

R65 12K GRAPHIC BASIC MANUAL

---

DER R65 12K GRAPHIC BASIC INTERPRETER IST EIN BASIC-INTERPRETER FUER DAS R65 COMPUTER SYSTEM. SEIN BEFEHLS-SATZ UMFAST ALLE STANDARD-BASIC-BEFEHLE SOWIE EINE REIHE VON ZUSAETZLICHEN BEFEHLEN UND FUNKTIONEN ZUR STRINGVERARBEITUNG, ZUR ERZEUGUNG VON VIDEO GRAPHICS SOWIE FILE HANDLING MIT MAXIMAL ZWEI TAPE UND ZWEI FLOPPY DISC DRIVES. DER INTERPRETER IST MIT DER R65 SYSTEM SOFTWARE VERKNUEPFT, DIE DIE KONTROLLE UEBER ALLE INPUT/OUTPUT OPERATIONEN UEBERNIMMT.  
DIESES MANUAL IST KEINE BESCHREIBUNG DER PROGRAMMIER-SPRACHE BASIC, SONDERN IM TEIL A WERDEN ALLE STATEMENTS UND FUNKTIONEN KURZ ERKLAERT, UND IM TEIL B WIRD AUF DIE ARBEIT MIT SPEZIELLEN TEILEN, DIE NICHT IM STANDART BASIC ENTHALTEN SIND, NAEHER EINGEGANGEN.

TEIL A: LISTE DER STATEMENTS UND FUNKTIONEN

---

1. ALLGEMEINE EIGENSCHAFTEN

---

MEMORY FUER INTERPRETER: \$2000-\$52FF (13 K)

START ADRESSEN: \$2000 FUER KALTSTART (LOESCHT ANWENDERPROGRAMM UND DATEN)

PROGRAMM MEMORY: \$2003 FUER WARMSTART  
JE NACH SYSTEMKONFIGURATION,  
DIE ANZAHL BYTES WIRD AM ANFANG AUSGEDRUCKT.

RANGE: REELLE ZAHLEN: 1.70000000E-38 BIS 1.70000000E+38  
9 SIGNIFIKANTE STELLEN

INTEGERS: 48 BIT INTERNE PRAEZISION  
(DEFINITION DURCH %)  
-32767 BIS +32768

STRINGS: 16 BIT INTERNE PRAEZISION  
0 BIS 255 CHARAKTER  
7 BIT ASCII (MIT KLEINEN BUCHST.)

LINIENLAENGE: MAX 48 CHARAKTER

GRAPHICS: 224 X 116 AUFLOESUNG

HARDWARE: 8 BIT X 8 BIT MULTIPLIKATION,  
MINDESTENS 2 TAPE/DISC DRIVES  
VIDEO DISPLAY MIT GRAPHICS  
KEYBOARD MIT KLEINBUCHSTABEN  
MINDESTENS 24 K RAM MEMORY

## 2. BASIC STATEMENTS

ABKUERZUNGEN:	E1      REELLE EXPRESSION N1      INTEGER EXPRESSION (16 BIT) D1      8 BIT INTEGER EXPRESSION L1      16 BIT INTEGER ZAHL S1      STRING EXPRESSION V1      VARIABLE
AUTO L1,D1	GENERIERT AUTOMATISCH LINIENNUMMERN,
AUTO L1	BIS IRGEND EIN ANDERER BEFEHL AUSGE-
AUTO .D1	FUEHRT WIRD. L1 (DEFAULT 10) IST DIE
AUTO	ERSTE GEWUENSCHTE NUMMER, D1 (DEFAULT 10) IST DAS INCREMENT.
CLOSE D1	ABSCHLIESSEN DES FILES MIT DEM SYMBO- LISCHEN NAMEN D1.
CLOSE	ABSCHLIESSEN ALLER OFFENER FILES
CLR	LOESCHT ALLE VARIABLEN SOWIE ALLE OFFEENEN FILES (OHNE SIE ABZUSCHLIESSEN).
CONT	WEITERFAHREN IM PROGRAMM. IST NACH FOLGENDENEREIGNISSEN ERLAUBT. 1. NACHDEM MIT HILFE DER ESC-TASTE GESTOPPT WURDE 2. NACHDEM EIN STOP BEFEHL AUSGEFUEHRT WURDE 3. NACHDEM AUF EIN INPUT BEFEHL MIT RETURN GEANTWORTET WURDE
COPY D1,D2,D3	KOPIERT CHARACTERS VOM DEVICE D1 AUF DEVICE D2. D1 UND D2 KOENNEN I/O DEVICES ODER SYMBOLISCHE FILES SEIN. ABGEBROCHEN WIRD DAS KOPIEREN BEI EINER EOF MARKE, ODER ZUSAETZLICH BEI EINEM CHARAKTER MIT ASCII CODE D3, FALLS D3 GEgeben IST. (BSP: D3=13 (RETURN) KOPIERT BIS ANS ENDE EINER LINIE). DER ABBRUCH-CODE WIRD NOCH VOM INPUT-DEVICE GELESEN, ABER NICHT MEHR KOPIERT.
DATA DATALIST	DEFINIEREN VON DATEN FUER READ STATE- MENTS. DATEN KOENNEN ENTWEDER NUMERISCH ODER STRINGS SEIN, JE NACH DEM TYP DER VARIABLE IM READ-STATEMENT. DIE NUMERI- SCHEN VARIABLEN SIND IM FREIEN FORMAT, GETRENNNT DURCH OBLIGATORISCHE KOMMAS. LEERSTELLEN WERDEN UEBERLESEN. STRINGS KOENNEN, MUessen ABER NICHT IN ANFUEHRUNGSZEICHEN STEHEN. FALLS INNER- HALB EINES STRINGS KOMMAS, KLEINE BUCH- STABEN ODER RESERVIERTE WOERTER VERWEN- DET WERDEN SOLLEN, MUSS DER STRING IM ANFUEHRUNGSZEICHEN STEHEN. DIE EINZELNEN DATA-STATEMENTS WERDEN DER REIHE NACH ABGEARBEITET, FALLS NICHT

DER DATA-POINTER DURCH EIN RESTORE GEÄNDERT WIRD.

DEF FN<sub>U1</sub>(<sub>V2</sub>)=<sub>E1</sub> DEFINIEREN VON FUNKTIONEN VON EINER NUMERISCHEN VARIABLEN. <sub>V1</sub> IST DER NAME DER FUNKTION. <sub>V2</sub> IST EINE LOKALE VARIABLE, DIE AUCH IN <sub>E1</sub> VERWENDET WERDEN KANN UND BEIM AUFRUF DER FUNKTION (SIEHE FUNKTION FN(<sub>E2</sub>)) GESETZT WIRD. DER WERT DER GLOBALEN VARIABLEN <sub>V2</sub> WIRD WEDER GEÄNDERT, NOCH KANN ER IN <sub>E1</sub> BENÜTZT WERDEN. <sub>V2</sub> MUSS EINE NUMERISCHE, REELLE VARIABLE SEIN. INSBESONDRE SIND KEINE FUNKTIONEN VON STRINGVARIABLEN ERLAUBT.

DELETE <sub>S1</sub>.<sub>D1</sub>.<sub>D2</sub> LOESCHT DAS FILE MIT DEM NAMEN <sub>S1</sub> (KEIN DELETE <sub>S1</sub> DEFAULT) UND DEM CYKLUS <sub>D1</sub> (DEFAULT=0) AUF DEM DISC DRIVE <sub>D2</sub> (DEFAULT=0).

DIM <sub>V1</sub>(<sub>D1</sub>,...),... DIMENSIONIERUNG VON ARRAYS. ES SIND BIS ZU 127 DIMENSIONALE ARRAYS VON REELLEN ZAHLEN, INTEGERN (BSP. <sub>A2</sub>(5,3)) UND STRINGS (BSP. <sub>B1\$</sub>(3,9,9)) ERLAUBT. DIE ANZAHL ELEMENTE PRO DIMENSION IST MAX 32000, WOBI DIE INDEXZAHLEN VON 0 AUS LAUFEN.

BENOETIGTER SPEICHERPLATZ:

<sub>D</sub>=ANZAHL DIMENSIONEN. <sub>E</sub>= TOTALE ANZAHL ELEMENTE, <sub>B</sub>=2 FUER INTEGERS, <sub>B</sub>=3 FUER STRINGS UND <sub>B</sub>=5 FUER REELLE ZAHLEN.  
ANZAHL BYTES = 2 + 2 X <sub>D</sub> + <sub>E</sub> X <sub>B</sub>  
STRING ARRAYS BRAUCHEN DAZU NOCH DYNAMISCH EIN BYTE PRO BUCHSTABE JEDES STRINGS.

DIR <sub>D1</sub> DIR AUSDRUCKEN DES DIRECTORY DES DRIVES <sub>D1</sub> (DEFAULT=0). KANN MIT ESCAPE ABGE BROCHEN WERDEN.

DRAW <sub>N1</sub>,<sub>N2</sub>,<sub>D1</sub> DRAW <sub>N1</sub>,<sub>N2</sub> ZEICHNET AUF DEM GRAPHISCHEN DISPLAY EINE GERADE VOM GEGENWAERTIGEN ORT DES GRAPHIC CURSORS ZUM ORT <sub>N1</sub>,<sub>N2</sub>. DER GRAPHIC CURSOR WIRD AUF <sub>N1</sub>,<sub>N2</sub> GESETZT. <sub>D1</sub> GIBT DIE ART DES STRICHES AN (SIEHE BEFEHL PLOT, DEFAULT=0=WEISS)

END BEendet die Ausfuehrung eines BASIC-PROGRAMMS.

FILES ZEIGT DIE FOLGENDEN ANGABEN UEBER ALLE IM MOMENT OFFENEN SEQUENTIELLEN FILES AUF DEM BILDSCHIRM:  
NO SYMBOLISCHER FILE-NAME.  
DRIVE CODE FUER DEN TAPE/DISC DRIVE.  
DIR RICHTUNG (0=LESEN, 1=SCHREIBEN).  
RECORD NO DES ZULETZT GELESENEN ODER GESCHRIEBENEN RECORDS.

FOR V1=E1 TO E2 STEP E3 AUSFUEHRUNG EINER SCHLEIFE.

DIE VARIABLE V1 ERHAELT ZUERST DEN WERT E1, DANN WIRD IN JEDEM LOOP E3 ADDIERT (DEFAULT (KEIN STEP E3) =1), BIS V1 GROESSER ALS E2 IST. FALLS E3<0, WIRD GESTOPPT, FALLS V1>E2. BEIM (ANSICH NICHT ERLAUBTEN) VERLASSEN EINER FOR/NEXT SCHLEIFE AUF ANDEREM WEG ALS UEBER DAS ERFUELLEN DES ABBRUCHKRITERIUMS, IST VORSICHT GEBOTEN. (INDEX VARIABLE DANN SONST NICHT MEHR VERWENDEN).

FORMAT C D1,D2 SETZEN EINES FORMATES FUER NUMERISCHEN OUTPUT (GILT GLOBAL). C IST DER FORMATCODE, D1 DIE FELDLAENGE (MAX. 15) UND D2 DIE ANZAHL STELLEN (MAX. 8, DEFAULT =0). SIEHE INPUT/OUTPUT IM TEIL B DIESES MANUALS.

GET #D1; V1,V2.... LESEN VOM DEVICE D1 (DEFAULT (KEIN #D1;) =0). FALLS V1 EINE STRINGVARIABLE IST, WIRD EIN BUCHSTABE GELESEN, SONST EINE ZIFFER. BEACHTE DIE BEDEUTUNG DER DEVICE-ZAHL:  
 D1=0: LESEN VOM KEYBOARD, WARTEN AUF KEY  
 D1=1: LESEN VOM KEYBOARD, RUECKKEHR MIT LEEREN STRING, FALLS KEIN KEY GE-DRUECKT.  
 D1=2: LESEN VOM TTY-INPUT  
 D1=3: LESEN VOM KEYBOARD UND GLEICHZEITIG SCHREIBEN IM GRAPHIC-DISPLAY.  
 D1>=4: LESEN VON EINEM SYMBOLISCHEN FILE

GOSUB L1 AUFRUF EINER BASIC-SUBROUTINE MIT DER LINIENNUMMER L1.

GOTO L1 FORTSETZUNG DER PROGRAMMAUSFUEHRUNG BEI DER LINIE L1.

IF E1 THEN L1 FORTSETZUNG DER PROGRAMMAUSFUEHRUNG BEI DER LINIE L1, FALLS E1<>0 ODER "WAHR". FALLS DIE BEDINGUNG NICHT ERFUELLT IST, WIRD BEI DER NAECHSTEN LINIE WEITERGEFAHREN.

IF E1 GOTO L1 FUNKTIONIERT GENAU GLEICH WIE IF .. THEN  
 IF E1 THEN STATEMENT... FUEHRT DIE STATEMENTS, DIE NOCH AUF DIESER LINIE LIEGEN, AUS, FALLS DIE BEDINGUNG ERFUELLT IST.

INPUT #D1; V1,V2... LESEN VON EINEM DEVICE (SIEHE GET).  
 INPUT V1,V2... IM GEGENSATZ ZU GET WIRD JEDOCH NICHT

EIN BUCHSTABE, SONDERN EINE GANZE ZAHL ODER EIN STRING GELESEN (GLEICHE SYNTAX WIE IN DATA).  
 FALLS VOM KEYBOARD (D1=0) NUR MIT RETURN GEANTWORTET WIRD, STOPT DAS DAS PROGRAMM UND KANN SPAETER MIT

CONT WEITERGEFUEHRT WERDEN. DAS IST EIN AUSGEZEICHNETES MITTEL, UM BASIC PROGRAMMFEHLER ZU ENTDECKEN.  
INPUT "TEXT":U1,U2 SCHREIBT EINEN TEXT AUF DEN BILDSCHIRM, BEVOR EIN INPUT VERLANGT WIRD.

LET V1=E1 ORDNET DER VARIABLEN V1 DEN WERT E1 ZU.  
V1=E1 DER BEFEHL LET KANN WEGGELASSEN WERDEN.

LIST L1-L2 AUSLISTEN DES PROGRAMMES ODER SPEZIFIZIERTER TEILE DAVON AUF DEM BILDSCHIRM  
LIST L1 ODER AUF DEM MIT D1 ANGEgebenEN DEVICE  
LIST L1- (D1 KANN AUCH EIN SYMBOLISCHES FILE  
LIST -L2 SEIN)  
LIST  
LIST #D1;....

LOAD S1,D1,D2 LADEN DES BASIC-PROGRAMMS MIT DEM FILE-NAMEN S1 UND DEM CYKLUS D1 (DEFAULT=0)  
LOAD S1,D2 VOM DISC/TAPE DRIVE D2. BEIM LESEN VON  
LOAD S1 EINEM TAPE DARF S1 AUCH WEGGELASSEN  
LOAD WERDEN (DEFAULT=ERSTES ENTDECKTES FILE).  
ES MUSS SICH UM EIN BLOCKFILE (TYP B)  
VOM SUBTYP B (BASIC PROGRAMM) HANDELN.

MERGE S1,D1,D2 ANHAENGEN EINES PROGRAMMS VON EINEM  
DISC ODER TAPE-DRIVE AN DASJENIGE, DAS  
SICH SCHON IM MEMORY BEFINDET.

ACHTUNG! MERGE KANN NUR RICHTIG FUNKTIONIEREN,  
WENN DIE FOLGENDEN BEDINGUNGEN ERFUELLT  
SIND: 1. BEIDE PROGRAMME MUESSEN IM  
MEMORY PLATZ HABEN. 2. DAS ZWEITE  
PROGRAMM MUSS HOECHERE LINIENNUMMERN  
ALS DIE HOECHSTE IM ERSTEN VORKOMMENDE  
HABEN.  
FUNKTION VON MERGE SONST WIE LOAD.

MOVE N1,N2 SETZT DEN GRAPHICS CURSOR AUF DIE KOORDINATEN N1,N2.

NEW LOESCHT DAS BASIC-PROGRAMMMEMORY SOWIE  
ALLE VARIABLEN UND OFFENEN FILES.

NEXT V1 GEBRAUCHT EIN EINER FOR - SCHLEIFE.  
NEXT V1,U2,... MEHRERE SCHLEIFEN KOENNEN IN EINEM NEXT  
ENDEN.

NEXT DIE ANGABE DER INDEXVARIABLEN IST FAKULTATIV.

ON D1 GOTO L1,L2,L3,... FUEHRT PROGRAMMAUSFUEHRUNG  
BEI DER D1-TEN LINIENNUMMER LI WEITER.  
FALLS D1=0 ODER GROESSER ALS DIE ANZAHL  
LI, WIRD AUF DER NAECHSTEN LINIE WEITER-  
GEFAHREN.

ON D1 GOSUB L1,L2,L3,... ANALOG

OPEN D1,S1,D2,D3,D4,N1 ERÖEFFEN EINES SEQUENTIELLEN  
FILES. D1 IST DER SYMBOLISCHE FILENAME

(EINE ZAHL ZWISCHEN 4 UND 255), S1 IST DER FILENAME AUF DEM DISC/TAPE DRIVE, D2 (DEFAULT=0) IST DER CYKLUS, D3 DIE DRIVE-NUMMER, D4 IST DER INPUT/OUTPUT CODE (0=LESEN, 1=SCHREIBEN). N1 (DEFAULT=0) BEDEUTET BEI TAPE-FILES DIE STARTPOSITION (NUR BEI SCHREIBFILES BENOETIGT).

OUT D1	ALLER OUTPUT, DER AUF DEN BILDSCHIRM GEHT, WIRD AUCH AUF DAS DEVICE D1 GE-SCHRIEBEN (WIE AUTOPRINT). D1 KANN AUCH EIN SYMBOLISCHES FILE SEIN. D1=0 ODER KEIN D1 STOPT DIESEN AUTOMATISCHEN OUTPUT MODE. DA DIE AUTOPRINT-FUNKTION DER SYSTEMSOFTWARE VERWENDET WIRD, KANN NICHT GLEICHZEITIG AUCH NOCH AUTOPRINT VERLANGT WERDEN.
PLOT N1,N2,D1	SCHREIBT EINEN PUNKT AUF DEM GRAPHIC-DISPLAY BEI DEN KOORDINATEN N1,N2 UND SETZT DEN GRAPHIC-CURSOR. D1 (DEFAULT=0) BEDEUTET DIE ART DES STRICHES: D1=0 WEISS UEBERMALEND D1=1 SCHWARZ/WEISS INVERTIEREND D1=2 SCHWARZ UEBERMALEND
PLOT NEW	INITIALISIERT GRAPHICS. DIESER BEFEHL MUSS GEgeben WERDEN, BEVOR ANDERE GRAPHICS-BEFEHLE AUSGEFUEHRT WERDEN DUERFEN. BEACHTE, DASS DANN IM ALPHANUMERISCHEN VIDEO-MEMORY NUR NOCH 16 LINIEN FREI SIND. PLOT NEW FUEHRT AUCH NOCH AUTOMATISCH EIN PLOT CLR UND PLOT CONT AUS.
PLOT CLR	LOESCHT DEN GRAPHISCHEN DISPLAY
PLOT WAIT	INITIALISIERT ANIMATION GRAPHICS. DER PLOT WIRD VON NUN AN IN EINEN BUFFERGESCHRIEBEN (\$0C00-\$13FF), VON DEM AUS ER SCHNELL AUF DEN BILDSCHIRM KOPIERT WERDEN KANN. ALLE PLOTBEFEHLE WIRKEN NUN AUF DEN BUFFER. "PLOT NEW" LOESCHT DIESE PLOTART.
PLOT NEXT	KOPIERT DEN ANIMATION BUFFER AUF DEN BILDSCHIRM.
PLOT STOP	RUECKKEHR IN DEN ALPHANUMERISCHEN DISPLAY, OHNE DEN INHALT DES GRAPHISCHEN DISPLAYS ZU VERLIEREN. (TOGGLE MIT KEY A-G MOEGLICH).
PLOT CONT	RUECKKEHR IN DEN GRAPHISCHEN DISPLAY.

PLOT END LOESCHT INITIALISATION DES GRAPHISCHEN DISPLAYS. INHALT IST VERLOREN. DAS VIDEO MEMORY IST WIEDER VOLL FUER DIE ALPHANUMERISCHE ANZEIGE VERFUEGBAR.

BEACHTE, DASS BEFEHLE WIE PLOT STOP UND PLOT CONT NUR DIE MOMENTALE DISPLAY-ART AENDERN. UNABHAENGIG DAVON KANN MIT DEN ENTSPRECHENDEN BEFEHLEN INS ALPHA-NUMERISCHE UND GRAPHISCHE MEMORY GESENDSCHRIEBEN WERDEN. BEIDE SIND COMPUTER-INTERN VONEINANDER UNABHAENGIG, AUSSER DASS ERSTERES NACH DER INITIALISIERUNG DER GRAPHISCHEN MEMORY NUR NOCH 16 ZEILEN LANG IST. NEBEN DEN BASIC BEFEHLEN PLOT STOP UND PLOT CONT HAT DER BENUETZER AUCH UEBER DIE KEYS A-G ZUGRIFF ZUR DISPLAY-ART UND KANN BEIDE MEMORIES MIT DEN KEYS CSC UND CLG INDIVIDUELL LOESCHEN.

POKE N1,D1 SETZT DIE MEMORY-ZELLE MIT ADRESSE N1 AUF DEN WERT D1.

PRINT #D1; PRINT LIST SCHREIBEN AUF DAS DEVICE D1  
 PRINT PRINT LIST SCHREIBEN AUF DAS DEVICE 0  
 PRINT #3,D2; PRINT LIST AUF GRAPHICS KANN DIE STRICHART (D2, DEFAULT=0) ANGEgeben WERDEN (SIEHE PLOT)

OUTPUT DEVICES SIND:

D1=0	BILDSCHIRM
D1=1	SCHREIBMASchine
D1=2	HEATHKIT PRINTER
D1=3	GRAPHIC DISPLAY (7X8 MATRIX, MIT GRAPHICS POINTER UNTEN LINKS)
D1>3	SYMBOLISCHES FILE

PRINT LIST:

BESTEHT AUS EINZELNEN PRINT-ITEMS, DIE DURCH DIE FOLGENDEN ZEICHEN GETRENNNT WERDEN:

- , TABULIERT AUF NAECHSTE 12-STELLE
- ; TABULIERT NICHT

ES KANN AUCH KEIN TRENnzeichen VERWENDET WERDEN, FALLS DIE SYNTAX DES 1. ITEMS KLAR DESSEN ENDE MARKIERT.

PRINT ITEMS:

TAB(D1)	TABULIERT AUF POSITION D1
SPC(D1)	SCHREIBT D1 LEERZEICHEN
EC D1,D2)	FORMAT SPEZIFIKATION FUNKTIONIERT WIE FORMAT(GLOBAL!)
E1	DRUCKT DEN WERT VON E1 AUS
S1	DRUCKT DEN STRING S1 AUS
"TEXT"	DRUCKT DEN TEXT AUS AM ENDE EINES PRINT BEFEHLS ERFOLGT EIN AUTOMATISCHES CR LF, FALLS NICHT MIT , ODER ; GEENDET WIRD.

READ V1,V2,... LESEN VON DATEN AUS DATA-STATEMENTS

REM TEXT REMARK TEXT	ANMERKUNG, GANZE LINIE WIRD NICHT BEACHTET.
REN N1,N2	NEUNUMMERIERUNG DER LINIEN. NEUE WERTE VON N1 AUS MIT INTERVALL N2.
RESTORE L1 RESTORE	SETZT DATA POINTER AUF BEGINN DER LINIE L1 (DEFAULT = 0 = START DES PROGRAMMS)
RETURN	RUECKKEHR AUS EINER SUBROUTINE
RUN	STARTEN DES BASIC PROGRAMMS. ALLE VARIABLEN WERDEN AUF "UNDEFINIERT" GESETZT.
STOP	STOPT PROGRAMMAUSFUEHRUNG MIT ANGABE DER LINIENNUMMER (Z.B. BREAK AT LINE 100)
STORE S1,D1,D2,N1	ABSPEICHERN EINES BASIC-PROGRAMMS.
STORE S1,D1,D2	S1 IST DER NAME DES ERZEUGTEN FILES,
STORE S1,,N2	D1 (DEFAULT=0) DER ZYKLUS, D2 (DEFAULT=0) TAPE ODER DISK-DRIVE, N1 IST EINE
STORE S1	FAKULTATIVE BANDPOSITION (DEFAULT=0).
SYS N1	AUSFUEHREN EINER MACHINENSPACHSUBROUTINE BEI DER ADRESSE N1
WAIT N1	WARTET N1 MILISEKUNDEN.

## 2. FUNKTIONEN

---

FOLGENDE FUNKTIONEN SIND IM R65 12K GRAPHIC BASIC ENTHALTEN:

ABS(E1)	ABSOLUTWERT DES AUSDRUCKS E1
N1 AND N2	LOGISCHES AND DER BEIDEN INTEGERAUSDRECKE N1 UND N2.
ASC(S1)	ASCII-CODE DES BUCHSTABENS S1
ATN(E1)	ARCUS TANGENS VON E1
CHR\$(D1)	CHAR MIT ASCII-CODE D1 (UMKEHRFUNKTION VON ASC(S1))
COS(E1)	COSINUS VON E1
EXP(E1)	EXPONENTIALFUNKTION
FNU1(E1)	BERECHNUNG DER MIT HILFE VON DEF FN1 DEFINIERTEN FUNKTION.
INT(E1)	GANZER ANTEIL VON E1
LEFT\$(S1,D1)	D1 ERSTE BUCHSTABEN DES STRINGS S1

LEN(S1)	LAENGE DES STRINGS D1
LOG(E1)	NATUERLICHER LOGARITHMUS VON E1
MID\$(S1,D1)	RECHTER TEIL DES STRINGS S1 AB BUCHSTABE NUMMER D1 BIS ANS ENDE VON S1
MID\$(S1,D1,D2)	AUSSCHNITT AUS DEM STRING S1 VON D1 BIS D2
NOT N1	LOGISCHES NOT DES INTEGERS N1
N1 OR N2	LOGISCHES OR DER INTEGER N1 UND N2
PEEK(N1)	WERT DER MEMORY-ZELLE N1
POS(D1)	POSITION DES CURSORS IN DEVICE D1 (NICHT ERLAUBT FUER SYMBOLISCHE FILES)
RIGHT\$(S1,D1)	DIE D1 LETZTEN BUCHSTABEN DES STRINGS S1
RND(E1)	RANDOM-ZAHL: E1=0: LETZTE RANDOM ZAHL E1<>0: ECHTE NEUE RANDOM ZAHL
SGN(E1)	0, FALLS E1=0 1, FALLS E1>0 -1, FALLS E1<0
SIN(E1)	SINUS VON E1
SQR(E1)	QUADRATWURZEL VON E1
STA(D1)	INTERPRETER STATUS INFORMATION: STA(0) ANZAHL FREIE BYTES IM MEMORY STA(1) LAENGE DES PROGRAMMS (BYTES) STA(2) VIDEO CONTROLER STATUS STA(2) AND 128 1=INVERSE VIDEO ON STA(2) AND 64 1=GRAPHICS ANGEZEIGT STA(2) AND 32 1=GRAPHICS INITIALISIERT STA(2) AND 16 1=DISPLAY CONTROL KEY STA(2) AND 8 1=AUTO OUTPUT GESETZT STA(2) AND 2 1=IBM PRINTER GESTOPT STA(3) TESTET DOT IN GRAPHICS MEMORY, DER VOM MOMENTANEN GRAPHICS CURSOR ADDRESIERT WIRD (1=WEISS,0=SCHWARZ) STA(4-255) INFORMATION UEBER SYMBOLISCHES FILE 0 = NICHT OFFEN 1 = OFFENES READ FILE 2 = OFFENES WRITE FILE
STR\$(E1)	STRING DES WERTES VON E1
TAN(E1)	TANGENS VON E1
USR(N1,E1)	MACHINENSPRACHFUNKTION AN DER ADRESSE N1. IM FLP-ACCUMULATOR WIRD DER WERT VON E1 UEBERGEBOEN, DIE FUNKTION ERHAELT

DANN DEN BERECHNETEN WERT VOM FLP-ACCU-MULATOR.

VAL(S1) NUMERISCHER WERT DES STRINGS S1 (UMKEHR-FUNKTION VON STR\$(E1))

## TEIL B SPEZIELLE MOEGLICHKEITEN

---

### 1. NUMERISCHER INPUT

---

DER NUMERISCHE INPUT (STATEMENTS INPUT, DATA SOWIE IM PROGRAMM IN EXPRESSIONS) ERFOLGT IM FREIEN FORMAT, D.H. FIXKOMMA-, EXPONENTIAL- UND HEX-DARSTELLUNG KOENNEN BELIEBIG VERWENDET WERDEN.

BEISPIELE FUER ERLAUBTEN NUMERISCHEN INPUT:

1.00, 1+7.48, -2.3385555554444444, 127, 1E-3, 1E3,  
127.28E+22, +\$FF, -\$12C, #1

BEIM HEXADEZIMALEN INPUT (DEFINIERT DURCH DAS \$-ZEICHEN AM ANFANG) IST DER MAXIMALE WERT \$7FFFFFF, HEX-ZAHLEN ENTAHLTEN KEINE BRUCHTEILE.

### 2. NUMERISCHER OUTPUT

---

DER NUMERISCHE OUTPUT KANN FORMATTIERT WERDEN MIT DEM BEFEHL FORMAT (SIEHE TEIL A DES MANUALS) ODER MIT DEM PRINT-ITEM [(D1,D2)]. JEDO FORMATTIERUNG GILT, BIS SIE DURCH EINE ANDERE FORMATTIERUNG AUFGEHOBEN WIRD. DIE BASIC-DEFAULT FORMATTIERUNG A0 WIRD AUCH DURCH CLR UND NEW GESETZT.

FELDLAENGE: IN ALLEN FORMATTIERUNGSAARTEN IST D1 DIE FELDLAENGE. ALLE ZAHLEN WERDEN RECHTSJUSTIERT IN DAS SPEZIFIZIERTE FELDGESCHRIEBEN. FALLS DAS FELD ZU KLEIN IST, WIRD UEBER DAS FELD HINAUSGESCHRIEBEN. D.H. BEI EINEM KLEINEN FELD ERFOLGT DER OUTPUT WIE OHNE FELD-SPEZIFIZIERUNG. FALLS DIE FELDLAENGE D1=0 IST, WIRD DIE ZAHL MIT EINEM LEERZEICHEN ABGESCHLOSSEN. FALLS D1>0 IST, WIRD KEIN LEERZEICHEN AM ENDEGESCHRIEBEN. DIE MAXIMALE FELDLAENGE IST 15. POSITIVE ZAHLEN BEGINNEN MIT EINEM LEERZEICHEN (AUSSER IM H-FORMAT), NEGATIVE MIT EINEM MINUSZEICHEN.

AUTOMATISCHE FORMATTIERUNG: A D1

DIE ZAHL WIRD NORMALERWEISE OHNE EXPONENT GESENDRIBEN, MIT SOVIELEN STELLEN, WIE NOTIG. ZAHLEN GROESSER ALS 99999999 ODER KLEINER ALS 0.01 WERDEN MIT EXPONENT GESENDRIBEN.

A 0 (DEFAULT) IST DIE STANDART-BASIC FORMATTIERUNG.

FIXKOMMA-FORMATTIERUNG: F D1,D2

ES WIRD IM FIXKOMMA-FORMAT MIT D2 STELLEN NACH DEM KOMMA GESENDRIBEN (RUNDUNG). FALLS DIE ANZAHL STELLEN WEGEN DER GROESSE DER ZAHL GROESSER ALS 9 IST, WERDEN

STELLEN NACH DEM KOMMA WEGGELASSEN. ZAHLEN GROESSER ALS 99999999 WERDEN IM E-FORMAT GE SCHRIEBEN. IM GEGEN-SATZ ZUM A-FORMAT ERSCHEINEN SEHR KLEINE ZAHLEN ALS 0.

EXPONENTIAL-FORMATTIERUNG: E D1,D2  
ES WIRD MIT EXPONENT UND D2 STELLEN NACH DEM KOMMA GE SCHRIEBEN (RUNDUNG).

HEX-FORMATTIERUNG: H D1,D2  
ES WIRD ZUERST EIN \$-ZEICHEN, NACHHER DIE ZAHL IN HEX GE SCHRIEBEN. MAX WERTE +\$7FFFFFF, -\$7FFFFFF.  
NEGATIVE ZAHLEN WERDEN IM KOMPLEMENT GE SCHRIEBEN (8-HEX-DIGIT-2-KOMPLEMENT). D2 GIBT DIE MINIMAL ZU SCHREIBENDEN STELLEN AN. WENN NOTIG, WERDEN NULLEN AM BEGINN DER ZAHL GE SCHRIEBEN.

BEISPIELE: (# # MARKIEREN DAS FELD IN DEN BEISPIELEN)

WERT:	FORMAT:	OUTPUT:
1	F6,2	# 1.00#
1	F0,2	#1.00 #
1.5	F5,0	# 1#
127	E6,1	#1.3E+02#
128	H8,4	# \$0080#
31	H8,0	# \$1F#

D1 UND D2 SIND EXPRESSIONS, D.H. ES KANN AUCH DYNAMISCH INNERHALB EINES PROGRAMMS DIE FORMATTIERUNG BERECHNET WERDEN.

### 3. INPUT/OUTPUT

---

DAS R65 GRAPHIC BASIC ENTHAELT EINE GANZE REIHE VON MOEGLICHKEITEN, UM MIT DEN INTERFACES DES R65 COMPUTER SYSTEMS ZU VERKEHREN.

MIT "PRINT", "LIST" UND "OUT" KOENNEN DIE ALPHANUMERISCHE VIDEO-ANZEIGE, DIE IBM-SCHREIBMASCHINE, EIN RS 232 INTERFACE FUER EINEN SCHNELLEN DRUCKER ETC. DIE GRAPHISCHE VIDEO-ANZEIGE SOWIE SEQUENTIELLE FILES AUF TAPE UND DISK ANGESTEUERT WERDEN.

MIT "GET" UND "INPUT" KANN VOM KEYBOARD, EINEM TTY-EINGANG, SOWIE VOM SEQUENTIELLEN FILES AUF TAPE UND DISK GELESEN WERDEN.

BASIC PROGRAMME KOENNEN MIT "LOAD", "STORE" UND "MERGE" GE SPEICHERT UND GELADEN WERDEN.

DAS ABLESEN DER SENSE SWITCHES GE SCHIEBT MIT "PEEK":  
PEEK(\$1700) AND 32 = 1 IF SENSE SWITCH 1 SET  
PEEK(\$1700) AND 16 = 1 IF SENSE SWITCH 2 SET  
PEEK(\$1700) AND 8 = 1 IF SENSE SWITCH 3 SET

#### 4. GRAPHICS

EINE DER WICHTIGSTEN EIGENSCHAFTEN IST DIE MOEGLICHKEIT, GRAPHISCHE DARSTELLUNGEN AUF DEM BILDSCHIRM ZU ERZEUEN. DURCH "PLOT NEW" WIRD DAS VIDEO MEMORY SO RUPFGETEILT, DASS ALPHANUMERISCHE UND GRAPHISCHE DARSTELLUNG UNBAHAENGIG VONEINANDER MOEGLICH SIND. DIE GRAPHISCHE DARSTELLUNG GESCHIEHT IN EINEM KOORDINATENSYSTEM VON 224\*118 DOTS. DABEI GEHT DIE X-ACHSE VON 0 BIS 223 UND DIE Y-ACHSE VON 0 BIS 117. PUNKTE, DIE AUSSERHALB DES FELDES FALLEN, WERDEN AUF DEM RAND ANGEZEIGT.

DAS ZEICHNEN GESCHIEHT MIT HILFE EINES GRAPHIC-CURSORS, DER VON DEN VERSCHIEDENEN BEFEHLEN GESETZT WIRD.

ZEICHNEN VON PUNKTEN: PLOT X,Y

ZEICHNEN VON GERADEN: MOVE X1,Y1: DRAW X2,Y2

WEITERHIN IST MIT PRINT #3 AUCH DAS BESCHRIFTEN MOEGLICH (NUR 64-ZEICHEN ASCII CODE).

ZEICHNEN UND BESCHRIFTEN KANN IN WEISS, SCHWARZ ODER INVERTIEREND ERFOLGEN.

MIT DER FUNKTION STA(3) LAESST SICH AUCH BESTIMMEN, OB EIN GEGEBENER PUNKT IM MOMENT SCHWARZ ODER WEISS IST.

DIE BEFEHLE "PLOT WAIT" UND "PLOT NEXT" ERLAUBEN DIE ARBEIT MIT EINEM VIDEO BUFFER ZUR ANIMATION.

#### 5. ARBEITEN MIT SEQUENTIELLEN FILES

ES BESTEHEN AUSGEDEHNT MOEGLICHKEITEN, UM MIT SEQUENTIELLEN ASCII FILES AUF TAPE UND DISK ZU ARBEITEN. SOLCHE FILES WERDEN MIT "OPEN" ERoeffnet und mit "CLOSE" abgeschlossen. Beim eroeffnen muss die Richtung (LESE- oder SCHREIBFILE) bestimmt werden.

Das SCHREIBEN auf SCHREIBFILES kann mit "PRINT", "LIST" und "OUT" erfolgen.

Das LESEN geschieht mit "GET" oder "INPUT".

SPEICHERN von DATEN mit SEQUENTIELLEN FILES:

```
BSP: 10 OPEN 5,"DATENFILE",01,2,1,100  
      20 PRINT #5; I","J","K  
      30 CLOSE 5
```

BEACHTE, DASS DIE INDIVIDuellen ZAHLEN EXPLIZIT MIT KOMMAS GETRENNT WERDEN MUessen, WENN SIE AUF EINER LINIE ZU STEHEN KOMMEN. DATEN AUF EINER LINIE, DIE NICHT DURCH KOMMAS GETRENNT SIND, KOENNEN NICHT MIT "INPUT" GELESEN WERDEN, EBENSO, WENN DIE GE SCHRIEBENEN LINIEN LAENGER ALS 48 ZEICHEN SIND.

LESEN von DATEN von SEQUENTIELLEN FILES:

```
BSP: 10 OPEN 6,"DATENFILE",01,2  
      20 INPUT #6; I,J,K  
      30 CLOSE 5
```

TEXTE WERDEN AM EINFACHSTEN MIT "GET" EINGELESEN.

DAS AUTOMATISCHE KOPIEREN IST MIT COPY MOEGLICH.

BSP: KOPIEREN EINER LINIE VOM FILE 5 AUF DEN BILDSCHIRM  
COPY 5.0,13

FALLS IRGENDWELCHE SCHWIERIGKEITEN BEIM LESEN VON SEQUENTIELLEN DATENFILES AUFTREten, KANN MIT DEM BASIC-PROGRAMM "SHOW DATA" DAS FILE IM DETAIL (MIT ALLEN KONTROLLZEICHEN ETC.) BETRACHTET WERDEN.

DAS EINE EINES FILES WIRD BEIM LESEN MIT EINER EOF-MARKE (\$1F) MARKIERT. DIESE SOLLTE MIT "GET" UEBERPRUEFT WERDEN.

ALLE SEQUENTIELLEN BASIC-OUTPUT-FILES HABEN DEN SUBTYP D.