MicroWord

Beschreibung zum Texteditor M i c r o W O R D 1.5

© by R. Brosig 13.09.88, IG-HC Dresden

„MicroWORD 1.5“ ist ein gegenueber V.1.3 stark erweitertes Textsystem, welches sich durch Einfachheit, Kompaktheit und hohe Arbeitsgeschwindigkeit auszeichnet. Mit 22 Kommandos sowie 21 Editierfunktionen werden alle fuer ein Textsystem wichtigen Funktionen ermoeglicht. Durch die automatische Steuerzeichenkonvertierung koennen Texte verschiedener Herkunft bearbeitet werden.

1. Speicheraufteilung

```
 100H-13ADH  Programm
13AEH-13F8H  Arbeitszellen
13F9H-1418H  Stringpuffer
1419H-1455H  Phys. V.24-Druckertreiber
1456H-17FEH  SAVE-Bereich
   ab 17FFH  Textpuffer
```

2. Der Textspeicher

Als Textspeicher dient der Speicherbereich ab 17FFH. Bei 64K-RAM stehen somit max. 54220 Textzeichen zur Verfuegung. Wird der SAVE-Bereich (Merkzettel zur Textmischung) mit als Textpuffer genutzt (Text ab 1456H), sind sogar 55116 Zeichen nutzbar.

Die genaue Lage des Textpuffers kann durch das A-Kommando bei Bedarf entsprechend eingeschraenkt werden. Dabei kann die Definition nur im Bereich von 1456H bis Speicherobergrenze erfolgen (max. EBFFH). Wird die Obergrenze des Textspeicherbereiches ueberschritten, wird durch eine Einfuegung eines Zeichens 03H der ueberstehende Text abgeschnitten.

Waehrend 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 zusammengefuegt wird. Durch einen sofortigen Neustart wird der Text automatisch wieder reorganisiert.

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

3. Allgemeine Funktionen

Nach Start des Texteditors wird ein Speichertest durchgefuehrt, der einen evtl. groesser 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 moeglichen Kommandos,
- die Zeile, in der sich der Cursor befindet (nach Start immer in der letzten)<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 verdeckten Zeichens <CD> an.

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

4. Der Kommandomodus

Der Kommandomodus ist durch das Kommandomenue gekennzeichnet. Durch Eingabe des entspr. grossen oder kleinen Buchstabens wird das Kommando ausgefuehrt. nach Ausfuehrung des Kommandos wird immer der Bildschirmmodus erreicht. Alle Zeicheneingaben, die kein Kommando aufrufen, fuehren zum Abschluss des Kommandomodus und damit in den Bildschirmmodus. Folgende Kommandos koennen verwendet werden:

<A> - assign

Damit kann der Textpufferbereich neu definiert werden. Dadurch ist es auch moeglich, 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 fuehrende 0) erfolgen.

<B> - bye

Rueckkehr 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 fuenf 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 zurueckgeschaltet. Die Markierung wird durch andere Kommandos ausgewertet. Diese Funktion kann alternativ auch durch eine entsprechende Steuertaste (^V) im Bildschirmmode ausgefuehrt werden.

<%> - delete part

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

<&> - all clear

Loeschen des gesamten Textpuffers.

<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 Verhaeltnis der beiden Textteile beliebig sein. Die Suche erfolgt dabei grundsaetzlich ab aktueller Cursorpos. Bei der ersten Uebereinstimmung wird der Curuor 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 / S4-A =Ersetzen/Wiederholen). Die Find- Routine startet immer auf der aktuellen Zeilenposition (ggf. vorher Kommando S ausfuehren. Ist die gefundene Zeichenfolge nicht die gewuenschte, so kann mit Hilfe der Cursorfunktionen die Zeichenkette uebergangen werden.

<H> - print part

Ueber 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 Moeglichkeit 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/Seite (lines/page),die man mit einer Dezimalzahl (>9) beantwortet. Dabei ist die Laenge der Seite in Zeilen gemeint und nicht die Anzahl der Textzeilen auf der Seite (normale Seite=72). Bei Zeilenzahl=0 bzw. keine Eingabe erfolgt keine Seitenformatierung.

Bei geforderter Seitenformatierung wird abgefragt, mit welcher Numerierung 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 <I>, nur dass der komplette Text gedruckt wird.

<L> - line-end

Mit diesem Kommando wird die rechte Zeilenbegrenzung eingestellt (10-255). 6 Pos. vor dieser Begr. erfolgt eine Randwarnung, bei Ueberschr. 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. Zeilenlaenge schaltet die Funktion aus.

<R> - read file

Es wird ein Text ueber die Headersaveschnittstelle in den Textpuffer geladen. Ein evtl. vorhandener Text wird ueberschrieben.

<W> - write file

Der gesamte Text, der sich im Textpuffer befindet, wird ueber die Headersaveschnittstelle ausgegeben.

<V> - write part

Ein vorher im <M>-Kommando markierter Textteil wird bis zur aktuellen Cursorposition ueber die Headersaveschnittstelle ausgegeben.

<D> - read part

Der ueber die Headersaveschnittstelle eingelesene Text wird an der Cursorposition eingefuegt. Dabei ist darauf zu achten, dass der eingefuegte 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 koennen Texte, die fuer 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 veraendert werden.

<=> - NL->CRLF

Wandelt ueber 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 zurueckgewandelt. Es ist zu beachten, dass der Text durch die Wandlung in CRLF um die Zeilenzahl laenger wird und deshalb der freie Speicherplatz diesem Fakt entsprechen muss. Andernfalls erfolgt eine Fehlermeldung.

<I> - in save

Dieses sehr leistungsfaehige 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 Laenge dar, der direkt hinter dem Editor plaziert ist. Ist der markierte Bereich groesser als der SAVE, erfolgt eine Fehlermeldung, der Text wird nicht in den SAVE uebernommen. Sollte der SAVE-Puffer vom normalen Textpuffer mitbenutzt werden muessen, so erfolgt eine entsprechende Fehlermeldung, wenn trotzdem eine SAVE- Funktion ausgefuehrt werden soll.

<O> - out save

Der Inhalt des im <I> Kommando gefuellten SAVE's wird an der aktuellen Cursorposition eingefuegt. Dabei erfolgt eine Kontrolle ueber den noch verfuegbaren 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 Raendern Text abgeschnitten sein, der nicht mehr auf den Bildschirm passt. Alle Editierfunktionen werden durch entsprechende Tastatureingaben ausgeloest. Das Programmsystem enthaelt dazu eine Tastaturkonvertierungsroutine, die es ermoeglicht, jeden beliebigen Tastaturcode als Funktionscode festzulegen. Im Folgenden werden die entsprechenden Speicherzellen, in denen der gewuenschte Tastaturcode eingetragen werden kann, aufgelistet:

```
Adr|Code|S4|^|   Bedeutung
HEX|(z.B)
_____________________________
129H 09H Q  I  Cursor rechts
12AH 08H P  H  Cursor links
12BH 0AH R  J  Cursor runter
12CH 0BH S  K  Cursor hoch
12DH 01H I  A  Cursor eine Seite vor
12EH 05H M  E  Cursor Seite zurueck
12FH 10H @  P  Zeichen links loeschen
130H 7FHS2GDEL Zeichen loeschen
131H 03H K  C  Bild-/Komm.-Mode
132H 1BH   ESC Ank. Steuerzeichen
133H 12H B  R  Wandlung Buchstabe
134H 15H E  U  CAPS (Tausch gr/kl))
135H 0FH W  O  Loeschen einer Zeile
136H 11H A  Q  Ersetze/Wiederhole
137H 0CH T  L  FORM FEED
138H 0DH U  M  ENTER
139H 14H D  T  Tabulator
13AH 06H N  F  Tabulator
13BH 0EH V  N  NUL
13CH 13H C  S  Einfuegen/Ueberschreiben
13DH 16H F  V  Marke setzen
13EH 18H    X  Cursor Wort rechts
13FH 19H    Y  Cursor Wort links
140H 1CH    <  Reclaim,Zeichen zurueckholen
141H 0         Reserve
```

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

- 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 geloescht werden soll, um den Cursor ein Wort zurueckzusetzen.

- 4. Reclaim, diese Sonderfunktion haengt eng mit der gesamten Arbeitsweise des Editors zusammen. Mit dieser Taste ist es moeglich, ein versehentlich ueberschriebenes oder geloeschtes Zeichen/Wort zurueckzuholen, wenn dazwischen keine Cursorfunktionen oder andere Manipulationen des Textes erfolgten.
- 5. Insert/Overwrite, in Abhaengigkeit dieser Einstellung, die in der Statuszeile auch angezeigt wird, erfolgt die Eingabe neuen Textes entweder indem der alte Text ueberschrieben (O) oder eingefuegt (I) wird.

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

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

(die Ausschaltung der Unterstreichungsfunktion beim Drucker erfolgt uebrigens durch die Folge 1BH 2DH 00H –> ESC,ESC,-,^N). Man erkennt, dass vor j e d e m Steuerzeichen die Ankuendigungstaste gedrueckt werden muss. Vorsicht ist bei der Einfuegung 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 Gruenden mit der Kodierung 03H im Text gearbeitet werden, so empfiehlt es sich, das ETX-Zeichen umzudefinieren (Zelle 127H). Folgende Steuerzeichen koennen bzw. muessen ohne Ankuendigungs taste in den Text eingefuegt werden:

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

Die Steuerzeichen werden im Text als Grafikzeichen mit einem in Zelle 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 tatsaechlich 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 dadurch nicht mehr dem ASCII-Format entspricht, und meist seine Druckfaehigkeit verliert. Man sollte auch immer daran denken, dass zum Editieren die Grafikfunktion der Tastatur wieder ausgeschaltet werden muss.

Fuer 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 koennen. 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

Der Anschluss eines Druckers mit V.24-Schnittstelle ist realisiert. Der Anschluss des Druckers erfolgt ueber das Z1013 E/A-Modul von RobotronRiesa.

Um andere phys. Druckertreiber nutzen zu koennen, kann auf der Adresse 112H der entsprechende Sprung eingetragen werden. Dort kann z.B. ein Sprung auf DRAKK des Sprungverteilers gelegt werden (falls kein externer log. Druckertreiber verwendet werden soll, kann auch ein Sprung auf ZEIDR erfolgen). Der angehaengte phys. V.24-Treiber kann dann natuerlich geloescht bzw. anderweitig genutzt werden.

Sollte der verwendete Drucker kein NL (1EH) als Steuerzeichen verstehen, so ist dieses im phys. Treiber in CR-LF (0DH-0AH) aufzuloesen (im angehaengten V.24- Treiber realisiert). Durch Aenderung folgender Zellen ist es moeglich, auch den USER-Port der Grundplatine als Druckerschnittstelle zu nutzen:

```
Datenausgang: bit 0
DTR-Signal:   bit 7
```

```
1428H 35H->01H
142CH 35H->01H
1437H 34H->0
1439H 08H->80H
1448H 34H->0
```

Zeitkonstanten fuer 9600 bit/s:

```
        | 2-MHz | 4-MHz
-----------------------
144EH ->|  0AH  |  17H
```

Hier noch einmal ein kleines Beispielprogramm fuer die Einbindung einer einfachen Centronics-Schnittstelle ueber den freien PIO-Port:

```
      ORG   1419H

ST:   PUSH  AF
      LD    A,I
      LD    (MINTR),A  ;retten I-Reg.
      LD    A,0FH      ;PIO Initialisierung
      OUT   1
      LD    A,L(INTAB) ;INT-VEKTOR
      OUT   1
```

```
      LD    A,87H
      OUT   1
      LD    A,H(INTAB) ;I-REGISTER
      LD    I,A
      POP   AF
      CALL  AUSG
RUECK:LD    A,(MINTR)
      LD    I,A        ;Rueckschreiben I-Reg.
      RET
;
AUSG: CMP   1EH
      JRZ   KONV-#     ;Aufloesen
AUSG1:OUT   0
DYST: DI
      CALL  INKEY      ;BREAK?
      CMP   3
      SCF              ;Fehler
      RZ
      CCF              ;CY=0
      EI
      JRNC  DYST-#     ;Warten auf INT
      RET
;
;Konvertierung des Zeilenendezeichens
;"1EH" in die Folge "0DH,0AH"
;
KONV: LD    A,0DH
      CALL  AUSG1
      XOR   A            ;C-Flag ruecksetzn
      LD    A,0AH
      CALL  AUSG1
      JR    RUECK-#
;
;Interruptroutine
;
INTS: DI
      SCF
      RETI
;
MINTR:BER   2
;
      ORG #*0FFFEH+2      ;gerade Adresse
;
INTAB:DA    INTS
;
      ORG 112H
;
DRZEI:JMP   ST
;
      END
```

Anmerkung:

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-Systeme verwendbar, bzw. es lassen sich auch CPM-kompatible Texte bearbeiten (Es muss beachtet werden, dass das ETX-Zeichen im CPM nicht 03, sondern 1AH ist→ umdefinieren oder als Steuerzeichen in den Text schreiben).

7.Implementierunsabhaengige Anpassungsarea

```
100H   JMP ANF     ;PROGRAMMANF.
103H   JMP WSTRT   ;WARMSTART
106H   JMP OUTCH   ;AUSG ZEICHEN AKKU
109H   JMP INCHR   ;EING ZEICHEN AKKU
10CH   JMP INKEY   ;TASTATURABFRAGE
112H   JMP DRAKK   ;DRUCKEN EIN ZEICHEN
      (JMP ZEIDR)   IM AKKU, ZEILENDE=1EH
115H   JMP DRDEL   ;RUECKSETZEN DES
                   ;EXT.LOG.DR.TR.
118H   JMP SARUF   ;HEADERSAVE
11BH   JMP LORUF   ;  -"-
11EH   JMP BEEP    ;BEEPERZEUGUNG
121H   JMP SYS     ;SYSTEMRUECKKEHR
```

Alle Spruenge ausser DRAKK (ZEIDR) und DRDEL(kurzg.) zeigen original auf den Sprungverteiler. Bei Nichtimplementierung einzelner Sprungverteilerroutinen sind diese unbedingt mit RET (C9H) kurzzuschliessen! Das gilt besonders fuer die BEEP-Funktion. Bei Verwendung des .A2-Monitors von Riesa gibt es Schwierigkeiten mit der INKEY-Funktion, da dieser Monitor auch ueber den RST 20H, DB 4 nicht kompatibel zur alten INKEY-Routine ist. Mit einem kleinen Zusatzprogramm kann die neue Routine zur alten kompatibel gemacht und an Stelle dieser eingebunden werden:

```
INKEY:  PUSH    IX     ;ANDERE REG. IN ALTER R. AUCH N.
    LD    BC,1000H ;REPEAT-ANSPRECHZEIT
WIC:    PUSH    BC
    RST    20H
    DB    4     ;INKEY AUS NEUEM MON.
    POP    BC     ;HL=37H
    LD    IX,4
    CMP    (IX+0)
    JRNZ    USMLZ-#     ;UNGLEICH ALTEM ZEICHEN
    OR    A
    JRZ    RSMB-#     ;KEINE TASTE GEDR.
TIME:   DEC    C
    JRNZ    TIME1-#
    BIT    4,M     ;MERKBIT ZELLE 37H
    JRNZ    KURZ-#
    LD    DE,800H     ;REPETIERFREQU.
TIME2:  DEC    DE
    LD    A,E
```

```
    OR    D
    JRNZ    TIME2-#
KURZ:   DJNZ    WIC-#
    SET    4,M
    XOR    A
    JR    USMLZ-#
RSMB:   RES    4,M
USMLZ:  LD    (IX+0),A
    POP    IX
    RET
```

Wichtige Adressen und Zellen zur Anpassung:

```
-Cursordarstellung:

 Code1    :   124H  ->C7H
 Code2    :   125H  ->FFH
 mit 7.bit:   126H  ->0/80H

-Ende-/Anfangtextzeichen:

 ETX:  127H  ->03H
 STX:  128H  ->02H

-Textpuffer-Daten:

 TPANF: 15BH ->17FFH
 TPEND: 15DH ->DFFFH

-Bildschirm:

 BWS-Anfang : 15FH ->EC00H
 Spaltenzahl: 161H ->32
 Zeilenzahl : 163H ->32

-Tastatur:

 Adr. Tast.-puf.-zelle: 165H->04H
 (wird zur blinden Abfr.
  auf 0 gesetzt)

-absolute Textbereichsgrenzen:

 untere Grenze: 167H ->1456H
 obere  Grenze: 169H ->EBFFH

-Zeichenoffset bei Steuerzeichen:

 16BH -> 0A0H
```

Diese Zellen sind im allg. konstant, koennen aber bei Hardwareaenderungen (z.B.groesserer

Bildschirm) entspr. geaendert werden.

Viel Spass mit „MicroWORD 1.5“

R. Brosig, IG-HC Dresden

PS: MicroWord 1.5 wird mit einem Ladebild ausgeliefert, welches nach dem Laden automatisch das Laden des Editors einleitet, sofern das Headersave ueber den Sprungverteiler erreichbar ist. Dieses Ladebild nimmt keinerlei Speicher in Anspruch, da dieses durch den Editor ueberladen wird. Fehlerhinweise oder Verbesserungsvorschlaege werden gern entgegengenommen.