MicroWORD

„MicroWORD 1.5“ ist ein extrem schneller, vom Z1013 adaptierter Texteditor.

© by R. Brosig 1988, IG-HC Dresden
Version Z9001: V. Pohlers, April 2020



Download

- Paket

Beschreibung

„MicroWORD 1.5“ ist ein Textsystem, welches sich durch Einfachheit, Kompaktheit und hohe Arbeitsgeschwindigkeit auszeichnet. Mit 23 Kommandos sowie 21 Editierfunktionen werden alle für ein Textsystem wichtigen Funktionen ermöglicht. Durch die automatische Steuerzeichenkonvertierung können Texte verschiedener Herkunft bearbeitet werden.

1. Speicheraufteilung

```
 300H-     Programm
           Arbeitszellen
 ab 1800H  Textpuffer
von EOR 1K SAVE-Bereich
```

Ohne Beschreibung 100H-17FFh speichern

2. Der Textspeicher

Als Textspeicher dient der Speicherbereich ab 17FFh. Bei 64K-RAM stehen somit max. 42000

Textzeichen zur Verfügung. Wird der SAVE-Bereich (Merkzettel zur Textmischung) mit als Textpuffer genutzt, sind sogar 1000 Zeichen mehr nutzbar. Die genaue Lage des Textpuffers kann durch das A-Kommando bei Bedarf entsprechend eingeschränkt werden. Dabei kann die Definition nur im Bereich von 1747h bis Speicherobergrenze erfolgen (max. 0BFFFH). Wird die Obergrenze des Textspeicherbereiches überschritten, wird durch eine Einfügung eines Zeichens 03H der überstehende Text abgeschnitten. Während der Bearbeitung wird der Text an Cursorposition in zwei Teile geteilt, wobei der hintere Teil an die Textpufferobergrenze geschoben wird. Ein Austritt aus dem Editor mit Reset oder dgl. bewirkt, dass der Text nicht wieder zusammengefügt wird. Durch einen sofortigen Neustart wird der Text automatisch wieder reorganisiert.

Achtung: Der Editor benötigt den gesamten definierten Textpufferbereich, auch wenn der zu bearbeitende Text kleiner ist.

3. Allgemeine Funktionen

Nach Start des Texteditors wird ein Speichertest durchgeführt, der einen evtl. grösser def. Textpuffer auf das Speicherende begrenzt (Anzeige 'M'). Danach wird das Ende eines evtl. im Textpuffer liegenden Textes gesucht (Anzeige 'I'). Befinden sich darin CRLF-Steuerzeichen, so werden diese in NL gewandelt. CR-Steuerzeichen allein werden nur dann in NL gewandelt, wenn sich keine NL's im Text befinden. Nach dem Ende der Initialisierungsphase wird ein Kommandomodus erreicht. Dieser zeigt

- das Menue der möglichen Kommandos,
- die Zeile, in der sich der Cursor befindet <LINE>,
- die Cursorposition innerhalb der aktuellen Zeile <POS>,
- den freien Platz im Textpuffer dezimal <FREE>,
- die aktuelle Cursor-Textposition in hex <TXP>,
- das Textpufferende in hex <ETB>,
- das definierte Zeilenende (max. 255) dezimal <LE>,
- den eingeschalteten Editiermode (Insert'I'/Overwrite'O')<M>
- sowie den Hexcode des durch den Cursor im Bildmode vedeckten Zeichens an <CD>.

Für den Fall, dass noch kein Text im Textpuffer liegt, empfiehlt es sich, den Textpuffer als erstes komplett zu löschen <&>. Der SAVE (Merkzettel) wird beim Start automatisch gelöscht.

4. Der Kommandomodus

Der Kommandomodus ist durch das Kommandomenue gekennzeichnet. Durch Eingabe des entspr. großen oder kleinen Buchstabens wird das Kommando ausgeführt. nach Ausführung des Kommandos wird immer der Bildschirmmodus erreicht. Alle Zeicheneingaben, die kein Kommando aufrufen, führen zum Abschluss des Kommandomodus und damit in den Bildschirmmodus. Folgende Kommandos können verwendet werden:

<A> - assign
Damit kann der Textpufferbereich neu definiert werden. Dadurch ist es auch möglich, mehrere Texte im Speicher zu halten. Es ist zu beachten, dass der Text auf der Adresse Textpufferanfang+1 beginnt, da auf der 1. Pufferzelle das STX-Zeichen (02) eingetragen wird. Die Eingabe der Adressen kann dezimal oder hexadez. (nachstellen eines H, ohne führende 0) erfolgen.

<B> - bye

Rückkehr zum Monitor und Reorganisation des Textes.

<S> - start
Cursor wird auf Textanfang gesetzt.

<E> - end
Cursor wird auf Textende gesetzt.

<N> - number
Cursor wird nach Eingabe der Zeilennummer mit maximal fünf Ziffern auf die entsprechende Zeile gesetzt.

<M> - set mark
Der Cursor wird im Bildschirmmodus vor Anwendung des Kommandos auf eine bestimmte Textstelle positioniert, anschließend wird mit diesem Kommando diese Stelle markiert und wieder in den Bildschirmmodus zurückgeschaltet. Die Markierung wird durch andere Kommandos ausgewertet. Diese Funktion kann alternativ auch durch eine entsprechende Steuertaste (**^V**) im Bildschirmmode ausgeführt werden.

<%> - delete part
Mit diesem Kommando kann ein Textteil, dessen Anfang mit dem <M>-Komm. gekennzeichnet wurde, bis zur aktuellen Cursorposition gelöscht werden. Mögliche Fehler sind das Fehlen der ersten Markierung bzw. die Markierung liegt hinter der Cursorposition.

<&> - all clear
Löschen des gesamten Textpuffers. Der Textpuffer wird dazu nicht physisch gelöscht, sondern es wird am Anfang des Textpuffers das ETX-Zeichen eingetragen.

<F> - find/change (ab Cursorp.)
Soll eine Textfolge im Textpuffer gesucht werden und ggf. durch eine andere Textfolge ersetzt werden, so muss nach Aufruf dieses Kommandos der Text, der gesucht werden soll, eingegeben werden, durch '/' getrennt kann anschließend der einzusetzende Text angegeben werden. Die gesamte Textfolge ist auf 32 Zeichen begrenzt. Dabei kann das Verhältnis der beiden Textteile beliebig sein. Die Suche erfolgt dabei grundsätzlich ab aktueller Cursorpos. Bei der ersten Übereinstimmung wird der Cursor auf das erste Zeichen dieser Zeichenkette gesetzt und in den Bildschirmmodus umgeschaltet. Im Bildschirmmodus existiert eine Funktionstaste, die diesen Kommandokomplex ohne erneute Eingabe der Zeichenkette wiederholt startet (CTRL-Q =Ersetzen/Wiederholen). Die Find-Routine startet immer auf der aktuellen Zeilenposition (ggf. vorher Kommando S ausführen).
Ist die gefundene Zeichenfolge nicht die gewünschte, so kann mit Hilfe der Cursorfunktionen die Zeichenkette übergangen werden.

<H> - print part
Über die Druckerschnittstelle kann mit diesem Kommando der Text zwischen einer im <M>Kommando markierten Stelle und der aktuellen Cursorposition formatiert ausgedruckt werden. Es besteht die Möglichkeit der Einstellung des Seitenformates und der Auswahl der zu druckenden Seiten. Eine Arbeit mit Seitenstop kann eingestellt werden. Nach Aufruf des Kommandos erfolgt die Aufforderung zur Eingabe der Zeilenzahl/Sei- te (lines/page), die man mit einer Dezimalzahl (>9) beantwortet (normale Seite=72). Bei Zeilenzahl=0 bzw. keine Eingabe erfolgt keine Seitenformatierung. Bei geforderter Seitenformatierung wird abgefragt, mit welcher Nummerierung die gedruckten Seiten beginnen sollen. Bei leerer Eingabe wird ohne Fußzeile, in der die Seitennummer untergebracht ist, gedruckt. Anschließend wird abgefragt, ob ein Druckstop nach jeder

Seite erfolgen soll. Wird diese Frage bejaht, stoppt der Druck definiert nach jeder Seite. Im anderen Fall wird der Text bis zum Ende ohne Pause gedruckt.

<P> - print file
wie <H>, nur dass der komplette Text gedruckt wird.

<L> - line-end
Mit diesem Kommando wird die rechte Zeilenbegrenzung eingestellt (10-255). 6 Pos. vor dieser Begrenzung erfolgt eine Randwarnung, bei Überschreiten der Pos. erfolgt ein Wortumbruch bis zum vorhergehenden Leerzeichen oder TAB. Wird innerhalb der letzten 6 Zeichen ein '-' eingegeben, so erfolgt automatisch ein Zeilenabschluss. Die Eingabe einer 0 als max. Zeilenlänge schaltet die Funktion aus.

<R> - read file
Es wird ein Text von Kassette in den Textpuffer geladen. Ein evtl. vorhandener Text wird überschrieben.

<W> - write file
Der gesamte Text, der sich im Textpuffer befindet, wird auf Kassette ausgegeben.

<V> - write part
Ein vorher im <M>-Kommando markierter Textteil wird bis zur aktuellen Cursorposition auf Kassette ausgegeben.

<D> - read part
Der von Kassette eingelesene Text wird an der Cursorposition eingefügt. Dabei ist darauf zu achten, dass der eingefügte Text kleiner als der freie Textbereich ist. Es ist wichtig, dass der Einlesevorgang nicht vorzeitig abgebrochen wird, da es sonst zu einem Textverlust kommen kann.

<X> - expand
Hiermit wird der zweizeilige Bildmode eingeschaltet. Damit können Texte, die für ein breiteres Format geschrieben sind, ohne Bildrollen gelesen und bearbeitet werden.

<C> - compact
Der zweizeilige Bildmode wird wieder ausgeschaltet.

<T> - write repeat
Die vorangegangene Write-Funktion wird wiederholt (Sicherheitskopie erzeugen). Zwischen der ersten Schreiboperation und diesem Kommando darf die aktuelle Cursorposition nicht verändert werden.

<=> - NL→CRLF
Wandelt über den gesamten Text das Zeilenendezeichen NL (1EH) in die Steuerzeichenfolge CRLF (0DH,0AH). Anschließend wird das Kommando <W> aufgerufen, sodass der gewandelte Text auf Kassette ausgelagert werden kann. Nach Beendigung oder Abbruch der Save-Funktion wird der Text wieder zurückgewandelt. Es ist zu beachten, dass der Text durch die Wandlung in CRLF um die Zeilenzahl länger wird und deshalb der freie Speicher- platz diesem Fakt entsprechen muss. Andernfalls erfolgt eine Fehlermeldung.

<-> - NL→CR
Wandelt über den gesamten Text das Zeilenendezeichen NL (1EH) in das Steuerzeichen CR (0DH). Anschließend wird das Kommando <W> aufgerufen, sodass der gewandelte Text auf Kassette ausgelagert werden kann. Nach Beendigung oder Abbruch der Save-Funktion wird der Text wieder

zurückgewandelt.
Dieses Kommando ist speziell für Assember-Quelltexte des Z9001 gedacht. EDIT kodiert das Zeilenende nur mit 0Dh.

<I> - in save
Dieses sehr leistungsfähige Kommando dient im Zusammenhang mit dem <O> Kommando dem Mischen von Textteilen aus dem gleichen Textpuffer, oder aus einem anderen. Ein Textbereich von einer definierten Marke bis zur Cursorposition wird in den SAVE geladen. Der SAVE stellt einen Sondertextpuffer von 1K Länge dar, der direkt hinter dem Editor platziert ist. Ist der markierte Bereich grösser als der SAVE, erfolgt eine Fehlermeldung, der Text wird nicht in den SAVE übernommen. Sollte der SAVE- Puffer vom normalen Textpuffer mitbenutzt werden müssen, so erfolgt eine entsprechende Fehlermeldung, wenn trotzdem eine SAVE/Funktion ausgeführt werden soll.

<O> - out save
Der Inhalt des im <I> Kommando gefüllten SAVE's wird an der aktuellen Cursorposition eingefügt. Dabei erfolgt eine Kon- trolle über den noch verfügbaren Speicherplatz.

5. Der Bildschirmmodus

Im Bildschirmmodus wird der Bildschirm entsprechend des eingegebenen Textes aktualisiert. Das Textfenster wird dabei so dargestellt, dass der Cursor im Bildfenster erscheint. Dabei kann an den Rändern Text abgeschnitten sein, der nicht mehr auf den Bildschirm passt. Alle Editierfunktionen werden durch entsprechende Tastatureingaben ausgelöst.
Das Programmsystem enthält dazu eine Tastaturkonvertierungsroutine, die es ermöglicht, jeden beliebigen Tastaturcode als Funktionscode festzulegen. Im Folgenden werden die entsprechenden Speicherzellen, in denen der gewünschte Tastaturcode eingetragen werden kann, aufgelistet:

```
ab Adr. CODTB 32Ch

Code| CTRL  | Spezial | Bedeutung
_____________________________________
 09H  CTRL-I   -->      Cursor rechts
 08H  CTRL-H   <--      Cursor links
 0AH  CTRL-J   cu down  Cursor runter
 0BH  CTRL-K   cu up    Cursor hoch
 01H  CTRL-A            Cursor eine Seite vor
 05H  CTRL-E            Cursor Seite zurueck
 13H  CTRL-S   PAUSE    Zeichen links loeschen
 1FH  DEL      DEL      Zeichen loeschen
 03H  CTRL-C   STOP     Bild-/Komm.-Mode
 1BH  ESC      ESC      Ank. Steuerzeichen
 12H  CTRL-R            Wandlung Buchstabe
 15H  CTRL-U            CAPS (Tausch gr/kl))
 0FH  CTRL-O            Loeschen einer Zeile
 11H  CTRL-Q            Ersetze/Wiederhole
 0CH  CTRL-L            FORM FEED
 0DH  CTRL-M   ENTER    ENTER
 14H  CTRL-T            Tabulator
 06H  CTRL-F            Tabulator
 0EH  CTRL-N            NUL
```

```
 1AH  CTRL-S   INS      Einfuegen/Ueberschreiben
 16H  CTRL-V            Marke setzen
 18H  CTRL-X   TAB -->| Cursor Wort rechts
 19H  CTRL-Y   TAB |<-- Cursor Wort links
 1CH  LIST     LIST     Reclaim,Zeichen zurueckholen
 0                      Reserve
```

Neben den von der Version 1.3 her bekannten Funktionen wurden 5 neue Funktionen eingeführt:

1. Marke setzen; damit kann im Bildmode eine Marke gesetzt werden ohne diesen zu verlassen.

2./3. Cursor Wort links/rechts; dadurch kann man schnell Positionen in einer Zeile anfahren. Weiterhin hilft diese Funktion, wenn z.B. das letzte Wort gelöscht werden soll, um den Cursor ein Wort zurückzusetzen.

4. Reclaim; diese Sonderfunktion hängt eng mit der gesamten Arbeitsweise des Editors zusammen. Mit dieser Taste ist es möglich, ein versehentlich überschriebenes oder gelöschtes Zeichen/Wort/Textteil zurückzuholen, wenn dazwischen keine Cursorfunktionen oder andere Manipulationen des Textes erfolgten.

5. Insert/Overwrite, in Abhängigkeit dieser Einstellung, die in der Statuszeile auch angezeigt wird, erfolgt die Eingabe neuen Textes entweder indem der alte Text überschrieben (O) oder eingefügt (I) wird.

Steuerzeichen, die von der Tastatur ohne Ankündigung angesprochen werden, aber durch das Programmsystem nicht bearbeitet werden, erzeugen eine Fehlerausschrift. Sollen Steuerzeichen in den Text eingefügt werden, so sind diese jeweils mit der Ankündigungstaste anzumelden (entspr. Kodierungstabelle z.B. ESC). Will man z.B. die Folge 1BH 2DH 01H (Unterstreichung ein für einen EPSON-kompatiblen Drucker) eingeben, so ist folgende Tastenfolge notwendig:

```
ESC,ESC -> 1BH
-       -> 2DH
ESC,^A  -> 01H
```

(die Ausschaltung der Unterstreichungsfunktion beim Drucker er- folgt übrigens durch die Folge 1BH 2DH 00H –> ESC,ESC,-,^N). Man erkennt, dass vor j e d e m Steuerzeichen die Ankündigungstaste gedrückt werden muss. Vorsicht ist bei der Einfügung des ETX-Zeichens (03H) geboten. Bei einem neuen Start des Editors wird dieses Zeichen als Textende erkannt, und der dahinterliegende Text abgeschnitten. Muss aus irgendwelchen Gründen mit der Kodierung 03H im Text gearbeitet werden, so empfiehlt es sich, das ETX-Zeichen umzudefinieren (Zelle 32AH). Folgende Steuerzeichen können bzw. müssen ohne Ankündigungstaste in den Text eingefügt werden:

TAB Kodierung 09H; Tabulatorstop jede 8. Spalte
NL Kodierung 1EH; Neue Zeile als Wagenrücklauf/Zeilenschaltung
FF Kodierung 0CH; Neue Seite (wird nur beim Druck ausgeführt, Anz. im Text als Grafikzeichen)
NUL Kodierung 0; schreibt eine physische 0 in den Text, da 0 von Tastatur aus nicht erzeugbar ist, aber für bestimmte Effekte (Drucker-St.-Zeichen) benötigt werden.

Die Steuerzeichen werden im Text als Grafikzeichen mit einem in Zelle OFFS 16BH zugeordneten Offset dargestellt. Sollen die Steuerzeichen dem Zeichensatz entsprechend dargestellt werden, so ist der Offset auf 0 zu setzen: 16BH→0 Ist man sich nicht sicher, welches Steuerzeichen tatsächlich im

Speicher steht, so kann der Cursor auf die entsprechende Stelle gefahren werden. In der Statuszeile erscheint dann der entsprechende Hexcode.
Weiterhin sind im Text Grafikzeichen zugelassen und eingebbar (außer FFH). Es ist aber zu bedenken, dass der Text da- durch nicht mehr dem ASCII-Format entspricht, und meist seine Druckfähigkeit verliert (außer IBM-Zeichensatz). Man sollte auch immer daran denken, dass zum Editieren die Grafikfunktion der Tastatur wieder ausgeschaltet werden muss.

Für die Cursordarstellung sind zwei Codes (je nach dem, ob sich unter dem Cursor ein Zeichen befindet) vorgesehen, die in Zelle 124H bzw 125H eingetragen werden können. Wird in Zelle 126H eine 80H eingetragen, so erfolgt die Cursordarstellung mit gesetztem 7.Bit (z.B. wenn im Zeichensatz statt den Grafikzeichen der inverse ASCII-Zeichensatz abgelegt ist→Cursordarstellung als inverses Zeichen).

6. Die Druckerschnittstelle

Es muss ein Druckertreiber geladen sein.

Da MicroWord den gesamten Speicher nutzt, muss EOR vor dem Druckertreiber stehen!

Beispiel

```
>V24     Laden der V24A1..A3-Treiber (wie im Druckermodul)
>EOR B6FF    Setzt EOR vor V24 (beginnt ab B800, 100h Byte f. System)
>V24A3   den direkten Druckertreiber init.
>WORD        MicroWord (re)starten
```

Sollte der verwendete Drucker kein NL (1EH) als Steuerzeichen verstehen, so ist dieses im phys. Treiber in CR-LF (0DH-0AH) aufzuloesen (im enthaltenen Druckertreiber realisiert).

Anmerkung zur Nutzung vom MicroWORD

Der erzeugte Quelltext ist kompatibel zu den meisten anderen Textsystemen und Assemblern wie z.B. Texteditor Scf, MTX oder Assembler Scf und ist rein ASCII-codiert. Durch die Steuerzeichenwandelmoeglichkeit NL→CRLF ist der Text auch fuer CPM-Sys- teme verwendbar, bzw. es lassen sich auch CPM-kompatible Texte bearbeiten (Es muss beachtet werden, dass das ETX-Zeichen im CPM nicht 03 (^C), sondern 1AH (^Z) ist → umdefinieren oder als Steuerzeichen in den Text schreiben).

7. Wichtige Adressen und Zellen zur Anpassung

(korrekte Adressen s. Listing)

```
-Cursordarstellung:

 Code1    :   424H  ->C7H
 Code2    :   425H  ->FFH
 mit 7.bit:   426H  ->0/80H
```

```
-Ende-/Anfangtextzeichen:

 ETX:  427H  ->03H
 STX:  428H  ->02H

-Textpuffer-Daten:

 TPANF: 45BH ->17FFH
 TPEND: 45DH ->DFFFH

-Bildschirm:

 BWS-Anfang : 45FH ->EC00H
 Spaltenzahl: 461H ->40
 Zeilenzahl : 463H ->24


-absolute Textbereichsgrenzen:

 untere Grenze: 467H ->174Fh
 obere  Grenze: 469H ->BFFFH


-Zeichenoffset bei Steuerzeichen:

 46BH -> 0A0H


Diese Zellen sind im allg. konstant, koennen aber bei Hardware-
aenderungen (z.B.groesserer Bildschirm) entspr. geaendert
werden.
```

Viel Spass mit „MicroWORD 1.5“
R. Brosig, IG-HC Dresden

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/z9001/software/microword?rev=1588230010

Last update: **2020/04/30 07:00**