

Roland Mester, Kurt Peter

Bildschirm-Editor für AIM-65 und PC-100

Der hier beschriebene Texteditor eignet sich für die Computer AIM-65 (Rockwell) und PC-100 (Siemens/System-Elektronik) mit angeschlossenem Video-Interface. Er arbeitet bildschirmorientiert und kann auch leicht durch neue Kommandos erweitert werden. Das Programm ist als Assembler-Listing sowie in mehreren EPROM-Versionen von unserem Software-Service lieferbar. Das Konzept ist aber sicher auch für viele Leser interessant, die den Editor nicht selbst einsetzen.

Der „Screen Editor V 3.1“ ist ein bildschirm- und cursororientierter Editor, der alle wesentlichen Funktionen zur Texterfassung und -bearbeitung bereitstellt. Er arbeitet mit allen Video-Interfaces zusammen, deren Bildwiederholungspeicher dem Rechner wie normales RAM zur Verfügung steht, z. B. ab den Adressen hex 9000 oder 9800. Er ist voll mit dem im AIM-65 eingebauten Texteditor kompatibel, benutzt allerdings sehr viele ROM-Routinen, so daß er sich nur mit erheblichem Aufwand auf andere Computertypen umschreiben ließe. Andererseits paßt er gerade dadurch noch in ein 4-KByte-EPROM (z. B. im Adressenbereich B000...BFFF oder D000...DFFF).

Bildschirm als Textfenster

Der Bildschirm liegt wie ein Fenster über dem Text, das entweder zeilenweise oder seitenweise über den Text verschoben werden kann. Eingabe und Änderungen am Text sind mit einem sichtbaren Cursor und einem leistungsfähigen Befehlssatz möglich. Da einerseits häufig benutzte Kommandos schnell erreichbar sein sollten, andererseits aber keine unüberschaubare Doppelbelegung von Tasten mit CTRL-Zeichen sinnvoll ist, wurde der Befehlssatz in eine Direkt- und eine Indirekt-Ebene aufgeteilt. In

der direkten Ebene sind alle Kommandos integriert, die man sehr häufig braucht, z. B. Cursorbewegungen. Sie sind über Control-Tasten zugänglich. Alle anderen liegen in der indirekten Ebene, von der aus man die textübergreifenden Kommandos mit Kurzbezeichnungen erreichen kann. Diese Befehle können von der Kommandozeile aus gegeben werden, die über dem Text invers dargestellt wird und in der auch alle Editor-Meldungen erscheinen.

Der Screen-Editor kann auch parallel mit irgendwelchen Sprachen (Basic, Pascal usw.) benutzt werden, da der benötigte Zero-Page-Bereich auf den Stack gerettet wird. Fehlbedienungen jeder Art werden so abgefangen, daß der Benutzer in seiner Freiheit praktisch nicht eingeschränkt wird. Die Kommandoabkürzungen sind, soweit sinnvoll, dem Englischen entliehen.

Im folgenden wird unterschieden zwischen einer offenen und einer geschlossenen Zeile: Letztere ist durch ein Return-Zeichen abgeschlossen. Neue Zeilen werden prinzipiell erst nach dem Drücken der Return-Taste in den Hauptspeicher übernommen. Deshalb kann man auch eine offene Zeile nicht verlassen, sonst wäre sie ja verloren: Entweder drückt man Return, oder man löscht sie. „Return“ wird (im Gegensatz zum AIM-

Editor) auf dem Bildschirm als sichtbares Zeichen dargestellt.

Die Initialisierung

Mit jedem Einstieg wird der DILINK-Vektor des AIM auf hex EF05 gesetzt. Damit wird ein eventuell vorhandenes Video-Treiber-Programm außer Betrieb gesetzt. Die Startadresse ist der Beginn des jeweiligen EPROM-Bereichs, z. B. B000 oder D000. (In diesen beiden Fällen ist der Start auch mit den AIM-Tasten 5 bzw. N möglich.) In der Kommandozeile erscheint „FROM=“, womit nach der Text-Startadresse gefragt wird, dann „TO=“ zur Eingabe der Text-Endadresse. Hier kann man noch mit Escape zum Monitorprogramm zurückkehren, ohne Speicherinhalte zu zerstören. Drückt man nach FROM sofort Return, so wird hex 0200 eingesetzt; ein Return sofort nach TO veranlaßt den Editor, nach der Obergrenze des Arbeitsspeichers zu suchen.

Will man später in den Editor einsteigen, ohne Texte zu löschen, so kann das an der Anfangsadresse + 3 geschehen, also z. B. D003 oder B003 (entspr. AIM-Taste 6). Dabei werden allerdings alle Tabulatorpositionen gelöscht und fünf Standardpositionen in Zehnerabständen eingetragen. Außerdem wird als Standardsprungtaste „ß“ (PRINT auf der ASCII-AIM-Tastatur) festgelegt.

Texteingabe

Die oberste Zeile des Bildschirms wird invers dargestellt und bleibt immer stehen. Sie dient als Kommando- und Meldezeile. Die restlichen 15 Zeilen (bei einem Video-Interface mit 16 Zeilen) sind dem Text vorbehalten. Alle Tasten werden, sobald sie länger als 0,5 s gedrückt werden, automatisch etwa 10mal pro Sekunde wiederholt. Es ist gemischte Groß- und Kleinschreibung möglich. Die Shift-Arretierung (nur Großbuchstaben) ist nach der Initialisierung stets eingeschaltet; mit CTRL-A kann sie aus- und wieder eingeschaltet werden. Sie beeinflußt nur die Buchstabentasten. Die Tastatur wird vom Programm selbständig decodiert, so daß auch PRINT und LF unabhängig von ihrer sonstigen Wirkung verfügbar sind und z. B. für die Darstellung deutscher Umlaute verwendet werden können. Eine Unterstreichung von Zeichen ist über Shift-DEL erreichbar. Die Zeilenbreite ist entsprechend dem AIM-Format auf 61 Zeichen begrenzt, das sind 60 darstellbare Zeichen und ein

Return. Die nachträgliche Verkürzung von Zeilen ist jederzeit möglich, indem man an der jeweiligen Stelle Return drückt. Alles, was rechts davon steht, wird dabei gelöscht. Einfügungen am Zeilenende sind möglich, indem man das Return-Zeichen (CR) überschreibt. In der ersten Spalte einer Zeile ist Return verboten: Leerzeilen gibt man durch die Folge Space-Return ein.

Die Tastaturdecodierung des Editors arbeitet mit einer gegenüber dem AIM-Monitorprogramm verdoppelten Entprellzeit. Auch eine sehr schnelle Eingabe wird dadurch noch nicht behindert. Control-Zeichen werden bei der Eingabe nicht akzeptiert, soweit es sich nicht um Kommandos handelt. Man kann sie jedoch nachträglich einfügen, indem man zunächst irgend ein Sonderzeichen eingibt und es später durch die entsprechenden Control-Zeichen (z. B. zur Druckersteuerung) ersetzt.

Der Cursor wird dadurch dargestellt, daß das unter ihm liegende Zeichen invers erscheint. Eingaben sind nur bei sichtbarem Cursor möglich. Beim Überschreiten der 50. Position einer Zeile wird ein kurzer Piepton ausgegeben, um auf das nahende Zeilenende hinzuweisen.

Der Screen-Editor dient ausschließlich zum Erstellen und Editieren von Texten auf dem Bildschirm. Da er mit dem AIM-Editor kompatibel ist, kann man die Texte mit letzterem z. B. auf Band abspeichern und wieder laden.

Der Befehlssatz

CTRL-L (Cursor Left)
Cursor nach links, ohne Zeichen zu löschen.

CTRL-R (Cursor Right)
Cursor nach rechts. Blockiert am rechten Zeilenrand.

CTRL-U (Cursor Up)
Cursor nach oben. Beim Erreichen des oberen Bildschirmrandes rollt der Text nach unten. Blockiert am Textanfang, während CTRL-I und wenn in eine offene Zeile bereits etwas eingetragen wurde.

CTRL-D (Cursor Down)
Cursor nach unten. Beim Erreichen des unteren Bildrandes rollt der Text nach oben; unterhalb der letzten Zeile wird eine freie Zeile zur Verfügung gestellt. Blockiert, wenn der Cursor in der letzten

Textzeile steht, während CTRL-I und wenn in eine offene Zeile bereits etwas eingetragen wurde.

CTRL-E (Erase Character)
Löscht das Zeichen, auf dem der Cursor steht. Blockiert, wenn der Cursor auf CR oder am rechten Rand steht.

DEL (Delete Character)
Löschen des links vom Cursor stehenden Zeichens. Nur am Ende einer offenen Zeile möglich.

CTRL-F (Fill Character)
Einfügen von Zeichen links von der Cursorposition. In der Kommandozeile erscheint „FILL“. Dieser Modus wird durch nochmaliges CTRL-F oder jedes andere Kommando verlassen. Blockiert, wenn rechts keine Zeichen stehen oder wenn die Zeile voll ist.

CTRL-K (Kill Line)
Zeile löschen (bei offenen und geschlossenen Zeilen möglich).

CTRL-I (Insert Line)
Zeile über der Cursorposition einfügen. Der Insert-Modus wird in der Kommandozeile angezeigt. Die neue Zeile kann man entweder beschreiben und mit Return verlassen oder durch erneutes CTRL-K aufgeben. Blockiert, wenn CTRL-I selbst gerade ausgeführt wird oder wenn man sich in einer offenen Zeile befindet.

CTRL-T (Display Top Lines)
Anzeige der ersten 15 Textzeilen auf dem Bildschirm. Blockiert, wenn der Cursor schon auf der obersten Zeile steht, in eine offene Zeile bereits etwas eingetragen wurde oder CTRL-I gerade ausgeführt wird.

CTRL-B (Display Bottom Lines)
Setzt den Cursor ans Textende, zeigt die letzten 14 Textzeilen an und stellt eine Leerzeile zur Fortsetzung zur Verfügung. Blockiert, wenn CTRL-I gerade ausgeführt wird, wenn in eine offene Zeile bereits etwas eingetragen wurde oder wenn man schon auf der untersten Zeile steht.

CTRL-N (Display Next Page)
Nächste Seite, d. h. die nächsten 14 Zeilen anzeigen. Die unterste Zeile der vorhergehenden Seite wird die oberste der neuen. Blockiert wie CTRL-B.

CTRL-P (Display Previous Page)
Vorhergehende (previous) Seite anzeigen, entsprechend CTRL-N.

CTRL-A (Alter Shiftblock)
Shift-Lock ein- oder ausschalten. Der Zustand wird in der Kommandozeile angezeigt.

CTRL-S (Split Lines)
Aufspalten einer Zeile in zwei. Dazu geht man in die entsprechende Zeile und drückt dort, wo eine neue begonnen werden soll, auf Return. Blockiert, wenn die Zeile noch nicht abgeschlossen ist, während CTRL-I und wenn der abgetrennte Zeilenteil keine Zeichen enthalten würde.

CTRL-J (Join Lines)
Zwei aufeinanderfolgende Zeilen zusammefügen. Nach CTRL-J wird das Return-Zeichen der aktuellen Zeile durch einen Leerraum ersetzt. Blockiert, wenn kein Platz mehr für die nächste Zeile ist, während CTRL-I und wenn die aktuelle Zeile noch nicht abgeschlossen ist.

CTRL-H (Cursor Home)
Cursor home: setzt den Cursor an den linken Rand. Blockiert, wenn er bereits dort steht.

CTRL-C (Copy Line)
Dupliziert die Zeile, auf der der Cursor gerade steht. Blockiert, wenn die Zeile noch nicht abgeschlossen ist.

CTRL-PRINT
Setzt an die Cursorposition eine von maximal 20 Tabulator-Positionen. Blockiert, wenn dort schon eine Tabulatorposition sitzt.

CTRL-F3
Löscht den Tabulator an der Cursorposition. Blockiert, wenn dort kein Tabulator ist.

β (Shift-PRINT)
Tabulatorsprung. In der Indirekt-Ebene kann man auch ein anderes Zeichen als Tabulatorsprung vereinbaren.

ESC
Damit steigt man aus dem Text aus und landet in der Kommandozeile. Nun hat man die Befehle der Indirekt-Ebene zur Verfügung. Nochmaliges ESC bewirkt einen Sprung zu einer User-Adresse. Mit Return kommt man von der Kommandozeile zurück in den Text. Blockiert, wenn CTRL-I gerade ausgeführt wird oder wenn in eine offene Zeile bereits etwas eingetragen wurde. ESC wird übrigens auch mit CTRL-F1 ausgeführt, Return mit CTRL-M.

Indirekte Kommandos

Ist der Cursor z. B. nach ESC in die Kommandozeile gesprungen, befindet man sich im indirekten Modus. Hier müssen alle Kommandoaufrufe mit Return abgeschlossen werden. Die Befehle müssen hier mit ihrem Namen aufgerufen werden und lassen sich auch mit mindestens einem Buchstaben abkürzen. Ihre Namen werden in die Kommandozeile nach dem Aufruf eingeblendet.

Ist der Textspeicherbereich aus irgend einem Grund nicht mit einem Null-Byte abgeschlossen, so landet man nach der Ausführung einiger Indirekt-Kommandos im AIM-Editor (Anzeige von END).

Die Zero-Page-Variablen von hex B2 bis FF sind dann u. U. zerstört. Der Mangel läßt sich beheben, indem man wieder in den Screen-Editor einsteigt, sich die Editor-Parameter anzeigen läßt und in die Speicherzelle, auf die „END=" zeigt, vom Monitor aus hex 00 einträgt.

Folgende indirekte Kommandos gibt es:

?

In der Kommandozeile erscheinen folgende vier Adressen: FROM = Startadresse des Textes, TO = Endadresse des Text-Speicherbereichs, END = momentane Endadresse des Textes, NOW = Adresse des ersten Zeichens der Zeile, in der der Cursor steht.

Return

Zurück in den Texteingabe-Modus. Dann stehen wieder die Befehle der Direkt-Ebene zur Verfügung.

ESC

Sprung zu einer User-Routine. Diese Routine sollte den Stackpointer wieder auf hex FF setzen. Den Befehl kann man z. B. zur Bildschirminitialisierung verwenden. Seine Programmierung geht aus dem Assemblerlisting hervor, das unser Software-Service zum Screen-Editor liefert.

QUIT

Ausstieg aus dem Editor, Sprung zum Monitor inklusive Stack-Bereinigung.

RANDAUSGLEICH

Der Text wird links- und rechtsbündig im Speicher ausgerichtet, indem der Text in zu kurzen Zeilen mit Leerräumen aufgefüllt wird. Zeilen, die einen Absatz bilden, müssen vor dem Return-Zeichen einen Leerraum enthalten. Die

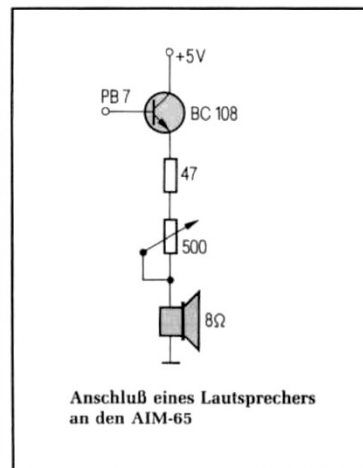
Ausführung erfolgt ab der aktuellen Textzeile, also vorher CTRL-T drücken! Bei langen Texten kann die Bearbeitung einige Minuten dauern, 30 KByte Text z. B. rund eine halbe Stunde.

COMPRESS

Diese Routine bewirkt, daß Zeilen, die noch nicht ganz voll sind, mit kompletten Worten der jeweils nächsten aufgefüllt werden. Ansonsten gilt das Gleiche wie für den Randausgleich.

TABULATORZEICHEN

Nach der Initialisierung des Screen-Editor dient CTRL-PRINT als Tabulator-



sprung. Nach diesem Kommando wird das jeweils definierte Zeichen (z. B. ß) angezeigt. Ist man damit zufrieden, braucht man nur Return zu drücken, andernfalls das gewünschte Zeichen, gefolgt von irgend einer anderen Taste als Bestätigung.

FIND

Suchen eines maximal 20 Zeichen langen Strings im Text. Es wird nicht zwischen Groß- und Kleinschreibung unterschieden. Der Cursor steht, wenn das Wort gefunden wird, an dessen Anfang; dann gibt es drei Möglichkeiten: Mit ESC unterbricht man die Suche und kehrt in die Indirekt-Ebene zurück. Die Leertaste bewirkt ein Weitersuchen.

Und mit Return oder irgend einer anderen Taste kann man ohne Umwege in den Text zurückkehren. Bei erfolgloser Suche ertönt ein kurzer Piepton. Befindet sich der Suchstring mehrfach in ei-

ner Zeile, so wird er auch mehrfach gefunden – im Gegensatz zum AIM-Editor!

CHANGE

Dieser Befehl dient zum Ändern von Zeichenfolgen im Text. Zunächst funktioniert er ähnlich wie FIND. Nach erfolgreicher Suche hat man außer ESC, Leertaste und Return zusätzlich die Möglichkeit, CTRL-C zu drücken, so daß die gewünschte Änderung ausgeführt wird. Anschließend wird das nächste Auftreten des zu ändernden Strings gesucht. Selbstverständlich ist es auch möglich, einen String zu löschen, indem man nach TO gleich Return eingibt. Der alte String darf maximal 20, der neue maximal 25 Zeichen lang sein. Der Befehl blockiert, wenn die Änderung zu einer zu großen Zeilenlänge führt.

MOVE/COPY

Nach SOURCE FROM muß man angeben, wo der zu transferierende Block beginnt, nach SOURCE TO, wo er endet, nach DESTINATION den Zielpunkt des Transfers. Dies geschieht, indem man jeweils den Cursor an den Anfang der gewünschten Zeile setzt und die Return-Taste betätigt. Der Befehl kann z. B. bei 30 KByte Text rund 45 Minuten dauern.

KILL

CTRL-K ist für längere Textblöcke zu langwierig. Deshalb kann man mit KILL ganze Blöcke löschen. Nach FROM landet man im Text, und man kann den Blockanfang mit Return markieren, ebenso das Blockende nach TO. Die Ausführung dauert maximal vier Sekunden.

Erweiterungen

Um den schon erwähnten Piepton über einen Lautsprecher ausgeben zu lassen, genügt eine einfache Schaltung (Bild). Die Originalversion des Editors wurde für das Video-Interface CRT-2 von Graf (Kempton) entwickelt. Um das Return-Zeichen und deutsche Umlaute darstellen zu können, ist es sinnvoll, das Zeichengenerator-EPROM entsprechend zu ändern. Angaben hierzu, zur Änderung des AIM-Monitorprogramms für größere Zeilenbreiten als 61 sowie Erweiterungs-Anregungen enthalten die Unterlagen, die vom Franzis-Software-Service (Postfach 37 01 20, 8000 München 37, Tel. 0 89/51 17-3 31) gegen eine Schutzgebühr bezogen werden können. Dort ist auch ein programmiertes 4-KByte-EPROM vom Typ 2532 erhältlich, das entweder in den Basic- oder Assembler-Steckplatz des AIM paßt.