

So, jetzt ist es soweit. Ein großer Schritt ist vollbracht (zumindest für mich). Ihr habt mit dieser Diskette die aktuellen Versionen der Programme *DiskWorks*, *DataBrowser* und *CalCula* vorliegen. An dieser Stelle möchte ich noch mal darauf hinweisen, daß es sich bei den



beiden letztgenannten Programmen nur um Vorabversionen (0.5 und 0.9) handelt, an denen ich noch arbeite. Aber Ihr könnt **auch** schon damit arbeiten und mir Anregungen und Fehlermeldungen mitteilen. Ich bitte sogar darum!

Dann liegt der Diskette eine Datei bei, die FKEY heißt. Das ist die Definitionsdatei für die Funktionstasten in Basic. Sie müssen mit BLOAD "S2:FKEY",#Bank,&Adresse geladen werden. #Bank und &Adresse hängen von Eurem Speicherausbau ab. Standard ist #0,&C008 bei keiner Basic-Speichererweiterung (11 KB), RAM-Disk oder Speichermodulen, die keinen Einfluß auf den Basic-Speicher nehmen. Bei allen anderen Konfigurationen müßt ihr die Bank und die Adresse selbst eintragen. FKEY ist nur ein Vorschlag. Ich habe allerdings lange daran gebastelt, bis ich sie so hatte, daß - so denke ich - man am besten damit arbeiten kann.

Und dann ist da noch... Nein, nicht *ist* sondern *sind* da noch Dateien mit der Endung .CFG. Wer die Anleitungen der Programme schon gelesen hat, weiß bereits worum es geht. Es sind die Konfigurationsdateien für die Makro-Definitionen. Denn ab den vorliegenden Versionen aller KiKiSoft Programme gibt's jedes Programm nur noch mit Makro-Interpreter. Das erspart mir **Aben** möchten, weil sie bestimmte wenn sie wiederum KiKiWare verlas- mehr an Flexibilität in der Benutzung der KiKiWare. Es ist nun möglich, von DW aus eine Datei zu edieren, den Editor zu verlassen, (wenn FKEY geladen) durch Drücken von F6 (&) wieder DW zu starten, "R" zu betätigen und damit CalCula zu laden, ein paar Berechnungen durchzuführen, "q" (kleines Q gibt's mit Shift und "Q") zu drücken, damit wieder DiskWorks zurückzuholen, eine BAK-Datei zu löschen... und jetzt könnte ich noch mal eine Seite so schreiben, das dürft Ihr dann im Geiste selbst ergänzen. Lange Rede, kurzer Sinn: KiKiWare ist jetzt sozusagen untereinander 'verbunden'. BLOAD, LOAD etc. gehört der Vergangenheit an. Zumindest das ~~manuelle~~ Eingeben dieser Befehle könnt Ihr größtenteils vergessen. Nur wenn Ihr im Basic seit, müßt Ihr noch Basic-Befehle kennen und eingeben. Die beiliegenden CFG-Dateien sind nur Beispiele und können mit jedem Standard-TextEditor (z. B. Text+) individuell verändert werden.

Christian Becker	ab 1. 9. 1993
Hörder Str. 282	Goebenstr. 10
58454 Witten	44135 Dortmund

Eine ausführliche Beschreibung der Makro-'Sprache' findet Ihr in der Beschreibung von DiskWorks. Für DataBrowser und CalCula gibt es nur eine Einschränkung: da keine Dateinamen zur Verfügung stehen, kann der Befehl "@" der Makro-Sprache nicht angewendet werden. Ansonsten alles wie in DiskWorks. Sollte es einmal notwendig sein, ein Basic-Programm ablaufen zu lassen, so kann jeder selbst bei Beendigung desselben dafür sorgen, daß DW (meine Empfehlung) wieder gestartet wird.

Auf der Diskette ist noch eine kleine Uhr, CLOCK benannt. Einmal gestartet, gibt sie auf Betätigen der Shift-Tasten die aktuelle Uhrzeit (des Rechners) aus (von jedem Programm aus). Clock verschwindet wieder, wenn der Rechner aus Basic mit OFF ausgeschaltet wurde. KiKiSoft-Programme entfernen Clock beim Ausschalten des Rechners nicht (CALL 5 aus Basic auch nicht).

Damit jeder überprüfen kann, ob er mir noch 1000 Deutschmark auf mein Konto überweisen kann (KiKiWare ist eigentlich mehr wert), liegt der Diskette noch das Programm FINANZ.BAS (leider nur in BASIC) bei. Es bietet mehrere finanzmathematische Funktionen an. Schaut nach, ob Ihr es gebrauchen könnt. Vielleicht bastel ich sogar noch mal ein wenig daran herum, damit es (Benutzer-) freundlicher wird.

Da auch ich nicht unfehlbar bin, kann sich jeder, der Probleme mit meinen Programmen hat, selbstverständlich auch an mich wenden. Ruft mich an oder - noch besser - schreibt mir eine kurze Notiz. Zu jedem Teil von KiKiWare gibt's in jedem Falle eine Anleitung (FINANZ.BAS ist nicht KiKiWare). Eine Haftung für eventuell damit entstandene Schäden aller Art schließe ich allerdings aus.

Ja. Und was gibt es sonst noch zu sagen? Ich glaube nichts mehr außer: viel Spaß und Erfolg mit KiKiWare by KiKiSoft!

Euer Christian (Nur wo KiKiWare draufsteht ist auch KiKiWare drin!)

DiskWorks V2.0

Hier nun eine Anwenderhilfe für DiskWorks (im folgenden des öfteren mal DW genannt). Diskworks ist eine Benutzeroberfläche für den Sharp PC-1600.

Mit dem Programm können die leidigen und relativ unkomfortablen Disketten-Befehle (Kill the Wildcards!) gestrost vergessen werden. Selbst das Wissen über ganze Befehlsfolgen kann - nachdem es einmal definiert wurde - in den Papierkorb wandern.

Mit Diskwoks können auf einfache Weise Dateien kopiert, gelöscht, umbenannt, geschützt und angesehen werden. DW formatiert und kopiert Disketten im ganzen.

Als besonderes Bonbon können mit DiskWorks alle möglichen Programme (Basic und Maschinensprache) auf unterschiedliche Weise gestartet werden. DW arbeitet mit dem Diskettenlaufwerk CE-1600P und allen kompatiblen RAM-Disks ohne Probleme zusammen (von mir mit 32, 64 und 128, von Gernot mit 256KB getestet). Die Kopierfunktionen unterstützen sogar die serielle Schnittstelle (kann auch von DiskWorks aus konfiguriert werden).

Drüber hinaus bietet DW eine Info-Funktion, die Auskunft über das angewählte Laufwerk (freier Platz, Anzahl Dateien), den Zustand der Batterien und das Datum und die Uhrzeit gibt.

Wer einen IBM-kompatiblen PC sein sein Eigen nennt, der kann von mir noch ein Programm zur Sicherung von Daten und Programmen auf den PC (über die serielle Schnittstelle) bekommen. Es ist ebenfalls von KiKiSoft entwickelt und hat im Gegensatz zu den einschlägigen Programmen von YellowComputing und ECPS den Vorteil, nichts zu kosten (Kabel sind allerdings selbst zu löten)!

DiskWorks liegt nun in der Version 2.0 vor. Gegenüber den Versionen 1.0 und 1.1 haben sich einige große und auch kleine Dinge geändert.

Die Input-Routine bei der Eingabe von Dateinamen etc. wurde überarbeitet und funktioniert jetzt korrekt. Bei Fragen nach ja nein, oder bei der Aufforderung eine beliebige Taste zu betätigen blinkt jetzt ein Cursor. Das macht das ganze etwas verständlicher. Man weiß jetzt, wann der PC1600 etwas vom Anwender erwartet.

Die Makro-Sprache (siehe DW.CFG) wurde um zwei wesentliche Features erweitert.

Und last but not least wurde ein wenig an der Copy-Routine geschraubt. Ich kann jetzt auch beim Kopieren den Namen des zu kopierenden Files ändern. Das hat sich bei mir manchmal als nützlich herausgestellt.

DW beginnt bei der Adresse &D000 und belegt auf einer RAM-Disk DW ca. 6000 Byte hinzu kommt noch eine Datei DW.CFG, die selbst zu erstellen ist bzw. seinen Bedürfnissen anzupassen ist. Kurze Basic-Programme werden somit selbst bei keinen zusätzlichen RAM-Karten nicht beeinflusst.

DW.CFG wird von DW ab &CC00 abgelegt. Für jede Datei

auf dem Quell-Laufwerk werden 19 Byte benötigt (hinter DW). Das bedeutet, daß bei gesetztem MAXFILES=1 ca. 130 Dateien auf dem Medium sein dürfen. Mehr Informationen dazu in einem späteren Abschnitt. An dieser Stelle sei nur gesagt, daß zum Betrieb von DW die Datei DW.CFG auf dem Laufwerk S2: vorhanden sein sollte. Anderenfalls gibt DW beim Start die Fehlermeldung 152 (Datei nicht gefunden) aus und Makros können nicht verwendet werden

Aber zunächst noch mal ein Wort zu den Mindestanforderungen für DW. Da es als Benutzeroberfläche gedacht ist, um Programme zu starten, zu kopieren, ihnen andere Namen zu geben, sie zu löschen und viele andere nützliche Funktionen enthält, welche die tägliche Arbeit mit dem PC-1600 vereinfachen, muß es auf jedem System laufen. Also auch auf dem Grundsystem ohne irgendwelche anderen Programme.

Das tut es (bis auf wenige Ausnahmen) auch. Fast alle Funktionen können verwendet werden wenn kein Speichermodul im PC-1600 steckt. Die Funktion zum Kopieren von Disketten bildet hier die Ausnahme. Da der Kopiervorgang in fünf Schritten erfolgt, müssen die 64KByte der Diskette auch irgendwo im Speicher Platz finden (jeweils durch 5 dividiert).

Den notwendigen Speicher sucht sich DW dann auch, und zwar auf dem Speichermodul S1: und dort im Speicherbereich von &80C5 bis &BFFF der Bank 0. Wenn das Modul nicht da ist, geht das natürlich in die Hose. Wenn dort wichtige Daten stehen, sind sie nach dem Kopieren der Diskette weggeputzt. Also hier ist Vorsicht geboten.

Ich bin einen weiteren Kompromiß eingegangen, den ich nicht verschweigen möchte. Es geht um das Kopieren von Dateien von Laufwerk X: auf das Laufwerk Y:.. Das geht leider nicht schneller als mit der original-Betriebssystem-Version des Copy-Befehls. Man wird zum Diskjockey. Aber damit diese Funktion ohne Speichermodule funktioniert, geht es nicht anders (vielleicht ein bißchen, vielleicht kommt das ja noch mal).

Man kann DW auch von Cassette starten, es versucht jedoch immer zuerst Laufwerk S2: zu suchen und darauf vorhandene Files einzulesen.

Aber zurück zur Bedienung. Dazu zunächst einmal eine Beschreibung des Erscheinungsbildes von DiskWorks nach dem Starten und nach der Eingangsmeldung.

-> FILE1	.	4711
FILE2	.	4712
FILE3	BAS!	4712
INF CPY KIL MSK S2:		X:

So kann's aussehen, wenn DW gestartet wurde. Vorausgesetzt natürlich, Laufwerk S2: ist vorhanden und enthält die angezeigten Files. Ist der PC nicht an den CE-1600P (mit Floppy) angedockt, dann ist als Ziellaufwerk nicht X: sondern COM: eingetragen.

Die oberste angezeigte Datei ist immer die 'aktuelle' Datei. Sie kann ich zur Bearbeitung markieren, starten (falls Programm) oder ansehen. Aus diesem Grund erscheint auch der Pfeil in der obersten Zeile.

File3.BAS im Beispiel wird von einem Semikolon ";" abgeschlossen. Das bedeutet, daß diese Dateien mit dem Schutzflag versehen ist und nicht versehentlich gelöscht werden kann.

Die Zahl hinter dem Dateinamen bezeichnet die Länge der jeweiligen Datei.

Sind mehr als drei Files auf dem Laufwerk abgelegt, so kann mit den Cursortasten (runter und rauf) fileweise gesprungen werden. Cursor runter in der oben angezeigten Situation würde FILE2 zum aktuellen File machen. FILE1 verschwindet und FILE3.BAS rückt nach.

Analog funktioniert das mit der Cursortaste hoch. Wenn man an den jeweiligen Rändern angekommen ist (erster oder letzter File), dann hat das Betätigen der Cursortasten keine Auswirkung. Es piept dann auch nicht (DW ist schließlich eine Benutzeroberfläche und kein Musikprogramm!).

Möchte man schnell zum Anfang oder zum Ende der Dateiliste, so nimmt man die Cursortasten links und rechts (am Zehnerblock). Rechts springt zum Ende, links zum Anfang.

Um Dateien bearbeiten zu können (umbenennen, kopieren, löschen), müssen sie markiert sein. Das wird fileweise mit der SPACE-Taste gemacht. Mit ihr wird die Markierung jeweils invertiert. Also sowohl ein- als auch ausgeschaltet.

Die Markierung aller Files invertieren, die aktuell eingelesen sind und zur Filemaske passen, kann man mit der RCL-Taste. Die Doppelpfeiltaste hebt die Markierung aller Files auf!

Hier muß ich einmal kurz auf die Menüfunktionen vorgreifen. Um nämlich das angezeigte Quell- und Ziellaufwerk schnell zu tauschen dient die Taste KBII. Mehr zu dem Thema der Laufwerke später.

Die aktuelle Datei (erste in der Anzeige) kann, ob BASIC- und ML-Programm, über die ENTER-Taste gestartet werden. DW erkennt den Dateityp selbständig und unabhängig vom Namen.

Mit OFF kann an jeder Stelle in DW der Rechner ausgeschaltet werden. MODE dient auf der obersten Menüebene (Files werden angezeigt, eine Menüzeile als unterste Zeile im Display) zum Beenden von DiskWorks. Ansonsten kann mit MODE jede Operation vor der endgültigen Ausführung (innerhalb von Sicherheitsabfragen etc.) abgebrochen werden.

Und jetzt zu den Funktionstastenmenüs. Zu der untersten Menüzeile (INF, CPY, KIL, MSK, S2:, X:) existiert ein Gegenstück auf der sogenannten zweiten Menüebene (NAM, PRO, VIE, PAR, INI, DCP). Zwischen den beiden Menüebenen kann mit einem Tastendruck (DEF) umgeschaltet werden.

Die angezeigten Funktionen werden dadurch aktiviert, daß die darunter liegende Funktionstaste (!, ", #, \$, %, &) betätigt wird.

Aber nun zu der Bedeutung der Kürzel in den Menüzeilen.

Erste Menüebene:

INF: dieser Menüpunkt gibt Auskunft über das aktuell ausgewählte Laufwerk, die vorhandenen Files, die gewählte Filemaske, das Datum und den Stand der Batterien (intern und falls angeschlossen auch extern).

Die oberste Zeile gibt den freien Speicherplatz auf der RAM-Disk S2: bzw. des Quelllaufwerks aus. Die Filemaske ist die Anzeigeschablone für anzuzeigende Files. Dabei wird das *.* in die oben gezeigte Beispielmaste umgesetzt. Über die Anzahl Files, die zu der Filemaske passen, und wieviele davon aktuell markiert (0) sind, gibt die dritte Zeile Auskunft. Die unterste Zeile zeigt Datum, Uhrzeit, interne Batteriespannung (I=) in Prozent und die externe Batteriespannung (CE-1600P, E=), falls das Gerät angeschlossen ist.

Bei angeschlossenem Netzteil kann es schon einmal vorkommen, daß eine Spannung über 100% angezeigt wird. Naturgemäß ist die Umwandlung der analogen Spannung auf dem PC-1600 recht ungenau. Um die Spannung möglichst genau messen zu können, werden mehrere Messungen durchgeführt und anschließend ein Mittelwert gebildet. Trotzdem können noch geringe Sprünge und Ungenauigkeit auftreten. Wenn einem also ein Messergebnis 'spanisch' vorkommt, einfach noch einmal die INF-Funktion aufrufen...

CPY: hiermit werden Files kopiert (CoPY). Dazu müssen sie markiert werden. Die Dateien werden dann vom Quelllaufwerk auf das Ziellaufwerk übertragen. Datum und Uhrzeit der Dateien bleiben dabei erhalten.

Zum Kopieren fragt DW, ob das Ziellaufwerk bereit ist. Mit 'J' beginnt der Kopiervorgang. Mit 'N' wird er nicht gestartet. MODE bricht ab.

Mit dem CoPY-Befehl kann selbstverständlich auch über die serielle Schnittstelle kopiert werden. Für Leute, die einen PC (IBM-Clone, Atari der MAC oder sonstwas in der Art) nebenbei besitzen können somit prima Datensicherungen auf den PC betreiben (siehe auch Angebot in der Einleitung).

Auf den PC kopieren ist sehr einfach. Es werden alle zu kopierenden Files markiert und anschließend die CoPY-Funktion gestartet. DW fragt vor jedem File, ob die Schnittstelle bereit zur Aufnahme der Daten ist. MODE oder 'N' bricht hier den Kopiervorgang ab. 'J' startet die Übertragung.

Der Rückweg vom PC ist etwas mühsamer. Da DW nicht wissen kann, wie die zu übertragende Datei heißt, wird vor jeder Übertragung auf den PC-1600 nach einem Dateinamen gefragt. Anschließend wird so verfahren wie bei der Übertragung auf den PC.

Jetzt noch was zum Thema Backup von größeren RAM-

Disks. Wer eine 128 oder gar eine 256 KByte RAM-Disk sein Eigen nennt, der möchte sicherlich ab und an ein Backup durchführen. Das geht selbstverständlich nur auf mehrere Disketten. Mit DW ist das Kopieren jedoch sehr einfach. Es werden einfach alle Files der RAM-Disk markiert (RCL nach Start) und die Kopierfunktion gestartet. Ist die Diskette voll, gibt es eine Fehlermeldung und DW hat alle schon erfolgreich kopierten Files übertragen und die Markierung entfernt.

Nun wird einfach die nächste Diskette für das Backup eingelegt und der Kopiervorgang wieder gestartet. Dies macht man, bis DW alle Files kopiert hat.

DiskWorks überschreibt übrigens Files gleichen Namens immer auf der Zieldiskette. Da ist etwas Vorsicht geboten!

Der Kopierfortschritt wird in der untersten Displayzeile durch "." (Leseoperation) und "o" (Schreiboperation) angezeigt.

KIL: Mit KIL kann gelöscht werden (KILI). Sind Files markiert, so werden diese gelöscht. Dazu fragt DiskWorks den Benutzer noch, ob das Löschen jedes Files einzeln bestätigt werden soll. 'J' sorgt dafür, daß für jede markierte Datei eine Sicherheitsabfrage erfolgt. 'N' löscht sofort. MODE bricht ab.

Ist keine Datei markiert, so kann auch mit Wildcards gelöscht werden. Zum Beispiel "*" löscht alle Files auf dem Quelllaufwerk. Und das unabhängig von der eingestellten Filemaske!

Auch hier gilt: aufgepaßt! Gelöschtes ist unwiederbringlich weg!

MSK: legt die Filemaske fest (MaSKe). Um z. B. alle Basic-Files anzuzeigen, kann nach Betätigen von '\$' bzw. F4 '*.BAS' eingegeben werden. Anschließend werden nur noch Files mit der Endung *.BAS im Display angezeigt. Um eine bestehende Maske zu löschen, muß lediglich die ENTER-Taste betätigt werden. MODE bricht ab und verändert nichts.

S2: zeigt das aktuell gewählte Quelllaufwerk an. Durch '%' bzw. F5 kann das Laufwerk gewechselt werden. Die Auswahl geschieht wieder über Funktionstasten F2 bis F6. F2=COM:, F3=X:, F4=Y:, F5=S1:, F6=S2:. MODE bricht ab und ändert nichts. Mit COM ist das durch SETDEV gewählte serielle Port des PC-1600 gemeint. Die in BASIC-eingestellten Übertragungsparameter werden ebenso übernommen.

X: zeigt das aktuell gewählte Ziellaufwerk an. Durch '&' bzw. F6 kann es gewechselt werden. Siehe S2: bzw. Quelllaufwerk.

Zweite Menüebene:

NAM: dient zum Umbenennen von Dateien. Es werden entweder einzeln alle markierten Files angezeigt und nach dem neuen Namen gefragt oder (wenn kein File markiert ist) ein reNAME mit Wildcards erfragt. NAME "*.BAK" AS "*.OLD" benennt alle Files mit der Endung 'BAK' in Files mit der Endung 'OLD'. MODE bricht auf jeder Ebene

des Befehls ab.

PRO: Schützen von Dateien. Ist eine Datei mit einem Schreibschutz versehen (bei DW durch ein '!' hinter dem Dateinamen angezeigt), dann kann sie nicht gelöscht oder überschrieben werden (von den Betriebssystem- oder DW-Funktionen).

Mit PRO kann nun der Schutz der aktuellen Datei im Display (oberste Zeile) ein- bzw. ausgeschaltet werden.

VIE: Ansehen von Files im Hex-Modus (inkl. ASCII). Dies kann eine sehr nützliche Funktion sein. Man kann 'mal eben kurz' in beliebige Files 'hineinsehen'. Sie werden im Hex-Modus angezeigt. Hinter den Hexziffern erscheint im Display das entsprechende ASCII-Zeichen.

Ist das Display voll, hält DW an. Mit einer beliebigen Taste wird die nächste Zeile angezeigt (alle anderen werden eine Zeile hoch geschoben).

MODE dient auch hier wieder zum Beenden der Fileanzeige.

COM: stellt die Kommunikationsparameter für die serielle Schnittstelle ein. Die Standardeinstellung nach dem Start von DiskWorks entspricht den BASIC-Befehlen:

SETDEV "COM1:" und
SETCOM"COM1:" ,9600,8,N,1,X,N

in Folge.

Nun kann die Setcom-Einstellung mit den Funktionstastentennis in dieser Funktion beliebig geändert werden.

INI: mit dieser Funktion wird eine Diskette im Laufwerk X: formatiert (INIt-Befehl von BASIC). DW fragt zunächst, ob eine Diskette in X eingelegt wurde. Wenn mit 'J' geantwortet wird, dann wird die in X: einliegende Diskette gnadenlos formatiert (initialisiert). Also Vorsicht, alle Daten sind damit von der Diskette gelöscht!

DCP: kopiert ganze Disketten von Laufwerk X: nach Laufwerk X: (DiskCoPy). Dazu beachte die eingangs gemachten Anmerkungen über die benötigten Hardware-Konfigurationen.

DW fragt nach dem Start der Funktion nach der SOURCE-Disk (Quelldiskette), liest einen Teil der Diskette in den Speicher und fragt anschließend nach der TARGET-Disk (Zieldiskette). Und dies so oft, bis die gesamte Diskette kopiert wurde (5 mal).

Die Konfigurationsdaten DW.CFG und DWMac

Wie eingangs erwähnt, ist zum Betrieb von DW eine Datei namens DW.CFG notwendig. Diese Datei wird von einem beliebigen Texteditor (z. B. BASIC mit LOAD*, SAVE* oder Text+) erstellt. Sie enthält vom Benutzer selbst zu konfigurierende Kommandos bzw. Kommandofolgen (Makros, daher auch der Name DWMac), die dann von DW aufgerufen werden können.

Die Datei muß als erstes und letztes Zeichen ein Hütchen ("^" = Shift + SPACE) enthalten. Anderenfalls kann DW abstürzen.

Und in der mit diesem Handbuch vorliegenden Version 2.0 von DiskWorks gibt es noch eine Besonderheit bei diesen Makros. Zunächst einmal eine schlechte Nachricht: zur Ausführung eines Makros wird nicht mehr automatisch DW verlassen. Und jetzt viele gute: man kann jetzt Makros sowohl in gewohnter Form (mit einer kleinen Änderung) als auch innerhalb von DiskWorks anwenden. Z. B. Markerien aller *.BAK und automatisches Löschen. Derer Anwendungen gibt es viele. Aber da ist dem Anwender keine Grenze gesetzt und seine Phantasie gefordert.

Jetzt aber nähere Informationen zu der Erweiterung der Makrosprache 'DWMac'.

Ein wenig Übertreibung ist immer ganz nett. Also nehmt das mit der Makrosprache nicht so ernst. Es handelt sich nämlich nicht um eine ausgezeichnete Programmiersprache, sondern vielmehr um das Aufschreiben von Tastenfolgen, die DW dann wieder abspielen kann.

Das nennt sich dann ein 'Makro starten'. Makros können sowohl innerhalb als auch außerhalb von DW ausgeführt werden. Es wird ja eigentlich 'nur' der KEYBUFF\$ des Betriebssystems verwendet.

Wichtig ist nur, daß man nicht versichtlich ein Makro, daß für das 'Außerhalb' bestimmt war, innerhalb von Diskworks ausführt. Das führt dann nämlich mit Sicherheit zum Absturz des Rechners.

Ein Makro hat im Prinzip einen recht einfachen Aufbau. Es fängt mit der Taste an, die nach dem Betätigen dazu führen soll, daß das Makro gestartet wird. Dann folgt ein Space und dann fängt die Befehlsfolge an, die ausgeführt werden soll.

Also:

<Taste><SPACE><Befehlsfolge>

Die <Taste> muß eine Taste sein, die man über die Tastatur erreichen kann, also die Menge [A..Z,a..z,0..9,!,",#,\$,%,&,?,:;,{}[,],(,),_,@,`',,;]. Jeder sollte damit mal experimentieren. Aber in etwa müßten das alle Zeichen sein.

<SPACE> muß auch nicht weiter erklärt werden, ist aber in jedem Fall notwendig.

Und jetzt zur <Befehlsfolge>. Da kann zunächst auch mal jedes Zeichen drinstehen, was über die Tastatur erreichbar ist. Z.B. die Folge <LOAD"S2:DW">. Tja, und da hätten wird dann auch gleich ein kleines Problem aufgedeckt (denkst Du), denn wie mache ich dem Rechner klar, daß ich hinter dem Load... auch noch ein ENTER gedrückt haben möchte? Kein Problem. Es gibt sogar zwei Lösungen dieses Problems. Die erste ist das Zeichen "~" (die Tilde). Anstelle der Tilde wird vom Makro-Interpreter (hört sich auch gut an, nicht war?) ein ENTER bei der Ausführung der Makros eingesetzt.

Jetzt soll es aber schon mal vorgekommen sein, daß man nicht nur ein ENTER, sondern z.B. den Code der OFF-Taste in einem Makro haben möchte. Oder gar den Code der CTRL-Taste. Oder der SML-Taste...

Tja, auch da gibt's eine Lösung. Tastencodes jeder Art kann man in DW.CFG als folgende Zeichenfolge schreiben:

\<ASCII-CODE\

Also den Code in zwei nach hinten gekippte Schrägstriche eingefast, z.B. \15\. Das bedeutet die OFF-Taste, da der ASCII-Code der OFF-Taste 15 ist.

Damit ihr nicht alle Codes erst noch herausfinden müßt, gibt's eine Tabelle mit den besonderen Codes. Ihr könnt auf diese Weise auch die normalen ASCII-Zeichen in den Makros abbilden. Aber es ist schließlich einfacher "LOAD" zu schreiben, als \76\79\65\68\.

Es gibt noch eine kleine aber nichtsdestotrotz feine Besonderheit. Mit der Zeichenfolge

\@

kann an der Stelle im Makro der Dateiname der obersten Displayzeile eingefügt werden. Also praktisch wie eine Variable, die je nach Situation einen andern Wert hat.

Taste	Code
F1	17
F2	18
F3	19
F4	20
F5	21
F6	22
DEF	27
KBII	4
CTRL	3
SHIFT	1
SML	2
DPfeil	9
RCL	25
Down	10
Up	11
BS	5
OFF	15
CL	24
MODE	31
Left	8
Right	12
\	92 oder \\
@	64
~	126

Und jetzt noch mal auf das 'Innerhalb' und 'Außerhalb'. In der Version 1.2 von DW wurde für ein Makro automatisch DW verlassen und somit automatisch auf den Basic-Interpreter 'losgelassen'. Das geschieht jetzt einfach nicht mehr, sondern man muß das Verlassen von DW einfach mit der MODE-Taste erzeugen. Aber wie die MODE Taste erzeugen? Na klar, gerade erst gelernt! Einfach \31\ als erstes Zeichen der Befehlsfolge und schon geht's in Richtung Basic-Interpreter.

Damit wird auch klar, warum man darauf achten muß, kein 'außerhalb' Makro 'innerhalb' abzuspielen. Angenommen wird hätten folgendes vor: Laden des Programms Text+ aus DW heraus per Makro. Text+ liegt auf S2: und heißt TXT.

Also würde die Befehlsfolge im Basic-Interpreter lauten "BLOAD"S2:TXT<ENTER>". In DW.CFG müßte aber folgendes vermerkt sein (wir nehmen mal an, daß wir die Taste "T" belegen wollen):

T \31\BLOAD"S2:TXT~

Also vorangestellt der Code für die Mode-Taste, dann das <SPACE> und abschließend der Code für die ENTER-Taste.

Sollte das \31\ vergessen werden, so führt DW das Makro

'Innerhalb' aus und bis zum ersten "T" von TXT passiert nichts Schlimmes. Aber bei dem "T" 'denkt' DW, es soll das Makro der Taste "T" ausführen... Das führt also dazu, daß immer wieder das Makro ausgeführt wird. Unter Umständen kann man das dann nicht mehr abbrechen. Da hilft dann nur noch der Reset-Schalter. Sollte der Fehler doch mal passieren, dann versucht in jedem Fall erst einmal die BRK-Taste. Vielleicht kann doch damit das Endlosmakro unterbrochen werden.

Sofort nach Drücken der definierten Taste wird der jeweilige Befehl (oder die Befehlsfolge) ausgeführt. Es erfolgen keine Sicherheitsabfragen, wie beim Starten von Programmen über Enter.

Die Datei DW.CFG darf allerdings niemals länger als 1024 Byte sein! Anderenfalls stürzt DW schon beim Start ab! Also paßt auf. Das tut DW an der Stelle nämlich nicht.

Im folgenden noch eine kurze Beschreibung meiner aktuellen DW.CFG-Datei. Vorab noch einmal: das erste und letzte Zeichen müssen ein Hütchen ^ sein!! Das erste Hütchen muß mit in der ersten Zeile stehen, daß letzte Hütchen muß unbedingt in einer eigenen Zeile stehen!

```
^C \31\BLOAD"S2:TXT~LS2:DW.CFG~E
T \31\BLOAD"S2:TXT~
X \31\BLOAD"S2:DEB~
F \31\LOAD"S2:FINANZ",R~
D \31\BLOAD"S2:DB~
! \31\BLOAD"S2:FKEY,#0,&C008~BLOAD"S2:DW~
U \31\BLOAD"S2:CLOCK~BLOAD"S2:DW~
K \31\BLOAD"S2:TK~
O \15\
M \9\25\
B M\18\
A \31\LOAD"S2:DWASM~
E \31\BLOAD"S2:TXT~L\@~E
Q \31\LOAD"S2:ST.BAS",R~
^
```

Die erste Zeile ist also schon mal ganz schön kompliziert. Das erste Zeichen ist das Hütchen, hat also mit dem eigentlichen Makro nichts zu tun. Dann folgt das 'C'. Das Makro wird also auf die Taste C gelegt. Das folgende Leerzeichen muß die Tastendefinition von dem eigentlichen Makro trennen. Nun fängt erst das Makro an. \31\ ist der Tastencode für die MODE-Taste, veranlaßt also den Makro-Interpreter, DW zu verlassen (man kann auch sagen: es wird für den Benutzer die Mode-Taste gedrückt). BLOAD "S2:TXT~ lädt das Programm Text+ (heißt bei mir TXT und liegt auf S2:). Die Tilde ~ ersetzt die Enter-Taste. Dann ist aber noch nicht das Ende erreicht. Jetzt wird für den Benutzer die Taste L gedrückt und der Dateiname S2:DW.CFG mit Enter ~ eingegeben. Nachdem jetzt die Datei geladen wurde kommt das letzte, was das Makro noch macht. Es startet mit E den Edit-Modus von Text+. Zusammengefaßt lädt das Makro also den Texteditor und die Datei DW.CFG zur Bearbeitung. Und das ganze aus DW heraus mit einem Tastendruck. Das erste Beispiel zeigt also schon eine ganze Menge der Makro-Sprache DWMac.

Jetzt weiter mit dem Makro, das der Taste F zugeordnet ist. Das ist zwar ein einfaches Makro, zeigt aber, daß man

nicht nur Maschinenprogramme starten kann, sondern auch Basic-Programme laden (LOAD"S2:FINANZ") und diese dann mit dem Commando ,R~ automatisch starten kann.

Nun betrachten wir das Makro, das mit dem Ausrufezeichen ! beginnt (der Taste ! zugeordnet ist). Das ist auch eine feine kleine Sache, auf die der aufmerksame Leser bestimmt schon gekommen ist. Erst lädt das Makro eine Funktionstastendefinition (BLOAD"S2:FKEY"), an die Stelle #0,&C008 und startet anschließend wieder das Programm DW, also DiskWorks.

Das gleiche Prinzip wird in dem Makro U verwendet. Nur wird dort die Hintergrunduhr geladen (aktivierbar mit Shift-Tasten, bleibt dann im Hintergrund und wartet auf Aktivierung). Anschließend wird ebenfalls wieder DiskWorks gestartet.

Ein besonders kurzes Makro ist der Taste O zugeordnet. Die einzige Aktion ist das Betätigen der Taste mit dem Tastencode 15. Das ist nichts anderes als die OFF-Taste. Also schaltet O den Rechner aus (Off).

Nach dem besonders kurzen Makro O nun mal ein ganz besonders vielseitiges Makro, was auf den ersten Blick gar nicht besonders aussieht. Es fängt mit E an, verläßt dann mit \31\ DW, startet S2:TXT~ und betätigt dort die Load-Funktion (L) und..? Ja, was ist denn das. Da steht ja das Klammeräffchen @ nach dem Backslash \! Und was soll das? Ganz einfach. E steht für Edit (ist ja klar, es wird ja der Editor Text+ geladen) und soll eine Datei edieren. Aber welche Datei? Auch ganz einfach. Nämlich die aktuelle Datei, also die Datei, die im Display an oberster Stelle steht. Ist doch klasse, oder? Ein universelles Makro zum edieren beliebiger Dateien. Eins für alle! Hier liegt jetzt jedoch ein wenig Verantwortung beim Benutzer, denn es macht unter Umständen nicht viel Sinn, einen Debugger mit der Textverarbeitung zu traktieren. DW's Makrointerpreter prüft an der Stelle nämlich kein bißchen! Wie soll er auch?

Das letzte Makro ist ganz speziell für unseren gemeinsamen Freund Gernot geschrieben. Der nervt mich nämlich ständig damit, daß ich mit dem Verlassen meiner schönen Programme seine Benutzertragfläche 'Status Klo Dry' aufrufen soll (hihi und sorry Gerni, in der Beschreibung zu Deiner Fläche hast Du jetzt einen Scherz über meine Software frei. Aber maximal einen, und gefälligst nicht so fies!). Das kann er jetzt gefälligst selbst machen. DW bietet alle Mittelchen dazu an!

Bevor jetzt Fragen auftauchen sei an dieser Stelle noch einmal darauf hingewiesen, daß die Makros nur dann ausgeführt bzw. gestartet werden können, wenn DiskWorks aktiv ist! Sobald DW verlassen wurde (außer über Makrobefehl \31), kann kein Makro mehr ausgeführt werden. Das kann auch nicht wünschenswert sein, da ansonsten niemand mehr mit dem PC-1600 arbeiten könnte, da beim Betätigen von zugewiesenen Tasten ständig ein Makro ausgeführt würde. Die Verfahrensweise nach dem Makrobefehl ist inzwischen klar. Wenn das Makro abgearbeitet wurde, das mit \31\ begann, ist's vorbei

mit dem Makro. Es sei denn, der letzte Makro-Befehl hat wieder DW gestartet und man möchte ein weiteres Makro ausführen. Damit kann man dann auch Makros miteinander verknüpfen und ineinander verschachteln. Aber Vorsicht!

Und noch ein Hinweis. Achtet darauf, daß Ihr Euch keine Endlosschleifen in die Makros einbaut. Es hilft dann unter Umständen nur noch ein Reset. Eine kleine Endlosschleife sei hier gezeigt:

```
Q \31\BLOAD"S2:DW~Q
```

Das Q ist die zugeordnete Makrotaste und nach dem Starten von DW (über das Makro) wird die Taste Q simuliert, also wieder DW verlassen und DW neu gestartet und dann wird wieder die Taste Q simuliert, mit der DW verlassen und wieder neu gestartet wird, worauf wieder die Taste Q simuliert wird... Ihr seht also, eine einfache Endlosschleife, aber doch sehr effektiv und unter Umständen verhängnisvoll. Achtet also darauf.

Sollte euch der Fehler passieren, daß Ihr eine Makro-Taste doppelt oder gar dreifach vergebt, so hat das keinen negativen Seiteneffekt, außer daß die zweite und dritte Definition einfach überhaupt keine Beachtung finden wird. Nachdem der Makrointerpreter das erste Makro mit der zugeordneten Taste gefunden hat, wird es auch schon ausgeführt, unabhängig davon, ob noch einmal ein Makro mit der gleichen Taste vorliegt.

Fehlermeldungen:

Die Fehlermeldungen entsprechen im allgemeinen den Fehlermeldungen des Basic-Interpreters (bei den File-Funktionen). Was genau stimmt, sind die Nummern der Fehlermeldungen. Somit können Erläuterungen zu den Fehlern in dem Originalhandbuch des Rechners nachgelesen werden.

Die einzigen beiden 'Nichtübereinstimmungen' sind die Fehler 98 und 99, 'Ziel = Quelle' und 'sonstiger Fehler!'. Letzter tritt dann ein, wenn die Fehlerursache nicht genau zu analysieren ist. Da ist dann die gute alte graue Zellenansammlung gefragt, die sich im oberen Teil der Halsausbeulung befinden sollte. Im folgenden eine Auflistung aller angezeigten Fehlermeldungen.

- 98 Ziel = Quelle!
- 99 sonstiger Fehler!"
- 151 Datei gibt's schon
- 152 Datei nicht gefunden!
- 154 Datei bereits offen!
- 155 Gerät nicht vorhanden
- 157 Falscher Dateiname!
- 159 Disk schreibgeschützt!
- 160 Keine Diskette!
- 161 Disk nicht formatiert!
- 162 Schreib- Lesefehler!
- 163 Falsche Disk(Wechsel)!
- 164 Diskette voll!
- 167 Fataler Fehler!
- 168 Akkus leer!

Allgemeines

Sollten Euch Fehler auffallen bitte ich um sofortige Meldung an mich. Nobody ist perfect (I'm Nobody).

KiKiSoft ist:

Christian (KiKi) Becker

Hörder Str. 282

5810 (ab 1.7.93:58454) Witten-Stockum

und ab 1. 9. 93 wohne ich in der Goebenstr. 10 in 44135 Dortmund.

Wenn Ihr - genauso wie ich - meint, daß DW ein Spitzenprogramm ist und ihr es wirklich gebrauchen könnt, dann bin ich keinesfalls abgeneigt, kleinen Geldspenden (20 bis 2000 Marx reichen in etwa) freudigen Herzens entgegenzusehen.

Zur Information: DiskWorks stammt aus einer Idee, die ich im Frühjahr 1992 hatte. Im Frühsommer habe ich dann mit der Realisierung begonnen (nach Basic-Studie). Auf dem 1992er Clubtreffen im Juli lag eine 'Dummy'-Version vor, die in etwa die Benutzeroberfläche andeutete. Damals waren Gernot und Harald Feuer und Flamme und ich fühlte mich animiert, das Programm weiterzuentwickeln.

Seit dem Spätsommer 1992 ist DW so weit, daß ich ständig damit arbeite (der eine oder andere hatte auch das Vergnügen, mit fehlergespickten Vorabversionen arbeiten zu müssen, äh dürfen). Aber die Entwicklung der Version 1.0 ist erst mit dem Monat Januar 1993 abgeschlossen worden. Nun, im April gab's die Version 1.1 (an allen Banken und Sparkassen) und heute (Ende Mai bis Anfang Juni) die Version 2.0. Und das Problem ist, daß ich schon wieder ein paar Ideen habe, was man noch in DiskWorks gebrauchen kann.

Insgesamt steckt also zur Zeit ein gutes Jahr Entwicklungszeit in dem Programm (mit vielen Denkpausen und Anregungen durch Gernot und Harald, Danke dafür).

Tja, danke für die Aufmerksamkeit und viel Spaß mit DiskWorks 2.0!