

**MOE**

**MOE**

**MOE**

**INSTRUTEK**  
8700 Horsens · Tlf. 05 · 61 11 00

**MICROCOMPUTERS**

**MICROCOMPUTERS**

**MICROCOMPUTERS**

**MICROCOMPUTERS**

**KIM TEXT EDITOR**

**USER MANUAL**



**MICROCOMPUTER FAMILY  
KIM TEXT EDITOR  
User Manual**

Copyright by MCDS  
3. englische Auflage 1977  
Auflagenhöhe 1. bis 3. : 15.000

The information in this manual has been reviewed and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. The material in this manual is for informational purposes only and is subject to change without notice.

MCDS verfügt über alle Rechte der deutschen und englischen Ausgabe. Nachdrucke und Vervielfältigungen, auch auszugsweise, sind nur mit ausdrücklicher Genehmigung der MCDS gestattet.

Auch bei dieser 3. englischen Auflage freuen wir uns über kritische Anregungen des Leserkreises, die zur kontinuierlichen Verbesserung der Qualität weiterer Auflagen Verwendung finden.

Satz, Druck und Vertrieb von MCDS Microcomputer Datensysteme GmbH,  
Luisenplatz 4, Postfach 11 0868, D-6100 Darmstadt, Tel. 06151/2 03 02,  
Telex 04 19 390 hysy d



## TABLE OF CONTENTS

INTRODUCTION		1
CHAPTER 1	INSTALLATION AND CHECKOUT	3
CHAPTER 2	SAMPLE USE OF THE KIM EDITOR	4
CHAPTER 3	BASIC EDITOR OPERATIONS	9
CHAPTER 4	USING PAPER TAPE AND CASSETTE TAPE	13
CHAPTER 5	OTHER COMMANDS	15
CHAPTER 6	CHANGING THE EDITOR	17
CHAPTER 7	CHANGING ASSEMBLER/EDITOR I/O DEVICES	23

## LIST OF TABLES

Table 1	KIM EDITOR COMMAND SUMMARY	2
Table 2	FIRST FEW WORDS OF EDITOR PROGRAM	3
Table 3	KIM EDITOR User-Modifiable Vectors in Page Zero	18



## INTRODUCTION

Congratulations on your purchase of the KIM Text Editor. The Text Editor is an integral part of the KIM Editor/Assembler package and will allow you to easily input and modify text to your KIM system memory area. Although primarily designed to work with the KIM Resident Assembler, any text material may be input and edited using the KIM Editor.

The KIM Editor features:

Line-numbered text entry and editing for ease of use

Single-letter mnemonic commands for easy operation

ROM resident--no need to load or protect the Text Editor program

Designed to work with any 650X-based system

User-defined commands for complete flexibility

Automatic linkage to the KIM Resident Assembler

Complete compatibility with KIM audio cassette interface and paper tape

Memory independent. Text may be stored anywhere in memory and multiple text files may reside in memory simultaneously.

Table 1 gives a brief summary of the KIM Editor commands.

Your KIM Editor has been carefully designed and thoroughly tested to assure correct operation and complete compatibility with your KIM system. If you suspect that your KIM Editor is not working correctly, contact the Manager of Product Support, MOS Technology Sales, Inc. 901 California Avenue, Palo Alto, CA 94304. Please include a brief terminal session which documents any suspected problem.

Table 1  
KIM EDITOR  
COMMAND SUMMARY

P A    Print all stored text.

P n    Print text beginning at line n.

F xyz Find and print all lines containing string xyz.

S      Print Status--hex origin and end of text and  
        decimal number of lines in the current text file.

R      Resequence line numbers by 5's.

Q      Query--return to BASE= query.

E      Exit through link vector--default is KIM.

?      Set ? as prompt character.

?x     Set x as prompt character.

X      JSR to user-defined command.

A      Go to Assembler with current file.

K<sup>C</sup>    (control K) exit to KIM monitor.

T<sup>C</sup>id   Dump current file to audio tape with ID of id.

T      Dump current file to paper tape.

NOTE: All commands must be terminated by a carriage  
return.



Chapter 1  
INSTALLATION AND CHECKOUT

Your KIM Editor/Assembler ROMs were designed to be installed on a KIM-5 ROM expansion board. The three ROMs which compose the editor and assembler should be mounted on your KIM-5 as described in the KIM-5 User's Manual. The editor, assembler, and KIMath floating-point math package reside in the highest 8K of KIM memory space. Thus, one of the KIM-5 memory address switches should be set to an address of E000.

Once you have inserted the ROMs in the KIM-5 and inserted the KIM-5 in your KIM motherboard, you can verify that the system is operating correctly by examining the first few words of the editor program. Table 2 shows the first few words of the editor program, starting at location F100.

Table 2

Location	Contents
F100	20
F101	A5
F102	F6
F103	A9
F104	00
F105	85
F106	CD
F107	D8

Chapter 2  
SAMPLE USE OF THE KIM EDITOR

To operate the Editor as described in this manual, the following minimum system is required:

- (A) A KIM-1 microcomputer
- (B) 4K of expansion memory
- (C) A KIM-4 motherboard
- (D) A KIM-5 ROM expansion board with the KIM resident assembler/text editor installed

Throughout this manual it is assumed that your random access expansion memory has a base address of 2000 (hex). Naturally, your expansion memory may be as large as you wish and located in any memory space which does not conflict with the KIM monitor or other ROM firmware. This manual also assumes that you have a TTY-like terminal connected to the KIM serial input/output port. If you are using parallel input and output ports or other nonstandard I/O configurations, you will have to write driver software to support such devices. See Chapter 7 for the appropriate modifications.

Entering the Editor

To start the editor, enter address F100 and hit the G key on your TTY. The editor will respond by typing BASE=. You should respond by typing the hexadecimal starting address at which you wish your block of text to begin. The editor will respond by typing N OR O?. If you wish to edit some text already placed in memory (through a tape dump, for instance), you would respond O, since you are editing old text. For the purpose of our example, we will enter an N since we

are creating a new text file. The editor will respond with a carriage return and line feed and wait for you to enter the first line of text. Example 1 shows this initial dialogue. Note that throughout this document what you type in will be underlined for clarity. Naturally, this underlining does not occur when you actually use the editor.

Example 1  
STARTING THE EDITOR

```
KIM          + Depress RS on KIM-1 and
              rubout on your TTY

XXXX XX

F100        + Starting address of Editor

F100 20G    + Go

BASE=2000   + Hex starting address of text

N OR O?N    + New text file
```

Entering Text

Since the Editor is primarily designed to prepare assembly language instructions to submit to the KIM resident assembler we will use as our example the small program shown on page 11 of the KIM-1 User Manual.

Every line of text entered into the editor must have a decimal line number between 1 and 9999. If the same line number is used twice, the new line replaces the previous line with the same line number. Here is what our sample would look like as it is typed in (Note that increments of 10 are used between each line number; this is not necessary but makes it easier to enter additional text in the future):

Example 2

ENTERING TEXT

```
10 POINTL = $FA  
20 POINTH = $FB  
30 VAL1  
40 VAL2  
50 PROG CLC  
60 LDA VAL1  
70 ADC VAL2  
80 STA POINTL  
90 LAD #00  
100 STA POINTH  
110 JMP START
```

Remember that you must type a carriage return at the end of each line you type in.

Listing Text

To list the text you have just typed in, type the character P for Print, a space, and the character A for All. The editor will then type back all of the text which you have typed in.

Example 3

LISTING TEXT

```
P A  
0010 POINTL = $FA  
0020 POINTH = $FB  
0030 VAL1  
0040 VAL2  
0050 PROG CLC  
0060 LDA VAL1  
0070 ADC VAL2  
0080 STA POINTL  
0090 LAD #00  
0100 STA POINTH  
0110 JMP START  
*ET
```

### Changing the Text

To make our sample program acceptable to the assembler we will now have to add and change several lines. To add a line between two existing lines, simply type a line number between the two values and enter the text. (Notice the new line 25 entered in Example 4.)

To change an existing line, type the same line number and the new contents for that line. (Note in Example 4 how lines 30, 40, and 90 were changed.) Note that a single character in a line cannot be corrected; the entire line must be retyped.

#### Example 4

##### CORRECTING TEXT

```
25  *=$0000
30  VAL1 *+=+1
40  VAL2 *+=+1
90      LDA #00
120 .END
26  START = $1C00
P A
0010 POINTL = $FA
0020 POINTH = $FB
0025 *=$0000
0026 START = $1C00
0030 VAL1 *+=+1
0040 VAL2 *+=+1
0050 PROG CLC
0060      LDA VAL1
0070      ADC VAL2
0080      STA POINTL
0090      LDA #00
0100      STA POINTH
0110      JMP START
0120 .END
*ET
```

Note that the \*ET at the end of a print command signifies the End of Text.

### Saving the Text on Audio Cassette Tape

To transfer your edited text to audio cassette for storage, first insert a cassette in your recorder and assure that the cassette recorder is properly

connected to your KIM system as outlined in the KIM-1 User's Manual. To begin transfer of the text file to the audio cassette, hold down the control button on your TTY (marked CTRL), and strike the character "T." Then release the control button and type a two-digit decimal number between 01 and 99 (do not use the hexadecimal numbers A-F). Before striking the carriage return, put your cassette recorder in record mode and start the tape. Then strike the carriage return. The editor will automatically insert the starting and ending address of your text and the ID you have typed into the correct locations in KIM. The editor will then transfer control to the audio cassette routines. When the tape has been properly recorded, control will return to the KIM monitor.

#### Reloading the Text from Audio Cassette

When you wish to place the text you previously stored on cassette back into memory for further editing or assembly, first prepare your cassette system for playback. Then set location 00F1 to 00, put the ID you used when recording the cassette into location 17F9, and address location 1873. Depress the G key and play back the tape. When the tape has been read in, start the editor (location F100), give the starting address of the text in response to the BASE= query and answer "O" to the N OR O? query. The text file is now ready for further processing.

If you had used an ID of 43 to record the sample program, the playback sequence would look like:

```

KIM
XXXX XX F1
00F1 XX 00.  + Clear decimal mode
00F2 XX 17F9
17F9 XX 43.  + Enter ID
1780 XX 1873
1873 XX G
(Start tape playback here)
KIM
0000 XX F100 + Playback complete, go to editor
F100 XX G    + Start editor
BASE= 2000  + Give base
N OR O? O   + Old file

```

## Chapter 3

### BASIC EDITOR OPERATIONS

Now that you have had an opportunity to see the basic operations of the editor we will now examine each editor capability in detail.

#### Entering Text

As we saw in Chapter 2, text is entered into memory by typing in the line of text preceded by a line number. Line numbers may range from 1 to 9999; there is no necessity to type leading zeros, the editor will insert them automatically. Note that any line typed to the editor which does not contain a line number will be interpreted as an editor command. Any command typed in which is not recognized by the editor will result in the error message BAD COM.

If you type in a portion of a text line and realize you have made a mistake, there are two methods of correcting your error. If you depress the control key and hit the character "X" (control X) the line will be ignored and a carriage return/line feed will be issued by the editor. If you wish to "back-space" and retype an erroneous character, depress the shift button and strike the 0 key (+). The editor will "back up" one character each time you strike the + key. You may strike the + key as many times as necessary to delete previous characters.

#### Listing Text

Chapter 2 showed how to list the entire text file by typing P A. If you wish to only list a portion of the text, type P, a space, and a line number. The editor will then type all of the text beginning with that line number and continuing through the end of the file. If you wish to only list a few lines of

text, type P, the first line of text you wish to see, and a carriage return. The editor will then begin listing at that point. When you have seen enough of the text, depress the break button. This will signal the editor that you wish it to stop listing; it will stop printing and wait for another command to be entered.

#### Adding a New Line

Each line you type will automatically be inserted in the text file in line number sequence. Thus, if you wish to add a new line between old lines 20 and 30, simply give it a line number between 21 and 29. If you wish to insert a new line between two lines with adjacent line numbers (e.g., insert a line between line numbers 24 and 25), you must resequence the existing line numbers by using the resequence command (see below). There is no restriction on the number of text lines or the size of the text file which the editor can handle. The only restriction is the amount of memory available for such storage.

#### Resequencing Lines

The KIM resident assembler ignores the line numbers in your file when doing an assembly. Thus, the line numbers are a convenience for you when you are editing text. If you wish to resequence the line numbers, simply type an R and a carriage return. The editor will automatically resequence all the line numbers in your text file. The first line will be the line number 5 and each line number which follows will be incremented by 5.

#### Example 5

#### RESEQUENCING LINES

```
1 LINE 1  
2 LINE 2  
3 LINE 3  
R  
P A
```



```
0005 LINE 1
0010 LINE 2
0015 LINE 3
*ET
```

### Locating Characters in the Text

In a lengthy text file it is often desirable to be able to find all lines in the text which contain certain characters or groups of characters. This is done with the Find command. Example 6 illustrates the use of the Find command. First the entire text file is listed using the P A command, then all lines which contain references to POINTH are printed by issuing the command F POINTH. The second portion of the example shows finding all lines which contain an asterisk by typing F \*. You must always type a space after the F.

#### Example 6

##### FINDING SPECIFIED TEXT

```
P A
0010 POINTH = $FA
0020 POINTH = $FB
0025 *= $0000
0030 VAL1 = *+1
0040 VAL2 = *+1
0050 PROG CLC
0060 LDA VAL1
0070 ADC VAL2
0080 STA POINTL
0090 LDA #00
0100 STA POINTH
0110 JMP START
0120 .END
*ET
```

```
F POINTH
0020 POINTH = $FB
0100 STA POINTH
*ET
```

```
F *
0025 *= $0000
0030 VAL1 = *+1
0040 VAL2 = *+1
*ET
```

### Exiting from the Editor

To return to the KIM monitor, depress the control key and strike the character "K" (control K) followed by a carriage return.

The user may also return to the KIM monitor by typing an "E" (Exit).  
(The meaning of the "E" command may be modified by the user. See Section 6.)

## Chapter 4

### USING PAPER TAPE AND CASSETTE TAPE

#### Using Paper Tape

Once you have finished editing a text file you may wish to punch it out on paper tape. To do this, turn on the TTY paper tape punch and type a T and a carriage return. The editor will then punch 20 null characters on the tape and dump the file onto the paper tape in ASCII format. Once the tape punch is complete the editor will punch another series of null characters to terminate the punching. Turn the paper tape punch off and remove the tape from the punch.

To load a paper tape into the editor, start the editor and type in the base address at which you wish the text on the paper tape to begin. Answer the N OR O query with an N to indicate that you are putting in a new text file from paper tape. Place the paper tape in your reader, and briefly move your reader switch to the start position. As each line is read, KIM will stop the paper tape reader, process the line, and restart the reader. Make sure that you place the paper tape in the reader with the null characters punched by the editor under the read head. Note that it is not necessary to load the text on paper tape into memory to the same address from which it was originally dumped.

#### Audio Tape

The loading and dumping of audio tape from your cassette recorder to the editor is completely documented in Chapter 2 of this manual. Note that while the paper tape may be placed anywhere in memory when read back in, the cassette tape will always be reloaded in the memory area from which it was originally dumped. If you wish to load audio cassette data into a memory area other than

that from which it was originally recorded, this may be done by using an ID of FF on playback. See Section 4.2 of the KIM-1 User Manual for details.

## Chapter 5 OTHER COMMANDS

### The Status Command

Typing an S command to the editor will result in the editor typing back three numbers. The first number is the hexadecimal starting address of the current text file. The second number is the hexadecimal address of the end of the current text file. The third number is the decimal number of lines contained in the current text file. If you are entering a text file which you suspect may approach the capacity of your available memory, you can check the amount of memory being used from time to time as you enter the text. You should note that the S command will return incorrect information if no text has been entered into your current text file.

### The Quit Command

Typing a Q command to the editor returns you to the BASE= query of the editor. You may then specify a new origin for further text entry of other text files. Previously entered text files will not be disturbed unless the files overlap in memory.

### The Assemble Command

Typing an A command to the editor will terminate editor function and transfer control to the KIM resident assembler program. The editor automatically configures the assembler for a memory-to-memory assembly of the current text file. You should note that the transfer is made to the assembler cold start entry point so this command may not be used in multiple file assembly. For the second and succeeding files of a multiple file assembly, you must exit from the

editor to the KIM monitor and then enter the resident assembler at its warm start point. See the assembler documentation for further details.

## Chapter 6

### CHANGING THE EDITOR

#### Changing the Prompt Character

It is sometimes convenient to have the editor type a prompt character at the beginning of each line when it is ready for input from the user. When the editor is initially entered, this prompt character is set to a null. If you wish a prompt character you simply type a question mark and carriage return and the editor will begin each line of input with a question mark as a prompt. If you wish to use a prompt character other than the question mark, type a question mark, the character you wish to use as a prompt, and a carriage return. For instance, typing a question mark, asterisk, carriage return will set the prompt character to an asterisk. To "turn off" the prompt character, simply type a question mark followed by a null (generated by depressing the control and shift buttons on your TTY and striking the P key).

#### Modifying the Exit Point

Three parts of the editor may be changed by the user through modification of page zero memory locations. These are (1) the destination of the exit command, (2) the definition of user-defined commands, (3) the input/output devices supported by the editor.

Changing I/O devices is covered in Chapter 7. In each case, the editor normally presets these page zero locations. The exit command is preset to return to the KIM monitor, the user-defined command is defined as a return to the editor, and the input/output devices are defined to be the TTY. If you

Table 3  
 KIM EDITOR  
 User-Modifiable Vectors  
 in Page Zero

<u>Name</u>	<u>Location</u>	<u>Purpose</u>	<u>Default Value</u>
USRCMD	ED, EE	Starting address of user-defined command subroutine	F617
LINK	EB, EC	Starting address to jump to when exit command is issued	1C14
OUTPUT	E7, E8	Output character subroutine address (should include BREAK test)	F6BA
INPUT	E5, E6	Input character subroutine address	1E5A
ENDLIN	E9, EA	End-of-line subroutine address	1E2F
Page Zero Locations for User-Defined:			
OT	DB, DC	address of beginning of text	N/A
EOT	DD, DE	address of end of text	N/A
TXBUF	84-CB	contents of last line typed by user	N/A



wish to change any of these three items, you must preset the locations for all three options and then enter KIM through its alternate start point, location F103.

For example, if you wished to Exit to your own program starting at location A000, you would load 00 in 00EB and A0 in 00EC. You would also have to load the default values for USRCMD, OUTPUT, INPUT, and ENDLIN as shown in Table 3. You would then enter the text editor at location F103 instead of F100.

#### User-Defined Commands

Any command typed to the editor which begins with the character "X" will cause the editor to do a JSR to the subroutine pointed to by bytes 00ED and 00EE in page zero. The editor presets 00ED and 00EE to point to an RTS instruction in the editor, thus an X command is usually ignored. You may preset 00ED and 00EE to contain the address of a subroutine you have written to perform some special editing function--you can extend the editor to meet your own desires!

#### Memory Organization for User-Defined Commands

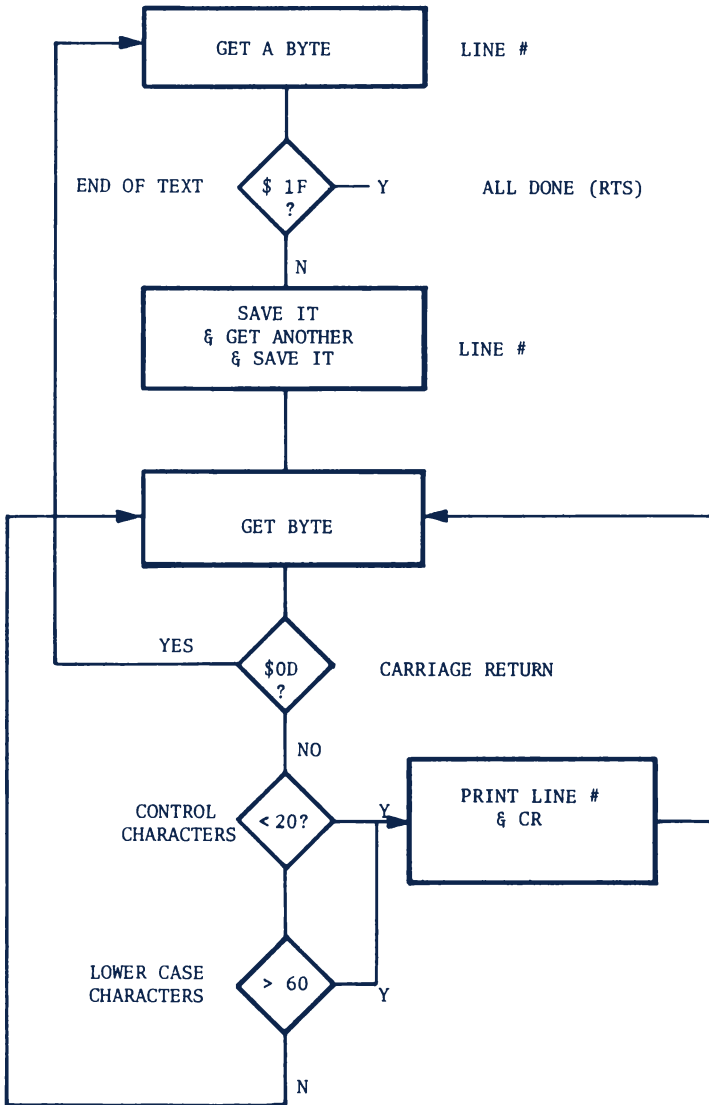
The following information will help you in writing your own commands:

1. The command you typed in (which began with an "X") is stored in the line buffer--locations 0084 through 00CB.
2. The end of text in the buffer is marked with an ASCII carriage return character (hex 0D).
3. The address of the beginning of your text file in memory is stored in locations 00DB and 00DC (low order byte first).
4. The address of the end of your text file is stored in 00DD and 00DE.
5. The last byte in your text file contains hexadecimal 1F.

6. Each line in the text file begins with the decimal four-digit line number packed in two bytes, thus line number 1763 would be packed as 17 as the first byte and 63 as the second byte.
7. Each line in the text file ends with an ASCII carriage return character ( $\text{\textbackslash}D$ ).
8. Your subroutine returns to the editor with an RTS instruction.

Program 1 shows a KIM assembly of a sample user-defined subroutine to check a source program for nonprinting characters (control characters or ASCII lower-case characters). Note that you would have to preset location 00ED to 00 and location 00EE to 02 (the starting address of the subroutine is 0200). You would also have to supply the default vectors for INPUT, OUTPUT, ENDLIN and LINK as shown in Table 3, and start the editor at location F103 instead of F100.

PROGRAM 1  
FLOW-CHART



KIMASM

LINE #	LOC	CODE	LINE	
0005	0200		PRTBYT	= \$1E3B
0010	0200		CRLF	= \$1E2F
0015	0200	A0 00	CKVAL	LDY #0 ; CLEAR Y
0020	0202	A5 03		LDA \$DB ; GET OT
0025	0204	85 00		STA \$00
0030	0206	A5 DC		LDA \$DC
0031	0208	85 01		STA \$01
0040	020A	B1 00	BEGIN	LDA (\$0),Y ; FIRST LN BYTE
0045	020C	C9 1F		CMP #\$1F ; *ET?
0050	020E	F0 ** **		BEQ END ; YEP..
0055	0211	85 02		STA \$02 ; NUP..
0060	0213	20 ** **		JSR INCPTR ; READY FOR NEXT
0065	0216	B1 00		LDA (\$0),Y ; GET IT
0070	0218	85 03		STA \$03
0075	021A	20 ** **		JSR INCPTR ; NEXT...
0080	021D	B1 00	NEXT	LDA (\$0),Y ; GET TEXT
0085	021F	20 ** **		JSR INCPTR
0090	0222	C9 0D		CMP #\$0D ; CR?
0095	0224	F0 E4		BEQ BEGIN ; END OF LINE
0100	0226	C9 20		CMP #\$20 ; CONTROL CHAR?
0105	0228	30 ** **		BMI PRINT ;YEP..
0110	022B	C9 61		CMP #\$61 ; LOWER CASE?
0115	022D	10 ** **		BPL PRINT ;YEP..
0120	0230	4C 1D 02		JMP NEXT ;LOOP BACK
0125	0233	A5 02	PRINT	LDA \$02 ; 1ST BYTE
0130	0235	20 3B 1E		JSR PRTBYT
0135	0238	A5 03		LDA \$03 ; AND SECOND
0140	023A	20 3B 1E		JSR PRTBYT
0143	023D	A0 00		LDY #\$0 ; RESET Y
0145	023F	20 2F 1E		JSR CRLF
0150	0242	4C 1D 02	DONE	JMP NEXT
0155	0245	E6 00	INCPTR	INC \$00
0160	0247	D0 ** **		BNE END
0165	024A	E6 01		INC \$01
0170	024C	60	END	RTS
0180	024D			.END

ERRORS = 0000

SYMBOL TABLE

PRTBYT	1E3B	CRLF	1E2F	CKVAL	0200	BEGIN	020A
END	024C	INCPTR	0245	NEXT	021D	PRINT	0233
DONE	0242						

END OF ASSEMBLY

## Chapter 7

### CHANGING ASSEMBLER/EDITOR I/O DEVICES

The assembler and editor are preset to use the KIM monitor serial input/output routines. The assembler/editor can be used with other 6502-based systems by changing the values of INPUT, OUTPUT, and ENDLIN in page zero (see Table 3). To use the editor/assembler with a non-KIM system, load the starting addresses for the INPUT, OUTPUT and ENDLIN routines for your system in the appropriate page zero locations. You should also load the default values for USRCMD and LINK. Enter the editor at location F103 rather than F100 if you are using nonstandard I/O devices.

#### INPUT Routine

This routine is pointed to by locations 00E5 and 00E6 and is called every time the assembler or editor needs a character input. The default is an address of 1E5A, the KIM GETCH routine.

#### OUTPUT Routine

This routine is pointed to by locations 00E7 and 00E8 and is called every time the assembler or editor wants to print a character. The default is an address of F6BA, a routine in the editor which checks to see if the BREAK button is depressed before going to the KIM OUTCH routine at 1EA0. If the BREAK button is depressed, the output routine clears the stack and jumps to location F14D, the editor command input point.

Note: Since the editor and assembler share the same I/O routines, depressing the BREAK button while the assembler is printing will stop the assembler and return control to the editor at its command input point.

### ENDLIN Routine

This routine is pointed to by locations 00E9 and 00EA and is called every time the assembler or editor wants to print a carriage return/line feed. The default is an address of 1E2F, the KIM CRLF routine. Note that KIM's CRLF routine also sends six null characters, which may be displayed on CRT terminals.

### Sample Modification of I/O Vectors

The TIM microcomputer kit is similar to KIM but uses different locations for its I/O routines. They are:

<u>Purpose</u>	<u>Name</u>	<u>Location</u>
TTY input	RDT	72E9
TTY output	WRT	72C6
Carriage return	CRLF	728A

So to modify the assembler/editor to work with the TIM the following page zero locations should be changed:

<u>Location</u>	<u>Contents</u>	<u>Purpose</u>
E5	E9	Input
E6	72	
E7	C6	Output
E8	72	
E9	8A	CR/LF
EA	72	

The editor should be started at F103 to preserve these locations. Appropriate values for USRCMD and LINK must also be provided.



## NOTIZEN



ALLES ÜBER MOS -

# MICRO'S!

VOM DATENBLATT  
ÜBER HANDBÜCHER  
BIS ZUM 4 - FARB - POSTER



microcomputer  
datensysteme gmbh

luisenplatz 4-d 61 darmstadt ☎ 06151/23959

**MICROCOMPUTER-HANDBÜCHER (Netto + MWSt. + Versand)**

Wir bestellen zu Ihren bekannten Bedingungen:

1.10 KIM-1 User Manual	englisch	DM	19,80	Stck: _____	DM: _____
1.11 KIM-1 Handbuch	deutsch	DM	19,80	Stck: _____	DM: _____
1.20 KIM-2/3/4 User Manual Expansion Modules					
Memory/Errata Sheet/Motherboard	englisch	DM	9,25	Stck: _____	DM: _____
1.30 KIMath Subroutines Programming Manual	englisch	DM	15,45	Stck: _____	DM: _____
1.40 KIM Text Editor User Manual	englisch	DM	8,90	Stck: _____	DM: _____
1.50 KIM Assembler Manual Preliminary	englisch	DM	10,20	Stck: _____	DM: _____
1.60 Programming Manual	englisch	DM	28,60	Stck: _____	DM: _____
1.61 Programmierhandbuch	deutsch	DM	28,60	Stck: _____	DM: _____
1.62 THE FIRST BOOK OF KIM-1					
DAS KIM-REZEPTBUCH	englisch	DM	19,80	Stck: _____	DM: _____
1.70 Hardware Manual	englisch	DM	24,90	Stck: _____	DM: _____
1.71 Hardware Handbuch	deutsch	DM	24,90	Stck: _____	DM: _____
1.80 TIM Terminal Interface Monitor Manual	englisch	DM	12,90	Stck: _____	DM: _____
1.90 Cross Assembler Manual Preliminary	englisch	DM	12,70	Stck: _____	DM: _____
1.99 Bücherliste über weitere ca. 60 Bücher mit Kurzbeschreibung		DM	--,-	Stck: _____	DM: _____

**MICROPROCESSOR DATENBLÄTTER (Netto + MWSt. + Versand)**

2.10 Spectrum of Products	englisch	DM	0,95	Stck: _____	DM: _____
2.20 MCS 6 5 0 0 Microprocessors	englisch	DM	0,95	Stck: _____	DM: _____
2.30 MCS 6 5 2 0 Microprocessors	englisch	DM	0,95	Stck: _____	DM: _____
2.31 MCS 6 5 2 2 Microprocessors	englisch	DM	0,95	Stck: _____	DM: _____
2.40 MCS 6 5 3 0 Microprocessors	englisch	DM	0,95	Stck: _____	DM: _____
2.60 MCS 6 5 0 X Instruction Set 6502-6515	englisch	DM	1,10	Stck: _____	DM: _____
2.80 MCS 6 5 3 2 Microprocessors	englisch	DM	0,95	Stck: _____	DM: _____

**MICROCOMPUTER-LOSEBLATT-BIBLIOTHEK (Netto + MWSt. + Versand)**

3.10 Loseblattsammelordner aus Kunstleder für alle Bücher und Datenblätter		DM	4,95	Stck: _____	DM: _____
3.20 MICRO-Info Hardware ca. 1400 Seiten (FMI)	deutsch	DM	42,00*	Stck: _____	DM: _____
3.23 MICRO-Info Software ca. 400 Seiten (FMI)	deutsch	DM	38,00*	Stck: _____	DM: _____
3.26 Kontinuierliche automatische Ergänzung zur (FMI) 2-monatlich ca. 150 Seiten, 10 DM/S	deutsch	DM	15,00	Abo: _____	DM: _____

\* ) Gilt nur in Verbindung mit Abo, sonst 84,-/76,- DM.

**MICRO-COMPUTER-POSTER UND PROSPEKTE (Netto + MWSt. + Versand)**

4.10 KIM-1 Microcomputer Module, 4-Farb-Poster Schaltschema <input type="checkbox"/> = ungefaltet <input type="checkbox"/> gefaltet A 4		DM	2,95	Stck: _____	DM: _____
9.00 MCDS-HARDWARE + SOFTWARE PRO- DUKTE Schnell-INFO		DM	0,80	Stck: _____	DM: _____

**COMPUTERJOURNALE (Enduserpreise incl. MWSt. + Porto)**

11.10 JOCE & N: Europas einziges Computer- journal in deutsch und englisch für Hobby, Forschung und Ausbildung	englisch/deutsch	DM	4,00	Stck: _____	DM: _____
	Abonnement	DM	40,00	Stck: _____	DM: _____
11.11 KILOBAUD: Das bekannte US-Journal des MICRO + MINICOMPUTERMARKTES	englisch	DM	7,50	Stck: _____	DM: _____
	Abonnement	DM	75,00	Stck: _____	DM: _____
11.13 INTERFACE AGE: Ebenso bekanntes wie beliebtes US-Computer- Journal	englisch	DM	6,00	Stck: _____	DM: _____
	Abonnement	DM	60,00	Stck: _____	DM: _____
11.15 THE NEW HOBBY COMPUTERS, der BESTE Jahresextrakt	englisch	DM	15,75	Stck: _____	DM: _____
11.16 THE HOBBY COMPUTERS ARE HERE, der BESTE Jahresextrakt	englisch	DM	15,75	Stck: _____	DM: _____
11.17 „73“ MAGAZINE AMATEUR RADIO für Funkamateure in der BRD, 3500 Abos	englisch	DM	7,00	Stck: _____	DM: _____
	Abonnement	DM	70,00	Stck: _____	DM: _____

**COMPUTER CLUB EUROPE E.V.**

21.00 MITGLIEDSCHAFT, kostenfreien Bezug des JOCE & N + europaweite Kommunikation	Beitrag/Jahr	DM	85,00	<input type="checkbox"/>	_____
Rabatte auf Soft- und Hardware und vieles mehr.	Student/Jahr	DM	45,00	<input type="checkbox"/>	_____

Alle Preise verstehen sich netto fob Darmstadt.

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_

Straße: \_\_\_\_\_

Postleitzahl-Landeskennbuchstabe: \_\_\_\_\_ Ort: \_\_\_\_\_

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_



**MOS**  
MOS TECHNOLOGY, INC.



**microcomputer  
datensysteme gmbh.**

**loisenplatz 4  
d 61 darmstadt  
☎ 0 61 51/23959**



**This was brought to you  
from the archives of**

**<http://retro-commodore.eu>**