum Beispiel das Eindeutschen von Fehlermeldungen und Menütexten in der doch vorwiegend von der englischen Sprache beherrschten Software.

Doch zuerst wollen wir auf das Erstellen von eigenen Resource-Dateien etwas näher eingehen. Dazu sei noch kurz bemerkt, daß dieser Artikel keine vollständige Anleitung zum Resource Construction Set bietet, da dies zu umfangreich wäre.

Das Programm meldet sich nach dem Laden mit zwei Fenstern und dem Desktop auf dem Bildschirm (siehe Bild 1). Das erste Fenster ist das sogenannte **Resource-Partbox-Fenster**, in dem sich fünf verfügbare Baumarten (Trees) befinden. Dazu sei gesagt, daß das GEM die Daten einer Resource-Datei in einer Baumstruktur erwartet. Das Wort Tree erklärt sich aus dem Englischen (Tree = Baum) und wird wegen der baumartigen Verzweigung der weiteren Module, den sogenannten Objects, so bezeichnet. Die Wurzel bildet eine der Baumarten, von denen aus dann mittels der Objects immer weiter verzweigt werden kann.

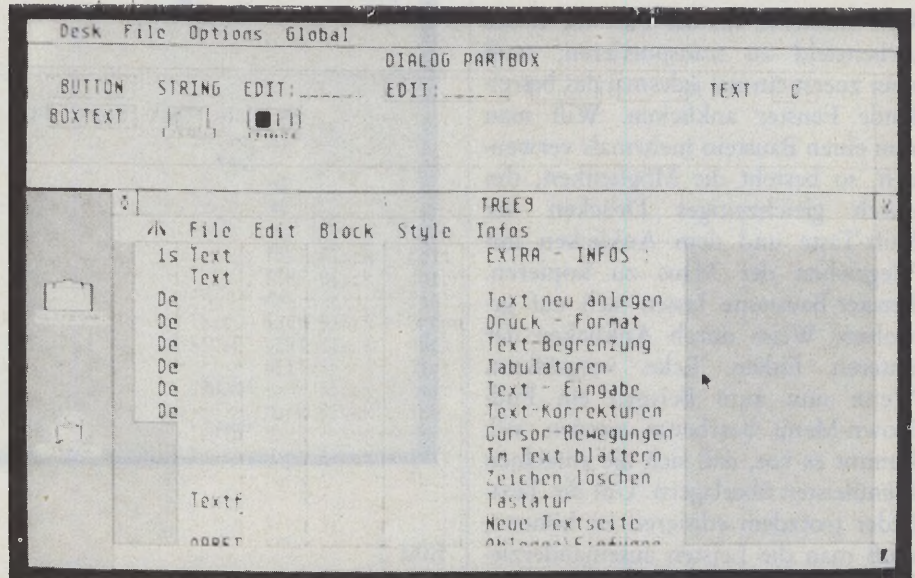


Bild 1

Die verschiedenen Baumarten

Man muß sich jetzt entscheiden, ob man ein Meldefenster (**Alert**) oder zum Beispiel ein Pull-Down-Menü erstellen will. Die dazu benötigten Baumarten werden im folgenden kurz erläutert.

Unknown

Das Fragezeichen dient normalerweise als Platzhalter im Arbeitsfeld, wenn man noch nicht weiß, als welche Baumart man das Feld definieren möchte.

Free

Wie der Name schon sagt, läßt sich mit einem Free-Baum fast ohne Auflagen montieren. Er wird hauptsächlich bei der Erstellung system-spezifischer Bäume verwendet.

Menü

Mit dieser Baumart lassen sich Pull-Down-Menüs erzeugen.

Dialog

Alle Fenster, die eine Eingabe erwarten, werden mittels dieser Baumart erstellt.

Alert

Wie oben schon erwähnt, kann man hiermit Meldefenster konstruieren.

Natürlich kann man mit dem Resource Construction Set auch bereits bestehende Dateien verändern. Ein einfaches Beispiel dafür ist das Eindeutschen von Textsequenzen in fremdsprachigen Programmen, wie auf unserem Bild 2 zu sehen ist. Hier wurden die Pull-Down-Menüs des Textverarbeitungsprogrammes 1st_Word vom Englischen ins Deutsche übersetzt.

Die Mausführung

Hat man sich für eine Baumart entschieden, klickt man sie im Resource-Partbox-Fenster an und „zieht“ sie mit gedrückter Maustaste in das Arbeitsfeld. Läßt man die Maustaste wieder los, erscheint sofort ein Fenster, in dem man seinen Baum benennen kann (siehe Bild 3). Außerdem hat sich noch das Partbox-Fenster verändert. Fast jede Baumart verfügt nämlich über eine eigene Partbox, in der die weiteren Bausteine zum Erstellen des Baumes enthalten sind (siehe Bild 4). Um diese Bausteine in den Baum zu implementieren, muß man diesen zuvor im Arbeitsfeld doppelt anklicken. Hierauf erhält man ein neues Arbeitsfeld, in dem man nun nach Herzenslust montieren und ändern kann. Dazu gibt es

ein paar einfache Tips zur Mausführung, die die Arbeit ungemein erleichtern.

Um Bausteine aus der Partbox in das Arbeitsfeld zu transportieren, muß man zuerst einmal jedesmal das betreffende Fenster anklicken. Will man nun einen Baustein mehrmals verwenden, so besteht die Möglichkeit, ihn durch gleichzeitiges Drücken der Shift-Taste und dem Anklicken mit Wegziehen der Maus zu kopieren. Fenster-Bausteine lassen sich auf gewohnte Weise durch Anklicken der unteren linken Ecke vergrößern. Wenn nun zum Beispiel ein Pull-Down-Menü bearbeitet werden soll, kommt es vor, daß sich die einzelnen Menüleisten überlagern. Um die Textfelder trotzdem editieren zu können, muß man die Leisten auseinanderziehen. Dies geschieht mittels der Maus und der Control-Taste. Mit derselben Methode werden sie nach der Verarbeitung wieder überlagert.

Zur Gestaltung der Fenster und Texte stehen vielfältige Möglichkeiten zur Verfügung (siehe Bild 5), von denen wir einiges kurz erklären wollen.

Selectable

Man kann das Objekt während des Programmverlaufes aktivieren.

Checked

Durch ein Häkchen am linken Rand des Objekts wird angezeigt, ob das Objekt aktiviert wurde.

Editable

Das Objekt enthält einen editierbaren Text.

Default

Dieses Objekt wird durch die Return-Taste ausgewählt. Es erscheint mit einem dickeren Rand auf dem Bildschirm.

Shadowed

Das Objekt erhält einen Schatten.

Touchexit

Diese Möglichkeit erspart ein Anklicken, da durch bloße Berührung des Objekts der Dialog beendet wird.

Exit

Hiermit beendet man einen Dialog mittels Anklicken.

Outlined

Das Objekt wird zusätzlich umrandet.

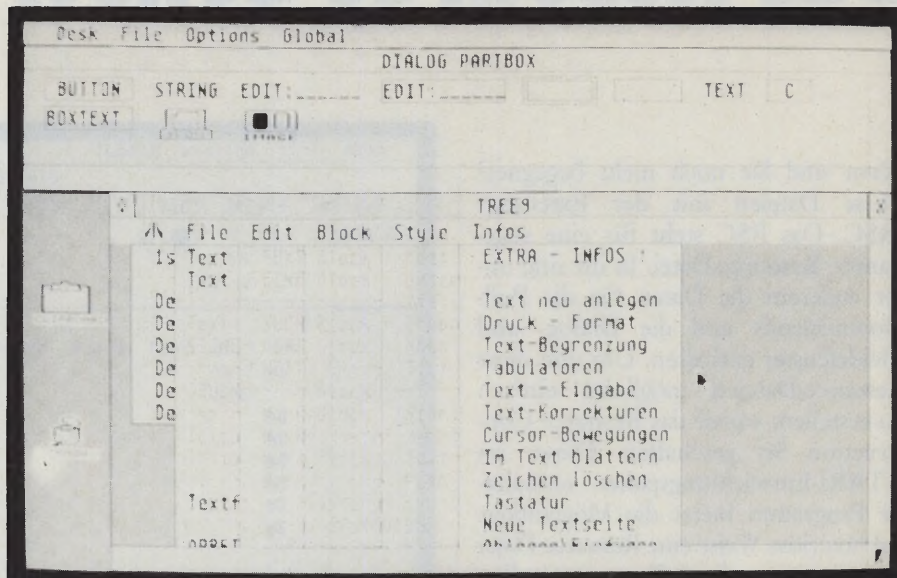


Bild 2

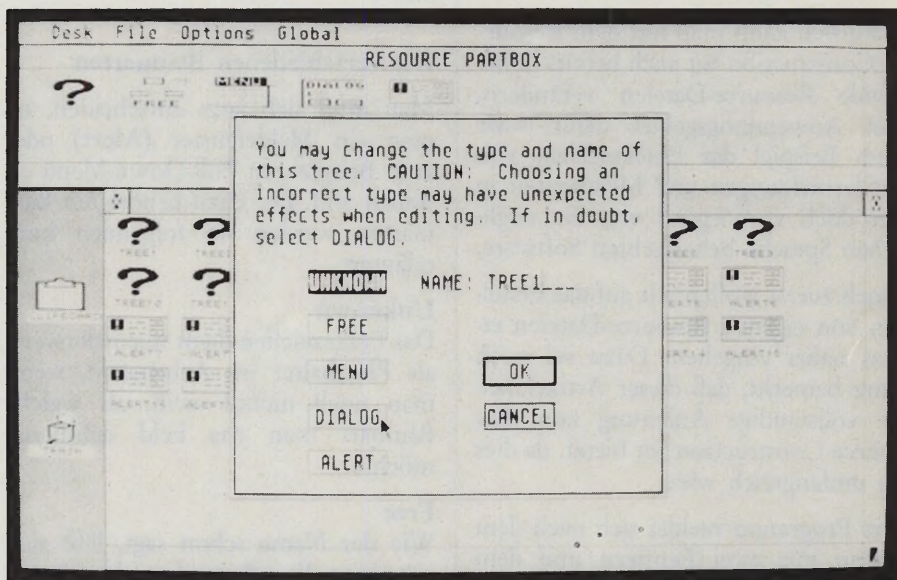


Bild 3

Disabled

Das Objekt erhält graue Schrift.

Radio Butn

Es kann immer nur ein Objekt aus mehreren ausgewählt werden.

Crossed

Hiermit wird eine Box durchkreuzt.

Die erstellten Dateien

Das Resource Construction Set erzeugt beim Abspeichern drei Dateien.

NAME.RSC

Das eigentliche Resource-File.

NAME.H

Eine Include-Datei für C-Compiler, die die symbolischen Konstanten beinhaltet.

NAME.DEF

In dieser Datei legt das Resource Construction Set die Namen der Bäume und Objekte und die Baumarten ab.

Die Dateien mit der Extension .RSC und .H können dann später in die eigenen Programmroutinen eingebunden werden. Die DEF-Datei dagegen wird nur vom Resource Construction Set benutzt. Natürlich muß man in seinem eigenen Programm die nötigen Abfragen und Definitionen selbst programmieren. Das Resource Construction Set nimmt einem nur einen Teil der Arbeit ab.

Will man bereits bestehende Resource-Dateien umändern, wird man feststellen, daß auf der Diskette nur die RSC-Datei vorhanden ist. Das kommt daher, daß die H-Datei bereits in das Programm eingebunden ist und die DEF-Datei – da nicht benötigt – sich gar nicht auf der Diskette befindet. Dies kann man aber ruhig ignorieren. Nach dem Laden der Datei erhält man nur Bäume der Typen Alert und Unknown. Um die Unknown-Bäume zu bearbeiten, muß man sie zuvor doppelt anklicken und am besten in Dialog-Bäume umbenennen. Die Weiterverarbeitung erfolgt, wie gehabt. Machen Sie sich aber vor der Veränderung der Original-Datei eine Sicherheitskopie, da die Ergebnisse der Manipulation nicht immer positiv ausfallen können. Zum Abschluß sei noch gesagt, daß man öfters zwischenspeichern sollte, da das Programm bei unsachgemäßer Bedienung gerne das System zum Absturz bringt.

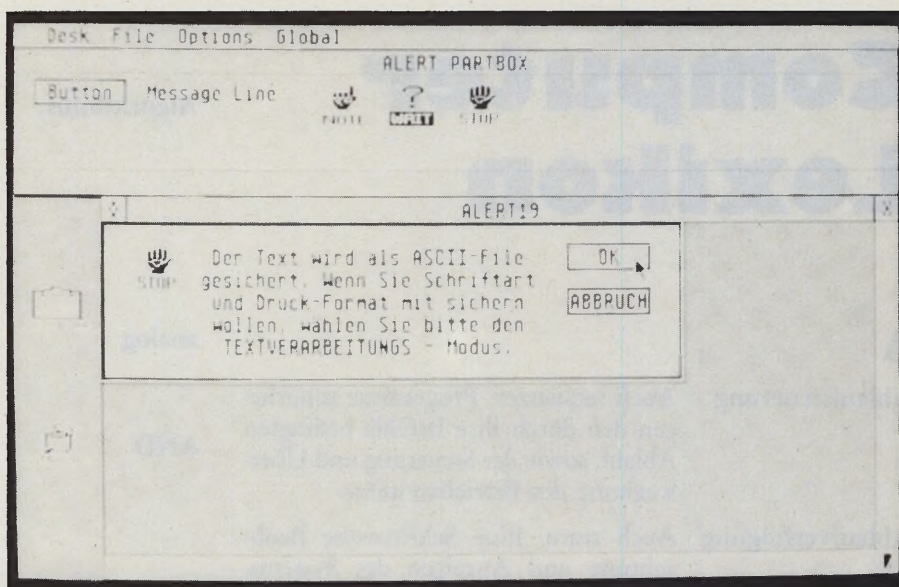


Bild 4

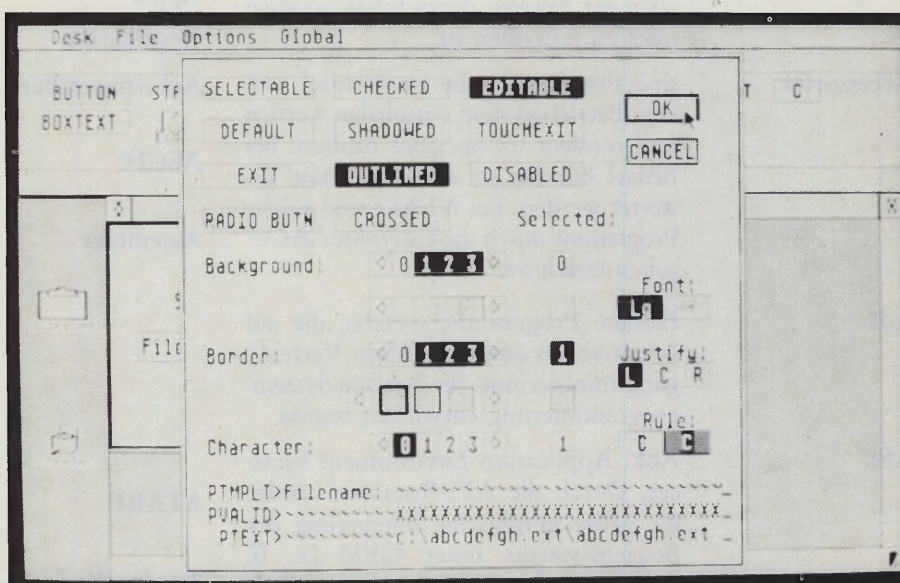


Bild 5

Computer preiswert Computer preiswert Computer preiswert Computer preiswert Computer

Neu! **Farbmonitor** für 260/520 +

nur **699,— DM**

Mit Umtauschgarantie!

Toll! **Panasonic Drucker KX-P1080 NLQ**

nur **699,— DM**

Disketten 5,25 "

— **Sentinel Disketten 2DD 135 TPI**

ab **7,90 DM**

2 D ab 3,50

Profi-Diskmonitor nur **49,— DM**, Endlospapier 12"x 240 mm nur **16,90 DM/1000 B**

Lühr's Computerladen · 2245 Tellingstedt · Hauptstraße 1 · ab 14.00 Uhr · (0 48 38) 679

Computer Lexikon

A

Ablaufsteuerung Auch sequenzer. Programme unterliegen den durch ihre Befehle bedingten Ablauf, sowie der Steuerung und Überwachung des Betriebssystems.

Ablaufverfolgung Auch trace. Eine Schrittweise Beobachtung und Anzeigen des Systems. (z. B. eines Programmablaufes)

access Auch Zugriff. Schreiben und Lesen von Speicherinhalten einschließlich suchen der Adresse, deren Inhalt zu lesen oder zu schreiben ist.

Accessories sind Programme, die automatisch mit dem Betriebssystem eingeladen werden und resident im Speicher bleiben. Bei Bedarf können sie aufgerufen und gestartet werden. Bei ATARI sind solche Programme durch die Extension '.ACC' gekennzeichnet.

Ada Höhere Programmiersprache, die auf Initiative des amerikanischen Verteidigungsministeriums für Realzeit-Systemprogrammierung entwickelt wurde.

AES Abk. Application Environment Services. Durch die AES-Routinen erfolgt die ganze graphische Steuerung des Betriebssystems unter GEM (z. B. Steuerung der Maus und der Pull-Down Menüs).

Adapter Gerät zum Anpassen von Hard- oder Softwarekomponenten, die ursprünglich nicht für den gemeinsamen Einsatz bestimmt waren.

Akkumulator Bei „Ein Adress Maschinen“ ist es notwendig, ein zentrales Register zu haben, mit dessen Inhalt die Operanden verknüpft oder in das das Ergebnis einer Operation abgespeichert werden kann.

Akustik-Koppler Ein Gerät zur Verbindung des Telefonhörers mit einer Datenübertragungseinrichtung, die mit einem Rechner verbunden ist.

Algol Eine höhere Programmiersprache, die zu Beginn der 60er Jahre entwickelt wurde.

Algorithmus

Schrittweise Spezifikation der Lösung eines Problems. Ausgehend von einem bestimmten Problem wird ein Algorithmus zur Lösung entworfen. Dieser erstellte Algorithmus kann dann in ein Flußdiagramm, dieses dann in ein Programm, übertragen werden.

analog

Charakterisiert einen stetigen Bereich von Spannungs- oder Stromstärkewerten. Gegensatz: digital.

AND

Eine logische Operation, die nach folgender Regel definiert wird: sind A und B gleich 1, so ist das Ergebnis gleich 1, ist A oder B gleich 0 ist das Ergebnis 0.

APL

Abk. A Programming Language. Eine höhere Programmiersprache.

Apple

Amerikanischer Computerkonzern.

Applikation

Anwenderprogramm.

Arbeitsspeicher

Ein Speicherbereich, der für vorübergehende Arbeiten zugewiesen wird.

ASCII

Abk. American Standard Code for Information Interchange.

Assembler

1. Assembler ist eine maschinenorientiert Programmiersprache und
2. das jeweilige Übersetzungsprogramm, das die Befehle aus dem Quellprogramm in der Assemblersprache in das Objektprogramm oder die interne Maschinensprache übersetzt (assembliert).

ATARI

Hersteller von Personal-Computern und Telespielen.

Ausdruck

1. (hard copy) Auf Papier gedruckte Ausgabe eines Computer Programms.
2. (listing) Eine gedruckte Darstellung eines Computerprogramms.

AUTOCODE

Eine maschinenorientierte Programmiersprache, die mit Makrobefehlen arbeitet.

autonom

Computer werden als autonom bezeichnet, wenn sie keinerlei Verbindung zu anderen Systemen haben.

B

Band

Preiswertes Massenspeichergerät für die Abspeicherung von Programmen.

Bank

Logische Speichereinheit (normalerweise 64k).

BASF	Abk. Badische Anilin und Soda Fabrik. Hersteller von magnetischen Speichermedien.
BASIC	Abk. Beginner's All-purpose Symbolic Instruction Code. BASIC ist wegen seiner leichten Verständlichkeit und Erlernbarkeit auf fast allen Computersystemen verfügbar. Von System zu System gibt es aber erhebliche Unterschiede, so daß man von vielen sogenannten BASIC-Dialekten spricht.
Batch-Datei	Eine Batch-Datei (Stapeldatei) ist eine ASCII-Datei mit der Extension BAT, die Betriebssystem-Befehle und Programmaufrufe enthält.
Baud	Anzahl von Bits, die in einer Sekunde übertragen werden.
Baustein	Auch Modul. Durch unterschiedliche Kombination lassen sich Systeme für verschiedene Zwecke zusammenstellen.
Befehl	Anweisung die den Prozessor mittels eines Programms zur Abarbeitung von bestimmten Programmroutinen veranlaßt.
bidirektional	Beschreibt die Eigenschaft, daß Daten über eine Verbindung in beide Richtungen übertragen werden können.
bidirektionales Drucken	Es wird sowohl vorwärts als auch rückwärts gedruckt, d. h. es wird zuerst von links nach rechts und dann von rechts nach links gedruckt.
Bildelement	Auch Pixel genannt, kleinste Einheit eines Bildes wie z. B. ein Punkt auf einem Bildschirm.
binär	Zweiwertig, das Wort bezeichnet die Eigenschaft einer Speicherzelle.
BIOS	Abk. Basic Input/Output System. Ein Teil des CP/M-Betriebssystems.
Bit	Kurzform für „binary digit“. Ein Bit kann den Wert 1 oder 0 annehmen. Bits werden zu größeren Einheiten zusammengefaßt. Halbbyte(4), Byte(8) oder Worten(16, 24, 32, 36 oder mehr).
Blank	Leerzeichen.
Booten	Laden eines Betriebssystems.
Bootsektor	In ihm steht ein einleitender Ladesatz, der automatisch beim Formatieren generiert wird. Dieser Ladesatz dient zum überprüfen, ob eine richtig formatierte oder intakte Diskette im Laufwerk ist.

DATENBLATT GFA-Basic für ATARI ST

GFA-eigener Bericht

GFA-Basic-Interpreter für ATARI ST.

- Kompakter Hochgeschwindigkeits-Interpreter
- 11-stellige Genauigkeit
- Strukturiertes Programmieren
- Einfachste GEM-Programmierung
- Komfortabler Editor

Die Daten im einzelnen:

1. Rechengenauigkeit

- 11 Stellen ● Exponenten bis +- 154
- Strings bis 32 767 Zeichen ● 32 Bit Integers (+- 2 Milliarden)

2. Geschwindigkeit

- Hohe Ausführungsgeschwindigkeit
- Schnelle Garbage Collection

3. Graphik

- Ausgefüllte und abgerundete Rechtecke ● Einzelne Punkte ● Raster Graphik (Software Sprites)
- Mausabfrage durch BASIC-Befehle

4. GEM-Programmierung

- Erstellung und Nutzung von Menüleisten durch BASIC-Befehle ● Erstellung und Nutzung von Dialogfenstern durch BASIC-Befehle

5. Editor

- Statt Edit-Window ein zweiter Bildschirm zum Editieren mit Umschaltung auf Knopfdruk, ohne Wartezeit wegen Windowverwaltung ● Bildschirmorientiert, ohne Zeilennummern, Blockoperationen ● Auf Wunsch formatiertes Listing, automatisches Einrücken von Schleifen ● Überprüfung der Programmstruktur ohne Testlauf möglich, so werden z. B. FOR-Befehle ohne das zugehörige NEXT erkannt

6. Strukturierte Programmierung

- WHILE-WEND ● REPEAT-UNTIL ● DO-EXIT-LOOP ● Procedures und Funktionen mit lokalen Variablen ● Formatiertes Listing ● Keine Zeilennummern ● Bit-Variablen (TRUE-FALSE) ● Bit-Packing in ARRAYS

7. Weitere Features

- Interpreterlänge maximal 60 KByte ● Ausführliche Fehlermeldungen in Deutsch und in anderen Sprachen (Englisch, Französisch, etc.) ● Mehrzeilige Funktionsdefinitionen ● Ansteuerung des AES durch BASIC-Befehle (ohne POKE) ● Durch Interrupt gesteuerte Programmausführung (begrenzt Multitasking), z. B. bei Menüauswahl mit der Maus automatische Unterbrechung des laufenden BASIC-Prog. und Ausführung eines BASIC-Unterprogrammes (mit lokalen Variablen) ● Debug-Hilfen ● Mitgeliefert wird ein RUN-ONLY-Interpreter, eine Version des Interpreters, mit dem Prog. NUR ausgeführt, aber NICHT editiert oder gelistet werden können ● Dieses Programm darf FREI kopiert werden, sodaß es möglich ist, Programme in GFA-BASIC zu vertreiben, ohne daß der Kunde den Interpreter selbst erwerben muß ● Lieferbar: spätestens Juni 1986 ● Der dazugehörige Compiler mit vollem Sprachumfang folgt bis Ende des Jahres 1986

Preis für Interpreter und Compiler voraussichtlich je DM 149,-

Burroughs	Amerikanischer Hersteller von DV-Sytemen.		
Bus	Verbindung zwischen mehreren Komponenten, über die Signale ausgetauscht werden. In einem Mikroprozessor befinden sich 3 Busse: Daten-, Adreß- und Kontrollbus.	control	Steuern. Bei einer DV-Anlage Abwicklung der einzelnen Befehle eines Programms einschließlich aller Soft- und Hardware Komponenten.
business Computer	Andere Bezeichnung für Mikrocomputer, welche meistens im Geschäftsreich eingesetzt werden.	coursewriter	Eine Programmiersprache für Pädagogen und Lehrer, die Programme für computerunterstütztes Lernen schreiben.
Byte	1. Ein Byte besteht aus 8 Bits und einem Prüfbit. 2. Bekannte Fachzeitschrift für Computer-Anwender.	CP/M	Abk. Control Program for Mikrocomputers. Ein sehr stark verbreitetes Betriebssystem für 8-Bit Rechner mit den Mikroprozessoren 8080, Z80 und 8085.
C		CPU	Abk. Central Processing Unit (Zentral-einheit).
C	Eine maschinennahe Programmiersprache, die mit dem Betriebssystem UNIX entwickelt wurde.	CR	Abk. Carriage Return (Wagenrücklauf).
CAD	Abk. Computer-Aided Design. Computer unterstützter Entwurf.	Cursor	Markierung, Zeichen oder Zeiger. Laufzeichen für die Kennzeichnung einer Stelle in einem Bildschirm, die gerade beschrieben werden soll. Wird als blinkendes Leerzeichen oder Unterstreichung, auch als Pfeil dargestellt.
CBASIC	Ein BASIC-Compiler für Mikrocomputer. Übersetzt BASIC, statt es wie die meisten Systeme zu interpretieren.	D	
cartridge	Modul.	Datei	Unter Datei versteht man eine Sammlung von Daten mit ähnlichem Inhalt und meist gleichartiger Form, die zusammengefaßt und in einem Speicher untergebracht sind. Das Grundelement einer Datei ist ein Datensatz, der zusammengehörige Daten enthält, die beliebige Elemente beinhalten.
CBM	Abk. Commodore Business Machines.	Datenbank	Eine systematische Organisation von Dateien für zentrale Zugriffe, Suchoperationen und Veränderungen.
Centronic Schnittstelle	Eine Schnittstelle für 8-Bit-Parallelübertragung, die sehr weit verbreitet ist.	Datexnetz	Ein digitales Postnetz für die speziellen Bedürfnisse der Datenfernübertragung. Die Leistung liegt zwischen 200 und 48 000 Baud.
Chip	Kleiner elektronischer Baustein, der als Grundelement von Computer-Systemen verwendet wird.	density	Dichte, Aufzeichnungsdichte.
CMOS	Abk. Complementary MOS. Eine Technologie, die durch ihren geringen Stromverbrauch charakterisiert ist.	Desktop	Benutzeroberfläche des TOS.
COBOL	Abk. Common Business Oriented Language. Eine höhere Programmiersprache für kaufmännische Anwendungen.	Desk-Accessories	Anwenderprogramme die im Pull-Down Menü „Desk“ installiert werden und somit in jedem GEM-Programm aufgerufen werden können.
Code	Ein Code wird als ein Verschlüsselungssystem bezeichnet.	digital	DV-Anlagen werden als Digitalrechner bezeichnet, wenn die interne Zeichendarstellung durch Ziffern z. B. durch das binäre Zahlensystem erfolgt.
COMOL	Abk. Common Algorithmic Language. Eine höhere Programmiersprache die aus BASIC abgeleitet ist.	directory	Das Directory gibt Auskunft welche Programme und freier Speicherplatz sich auf einem Datenträger befinden.
Commodore	Hersteller von Home- und Personal-Computern.		
Compiler	Übersetzungsprogramm, das Anweisungen einer höheren Programmiersprache in eine Menge binärer Anweisungen umwandelt, die dann vom Prozessor durchgeführt werden können. Der Compiler nimmt nur einmal die Umformung in binäre Anweisungen vor. Wird das Programm		

Diskette	Ein Speichermedium in Form einer flachen Scheiben mit magnetischer Oberfläche. Sie befindet sich in einer speziellen Hülle mit einer Aussparung wo ein Lese- oder Schreibkopf auf die Scheibe zugreifen kann.
DOS	Abk. Disc Operation System. DOS ist ein Betriebssystem, daß von IBM entwickelt wurde und weit verbreitet ist.
double density	Doppelte Aufzeichnungsdichte bei Disketten. Je nach Gerät können sie wahlweise mit einfacher oder auch doppelter Dichte beschrieben werden.
double sided	Auf beiden Seiten benutzbare Disketten. Dies ist je nach Gerät oder Diskette nicht immer möglich.
Duplexbetrieb	Übertragung von Daten auf Leistungssysteme gleichzeitig in beide Richtungen, d. h. das System sendet und empfängt zur selben Zeit Daten.

Fortsetzung folgt

KENNEN SIE UNS?

Wir sind führender Hersteller, Entwickler und Vertreiber professioneller Computer Hard- und Software.

Unsere langjährige Erfahrung auf dem Computer Sektor und die Kreativität unseres erfahrenen, gut eingespielten Entwicklungsteams machen uns zu Ihrem Ansprechpartner, wenn es um die volle Ausnutzung Ihres

ATARI ST® Systems geht.

Wir bieten eine große Palette an Hard- und Softwarezubehör an.

Vergleichen Sie selbst:

SOFTWARE:	
- Diskettenmonitor	DM 89,-
- Kalkulator	DM 59,-
- u.s.w.	
HARDWARE:	
- Eprom-Platine 128 KB	DM 99,-
- Steckplatzerweiterung	DM 189,-
- PAL-Interface	DM 298,-
- Epromprogrammiergerät	DM 349,-
	(in verb.)
- Drucker kabel	DM 58,-
- Akustikkoppler	DM 425,-

ST

Fordern Sie noch heute unseren Gratis-

Katalog an!

Jeder Einsender erhält kostenlos ein Heft mit Tips und Tricks zum Atari-ST!

Computertechnik

Z. Zaporowski

Vertrieb in der Schweiz:

Vinckestraße 4
5800 Hagen 1
Tel. 02331/14344
Mo.-Fr., 9.00-13.00,
15.00-17.30

MFS. Sägger, CH-3185 Schmitt
Tel. 037-36 2060

Händleranfragen erwünscht!

LOCATE UND INKEY\$ im ST-BASIC

Die Befehle INKEY und GOTOXY funktionieren im ST-BASIC leider nicht korrekt, was in einigen Fällen zu erheblichen Problemen führen kann. So war eine Tastaturabfrage im laufen Programm bisher nicht möglich, ebenso die korrekte Positionierung des Cursors. Mit diesen zwei kleinen Routinen können diese Fehler beseitigt werden.

LOCATE: Positioniert den Cursor an der definierten Stelle des Bildschirms.

Format: XLOC = .. : YLOC = .. : Gosub LOCATE

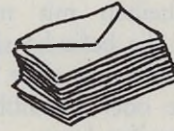
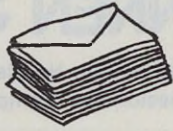
INKEY: Liest ein Zeichen von der Tastatur und übergibt es in TASTE\$, ohne den Programmablauf zu unterbrechen. Falls bei Aufruf des Unterprogramms eine Taste gedrückt ist, enthält die Variable KEY eine Eins.

Format: Gosub INKEY

```

65200 LOCATE:
65202 ' --> XLOC ; YLOC
65204 gotoxy 0,y
65206 modus= 2 :gosub grafikmodus.
65208 print tab(x+1);
65210 modus= 1 :gosub grafikmodus
65212 return
65214 '-----
65216 '
65220 INKEY:
65222 ' <-- TASTE$ , KEY
65224 poke contrl,31 : poke intin ,1
65226 vdisys
65228 key =peek(contrl+8)
65230 taste$=chr$( peek(intout))
65232 return
65234 '-----

```



„Monitor scharf gemacht“

Prima! Sehr erfreulich, wie gut Sie sich auf die Probleme der ST-Käufer einstellen. Die Monitoreinstellungen haben auch mir geholfen. Bis auf einen Bildfehler den ich nicht beheben konnte. Das Bild steht nicht waagrecht, sondern fällt rechts um ca. 10–15 Grad ab. Verstellen der Ferritrings nützt auch nichts. Was soll ich machen?

P.S. Profis schalten ihre Groß-Computer und Terminals nie aus. Um das ständige Laden des TOS zu vermeiden, würde ich auch gerne Computer, Diskettenlaufwerke und Monitor (in Dunkelstellung) Tag und Nacht anlassen. Gibt es irgendwelche Probleme? Patrick Eitner, Bonn

a) Um Ihren Bildfehler zu beheben, müsste man die gesamte Ablenkheit, die auf dem Bildröhrenhals fixiert ist, verdrehen. Diese Arbeit sollte aber nur von einem Fachmann vorgenommen werden, da dabei leicht die Bildröhre beschädigt werden kann (Impulsionsgefahr),

b) Es spricht nichts dagegen, die Geräte eingeschaltet zu lassen. Sie sollten jedoch auf eine ausreichende Kühlung der Netzteile achten (Freistellen der Netzteile genügt) und daß der Helligkeitsregler des Monitors voll zurückgedreht wird.

Die Redaktion

Eure Kurse sind ganz gut und schön, aber eine Frage, wen interessiert schon ein LOGO-Kurs. Es gibt da zwei Gruppen von Anwendern: Die Unternehmer und die Hobbyanwender. Ein Unternehmer hat wenig Zeit und Lust sich auch noch mit einer Programmiersprache herumzuergern. Er wird sich folglich fertige Software kaufen.

Wie ist es aber mit der zweiten Gruppe, den Hobbyanwendern? Wird jemand, der von einem C 64 oder einem CPC aufgestiegen ist, sich wirklich mit LOGO beschäftigen wollen, nachdem er sich in mühevoller Arbeit auf seinem ersten Computer in Basic und vielleicht noch in Assembler eingearbeitet hat? Wohl kaum, denn es gibt ja noch das sehr leistungsfähige ST Basic. Interessanter wäre da wohl ein C-Kurs, da GEM größtenteils in C programmiert wurde.

Christoph Bastuck, Lebach/Hoxberg (Brief sinngemäß gekürzt)

Antwort: Der Logo-Kurs kam ganz einfach deswegen zustande, weil beim Kauf des ST die Sprachen BASIC und LOGO mitgeliefert werden und somit jedem ST-Besitzer zur Verfügung stehen. Außerdem kann diese Sprache doch für den ein oder anderen durchaus interessant sein. Für die Zukunft sind auch Kurse in Assembler und C geplant.

Die Redaktion

Vier gewinnt

Mit welchem Compiler wurde dieses Spiel lauffähig compiliert?

K. Amadeus, Bad Münster

Antwort: Mit dem GST-C-Compiler

Die Redaktion

Wie lang darf das Diskettenkabel maximal sein?

Patrick Eitner, Bonn

Antwort: Wir betreiben seit einiger Zeit ein Laufwerk mit etwas über einem Meter Flachbandkabel, und hatten noch keine Probleme.

Die Redaktion

Ich benutze meinen ATARI ST mit 1st WORD zur Textverarbeitung und betreibe ihn mit einem EPSON FX-85 Matrixdrucker. Wie kriege ich den Drucker dazu, mir in anderthalbzeiligem Zeilenabstand ohne Eingriff ins Programm innerhalb eines Textdokumentes zu variieren?

Im voraus vielen Dank für Ihre Mühe.

Johannes Köndgen, Bargteheide

Antwort: Die Vorgehensweise können Sie dem Artikel 'Druckeranpassung 1st WORD' entnehmen. Dort werden verschiedene Möglichkeiten zum Ändern des Zeilenabstandes beschrieben, die allerdings den Zeilenabstand nur verkleinern. Wir empfehlen den variablen Zeilenvorschub (Beispielsweise n/216"). Dabei können Sie exakt jeden Abstand beim Linefeed einstellen und einem Grafikzeichen zuordnen. Näheres im Druckerhandbuch.

Die Redaktion

Habe beim Kauf des 520ST+ nur TOS, LOGO + BASIC bekommen. Wie komme ich am billigsten zum GEM?

Danke im voraus für Antwort

E. Appenzeller

Sie haben Glück. Aufgrund der Preispolitik von ATARI ist GEM momentan besonders preisgünstig. Sie finden es direkt unter Ihrer Tastatur, wenn Sie das TOS eingeladen haben. Dort ist es sinnigerweise bereits enthalten.

Die Redaktion



في المحلة الأخيرة ST كومبيوتر

انتم كسبتم word 5.1 الالة الدوى من كل حرف

بحرية التبعين والاختيار على الالة الكاسيه
او المطبعه الكهربيه الصغيره

مكتلتى انا عندي حديده في دوله بحريه وانا
بدي اكتب له خطابات باللكومبيوتر

ماذا يجب عمله وحل هذا سئلت عليه مع اللومبيوتر
اني سون اكون سعيد جدا اذا قدتم

ما عدتى اخطوني بعنه المعلومات انى اتمنى

انه قدروا انه تعطونى معلومات منه حيا
الموضوع وانه قدروا على قراءه خطا

وتبدوا على لهذه المشكله شكرا جزيلا

الراسل
يوسف نبيل

Wir freuen uns, daß die ST auch im fernen Ägypten gelesen wird. Die Welt ist bekanntlich klein. Bezüglich Ihrer Frage können wir Ihnen und Ihrem Freund behilflich sein. Ja, jedes beliebige Zeichen ist mit 1st Word und einem grafikfähigen Drucker druckbar. Wie dies funktioniert, können Sie in dieser Ausgabe der ST lesen. Mit ein wenig Geduld lassen sich auch solche komplizierten Zeichen ausdrucken, jedoch treten gerade bei den arabischen Zeichen Schwierigkeiten auf, da diese bekanntlich von rechts nach links geschrieben werden. Jedoch ist auch dieses möglich.

Durch die Steuer-codes:

01, 04, 19, 86 ★ unidirektional revers

wird der Drucker veranlaßt ausschließlich von rechts nach links zu drucken. Jedoch ist die Anpassung sehr kompliziert und erfordert einige Arbeit. Falls Sie aber unbedingt Textverarbeitung in arabischer Sprache betreiben wollen, so experimentieren Sie ein wenig. Viel Erfolg.

Grüße auch an Ihren Freund und an die Sphinx

Die Redaktion

BIETE SOFTWARE

ATARI 520 ST

Hervorragende Programme für Ihren ATARI ST: Diskettenmonitor · Calculator · FLOYD-Monitor · Bundesligatabellenverwaltung · VIP-Professional. Info gegen 2,- DM von JJC, Crispinstr. 4, 46 Dortmund 50

Verkaufe ST-Software. Große Auswahl an Finanz- u. Geschäftsprogrammen z. B. Rendite, Bilanz, Kfz. u.v. mehr. Fordern Sie Liste an. Josef Harand, Fritz-Dächert-Weg 44, 6100 Darmstadt-Eberstadt, Tel.: 0 61 51 / 5 82 97

NEU FÜR ST/ST+E

Mathematische Programme, Grafik UTL, Anwendersoftware uvm. Tel. 0 61 44 / 17 38

Lohn/Einkommensteuerberechnung f. 85 auf 260/520 DM 48 incl. Disk u. Script. Dieter Hartjes, 4040 Neuss, Cranachstr. 29, 0 21 01-46 61 07

Hex-File-Editor unter GEM Disk DM 49,- Hartmut Reck, Herbsdorfer Str. 14, 8221 Nußdorf Tel. 0 86 69 / 66 28

Schreibe Datenbank-Prg in BASIC nach Ihren Wünschen Fuß, Traveredder, Kl. Rönna

Verkaufe GSTC-Compiler Preis 250,- DM. 02932 / 3 84 93

Wer sagt denn es gäbe für ATARI ST keine Software? Größter Katalog geg. 2,- DM in Briefm. von Stofke + P. Rudolph GBR. 56 WPT-1. Postfach 13 06 24

★★★ ATARI ST ★★★ ★★★ METACOMCO ★★★

Macro-Assembler
Linker, Editor nur DM 119,-
Pascal,
Iso-Standard nur DM 269,-
Lattice C-Compiler
IBM-komp. voll. Kernighan +
Richie-Standard, GEM (VDI +
AES-Bibliothek) nur DM 299,-
Porto, Verpackung DM 4,-
★★ D. M. B. Software ★★
Postf. 11 15, 7501 Marxzell 1
Telef. bestellen täglich von
17-21 Uhr. Tel: 072 48-55 75

Biete Software

ATARI-ST-Software in großer Auswahl ab Lager lieferbar. Kostenlose Preisliste anfordern. HEINICKE-ELECTRONIC, Kommenderiestr. 120, 4500 Osnabrück, Tel. 05 41 / 8 27 99, Tx 944 966

SM Lager/Kunden/Rechnung f. DM 380,-. Tel. 0 41 53 / 68 31

CAD für Atari ST DM 1348 aktuelle Spielprogramme sowie Fachliteratur. Info: Buchh. Werner Finke Kipdorf 32, 56 Wuppertal-1

SUCHE SOFTWARE

Superbase für ST und andere Software ges. Tel. 0 21 01-8 01 39

Lottozahlen 6/49 (ZDat. + ZZ) f. 520 ges. Petersen, Putjerr. 28, 2385 Schuby.

Suche Fortran - Zahle Höchstpreis. T. 0 68 21 / 2 56 85

VERSCHIEDENES

SUCHE 260/520 mit Gegen-geschäft verbunden (Keyboard-Organ-etc.) Preisausgleich mögl. Angebote an: H. Schweizer, 6520 Worms, Eisenberger Str. 2

ST Floppystecker & 1 m KA. & Anschl. Pl. 26 DM / Centr.-KA. 45 DM / Scartka. 59 DM / PBL. Domain incl. Disk 19 DM. Tel. 02 09 / 78 53 56

Anwalt sucht Programmierer in C zur gemeinsamen Erstellung eines Anwalts-/Notarprogramms. Tel. Essen 77 00 01

KONTAKTE

ST-Anwender im Raum Neuss + D-dorf ges. 0 21 01 / 5 00 51

Suche ST-Besitzer im Raum ROW/STD zwecks Interessengemeinschaft. 0 42 82 / 51 38

BIETE HARDWARE

Schweiz
ATARI 520+ / 260 neu zu verkaufen zu absoluten Tiefstpreisen!!!
Tel: 0 34 / 34 61 45 93

Schweiz, verkaufe neue ATARI 520ST+, kompl. mit Orig. Floppy u. Monitor Brutto 2995,-. Netto 2 680,-. CH/034 61 45 93 (18⁰⁰21⁰⁰ h)

1 MByte für 520 ST / 260 ST inkl. Einbau OS 2 500,- Robert Di Blasi Hüttelbergstr. 39, A-1140 Wien

030 / 3 95 29 20

Hardware

Software

Reparatur

Carl Computer

Der Floppy Preisbrecher!
3.5" Floppy für ATARI ST 720 kB, eingb. Netzteil, Metallgehäuse, Abm. nur: 240 x 115 x 40 mm, anschlussfertig, Einführungspreis nur 498,- solange Vorrat. Fa. W. Ladz, Erzbergerstr. 27 4790 Paderborn
T. 0 52 51 / 3 63 96,
Liste 1 DM Porto.

Disketten 1 D 6,70 - Laufw. 2 x 1 MByte, Anschlussfertig 989,- Atari auf Anfrage AB-Computer, 5 Köln-41, Wildenburgstr. 21, 02 21 / 4 30 14 42

SF 314 neu - originalverpackt DM 650,- 06 31 / 2 31 93

Günstige Laufwerke:

BASF 6164, 3.5", 1 MB, 349,-
BASF 6138, 5.2", 1 MB, 359,-
Tel. 0 89 / 80 68 23, 18-22 Uhr

TAUSCH

Biete Pascal für Macro-Assembler von Metacomco od. entspr. Software. 06 31 / 2 31 93

Leser antworten Leser

Diese neue Rubrik ist ein Teil unserer gemeinsamen Gestaltung. Jeder der Lust und Kenntnis hat, kann seine Antwort oder auch Frage zu uns schicken. Wir werden dann die Besten bzw. Wichtigsten veröffentlichen. Um Ihnen das Antworten auf eine Frage schmackhaft zu machen, verlosen wir unter den abgedruckten Antworten eine Diskettenbox für 50 3 1/2-Zoll Disketten. Wir hoffen, daß dadurch denjenigen, die verzweifelt an einem Problem hängen, eine Möglichkeit gegeben wird, diese zu lösen.

Meine Frage ist: Kann ein Laufwerk Teac 55 G oder ein Teac 55 GF an den ST angeschlossen werden? Wie muß das GF-Laufwerk eingestellt sein, damit es wie ein F-Laufwerk arbeitet?

Harald Thienel

Ich benutze meinen Atari 520+ mit 1ST_WORD zur Textverarbeitung und betreibe ihn mit einem EPSON FX-85 Matrixdrucker.

Wie kriege ich den Drucker dazu, mir

in anderthalbzeiligem Zeilenabstand auszudrucken? Ist es vielleicht sogar möglich, den Zeilenabstand ohne Eingriff ins Programm innerhalb eines Textdokuments zu variieren?

Johannes Köndgen

Trotz neuestem TOS ist mein Nadel-drucker DRH 80 von Triumph Adler mit RS 232 C (V24 - DIN 66020) Schnittstelle nicht zu überreden, mit LPRINT oder LLIST etwas auszudrucken.

Da hilft auch keine Drucker-Einstellung und RS 232 Konfiguration. Wer kann helfen?

Fritz Steube

Leider hat ja das wohlgeleitene BASIC so seine Macken, u. a. keinem INKEY \$. Die in dem Programm benutzte Abfrage E\$ = INPUT \$(1) öffnet ärgerlicher Weise für einen Moment das COMMAND-Fenster.

Abhilfe schafft hier die Abfrage E\$ = CHR\$(INP(2)) die die Tastatur direkt nach der Eingabe eines ASCII-Zeichen abfragt und das graphische Gesamtbild nicht nach jedem Tastendruck zerstört.

Hans-Christian Gondolan

Ich hoffe, daß Sie noch nicht mit Leserfragen überlastet sind, und mir freundlicherweise folgendes Problem lösen: Ich besitze einen Loewe Opta (MCP 114), Farb Portable, mit RGB Ausgaben. Die Frage ist: Wie wird der „alte 520 ST“ angeschlossen?

Manfred Salomon

Bitte bringen Sie in Ihrer Zeitschrift auch wie man Daten von anderen Computern (Ich habe einen Comodore 8296, welcher aber hauptsächlich im 8032 Modus läuft) auf den ST übertragen und dann auf der Floppy speichern kann. Aber bitte so schön ausführlich und verständlich wie ihre Anleitung für das Druckerkabel in Heft 1.

Klaus Rapp

Meine Bemühungen, Turbo-Pascal zu erstehen, schlugen bisher fehl; daher frage ich Sie, ob Sie mir eine Bezugsadresse und evtl. auch einen Preis für dieses Programm nennen können.

Kay Steffen

Mit 5 Mark sind Sie dabei!

Die ST-Redaktion bietet für Sie ab sofort einen neuen Service an. Dieser wurde durch die große Anzahl von Leserbriefen angeregt. Für nur 5,- DM Unkostenbeitrag **pro Diskette** für das Kopieren und Bearbeiten, bieten wir folgende Public Domain Programme als Kopierservice an:

1. Diskette
 - Neochrome V.0.6: luxuriöses Malprogramm für alle Farbmonitorbesitzer
 - Dr. Doodle: Malprogramm für monochromen Monitor

2. Diskette
 - Joshua: Monitor und Diskmonitor
 - CP/M Emulator (CP/M 80)
 - Megaroids: Spiel für Monochrom- und Farbmonitor
 - Diskformatter: erweitert Diskkapazität auf 399 bzw. 809 kByte
 - Kartei-Kasten: luxuriöses Karteiprogramm unter GEM
 - 50/60 Hz-Umschaltung für Monitor im Autostart-Ordner

Zwei weitere Disketten haben wir in unseren Public-Domain Kopierservice aufgenommen. Es handelt sich dabei um ein komplettes **FORTH** System (volksFORTH-83), entwickelt von der **Forth-Gesellschaft eV** in Hamburg. Das System beinhaltet einen Screen-Editor, Interpreter, Compiler sowie einen vollständigen 68000 Assembler. Diskette 3 und 4 gehören beide zu dem System.

3. Diskette
 - Assembler
 - Editor
 - Tools
 - Tasker
 - Printer

4. Diskette
 - Grundsystem
 - Arbeit-System
 - Sys. Quelltext
 - Copyall (ein sehr empfehlenswertes Copy-Programm)

Für eine problemlose und schnelle Bearbeitung müssen Sie folgendes beachten:

- Schicken Sie uns nur einseitig formatierte Disketten.
- Außerdem einen ausreichend frankierten Rückumschlag (DM 0,80 reichen nicht), auf dem Ihre Adresse steht.
- In Ihrem eigenen Interesse verwenden Sie bitte gepolsterte Umschläge mit der Aufschrift „Datenträger“.
- Außerdem vermerken Sie bitte auf dem uns zugesandten Brief „Public Domain“.
- Legen Sie pro Diskette DM 5,- als Scheck bei.

Die Bezugsadresse lautet: Uwe Bärtels · ST Redaktion
Postfach 1131 · 6242 Kronberg

Kleinanzeigen

Privatanzeigen kosten pro Zeile und Spalte DM 5,- incl. MwSt.

Gewerbliche Anzeigen kosten DM 7,- pro Zeile und Spalte.

Zur Einsendung bitte die beiliegende Karte verwenden.

Einsendeschluß ist der 2. April 1986.

Verspätete Einsendungen kommen in die Juni-Ausgabe.

ACHTUNG!

Nachlieferung von ATARI ST-Zeitschrift Nr. 1

Premierenausgabe

Der HEIM-Verlag druckt dieses Heft wegen der großen Nachfrage nach.

Beachten Sie bitte die folgenden Hinweise, wenn Sie dieses Heft noch haben wollen:

1. Sie sind Abonnent und haben Heft 1 bestellt

Dann erhalten Sie es mit der nächsten ABO-Lieferung.

2. Sie sind kein Abonnent, haben Heft 1 jedoch bereits bestellt oder wollen es noch bestellen:

Dann übersenden Sie bitte einen Scheck in Höhe von 7,- DM. Nach Eingang des Betrages wird Ihnen die Nr. 1 nach Drucklegung kostenfrei übersendet.



- **Große Softwareübersicht**
Wir sagen Ihnen welche Programme, wo und zu welchem Preis zu bekommen sind.
- **CEBIT**
Wir waren dabei, ein Bericht über alle Neuigkeiten.
- **C-Kurs**
Ein ausführlicher Kurs mit vielen Programmbeispielen.
- **Betriebssystem (Teil 2)**
- **Terminalprogramme im Test**
- **ST der Zeichenkünstler**
Malprogramme im Vergleich

Die Maiausgabe erscheint am 25.4.86

Impressum

ST-Computer

Herausgeber: Heim Fachverlag,
Heidelberger Landstraße 194,
6100 Darmstadt 13,
Telefon (06151) 5 53 75

Verlagsleiter: H.-J. Heim

Vertrieb: H.-J. Heim
U. Heim
H. Schultheis

Redaktion: Uwe Bärtels (UB-Chefredakteur),
Harald Schneider (HS), Marcelo Merino
(MM), Harald Egel (HE)

ST-Redaktion: Uwe Bärtels, Postfach 11 31,
6242 Kronberg

Redaktionelle Mitarbeiter: F. Köstel,
Markus Nerdling (MN), K. Schultheis

Titelseite: Klaus Ohlenschläger

Anzeigen: Anzeigenleiter U. Heim,
Telefon (06151) 5 53 75

Anzeigenpreise nach Preisliste Nr. 1
gültig ab 1.1.86

Erscheinungsweise: 11 x jährlich

Bezugspreis: Einzelheft DM 6,-.
Jahresabonnement DM 60,- inklusive der
gesetzlichen Mehrwertsteuer und den Zu-
stellgebühren für 11 Ausgaben.

72,- DM inkl. Versand
(Ausland, Normalpost)

93,- DM inkl. Verand
(Ausland, Luftpost)

Bezugsmöglichkeiten: ATARI-Fachhändler,
Zeitschriftenhandel, Kauf- und Warenhäu-
ser oder direkt beim Verlag unter obiger
Adresse.

Druck: Ferling Druck Darmstadt

Manuskripteinsendungen: Programmlis-
tings, Bauanleitungen und Manuskripte
werden von der Redaktion gerne angenom-
men. Sie müssen frei von Rechten Dritter
sein. Mit ihrer Einsendung gibt der Verfasser
die Zustimmung zum Abdruck und der
Vervielfältigung auf Datenträgern im Heim
Verlag. Honorare nach Vereinbarung. Für
unverlangt eingesandte Manuskripte wird
keine Haftung übernommen.

Urheberrecht: Alle in der ST-Computer er-
schienenen Beiträge sind urheberrechtlich
geschützt. Reproduktion gleich welcher
Art, ob Übersetzung, Nachdruck, Vervielfäl-
tigung oder Erfassung in Datenverarbeitungs-
anlagen sind nur mit schriftlicher
Genehmigung des Heim Verlages erlaubt.

ATARI® ist eingetragenes Warenzeichen
der Atari Corporation.

GEM® ist eingetragenes Warenzeichen
der Digital Research.

Veröffentlichungen: Sämtliche Veröffent-
lichungen in ST erfolgen ohne Berücksichti-
gung eines eventuellen Patentschutzes, auch
werden Warennamen ohne Gewährleistung
einer freien Verwendung benützt.

Haftungsausschluss: Für Fehler in Text, in
Schaltbildern, Aufbausketzen, Stücklisten
usw., die zum Nichtfunktionieren oder evtl.
zum Schadhaftwerden von Bauelementen
führen, wird keine Haftung übernommen.

© Copyright 1985 by Heim Verlag.

hardware
software
organisation
service



Heeper Str. 106-108, 4800 Bielefeld 1, 0521/61663

Kein Kabelsalat mehr mit dem Gehäuse für ATARI ST

**DM
198,-**
inkl. MwSt.



- Zentrale Stromversorgung für alle Geräte einschl. 2 Drucker
- Einbaumöglichkeit von 2 Diskettenlaufwerken
- Rechner (Tastatur) kann komplett unter das Gehäuse geschoben werden (Staubschutz)
- Massives Blechgehäuse

ATARI ST-Gehäuse erhalten Sie bei den autorisierten Fachhändlern

Wir haben nicht nur den Computer,
wir haben auch die entsprechende Software:

- Crossreferenzprogramm f. BASIC	65,- DM
- Diskmonitor	65,- DM
- Dateiverwaltung	490,- DM
- Adressverwaltung	490,- DM
- Textverarbeitung	120,- DM
- Finanzbuchhaltung	700,- DM
- Lohnprogramm	640,- DM
- Auftragsbearbeitung	655,- DM
- Bestellwesen	600,- DM
- Branchenpakete je nach Umfang	

Wir beraten Sie gern.
Telefon 0 80 31-6 80 21

**SZEREDY
COMPUTING**

SZEREDY COMPUTING GmbH · Theodor-Gietl-Str. 3 · 8200 Rosenheim

MEGA-ATARI
PERSONALCOMPUTER DES JAHRES
1985

SOFTWARE FÜR ATARI ST

Produkt	VK-Preis	Produkt	VK-Preis
Adress-Perfect läuft unter GEM, suchen & sortieren nach allen Feldern, Schnittstelle zu beliebiger Textverarbeitung, SERIENBRIEF-SCHREIBUNG	148,-	TBC-LINT LINT - Das Präparat gegen Typenkonflikte, schlechte Argumentübergabe, Variablenmüll, Strukturfehler usw. Gegenanzeige: Nicht anwenden bei Allergie gegen saubere Programmierung.	298,-
Side-Click mit Taschenrechner, Kalender, Uhr Terminplanung und Überwachung usw.	198,-	The Best C Lint	
VT 100 unter GEM mit Upload/Download, Protokolldruck und voll VT 100 kompatibel	248,-	TBC-Compiler und LINT	748,-
C-Library Standard	128,-	ST System-Monitor	298,-
C-Library Extra	128,-	Profi-Monitor Direkt Assembler, Disassembler, Debugger, Hex-Ascii-Binär-Editieren	
C-Library Grafik	128,-	Profi-Assembler Def preiswerte, komfortable ST Assembler	198,-
C-Library GEM 1	128,-	Editor-Disk läuft unter GEM, muß man einfach haben!	148,-
C-Library GEM 2	128,-	ARST Für TBC und Entwicklungspaket!	98,-
C-Library Resource Die große Bibliothek für C-Programmierer näheres siehe INFO 1/86	128,-	Disc-Monitor Der bewährte, jetzt noch besser!	178,-
C-Library UNIX 1	158,-	Disketten-Monitor	248,-
C-Library UNIX 2	158,-	Fastos Cartridge DOPPELTE GESCHWINDIGKEIT bei allen Floppy-Arbeiten	
TBC-Compiler inkl. Assembler, Linker & UNIX Libraries - Verarbeitet UNIX Sourcecode - Optimiert wie kein anderer!!! Softwareentwicklung auf ST - nur mit TBC	498,-	File-Copy + Wie oben, jedoch für Files. Wahnsinnig schnell!!!	98,-
		Drucker-Disk Drucker-Treiber, Drucker-Spooler, Druck-Programm	98,-

Fordern Sie unsere Info an!



RDS SOFTWARE
KNOW HOW IS OUR BUSINESS

Jakobstraße 8 a · 6096 Raunheim · ☎ (0 61 42) 4 31 42

GENERALVERTRETUNG
SCHWEIZ:

SAMA Electronics
Bahnhofstraße 7
CH-5400 Baden

Alle Preise sind empfohlene
Endverbraucherpreise incl.
14 % Mehrwertsteuer. Änder-
ungen vorbehalten. Nähere
Informationen erhalten Sie
bei Ihrem ATARI-Händler
oder direkt bei uns.

FORTH-SYSTEME ANGELIKA FLESCH

● FORTH-System

4 x FORTH

- Level 1 – extrem schneller FORTH-Compiler 32 Bit **498,- DM**

Dieser Forth-Compiler ist extrem schnell (s. Testbericht ANTIC 12/85). Er bearbeitet ca. 100 000 Leerschleifen pro Sekunde, übersetzt ca. 20 KB-Source in 3 sec. incl. Linking und unterstützt Multitasking sowie Multiuser-Fähigkeiten.

Andere Software für den ATARI ST

● H & D BASE

348,- DM

In FORTH geschriebenes dBase II kompatibles Daten- u. Dateiverwaltungs-System mit Ausnützen der ATARI-spezifischen Möglichkeiten (Windows, Maus etc.). Unterstützt den kompletten Speicherbereich. Volle Datei-Kompatibilität.

● ST-Colouring-Book

125,- DM

Pictures für Neochrome (2 Discs)

Ausführliche Information bei

FORTH-SYSTEME ANGELIKA FLESCH

7820 Titisee/Neustadt

Postfach 12 26

Tel. 0 76 51 / 16 65 + 33 04

Händler-Anfragen erwünscht.