

```
##      ##
###    ###
####  ####
## #### ##
## ## ##
##    ##
##    ##
##    ##
```

```
#####
#####
##
##
##
##
##
##
```

```
##      ##
##      ##
##      ##
####
####
##      ##
##      ##
##      ##
```

I n f o
1

M T X U s e r - C l u b D e u t s c h l a n d
=====

1. Zweck: Austausch von Tips & Tricks u.s.w.
2. Programme (nur Selbstgeschriebenes): Tausch von kurzen und einfachen Routinen, Besprechung von guten Programmen damit der Autor diese dann an Clubmitglieder verkaufen kann. Programme einfach an uns schicken, und wir liefern Verbesserungshinweise, Besprechung, ...
3. Mitglied kann Jeder werden! Keine Aufnahme oder Beitragsg Gebühr!
4. Verpflichtungen keine!

Bitte: Einsendung von Tips & Tricks, Fragen, Antworten, kurzen Routinen, und was noch so zusammenkommt und andere interessieren könnte.

5. Club-Info wollen wir ca. monatlich verschicken. Das hängt von allen ab, da wir ja nicht rund um die Uhr am Computer sitzen können. Da brauchen wir die Hilfe aller Mitglieder!
6. Kosten: Wir berechnen ausschließlich Selbstkosten (Porto, Verpackung,..) Verständlicherweise verschicken wir nichts, wenn kein Geld da ist (s.u.)

Da wir es doof finden, das allererste Info in dem steht, wer wir sind nur gegen Freiumschlag zu verschicken, ziehen wir dafür bei Jedem dafür DM -.70 vom Konto ab. Einspruch ist selbverständlich jederzeit möglich!

7. Geld/Konto: Für Jedes Mitglied führt Herbert Herberg ein Konto, von dem die entstehenden Kosten jeweils abgehen. Der Kontostand wird regelmäßig mitgeteilt, und kann selbverständlich jederzeit erfragt werden!

Einzahlungen bitte möglichst auf's Club-Konto!(Absender nicht vergessen)
Postgiroamt Hamburg, BLZ 200 100 20,
Herbert Herberg Sonderkonto C, Nr. 3480 00-200

8. Kontaktadressen: (alle derselbe Club!!)
- | | | |
|-----------------|-------------------|-----------------|
| Herbert Herberg | Thomas Pflaum | Martin Müller |
| Sonnenau 2 | Leipziger Platz 1 | Kroosblöcke 72 |
| 2000 Hamburg 76 | 8500 Nürnberg 70 | 2000 Hamburg 74 |
| (040) 200 87 04 | (0911) 51 35 21 | (040) 655 08 19 |

Endlich ist es soweit!! Das erste größere Info geht raus. Wir bitten um Entschuldigung bei allen, die schon lange warten und warte und warten ... Vielen Dank für die Zahlreichen Zuschriften und Überweisungen. Bitte weiter so!!

Nun erst mal zum Grund der Verspätung: Da ich bei Vobis die FDX-Station erstanden habe, wollte ich auch schließlich damit diesen Text mit NewWord verfassen, ... aber wie das Schicksal so spielt habe ich einen MTX 512 aus einer schlechten Serie: Er will nicht mit der Floppystation zusammenarbeiten! Nach Aussagen von Firma Profisoft, von denen mein MTX stammt, und wo er schon seit 5 Wochen zur Reparatur ist hörte ich erst, das geht schnell, dann irreparabel und 'Sie bekommen einen neuen', aber wegen Lieferengpass höre ich heute, daß meiner doch repariert werden soll. Fehler sind einfach Timingprobleme! Derzeit benutze ich den MTX 512 von Martin Müller, damit ich meine Floppy mal benutzen kann. Übrigens warteten Thomas Pflaum und ich schon seit dem Frühjahr darauf, daß Profisoft uns mal eine Floppy liefert, und wir haben uns beide kürzlich eine von Vobis gekauft.
Herbert Herberg

↳ Nun zum MTX:

USERERROR, USERINT, USRRST, USERIO, USER: (Herbert Herberg)

Wenn ein Fehler auftritt, dann ruft der MTX die Fehlermeldungsroutine auf. Damit man seine eigenen Erweiterungen einbauen kann, steht die Adresse der Fehlerroutine in USERERROR als JMP-Befehl. Hier kann man also recht einfach erst eine weitere Überprüfung selbst machen, um so das BASIC zu verändern.

Wenn im Interruptflag INTFF (#FD5E) eines der Bits 4-6 auf 1 gesetzt werden, dann springt der MTX an die Adresse USRINT, wo ein JMP-Befehl stehen muß. Dazu auf der nächsten Seite eine Beispielroutine.

Beim Nonmaskable Interrupt springt der MTX an die Adresse USRRST (wo normalerweise ein JP-Befehl zum PANEL steht).

Bei der Tastaturabfrage springt der MTX an USERIO, wo ein JP-Befehl mit der Adresse des implementierten Tastatortreibers steht.

USER ist ein BASIC-Befehl, der an die Adresse USER springt, wo ein JMP-Befehl, oder ein RET stehen muß. Dies ist also ein frei definierbarer BASIC-Befehl! Allerdings nicht beim FDX-BASIC (ersetzt durch DISC!!).

↳ PRINT USING (C) Gabriel Schmidt

```
1000 LET X$=STR$(INT(ABS(X)*10^N+.5)*SGN(X))
1010 IF N>0 THEN LET X$LEFT$(X$,LEN(X$)-N)+". "+RIGHT$(X$,N) ELSE LET N=-1
1020 FOR Z=1 TO 12+N-LEN(X$):PRINT " ";:NEXT
1030 PRINT X$
1040 RETURN
```

Parameter: X=Zahl die Ausgegeben werden soll, N=Nachkommastellen

NewWord: (Herbert Herberg)

Man kann mit dem Installationsprogramm zu NewWord unter anderem vier verschiedene Drucker-Customs programmieren (in NWINSTAL: H Special Patches, Schirm 3). Weiß irgendjemand wie?? Wie kann ich die korrekten Codes für Standard/Alternate Pitch eingeben (damit der Drucker bescheid weiß)?

Es soll die folgenden NewWord-Manuals geben: **NewWord Encyclopedia** und **NewWord Installation Manual**. Wer hat/kennt diese? Ich hätte sie sehr gerne mal in Händen! (ggf. leihweise)

Routine, die Jegliches Programm ohne Disable Interrupt anhält, und weiterlaufen läßt (USERINT) (C) Herbert Herberg

Hier der Assemblercode: (Sollte als 1. Zeile ins BASIC-Programm)

```

LD    A,#C3          ;JP-Befehl
LD    (#FA98),A      ;USERINT
LD    HL,UINT        ;HL=Adresse der Routine
LD    (#FA99),HL     ;Damit: USERINT= JP UINT
LD    HL,#FD5E       ;INTFF
SET   5,(HL)        ;Setzt User-Interrupt
RET                                ;Rücksprung ins BASIC
UINT: CALL #79       ;A-Register = INKEY$
CP    134            ;Taste F7 ?
JR    Z,OFF         ;Ja, dann Interrupt ausschalten
CP    131            ;Taste F4 ?
RET   NZ            ;Nein, also nichts relevantes, zurück
WAIT: CALL #79      ;A-Register = INKEY$
CP    135            ;Taste F8 ?
JR    NZ,WAIT       ;Nein, dann warten!
RET                                ;Ja, dann zurück
OFF:  LD    HL,#FD5E ;INTFF
RES   5,(HL)        ;User-Interrupt ausschalten
RET                                ;zurück

```

Funktioniert wie folgt: Im INTFF (Interrupt-Flag) wird das Bit 5 gesetzt, um den User-Interrupt zu aktivieren, d.h. regelmäßig wird vom MTX ein Aufruf CALL USERINT ausgeführt. An diese Adresse wird ein JP UINT geschrieben, so daß dann stets UINT angesprungen wird. Ist die Taste F7 gedrückt, dann wird der User-Interrupt ausgeschaltet. Ist F4 gedrückt, dann hält Jegliches Programm (BASIC, ASSEMBLER, LLIST,...) an, bis die Taste F8 gedrückt wird. Der Interrupt wird im MTX Jede 1/125 Sekunde generiert. Statt den Tasten F4,F7 und F8 können beliebige andere außer BRK, ALPHA LOCK, SHIFT, CTRL und den RESET-Tasten verwendet werden. In den entsprechenden CP-Befehlen muß der passende ASCII-Code eingesetzt werden. Während der User-Interrupt aktiviert ist (Ausschalten mittels F7), sind die verwendeten Tasten für andere Zwecke nicht verfügbar, und die Tastatur reagiert im Dialog etwas empfindlicher, da das Auto-Repeat der Tastatur auch über diesen Interrupt (INTFF Bit 2) gesteuert wird. Wenn also im Dialog ein Programm bearbeitet werden soll, empfiehlt sich ein kurzer Druck auf F7, um den User-Interrupt auszuschalten. Herbert Herberg

ALICE-W Fragen:

1. Wie kommt man aus Grimbley Forest raus?
2. Was kann man mit dem Caterpillar auf dem Mushroom tun?
3. Wie kommt man am Drachen vorbei, bzw. wie mit dem Schwert dorthin?

32 Seiten ROM-Info (Herbert Herberg)

Wir haben leider noch kein kommentiertes ROM-Listing und sind hier für jeden Hinweis sehr dankbar!

Inhalt der 32 Seiten: Cassettenhandling, ROM-Jump-Tables, Joystick & Keyboard-Tips, VRAM & Videocontroller-Info. Wer diese Unterlagen haben möchte, kann sie bei Herbert Herberg für DM 8.- (incl. Porto + Verpackung) bestellen.

TAXAN-Farbmonitor nicht gut!! Liefert oft nur Schwarz/Weiß!!

FDX Single-Disk-Routinen: (Herbert Herberg)

1. **Formatieren:** FRM.COM **ACHTUNG:** Alle wichtigen Disketten, insbesondere Systemdisketten sollten einen **Schreibschutz** (zugeklebte Kerbe der Diskette) haben!!
2. Mit dem Programm SYS.COM können die (anders nicht verwendbaren) **Systemspuren** der Diskette mit CP/M beschrieben werden.
3. Mit COPY.COM können Diskettendateien mit nur einem Laufwerk **kopiert** werden.

Beachten Sie bei Jeder Meldung den Hinweis, welche Diskette eingelest werden soll!! 'source disc' = Diskette mit Daten, 'destination disc' = Diskette, auf die geschrieben werden soll, 'system disc' = Diskette mit den o.g. Programmen (mit Schreibschutz!!). Diese drei Routinen sind auf der mitgelieferten Diskette 'Disc Basic and Utilities'. Die Beschreibung hierzu im FDX-Handbuch sind in der Rubrik Disc Basic & Utilities gleich zu Beginn. Die auf der CP/M 2.2 System Diskette gelieferten Routinen (FORMAT.COM, SYSGEN.COM, PIP.COM) sind nicht für Systeme mit einem Laufwerk geeignet!

FDX-BASIC: (Herbert Herberg)

Wer keinen Monitor an der Floppystation anschließen will, sondern BASIC mit dem MTX-Schirm betreiben will (evtl. mangels 80-Zeichen-Monitor) kann folgendes tun:

Auf eine formatierte (FRM.COM oder FORMAT.COM) Diskette mit Systemspuren (SYS.COM oder SYSGEN.COM) die Files **FDXB.COM** und **STARTUP.COM** kopieren und das Kommando **startup fdxb 40** eintippen. Es bewirkt, daß beim Einschalten des Computers mit dieser Diskette im Laufwerk B sofort **FDXB 40** als Befehl ausgeführt wird. Dieser ruft das **FDX-BASIC** auf, daß mit dem Videocoutput im MTX (und nicht der 80-Zeichen-Karte in der FDX) arbeitet. Aber es kennt Floppy-Befehle! (Zum Formatieren, ... siehe den vorigen Abschnitt!)

Merkwürdige BASIC-Befehle: (Herbert Herberg)

Einige BASIC-Tokens sind zwar Belegt, haben aber noch keinen Zweck oder bewirken dasselbe wie andere! Außer den im Handbuch und dem o.g. USER (Nur bei MTX ohne FDX) gibt es noch die folgenden Befehle: (alles andere, was man sonst entdeckt ist redundant!)

BASIC-Function INP: LET I=**INP**(P) liefert in I den Wert vom Port P, ist das Gegenstück zu OUT

ASSEMBLER-Befehle: **DW** = Define Word (entsprechend **DB** = Define Byte)

WO: DW #4007

LD HL,(WO) liefert in HL den Wert #4007,

Hingegen

WO: DB #40,#07

LD HL,(WO) liefert in HL den Wert #0740

P = print, zu verwenden wie L, aber für den Drucker
PWO listet ab WO auf den Drucker

C = CLS löscht den Bildschirm (zu verwenden wie T)

FDX-Handbuch: (Herbert Herberg)

Wer hat die FDX-Doppellaufwerkkonfiguration fertig gekauft? Die dort beige-fügten Unterlagen sollen ausführlicher sein als die für die DM 1500.- FDX-Single-Disc-Station von Vobis. Insbesondere der Abschnitt 'Technical Manual' ist hier sehr kurz geraten (20 Zettel incl. Einbauanweisung für RS 232C-Karte). Insbesondere interessieren mich Unterlagen über den Zugriff auf die 80-Zeichen-Karte. (Siehe auch NewWord oben bzgl. fehlenden Unterlagen: Encyclopedia, Installation Manual.)

ROM-Listing selbstgemacht: (Herbert Herberg)

Aufruf: GOTO 3

Folgendes Programm kopiert einen bestimmten Teil der ROM's oder des Systembereiches beim FDX-BASIC in eine Assemblerzeile, die man dann mühelos auf den Drucker schicken kann. Warnung: 4kByte liefern ca. 37 Seiten auf den Drucker (ohne Vorschub):

Um in Assembler dieselbe Zeile x-fach einzugeben kann man folgenden Trick anwenden: Die Zeile einmal oben eintippen und mit <RET> einspeichern; dann mit <CLS> <RET> <T> zum Beginn positionieren (wo diese eine Zeile steht) und <RET> um in den Insert-Modus zu kommen, wobei die Zeile angezeigt wird; durch <RET> wird dieselbe Zeile eingefügt, existiert also zweimal; u.s.w (gef. RET-Taste belasten/festhalten).

1 CODE

```
8007 DS 255 ;Soviel DS, daß Zeile 2 bei 9000 (Hex) anfängt.
      u.s.w. ;sen dann die unteren 3 Byte der Adressen.
```

2 CODE

```
9000 LDIR ;4 kByte mit LDIR vollgeschrieben.
9002 LDIR ;Auf keinen Fall einen anderen Assembler-Befehl nehmen,
9004 LDIR ;da keiner diese Aufgabe besser erfüllt. Wenn hier mit
      ;DS 255, oder LD (IX+0),0 gearbeitet wird, wird ein
      u.s.w. ;Symbol-Table (in dem dann die 255 bzw. 0 stehen) angelegt,
      ;der dann das ganze Projekt unterminiert!!
```

```
AFFE LDIR
B000 RET
```

3 CODE

```
DI
LD HL,anfangsadresse des ROM/System-Bereiches
LD DE,#9000 ;Ziel
LD BC,#2000 ;Länge
LD A,ROM-page mal 16 (nur für ROM, nicht für FDX)
OUT (0),A ;Nicht für FDX, umschalten auf entspr. Seite im MTX-ROM
LDIR
LD A,0
OUT (0),A ;Nicht für FDX
EI
RET
```

Zeile 2 enthält dann das Gewünschte, und es kann dann mit LLIST 2,2 oder mit ASSEM 2, und dann dem P-Befehl ausgedruckt werden. **Aber** wie aus den ROM-Unterlagen hervorgeht stehen überall Daten herum, die so nicht entdeckt werden!! Die Assembler-Adressen oben sind für den MTX 500 und für FDX-Systeme. Für den MTX 512 (oder erweiterten MTX 500) ohne FDX wird aus #8007 eine #4007, #9000 -> #5000, #E000 -> #7000

NEW / RESET (Herbert Herberg & Martin Müller)

An ein mit o.g. Methoden gelöscht Programm kann man noch drankommen, mit POKE 64167,1. Dann kann man etwas noch listen. Um vollständig damit arbeiten zu können: im Handbuch stehen hinten die Adressen BASTOP, BASTPO, NBTOP, in die jeweils das Ende des BASIC geschrieben werden muß. Allerdings darf ein solches Programm kein Noddy enthalten.

Druckerausgabenveränderung: (Herbert Herberg)

Ein Programm hierzu liefern wir demnächst!

Aber hier ein paar Adressen aus dem ROM bzw. FDX-BASIC: (B = B-Register)

Output B to Screen **#0CDC**

Output B to Centronics **#0CE0**

Output B to RS 232C 0 **#0CCF**

Alle drei werden bei jeglicher Ausgabe mittels LIST oder LPRINT angesprochen, und verändern das A-Register. (CALL #0CF3 überprüft nur die BREAK-Taste, und unterbricht ggf.)

Wer also einen eignen Zeichensatz, oder ähnliches (z.B. einschalten der Sonderzeichenebene für Olympia Compact 2) auf den Drucker schicken will, kann zumindest beim FDX-BASIC einfach an die Stelle der entspr. Routine einen JP-Befehl zu einer selbstgeschriebenen Konvertierungsroutine setzen. Die eigentliche Ausgabe des Zeichens würde ich aber in der selben Form wie im System programmiert belassen. Achtung: wenn irgendwelche Register außer dem AF gebraucht werden, sollten diese mit PUSH/POP gesichert werden. **Warnung:** Ich warne Jeden, der sich nicht mit Assembler auskennt davor hier allzuviel herum zu probieren! Sonst bleibt nur noch RESET!! Außerdem würde ich nicht in den 'Output to Screen'-Teil eingreifen!

Wenn jemand sich da etwas zurechtgebastelt hat würde ich mich freuen, wenn ich eine Cassette oder Diskette incl. Kommentaren bekommen könnte!

BASIC

Ab Hex 4000 (MTX 512 o. FDX) bzw. Hex 8000 wird das BASIC-Programm abgespeichert. Die Art und Weise der Abspeicherung ist in den 32 Seiten ROM-Unterlagen enthalten.

Cassette:(Martin Müller & Herbert Herberg)

Vor dem SAVE sollte man CTRL und G gleichzeitig herunterdrücken. Dabei wird der Ascii-Character 7 ausgegeben: BELL (Glocke). Dies bewirkt ein Aufhören des Rauschens bei nicht vollständig ausgeschaltetem Tongenerator.

FEXPAND, USERNOD (Herbert Herberg)

An den o.g. Adressen können JP-Befehle zu selbstgestrickten Erweiterungen für das PANEL bzw. für NODDY stehen, die mit RET beendet werden müssen. Beim PANEL wird erst die Expansion, und dann das eigentliche PANEL aufgerufen! Wie das bei NODDY gehandhabt wird, weiß ich nicht.

Zu Verkaufen:

Centronics Matrix-Drucker (80 und 132 Zeichen Je DIN A4-Zeile; 6 Zeichensätze; Breitschrift; eingebaute Centronics-Buchse (parallele Schnittstelle); Einzel-, Endlos- und Rollenpapierverarbeitung; Halterung für Rollenpapier; voll New Word und Supercalc-Fähig (Fettdruck etc.)) incl. Handbuch und Druckerkabel für MTX (Flachbandkabel) billig abzugeben! VB DM 300.-.

Herbert Herberg, Sonnenau 2, 2 Hamburg 76, (040) 200 87 04

Fragebogen: (Herbert Herberg)

Bitte füllen Sie den beigefügten Fragebogen soweit aus wie Sie wollen und schicken Sie ihn an uns. Überprüfen Sie aber bitte sorgfältig Ihren Namen, Ihre Adresse und Ihre Telefonnummer in der Mitgliederliste und weisen uns auf Fehler hin.

Datenbankprogramm, das auf MTX ohne CP/M läuft gesucht!

NewWord/FDX läuft nicht, dann ist vermutlich wie bei mir (Herbert Herberg) der MTX defekt! Umtauschen!!

M.CODER ist ein BASCI-Compier für MTX! Wer kennt ihn, hat ihn, weiß wo es ihn gibt?

Kritik & Verbesserungsvorschläge hören/lesen wir jederzeit gerne! Nur so erfahren wir, was wir falsch machen! Aber auch zu wissen, was gut ist hilft uns.

Wie Sie sicherlich gemerkt haben, sind die einzelnen Artikel nicht sortiert. Das würde einfach zu viel Zeit kosten. Alles wird einfach in der Reihenfolge eingetragen, wie es hier eintrifft und bearbeitet wird.

Kontostand: (Herbert Herberg)

Anbei mit diesem Info finden Sie eine kleinen Zettel mit Ihrem Kontostand. Falls Sie irgendwelche Fragen oder Beanstandungen oder Zweifel haben lassen Sie es mich bitte umgehend wissen. Bitte beachten Sie die folgenden evtl. bei Ihnen entstandenen Posten: DM -.70 für 1. Information (s.o.), Kosten für 32 Seiten ROM-Info und zugestellte Kassetten/Disketten und DM 4.70 für dieses Info!

HALLO MTXer!

7

Das nächste mal gibt's mehr als die folgenden fragen und antworten und fragen.. und das kleine fract'1; womöglich auch neuigkeiten aus osnabrück,berlin und aachen? Aber damit die erste nummer raus kann erstmal bloß dies:

nein sven, man kann nichts auf den rand schreiben oder zeichnen (es sei denn du überlistest den multiplexer im blockschaltbild auf seite 206,schätze ich - abweichende meinungen und lernstoff immer willkommen - wie alles - bis auf die besonders eklige reklamepost, die jetzt ins haus kommt ---) und worauf auch viele hoff(t)en: nein, die befehlsbelegung der funktionstasten ist fest! Allerdings kann man im anwenderprogramm dafür davon steuern lassen, was man mag. Das tut NewWord ja auch sehr schön und viel "praktischer" als das MTX-basic.

Eine andere oft gestellte frage wird dagegen mehrfach positiv beantwortet, nämlich 'wie drucke ich den VS4 am besten?'

Wie die glücklichen besitzer des deutschen handbuchs auf seite 118 gelesen haben ist das gar nicht so schwer, aber langsam. Und klein.

Dagegen faßt die routine im Info2 die ursprünglich 8 kritischen zeilen (!) zusammen und läuft (geschätzt) doppelt so schnell. - A.S.hat eine vergrößerungsroutine verfaßt, die beeindruckender aussehende bilder erzeugt als die kleinen 1:1 pixel->dot formate. Wie man flotter druckt zeigt Muehrlings lösung. Und als alleskönner noch eine 1:1::2:3 doppelt saubere version aus dem mutterland unserer Maschine. Ich hoffe bis zur nächsten nummer die fractals damit kopiererfähig zu bekommen..

Im gleichen Info2 wird auch nach NewWord gefragt. Hierzu kann gesagt werden, daß sowohl der fragesteller wie auch ich damit sehr zufrieden sind. Klar - auch dieser text wird damit geschrieben. Bloß schade, daß ich keine zeit (und fertigkeit -noch..) habe ihn denn auch entsprechend auszufeilen, wie es mit diesem werkzeug möglich ist.

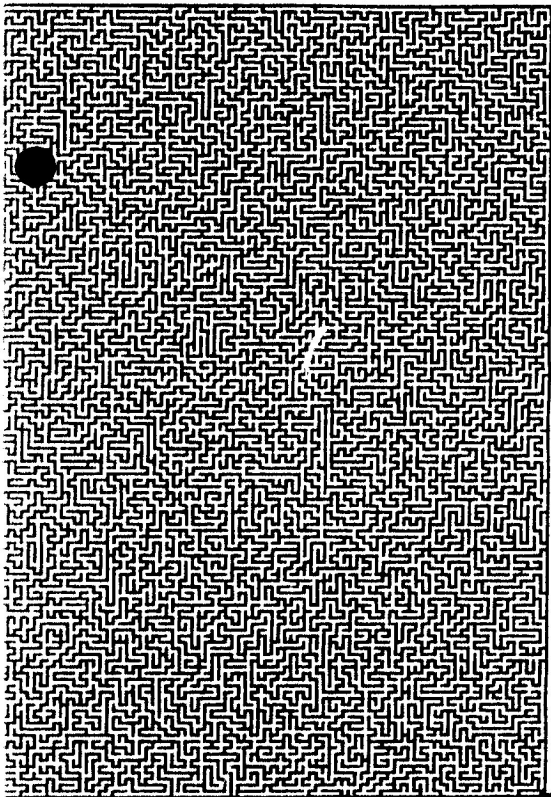
Die zuverlässigkeit der datenabspeicherung auf kassette nimmt natürlich!!! besonders ab wenn man nicht ausdrücklich den soundbuffer löscht (SBUF 2) und alle kanäle auf 0 setzt..aber es geht trotzdem ganz gut und klingt echt wild.

Auf einen farbmonitor werde ich erstmal verzichten; der s/w (Sanyo) bringt den text optimal und aus der 80zeichenfarbkarte kommen nicht halb so viele farben und bloß 90x160 pixel. Dafür steht der fernseher daneben und liefert denn guten alten VS4 gleichzeitig. Dezidierter äußern sich etliche stimmen gegen den TAXAN Vision EX, der wie ich erfahre nicht mal farbig läuft..

FRE(N) entspricht nutzloserweise EXP(N) und liefert nicht etwa den freien raum. OFF ist nobby und heißt eigentlich *0 oder *OFFSTACK. Die funktion LEN(A\$) zählt bei mir die zur darstellung erforderlichen stellen, wozu eben auch ein nicht sichtbares positives vorzeichen gehört, weswegen LEN(STR\$(23)) nicht 2 sondern drei bringt. Es soll abweichende erfahrungen geben!

Sicher hat jeder schon gemerkt, daß die wahrheitswerte null=falsch aber minus eins=richtig lauten und nicht wie im engl.Hdb.plus eins. Aber auch das einfache ist schwer und alles möglichekann passieren. Überhaupt wollen wir vorerst möglichst alles weitergeben, ohne filter - so gut wir können. So das wär's erstmal. Schade daß niemand Laufzeitmessungen geschickt hat...

Naja - kommt Zeit kommt Rat..tschüß, thomas.



HARDCOPY-ROUTINE FÜR DEN MTX FÜR EPSON-KOMPATIBLE DRUCKER IN VIERFACHER GRÖßE

COPYRIGHT 1984 BY ANDREAS SCHMITT, KAISERSTR. 54, 8000 MÜNCHEN 40

COPY IN 8 MIN 48 SEC

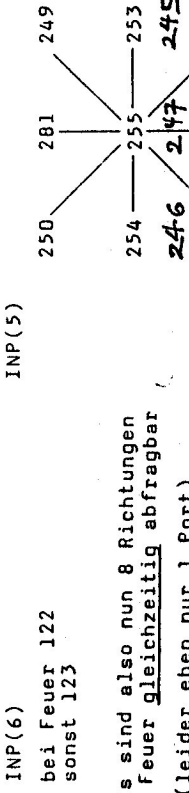
```

1000 VS 4
1005 REM*****BEGIN DER HARDCOPY-ROUTINE*****
1010 DIM A$(260): DIM B(4)
1020 LPRINT CHR$(27);"A";CHR$(8)
1030 LPRINT CHR$(27);"P";CHR$(0)
1040 FOR Y=192 TO 1 STEP -4
1050 LPRINT CHR$(27);"K";CHR$(0);CHR$(2);
1060 FOR X=0 TO 255
1070 FOR Z=1 TO 4
1080 LET B(Z)=ASC(IGR$(X,Y-Z,1))
1090 NEXT Z
1100 LET A$(X+1)=CHR$(192*B(1)+48*B(2)+12*B(3)+3*B(4))
1110 NEXT X
1140 FOR I=1 TO 256
1150 LPRINT A$(I);A$(I);
1155 NEXT I
1160 LPRINT
1170 NEXT Y
1180 STOP

```

Wenn diese BASIC-Lösung zu langsam ist, der findet in der Toolbox ein Assemblerprogramm.

W.W.'s Entdeckung: Wer hat sich noch nicht darüber geärgert, daß der Joystick mit INKEY\$ nur in 4 Richtungen abfragbar ist? Hier die Lösung: Benutzt man den linken Joystickport und fragt mit INP(5), INP(6) ab, so erhält man folgende Werte:



Es sind also nun 8 Richtungen und Feuer gleichzeitig abfragbar (leider eben nur 1 Port)

c) INTERRUPT-Steuerung, kein Problem mehr!

Zuerst wird in den PANEL gesprungen, man listet ab*FA98 auf und findet dort ein RET, gefolgt von 2 NGr's. Diese 3 Adressen (FA98-FAA0) muß man mit dem Objektcode eines Maschinensprachbefehls versehen um eine Assembleroutine 64mal in der Sekunde abarbeiten zu lassen:

z.B. D FA98 (in PANEL eingeben)

```

es erscheint: Du gibst ein:
FA98 XX (RET)
FA99 XX (RET)
FAA0 XX (RET)

```

Wenn Du jetzt erneut listest, erscheint bei FA98 JP #4010, das ist die Anfangsadresse des Maschinenprogrammes.

Hier eine interruptbetriebene Uhr:

```

0 GOTO 100
1 CODE

```

```

4010 LD A,169
OUT (2),A
LD A,95
OUT (2),A
LD A,(64857)
OUT (1),A
LD A,(64858)
OUT (1),A
LD A,58
OUT (1),A
LD A,(64859)
OUT (1),A
LD A,(64860)
OUT (1),A
RET

```

100 REM PROGRAMMANFANG

Gibt man jetzt GOTO 1 ein, passiert vorläufig gar nichts, bzw. das Programm läuft nur einmal durch.

Versucht man aber dann POKE 64862,31 (User-Bit in INTFF gesetzt) so erscheint die Uhr am Textbildschirm und erneuert sich fortwährend ohne jeglichen Eingriff.

ACHTUNG: Zum Programmieren mit POKE 64862,15 wieder ausschalten, da sonst Absturzgefahr herrscht!

c) Noch ein kleiner Tip: Der benötigte Speicherplatz eines Programmes entspricht ca. 256xPEEK(64205)+PEEK(64204)

d) Aber jetzt der RIESEN-Tip: Abspeicherung von Speicherblöcken des RAM's, ebenso einlesen ohne Verlust des Programmes!!! (aus YOUR COMPUTER Sept. 1984)

Assemblerlisting **siehe oben!**

← für Port L und Cursor-tasten: Siehe 32 Seiten Rom-Info!

```

1 REM -----fract'1-----<VERSION FRC1.BAS / 24.10 .1984>
2 GOTO 2000
5 VS 5: CLS : PLOT 69,0: FOR D=0 TO PI+PI STEP PI/2: ANGLE D: DRAW 90: NEXT
6 CSR 50,0: INK 4: PRINT "FRACTALS" " : INK 0: GOSUB 1100
50 RESTORE 51: READ WM,KM,CX,CY: REM <wertmax.,versuchstiefe,konstante:real/im>
51 DATA 41,50,-0.26,0.001
55 RESTORE 56: READ AS,BS,AZ,BZ,STZ,STS
56 DATA -63,128,-47,92,1,1
60 DIM C(15): RESTORE 61: FOR A=1 TO 15: READ C(A): NEXT : REM <FARBTAFFEL>
61 DATA 15,11,2,7,5,13,0,9,14,10,3,12,4,6,0
70 LET F=0: REM <farbsumme>
100 VS 4: FOR AT=0 TO 3: ATTR AT,0: COLOUR AT,15: NEXT : COLOUR 4,14: COLOUR 2,1
200 FOR Z=0 TO 191 STEP STZ: LET YS=(Z+AZ)/BZ: LET YS2=YS*YS: FOR S=0 TO 255 STEP STS: LET X=(S+AS)/BS: LET X2=X*X: LET K=0: LET Y=Y
S: LET Y2=Y2
210 LET K=K+1: LET Y=2*X+Y-CY: LET X=X2-Y2-CX: LET X2=X*X: LET Y2=Y*Y: IF K<KM AND (X2+Y2)<WM THEN GOTO 210
300 LET F=F+K: LET A=MOD(S,8): LET B=MOD(K,2): ATTR 2,B: IF A=7 THEN LET FC=INT(F/8): COLOUR 3,C(MOD(FC,14)+1): VS 5: CSR 73,0: PRIN
T FC;" " : VS 4: LET F=0
310 PLOT S,Z: NEXT : NEXT
400 VS 4: REM ----drucken----
410 LPRINT CHR$(27);"A";CHR$(8);: LPRINT CHR$(27);"P";CHR$(0)
420 FOR Y=24 TO 1 STEP -1: LPRINT CHR$(27);"K";CHR$(0);CHR$(1);: FOR X=0 TO 255: LET A$=BR$(X,8*Y-1,8): LPRINT A$;: NEXT : LPRINT :
NEXT : LPRINT
900 VS 4: GOTO 900
VS 4: COLOUR 2,15: ATTR 2,0: CLS : RUN : REM <--start mit cls
1100 VS 5: CSR 0,0: INK 0: PAPER 5: PRINT TIME$,: PAPER 0: RETURN : REM <zeit>
2000 DISC SAVE "FRC1.BAS": GOTO 5

```



Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

COMMAND 1: CTLSPR p,x

p = parameter and can be any of the six below:

- 0 Speed
1 to 255 to 0 (1 is fastest)
- 1 Distance
Tells the computer to move the sprite by 'x' pixels when requested.
- 2 Number of sprites
0 to 32 (The number of sprites must be at least 1)
- 3 Number of circling sprites
Sprites that will orbit when they go off the edge of the screen (must not exceed total number of sprites)
- 4 Plot sprite
A PLOT SPRITE can be chosen which will subsequently appear whenever a point is plotted. This sprite will move around the screen following any points or lines drawn by the BASIC GRAPHICS commands. This sprite can be any of the 32 defined in the normal way.
- 5 Number of moving sprites 0 to 32
This is the number of sprites that will move by themselves according to the x-speed and y-speed set in the SPRITE and ADJSPR commands.
- 6 Magnitude and size

x=0	size 8X8	mag 1
x=1	size 8X8	mag 2
x=2	size 16X16	mag 1
x=3	size 16X16	mag 2

COMMAND 2: GENPAT p,n,d1,d2,d3,d4,d5,d6,d7,d8

The GENPAT command is the command used to generate all types of patterns required by BASIC for characters and SPRITES. There are 5 modes.

- 1 To redefine an ASCII character. (CODES 32 TO 127)
- 2 To define a non ASCII character. (CODES 129 TO 154)
- 3 To define colour for each line of a character.
This only applies to user definable characters with codes 147 to 154.
- 4 To define an 8 by 8 sprite pattern.
- 5 To define each quadrant of a 16 by 16 sprite.

User definable characters have codes from 129 to 154.

Mode 1 allows the user to redefine one of the standard ASCII character patterns. Note that the ASCII characters are the ones which are most often used by the computer

Mode 2 allows the user to define his own character patterns without destroying any of the standard ASCII characters.

Mode 3 allows some of these user definable characters to be further defined by specifying an ink and paper colour for each of the eight rows of the character.

The values for ink and paper are as specified in the table in CHAPTER 1 but in this instance we are specifying two colours (ink and paper) at the same time. Each of d1 to d8 specify a paper and ink colour as a single number:

```

bit 0 1 2 3: 4 5 6 7
   ink  : paper
value = 16 * paper + ink
  
```

e.g. Red ink on blue paper

```

= RED : BLUE
=BLUE * 16 + RED
= 4 * 16 + 9
= 73
  
```

MODE	P	N
1	0	ASCII code (32 to 127)
2	1	user definable (code 129 to 154)
3	2	
4	3	pattern number 8 by 8 sprite pattern
5	4	pattern number 16 by 16 NW quarter
	5	pattern number 16 by 16 SW quarter
	6	pattern number 16 by 16 NE quarter
	7	pattern number 16 by 16 SE quarter

COMMAND 3: SPRITE

SPRITE (= create sprite)

SPRITE n,pat,yp,yp,xs,ys,col

n is sprite number 1 to 32
pat is pattern number 0 to 127 (size 0)
0 to 31 (size 1)
xp is position x off centre

yp is position y off centre (in range -4095 to 4095)
0,0 is defined as bottom left hand corner of screen i.e same as for plot.
NB Sprite coordinates are absolute and do not look at virtual screen origins (i.e assume a 32 by 24 graphic screen)

xs is the speed in the x direction range -128 to 127 where 1 unit of speed moves the sprite 1/8 pixel every master speed cycle as set by CTLSPR 0

ys is the speed in the y direction (plus = upwards) range -128 to 127

col = colour 0 to 15

COMMAND 4: MVSPR p,n,d

MVSPR is a general purpose command which combines 4 distinct functions:

p meaning

- 1 MOVEMENT
- 2 PATTERN SELECTION
- 4 REDIRECT
- 8 PLOT AT CENTRE

The functions are combined to allow complicated movements to occur whilst using only a single instruction. The type of activity is selected by p as in the table above. If combinations of activities are required, just add the p values together. Some examples are given below.

As before n selects the sprite number.

d is slightly more complicated as it must be able to reflect a value for several activities. If d is not in the range of any one of the chosen activities an error will occur.

MOVE (p=1) moves the sprite 1 step in the direction specified by d. The step size is set in CTLSPR 1 and the direction must be in the range 0 to 8 where directions 0 and 8 are the same.

PATTERN changes the sprite pattern to pattern number d. This pattern should have been defined in a GENPAT statement.

REDIRECT picks up the current velocity vector and switches it to the new direction.

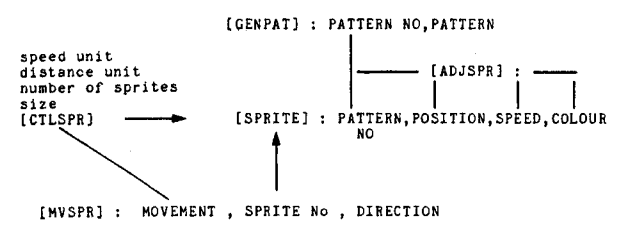
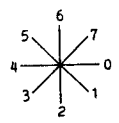
PLOT AT CENTRE causes a point to be plotted at the centre of the sprite specified by n. This is not directly affected by the value of d at all.

COMMAND 5: ADJSPR p,n,v

p	meaning	range of v
0	pattern	0 to 31 (size 1) 0 to 127 size 0
1	colour	0 to 15
2	x pos	0 to 255
3	y pos	0 to 255
4	x speed	0 to 255 (128 to 255 = neg)
5	y speed	0 to 255 (" " ")

COMMAND 6: VIEW direction, distance

direction = 0 to 7
distance = 1 to 255 to 0



CONTROL SEQUENCES

CTL Dn	Sets background colour to n
CTL E	Erase to end of line
CTL Fn	Sets foreground colour to n
CTL G	Sounds the bell
CTL H	Backspace, cursor left
CTL I	Tabulate the next block of eight
CTL J	Line feed, cursor down
CTL K	Cursor up
CTL L	Clear screen and home cursor
CTL M	Carriage return, cursor to left
CTL W	Tab back
CTL Y	Cursor forwards
CTL Z	Homes cursor
CTL	Page mode
CTL \	Scroll mode
CTL ^	Cursor on
CTL _	Cursor off

TABLE 4. Colour Assignments

COLOUR (HEX)	COLOUR	LUMINANCE (DC VALUE)	CHROMINANCE (AC VALUE)
0	Transparent	0.00	-
1	Black	0.00	-
2	Medium Green	.60	.60
3	Light Green	.80	.53
4	Dark Blue	.47	.73
5	Light Blue	.67	.60
6	Dark Red	.53	.53
7	Cyan	.80	.73
8	Medium Red	.67	.73
9	Light Red	.80	.73
A	Dark Yellow	.87	.53
B	Light Yellow	1.00	.40
C	Dark Green	.47	.60
D	Magenta	.60	.47
E	Grey	-	-
F	White	1.00	-

ESCAPE SEQUENCES

ESC S	Standard character font
ESC B0	American character font
ESC B1	English character font
ESC B2	French character font
ESC B3	German character font
ESC B4	Swedish character font
ESC B5	Spanish character font
ESC I	Inserts a blank line at cursor line
ESC J	Deletes the current cursor line
ESC K	Duplicates a line
ESC Xc	Simulates CONTROL character c

APPENDIX 1

ASCII CODES

ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC
NUL	00	0	/	2F	47	5E	94	
SOH	01	1	0	30	48	5F	95	
STX	02	2	1	31	49	60	96	
ETX	03	3	2	32	50	61	97	a
EOT	04	4	3	33	51	62	98	b
ENQ	05	5	4	34	52	63	99	c
ACK	06	6	5	35	53	64	100	d
BEL	07	7	6	36	54	65	101	e
BS	08	8	7	37	55	66	102	f
HT	09	9	8	38	56	67	103	g
LF	0A	10	9	39	57	68	104	h
VT	0B	11	:	3A	58	69	105	i
FF	0C	12	;	3B	59	6A	106	j
CR	0D	13	<	3C	60	6B	107	k
SO	0E	14	=	3D	61	6C	108	l
SI	0F	15	>	3E	62	6D	109	m
DLE	10	16	?	3F	63	6E	110	n
DC1	11	17	@	40	64	6F	111	o
DC2	12	18	A	41	65	70	112	p
DC3	13	19	B	42	66	71	113	q
DC4	14	20	C	43	67	72	114	r
NAK	15	21	D	44	68	73	115	s
SYN	16	22	E	45	69	74	116	t
ETB	17	23	F	46	70	75	117	u
CAN	18	24	G	47	71	76	118	v
EM	19	25	H	48	72	77	119	w
SUB	1A	26	I	49	73	78	120	x
ESC	1B	27	J	4A	74	79	121	y
FS	1C	28	K	4B	75	7A	122	z
GS	1D	29	L	4C	76	7B	123	
RS	1E	30	M	4D	77	7C	124	
US	1F	31	N	4E	78	7D	125	
space	20	32	O	4F	79	7E	126	
!	21	33	P	50	80	7F	127	DEL
"	22	34	Q	51	81			
#	23	35	R	52	82			
\$	24	36	S	53	83			
%	25	37	T	54	84			
&	26	38	U	55	85			
'	27	39	V	56	86			
(28	40	W	57	87			
)	29	41	X	58	88			
*	2A	42	Y	59	89			
+	2B	43	Z	5A	90			
,	2C	44		5B	91			
-	2D	45		5C	92			
.	2E	46		5D	93			

FUNCTION KEYS

The Function Keypad can be used to customise the computer for a particular application. There are eight keys marked F1 to F8.

F1	128	SHIFT and F1	136
F2	129	" " F2	137
F3	130	" " F3	138
F4	131	" " F4	139
F5	132	" " F5	140
F6	133	" " F6	141
F7	134	" " F7	142
F8	135	" " F8	143

If required, character patterns can be assigned to the function keys using the GENPAT statement.

For example,

```
10 GENPAT 1,129,32,80,136,136,248,136,136,0
```

will make F2 produce a character 'A'.

Virtual Screens - Byte format for each screen

Byte No	Contents
1	Screen Type, Auto Scroll, Cursor Flash, Page Mode
2	Current print position in virtual screen
3	2nd Byte of above
4	Absolute top left-hand corner
5	2nd Byte of above
6	Size of screen in characters
7	2nd Byte of above
8	Line width of Physical screen
9	Holds Cursor character
10	Border colour, Paper, Ink
11	Print colours: Ink, Paper; Print Attributes
12	2nd Byte of above
13	Non-print colours: Ink, Paper; Non-print Attributes
14	2nd Byte of above
15	Scroll count

CRVS n,t,x,y,w,h,s

To create your own virtual screen, use the CRVS command to define the required area and then select the screen using the VS command.

- n Virtual screen reference number. (0 to 7)
- t Type of screen. 0=text,1=graphics.
- x Coordinates of top left corner.
- y
- w Width of virtual screen in characters.
- h Height of screen in lines.
- s Width of screen. (40 for text, 32 for graphics)

If s is a different value, to the actual width of the screen, the virtual screen will become distorted. This can however be used to advantage. If for example s is set to 80 in a text screen, the virtual screen will only allow printing on alternate lines.

GRAPHICS FUNCTIONS

SPK\$ (screen peek)
Gives the character at the cursor location on the current text screen.

e.g LET A\$ = SPK\$

Uses: storing screens

GR\$ (x,y,b)
x and y are locations on the virtual screen

b is number of bits read.(If b = 1 equivalent to "POINT" function)

bits are vertical bits ie GR\$(20,190,4) gives a character made up as follows:

- bit 7 0
- bit 6 0
- bit 5 0
- bit 4 0
- Bit 3 pixel at 20,190
- bit 2 pixel at 20,189
- bit 1 pixel at 20,188
- bit 0 pixel at 20,187

DSI

Direct Screen Input

This command allows you to roam about freely within a screen only ending when carriage return is pressed. Within this instruction, the break key is not operational but will generate CTL C.

- CTL W = tab back
- CTL | = PMODE
- CTL \ = SMODE
- CTL ^ = CURSOR ON
- CTL _ = CURSOR OFF
- CTL D letter A to O = paper A to O (1 to 15)
- CTL F " " ink A to O (1 to 15)
- ESC I = insert line
- ESC J = delete line
- ESC K = duplicate line

EDITOR <variable list>

The editor gives the programmer the facility to accept input from a defined area of the screen. The area is defined by virtual screen 0 which may be set using the CRVS command.

For example:

```
10 CRVS 0,0,20,10,10,1,40
20 EDITOR A$
30 VS 5
40 PRINT A$
50 GOTO 20
```

The EDITOR leaves the current screen as screen 0 and must therefore be reset if printing is required on the full BASIC screen or any other screen. Line 30 resets the current screen to the full basic screen.

MTX 512 - HANBUCH - SUCHREGISTER

- A** ASCII-Codes - 127, Abkürzungen - 131, AND - 155, ADVANCE - 86, Abprüfen eines Progr. - 13, Assemble - 125, ASC - 132, ARC - 132, ANGLE - 131, 36,
- B** Bildschirmzeilen - 10, Bildschirmfarbe - 15, Bootische Ansteuerung - 135; BRANCH - 81, Bandtakt f. Kassette - SAVE - 176, 190, Bildschirmdruck (Drucker) - 164, Beugungsrichtung - 116, 118, Bootische Ansteuerung (Drucker) - 164, Beugungsrichtung - 116, 118, Bootische Ansteuerung (Text) - zu Bildschirm (grafik) - 117, BRK - 6, Bruchzahl - 152
- C** Centronics I/F - 237, 239; ^{91, 94, 10}CLS - 5; ^{118, 93}CLOCK - 29; CTL-Codes - 171; ⁵⁴CIRCLE - 137; ¹⁶⁶CSR - 63, 91, 790; CTRL-Taste - 93, CRYS - 92, Copy - 164, Cassetteneinanderanschluß - 7, CTLSFR - 140, CODE - 125, CP/M - 233, CHR\$ - 137,
- D** ^{45, 37, 141}DATA - 5; Druckerdruck - 25, 88; DIM - 53, 54, 59; DIR - 78, DISPLAY - 77, DST - 93; Disketten Betriebssystem - 233, Druckermatrix-Druck - 147, DRAW 97, Dokumentlauf - 53; Division-Reet - 152
- E** ⁴²ELSE - 9, 12, 41; ¹⁴⁸Erweitern - 11, 12; ESC-Codes - 171, 95; ENTER - 80, Erweiterungspalten - 152; Erweiternspalten-Druck - 197, EOL - 6,
- F** ⁷⁴⁵FARB-CODES - 15, 171; ⁷⁴⁵FIND-Diagramm - 31, FOR NEXT STEP - 47, ⁷⁴⁵Funktionsgruppen - 172, ⁷⁴⁵Funktionsgruppen - 180, 191, FORTH - 153,
- G** GOTO - 45, GOTO (Noddy) - 81, 86, Grafik - 91, GR\$ - 117, Grafik-Druck (Drucker) - 14; ⁷⁴⁵GR\$ - 147, Grafik-Rechen - 146, GENPAT - 146, Grad - 146, Grad - 146, Grad - 146

- H** Hierarchie - 7, 41, 136, Hochauflösende Grafik - 91, HOME - 5
- I** INK - 15, INPUT - 39, IF THEN ELSE - 41, INDIKATOR - 53, INKEY\$ - 67, IF (Noddy) - 81, INS - 5
- J**
- K** Klammerelemente - 21, ¹⁷⁶Kurzformen d. Eingabe - 131, 80, ¹⁷⁶Kassette-Druck - 176, 190; ¹⁷⁶Kassetteneinanderanschluß - 7, ¹⁷⁶Kreis-Grafik - 137,
- L** ¹⁷⁶LLIST - 25; ¹⁷⁶LOAD - 7; ¹⁷⁶Listung - 11; LIST - 11, ¹⁷⁶Logische Verknüpfungen - 41, ¹⁷⁶LEFT\$ - 68, ¹⁷⁶Labels - 81; ¹⁷⁶Länge einer Zeilenkette - 149, ¹⁷⁶LEN - 149
- M** ¹⁷⁶Matrixrechnung (= mathem. Verknüpfung mehrerer Tabellen) - 73, ¹⁷⁶MISMATCH - 10, ¹⁷⁶Multiplikation - 34, ¹⁷⁶Mittelwertberechnung - 49, ¹⁷⁶Mehrdimensionale Felder - 16, ¹⁷⁶MID\$ - 68, ¹⁷⁶MISSING SYMBOL - 85; ¹⁷⁶Monitor od. IV - 150, 191, 135, ¹⁷⁶Matr. Drucker - 147, ¹⁷⁶MIC-Prgr. - 129, ¹⁷⁶MOD - 152; ¹⁷⁶Mathematik - 152
- N** ¹⁷⁶NEXT - 153; ¹⁷⁶Noddy - 3, 77; ¹⁷⁶NEW - 9, 34, ¹⁷⁶NOT - 135, ¹⁷⁶Noddy-Programmierung - 79, ¹⁷⁶Noddy-Druck - 14; ¹⁷⁶Ausdruck - 88, ¹⁷⁶Noddy-Programmstart - 80, ¹⁷⁶NO DATA ERROR - 85, ¹⁷⁶Numerisches Isolerblock - 174

V ¹⁶⁷ VERIFY-13, Variablenamen-28, VS (virtuelle Screens)-92, VRL-45, 167

W Winkelfunktionen-65; Wurzeln-22, 65;

X

Y

Z Zeilennummer-Automatik-34, Zeichensätze-171, Zeilenduplizieren-171, Mehrere Befehle pro Zeile-63, Zeichenkettensprache-68, Zähler für Progr-Durchläufe-42, Zeichenkettenlänge-14;

O ON GOTO ON GOSUB-51, OR-135, Ordnerprogramme-55, 56, 57, OVERFLOW-85

OFFSTACK-86,

P Potenzrechnung-19, 22; PSU-1; PAGE-11, PAPER-15, Prioritätenfolge d. Operationen-136, 141, 143

PL0D-50, PAUSE-79, PAGE-TASK-73, POCAL-193, POCAL-193, Programmiersprachen-193

Programmkorrekturen-34, PRINT (Bildschirmformatieren)-63, PHI-36,

Q Quadrat zeichnen-97,

R ^{80, 28} RUN-8, Rechenhierarchie-19, 41, ¹³⁶ REM-33; ³⁷ READ-35, 43; RESTORE-61; RIGHT\$-68,

RAND, RND-69, RETURN-80, Rechen-Codes-185, 186; REM-64K-192, RGB-Memorie-

ROM-233, Radiant-Grad-96, Rechenfenster-65, Rest einer Division-152

S SBLF-120, SQR-65; ^{223, 140} SPRITES-225; ^{149, 151, 132, 144} SAVE-13; Stringverarbeitung-23, 25; Stringverknüpfung-24

STOP-36, STEP-47, Stringdimensionierung-53, Sortierprogramm-55, 56, 57, STACK-86

SCREEN-Modus-93, Systemvariablen-175, Spannungsversorgung-19, Stringvariablen-Nutzen-28

Speicherorganisation (Anleitung)-233, Speicherleiste (Anleitung)-197, SPIRAL-97

T Textverarbeitung mit Moddy-76, Tabulator-17, TIME\$-29, Temperaturrechnungen-42,

TV 68 Monitor-190, 194; TO-166, TAB-5, Tastenfunktionen-5,

U

Unterschrift:

Tagestands:

Datum:

Name:

Lehrang:

Benennung

Benennung

Benennung

Benennung

Punkte- schlüssel:	110	82	70	64	49
1	90	80	50	0	
2					
3					
4					
5					