

\*\*\*\*\*

# Programmkassette M 0191 für MRB Z1013

\*\*\*\*\*

Redaktion: IG-HC am IZ der TU-Dresden / AG Z1013 Dezember 1987

## Inhaltsverzeichnis

-----

Seite: A

-----

### Programme

Nr.	Anf.	Ende	Start	Typ	Name	RAM-Erw.	Zähler
-----							
01	E000	E3FF	E3EE	C	HEADERSAVE 5.95	ja	...
02	3000	3FFF	3FEE	C	HEADERSAVE 5.95	-	...
03	0100	037F	0100	C	HEADERPRINT 1.1	-	...
04	0100	1FFF	0100	C	TEXTED. Scf 6.1	-	...
05	0100	2FFF	0100	C	ASSEMBL. Scf 5.3	-	...
06	4000	5DFF	4000	C	OC-EDITOR 4.0	ja	...
07	0100	1FFF	0100	C	LOGICANALYSER Scf	-	...
08	0100	4080	1800	C	Hisoft-Pascal	ja	...
09	0100	3460	3450	C	CHESS-MASTER H	-	...
10	0100	0900	0100	C	WURMI	-	...
11	0100	2AFF	0300	C	HC-BASIC 85/1 H	-	...
12	2BC0	36F0	0000	B	INVASION	-	...
13	2BC0	4352	0000	B	OTHELLO	ja	...
14	2Boo	42AD	0000	B	HELEN	ja	...
15	2BC0	38BA	0000	B	KATZE-MAUS	-	...
16	2Boo	4F82	0000	B	GAMMON	ja	...
17	2BC0	A809	0000	B	TATUM	ja	...
18	2BC0	3DD8	0000	B	TRIPLETT	-	...
19	0100	3FFF	0100	C	DEMO-Z1013	-	...
20	0100	0300	0200	C	SPEICHERTEST	-	...

Seite: B (TEXTFILES)

-----

21	2000	3BBA	0000	I	HEADERSAVE 5.9		...
22	2000	3C31	0000	I	ASSEMBLER 5. 3		...
23	2000	3F58	0000	I	TEXTEDITOR 8<'1		...
24	2000	34D6	0100	I	SPRUNGVERTEILER		...
25	2000	2E7D	0000	I	LOGICANALYSER		...
26	2000	3B2A	0000	I	OCE 4.0		...
27	2000	26B6	0000	I	HEADERPRINT 1.1		...
28	2000	293C	0000	I	CHESS-MASTER H		...
29	2000	3E53	0000	I	HISOFT-PASCAL		...
30	2000	3C0B	0000	I	HC-BASIC TEIL 1		...
31	2000	2A5B	0000	I	HC-BASIC TEIL 2		...

BASIC-Programme mit KC-Interfaoe:

32	INVASION			
33	OTHELLO	(RAM-ERW.)		
34	HELEN	(RAM-ERW.)		
35	KATZMAUS		TYP	BEMERKUNG
36	GAMMON	(RAM-ERW.)	C	MC-PROGR.
37	TATUM	(RAM-ERW.)	B	BASIC
38	TRIPLETT		I	INTRODUCTION

## 2. Bedienung der Programmkassette

-----

Alle Programme auf dieser Kassette (ausser 2. Teil Seite 2) wurden mit einer speziellen SAVE/LOAD-Routine aufgezeichnet, welche vor Beginn der Arbeiten in den Rechner geladen werden muss. Dazu haben Sie zwei Moeglichkeiten (abhaengig vom vorhandenen RAM). Das Laden der SAVE/LOAD-Routine (Headersave) geschieht mit dem Monitorkommando L.

L E000 E3FF

bzw.

L 3C00 3FFF

Start/Restart: J E3EE

Start/Restart: J 3FEE

**Beim Laden der SAVE/LOAD-Routine KHTKR erst beim Ertoenen des ZWEITEN Vortones druecken!!**

Alle weiteren Programme werden mit dem Kommando @L ENTER bzw. @LXX ENTER (wenn Sie nicht selbst starten sollen) geladen. Bei den zu ladenden Programmen ist zu beachten, dass sie nicht laenger als der verfuegbare RAM, abzueglich des einen K fuer das Headersave, sein duerfen, wenn nur die Grundversion vorhanden ist. Notfalls koennen die Programme auch mit dem normalen Monitor-LOAD geladen werden (ENTER erst nach zweiten Vorton). Die BASIC-Programme im Headersave koennen nur dann aus dem HCBASIC heraus mit HLOAD geladen werden, wenn der Sprungverteiler auf FFFFH installiert ist. Ansonsten sind die BASIC-Programme mit dem KC 85/1-Interface zu verwenden (zu laden mit CLOAD"...).

Sollte der Einlesevorgang durch Fehlermeldung "bad record" unterbrochen werden, so stoppen Sie das Tape. Durch kurzes Zurueckspulen und Druecken von ENTER kann versucht werden, den Ladevorgang fortzusetzen (analog KC85/1). Der mitlaufende Memorypointer erscheint, wenn der Ladevorgang fortgesetzt wird.

Zu jedem sich nicht selbst erklaerenden Programm ist eine Beschreibung als Text-File in druckfaehigem ASCII-Code (mit Steuerzeichen NL{1EH}, FF{0CH}, TAB{09H}) auf der Kassette vorhanden. Mit Hilfe des Textverarbeitungsprogramms Scf koennen Sie den Text zeilenweise lesen (Kommando D).

Es ist auch moeglich die Texte stueckweise auf den Bildschirm zu laden: L EC00 EFFF (Bildwiederholtspeicher)

Anmerkung:

Einige Programme nutzen den Zusatzsprungverteiler (vorrangig Drucker- und Headersaveschnittstellen) und sind nur mit diesem lauffaehig!!

Die Rueckspulmoeglichkeit besteht sowohl im BASIC als auch in allen anderen Programmen und Dateneingaben.

-----~-----  
Die Weitergabe der SAVE/LOAD-Routine ist erwuenscht, wenn die Urheberschaft nicht veraendert oder verschleiert wird.

## Texteditor V 6.1 Scf/RB

### Programmstandort:

100H...1AFFH	Editor
1B00H...1FFFFH	Arbeits + SAVE-Bereich

Start: 100H

### Allgemeine Hinweise:

- \* Vor dem Start des Editors muss der Textspeicher (ab 2000H) mit 3 oder Text initialisiert werden.
- \* Nach dem Start fordert das Programm die Eingabe einer hexadezimalen Textanfangsadresse, welche mit ENTER abzuschliessen ist. Anschliessend geht das Programm in die Kommando-schleife, was am > zu erkennen ist.
- \* Die abgelegten Texte muessen intern mit ETX (03H) abgeschlossen sein. 70 mal der Wert 00H wird ebenfalls als Ende gewertet. Aus diesem Grund sollte der Speicher vorher mit 03H (ETX) belegt werden.
- \* Alle Kommandoeingaben bestehen aus mindestens einem Buchstaben. Die Zahlenangaben sind mit ENTER abzuschliessen. Alle nicht genutzten Kommandos werden ignoriert oder fuehren zur Fehleranzeige. Die Zahlenangaben erfolgen dezimal.
- \* Fuer die Magnetbandroutinen wurde das Headersave 5.95 physisch eingebunden (Standort: 700H - 0AFFH). Somit enthaelt der Editor ein eigenes Kassetteninterface.

### Kommandos:

A **xx**      maximale Zeilenlaenge ( < 130 )

B **xx**      Bildmodus mit aktiver Zeile **xx**  
Der Text wird auf dem Bild dargestellt und kann durch Cursorfunktionen korrigiert werden. Dabei wird eine aktive Zeile definiert, in der die Korrekturen stattfinden. Zusaetzlich werden die vier vorhergehenden und fuenf folgenden Zeilen aufgeblendet.  
Aus den Zeilen werden nur 28 Zeichen dargestellt. Ueberschreitet der Cursor die Position 20, so beginnt die aktive Zeile (bzw. bis Zeile 6 das gesamte Bild) nach links zu wandern.  
Der Cursor wird durch ein auf der Spitze stehendes Viereck dargestellt. Alle im Text befindlichen Steuerzeichen (00...1FH) werden durch die Zeichen ab A0H des Zeichengenerators dargestellt. Die Eingabe von Steuerzeichen ist nur im Insert-Mode moeglich und umfasst die Zeichen 10...17H.

Folgende Steuerfunktionen sind moeglich:

	Taste	Funktion
	^C	S4-K Ende Modus
	^H	<- Cursor links
	^I	-> Cursor rechts
	^K	S4-S Cursor hoch
	^J	S4-R Cursor runter (1Zeile vorwaerts)
	^A	S4-I Cursor ans Zeilenende
DEL	S2-G	Loeschen 1 Zeichen am Cursor
^E	S4-M	Cursor an Zeilenanfang
^Q	S4-A	Cursor rechts bis zum naechsten Leerzeichen
^P	S4-@	Cursor links bis zum naechsten Leerzeichen
^R	S4-B	Wandlung des Zeichens am Cursor von Gross- in Kleinbuchstaben oder umgekehrt und Cursor 1 Position nach rechts
^S	S4-C	Loeschen der Zeichen vom Cursor bis zum naechsten Leerzeichen (oder Zeilenende)
^T	S4-D	Loeschen der Zeichen vom Cursor bis Zeilenende
^G	S4-O	Bild wird so verschoben, dass gleiche Spaltenpositionen ueber-einander liegen (zusaeztl. Langlisting aus)
^V	S4-F	Langlisting ein
^M	Enter	wird in NL (1EH) gewandelt und eingefuegt

Alle Zeichen, welche keine Steuerwirkung besitzen, werden in den Text eingefuegt.

- C Tausch von CR o. CR-LF in NL (0DH,0AH ->1EH)
- D xx Anzeigen ab Zeile xx  
Mit Betaetigen von S4-K wird die Anzeige abgebrochen. S4-S bewirkt eine Zeilenrueckschaltung. Alle anderen Tasten bewirken die Ausgabe der naechsten Zeile.
- E Neunumerierung, sollte vor einer MBG-Ausg. ausgefuehrt werden. Vor seiner Nutzung aendern die Kommandos I,K,U nicht die Zeilennummerierung!
- I XX Insert nach Zeile XX  
Bis zum Abruch durch S4-K kann Text eingefuegt werden.  
Die Zeilen werden mit Enter abgeschlossen und dabei in den Text eingefuegt. Bei Ueberschreitung des im A-Befehl angegebenen Wertes wird die Zeile automatisch abgeschlossen.
- K XX (ENT) YY (ENT) loescht die Zeilen XX bis einschliesslich YY
- K XX (ENT,ENT) loescht die Zeile XX
- L Druckbefehle  
LT Listen Text  
LZ X Y Listen von Zeile X bis Y mit Zeilennummer

LS	Drucker synchronisieren
LE	Blatteinzug
LF	Formfeed
LC	Drucken im Direktmode (fuer Steuerzeicheneingabe). Die Steuerzeichen werden als Buchstabe eingegeben. Der Abschluss erfolgt durch FF. Die Druckereinbindung erfolgt durch Eintragen der Treiberadresse auf 449H/44AH (Uebergabe im Akku). Die Druckersynchr. erfolgt ueber Adr. in 447H/448H
M	Anzeige Menuebild mit allen Kurzzeichen # - String      * - dez. Zahl
N	Neustart des Programms
P	Loeschen der Paritaet ueber den gesamten Text
R (ENT)	Lesen von Magnetband Der alte Text wird ueberschrieben. Abfrage Typ: S - Assemblerquelle T - allg. Text I - Beschreibungen, Informationen Name: jedes eingeg. Zeichen ist signifikant
RZ XX (ENT)	Lesen von Magnetband und Anfuegen auf XX-ter Zeile (ansonsten wie R)
S XX (ENT) YY (ENT)	Uebernehmen der Zeilen XXX bis einschliesslich YY in den Save (Merkzettel)
T K1 K2	Tauschen von Zeichenketten Die Zeichenketten werden mit S4-K abgeschlossen und koennen 0...20 Zeichen lang sein. Es lassen sich alle Zeichen ausser 03H (=ETX = S4-K) tauschen.
U XX (ENT)	Saveinhalt nach Zeile XX einfuegen Das Kommando kann mehrmals genutzt werden, ohne 'S' erneut zu starten.
W	Magnetband schreiben W (ENT) Aufzeichnen des gesamten Textes WW (ENT) Wiederholen der Aufzeichnung WZ X (ENT) Y (ENT) Aufzeichnen von Zeile X bis Y
Y	Statusanzeige (gross)
Z	Statusanzeige (Zeilenzahl)
G	Wandeln des gesamten Textes in Grossbuchstaben
S4-K	Ende der Programmnutzung (Rueckkehr zum Monitor)

#### Nutzungshinweise:

1. Erfassen von Text
- Vor Nutzungsbeginn muss der Textspeicher mit 3 geloescht werden (Monitor --> K )
  - Bis zu einer Textlaenge von ca. 4 K ist es empfehlenswert im

Bildmodus (B) zu schreiben.

- Die Zeilenenden werden dabei als Dreieck dargestellt, Tabulatoren als kleiner Winkel.
- Bei laengeren Texten wird die Zeicheneingabezeit zu lang, so dass in den Insertmodus (I) uebergangen werden muss.
- Um die Eingabegeschw. im B-Mode zu erhoehen, ist es auch moeglich, mit N eine neue Anfangsadresse festzulegen, die mitten im erfassten Text liegen kann.  
Damit ist es auch moeglich, mehrere Texte zugleich im Speicher zu halten. Diese Texte koennen mit Hilfe des Saves gemischt werden.

## 2. Aendern von Text

- geaendert wird grundsaeztlich im Bildmodus
- da es bei sehr langen Texten passieren kann, dass die Zeiten sehr lang werden, ist auch folgende Vorgehensweise moeglich:
  - \* Abspeichern des zu aendernden Textbereichs in den Save (S)
  - \* Umweisen des Textspeichers auf einen freien Bereich (N)
  - \* Einfuegen des Saves in den neuen Text (U)
  - \* Bearbeiten des Textes (B)
  - \* Rueckspeichern in den Save (S)
  - \* Umweisen des Textspeichers auf den den alten Bereich (N)
  - \* Einfuegen an den alten Standort (U)
  - \* Loeschen des unbearbeiteten Bereichs im alten Text (K)

Autor: Gerd Schoenfelder (IZ)  
letzter Bearb.: Rainer Brosig (IZ)

\*\*\*\*\*  
**HEADER-SAVE V 5.9(C) by RB**  
 \*\*\*\*\*

-Programmstandort: E000H-E3FFH Start: E3EEH  
 eigentliche SAVE-LOAD-Routine E000H - E3EDH  
 fuer Grundversion Standort 3C00-3FFF, St:3FEE

-benutzte Zellen:

DATA: 13H Blockadresse des gelesenen Blocks  
 ZILAD: 25H zu lesende Blockadresse  
 SOIL: 16H Start of Input-Line  
 ARG1: 1BH Anfangsadresse File  
 ARG2: 1DH End -"-  
 ARG3: 23H Start -"-  
 DATA+2:15H Zeichenzaehler bei Nameneingabe  
 Pufferbereich Kopf: 0E0H-0FFH

-Die orig. SAVE/LOAD-Routine aus dem Monitor wird nicht benutzt.  
 -Beim Start wird die Kommandoschleife auf B0H geladen (B0H-B5H).  
 -Alle weiteren Aufrufe erfolgen mit @ ... .

**SAVE:** @S anfahr. endadr. startadr. **ENTER**

danach Abfrage TYP und FILENAME

TYP: C-COM-File selbststartend

T-TEXT-File

B-BASIC-Programm

M-Maschinenprogramm, nicht selbststartend

S-SOURCE-Assemblerquelle (Quelltext)

P-PASCAL-Proramm

D-DATA-Datenfelder

I-Introduction, Informationen

E-EPROM-Inhalte

SPACE-ohne Typ

FILENAME: max. 16 Zeichen (alles zugelassen, auch Space)

-Beim Abspeichern wird in jeden Blockkopf die physische Anfangs-  
 adresse eingetragen, welche beim Einlesen ausgewertet werden kann.  
 -Das File bleibt trotzdem unter Weglassen des Kopfblocks mit  
 dem originalen Monitor-Load lesbar!  
 -Die Einfuehrung eines Memorypointers erleichtert die Kontrolle  
 ueber den Abarbeitungsstand.  
 -Nach dem SAVE-Vorgang kann eine Verify-Funktion genutzt werden,  
 die nur die Blockchecksummen ueberprueft. Der Memorypointer zeigt  
 dabei wieder den Abarbeitungsstand und im Fehlerfall die Adresse  
 des fehlerhaften Blocks an. Die Kontrolle erfolgt erst nach dem  
 Lesen eines Kopfblocks und endet nach dem Lesen der zu dem File  
 gehoerenden letzten Kopfadresse. Alles, was gelesen wird, er-  
 scheint als ASCII-Interpretation auf der obersten Bildschirmzeile.  
 -externer Aufruf: **CALL SARUF** (E003H) moeglichst ueber Sprungver-  
 teiler (FFF4H). Der 1.Registersatz und AF' werden zerstoeert!

Parameteruebergabe: Zellen 1BH - Anfangsadresse

1DH - Endadresse

23H - Startadresse

Akku 3AH - Wiederholen der SAVE-Funktion  
 mit gleichem Kopf (gilt auch fuer  
 den Kopf eines vorangegangenen LOAD)  
 -Typuebergabe in H(IY), sonst H(IY)=0

-Bei Eingabe von **@S::** wird die SAVE-Funktion mit den alten  
 Kopf-Parametern ausgefuehrt. Diese koennen auch von der LOAD-Fkt.  
 stammen (Nutzung zum schnellen Kopieren von Files).

**LOAD :**    @L - Laden das naechstfolgenden Files ohne Namen- und Typkontrolle  
               @LN - Laden eines Files mit Namen- und Typkontrolle  
               Es gilt, dass alles signifikant ist, was eingegeben wird, das heisst, wird nur mit Enter quittiert, erfolgt fuer den Typ oder den Namen keine Kontrolle. Es muessen nur so viele Zeichen eingegeben werden, wie fuer eindeutige Signifikanz notwendig sind.

**Achtung:**    Es muss darauf geachtet werden, dass der gewaehlte Name auf dem Bildschirm stehen bleibt, da kein Nameingabepuffer existiert.

- Erfolgt der Aufruf mit **X**, z.B. @LNX oder @LXX, wird nach dem Laden auf einen Autostart bei COM-FILE's verzichtet (X=alle Zeichen ausser SPACE).
- Der Rechner befindet sich so lange in einer Warteschleife, bis ein gueltiger Kopf gelesen oder mit BREAK (S4-K, CTRL-C) abgebrochen wird. Der Abruch gelingt nur bei anliegendem Signal, ansonsten mit RESET. Ungueltige Koepfe werden angezeigt. Nach sieben Koepfen ist der gewaehlte Name aus dem Bildschirmbereich gelaufen und es ist kein Laden mehr moeglich. Man sollte also wenigstens in die Naehe des gesuchten Files spulen.
- Soll ein File auf eine andere Adresse geladen werden, als die im Kopf angegebene geladen werden, ist es moeglich, eine neue Anfangsadresse anzugeben: @L XXXX. Es werden dabei automatisch der Autostart gesperrt und die neue Anfangs- und Endadresse errechnet sowie angezeigt. Diese Funktion ist nur ohne Namenssignifikanz nutzbar! (bei externem Aufruf auch mit)
- Sollte bei einem File, welches mit Blockadressen aufgezeichnet wurde, ein Lesefehler auftreten oder ein Block ueberlesen worden sein, so wird das Einlesen unterbrochen und eine Fehlermeldung ausgegeben. Durch kurzes Zurueckspulen und Druecken von ENTER kann versucht werden, den fehlerhaften Block noch einmal zu lesen (analog KC 85/1). Der Memorypointer erscheint, wenn der Ladevorgang fortgesetzt wird.
- Files ohne Kopfblocknummern (Versionen unter 5.0) koennen ab Version 5.8 nicht mehr gelesen werden. Es wird nur der Kopf angezeigt und das Einlesen wird mit "bad record" unterbrochen.
- Wird ein Blockkopf mit Inhalt 0FFFFH gelesen, wird der Ladevorgang mit "bad record" abgebrochen.
- externer Aufruf: **CALL LORUF** (E000H) moeglichst ueber Sprungverteiler (FFFF1H). Der 1.Registersatz und AF' werden zerstoeert.

Parameteruebergabe:

- \* H(IY)=0 - Typ wird abgefragt  
           H(IY)=Typkennzeichen -> Typvorgabe -> keine Abfrage Typ
- \* A=0 - ohne signifikante Kopfkontrolle  
   A=4EH - mit signifik. Kopfkontrolle, Typ- & Namenabfrage  
   A=41H - ohne Kopfkontrolle
- \* L(IY)=20H - Freigabe des Autostarts bei COM-FILE's

- Beim Laden auf eine neue Anfangsadresse muss auf Zelle 1BH die neue Anfangsadresse uebergeben werden ( > 0FFH), ansonsten muss die Zelle mit 0 initialisiert werden.
- Fuer den externen Aufruf wurden noch weitere Unterprogramme zu- gaenglich gemacht:



```

CALL BLMK (E00CH; SPV: FFC7H) - Lesen eines Blocks
Parameteruebergabe:
Zellen 25H/26H * Kopfinhalt des zu lesenden Blocks
HL           * Ladeadresse des Blocks
Return:
Zellen 25H/26H * Kopfinhalt + 20H
HL           * HL:=HL+20H

CALL BSMK (E00FH; SPV: FFC4H) - Schreiben eines Blocks
Parameteruebergabe:
HL * Quelladresse Block
IX * Kopfinhalt
DE * Anzahl der Sync.-Bits
Return:
HL * HL:=HL+20H

CALL SUCHK (E012H; SPV: FFC1H) - Suchen eines Kopfblocks und
Uebergabe des Inhalts im Kopfpuffer (E0-FFH), wo er
vom aufrufenden Programm ausgewertet werden kann.

CALL AKP   (E015H; SPV: FFB EH) - Aufbereitung Kopfpuffer
Parameteruebergabe wie bei SARUF

```

```

Kopfaufbau: Byte 0 - 1  Anf.adr.
               2 - 3  Endadr.
               4 - 5  Startadr.
               6 -0BH frei fuer Zusatzinformationen
                  0CH Typkennzeichen
               0DH-0FH 3 x 0D3H = Kopfkennzeichen
               10H-1FH 16 Byte Namensblock

```

**Hinweise, Fehler bitte an R. Brosig, W.Florin 2c, COSWIG, 8270.**



```

*****
***                                     ***
***           A S S E M B L E R           ***
***                                     ***
***                   V 5.3                   ***
***                                     ***
***                   01.10.87                   ***
***                                     ***
*****

```

\* Programmstandort :

```

    100H...5C0H  Zusatzmonitor
    1000H...2C6FH Assembler
    2E00H...2F6FH Systemzellen
    800H...0FFFH Symboltabelle

```

\* Programmstart :

```

    100H  Neustart , Befehl '#HALT' = 'END'
    1003H  Restart
    1006H  Restart , BEFEHL '#HALT' = Kommentar

```

\* Quellformat:

- Die Marken sind max. 5 Zeichen lang und beginnen mit Buchstaben.
- Marken werden mit Doppelpunkt abgeschlossen.
- SPACE und TAB werden als Trennzeichen gewertet. Sie koennen mehrfach auftreten.
- Das Namensfeld kann entfallen.
- Die Quellzeile darf ohne Kommentar max. 85 Zeichen lang sein. Der Kommentarteil wird bei der Uebersetzung ausgeblendet.
- Die Zeile kann mit NL oder CRLF abgeschlossen werden.
- Eine leere Zeile wird als Kommentar gewertet.
- Registernamen gelten als vereinbarte Symbole, die Befehlsmnemonik nicht.

\* Variablen:

- Alle 16-bit-Werte koennen durch einen mehrteiligen Ausdruck aus Variablen gebildet werden.
  - Alle 8-bit-Konstanten koennen mit einem Ausdruck definiert werden. Es werden dabei die niederen 8 Bit der Variablen verwendet und die oberen 8 Bit geprueft.
  - Ein Ausdruck kann durch + und - verknuepft werden. Er kann mit - anfangen. Durch das Zeichen \* werden die Teile und-verknuepft, durch / oder-verknuepft. Die Verknuepfungen werden von links nach rechts ohne Prioritaet abgearbeitet.
  - Als Variable / Konstante sind zugelassen :
    - 4 Zeichen hex, mit Ziffer beginnend und mit 'H' abgeschl.
    - 4 Zeichen oktal, mit 'Q' abgeschl.
    - 4 Zeichen dezimal ohne Abschluss
- Name
- # ergibt den Wert des PC vor Befehlsausfuehrung
- 1 Zeichen Literal in Hochkomma
- 2 Zeichen Literal in Hochkomma
- (linkes Zeichen = NWT )

16 Zeichen dual, mit 'B' abgeschlossen  
Die Werte koennen vorzeichenbehaftet sein.

\* Spezielle Formate:

- Die Befehle LD X,H(nn)  
und LD X,L(nn)  
sind zugelassen. nn kann dabei ein Ausdruck sein.
- Bei den Anweisungen DA,ORG,EQU,BER darf nur ein  
Wert erscheinen, der durch einen Ausdruck gebildet  
werden kann.
- Beim Befehl DB sind numerische Werte, Marken und  
Textketten zugelassen. Sie sind durch Komma zu trennen.
- Fuer (HL) muss M geschrieben werden.

\* Programmende durch 'END' oder '#HALT' gleichberechtigt.  
Die Bewertung des Befehls '#HALT' ist abhaengig von der Start-  
adresse.

\* Zusatzbefehle:

- HEX Definition von 2-stelligen Hex-Konstanten, durch  
Komma getrennt. Der Buchst. H und Fuehrungsnullen  
entfallen.  
z.B.: HEX 11,55,AA,1B,F3,00,DC
- PRNT Listensteuerung volle Liste
- PRNO "=" Fehlerliste
- PRTH "=" Hex-Liste
- PN wird ignoriert
- TITL "="
- alle Befehle mit LX,HX,LY,HY
- SLL Rotation

\* Fehlerausschriften:

- M Speicherfehler
- U unbekannter Befehl
- K Konstantenfehler (Wert zu gross)
- F Formatfehler
- A Adressenfehler
- N unbekannter Name
- T Tabellenueberlauf
- L Befehlszeile zu lang (ohne Komm.)
- V Adressenversatz zwischen 1. und 2. DL
- I Input-Fehler

\* Systembedienung:

- Start: 100H
- Restart: 1003H

- Durchlaufsteuerung:

- N Neustart (loescht Symboltabelle)
- 1 1. DL (keine Liste, sonst = 2.DL)
- 2 2. DL (Uebersetzung)
- ^C Ende der Nutzung (CTRL-C = 03H = S4-K)

Es koennen mehrere Programme hintereinander uebersetzt  
werden. Ein Programm kann dabei auf Marken aus den anderen  
Programmen zugreifen.

- Laufsteuerung (Anforderung mit 'A >' )

Ausgabeformat S Symbolliste  
(Zeile - Name - Wert)

	H	Hexliste (ohne Quellliste)
	F	Fehlerliste
	L	volle Liste
	M	Symboltabelle ausgeben (ohne Lauf) (Name - Wert )
	P	Zuschalten des Druckprotokolls
	V	Seitenvorschub und Abschalten des Druckers
	I	Initialisierung des Druckers
Zeitsteuerung	N	max. Geschwindigkeit
	T XXXX	Zeitlauf (Zeit in ms bis max. 1000)
Adressen	A XXXX	Zwangsadr. fuer ORG (Aenderung der ORG-Anweisung im Dialog)
	O XXXX	Offset zu ORG (ergibt Speicherbereich, auf dem der MC abgelegt wird Zweier-Komplement in Hex)
	Q XXXX	Quellanf.-Adr. (nur bei Mehrfachuebersetzungen im 2.DL)
	K	Aufhebung der A-Anweisung
Lauf	R	Run
	^C	Ende der Nutzung

Bei Zeitsteuerung = T kann die Uebersetzung durch SPACE nach jeder Zeile unterbrochen werden. Fortgesetzt wird durch Betaetigen einer beliebigen Taste.

\* Interne Funktionen

Marken mit (TAB) als erstes Zeichen koennen mehrfach verwendet werden. Ihr Wert wird auch im 2.DL veraendert, wodurch sie nur fuer Rueckwaertsspruenge zu verwenden sind oder unmittelbar vor ihrer Nutzung zu definieren sind.

Die Eingabe von ORG-Adressen wird gekellert . Es koennen max. 10 Adressen eingegeben werden. Die Adresse 0000 ruft dabei keine Wirkung hervor .

\* Modifikationen:

- 1) Bei fehlendem Druckeranschluss muss auf den Adressen 447H/448H und 449H/44AH 0D4H/03H eingetragen werden. Adresse Druckertreiber in 449H/44AH eintragen. Adresse Druckersynchr. in 447H/448H    "-"
- 2) Die Lage und Groesse der Symboltabelle kann durch Veraenderung der folgenden Zellen geaendert werden:

< 2C02H / 2C03H >	=	Anfangsadresse
< 2C04H / 2C05H >	=	Laenge

Autor: Gerd Schoenfelder  
letzter Bearbeiter: Rainer Brosig



\*\*\*\*\*  
**Sprungverteiler fuer Z 1013-Erweiterungen**  
 \*\*\*\*\*

Der Zusatzsprungverteiler wurde entsprechend einer Einigung innerhalb der Interessengemeinschaft Heimcomputer des IZ an der TU-Dresden folgendermassen festgelegt:

Beginnt auf Adresse **FFFFH** abwaertsfuehrend:

**FFFDH - JMP INKEY** holt ein Zeichen von Tastatur in den Akku;  
 kommt beim 2. Aufruf nur zurueck, wenn Taste  
 zwischendurch losgelassen wurde

**FFFAH - JMP POLL** bringt immer ein Zeichen im Akku zurueck, egal  
 ob Taste losgelassen wurde oder nicht

**FFF7H - JMP STAT** uebergibt Tastaturstatus im Akku  
 A=0 - keine Taste gedrueckt  
 A=FFH - Taste gedrueckt  
 die Abfrage erfolgt ohne Ruecksicht, ob die  
 Taste schon vor dem Aufruf gedrueckt war und  
 hinterlaesst trotz gedrueckter Taste den Status  
 'letztes Zeichen war 0' ((Zelle 4)=0) um eine  
 evtl. nachfolgenden INKEY-Routine nicht zu  
 sperren

**FFF4 - JMP SARUF** ruft die SAVE-Routine des Headersave  
 ! zerstoert 1. Registersatz + AF'  
 Parameteruebergabe:  
 Zellen 1BH - anadr.  
 1DH - endadr.  
 23H - strtadr.  
 Akku 3AH - Wiederholen der SAVE-Funktion mit  
 gleichem Kopf  
 H(IY) Typvorgabe (in ASCII), sonst 0

**FFF1H - JMP LORUF** ruft LOAD-Routine des Headersave  
 ! zerstoert 1. Registersatz + AF'  
 Parameteruebergabe:  
 Zellen 1BH - neue Anfangsadresse des Files  
 sonst 0  
 Akku 0 - ohne signifikante Kontrolle  
 4EH - mit signifikanter Kopfkontrolle  
 (Typ) + Namenabfrage  
 H(IY) 0 - Typ wird abgefragt  
 Typkennzeichen (in ASCII), keine Typabfr.  
 L(IY) 20H - Freigabe Autostart bei COM-Files

**FFEEH - JMP ZMINI** Initialisierung der Z-Monitorrufe auf B0H

**FFEBH - JMP DRDEL** setzt den logischen Druckertreiber zurueck

**FFE8H - JMP DRAKK** uebergibt den Akkuinhalt an den logischen  
 Druckertreiber

**FFE5H - JMP BSDR** druckt den Inhalt des BWS und kehrt in das  
 rufende Programm zurueck

**FFE2H - JMP HARD** uebergibt den Akkuinhalt an logischen Drucker-  
 treiber wenn ein Flag im Rechner gesetzt ist;  
 wandelt CR (0DH) in NL (1EH/0DH-0AH)

	! nur verwenden, wenn Programm eine eigene Bildschirmverwaltung hat
<b>FFDFH - JMP DRZEL</b>	wie DRAKK, nur das der Inhalt von 1BH uebergeben wird (vorgesehen, um im BASIC mit POKE zu drucken)
<b>FFDCH - JMP BEEP</b>	erzeugen eines kurzen Signals
<b>FFD9H - JMP ASTA</b>	Ausgabe Akkuinhalt als ASCII-Zeichen an PUNCH
<b>FFD6H - JMP BSTA</b>	Ausgabe Akkuinhalt als Byte an PUNCH
<b>FFD3H - JMP AIN</b>	Eingabe eines ASCII-Zeichens vom LBL in den Akku
<b>FFD0H - JMP BIN</b>	Eingabe eines Bytes vom LBL in den Akku
<b>FFCDH - JMP DRINI</b>	Initialisierung des logischen Druckertreibers
<b>FFCAH - JMP ZEIDR</b>	uebergibt ein Zeichen im Akku an physischen Druckertreiber
<b>FFC7H - JMP BLMK</b>	Lesen eines Blocks vom Headersave Parameteruebergabe: Zellen 25H/26H * Kopfinhalt des zu lesenden Bl. HL * Ladeadresse des Blocks Return: Zellen 25H/26H * Kopfinhalt + 20H HL * HL:=HL+20H Abbruch des Lesens bei Kopfinhalt=0FFFFH oder DMA > Endadr in ARG2 (1DH)
<b>FFC4H - JMP BSMK</b>	Schreiben eines Blocks im Headersave Parameteruebergabe: HL * Quelladresse Block IX * Kopfinhalt DE * Anzahl der Sync.-Bits Return: HL * HL:=HL+20H
<b>FFC1H - JMP SUCHK</b>	Suchen eines Kopfblocks und Uebergabe des Inhalts im Kopfpuffer (E0-FF), keine Auswertung
<b>FFBEH - JMP AKP</b>	Aufbereitung Kopfpuffer mit Namenabfrage Parameteruebergabe wie bei SARUF
<b>FFBBH - JMP GETST</b>	Abfrage der Joysticks und Uebergabe des Er- gebnisses in BC (B-links,C-rechts) mit folgen- den Bit-Bedeutungen (Belegung mit 1): Bit 0 - links 1 - rechts 2 - runter 3 - hoch 4 - Aktionstaste Z-Flag=1, wenn keine Betaetigung vorliegt CY-Flag=1, wenn Spielhebel nicht angeschlossen
<b>FFB8H - JMP SOUND</b>	Ausgabe einer vollen Periode auf die Tonband- buchse, sowie auf Bit 7 vom Systemport (User-P) Uebergabe der Periodendauer in C mit T=n*33mks+20mks (2MHz)



\*\*\*\*\*  
**LOGIKANALYSATOR - PROGRAMM**  
 \*\*\*\*\*

Das Programm zur Logik-Zustands- und Logik-Zeitanalyse laesst sich auf jedem K1520 implementieren. Mit ihm koennen langsame Vorgaenge, wie sie bei E/A-Geraeten und Datenuebertragungen auftreten, verfolgt werden. Die Ausgabe erfolgt auf Bildschirm, Systemdrucker und/oder dem graphischen Drucker K6311. Das Programm benoetigt 2 KByte ROM und 2,5 KByte RAM. Die zeitliche Aufloesung reicht bis zu einer Impulsbreite von 25 us. Die Datenmenge umfasst 2 Kbyte. Als Eingangspegel am Standardtor sind TTL-Signale verarbeitbar.

Die Zeitangaben gelten fuer eine Rechnertaktfrequenz von 2 MHz.

## 2. Technische Parameter

Datenbreite : 8 Bit  
 Speichertiefe : 2048 Bit  
 Startadresse : 100 H  
 Restartadresse : 1000 H !!!!!  
 Eingangspegel : 0 ... 5 V ungepuffert  
 Standardtor : 00 H (PIO-Tor A)

### 2.1. Zeitanalyse

Aufloesung : 10 us  
 25 us ... (10 us) ... 2585 us  
 Triggerung : Vortriggerung  
 Start, wenn (Daten \* Maske) = Triggerwort  
 (Bei Aufloesung 4 us ist die Maske = 0FFH)

### 2.2. Zustandsanalyse

Aufloesung : ca. 30 us je Halbtakt  
 Takt : aus Datenstrom  
 Erlaubte Taktbits werden ODER-verknuepft.  
 Takt = low wird zuerst aufgezeichnet  
 Triggerung : Vortriggerung  
 Start, wenn :  
 (Daten \* Maske) exor (XOR-Maske) = 0

## 3. Bedienung

Alle Einstellungen erfolgen im Dialog. Ausgaben mit '?' erfordern die Eingabe von J oder N, alle anderen nume-

rische Werte. Die Eingaben erfolgen alle in hexadezimaler Form.

### 3.1. Aufnahmeparameter

Die Parameter der Datenaufnahme lassen sich vor jeder Messung neu einstellen. Die Entscheidung dazu wird im Dialog abgefordert. Der Rechner fordert nur die fuer die eingestellte Messart erforderlichen Werte ab, wobei die logischen Verknuepfungen der triggerworte den Angaben aus 2.1. und 2.2. zu entnehmen sind.

### 3.2. Datenanzeige

Die Datendarstellung und der Neustart von Messungen erfolgt von einem zentralen Anzeigebild aus.

#### 3.2.1. Neustart Messung

Das Programm wird entsprechend einem Restart neu gestartet. Es entfaellt allerdings die Auswahl der Grundbetriebsart (Zustand / Zeit).

#### 3.2.2. Neustart Programm

Es erfolgt ein voelliger Neustart einschliesslich Initialisierung.

#### 3.2.3. Anzeige Zeitdiagramm

Es wird der Speicherinhalt zusammen mit allen wichtigen Parametern dargestellt. Auf dem Bildschirm werden 30 Taktzustaende aufgeblendet. Ausser den Messdaten sind folgende Angaben dargestellt :

T-A	:	Seitenanfang ( = Adresse linker Rand )
POINT	:	Adresse Pointer
DT	:	Differenz zwischen Marke ( M ) und Pointer in Takten
HEX	:	Pointerbyte hexadezimal
DEZ	:	Pointerbyte dezimal

Als Kommando werden folgende Eingaben akzeptiert :

U	:	30 Takte vorwaerts (Bild + Pointer)
D	:	30 Takte zurueck        "-"
CUR	:	Cursor rechts = 10 Takte vor
CUL	:	Cursor links = 10 Takte zurueck
+	:	Pointer 1 Takt vor
-	:	Pointer 1 Takt zurueck
X +/-	:	Pointer X Takte vor/zurueck
M	:	Marke wird auf Pointer gesetzt
ENTER	:	Ende

Autor: Gerd Schoenfelder

\*\*\*\*\*  
**OBJEKTCODEEDITOR OCE FUER Z1013**  
 \*\*\*\*\*

Der Objektcodeeditor enthaelt einen Assembler zur Uebersetzung von Befehlen im Mnemonikcode in Maschinencode und den Reassembler zur Rueckuebersetzung.  
 Ausser dem Maschinencode wird noch eine Markentabelle abgespeichert.

Programmstandort:

4000H ...5DFFH

OCE-Stack:

200H abwaerts

Hinweise zur Benutzung:

- adr = Hexzahl oder vereinbarte Marke
- ein Programmende ist eine Adresse, ab der mind. viermal der Speicherinhalt 'FF' steht
- Verschieben von Progr.teilen nur in mit '00' oder 'FF' belegten Speicher moeglich (bei M+E beachten)
- es ist deshalb guenstig, den Speicher mit 'FF' zu initialisieren !

Nach dem Start des OCE auf der Adresse 4000H kann die Adresse der Markentabelle eingegeben werden. Es muessen davor 20 Speicherplaetze freier Speicher sein fuer Merkworte des OCE.  
 Danach befindet sich der OCE im Anzeigemodus.

+ Markentabelle

- Ende der Tabelle : '00'
- max. 127 Marken
- jede Vereinbarung und jeder Zugriff braucht 5 Byte
- ein Ueberlaufen wird nicht verhindert

+ Anzeigemodus

- Programmanfaenge und markante Programmstellen mit Globalmarken versehen ! (3.Zeichen = Buchstabe)
- Kommandos im Anzeigemodus :

Taste      Funktion

```

-----
K   direkte Maschinencodееingabe
T   Uebergang zum Testmodus ab Anzeigeadr.
H   Vereinbarung der Anzeigeadr. als Haltepunkt
R   Uebergang zum RAM-Modus
Q   Uebergang zum Quellcodееingabemodus
DOWN naechster Befehl
UP   vorstehender Befehl
N    32 x DOWN
Z    32 x UP ( 'UP' und 'Z' nur bei vorhandener Globalmarke
      bis 256 Byte vor akt. Adresse )
+/-  Anzeigeadr. incrementieren
-/-  Anzeigeadr. decrementieren
A    Anzeige ab vorstehender Globalmarke
G    G CR Anzeige ab Programmende
     G 0 CR Anzeige ab der mit EXT vereinbarten Marke
     G adr CR Anzeige ab adr
B    Bindekommando (ersetzt symb. Adr. durch hexadez.)
     erzeugt lauffaehiges Programm
     B CR Binden auf Anzeigeadr.
     B adr CR
  
```

M Verschieben von Programmen mit Marken  
M adr CR CR Verschieben von adr bis Programmende auf Bereich  
ab Anzeigeadr.  
M adr1 CR adr2 CR von adr1 bis ausschliesslich adr2 auf Be-  
reich ab Anzeigeadr.  
V Verschieben nur der Marken (Markenumrechnung)  
V adr CR von adr bis Progr.ende auf Anzeigebereich  
V adr1 CR adr2 CR von adr1 bis ausschl. adr2 auf Anzeigeadr.  
E Eingabebereich schaffen  
E CR 1 Byte  
E adr CR ab akt. Adr. bis ausschl. adr  
das folgende Programm wird entsprechend verschoben  
S Streichen von Befehlen  
S CR 1 Befehl streichen  
S adr CR von Anzeigeadr. bis adr streichen  
L Anzeige ab Programmzaehler vor dem letzten Schritt im Testmode  
O Quellprogrammausgabe  
O adr CR Programm ab adr als Quelle in RAM geschrieben  
I Quellprogrammeingabe  
I adr CR Quellprogramm ab adr wird in Anzeigebereich ueber-  
nommen (gl. Format wie oben)  
P Reassemblerliste drucken  
akt. Zeile mit '\*' gekennzeichnet  
Unterbrechung 'S, Fortsetzung 'Q  
bei Ende wird Seitenvorschub erwartet - F  
P adr CR : adr= Endadr. abs. oder symbolisch  
P CR bis Progr.ende

+ Quellcodeeingabemodus

- Zahlen dez. oder hex. mit Ziffer beginnend  
(z.B. : 0BFFEH )
- Marke 3 Zeichen, erstes = Buchstabe
- alte Marken ueberschreibbar
- keine Marke: Tabulator oder 1 Leerzeichen
- nach Marke kann Offset bis +127/-128 stehen
- Op.code, Operanden nur durch Leerzeichen getrennt
- Ueberschreiben von Quellcode vermeiden! Vorsicht!
- kein Kommentar!
- Korrekturen: mit S Befehl streichen  
mit E Freiraum schaffen  
Eingabe neuer Befehl
- Fehlermeldungen:  
F Operations- oder Operandenfehler  
M Marke schon vorhanden  
O Offset zu gross
- Uebernahme mit CR, nochmaliges CR naechster Befehl, jede  
andere Taste - Anzeigemodus
- Pseudooperationen:  
DB n DW nn  
DM "text" MAR:EXT adr -ext. Marken deklarieren
- # Speicherplatzzaehler

+ Binden

- mit 'B' auf waehlbare adr oder nach 'T' auf die Anzeigeadr.
- nur das erstellte Programm als gesamtes (1 Schritt)
- mit EXT vereinbarte Marken werden nicht verschoben
- nach Fehlermeldung mit CR weitere Fehler angezeigt  
( Bindeoperation unterbrochen!)
- MTB beim Binden nicht veraendert!

+ RAM-Modus

- Funktionen des RAM-Modus:

- \* Anzeige eines bel. Speicherbereichs in HEX oder ASCII
- \* Speichereingabe
- \* Kopieren eines Bereichs in anderen
- \* Vergleich
- \* Suchen nach Bytes
- \* Speicherbereich ausdrucken

- Kommandos:

Taste      Funktion

-----	
H	Speicher in HEX anz.
S	"      " ASCII "
-	+1
-	-1
DOWN	+8
UP	-8
N	naechste Seite
Z	1 Seite zurueck
G	direkte Positionierung
CR	uebernehmen Eingabefeld in akt. Pos. (HEX)
T	ASCII-Code des naechsten ueber Tast. eingeg. Zeichens in Eingabefeld uebernehmen
I	Fuellen eines Bereiches
	I adr CR ab akt. Adr. bis ausschl. adr
M	Kopieren eines Bereichs
	M adr1 CR adr2 CR -von adr1 bis ausschl. adr2 in akt. Ber.
V	Vergleichen
	V adr1 CR adr2 CR -Anzahl der Differenzen angezeigt
	Kursor auf ersten Unterschied
	Fortsetzung: CR Wert aus Eingabefeld (Vergl.bereich)
	wird auf akt. Adr. uebernommen
	V Fortsetzung Vergleich
	BREAK Abbruch
L	Suchen einer Datenfolge
	L adr1 CR adr2 CR Datenfolge ab adr1 bis ausschl. adr2
	wird im Speicher gesucht
	wenn gefunden, auf akt. adr:
	CR Suche fortsetzen
	BREAK Abbruch, Bereich vor Suche angezeigt
P	Ausdruck Speicherbelegung
	P adr CR -anlog P im Anzeigem.

+ Testmodus

- T Abarbeitung akt. Befehl und protokollieren  
normal naechster (vom Progr. auszuf.) Befehl, wenn zuvor  
im A-Modus Haltepkt. festgelegt, dann dieser
  - N Abarbeitung akt. Befehl und Protokoll, naechster im  
Speicher stehender Befehl
  - L direktes Abarbeiten (Echtzeit)  
0FFH wirkt als Softwarehaltepunkt, naechster Bef. protok.
  - R Aendern Registerinh. mit CR bestaetigt  
BREAK nicht uebernommen
- Achtung: bei 'T' und 'N' I-Reg. nicht veraendert, IM2

+ Tastencodes: CR/0D,BREAK/03,DOWN/0A,UP/0B,LEFT/08,RIGHT/09  
(INS/05,DEL/13,TAB/17)

+ Drucker: eigene Routine enthalten, andere ueber Sprungverteiler  
auf 5D43H/5D44H; Uebergabe in A (JMP FFE8H)

letzter Bearbeiter:

Uwe Gast  
Ringfurther Weg 5  
Magdeburg  
3035

\*\*\*\*\*  
**C H E S S M A S T E R Z 1013**  
 \*\*\*\*\*

Programmstandort: 0B0H - 3400H  
 Headersave-Laden: @L, danach wird Progr. von 100H-3490H geladen;  
                   nach einem Autostart auf 3450H wird Programm  
                   nach 0B0H geladen und gestartet  
 Headersave-Saven: nur durch vorhergehendes Autostartverbot und @S::  
 Programmrestart: J 2800 oder @S

Kommandos:  
 -----

NE (NEW)           Aufstellung der Figuren  
 LE (LEVEL)       Spielstaerke 0 - 7  
 RA (RANDOMIZE)   Zufallsgenerator 0 - 4  
 RE (REFERE)      Rechner ist Schiedsrichter  
 SE (SELFPLAY)   Rechner spielt gegen sich selbst  
                   mit Quittung durch ENTER  
 AU (AUTO)       Rechner spielt automatisch gegen  
                   sich selbst  
 CO (COLOR)       Doppelfunktion:  
                   1.Farbwahl: Weiss - Rechner  
                               Schwarz - Spieler  
                   Rechner gibt weissen Zug vor,  
                   angenommen wird mit SPACE,ENTER  
                   nach NE hat der Rechner schwarz  
                   2. Zugvorschlag durch Rechner  
                   nach CO Zugvorschlag fuer eigene Partei -> nur mit  
                   ENTER quittieren -> vorgeschl. Zug kann durch dessen  
                   Eingabe ausgefuehrt werden  
  
 BO (BORD)        Eingabemodus fuer Schachprobleme  
                   z.B. KE4 = weisser Koenig auf E4  
                   -KE4 = schwaz. Koenig auf E4  
                   Uebergang in Spielmodus BO,SPACE,ENTER

Beispiele:  
 -----

NE,ENT           Aufstellen der Figuren  
 LE,ENT,4        Spielstaerke 4  
 RA,ENT,2        Zufallsgenerator 2  
 E2E6            Zug des weissen Spielers  
                   Rechner zeigt seinen Zug durch  
                   Zeichen im Anfangs- und Endfeld an  
                   Quittieren mit ENTER  
                   Kennton ueber TB-Geraet (Aufn.)  
 E1G1            Weiss - kurze Rochade  
 E1C1            Weiss - lange Rochade

- \* Falsche Zuege oder Nichtbeachten, dass Koenig im SCHACH ist,  
 wird durch "\*\*\*\*" angezeigt
- \* Abbruch des Spieles jederzeit mit "NE" moeglich
- \* PATT und MATT werden kurzzeitig angezeigt (mit Kennton)
- \* Ab Spielstaerke 2 koennen die Zuege bereits recht lange  
 dauern, deshalb TB auf Aufnahme, damit Rechner seinen Zug  
 mit Kennton melden kann
- \* Vor Programmstart ist es ratsam, fuer eine bessere Zugeingabe,  
 die Tastatur auf HEX-MOD zu schalten

(c) by N.H. & U.G.  
 Mai 1987

Die angepasste Version fuer den Z-1013 ist gegen einige Aenderungen geschuetzt, dieser Schutz beeintraehtigt die Arbeit mit dem Compiler in keiner Weise und braucht nicht entfernt werden. Als Kassetteninterface ist fuer diese Version der >HEADER< von Rainer Brosig ueber Sprungverteiler vorgesehen. Damit eroeffnen sich fuer den Anwender folgende Moeglichkeiten:

- Beim Letzteren wird der Paseditor und der Compiler zerstört ( wenn 'Y' eingegeben wird ) und es erfolgt ein Linken, anschliessend wird die SAVE-Routine angesprochen und das MC-Programm kann aufgenommen werden.Nach dem Absaven wird das MC-Programm automatisch gestartet.Ein aeusserer Start ist auf Adresse 03F9 H jederzeit moeglich. Zu Testzwecken waehrend der Erstellung ist ein Probelauf nach dem Compilieren moeglich, dazu ist die Frage 'Run?' mit 'Y' zu beantworten. Dieser Probelauf zerstört NICHT den Compiler!Das Kommando 'R'wiederholt den Start. STEUERKOMMANDOS:

```
(...)      Parameterangabe Pflicht!  
Iz1,i1    Insert ab Zeile mit Increment (*wie AUTO *)  
Lz1,z2    Listet Quelle ab Zeile1 bis Zeile2  
K(x1)     Anzahl der Zeilen bei List (*Siehe LINES*)  
Cz1,z2    Compile Zeile1 bis Zeile2  
D(z1,z2)  Delete Zeile1 bis Zeile2  
M(z1,z2)  Move Kopiert Zeile1 auf Zeile2  
N(z1,i1)  Renumber 1.Zeile,Increment  
Fz1,z2,$1,$2  Finde von Zeile1 bis2 $1 und gehe in EDIT  
Ez1       Edit Zeile1  
'SPACE'   Pointer rechts  
'<-'      Pointer links  
'->'      Tabulator  
'ENTER'   Abschluss Mode  
Q         Quit := Zustand vor EDIT  
R         Reload := Zustand vor Aenderung in EDIT  
L         List Zeile  
K         Kill aktuelles Zeichen  
Z         Kill bis Zeilenende  
F         Weitersuchen(siehe 'F'-Kdo)  
S         Substitute $1 gegen $2 (siehe 'F'-Kdo.)  
I         Insert bis 'ENTER'  
C         Change bis 'ENTER' (ueberschreiben)  
X         Pointer an das Ende setzen und Insert-Mode  
P(z1,z2)  Put (Text auf Band schreiben)  
G         Get (Text von Band lesen)  
Cz1       Compile ab Zeile1  
R         Run Probelauf  
T         :=MC Programm Adressen umrechnen+linken
```



```

B      Bye Systemaustritt
X      Ausgabe Anfangsadresse Textpuffer
W      Ausgabe der Quelle auf Drucker

```

```

*****
**                                     **
**      F E H L E R M E L D U N G E N   C O M P I L E R      **
**                                     **
*****

```

```

1      ZAHL ZUGROSS
2      ;ERWARTET
3      UNDEF. NAME
4      NAME ERWARTET
5      BEI CONST = UND NICHT :=
6      =ERWARTET
7      EINE ANWEISUNG DARF NICHT MIT DIESEM NAMEN BEGINNEN
8      :=ERWARTET
9      )ERWARTET
10     TYPFehler
11     .ERWARTET
12     FACTOR ERWARTET
13     CONSTANT ERWARTET
14     NAME IST KEINE CONSTANCE
15     THEN ERWARTET
16     DO ERWARTET
17     TO ODER DOWNT0 ERWARTET
18     ( ERWARTET
19     AUSDRUCK TYP KANN NICHT GESCHRIEBEN WERDEN
20     OF ERWARTET
21     ,ERWARTET
22     :ERWARTET
23     PROGRAM ERWARTET
24     VARIABLE EWARTET,DA PARAMETER VARIABLEL IST
25     BEGIN ERWARTET
26     BEI READ VARIABLE ERWARTET
27     AUSDRUECKE DIESES TYPs KOENNEN NICHT VERGlichen WERDEN
28     INTEGER / REAL ERWARTET
29     VARIABLENTYP KANN NICHT GELESEN WERDEN
30     DIESER NAME IST KEIN TYP
31     BEI REAL IST EXPONENT VERLANGT
32     SCALARER AUSDRUCK , KEIN NUMERISCHER ERWARTET
33     LEERSTRINGS VERBOTEN,CHR(0) VERWENDEN
34     (/ERWARTET
35     /)ERWARTET
36     ARRAYINDEX MUSS SCALAR SEIN
37     ..ERWARTET
38     (/ ODER , ERWARTET IN ARRAY DECLARATION
39     OBERGRENZE<UNTERGRENZE!
40     ZU LANG! MAX.256 ELEMENTE
41     ERGEBNIS VOM TYP NAME ERWARTET
42     , ODER /) ERWARTET IN SET
43     .. / , / /) ERWARTET
44     PARAMETERTYP MUSS NAME SEIN
45     LEERES SET KANN NICHT FACTOR IN NICHT-ZUWEISUNGSANW.
        SEIN
46     SCALARTYP (+REELLE ZAHL) ERWARTET
47     SCALARTYP REELLE ZAHLEN ERWARTET
48     SETS NICHT VERTRAEGLICH
49     < AND > SETS NICHT DAMIT VERGLEICHEN
50     FORWARD,LABEL,CONST,VAR,TYPE BEGIN ERWARTET

```

```

51  HEXAZAHL ERWARTET
52  SETS KOENNEN NICHT GEPOKET WERDEN
53  ARRAY ZU GROSS
54  END ODER : IN RECORDDEFINITION FEHLT
55  FELDDNAME ERWARTET
56  NACH WITH VARIABLE ERWARTET
57  WITH VARIABLE MUSS RECORD TYP SEIN
58  FELDDNAME NICHT MIT WITH ANWEISUNG IN VERBINDUNG
59  NACH LABEL VORZEICHENLOSE INTEGERZAHL
60  NACH GOTO VORZEICHENLOSE INTEGERZAHL
61  MARKE IN FALSCHER PROGRAMMEBENE
62  MARKE NICHT VEREINBART
63  PARAMETER VON SIZE MUSS VARIABLE SEIN
64  AUF ZEIGER NUR GLEICHHEITSTEST ANGEWANDT
65  ZULAESSIGE WRITE-PARAMETER FUER INTEGER: E:M:H
66  $ ENTHAELT KEIN ZEILENENDEZEICHEN
67  PARAMETER VON NEW,MARK,REALISE MUSS ZEIGERVARIABLE SEIN
68  PARAMETER VON ADDR MUSS VARIABLE SEIN

```

Folgende Pascalbefehle sind implementiert:

```

=====
FRAC      EXP      LN      ARCTAN    TAN      COS
SIN       INP      OUT     SIZE      ADDR     INLINE
ENTIER    USER     RANDOM  INCH    HALT     EOLN
PAGE      SQRT     ROUND   TRUNC   MAXINT   SUCC
PRED      ORD      PEEK    POKE    RELEASE  MARK
NEW       TOUT     TIN     CHR    ODD      ABS
SQR       FALSE    TRUE    BOOLEAN CHAR    REAL
INTEGER   READLN   READ    WRITELN WRITE   PACKED
NIL       FORWARD  PROGRAM OR     OF      TO
DO        IF       SET     NOT     MOD     DIV
VAR       AND      FOR     END     GOTO    WITH
TYPE     CASE     ELSE    THEN   LABEL   CONST
ARRAY    UNTIL    WHILE   BEGIN  RECORD  DOWNTO
REPEAT   FUNCTION  PROCEDURE

```

STRINGS MUESSEN ALS ARRAY OF CHAR VEREINBART WERDEN

```

*****
**                                           **
**           COMPILERSTEUERUNG             **
**                                           **
*****

```

Die Compilersteuerung erfolgt unter Benutzung der Kommentarklam-  
mern (\*,\*). Der Compiler erkennt eine Zeichenkette als Steuerzei-  
chen an, wenn vor ihr das reservierte Symbol '\$' eingetragen ist.  
'+' schaltet Mode ein , '-' schaltet Mode aus.  
'+' und '-' werden jeweils nach dem Steuerbuchstaben gesetzt.  
Zum Beispiel schaltet (\*\$L+\*) auf vollstaendiges Listing  
Folgende Buchstaben sind zulaessig:

	(+)	(-)
L(+)	VOLLST.LISTING	NUR FEHLERZEILEN
O(+)	OVERFLOWTEST BEI INTEGER(+/-)	ENTFAELLT (SCHNELLER)
C(+)	TASTATURABFRAGE BEI TEST	KEIN ENDLOSSCHLEIFENTEST
A(+)	TEST FELDDINDIZESGROESSE	TEST ENTFAELLT
I(-)	BEI INTEGER <>= RICHTIG AUCH BEI OVERFLOW	KOREKTUR WIRD NICHT DURCHGEFUEHRT
P.	DRUCKER EIN/AUS (ohne +/-)	

letzter Bearbeiter:

Nicolas Hildebrandt  
W.-Kuelz-Str. 34  
Weimar  
5300

\*\*\*\*\*  
**Beschreibung zum Headerprint 1.1 (C) by RB**  
 \*\*\*\*\*

Das Headerprint-Programm dient der automatischen Erfassung von Kassetteneinhalten, die mit dem Headersave-Programm aufgezeichnet wurden, auf einen Drucker.

Standort: 100H - 37FH Start: 100H

Anpassung: 103H: JMP DRDEL ;Ruecksetzen logischer Druckertreiber  
 106H: JMP DRAKK ;Uebergabe Zeichen (Akku) an Druckertr.

Nach dem Start des Programms wird abgefragt, ob ein neues Verzeichnis angelegt werden soll. Wenn ja, wird die Kassettenseite (A,B o. 1,2) sowie der Name der Kassette abgefragt (beliebiger String). Diese Angaben werden danach als Kopf ausgedruckt. Auf die Aufforderung START TAPE kann die Kassette entweder von vorn oder ab der aktuellen zu protokollierenden Position gestartet werden. Nun wird immer dann, wenn ein Programmkopf gelesen wird, der Inhalt des Kopfes als ASCII-String zeichenweise an den Druckertreiber uebergeben. Bei sehr langsamen Druckern ist zu beachten, dass der naechste Kopf erst dann gelesen werden kann, wenn der vorherige Kopf fertig gedruckt ist (kann vor allem bei sehr kurzen Programmen vorkommen). Der Abbruch des Programms kann, so lange ein Signal anliegt, mit BREAK (S4-K) erfolgen, ansonsten mit RESET.  
 Nach dem Start des Programms wird ein Programm zum Ruecksetzen des Druckertreibers angesprungen (kann auch zum Initialisieren benutzt werden). Wird ein solches Programm nicht benoetigt, kann der entspr. Sprung auf 103H mit RET (C9H) kurzgeschlossen werden.

Zweckdienliche Hinweise bitte an Rainer Brosig, W.-Florin-Str. 2c COSWIG, 8270.

## Korrekturen zur Programmkassette M0191 fuer Z1013

(sind blau eingearbeitet, außerdem gibt es Patche für Headersave und Pascal. Diese sind in der aktualisierten Kassette bereits erfolgt).

---

## Beiblatt zur Kassette

IG HC des Informatikzentrums  
der TU Dresden

Dresden, 10.12.1987

Liebe Teilnehmer unserer 1. Z1013-Tagung vom 5.12.87!

-----  
Bei der Herstellung unserer Tagungskassette "M0191" ist leider ein technischer Fehler aufgetreten, den wir erst nach dem Verkauf festgestellt haben. Wir bitten Euch vielmals um Entschuldigung. Die Kassett läßt sich nicht einwandfrei in den Z1013 einlesen. Wir möchten Euch nun wie folgt helfen. Ihr habt zwei Möglichkeiten die Programme zu erhalten:

1. Schickt die Originalkassette an Freund Steglich, er wird eine neu bespielte Kassette zurückschicken!
2. Wenn Ihr die Kassette behalten wollt, dann schickt bitte das Einlegeblatt Eurer Kassette mit Eurer Adresse versehen ab. Legt bitte 10,-M (Materialwert für die neue Kassette) in den Umschlag. Eine Postanweisung ist ebenfalls möglich.

Ohne den Nachweis für die gekaufte Kassette kann aus Kapazitätsgründen keine Bearbeitung erfolgen!

Reklamationsbearbeitung:      Klaus Steglich  
                                 Karl Str. 10  
                                 Hohen Neuendorf  
                                 8426

gez. Steglich/Brosig