

DiskWorks V2.0

Inhaltsverzeichnis

Einleitung.....	2
Überblick.....	3
Funktionsmenüs.....	5
Erste Menuebene:.....	5
INF.....	5
CPY.....	5
KIL.....	6
MSK.....	6
S2.....	6
COM.....	7
Zweite Menuebene:.....	7
NAM.....	7
PRO.....	7
HEX.....	7
COM.....	7
INL.....	8
DCP.....	8
Konfigurationsdatei (DW.CFG, DWMac).....	8
Überblick.....	8
„Macro-Sprache“ DWMac.....	8
Aufbau der Datei DW.CFG.....	9
Besonderheiten.....	9
Achtung!.....	10
Mein Beispiel.....	11
Erste Zeile.....	11
Ausrufezeichen!.....	12
U wie Umweg.....	12
Off.....	12
„Parameterübergabe“.....	12
Schlussbemerkung zu den DWMac.....	13
Fehlermeldungen:.....	13
Abschlussbemerkungen zu DW und dem Universum.....	14

Einleitung

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!) getrost vergessen werden. Selbst das Wissen über ganze Befehlsfolgen kann - nachdem es einmal definiert wurde - in den Papierkorb wandern.

Mit DiskWorks 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 „großen“ PC (Windows, Linux, Mac) sein Eigen nennt, der kann von mir noch ein Programm zur Sicherung von Daten und Programmen auf den PC (über die serielle Schnittstelle) nutzen. Hier ist einiges am Markt erhältlich – auch kostenlos. Kabel sind allerdings selbst zu löten oder zu kaufen. Hier ist einiges möglich, auch mit Arduino oder ESP32 Einheiten.

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 2.0 beginnt bei der Adresse &DOOO 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 &CCOO 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. 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. Sonst 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 Kassette starten, es versucht jedoch immer zuerst Laufwerk S2: zu suchen und darauf vorhandene Files einzulesen.

Überblick

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 es aussehen, wenn DW gestartet wurde. Vorausgesetzt natürlich. Laufwerk S2: ist vorhanden und enthält die angezeigten Files. Ist der PC nicht an den CE1600P (mit Floppy) angedockt, dann ist als Ziellaufwerk nicht X: sondern COM: eingetragen.

Die zuoberst angezeigte Datei ist immer die 'aktuelle' Datei, auch durch den Pfeil (->) angedeutet. Sie kann ich zur Bearbeitung markieren, starten (falls Programm) oder ansehen.

File3.BAS im Beispiel wird von einem Ausrufezeichen "!" abgeschlossen. Das bedeutet, dass diese Datei mit dem Schutzflag versehen ist und nicht versehentlich gelöscht werden kann .

Die Zahl hinter dem Dateinamen bezeichnet die Größe der jeweiligen Datei in Byte.

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 Zehner-Block). 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 File-Maske 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 oder 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.

Funktionsmenüs

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, HEX, COM, INI, DCP). DEF schaltet zwischen diesen beiden Zeilen hin und her.

Die angezeigten Funktionen werden dadurch aktiviert, dass die darunter liegende Funktionstaste (!, ", #, S, %, &) 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 File-Maske, 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 File-Maske ist die Anzeigeschablone für anzuzeigende Files. Dabei wird das '*.*' in die oben gezeigte Beispielmaste umgesetzt. Über die Anzahl Files, die zu der File-Maske passen, und wie viele davon aktuell markiert (0) sind, gibt die dritte Zeile Auskunft. Die unterste Zeile zeigt Da«n, 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 (COM1:) kopiert werden. Mit einem PC (Linux, Windows, Mac) können und sollten Datensicherungen auf den PC angefertigt werden. No backup, no mercy!

Zum Kopieren auf COM1: 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 1~ 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 unbedingt Vorsicht geboten!

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

KIL

Mit KIL kann gelöscht werden (KIL1). 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 File-Maske!

Auch hier gilt: aufgepasst! Gelöschtes ist unwiederbringlich weg!

MSK

Legt die File-Maske fest (MaSKe). Um z. B. alle Basic-Files anzuzeigen, kann nach Betätigen von 'S' 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

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.

COM

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

Zweite Menuebene:

NAM

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.

HEX

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 HexModus 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 Funktionstastenmenüs in dieser Funktion beliebig geändert werden.

INI

mit dieser Funktion wird eine Diskette im Laufwerk X: formatiert (INI-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).

Konfigurationsdatei (DW.CFG, DWMac)

Überblick

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. Markieren aller *.BAK und automatisches Löschen. Derer Anwendungen gibt es viele. Aber da ist dem Anwender keine Grenze gesetzt und seine Phantasie gefordert.

„Macro-Sprache“ DWMac

Jetzt aber nähere Informationen zu der Erweiterung der Sprache '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 versehentlich ein Makro, das für das 'Außerhalb' bestimmt war, innerhalb von DiskWorks ausführt. Das führt dann nämlich mit Sicherheit zum Absturz des Rechners.

Aufbau der Datei DW.CFG

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.

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 wahr?) ein ENTER bei der Ausführung der Makros eingesetzt.

Besonderheiten

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 eingefasst, z.B. \15\ Das bedeutet die OFF-Taste, da der ASCII-Code der OFF-Taste 15 ist.

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.

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\

Taste	Code
F1	17
F2	18
F3	19
F4	20
F5	21
F6	22
DEF	27
KBII	4
CTRL	3
SHIFT	1
SML	2
DoppelPfeil	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 nicht mehr automatisch, 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.

Achtung!

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 Gur 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.

Mein Beispiel

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, das 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\BLOADnS2: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~
^
```

Erste Zeile

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.

Ausrufezeichen!

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.

U wie Umweg

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. Also zurück zur „universellen Benutzeroberfläche“.

Off

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).

„Parameterübergabe“

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 Klammeröffchen @ 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?

Schlussbemerkung zu den DWMac

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 PC1600 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.

Es sei denn, der letzte Makrobefehl 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!
```

Abschlussbemerkungen zu DW und dem Universum

Sollten Euch Fehler auffallen bitte ich um sofortige Meldung an mich. Nobody ist perfect (I'm Nobody). Unter crido@gmx.de bin ich unter anderem erreichbar

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 Fehler gespickten 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 und 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!

XianBild

18. Juli 1993 (im Original, heute ist der 18.07.2024 und ich habe aktuell den PC-1600 reaktiviert und diese Anleitung überarbeitet.)