Arbeitsmaterial Informatik

Kleincomputer
KC 85/1, KC 87

Schülerrechenzentrum
Station Junger Techniker
„Johann Andreas Schubert"
Karl-Marx-Stadt

Karl-Marx-Stadt, 4.Maerz 1988

Schuelerrechenzentrum an der
Station Junger Techniker
"Johann Andreas Schubert"
Karl-Marx-Stadt, 9044
Str. Usti nad Labem 43/45

Inhalt:

Technische Daten des KC 85/1 1
Struktur und Blockschaltbild eines Rechners .. 2
Informatik - Begriffe 3
Der Weg zum Programm 4
KC - **BASIC** 5
Zusaetzliche Befehle des KC 85/3 7
BASIC - Fehlermeldungen 8
Wichtige Adressen des BASIC - Interpreters ... 8
BASIC - Tokens und Zeilenaufbau 9
Befehlsliste des KC - **PASCAL** 10
Grafikzeichen 12
Grafiktasten und Bildschirmaufbau 13
ASCII - Zeichen - Liste 14
Aufbau des Betriebssystems 15
Speicheraufteilung 15
Kanaladressen 15
Arbeitszellen des Betriebssystems 16
Unterprogramme des BOS - Systems 17
Programmierung von CTC und PIO 18
Registersatz des U 880 19
Befehlsliste des U 880 19
Flagregister des U 880 21
Editor-Assembler 'ASM' 22
Zusatzmonitor 'ZM' 23
Allgemeine Benutzungshinweise 24
Umrechnung Hexadezimal-Dezimal 25
Quellen 25

Arbeitsmaterial Informatik :

Beschreibungen und Tabellen zum

Kleincomputer KC 85/1, KC 87

Einleitung:

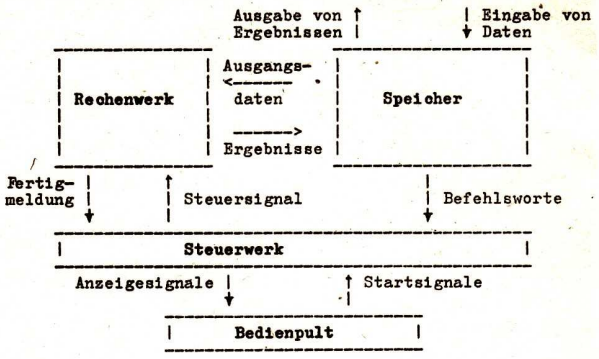
Der Kleincomputer KC 85/1 ist ein hochwertiges mikroelektronisches Erzeugnis der Industrie unseres Landes. Seine guten Gebrauchseigenschaften ermoeglichen den Einsatz in Industrie, Wissenschaft und Lehre.

Technische Daten:

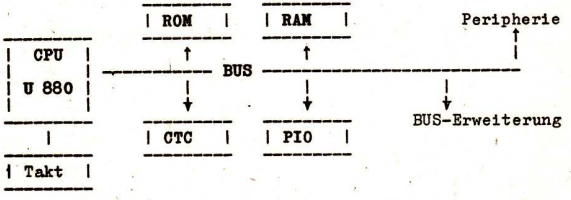
- 8 bit-Mikrorechner mit Mikroprozessor **U 880 D**
- Speicherkapazitaet:
  - 4 Kbyte-ROM Betriebssystem
  - 16 Kbyte-RAM Arbeitsspeicher
  - 2 Kbyte-RAM Bild/Farbspeicher
  - 2 Kbyte-ROM Zeichengenerator

  erweiterbar auf maximal 64 Kbyte
- Bildaufbau
  - 24 Zeilen x 40 Spalten
  - 20 Zeilen x 40 Spalten (waehlbar)
  - 1 Zeichen = 8 x 8 Bildpunkte
- je 8 Vorder- und Hintergrundfarben
- 96 Textzeichen und 128 Grafiksymbole, Pseudografik
- Kassettenmagnetbandgeraet als externer Massenspeicher
- Anschluss der Drucker der Serie K 63
- Programmiersprachen
  - KC-BASIC
  - Assembler (SYPS K 1520)
  - KC-PASCAL (Standard-PASCAL)
  - Mini-BASIC
- Anschluesse
  - 1 Buchse fuer Fernsehgeraet
  - 1 Buchse fuer Magnetbandgeraet
  - 2 Buchsen fuer Spielhebel
  - 4 Steckplaetze fuer Zusatzmodule
  - 1 Buchse fuer spezielle Anwendungen (PIO Port B)
- Masse: 3,9 kg
- Stromversorgung: 220 V $\pm$ 10 %, 30 W; 50 Hz $\pm$ 1 Hz
- Betriebstemperatur: + 5°C bis + 35°C
- Hersteller: **VEB Robotron-Messelektronik "Otto Schoen" Dresden**

Struktur eines Rechners :



Blockschaltbild :



Begriffserklaerung:

BUS ... Sammelleitung zur Informationsuebertragung
CPU ... Central Processing Unit,
Zentrale Verarbeitungseinheit
CTC ... Counter/Timer Circuit
Zaehler/Zeitgeber - Schaltkreis
PIO ... Parallel Input/Output
Eingabe/Ausgabe - Schaltkreis
RAM ... Random Access Memory, Lese-Schreib-Speicher
ROM ... Read Only Memory, Nur-Lese-Speicher

Informatik - Begriffe :

Adresse	... Kennzeichnung eines Speicherplatzes
Akkumulator	... Haupt-Rechenregister
alphanumerisch	.. Buchstaben, Ziffern und Sonderzeichen
ALU	... arithmetisch-logische Einheit
ASCII	... 7 bit-Code fuer Zeichendarstellung
Assembler	... maschinennahe Programmiersprache
BASIC	... 'Beginners All Purpose Symbolic Instruction Code', am weitesten verbreitete hoehere Programmiersprache
BCD	... Abk. fuer binaer codierte Dezimalzahl
Befehl	... Fuer die Ausfuehrung einer Operation im Rechner notwendige Angabe
Betriebssystem	.. Paket von Grundprogrammen, welche den Computer funktionsfaehig machen
Bit	... kleinste Speichereinheit
Byte	... 1 Byte = 8 bit, kleinste adressierbare Einheit, entspricht 1 Ziffer/Buchstabe
Compiler	... Uebersetzungsprogramm einer hoeheren Programmiersprache in Maschinencode
Cursor	... Markierung auf dem Bildschirm an der Stelle, wo das naechste Zeichen dargestellt wird
Daten	... ausgewaehlte Informationen zu einem bestimmten Sachverhalt
Datei	... Datenmenge
Diskette	... Folienspeicher
File	... Datei, Programm
Hardware	... geraetetechnische Ausstattung eines Computers
Interpreter	... Uebersetzungsprogramm einer hoeheren Programmiersprache, wobei erst waehrend des Programmlaufs uebersetzt wird
Interrupt	... Programmunterbrechung
IRM	... Bildwiederholspeicher
Kbyte	... Angabe der Speichergroesse 1 Kbyte = 1024 byte
Maschinensprache	Informationsform, welche der Computer entschluesseln kann
Monitor	... Bildschirm bzw. Dienstprogramm
PASCAL	... 1970 entwickelte hoehere Programmiersprache
Port	... Kanaladresse, Tor
Programm	... Algorithmus zur Loesung eines Sachverhaltes mit dem Computer
Register	... Speicherstellen des Mikroprozessors
Software	... alle Programme und Beschreibungen, die die Benutzung des Computers ermoeglichen
Stack	... Keller- oder Stapelspeicher
String	... Zeichenkette
V.24 Interface	.. weitverbreitete Schnittstelle zum Anschluss peripherer Geraete

Fuer das erfolgreiche Erstellen eines Programms ist es notwendig, die folgende Schrittfolge einzuhalten:

1. **Analyse der Aufgabenstellung**, d.h. Vertrautmachen mit den bekannten und unbekannten Groessen; Aufstellen eines Algorithmus zur Loesung der Aufgabenstellung.
2. **Grobplanung**, d.h. Zusammenstellen der beabsichtigten Moeglichkeiten aber auch Einschraenkungen des Programms. Hier ist es auch unbedingt notwendig, **alle** benoetigten Variablen genau zu definieren.
3. **Erstellen des 1.Struktogramms**
   In diesem Struktogramm werden die wesentliche Abfolge des Programms, die einzelnen Unterprogramme und Loesungsalgorithmen aufgenommen.
4. **Schrittweises Verfeinern des Struktogramms**
   Komplizierte und aufwendige Teile des Programms werden immer praeziser in weiteren Struktogrammen beschrieben. Ein grosser Aufwand an dieser Stelle hilft spaeter, wenn sich inhaltliche Fehler einstellen sollten.
5. **Aufstellen des Programms**
   Nun ist es moeglich, die Struktogramme in die entsprechende Programmiersprache, z.B. BASIC, zu uebertragen. Dieses erste konkrete Programm **muss** unter allen Umstaenden auf dem Papier und **nicht** gleich in den Kleincomputer geschrieben werden.
6. **Eingeben des Programms** in den Kleincomputer
7. **Erster Test** einzelner Teile und des ganzen Programms
   An dieser Stelle sollte vor allem die inhaltliche Korrektheit der Loesung getestet werden. Treten syntaktische Fehler auf, so sind diese schrittweise zu entfernen. Bei anderen, etwa inhaltlichen Fehlern, ist es oft ratsam, das Struktogramm zu kontrollieren und gegebenenfalls noch einmal mit 1. usw. zu beginnen. So unschoen das ist, spart man doch meist Zeit.
8. **Ausgestalten des Programms**
   Sind alle inhaltlichen und syntaktischen Fehler entfernt, kann man das Programm dahingehend ausgestalten, dasz Grafikelemente genutzt werden, der Bildschirm besser aufgeteilt wird usw.... Desweiteren sollte man moegliche Fehleingaben durch eine sinnvolle Programmierung ausschliessen. Ein Programm ist nur dann gut, wenn es sich **selbst** umfangreich dokumentiert.
9. **Letzter Test**
   Bei dem letzten Test kommt es nochmals darauf an, alle Funktionen, Unterprogramme usw. mit verschiedensten Werten zu ueberpruefen. Treten keine Probleme auf, so kann das Programm in der letzten Form endgueltig auf Kassette erfasst werden.
10. Zum Abschluss ist es notwendig eine gute **Dokumentation** fuer den Nutzer zu erstellen. Ziele, Aufgaben, Ergebnisse aber auch Grenzen des Programms sind darzustellen.

Start des BASIC-Interpreters: **BASIC** <ENTER>
Restart ohne Loeschen des Programms: **WBASIC** <ENTER>

KC - BASIC - Anweisungen:

Hinweis: <> ... notwendige Parameter
[] ... moegliche Parameter
() ... syntaktisch notwendige Klammern

```
AUTO [z] [,s]          ... Automatische Zeilennummerrierung ab
                           Zeile z im Abstand s
BEEP                   ... Erzeugung eines Tones
BORDER <rfarb>         ... Festlegen der Bildschirmrandfarbe
BYE                    ... Rueckkehr zum Betriebssystem
CALL <adr>             ... Aufruf eines Maschinenprogramms ab
                           der Adresse <adr> (dezimal !)
CALL* <adr>            ... s.o., Adresse hexadezimal
CLEAR [s][,e]          ... Loeschen aller Variablen und Ein-
                           stellen der Groesse des Zeichen-
                           kettenspeichers [s] sowie der
                           verfuegbaren Endadresse [e]
CLOAD "name"           ... Einlesen eines BASIC-Programms
CLOAD*"name";feldname  ... Einlesen eines Variablenfeldes
CLS                    ... Bildschirm loeschen
CONT                   ... Programmfortsetzung nach <STOP>,
                           <PAUSE>
CSAVE "name"           ... Abspeichern eines BASIC-Programms
CSAVE*"name";feldname  ... Abspeichern eines Feldes
DATA <k1> [,k2]..[,kn] ... Konstantenliste
DEEK (<adr>)           ... Lesen der Adressen <adr> und
                           <adr+1>
DEF FNY(x)= ausdruck   ... Definition einer Funktion
DELETE <a> [,e]        ... Loeschen der Zeilen <a> bis [e]
DIM feld (index [,index...])
                       ... Festlegen eines Variablenfeldes
DOKE <adr>,<w>         ... Schreiben des Wertes <w> auf die
                           Adressen <adr> und <adr+1>
EDIT <znr>             ... Editieren der Zeile <znr>
END                    ... Programmende
FOR var=<a> TO <e> [STEP <s>]
                       ... Programmschleife, Variable var
                           laeuft von <a> bis <e> in
                           Schritten von <s>
GOSUB <znr>            ... Sprung zum Unterprogramm ab Zeile
                           <znr>
GOTO <znr>             ... Unbedingter Sprung zur Zeile <znr>
IF ausdruck THEN anweisung [:ELSE anweisung]
                       ... Bedingte Handlungsanweisung
INK <vfarb>            ... Festlegen der Vordergrundfarbe
INP <port>             ... Lesen der Kanaladresse <port>
INPUT ["string";] var... Eingabe einer Zahl/Zeichenkette
JOYST (n)              ... Spielhebelabfrage (n = 1 oder 2)
[LET] var=ausdruck     ... Wertzuweisung
```

```
LINES [anz]              ... Festlegen der Anzahl der aufzu-
                             listenden Zeilen
LIST [znr]               ... Listen des Programms ab Zeile [znr]
LIST#1 "name"            ... Auslesen des Programms in ASCII-
                             Zeichen
LOAD#1 "name"            ... Einlesen eines mit LIST# ausgelese-
                             nen Programms
NEW                      ... Loeschen des BASIC-Programms
NULL <zahl>              ... Anzahl der auszugebenden Dummy-
                             Zeichen
ON x GOTO liste von zeilennummern
                         ... mehrfache Programmverzweigung
ON x GOSUB liste von zeilennummern
                         ... mehrfache Programmverzweigung
OUT <port>,<w>           ... Ausgabe des Wertes <w> auf die
                             Kanaladresse <port>
PAPER <hfarb>            ... Festlegen der Hintergrundfarbe
PAUSE [n]                ... Programmunterbrechung (n*0,1 sek)
PEEK (<adr>)             ... Lesen der Adresse <adr>
POKE <adr>,<w>           ... Schreiben des Wertes <w> auf die
                             Adresse <adr>
PRINT [liste]            ... Ausgabe von Text/Werten
?                        ... entspricht PRINT
PRINT AT (z,s);[liste] ... Ausgabe ab Zeile z und Spalte s
READ var [,var...]       ... Lesen von Variablen aus DATA-Zeilen
REM kommentar            ... Kommentarzeile
!                        ... entspricht REM
RENUMBER [aza] [,aze] [,nza] [,ab]
                         ... Umnummerieren von alter Zeile aza
                             bis aze mit erster neuer Zeile nza
                             im Abstand ab
RESTORE [znr]            ... Setzen des DATA-Zeigers
RETURN                   ... Ruecksprung aus Unterprogramm
RUN [znr]                ... Programmstart
STOP                     ... Unterbrechen eines Programms
TROFF                    ... Ausschalten des Kontroll-Modus
TRON                     ... Einschalten des Kontroll-Modus
WAIT i,j [,k]            ... spezielle Funktion zur Programm-
                             steuerung
WIDTH [zahl]             ... Laenge einer Ausgabezeile
WINDOW [za,ze,sa,se] ... Fenstereinstellung von Zeile za bis
                             ze und Spalte sa bis se
```

Funktionen :

```
ABS (x)                  ... absoluter Betrag
ATN (x)                  ... arctan x, Resultat im Bogenmasz
COS (x)                  ... Cosinus, cos x
EXP (x)                  ... Exponentialfunktion  e^x
INT (x)                  ... Ganzer Teil von x
LN  (x)                  ... natuerlicher Logarithmus von x
SGN (x)                  ... Vorzeichenfunktion
SIN (x)                  ... Sinus, sin x
SQR (x)                  ... Quadratwurzel von x
TAN (x)                  ... Tangens, tan x
```

String - Funktionen :

INSTR (A$,B$)	... ermittelt die Position, ab welcher A$ in B$ enthalten ist
LEFT$ (A$,x)	... liefert die ersten x Zeichen von A$
LEN(X$)	... Laenge des Strings X$
MID$ (A$,x,y)	... y Zeichen von A$, beginnend mit dem x-ten Zeichen
RIGHT$ (A$,x)	... letzte x Zeichen von A$
STRING$ (n,A$)	... Zeichenkette mit n mal String A$
STR$ (x)	... Umformen von x in einen String
VAL (A$)	... Numerischer Wert von A$

Sonstige Funktionen

ASC (X$)	... ASCII-Code des ersten Zeichens von X$
CHR$ (x)	... liefert das Zeichen des ASCII-Codes x
FRE (var)	... Groesse des freien Speichers
INKEY$	... Tastaturabfrage
POS (n)	... Abfrage aktuelle Schreibposition
RND (1)	... Zufallszahl zwischen 0 und 1
SPC (n)	... Ausgabe von n Leerzeichen
TAB (n)	... Sprung zur n-ten Spalte
USR (x)	... Aufruf eines Maschinenprogramms mit Parameteruebergabe

Mathematische Operationen :

```
+  -  /  *  ^  <  <=  <>  >=  >  =  NOT  AND  OR
PI = 3.14159
```

Zusaetzliche BASIC-Befehle des KC 85/3 :

BLOAD	... Laden eines Maschinenprogramms
CIRCLE <x>,<y>,<r>[,f]	... Kreis mit dem Mittelpunkt x,y dem Radius r und der Farbe f
COLOR <v>,<h>	... Farbeinstellung der Vordergrund- v und Hintergrundfarbe h
KEY <n>	... Festlegung der Funktionstaste n
KEYLIST	... Abfrage der Funktionstasten
LINE <xa>,<ya>,<xe>,<ye> [,f]	... Linie von xa,ya nach xe,ye/Farbe f
LOCATE <z>,<s>	... Platzieren des Cursors in die Zeile z und Spalte s
PRESET <x>,<y>	... Loeschen des Punktes x,y
PSET <x>,<y>	... Setzen des Punktes x,y
PTEST (<x>)	... Test des Punktes x
RANDOMIZE	... Initialis. des Zufallsgenerators
SOUND <z1>,<v1>,<z2>,<v2> [,l] [,t]	... Tonausgabe
SWITCH <m>,<k>	... Ab- und Zuschalten eines Moduls
VPEEK (<I>)	... Lesen des Bildwiederholspeichers
VPOKE <I>,<J>	... Schreiben in den IRM

BASIC - Fehlermeldungen :

```
BS      ... Feldelement ausserhalb des dimens.Bereiches
CN      ... <CONT> nicht moeglich
DD      ... Feld doppelt definiert
FC      ... Unzulaessiger Funktionsaufruf
ID      ... INPUT und DEF FN nicht im Kommandomodus
IO      ... Einlesefehler
LS      ... Zeichenkette zu lang (>255 Zeichen)
MO      ... ein Operand fehlt
NF      ... fehlerhafte FOR...NEXT-Schleife
OD      ... zu wenig Daten in DATA-Zeilen
OM      ... Arbeitsspeicher zu klein
OS      ... Zeichenkettenspeicher zu klein
OV      ... numerischer Ueberlauf
RG      ... RETURN vor GOSUB
SN      ... Syntaxfehler
ST      ... String zu komplex oder zu lang
TM      ... Falsche Zuordnung von Variablentypen
UF      ... Funktion nicht definiert
UL      ... Zeile existiert nicht
/0      ... Division durch Null
?REDO FROM START
        ... Fehlerhafte Eingabe bei INPUT
?EXTRA IGNORED
        ... Ueberfluessige Daten bei INPUT
??      ... Unvollstaendige Dateneingabe bei INPUT
UL ERROR (wiederholt)
        ... Programmabsturz
            (Korrektur evtl. mit AUTO-Flag = 0 !)
```

Wichtige Adressen des BASIC-Interpreters :

Adresse (dezimal)	Erklaerung
772-773	Systemzellen fuer Aufruf USR-Funktion
845	AUTO-Flag
861	Schreibschutz vor Ausladen
862	LIST/EDIT - Sperrzelle
863-864	Adresse Beginn BASIC-Programm
983-984	Adresse Ende BASIC-Programm

(hexadezimal) CALL-Rufe	
0300	Warmstart des BASIC-Interpreters
C48A	Aufbereiten gelesenes Programm
C9F6	Uebernahme der Parameter bei USR
D0B1	Rueckgabe der Parameter bei USR

Tokens der BASIC-Befehle :

	8	9	A	B	C	D	E
0	END	OUT	LIST	^	EXP	LOAD	PAPER
1	FOR	ON	CLEAR	AND	COS	TRON	AT
2	NEXT	NULL	CLOAD	OR	SIN	TROFF	COLOR
3	DATA	WAIT	CSAVE	>	TAN	EDIT	SOUND
4	INPUT	DEF	NEW	=	ATN	ELSE	PSET
5	DIM	POKE	TAB(	<	PEEK	INKEY$	PRESET
6	READ	DOKE	TO	SGN	DEEK	JOYST	BLOAD
7	LET	AUTO	FN	INT	PI	STRING$	VPEEK
8	GOTO	LINES	SPC(	ABS	LEN	INSTR$	VPOKE
9	RUN	CLS	THEN	USR	STR$	RENUMBER	LOCATE
A	IF	WIDTH	NOT	FRE	VAL	DELETE	
B	RESTORE	BYE	STEP	INP	ASC	PAUSE	
C	GOSUB	!	+	POS	CHR$	BEEP	
D	RETURN	CALL	-	SQR	LEFT$	WINDOW	
E	REM	PRINT	*	RND	RIGHT$	BORDER	
F	STOP	CONT	/	LN	MID$	INK	

Erklaerung: Bei der Eingabe einer Befehlszeile werden alle Anweisungen durch den BASIC-Interpreter in sogenannte Token umgewandelt.

Aufbau einer BASIC-Programmzeile :

Im Normalfall beginnen alle BASIC-Programme bei der Adresse 401H (dezimal 1025).

Zeilenaufbau (Beispiel):

```
Adr.    Pointer   ZNR                                        Zeilenende
401H.  |0F |04 |0A |00 |81|58|B4|33|33|A6|31|30|30| 00|
        +          10   FOR X = 3 3 TO 1 0 0
40FH   |1B |04 |14 |00 |9E|58|2C|CC|28|58|29| 00|
        +          20   PRINT X , CHR$ ( X )
41BH   |21 |04 |1E |00 |82| 00|
        +          30   NEXT
421H   |00 |00 |00 |
       ------------ Programmende
```

Pointer ... Adresse der naechsten BASIC-Programmzeile
ZNR ... Zeilennummer der aktuellen Zeile
Zeilenende ... wird immer mit Byte 00 abgeschlossen
Innerhalb der Zeilen erkennt man deutlich die Codierung der BASIC-Befehle in Tokens.

Befehlsliste des KC-PASCAL :

Start des KC-PASCAL:	**PASENTRY**	<ENTER>
Restart ohne Programmloeschen:	**PASREC**	<ENTER>
Programmstart:	**PASRUN**	<ENTER>

Reservierte Worte:

AND	ARRAY	BEGIN	CASE	CONST	DIV
DO	DOWNTO	ELSE	END	FORWARD	FUNCTION
GOTO	IF	IN	LABEL	MOD	NIL
NOT	OF	OR	PACKED	PROCEDURE	PROGRAM
RECORD	REPEAT	SET	THEN	TO	TYPE
UNTIL	VAR	WHILE	WITH		

Spezialsymbole:

+	-	*	/	=	<>	<	<=
>=	>	(	)	[	]	(*	*)
^	:=	.	,	;	:		

Vordefinierte Namen

Die folgenden Namen kann man sich als in einem das gesamte Programm umschliessenden Block definiert denken, womit sie stets verfuegbar sind.

CONST	MAXINT=32767;
TYPE	BOOLEAN=(FALSE,TRUE);
	CHAR (Erweiterter ASCII-Zeichensatz)
	INTEGER=-MAXINT..MAXINT;
	REAL (Eine Untermenge der reellen Zahlen)
PROCEDURE	WRITE; WRITELN; READ; READLN; PAGE; HALT;
	USER; POKE; INLINE; OUT; NEW; MARK; RELBASE;
	TIN; TOUT;
FUNCTION	ABS; SQR; ODD; RANDOM; ORD; SUCC; PRED; INCH;
	EOLN; PEEK; CHR; SQRT; ENTIER; ROUND; TRUNC;
	FRAC; SIN; COS; TAN; ARCTAN; EXP; LN; ADDR;
	SIZE; INP;

Tastenbelegungen: INS ... '[' DEL ... ']'

Haeufige Fehlermeldungen:

```
 ^1  ... Zahl zu gross
 ^2  ... Semikolon wird erwartet
 ^3  ... Nichtdeklarierter Name
 ^8  ... ':=' wird erwartet
^10  ... Falscher Typ
^25  ... 'BEGIN' wird erwartet
^30  ... Dieser Name ist kein Typ
^34/35 ... '[' bzw. ']' wird erwartet
^62  ... Marke ist nicht vereinbart
```

Editierkommandos:

Kommando	Erklaerung
I n,m	Einfuege-Modus ab Zeile n im Abstand m
L n,m	Listen des Textes von n bis m
K n	Anzahl der List-Zeilen
D n,m	Loeschen der Zeilen n bis m
M n,m	Kopiert Zeile n auf Zeile m
N n,m	Umnummerieren mit erster Zeile n, Abstand m
F n,m,f,s	Von Zeile n bis m wird String f gesucht und evtl. durch s ersetzt
E n	Editieren von Zeile n (Funktionen s.unten)
P n,m,s	Auslesen des Textes von Zeile n bis m unter dem Namen s auf Band
G ,,s	Einlesen des Textes von Band
C	Compilieren
R	Programmstart (eines compilierten Programms)
T	Compilieren und Erzeugen eines COM-Files (dabei wird der Compiler zerstoert)
B	Rueckkehr in das Betriebssystem

Funktionen bei Editieren einer Zeile:

-->	ein Zeichen weiter
<--	ein Zeichen zurueck (kein Loeschen)
^	Sprung zum naechsten TAB
ENTER	die editierte Zeile wird uebernommen
Q	Abbruch des Editierens, Zeile bleibt erhalten
R	der Textpuffer wird nochmals mit der alten Zeile geladen
L	listet den Rest der Zeile
K	loescht das Zeichen an der aktuellen Cursorposition
Z	loescht alle Zeichen ab der Cursorposition
F	sucht den naechsten mit 'F' definierten String
S	ersetzt den eben gefundenen mit dem vorher definierten Austauschstring
I	fuegt Zeichen ab der aktuellen Cursorposition ein. Funktion bleibt bis zum naechsten <ENTER> erhalten.
X	setzt Zeiger auf das Ende der Zeile, wobei gleichzeitig die Funktion 'I' beginnt
C	Ueberschreiben der aktuellen Cursorposition, Funktion bleibt bis <ENTER> erhalten.

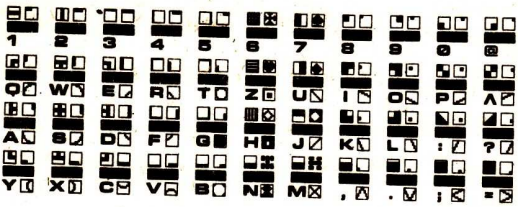
Fehlerbehandlung beim Compilieren:

Treten waehrend des Compilierens Fehler auf, so kann das Kommando 'C' abgebrochen und in den Editmodus uebergegangen werden:

E	Aufruf der angezeigten fehlerhaften Zeile
P	Aufruf der vorhergehenden Zeile

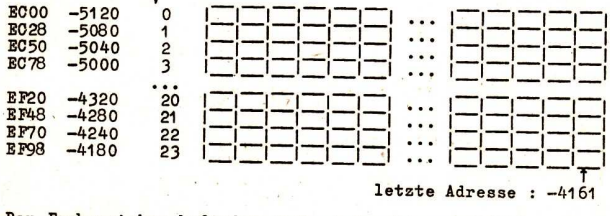
D	H	G	D	H	G	D	H	G	D	H	G	D	H	G	D	H	G	D	H	G
128	80		144	90		160	A0		176	B0		192	C0		208	D0		224	E0		240	F0	
129	81		145	91		161	A1		177	B1		193	C1		209	D1		225	E1		241	F1	*)
130	82		146	92		162	A2		178	B2		194	C2		210	D2		226	E2		242	F2	
131	83		147	93		163	A3		179	B3		195	C3		211	D3		227	E3		243	F3	
132	84		148	94		164	A4		180	B4		196	C4		212	D4		228	E4		244	F4	
133	85		149	95		165	A5		181	B5		197	C5		213	D5		229	E5	*)	245	F5	
134	86		150	96		166	A6		182	B6		198	C6		214	D6		230	E6		246	F6	
135	87		151	97		167	A7		183	B7	*)	199	C7		215	D7		231	E7		247	F7	*)
136	88		152	98		168	A8		184	B8		200	C8		216	D8		232	E8		248	F8	
137	89		153	99		169	A9		185	B9		201	C9		217	D9		233	E9		249	F9	
138	8A		154	9A		170	AA		186	BA		202	CA		218	DA		234	EA		250	FA	
139	8B		155	9B		171	AB		187	BB		203	CB		219	DB		235	EB	*)	251	FB	
140	8C		156	9C		172	AC		188	BC		204	CC		220	DC		236	EC		252	FC	
141	8D		157	9D		173	AD		189	BD		205	CD		221	DD		237	ED		253	FD	
142	8E	*)	158	9E		174	AE		190	BE		206	CE		222	DE		238	EE		254	FE	
143	8F		159	9F		175	AF		191	BF	*)	207	CF		223	DF		239	EF		255	FF	

Tastenbelegung fuer Grafikzeichen :



Erklaerung: links oben ... ohne Zusatztaste
rechts oben ... mit <SHIFT>
unten ... mit <CTRL>

Bildschirmaufbau :



Der Farbspeicher befindet sich ab der Adresse E800H.
Die Adresse des Farbatributes im Farbspeicher ergibt sich aus der Adresse des zugehoerigen Zeichens im Zeichenspeicher minus 400H (1024 dezimal).

ASCII - Tabelle :

1. Steuerzeichen

Hex	Dez	Taste	Funktion
02	2	CLLN	Loeschen aller eingeg. Zeichen
03	3	STOP	Abbruch der Eingabe
07	7	CTRL_G	Ausgabe eines Tones
08	8	<—	Cursor links
09	9	—>	Cursor rechts
0A	10	↓	Cursor runter
0B	11	↑	Cursor hoch
0C	12	CTRL_L	Bildschirm loeschen
0D	13	ENTER	Cursor an Anfang der Zeile
0E	14	CTRL_N	Bildschirminhalt drucken
10	16	CTRL_P	Drucker ein- und ausschalten
11	17	CTRL_Q	Kontrollton ein- und ausschalten
17	23	CTRL_W	Papiervorschub am Drucker
1B	27	ESC	
1D	28	RUN	

2. Buchstaben, Ziffern und Sonderzeichen

Dez		Dez		Dez		Dez	
32	Space	56	8	80	P	104	h
33	!	57	9	81	Q	105	i
34	"	58	:	82	R	106	j
35	#	59	;	83	S	107	k
36	$	60	<	84	T	108	l
37	%	61	=	85	U	109	m
38	&	62	>	86	V	110	n
39	'	63	?	87	W	111	o
40	(	64	@	88	X	112	p
41	)	65	A	89	Y	113	q
42	*	66	B	90	Z	114	r
43	+	67	C	91	[	115	s
44	,	68	D	92	\	116	t
45	-	69	E	93	]	117	u
46	.	70	F	94	^	118	v
47	/	71	G	95	←	119	w
48	0	72	H	96	`	120	x
49	1	73	I	97	a	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	c	123	{
52	4	76	L	100	d	124	|
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	TILDE
55	7	79	O	103	g	127	DEL

Betriebssystem:

Vorbild fuer das Betriebssystem des KC 85/1 ist das System CP/M 80. Der allgemeine Aufbau und alle Schnittstellen wurden uebernommen. Unterschiede existieren bei der Arbeit mit externen Datentraegern, da bei dem KC 85/1 ausschliesslich mit Magnetbandkassetten gearbeitet wird.

Bestandteile des Betriebssystems:

1. Console Command Programm (CCP) - Steuerung des Computers
2. BASIC Operating System (BOS) - Monitorunterprogramme
3. BASIC Input/Output System (BIOS) - Peripheriesteuerung

Speicheraufteilung:

Dezimal		Hexadezimal
0		0
	Arbeitsspeicher des Betriebssystems	
530		212H
	frei	
768		300H
	Arbeitsspeicher des BASIC-Interpreters	
1024		400H
	RAM Anwenderspeicher	
16384		4000H
	Erweiterung 1	
32768		8000H
	Erweiterung 2	
49152		C000H
	ROM-Erweiterung z.B. BASIC-Interpr.	
59392		E800H
	Farbspeicher	
60416		EC00H
	Bildspeicher	
61440		F000H
	Betriebssystem ROM	
65535		FFFFH

Kanaladressen :

Adresse	Erklaerung
128	CTC-Kanal 0, u.a. Kassettenarbeit
129 130,131	CTC-Kanal 1, frei fuer Anwender
136	Einstellung Farben, Bildschirm
137	PIO B, Daten
139	PIO B, Kommando
145	Tastatur-PIO B, Daten
202,203	E/A-Erweiterungsmodul
248-251	Analog-Digital-Wandler

Wichtige Arbeitszellen des Betriebssystems :

Adresse (hex.)		Bedeutung
0000	JMP WBOOT	Sprung zum Warmstart
0005	JMP BOS	Eintritt in BOS-Unterprogrammsystem
0013	JOYR:	Datenpuffer fuer Spielhebel 1
0015	LISW:	Schalter fuer Hardcopy (1 = ein)
0016	BSW:	Schalter fuer Kontrollton (1 = ein)
001B	DMA:	Zeiger auf Kassetten-E/A-Puffer
001D	HOUR:	Puffer fuer Stunden
001E	MIN:	Puffer fuer Minuten
001F	SEC:	Puffer fuer Sekunden
0024	LAKEY:	Merkzelle fuer letztes gueltiges Zeichen von Tastatur
0026	SHLOC:	Schalter fuer SHIFT LOCK
002B	CHARP:	akt. Spalte des Cursors (1-40)
002C	LINEP:	akt. Zeile des Cursors (1-24)
002D	CURS:	physische Adresse des Cursors
0036	EOR:	End of RAM-Adresse
003B	P1ROL:	1.zu rollende Zeile -1
003C	P2ROL:	letzte zu rollende Zeile +1
003D	P3ROL:	1.zu rollende Spalte -1
003E	P4ROL:	letzte zu rollende Spalte +1
003F	BUFF:	Puffer fuer das vom Cursor ueberschriebene Zeichen
005C	FCB:	Dateikontrollblock
	005C Name:	Dateiname
	0064 Typ:	Dateityp
	0069	Blockpruefsumme
	006B BLNR:	Gelesene Blocknummer
	006C LBLNR:	Zu lesende Blocknummer
	006D AADR:	Dateianfangsadresse
	006F EADR:	Dateiendadresse
	0071 SADR:	Startadresse
	0073 SBY:	Schutzbyte
0080	CONBU:	CCP-Eingabepuffer/Kassettenpuffer
0100	INTLN:	interner Zeichenkettenpuffer
01FE		Beginn Stack-Bereich
0200		Beginn Interruptadresstabelle
0202		Interruptadresse frei fuer Anwender CTC Kanal 1
E800		Beginn Farbspeicher
EC00		Beginn Zeichenspeicher (IRM)
EFC0		Systembyte
EFC1		64 bit Register fuer Speicherkonfiguration
EFC8	(-4152)	Cursor-Markierungszelle
EFE1		Adresse TTY-Treiber fuer LIST

Unterprogramme des BOS-Systems :

Rufnr.	Name	Erklaerung	Parameter
0	INIT	Neustart	
1	CONSI	Zeicheneingabe von Tastatur	ASCII-Zeichen in A
2	CONSO	Zeichenausgabe zum Bildschirm	Zeichen in E
5	LISTO	Ausgabe eines Zeichens zum Drucker	Zeichen in E
6	GETST	Spielhebelabfrage	Code der Spielhebel 1 in C, 2 in B
9	PRNST	Ausgabe eines Strings	Stringadresse in DE
10	RCOMB	Eingabe eines Strings Pufferinitialisierung: 1.byte Laenge des Strings 2.byte wirkliche Laenge nach Eingabe ab 3.byte String	Pufferadresse in DE
11	CSTS	Abfrage Konsolstatus	in A 0 keine Taste sonst ASCII-Code
13	OPENR	Einlesen Block 0	siehe FCB-Struktur
14	CLOSR	Lesen beenden	
15	OPENW	Ausgabe FCB	siehe FCB-Struktur
16	CLOSW	Ausgabe letzter Block	
17	GETCU	Ausgabe der Cursor-position	D: Zeile (hex.) E: Spalte (hex.) BC: Adresse
18	SETCU	Setzen Cursorposition	D: neue Zeile E: neue Spalte
20	READS	Lesen eines Blockes von 128 byte ab DMA	siehe FCB-Struktur
21	WRITS	Schreiben eines Blockes	siehe FCB-Struktur
22	SETTI	Stellen der Uhr	A: Stunden D: Minuten E: Sekunden
23	GETTI	Abfragen der Uhr	wie Nr.22
24	PRITI	Ausgabe der Uhr auf Bildschirm	Zieladresse in DE
25	INITA	Tastaturinitialisierung	
29	DCU	Cursor loeschen	
30	SCU	Cursor setzen	in BC Adresse zur.
31	COEXT	Komprimieren eines Strings	DE: Stringlaenge BC: Zieladr. zur.

Alle Unterprogramme werden durch den Systemruf **CALL 5** erreicht. Die Nummer des Unterprogramms ist vorher im **Register C** zu hinterlegen.

Programmierung von CTC und PIO :

1. PIO

	Bit	7	6	5	4	3	2	1	0
Kanalsteuerwort									
Modus Ausgabe		0	0	x	x	1	1	1	1
Interruptvektor		V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0
Steuerwort		I	x	x	x	0	0	1	1
Modus Eingabe		0	1	x	x	1	1	1	1
Modus bidirectional		1	0	x	x	1	1	1	1
Interruptvektor und Steuerwort wie Modus Ausgabe									
Modus Einzelbitsteuerung		1	1	x	x	1	1	1	1
Auswahlbyte - Belegung:		0 Ausgang					1 Eingang		
Interruptvektor		V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0
Steuerwort		I	x	x	Ma	0	1	1	1
Interruptmaske: Eingaenge mit Belegung 0 sind zugelassen									

2. CTC

	Bit	7	6	5	4	3	2	1	0
Interruptvektor		V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0
Kanalsteuerwort									
Modus Zaehler		I	1	-	-	-	ZK	x	1
Modus Zeitgeber		I	0	x	x	x	ZK	x	1
Bit 5 ... Vorteiler		0 = 16, 1 = 256							
Zeitkonstante		Dualwert zwischen 0 und 255							
Interruptperiode		**T = 2,5 MHz * Teiler * Zeitkonstante**							

Erklaerung:

I ... Kennzeichnung Interrupt erlaubt (Belegung 1) oder nicht erlaubt (0).
Ma ... Interruptmaske folgt (1) oder folgt nicht (0)
ZK ... Zeitkonstante folgt (1) oder folgt nicht (0)
x ... Belegung fuer einfache Faelle ohne Bedeutung

Registersatz des U 880 :

B C D E H L	Vordergrundregister (8 bit)
B' C' D' E' H' L'	Hintergrundregister (8 bit)
A A'	Akkumulator (8 bit)
F F'	Flagregister (8 bit)
IX IY	Indexregister (16 bit)
PC	Programmzaehler (16 bit)
SP	Stackpointer (16 bit)
I	Interruptregister (8 bit)
R	Refreshregister (8 bit)

Befehlsliste des U 880 :

8/16 bit-Ladebefehle, Arithmetik, Logik

	B	C	D	E	H	L	M	A	n
LD B	40	41	42	43	44	45	46	47	06
LD C	48	49	4A	4B	4C	4D	4E	4F	0E
LD D	50	51	52	53	54	55	56	57	16
LD E	58	59	5A	5B	5C	5D	5E	5F	1E
LD H	60	61	62	63	64	65	66	67	26
LD L	68	69	6A	6B	6C	6D	6E	6F	2E
LD M	70	71	72	73	74	75		77	36
LD A	78	79	7A	7B	7C	7D	7E	7F	3E
ADD	80	81	82	83	84	85	86	87	C6
ADC	88	89	8A	8B	8C	8D	8E	8F	CE
SUB	90	91	92	93	94	95	96	97	D6
SBC	98	99	9A	9B	9C	9D	9E	9F	DE
AND	A0	A1	A2	A3	A4	A5	A6	A7	E6
XOR	A8	A9	AA	AB	AC	AD	AE	AF	EE
OR	B0	B1	B2	B3	B4	B5	B6	B7	F6
CMP	B8	B9	BA	BB	BC	BD	BE	BF	FE
INC	04	0C	14	1C	24	2C	34	3C	
DEC	05	0D	15	1D	25	2D	35	3D	

x	(BC)	(DE)	(nn)	I	R
LD A,x	0A	1A	3A	ED57	ED5F
LD x,A	02	12	32	ED47	ED4F

x	AF	BC	DE	HL	SP
PUSH x	F5	C5	D5	E5	
POP x	F1	C1	D1	E1	
LD x,nn		01	11	21	31
LD x,(nn)		ED4B	ED5B	2A	ED7B
LD (nn),x		ED43	ED53	22	ED73
ADD HL,x		09	19	29	39
ADC HL,x		ED4A	ED5A	ED6A	ED7A
SBC HL,x		ED42	ED52	ED62	ED72
INC x		03	13	23	33
DEC x		0B	1B	2B	3B

Allgemeine Arithmetik, Steuerung und Interruptsteuerung

DAA	... 27	CCF	... 3F	SCF	... 37
CPL	... 2F	NEG	... ED44	NOP	... 00
HALT	... 76	DI	... F3	EI	... FB
IM0	... ED46	IM1	... ED56	IM2	... ED5E
RETI	... ED4D	RETN	... ED45		

Sprungbefehle, Unterprogrammaufrufe und Rueckkehr

	unbed.	C	NC	Z	NZ	PE	PO	M	P
JP	C3	DA	D2	CA	C2	EA	E2	FA	F2
CALL	CD	DC	D4	CC	C4	EC	E4	FC	F4
RET	C9	D8	D0	C8	C0	E8	E0	F8	F0
JR	18	38	30	28	20				

Restartrufe

nn	00	08	10	18	20	28	30	38
RST nn	C7	CF	D7	DF	E7	EF	F7	FF

JMP (HL) ... E9 DJNZ ... 10

Operationsbyte: fuer IX ... DD fuer IY ... FD

Austausch-, Stack- und Blockbefehle

EX DE,HL ... EB EXAF ... 08 EXX ... D9
EX (SP),HL.. E3 LD SP,HL.. F9

LDI	... EDA0	LDIR	... EDB0	LDD	... EDA8
LDDR	... EDB8	CPI	... EDA1	CPIR	... EDB1
CPD	... EDA9	CPDR	... EDB9	INI	... EDA2
INIR	... EDB2	IND	... EDAA	INDR	... EDBA
OUTI	... EDA3	OTIR	... EDB3	OUTD	... EDAB
OTDR	... EDBB				

Ein- und Ausgabebefehle

	B	C	D	E	H	L	F	A
IN	ED40	ED48	ED50	ED58	ED60	ED68	ED70	ED78
OUT	ED41	ED49	ED51	ED59	ED61	ED69		ED79

IN n ... DB OUT n ... D3

Spezielle Rotationsbefehle

RLCA	... 07	RRCA	... 0F	RLA	... 17
RRA	... 1F	RRD	... ED67	RLD	... ED6F

Bit- und Rotationsbefehle 1.Byte: CB

		B	C	D	E	H	L	M	A
Bit	0	40	41	42	43	44	45	46	47
	1	48	49	4A	4B	4C	4D	4E	4F
	2	50	51	52	53	54	55	56	57
	3	58	59	5A	5B	5C	5D	5E	5F
	4	60	61	62	63	64	65	66	67
	5	68	69	6A	6B	6C	6D	6E	6F
	6	70	71	72	73	74	75	76	77
	7	78	79	7A	7B	7C	7D	7E	7F
SET	0	C0	C1	C2	C3	C4	C5	C6	C7
	1	C8	C9	CA	CB	CC	CD	CE	CF
	2	D0	D1	D2	D3	D4	D5	D6	D7
	3	D8	D9	DA	DB	DC	DD	DE	DF
	4	E0	E1	E2	E3	E4	E5	E6	E7
	5	E8	E9	EA	EB	EC	ED	EE	EF
	6	F0	F1	F2	F3	F4	F5	F6	F7
	7	F8	F9	FA	FB	FC	FD	FE	FF
RES	0	80	81	82	83	84	85	86	87
	1	88	89	8A	8B	8C	8D	8E	8F
	2	90	91	92	93	94	95	96	97
	3	98	99	9A	9B	9C	9D	9E	9F
	4	A0	A1	A2	A3	A4	A5	A6	A7
	5	A8	A9	AA	AB	AC	AD	AE	AF
	6	B0	B1	B2	B3	B4	B5	B6	B7
	7	B8	B9	BA	BB	BC	BD	BE	BF
RLC		00	01	02	03	04	05	06	07
RRC		08	09	0A	0B	0C	0D	0E	0F
RL		10	11	12	13	14	15	16	17
RR		18	19	1A	1B	1C	1D	1E	1F
SLA		20	21	22	23	24	25	26	27
SRA		28	29	2A	2B	2C	2D	2E	2F
SLE		30	31	32	33	34	35	36	37
SRL		38	39	3A	3B	3C	3D	3E	3F

Flag-Register

Zero-Flag (Z) ... Flag wird 1 gesetzt, wenn Ergebnis 0
Cary-Flag (C) ... wird 1, wenn Uebertrag entstand
Sign-Flag (S) ... wird 1, wenn Ergebnis negativ
Parity/Overflow-Flag (P/V)
... wird 1, wenn Ueberlauf entstand oder bei einer logischen Operation das Ergebnis paarig ist

1. Editor-Assembler von robotron 'ASM':

Neustart von EDIT und ASM mit: <N> <ENTER>

Editkommandos: Alle Kommandos beginnen mit '#' !

```
#A              ... Anzeige und Aenderung Textpuffer
#B              ... Ruecksetzen auf Quelltextanfang
#E              ... Beenden EDIT, Rueckkehr in OS-Mode
#T [z] [,] ...  Listen des Quelltextes (Zeile [z], ab der
                Zeile [z])
#C <z> [,y]...  Einfuegen Kommentar an Zeile <z>
#N [z]          ... Streichen Kommentar (der Zeile [z])
#I <z>          ... Einfuegen von Text nach der Zeile <z>
#K <z>          ... Loeschen der Zeile <z>
#S <z>          ... Substituieren in der Zeile <z>
        Format: /alter String/neuer String/      <ENTER>
#W "name"   ... Schreiben des Quelltextes auf Band
#R "name"   ... Lesen des Quelltextes von Band
```

ASM-Kommandos:

Einstellen des Datenflusses:

```
HL:   Kopfzeile
MO:   Anlage des MC-Code: R ... Speicher, T ... Kassette
LO:   Anzeige des Textes: C ... TV, P ... Drucker

1.Schritt:    PASS 1      Erzeugen MC-File
2.Schritt:    PASS P      Ausgabe der Assemblerliste
              PASS A      Ausgabe Symbolreferenztabelle
              PASS X      Ausgabe Cross-Referenz-Tabelle
              PASS F      Ausgabe Fehlerliste
              PASS V      Veraenderung Datenfluss
              PASS M      Ausgabe MC-Code in Speicher oder
                          auf Kassette
              PASS E      ASM-Beenden, Rueckkehr in OS-Mode
```

Pseudoanweisungen des 'ASM', 'IDAS', 'ZM'

```
PN <name>     ... Programmname
END           ... Quelltextende
ORG <aadr>    ... Programmorganisation ab der Adresse <aadr>
EQU <ber>     ... Reserviert einen Bereich (<256 byte)
DB 0          ... Definiert 1 Byte
DA 00         ... Definiert Doppelbyte/Wort
DB '...'      ... Definiert Zeichenkette
```

Besonderheiten von SYPS K 1520:

Relative Spruenge werden mit '-#' gekennzeichnet.
Marken duerfen maximal 5 Zeichen lang sein.
Makros sind nicht gestattet.

2. Zusatzmonitor 'ZM' (SRZ):

Enthaelt 14 transiente Kommandos, u.a. einen Interpretativen Dialog-Assembler:

Kommandos	Erklaerung
ANALYSE	Banddaueranalyse
LOAD	Laden eines Maschinenprogramms
SAVE <aadr> <eadr> <sadr>	Ausladen eines Maschinenprogramms mit Anfangsadresse <aadr>, Endadresse <eadr> und Startadresse <sadr>
DISPLAY <adr>	Speicheranzeige
MODIFY <adr>	Eingabe von Maschinencode
MODIFY P <adr>	Ein- und Ausgabe zu Kanaladressen
JUMP <adr>	Start eines Maschinenprogramms
HEX-DEZ <zahl>	Umwandlung hexadezimal-dezimal-binaer
RE-ASM <adr>	Reassembler
COPY	Kopieren von Programmen
COLOR	Farbeinstellung
MEMORY	spezielle Speicheroperationen
MEMORY S <string>	Stringsuche im Speicher
PRINTER parameter	Druckerinitialisierung
FS	Fernschreiberansteuerung
IDAS	Interpretativer Dialog-Assembler

Kommandos des 'IDAS':

Neustart mit <N>, sonst beliebige von <N> verschied.Taste

Kommandos	Erklaerung
.A	Anzeige und Aenderung der Puffer
.E	Rueckkehr in das Hauptmenue
.K [z1[,z2]]	Loeschen der Zeilen z1 bis z2
.K	Loeschen des ganzen Quelltextes
.L	Listen des Quelltextes
.L <marke>	Listen ab Marke
.M	Auslesen Maschinencode auf Band
.R ... <N>	Einlesen eines Textes von Band
.R ... <J>	Schreiben in Textpuffer
.S	Umnummerieren
.T	Uebersetzen Quelltext in MC-Code
.W	Schreiben des Textes auf Band
.X	Anzeige der Registerinhalte
RUN	Uebersetzen und Starten des Programms

Achtung ! Bei Eingabe von Befehlen ohne Zeilennummer werden diese sofort ausgefuehrt.

Zur Gewaehrleistung eines fehlerfreien und erfolgreichen Arbeitens mit dem Kleincomputer ist ein sorgfaeltiges und gewissenhaftes Vorgehen notwendig:

1. Kleincomputer, Fernsehgeraet und Kassettenrecorder sind sicher aufzustellen. Das Einschalten erfolgt in der Reihenfolge Fernseher, Kassettenrecorder und erst **danach** der Kleincomputer.
2. Es sind **keine** Verbindungen/Kabel zu veraendern.
3. Zusatzmodule oder andere periphere Baugruppen sind **nicht zu entfernen.**
4. Das Druecken der <RESET>-Taste darf nur im aeussersten Notfall erfolgen. Programme sind prinzipiell mit den Anweisungen 'NEW' oder 'BYE' zu loeschen.
5. Ist es notwendig einen Kassettenrecorder anzuschliessen, muss besonders darauf geachtet werden, dass das Diodenkabel **nur** in die vorgesehene Buchse gesteckt wird. Ein falscher Anschluss fuehrt zur Zerstoerung des Kleincomputers.
6. Die Tasten sind ohne Kraftanstrengung zu betaetigen.
7. Essen und Trinken am Computerplatz ist nicht gestattet.
8. Tonkoepfe der Kassettenrecorder duerfen nicht verstellt werden, da sonst mit diesen Recordern aufgezeichnete Programme nicht mehr gelesen werden koennen.
9. Die Arbeitsgemeinschaftszeit ist sinnvoll zu nutzen. Computerspiele sind zwar auch nicht schlecht. Mit diesen kann man aber nur wenig lernen.
10. Ist es notwendig, doch einmal Zusatzmodule zu stecken oder zu entfernen, so ist dies nur im ausgeschalteten Zustand moeglich. Die Genehmigung des Arbeitsgemeinschaftsleiters ist einzuholen.
11. Wird die Arbeit mit dem Computer beendet, so ist das Geraet auszuschalten, **danach** Fernseher und Recorder. Die Netzstecker sind aus der Steckdose zu ziehen.
12. Sollten waehrend der Arbeit Stoerungen des Computers auftreten, so ist dies **sofort** dem Arbeitsgemeinschaftsleiter zu melden.
13. Fuer das Aufzeichnen von Programmen muessen unbedingt neuwertige Kassetten verwendet werden. Aeltere Magnetbandkassetten auf denen evtl. schon einmal Musik aufgezeichnet war, sind voellig ungeeignet.

Ausserdem sollte man stets beachten:

Schimpfen auf den Computer hilft nicht !
Er fuehrt nur das aus, was DU ihm eingegeben hast !

Umrechnungstabelle Dezimal-Hexadezimal :

	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1	16	17	18	19	20	21	22	23	24
2	32	33	34	35	36	37	38	39	40
3	48	49	50	51	52	53	54	55	56
4	64	65	66	67	68	69	70	71	72
5	80	81	82	83	84	85	86	87	88
6	96	97	98	99	100	101	102	103	104
7	112	113	114	115	116	117	118	119	120
8	128	129	130	131	132	133	134	135	136
9	144	145	146	147	148	149	150	151	152
A	160	161	162	163	164	165	166	167	168
B	176	177	178	179	180	181	182	183	184
C	192	193	194	195	196	197	198	199	200
D	208	209	210	211	212	213	214	215	216
E	224	225	226	227	228	229	230	231	232
F	240	241	242	243	244	245	246	247	248

•	9	A	B	C	D	E	F
0	9	10	11	12	13	14	15
1	25	26	27	28	29	30	31
2	41	42	43	44	45	46	47
3	57	58	59	60	61	62	63
4	73	74	75	76	77	78	79
5	89	90	91	92	93	94	95
6	105	106	107	108	109	110	111
7	121	122	123	124	125	126	127
8	137	138	139	140	141	142	143
9	153	154	155	156	157	158	159
A	169	170	171	172	173	174	175
B	185	186	187	188	189	190	191
C	201	202	203	204	205	206	207
D	217	218	219	220	221	222	223
E	233	234	235	236	237	238	239
F	249	250	251	252	253	254	255

Quellen :

- Betriebssystem KC 85/1,
- Programmierhandbuch Teil1 und 2 des KC 85/1,
- Beschreibung zum KC 85/1-Programm 'ASM',
  VEB robotron-Messelektronik "Otto Schoen" Dresden, 1986
- Dokumentation zum 'KC-PASCAL-Compiler',
  TU Karl-Marx-Stadt, Mugler, 1987
- URANIA-Sonderheft,
  Urania-Verlag, Grote/Voelz, 1987
- Mikroprozessoren - Mikroelektronische Schaltkreise,
  Militaerverlag, Barthold/Baeurich, 1985