

Kangaroo Software E.R.S.

This document describes the Kangaroo
operating system. Rev 5&6 software releases.

Author: Tony Ridolfo
HP-Corvallis

Date of this version <08/06/81>

Table of Contents

1	Introduction	
2	System Modes	
3	Keyboard and Display	
3.1	Keyboard	3-1
3.2	Calculator vs. Program Entry	3-3
3.3	Keyboard Layout	3-3
3.4	Lines and Windows	3-3
4	Editing Features	
4.1	Special Keys	4-3
5	Files	
5.1	Kangaroo memory	5-1
5.2	Workfiles	5-1
5.3	File Commands	5-2
5.4	Error Messages	5-4
5.5	The keys file	5-5
5.6	Other Files	5-6
5.6.1	The Calculator Variables Subfile	5-6
5.6.2	Lex and Rom files	5-7
5.6.3	DATA files	5-8
6	POWER ON and ATTENTION KEY	
6.1	Definitions	6-1
6.2	State Switching	6-2
6.2.1	Light Sleep to Deep Sleep	6-2
6.2.2	Light Sleep to Awake to Light Sleep	6-2
6.2.3	Deep Sleep to Awake	6-2
6.3	Sleeping Pil	6-3
6.4	Low Battery and Dead Battery	6-3
6.5	The ATTN key	6-4
6.5.1	Uses	6-4
6.5.2	Locking the Machine	6-4
6.6	System Reset	6-5
7	APPOINTMENT MANAGEMENT	
7.1	Overview	7-1
7.2	Commands	7-2
7.3	Filling in the Template	7-3
7.4	Repeating Alarms	7-3
7.5	Appointment Processing	7-5
7.6	Error Messages	7-6
8	Time Mode	
8.1	Time Template	8-1

<Kangeroo E.R.S.>
<08/05/91>

8.2	Time Commands	8-1
8.3	Timer Accuracy	8-2
8.4	BASIC Time Functions	8-3
9	PIL	
9.1	General PIL Commands	9-1
9.2	Device Dependent Data	9-3
9.3	Error Messages	9-3
10	Card Reader	
10.1	Specifications	10-1
10.2	Commands	10-1
10.3	Error Messages	10-3
11	BASIC	
11.1	Numeric data	11-1
11.2	Character string data	11-2
11.3	Arithmetic Operators	11-2
11.4	Relational operators	11-2
11.5	Logical or Boolean Operators	11-3
11.6	Menu of commands and BASIC Statements	11-3
11.7	Numeric Functions	11-13
11.8	String Functions	11-14
11.9	Print Functions	11-14
A	Compatability with 95A	
B	Command Abbreviation	
C	Key Codes	
D	Error Messages	

Introduction	CHAPTER 1
--------------	-----------

The overall product objectives of the Kangaroo project are:

- a) BASIC language portable computer.
- b) Selling price between \$700 and \$1500 for base machine.
- c) Utilization of low risk technologies to insure high production volume.
- d) Timely product introduction (FY '82).

The major features of Kangaroo are summarized below:

- a) HP-85 compatible BASIC (ANSI minimal) extended to ANSI Level 1 where possible.
- b) Totally CMOS memory, 8K or 16K bytes RAM
- c) Typewriter keyboard.
- d) 32 character full alphanumeric LCD display, with four annunciators.
- e) Manual card reader for off-line storage of programs or data (1.3K bytes/card).
- f) Built-in HPIL interface.
- g) Battery operation (rechargeable nicads).
- h) Three slots for plug-in ROMs (8K or 16K bytes each).
- i) One slot for add-on RAM (8K bytes).
- j) Integration of a real-time clock and alpha storage into a "pocket secretary".
- k) Advanced file system that supports multiple files and file types in RAM at the same time.
- l) Totally redefinable keyboard.

The following chapters describe the various features of the Kangaroo software and the user commands.

System Modes	CHAPTER 2
--------------	-----------

Kangaroo has four distinct modes of operation. They correspond to the various operational cases of the machine, and will be referred to as:

- a) Edit mode (either text or BASIC)
- b) program running mode
- c) TIME mode
- d) Appointment mangagement mode

Edit mode refers to the mode that Kangaroo is normally in for creating BASIC programs, creating text files, or doing calculations.

If the user is editing a BASIC file, the prompt is the greater than symbol ">". If the edit file is pure text, the prompt is a colon ":". (When editing a text file, any numbered line is considered pure text, embedded blanks are left in and no parsing is done. If a line does not have a line number, what is entered is parsed and executed.)

Program running mode refers to the actual running of a BASIC program. We distinguish it from the others since the keyboard is ignored while a program is executing unless the program is specifically asking for input. The ATTN key is the only active key when input is not expected, and acts as a break key. While a program is running, there is an annunciator lit in the display.

In TIME mode the current date and time are displayed in the LCD. The user will be able to set or adjust the time in this mode.

The appointment management mode refers to the pocket secretary functions of Kangaroo. Appointments and messages can be entered and scheduled in this mode. This mode is characterized by the appointment template in the display.

Pressing the TIME, APPT, or EDIT key switches Kangaroo into the mode pressed. The ATTN key turns the computer on and does various other housekeeping functions.

Keyboard and Display	CHAPTER 3
----------------------	-----------

3.1 Keyboard

The keyboard is capable of generating all 128 ASCII keycodes, representing letters, digits, punctuation, and control functions. Both upper and lower case letters are available. Normally, the keyboard is lower cased, i.e., all letters are lower case and the punctuation and digit keys are unshifted. Pressing SHIFT will shift the alphabetic keys to upper case and the digit and punctuation keys to upper case.

Pressing SHIFT-LOCK switches the alphabetic keys to all upper case characters. Unshifted characters are entered by pressing the SHIFT and any key simultaneously. Pressing LOCK returns to the typewriter keyboard.

The user can also enter Greek or special characters: pressing the CTL (control) key while simultaneously pressing a non-numeric key generates the special characters. The table below summarizes the action of CTL. However, use of these characters in a program conflicts with standard usage of control characters. Extreme care should be used, especially if Kangaroo is interfaced to a peripheral which interprets control codes embedded in data as control. The Kangaroo manual and other literature will discourage usage of control codes for program documentation.

By pressing the CTL and LOCK keys simultaneously, the alpha keys are either non-entry keys or a special "number pad". To get the regular alpha keys (CAPS mode only) the user presses the SHIFT and an alpha key simultaneously. To exit this mode the user presses the LOCK key (and returns to normal).

<Kangaroo E.R.S.>
 <08/06/91>

<< Standard Entry >>

KEY	SHIFT	CTL
a	A	deg
b	B	x)
c	C	
d	D	2
e	E	3
f	F	4
g	G	5
h	H	6
i	I	7
j	J	8
k	K	9
l	L	0
m	M	cr
n	N	2
o	O	3
p	P	4
q	Q	5
r	R	6
s	S	7
t	T	8
u	U	9
v	V	0
w	W	1
x	X	2
y	Y	3
z	Z	4
space		
1	!	
2	"	
3	#	
4	\$	
5	%	
6	&	
7	'	
8	(
9)	
0	[
-]	
+	@	
BACK		ESC
*	^	
;	:	
,	<	
.	>	
/	?	
=		

<< CTL-LOCK Entry >>

KEY	SHIFT
no entry	A
"	B
"	C
"	D
"	E
"	F
"	G
"	H
5	I
1	J
2	K
3	L
0	M
no entry	N
6	O
no entry	P
"	Q
"	R
"	S
"	T
4	U
no entry	V
"	W
"	X
"	Y
"	Z

3.2 Calculator vs. Program Entry

The keyboard can be used to perform calculator functions as well as entering a BASIC program or text file. To calculate the square root of 543.567 the user types

SQR(543.567)

and presses the RTN key. When Kangaroo encounters the RTN command, it will examine the statement in the display to determine if it is a BASIC statement (preceded by a line number), a system command, an edit command, or a calculator function. All statements except BASIC program statements entered in edit mode are executed immediately. BASIC statements are stored for later execution. In text mode if the command is preceded by a line number it is stored as text. If not, it is executed. In order to do arithmetic when in text mode, if the first "character" is a number, precede it with a plus (+) sign.

3.3 Keyboard Layout

The keyboard layout is shown in the attached figure.

3.4 Lines and Windows

A line in a Kangaroo contains up to 96 characters. Any 32 consecutive characters may be selected and displayed at any time. This is referred to as the window on the line. (This is a single line analog of the 85's 32 character CRT line and 96 character logical line.) Editing keys are provided to move and edit the part of the line in the display (see the next chapter).

The total length of the LCD buffer is 96 characters. On input there is usually a prompt, generated either by the operating system or the user in a running program. This prompt, be it the ">" or ":" or "?" or "What is your name? ", takes space away from the 96 character input buffer. Kangaroo will BEEP when there are five spaces left in any input line. After the input buffer is full, the content of the line will not change and the cursor will stay at the last position.

Editing Features	CHAPTER 4
------------------	-----------

Nine keys are provided for editing and changing the line in the LCD display. These keys are labeled: \uparrow , \downarrow , \leftarrow , \rightarrow , BACK, I/R, FET, DEL, and CLR.

In the discussion below, the term "line" refers to the 96 character logical line which Kangaroo sees as a single line. The 32 characters which can be seen in the LCD by the user at any time are referred to as the "window" or "display".

UP-ARROW : this key will bring the previous file line into the display. In essence it goes "backwards" in a file to the next lower numbered line. In APPT mode, the \uparrow key is used to search backwards in time in the appointment file.

SHIFT \uparrow : Brings the first line of the current file into the display.

CTL \uparrow : Sends out an $\{S$, i.e., roll up, to the current display device. Note, the Kangaroo LCD driver ignores this key sequence.

DOWN-ARROW: this key will bring the line with the next higher line number into the display. If no such line exists, the display will show the last line in the file. In APPT mode, the \downarrow key is used to search forward in time for appointments.

SHIFT \downarrow : Brings the last line of the current file or program into the LCD.


CTL \downarrow : Sends out an $\{T$, i.e., roll down, to the current display device. Kangaroo ignores this sequence.


LEFT ARROW: This key moves the cursor to the left until the left edge of the display is reached. Holding the \leftarrow key down when the cursor is already at the left edge will scroll the string in view to the right if there is more to the line than is currently being displayed.

SHIFT \leftarrow : Positions the cursor at the left edge of the line.

CTL \leftarrow : Positions the cursor 32 places to the left of its current position or at the left edge of the display.

RIGHT-ARROW: This key moves the cursor to the right until the right edge of the display is reached. If the cursor is already at the right edge of the display, the displayed string will scroll to the left until the end of the string is reached.

SHIFT  : Positions the cursor at the far right edge of the display.













CTL  : Positions the cursor 32 places to the right of its current position or at the right edge of the display.

BACK : The BACKspace key moves the cursor to the left one character at a time and erases the character at the old cursor position. This key has no effect if the cursor is at the left edge of the line. For those modes with protected fields (APPT and TIME), the protected fields will not be erased. (Note: this key acts the same as CTL-H.)

SHIFT-BACK: This is the same as BACK.


CTL-BACK : this is the ESCape key.

Caution: the LCD driver software recognizes standard escape sequences. These are:

	right arrow
	left arrow
	clear the LCD, and home cursor
	cursor to far left
	home cursor
	clear the LCD (homes cursor also)
	clear to end of line
	delete with wraparound (normal Kangaroo delete)
	delete without wraparound
	cursor off
	cursor on
	%c,r position cursor to column, row with row ignored

Hence, displaying escape sequences in the LCD will cause it to perform the indicated operation.

I/R : The Insert/Replace key is used to toggle the display in and out of character insert mode. Pressing the I/R key during normal entry mode (replace mode) changes the cursor to the insert cursor: a blinking triangle. It is used to mark the insertion spot and to indicate that Kangaroo is in insert mode. Inputting characters from the keyboard will cause the display, starting at the insert cursor, to move to the right, one character at a time, and the input characters to appear in the vacated positions.

Pushing  while in insert mode moves the insert cursor to the

left. Similarly, → moves the cursor to the right.

Backspacing in insert mode causes the character immediately to the left of the insert cursor to be deleted and the line to move in from the right filling the vacated position. The rationale is that the user most likely wants to delete or correct a character just inserted.

Pressing the I/R key while in insert mode returns the cursor to the normal replace mode: a flashing rectangle. Pressing ↑, ↓, or RTN while in insert mode will terminate the insert and perform the requested operation.

SHIFT-I/R: This is the literal escape key. Pressing it causes the next key to be displayed as its default definition, rather than any redefinition or implied action.

DEL : The DELETE key removes the character appearing at the current cursor position. The part of the line to the right of the cursor moves to the left one position to occupy the vacated position.

SHIFT-DEL: Clears that part of the line starting at the current cursor position and ending at the far right edge of the line. In appointment mode, SHIFT-DEL deletes the currently displayed appointment.

CLR : The CLR key is used to clear the displayed line and positions the cursor at the left edge of the display. Note this does not delete the current line, but only clears the LCD and input buffer.

4.1 Special Keys

The following key sequences are also defined:

FET : This key acts as a typing aid to bring program or text lines into the display. When pressed while in an workfile, the word FETCH appears and the user has several options:

FETCH <line number>	returns that numbered line
FETCH	returns the current line
FETCH <'ed string>[,<line number>]	searches memory for first occurrence of <string> starting at the given line number
FETCH key <'ed key>	returns the softkey definition of <key>

<Kangaroo E.R.S.>

<08/06/81>

SHIFT-FET: This sequence brings into the display the previous ERROR message while held down. Releasing the key brings back the line causing the error with the cursor positioned at the first character which caused the error.

CTL-FET : This brings back into the display the previous input line. Normally, CTL-FET will be used to retrieve an algebraic expression which was just evaluated.

SHIFT-RUN: This key sequence acts like a single-step key on a handheld calculator. While held down the next line to be executed is displayed. When released, the line is executed and any output necessary is generated. If there is no output statement, the line remains displayed. The program is then stalled at this line. Execution can be CONTinued, S-RUN again, or RUNned from that line number.

TAB : This key moves the cursor across protected fields in the display. It is useful in TIME and APPT modes and if a program has a protected input template. (No user mechanism will be provided in the mainframe to create protected fields. This feature will be added in an advanced programming ROM.)

SHIFT-TAB: This does a back tab across protected fields.

Files	CHAPTER 5
-------	-----------

5.1 Kangaroo memory

The essence of the Kangaroo computer is its file handling system. A file is nothing more than a contiguous area of memory which may be manipulated by the user. The operating system is designed to make the actual file operation transparent to the user. The use of CMOS memory enables Kangaroo to retain all programs or text that the user inputs. The following sections describe the file structure implemented to help the user manage his programs and data.

5.2 Workfiles

The Kangaroo operating system and internal data structure allow multiple files and file types in RAM. The user can manipulate these files in various ways and the purpose of this section is to illustrate the user/machine interface for handling files. The goals of the software interface are:

- a) to encourage economy of RAM space
- b) to keep number of commands small
- c) to not sacrifice power or mnemonic value.

So, how does the user create a file and use it? After the initial setting of the system clock (when the batteries are first inserted), the user should press the EDIT key. The user then sees the ">" prompt, indicating that Kangaroo is pointing to a file of type BASIC. It is called the WORKFILE. Any numbered line is saved in this file automatically. Moreover, if the file is named, all additions and changes are automatically saved when a new file is called up. If the file was not named, attempting to edit another file will cause an error. The current WORKFILE must be either purged or named before another file can be edited.

Files are referenced by a 'file specifier' or 'filespec'. The

syntax of a file specifier is:

<filename> [:device [/password]]

where

filename is a string of from one to eight alphanumeric characters or the period
:device is an optional two character device mnemonic referencing where the file resides if not internally
/password is an optional four character password for file security good only for external files

All lowercase letters entered in a filespec are converted to uppercase.

There are two system files that the user can manipulate:

keys - the file in which redefined keys are kept
appt - the appointment file

These two files are not quoted when accessed since they have reserved names.

There are two special device names:

:CARD - card reader
:PCARD - card reader with private card option.

Refer to the chapter on the card reader for more details.

5.3 File Commands

The commands for manipulating files within Kangaroo are described below. In this description we use

<file> - for filespec
<line #> - for line number
| - for alternate syntactical choice

All commands are programmable.

CALL <file> - Will allow one program to call another as a subroutine with embedded END statements acting as returns. The mainframe will not support parameter passage, but the Advanced

<Kangaroo E.R.S.>
<08/06/81>

Programming ROM will. Parameters can be passed via the ASSIGN#, PRINT#, and READ# commands.

CAT# <number> - returns a string containing the catalog of the numbered file. A negative number returns the currently running file, a '0' returns the current workfile, a positive integer returns that file in the catalog, or a null string if there are not that many files in the catalog.

CAT [<file>] | CARD | ALL - Gives a catalog listing showing file name, type, length, and time and date of creation. If just CAT is given, the catalog is of the current workfile; if a file name is given, that file only is cataloged; if ALL is given, all files in the system are cataloged, and the user can scroll through the Catalog using the UP and DOWN ARROW keys. In any case, pressing any other key returns the user to the current workfile.

COPY [<file>] TO <file> - Copies the first file to the second file. Either filespec can be replaced with the word CARD meaning the card reader.

EDIT BASIC - Brings in an unnamed workfile of type BASIC with the ">" prompt.

EDIT TEXT - Brings in an unnamed workfile of type TEXT with the ":" prompt.

EDIT <file>, type - Brings in the file named of type given if it exists and the type matches. If the file exists but is not of the given type, then an error is generated. If the file does not exist, an empty file of that type and name is created.

EDIT - Brings in a new empty workfile of type whatever the current prompt is.

LIST [<file>][,<line #>[,<line #>]] - The first parameter is the file to be listed (on the current DISPLAY IS device). If omitted, then the current workfile is listed. The second parameter is the starting line for the listing. The last parameter is the number of the last line to be listed. If omitted, then this defaults to the second parameter; thus listing only the given line. If both the line numbers are omitted, then the third parameter is defaulted to the last line in the file; thus listing the entire file.

MERGE <file>[,<line #>[,<line #>]] - This merges the named file into the current workfile. The optional line numbers work as in LIST. The user should in fact think of MERGE as LISTing one file into another. If the current workfile into which one is merging contains identical line numbers as the file being merged in, the

current workfile lines will be overwritten.

NAME <file> - This renames the current workfile to the given name and then gives the user a new unnamed workfile. NAME is equivalent to

RENAME TO <file> @ EDIT

PLIST [<file>][,<line #>[,<line #>]] - This is the same as LIST except to the current PRINTER IS device. Also, if the file named is a TEXT file, the line numbers are not output to the print device; thus giving an unnumbered listing.

PURGE [<file>] - Purges from memory the named or current (default name) file.

RENUMBER [<start line#>[,<increment>[,<old start line#>[,<old final line#>]] Renumbers the current workfile. Default is to renumber the entire file by tens starting at line 10. The first parameter permits starting other than at line 10; the second permits increments other than ten; the third and fourth parameters permit renumbering parts of the workfile. If a section only is to be renumbered, but there would be a collision with existing lines outside the range of section to be renumbered, then no renumbering is done. If the renumbering would cause overflow of line numbers, then the entire file is renumbered by ones starting at line one.

RENAME [<file>] TO <file> - This renames the first file to the second name. If the first file is omitted, the current workfile is renamed.

RUN [<file>][,<line #>] - The first parameter is the name of the file to be RUN (TEXT files cannot be RUN). The second is the starting line number, and defaults to the beginning of the file.

5.4 Error Messages

The error messages relating to files and their probable causes are:

Error #	Meaning	Probable Cause
60	Invalid file name	Used a system file name
61	Duplicate file name	
62	File does not exist	

63	Not enough memory	Too big, or too much
65	Invalid access	File is private
66	Invalid password	
67	Invalid line length	
68	Invalid file type	
69	workfile not named	Need a RENAME TO

5.5 The keys file

A special file is the "keys" file. This file contains the user defined "softkeys". Due to the differences in hardware, the Kangaroo DEF KEY statement works differently than the HP35A ON KEY# statement. The 85 used four special keys which could generate an interrupt in a program and a branch could then be made. Kangaroo has no such structure. Hence, our keys become typing aids and can be defined as follows:

To redefine a key type

```
DEF KEY "<key>","defining string"[;]
```

The defining string can be any string or string expression. If the optional semicolon is present, the defining string becomes a non transmitting key; i.e., whenever that key is pressed, the defining string is put into the display, but no action is taken until a RTN is pressed. On the other hand, if the semicolon is omitted, then the key definition becomes a transmitting key, i.e., as if there were a carriage return at the end of the string. Now whenever that key is pressed, the defining string is executed.

To change the DEF KEY definition use the literalize sequence (SHIFT-I/R):

```
DEF KEY "lit<key>","new defining string"[;])
```

For example, suppose the user types:

```
DEF KEY "P","plist"
```

Then every time he presses P, the computer will see PLIST instead and try to do that. Now suppose he wishes to undo this definition, he would type:

```
DEF KEY "litP","litP";
```

All but four keys may be redefined in this manner including the top row keys. They are ATTN, EDIT, RTN, and the literalize

sequence itself.

By using the literalize sequence, the back arrow key, and the insert key, unique typing aids can be created. For instance, the following sequence will create a typing aid that allows the user to type in the name of the file and it will be copied from a card:

```
DEF KEY 'CTL c','Copy card to "'lit litI/R';
```

pressing CTL-c results in the display:

```
Copy card to ' '
```

The file "keys" can be examined, listed, edited, renamed, renumbered, and copied like any other user file. If listed, the line numbers are the equivalent ASCII value of the key.

5.6 Other Files

Besides program files, key files, the appointment file, and text files, there is the system file and lex files. Lex files are assembly language files that reside in either RAM or ROM and that extend or modify BASIC commands or functions. The table below describes these files as they appear in the catalog:

File Type	Type	Access	Existence
appt	A	all	When appointments exist
keys	T	all	When keys redefined
<lex>	L	run only	When copied in from card or on Rom
system	S	op sys only	Always
<userprgm>	B	all	When named
<usertext>	T	all	When named

Those files that the operating system needs to do its work are lumped together into the system file. Included in this file is the device table for PIL, the assign# file pointers, and calculator variables.

5.6.1 The Calculator Variables Subfile

Calculator variables reside in a subset of the 'system' file called the VARS file. This file may be CLEARED, but no other file manipulations are allowed. Whenever a variable is assigned

outside of a running program, it is saved in this subfile. For example, if the user types

```
x=sin(3.45)/exp(2.5)
```

then the value of "x" is saved in this file until either the file is cleared or the variable x gets a new value. Calculator variables can be used to do calculations outside of a running program.

To see the power of the VARS file, suppose the user types

```
A = 6.02e23    (Avogadro's number)
```

This variable now exists in the VARS file and is accessible whenever no program is running or stalled (stopped without doing an END), to be used in calculator computations. Now suppose a program is run in which A is assigned the value 3.14. If the program is stopped by the ATTN key, and the user types A he will get 3.14. He may alter this variable A in the program and resume the program with this new value. After the program ends, if he types A, he will get 6.02E23. In other words, program variables may be interrogated and altered independent of calculator variables when a program is stalled via ATTN or STOP. Garbage collection is done by the operating system and is invisible to the user. The VARS subfile, however, is permanent and grows whenever a new variable is assigned a value outside of a running or stalled program. The command to clear this file is

```
CLEAR VARS
```

which purges the calculator variable file from memory.

5.6.2 Lex and Rom files

Lex files are files that contain new commands, e.g. if there were an assembly language file on a mag card that defined a new BASIC function, it would be referred to as a lex file. Files of programs or text that reside in plug-in Roms are called Rom files. Kangaroo is unique in that BASIC and text files may be put into Roms as well as assembly language. The ability to put assembly language files onto a card is envisioned only via the KEOS software development program. These files appear in the catalog as indicated, and allow customization and expansion of the operating system.

5.6.3 DATA files

Kangaroo does not distinguish DATA files from program files. In fact, DATA files are really BASIC programs which contain DATA statements. They can be created like any other BASIC program file, or by BASIC programs themselves by use of the ASSIGN# and PRINT# commands. Below are listed the commands used to create and access DATA files. However, since DATA files are nothing more than BASIC programs made up of DATA statements, any BASIC program may be considered as a DATA file for another program if it contains DATA statements or if a PRINT# writes to it.

ASSIGN#<num exp> TO <string expression> - where the <string expression> is a filespec, opens an existing file or creates one of the given filename.

ASSIGN#<num exp> TO * - closes the referenced file. (THIS IS A CARRYOVER FROM THE 85 AND IS NOT NEEDED IN KANGAROO)

PRINT#<num exp>;<print list> - writes data serially to the referenced file.

PRINT#<num exp>,<line num>;<print list> - writes data "randomly" to the file referenced to the line number given in the second parameter. If the line exists, it is overwritten; else, it is inserted into the file. Note that line numbers are in the range 0 to 9999 and the length of a record (or line) is at most 255 bytes. However, the display can only show 96 characters, and an error will be reported if the user tries to edit or list such a line.

READ#<num exp>;<read list> - serially reads from the referenced file into the read list. Note, the read starts at the first DATA statement in the file.

READ#<num exp>,<line num>;<read list> - reads starting from the given line number, if it is a DATA statement.

RESTORE#<num exp>[,<line num>] - does a RESTORE in the respective ASSIGN# file.

POWER ON and ATTENTION KEY	CHAPTER 6
----------------------------	-----------

6.1 Definitions

This chapter explains the power on/off protocol. There are several terms used to describe the power state and clock speed of the hardware and software, and though they seem synonymous in most respects, they are not always so. The first term is SPEED. It refers to the system clock and there are three states:

FULL SPEED: The system is fully powered up and the clocks are running at 613 KHz.

SLOW SPEED: The system is powered up, but the clocks are at 4.6 KHz to conserve power.

ZERO SPEED: The system clocks are turned off, but the real time clock and the RAM refresh circuitry are active.

In addition to the speed of the internal clocks is the AWAKE/ASLEEP terminology:

AWAKE: This is the same as FULL SPEED and except for the time it takes to process a key, is generally only when executing a command, reading a card, driving the beeper, running a program, or using PIL.

LIGHT SLEEP: This is any time the system is in SLOW SPEED waiting for input. The LCD, keyboard and all other interrupts are active in this state.

DEEP SLEEP: This is any time the system is in ZERO SPEED. The LCD is off. The only active key is the ATTN key. The comparator is active, but the appointment processing may or may not be active (see ALARM ON/OFF below).

DEAD: The batteries are dead, no power is being supplied to the system and everything is off.

6.2 State Switching

Going from one state to the next is automatic and transparent to the user. This section describes the state transition sequences.

6.2.1 Light Sleep to Deep Sleep

This occurs when the user types the word BYE, or presses SHIFT-ATTN, or by the software when no program, keyboard, clock, or I/O activity is detected for approximately five (5) minutes.

6.2.2 Light Sleep to Awake to Light Sleep

This occurs automatically when the system processes a keyboard, clock, or I/O request. As soon as the processor is done, the system is sent back to light sleep after a short (approximately 3 seconds) wait.

6.2.3 Deep Sleep to Awake

This transition can occur either by the user pressing the ATTN key or by an appointment coming due. The ability of the real time clock to turn on the machine can be overridden with the command

ALARM OFF

The triggering of appointments is turned off and can only be turned on by the command

ALARM ON

The other Appointment commands are described in the next chapter.

Timers may also wake the computer up if a BYE command was executed in a running program after a timer had been enabled. The program will start executing the timer command(s) in a normal manner.

6.3 Sleeping Pil

The PIL transformers consume a lot of power, and the PIL interrupt cannot wake Kangaroo up from DEEP SLEEP. The following command keeps Kangaroo from timing out to DEEP SLEEP:

STANDBY ON

This disables the timeout and does not allow the LIGHT SLEEP to DEEP SLEEP transition to occur without the user typing BYE or pressing SHIFT-ATTN. In this mode the PIL, if assigned, remains powered up ready to process data. To undo, type

STANDBY OFF or
BYE or
SHIFT-ATTN

6.4 Low Battery and Dead Battery

Kangaroo switches from AWAKE or LIGHT SLEEP to DEAD in the following sequence.

- 1) The software detects a low power signal from the hardware and the low battery annunciator is lit in the display.
- 2) The system sets up a lower level battery detect and continues to operate normally, with the annunciator on.
- 3) When the power level falls, the message "Low batteries" is sent to the display and all processing is halted. The system then goes to DEEP SLEEP. Any attempt to turn it on only results in the above message and going back to DEEP SLEEP. Note, since DEEP SLEEP turns off the LCD, the low battery annunciator is turned off also.

Kangaroo remains in this state until the power level goes back up above this lower detect level or the batteries finally expire. This is step 4:

- 4) The DEAD mode is entered when the power drops below the threshold needed to maintain the time-of-day clock and the RAM refresh circuitry.

The DEAD state switches to the AWAKE mode when fresh batteries are inserted and the user presses the ATTN key. The software then reinitializes RAM and asks the user to set the time (see Chapter 8).

6.5 The ATTN key

6.5.1 Uses

The ATTN key has several uses:

- a) Wake Kangaroo up when in DEEP SLEEP.
- b) Stop a running program.
- c) Stop a listing.
- d) Stop auto line numbering.
- e) Acknowledge an appointment in APPT mode.
- f) Clear the LCD and/or any error condition.
- g) Break card reader operations.
- h) Break PIL operations.

In essence the ATTN key is the system BREAK key. If in EDIT mode, whatever the system was doing is halted and Kangaroo is ready to accept a new command. (The user is saying in effect, "Machine, listen to me now.")

If in DEEP SLEEP, pressing the ATTN key returns the user to the edit file last in. The only time this does not happen is if the machine went into DEAD state while asleep. As noted above, the user is put into the SET TIME template in this case.

6.5.2 Locking the Machine

If a program is running but is not looking for input from the keyboard, all keys except the ATTN key are ignored. To get the same effect when no program is running, and to prevent unauthorized use of the machine, a mechanism exists for locking the keyboard. The syntax is:

LOCK <string expression>

The string expression becomes a 'password' that the user MUST give whenever the machine comes up from DEEP SLEEP, i.e.,

whenever the user presses the ATTN key to turn on Kangaroo. In this case the message

 Password?

is displayed, and the user must type in the password exactly as it was given in the LOCK command. This is an absolute secure; i.e., if the password is forgotten, then the only way to wake the machine up is to remove the batteries. Of course, all data, programs and appointments will be lost.

To cancel the PASSWORD, type LOCK '' or LOCK <new string>. The null string cancels the lock.

6.6 System Reset

Since CMOS is susceptible to latch up and the software may lock up, there are two system resets available. One is to remove the batteries. If the computer is on, removing the batteries for 3 seconds will cause a hard reset. (Removing the batteries when in DEEP SLEEP will not do a reset however; allowing the user to replace them and not lose the contents of memory. The distinction is whether or not the machine is in DEEP SLEEP. A latch-up condition that does not respond to the key sequence below will occur when the machine is either full awake or in light sleep.) The other is to press simultaneously

 SHIFT-CTL-CLR and hold for one second.

This causes a system interrupt and sends the software to the initial coldstart location of ROM. Currently, this is a software reset, and hence, if for some reason the keyboard is disabled, this will not work and the user will have to pull the batteries. Either method is fatal to all user files.

APPOINTMENT MANAGEMENT	CHAPTER 7
------------------------	-----------

7.1 Overview

The pocket secretary features of Kangaroo are implemented in the appointment management part of the software. The user has the ability to:

- *Store appointments and notes in a built-in electronic calendar
- *Set up alarms and be notified when an appointment is due.
- *Search the calendar for given items, dates, and messages.
- *Have an appointment execute any BASIC command.
- *Run BASIC programs according to a time table stored in the calendar.
- *Display the current date and time continuously.

There is an annunciator in the LCD which will light whenever an appointment comes due. Pressing the APPT key will then display the currently due appointment. If no appointment is due, pressing the APPT key will then display the following template:

Day Mo/Dy/Yr Hr:Mn AM #IN !Note

where:

Day is day of week
Mo is month
Dy is day of month
Yr is the year
Hr is the hour
Mn is the minutes
AM is AM, PM, or ** for 24 hour clock
stands for alarm number
IN is a code for the type of appointment
!Note is the user's message or command

The above fields are semi-protected, i.e., the user will not be

able to delete the separators (/ or : or #) nor the Yr if in year mode. Kangaroo will automatically figure out any field that is not explicitly specified.

7.2 Commands

The editing keys for manipulating files have slightly modified action in the appointment file:

APPT : when the APPT key is press the above template is put into the LCD unless an appointment is due, in which case the due appointment will appear. Pressing the ATTN key acknowledges the due appointment, and the next past due appointment, if any, is displayed. Otherwise the template is displayed.

CLR : clears the LCD and redisplay the template.

RTH : causes Kangaroo to compute any needed values not filled in, check for day/date mismatches, store the complete appointment in the appointment file, and display the complete appointment.

SHIFT UP-ARROW: positions the cursor at the start of the appointment file.

SHIFT DOWN-ARROW: positions cursor at the bottom of the file (most future appointment).

UP-ARROW: moves backwards in time through the appointment file.

DOWN-ARROW: moves forward in time.

SHIFT APPT: allows the user to see the interval for repeating alarms, and extended year information, while held down.

SHIFT DEL: if the display is unchanged, deletes the current appointment. If the display is altered and then SHIFT DEL is press, the old appointment is edited to the currently displayed appointment and replaced.

Past appointments appear in the LCD underscored, whereas appointments yet to go off are not underscored.

7.3 Filling in the Template

To facilitate filling in the template, the user is not required to complete all parts of the template. Areas not filled in have default values. For example, if the default template is in the display, the time will default to midnight if the day is changed; otherwise, it is the current time rounded up to the next minute. Days may be omitted and the date put in. Kangaroo will fill in the correct day of the week. Similarly, filling in the day of the week but not the date is permissible. Kangaroo will figure out the correct matching date. If the user wants to make an appointment for next Tuesday, say, he types Tue in the day field, and Kangaroo fills in the rest. Of course, the user must still insert the time and message.

The appointment management software tries to match the correct date and day, if both are given, to the end of the next year. If such a match does not occur, Kangaroo gives an error unless the user had previously set the EXTEND status in time mode. In this case, Kangaroo searches into the future until it finds a matching year with two asterisks (**) placed in the year field if the match is not in the current century. Pressing SHIFT-APPT brings into the display the year of the appointment.

7.4 Repeating Alarms

The code for the type of appointment wanted is included in the digit and letter after the bell character in the LCD. The digit refers to the level of alarm:

- 0 no beep
- 1 single short beep
- 2 single long beep
- 3 two tone alarm pattern
- 4 rapid sequence of beeps
- 5 long low tone followed by long high tone
- 6 siren sound
- 7 repeat 2 every 15 seconds until acknowledged
- 8 repeat 4 every 15 seconds until acknowledged
- 9 repeat 6 every 15 seconds until acknowledged

The letter choices and their meanings are:

<Kangaroo E.R.S.>
<08/06/81>

N-normal appointment, deletes only if user presses
SHIFT-DEL
R-repeating appointment, reschedule immediately after
going off
A-repeating appointment, reschedule only after user
acknowledges

If the user inputs an R or an A, the template changes to

Interval = Mo+Dy+Hr+Mn | DOW

and the user then fills in the repeating interval. DOW means day
of week for weekly repeating appointments. For example, suppose
the user typed the following into the appointment template:

Day 12/25/Yr 08:00 AM #1R !Christmas day

and press RTN. Kangaroo responds with

Interval = Mo+Dy+Hr+Mn | DOW

and the user fills it in

Interval = Mo+Dy+Hr+05 | DOW

Then the alarm will reschedule itself every 5 minutes until
deleted. Note that Kangaroo will determine that the day of the
week is Thursday (for the year 1980) and upon pressing RTN the
display will show:

Fri 12/25/81 08:00 AM #11 !Christmas day

The day of the week has several forms:

MON, TUE, WED, ...	
MO+, TU+, WE+, ...	meaning next occurring
MO-, TU-, WE-, ...	meaning just previous
MO1, MO2, MO3, ...	meaning first, second, etc., Monday of the current month

This last option can only go up to 5th day occurrence in any given
month, and some days and/or months may only allow up to the 4th
occurrence. This option is for scheduling appointments on, say,
the second Tuesday of the month. For example, suppose the
interval template were filled in as:

Interval = 1+ 7+Hr+Mn | Th3

Then a repeating appointment would be set for the third Thursday
of every other month. (The extra seven days forces the

appointment processor to go to the next month.)

7.5 Appointment Processing

In order to prevent possibly unexpected results when appointments go off, Kangaroo actually processes an appointment just before going into deep sleep. When an appointment becomes due, if there is a beep tone specified, Kangaroo will beep, and the annunciator will light up. However, if there is a message or if a program is to be run, Kangaroo will postpone showing the message or running the program until no other processing is being done. This is just prior to shutting down for deep sleep, unless, of course, it is already in deep sleep.

The user may turn off this feature so that no appointments are processed (including beeping and lighting the annunciator) by typing

ALARM OFF

This command turns off the appointment processing. To turn it back on the user types:

ALARM ON

and the normal appointment handling is reactivated (all past due appointments will go off at this time).

When in APPT mode, pressing

RUN

will cause any pending appointments to be processed, i.e., type A appointments are rescheduled, note fields are displayed, and BASIC statements are executed.

The user can also schedule programs to be run or any other BASIC command to be executed by replacing the

!Note

with

>command

For example, the following template will set a repeating alarm to run the program in RAM called "GETUP" every morning at 6:30 am

<Kangaroo E.R.S.>
<08/06/81>

Day 06/24/yr 06:30 am #1R >Run "GETUP"
Interval = 00+01+00+00 | DOW

Kangaroo will correctly fill in the missing template values so that the program will automatically run every morning at the same time.

7.6 Error Messages

The error messages possible in Appointment mode are given below:

Error#	Meaning	Probable Cause
63	insufficient memory	not enough RAM available
103	improper day field	misspelling day
104	improper date field	numbers out of range
105	day/date mismatch	day & date incompatible
106	improper time field	numbers out of range
107	improper alarm spec	not 0-9 or N or R
108	duplicate APPT	two exactly the same
109	improper repetition field	not I or W or bad number in interval field

Time Mode	CHAPTER 8
-----------	-----------

8.1 Time Template

When the user presses the TIME key, the current date and time are displayed continuously in the LCD in the format:

Day Mo/Dy/Year Hr:Mn:Sc AM _

If the user types SET in time mode, the following set-time template is displayed:

Set Mo/Dy/Year Hr:Mn:Sc AM

where:

Mo stands for the month
Dy stands for the day
Year is a 4 digit year (needed to set the century)
Hr:Mn:Sc are the hours:minutes:seconds respectively
AM is either AM, PM, or ** (for 24 hr time)

The default values are the values of the current time or, if the time has never been set, 01/01/000 00:00:01 AM, and are inserted wherever the user fails to give a value.

8.2 Time Commands

Besides the SET command the user can change the format of the month/day/year information or even speed up or slow down the internal clock with the following commands:

STATS - brings the status template into the display. The user can change the date format, time format, and the APPT year format. The template is:

<Kangaroo E.R.S.>
<08/06/81>

Date: MDY, ~Time: AM, Appt: YEAR

where:

MDY is month/day/year format for the date
whether in TIME or APPT mode
DNY means day#month#year format
~Time the ~ denotes approximate time, a
*Time denotes exact time (see EXACT below)
AM can be AM, PM, or ** (for 24 hr time)
YEAR if left as is instructs Kangaroo to only
try to match the date and days in the
next calendar year, and ignore the year
field, or if
EXTD means find the first match from the current
date up to the year 9999 (extended search)

EXACT - this command instructs Kangaroo to take the current time as exact. If this is the first time that EXACT has been typed, the internal clock is flagged as having been set exactly. The next time, and all subsequent times that EXACT is entered, Kangaroo figures a correction factor based on the total number of "ticks" of the real time clock and the changes to the time that the user has entered by adjusting the clock since last typing EXACT.

ADJUST - results in the display

ADJUST (N) + Hr+Mn+Sc.t

To set the clock ahead one hour, place a 1 in the Hr field and press RTN. The time adjust can be plus or minus (+/-) hours, minutes, seconds, and tenths of a second. The (N) field allows for internal correction to the clock. The internal routines do automatic error correction on the clock when the user types EXACT. When the user ADJUSTs the clock, a count is kept of the number of 'ticks' of the real-time clock between EXACTs that have been adjusted. Normal adjustments are made modulo 30 minutes internally. For large variations the user can make the adjustment Absolute by replacing the N with an A.

RESET - this command clears the adjustment and error factors and clears the exact flag.

8.3 Timer Accuracy

The above commands referred to the accuracy of the real time

clock. Kangaroo has no internal trim capacitor for the clock crystal; hence HP cannot adjust it for accuracy at the factory. However, the user can make a software correction via the EXACT, ADJUST and SET commands. The STATS template denotes the EXACT setting via the '*' in front of the word TIME. Otherwise, it is a tilde '~' to denote approximate. The clock should be accurate to 3 minutes per month worst case. However, environmental factors and aging can affect the crystal accuracy.

The timer based commands such as those in the next section should only be used to tenths of a second. The keyboard, LCD, PIL, and system software together make resolution greater than tenths of dubious value.

8.4 BASIC Time Functions

Kangaroo allows BASIC to access the real time clock via the ANSI standard functions below:

TIME\$ = a string giving the current time in HH:MM:SS format.

DATE\$ = a string giving the date in YY/MM/DD format.

TIME - a function returning the number of seconds since midnight.

DATE - a function returning the year and day of year in the format YYDDD, e.g., 81013 is Jan 13, 1981.

ON TIMER #<integer>,<seconds> <command> - sets up timer interrupts for running programs. A program may contain up to 1000 ON TIMER statements (limited only by available RAM), which can instruct Kangaroo to execute any 'legal' after THEN' programmable command or commands separated by @. After the command is executed, if it was not a GOTO, control is returned to the next line to be executed as if the timer had not gone off. The ON TIMER command is an interrupt, and as such, when it is finished processing, control returns to the line from whence it was called.

Kangaroo also has three other time related BASIC commands. All of them use seconds as their parameter:

BEEP [<frequency in Hz>[,<seconds>]] - the seconds is the amount of time the given frequency is heard from the beeper. NOTE:THESE ARE APPROXIMATE! If the seconds are omitted, the default duration is approximately 0.1secs. If the frequency is omitted, the default is approximately 5000 Hz.

<Kangaroo E.R.S.>
<08/06/81>

DELAY <seconds> - the LCD output rate. Results and listings are held in the display until the delay is finished. The default value is one second.

WAIT <seconds> - causes a program to suspend execution for the specified number of seconds and then resume.

PIL	CHAPTER 9
-----	-----------

9.1 General PIL Commands

All peripherals for Kangaroo will interface via the PIL loop system. Kangaroo requires that any peripheral must have its own processor to interpret the commands it sends. The commands that are integral to Kangaroo are general in nature for initialization and writing to peripherals. Specific peripherals will have their own manuals explaining how they interpret these general commands. An advanced I/O ROM (AIQ) will be written for Kangaroo, but will be sold separately. All of the commands below are programmable.

ASSIGNIO - This command initiates the auto addressing features of PIL. The loop can contain 30 peripherals. Kangaroo assigns a number to each peripheral and then the display shows:

```
nn Devices on loop
Device #01=":_ "
```

The user types in a two letter mnemonic which stands for the device name and can be used in the **PRINTER IS** and **DISPLAY IS** commands in programs. For example, if the loop has (in order) a TV, a printer, a Special K and a plotter then the following could be a history of what was typed:

```
ASSIGNIO
04 Devices on loop
Device #01=":TV"
Device #02=":PR"
Device #03=":SK"
Device #04=":PL"
```

The device mnemonics can appear in a string after the **ASSIGNIO** statement, e.g.,

```
ASSIGNIO ":TV,:PR,:SK,:PL"
```

CLEAR LOOP - sends device clear to all peripherals on the loop.

CLEAR ":dev" - sends a device clear to the named device.

DISPLAY IS ":dev1[,;dev2[,...]]" - makes the named device(s) the target device(s) for all DISP statements and display functions. All output that would normally go to the LCD is also routed to the named device(s).

DISPLAY IS * - redirects all display output back to the LCD. (Unlistens the DISPLAY IS devices.)

ENDLINE <string expression> - outputs <string> at end of each print statement rather than the normal CR/LF. The expression can be from 0 to 3 characters long.

LISTIO - lists the device numbers and their assigned mnemonics.

OFFIO - This shuts off the loop, i.e., all PRINT and DISP commands default to the LCD. Kangaroo remembers the loop configuration and, as long as the loop is not changed, can be restored with the RESTOREIO command below.

PRINTER IS ":dev1[,;dev2[,...]]" - sets up named device(s) as the target of all PRINT and PLIST statements.

PRINTER IS * - redirects all printed output to the current DISPLAY IS device(s).

RESTOREIO - This command restores the loop to its previous configuration, without having to reassign device names. Note: as long as the number of devices to be assigned has not changed, Kangaroo will assume the geometry of the loop has not changed.

The above commands could be used when the user moves Kangaroo from one physical location, say his home where he has several peripherals, to another, say his job where he has no peripherals, and back to the first location. The OFFIO command will save the loop configuration and device names until a new ASSIGNIO command is entered or RESTOREIO is typed.

RIO (<reg number>) - numeric function which returns the integer value or the contents of PIL chip register number requested. The register number must be between 0 and 7 or else an error results.

STANDBY ON - Inhibits the PIL timeout so that Kangaroo remains on to process PIL interrupts. Note, this command can be important when dealing with peripherals which do not pass on information while they are "busy".

STANDBY OFF - disables the STANDBY ON command so that normal timeout takes place.

WIO (<reg number>,<numeric expression>) - writes the numeric expression as an integer value directly to the PIL chip register

specified. Caution, at this time, there is no good documentation on the PIL chip for directly accessing and writing to the chip. Such things as turning on the oscillators, addressing the loop, declaring talker actives, etc. are the responsibility of the user.

9.2 Device Dependent Data

Kangaroo will send device dependent data around the loop for specially controlling specific devices. To do this, the user makes the device the display device and then uses DISP CHR\$(xx), where xx is the decimal equivalent of the device dependent data. For example,

```
DISP CHR$(214)
```

puts Special K into double wide, column out mode.

Other devices may use device dependent data as commands which cause them to perform predetermined operations. These commands can be sent by making the device the print device and sending escape sequences. For example, the TV interface will interpret

␣J

as the command "home the cursor to the upper left hand corner of the screen." The user will have to refer to the specific peripheral manual to find out whether or not that peripheral interprets certain data frames as device dependent data (ie, commands). Most listeners (eg, printers and plotters) recognize escape sequences as commands; whereas most talkers (eg, cassette drivers) need HPIL command frames separate from data frames.

9.3 Error Messages

The error conditons that Kangaroo detects when using the loop are:

Error #	Message	Probable Cause
72	loop timeout	PIL unplugged or peripheral not powered up.
74	more names than devices	Either parameter list too

<Kangaroo E.R.S.>
<08/06/81>

		long in ASSIGNIO or fewer devices on RESTOREIO
75	no loop response	Peripheral not powered
76	invalid transmission	Hardware error
77	not controller	Kangaroo is not active controller
78	loop not assigned	need to ASSIGNIO
79	loop was not saved	tried RESTOREIO without first using OFFIO

Card Reader	CHAPTER 10
-------------	------------

10.1 Specifications

The card reader can store up to 650 bytes/side on the magnetic cards; so, the base machine memory can be stored on five cards. The card contains the following information besides the data:

- Data type: BASIC program, text or appointment.
- Size: number of bytes required for load.
- Number of tracks in file.
- Track number of this side.
- Filename.
- Time/date: date and time of file creation.
- Checksums
- Password: file security, not displayed.

In addition, the card may contain a write protect flag.

10.2 Commands

The user commands are based on the COPY command. Recall that a file is specified by

filespec = "filename[:device[/password]]"

For the card reader the device is either CARD for card or PCARD for private card. To use the card reader to copy a file into RAM, type

COPY CARD TO <filespec>

or

COPY <filespec1> TO <filespec2>

where filespec1 must include the device CARD or PCARD. This copies the card contents into RAM, if there is available memory.

If the optional password is used, it must match that on the card or the copy will be aborted. In the first case the "filename" is the one created in RAM and that on the card is ignored; in the second, "filename2" is used, but "filename1" on the card must match "filename1" the user gives.

COPY <filespec> TO CARD

or

COPY <filespec1> TO <filespec2>

where the device is either CARD or PCARD, copies the RAM file to a card. As above, the second form allows a different name for the file in RAM and on the card.

Note, the PCARD option secures the file so that it cannot be listed or recopied to another card. This option is for securing software (ours or an OEM's). Such a private program can only be loaded, purged, or run. Listing and duplicating are prohibited.

PROTECT - This writes a write protect code before the header. To protect a card takes two passes: the first to write data, the second to write the protect flag.

UNPROTECT - This erases the write protect.

CAT CARD | ":CARD" | ":PCARD" - This allows the user to read the header information off the card without actually reading in the data stored on it. Kangaroo will display the file name, file type, track number, number of tracks, and date of creation of the card read.

Suppose "PRGM1" is a BASIC program 1700 bytes long, and suppose that the user wishes to protect the card. Then the following sequence would occur (all caps are user inputs):

```
COPY "PRGM1" TO CARD
03 Tracks required;
Copy to card: align & [RTN]
Pull card ...
Verify card: align & [RTN]
Pull card ...
Track 1 done, insert track 2
Copy to card: align & [RTN]
Pull card ...
Verify card: align & [RTN]
Pull card ...
Track 2 done, insert track 3
Copy to card: align & [RTN]
Pull card ...
```

<Kangaroo E.R.S.>
<08/06/81>

```
Verify card: align & [RTN]
Pull card ...
PROTECT
Protect card: align & [RTN]
Pull card ...
PROTECT
Protect card: align & [RTN]
Pull card ...
PROTECT
Protect card: align & [RTN]
Pull card ...
```

Now suppose another Kangaroo user wishes to copy this card into his machine. The sequence could be (assuming he loads track #2 first):

```
CAT CARD
Catalogue card: align & [RTN]
Pull card ...
<Track 2 of 1>
PRGM1 B 1700 12/25/81 09:30
COPY CARD TO "MYCOPY"
Copy from card: align & [RTN]
Pull card ...
Track 2 done, insert track 1
Copy from card: align & [RTN]
Pull card ...
WARNING: too fast
Copy from card: align & [RTN]
Pull card ...
Track 1 done, insert track 3
Copy from card: align & [RTN]
Pull card ...
WARNING: too slow
Copy from card: align & [RTN]
Pull card ...
```

10.3 Error Messages

The card reader is "smart", i.e., the order of the cards or repetition is immaterial, and the headers are checked to make sure the cards are from the same file, there is enough memory, or if the cards are blank. If the software detects an error (due to timing or a write protect), the appropriate error message is displayed.

<Kangaroo E.R.S.>
<08/06/91>

Error#	Meaning	Probable Cause
22	read/write error	Bad card, or timeout
26	filename mismatch	Wrong filespec in COPY
27	write protected	Card has write protect flag need to UNPROTECT
28	not part of file	User mixed up his cards
29	verify failed	Bad record, try again
63	not enough memory	
66	bad password	Guess again!

BASIC	CHAPTER 11
-------	------------

The language definition for KANGAROO adheres to the ANSI Standard X3.60-1978, "American National Standard for Minimal BASIC, January, 1978". In some areas, KANGAROO includes enhancements to minimal BASIC, and in such cases an effort has been made to conform to:

- a) committee recommendations of subgroups of X3J2
- b) HP 85A
- c) proposed ANSI Level 1 BASIC, as of Dec., 1980

There are incompatibilities between b) and c) above, and Kangaroo took the stand that the syntax should support Level 1 wherever there was a conflict. This has led to a few spelling differences with the 85, but the capabilities are still there. Appendix A documents those differences.

11.1 Numeric data

Kangaroo allows three types of numeric data: REAL (floating point), SHORT (floating point), and INTEGER (fixed point).

- * Real numbers are stored with 12 digit mantissa, three digit exponent, and one digit mantissa sign. The mantissa is in sign-magnitude form, and the exponent is in sign-complement form (10's complement). Range + or - 9.9999999999E+ or - 499
- * Short numbers are stored with 5 digit mantissa, two digit exponent, and one digit sign. The mantissa is in sign-magnitude form, and the exponent is in sign-complement form (10's complement). Range is + or - 9.9999E+ or - 99
- * Integers are stored with 5 digits. Range is + or - 99999

11.2 Character string data

BASIC will handle character strings composed of any valid ASCII characters. The character strings, or strings for short, can be of any length from zero characters to a maximum limited only by available memory. The string of length zero is called the null string.

Minimal BASIC requires that the default length should be 19, but Level 1 says it should be as long as a line. The intent is probably the input line length. Kangaroo has a 32 character line; hence, the default string length has been increased to 32 characters. This is also the length of a string function.

11.3 Arithmetic Operators

Operator	Example
+ Add	A+B
- Subtract (or if unary, negate)	A-B or -A
* Multiply	A*B
/ Divide	A/B
\ or DIV Integer divide (IP(A/B))	A DIV B or A\B
^ Exponentiate	A^B

11.4 Relational operators

Operator	Example
= Equal	A=B
< Less than	A Greater than	A>B
<= Less than or equal	A<=B
>= Greater than or equal	A>=B
<> or # Not equal	A<>B or A#B

When relational operators are evaluated in an expression they return the value 1 if the relation is found to be true, or the

value 0 if the relation is false. For instance, A=B is evaluated as 1 if A and B are equal in value, and as 0 if they are unequal.

Strings are equal if they are the same length and contain exactly the same characters in the same order.

String inequalities compare each string character by character until an inequality is found, or until one string terminates. If an inequality is found, the character with lower ASCII code will be the lesser. If no inequality is found and lengths are unequal, the longer string is the greater.

11.5 Logical or Boolean Operators

Operator		Example
AND	Logical "AND"	A AND B
OR	Logical inclusive "OR"	A OR B
EXOR	Logical exclusive "OR"	A EXOR B
NOT	Logical complement	NOT A

Like the relational operators, the evaluation of an expression using logical operators results in the value of 1 if the expression is true, the value 0 if the expression is false.

11.6 Menu of commands and BASIC Statements

This section gives the syntax and a short explanation of all commands and BASIC statements. The language conforms to ANSI Minimal BASIC and extensions conform to the PROPOSED ANSI Level 1 standard where appropriate. This has caused some differences with the 85, which are enumerated in Appendix A.

ALARM OFF - turns off the appointment processing for future appointments.

ALARM ON - reenables appointment processing.

ASSIGN#<number> TO <filespec> - used to open a data file. Subsequent access is by PRINT#<number> and READ#<number>.

ASSIGN#<number> TO * - closes the data file.

ASSIGNIO [<assign list>] - initiates the PIL auto assign

sequence.

AUTO [<beg line>[,<increment>]] - allows program lines to be numbered automatically. If no parameters are specified, numbering begins with ten and is incremented by ten. Pressing ATTN turns off AUTO.

BEEP [<freq>[,<secs>]]- outputs an audible sound. If no parameters are specified, the frequency is approximately 5000 Hz, and duration is .1 second. By specifying parameters, the user can change only the frequency, or both frequency and duration. Note: the frequency is only approximate.

BEEP OFF - turns off the beeper so that no beep is output.

BEEP ON - reenables the beeper.

BYE - turns Kangaroo off. Same as SHIFT ATTN.

CALL <filespec> - subroutine call to another program. Embedded END statements act as returns to calling program. Parameters may be passed via DATA files.

CAT - produces a catalog listing of information about the current workfile.

CAT ALL - catalogs all files in memory.

CAT APPT - *DISPLAYS THE CATALOG ENTRY OF THE APPT. FILE.*

CAT CARD - returns the header information on a mag card.

CAT KEYS - *DISPLAYS THE CATALOG ENTRY OF THE KEYS FILE.*

CAT <filename> - catalogs the named file.

CLEAR LOOP - sends device clears around the PIL to all devices.

CLEAR ":dev" - sends specified device a device clear.

CLEAR VARS - clears the VARS file.

CONT [<line>] - continue execution of the program at the specified line, or where it was halted, without altering program conditions or modes.

COPY <filespec1> TO <filespec2> - device to device transfer of files.

DATA - provides constants and quoted or unquoted text from which READ can obtain values for numeric and string variables.

DEFAULT OFF - returns machine to normal error processing.

DEFAULT ON - prevents the following math errors from halting

<Kangaroo E.R.S.>
<08/06/91>

program execution by providing default values for out-of-range results which occur in computations or assignments. The default values allow a program to execute completely, using the default values, rather than stopping due to any of these math errors. The default values are:

ERROR	DEFAULT VALUE
Integer precision overflow	+ or - 99999
Short precision overflow	+ or - 9.9999E99
Real precision overflow	+/- 9.999999999999E499
Underflow (all data types)	0
SEC, TAN(N*PI/2), N:odd integer	9.999999999999E499
Zero to negative power	9.999999999999E499
Negative Y ^ Non-integer X	ABS(Y) ^ X
Zero ^ zero	1
Division by zero	+/- 9.999999999999E499
CSEC(N*PI)	9.999999999999E499

DEF FN [<param list>] - single or multiline function definition. Allows optional parameters.

DEF KEY <key>,<redef>[;] - redefines given key. The absence or presence of a trailing semicolon determines whether or not the redefinition is transmitting.

DELAY <secs> - used to speed up or slow down the output rate to the LCD.

DELETE [<beg line>[,<end line>]] - delete a line or section of a program. If only one line identifier is specified, just that line is deleted. If two line identifiers are specified, the block of lines is deleted. If no parameters, then the current line is deleted.

DIM <var list> - used to declare the number of dimensions and the maximum number of elements in each dimension for REAL precision array variables and initializes all elements to "undefined". The maximum number of elements is limited only by available memory. The number of elements can be calculated as available memory divided by element size, where element size is 3 for integers, 4 for shorts, and 8 for reals.

DISP <disp list> - causes the items specified in the display list to be displayed on the LCD (or DISPLAY IS device). Except for the output device, this is the same as a PRINT statement.

DISPLAY IS ":dev" - specifies the device on which the normal system messages are displayed (in addition to the LCD).

DISPLAY IS * - specifies the LCD as the device on which normal system messages are displayed.

<HP Private>

DISP USING - Except for the output device this is the same as the PRINT USING command. See it below.

EDIT - gives user an unnamed workfile of type last in.

EDIT BASIC - gives user an unnamed BASIC workfile.

EDIT TEXT - gives the user an unnamed text workfile.

EDIT <filespec> [, <filetype>] - returns workfile named (after checking for type match if necessary).

END - the highest numbered line of a program, and terminates program execution. Kangaroo does not require the END statement since the operating system already has one at the end of the file.

END DEF - the last line of a multiple line function definition.

ENDLINE <string> - at end of every printed line the given string is output. Normally this is a carriage return/linefeed. The string is limited to 0 to 3 characters.

FETCH [<line>] - gets the given line number, ready for editing. The default is the current line.

FETCH KEY <key> - gets the given key ready for editing.

FETCH <string>[, <line>] - searches the current file for the given string, starting at the optional line, returning that line if found ready for editing with the cursor at the beginning of the string. Note, the search is for the given string and hence capitals or the lack of them are significant.

FOR <var>=<start value> TO <end value> [STEP <increment>] - the FOR statement is used with the NEXT statement and defines how many times a FOR-NEXT loop is to be executed. The loop counter must be a simple variable. If no increment value is specified, it defaults to one.

GOSUB <line> - transfers program control to the subroutine beginning at the specified line.

GOTO <line> - transfers program control to the specified line.

IF <boolean exp> THEN <legal statement> [ELSE <legal statement>]
- the IF...THEN...ELSE statement provides conditional branching. If the expression is evaluated as true, execution is transferred to the specified line or the statement is executed. If false and ELSE is not specified, execution continues on the next line. If

<Kangaroo E.R.S.>

<08/06/51>

false and ELSE is specified, execution is transferred to the specified ELSE line, or the specified ELSE statement is executed.

The following statements cannot follow THEN, ELSE, ON ERROR, or ON TIMER:

DATA	IMAGE
DIM	INTEGER
DEF FN	NEXT
END	OPTION ANGLE
END DEF	OPTION BASE
FOR	REAL
IF	SHORT

IMAGE - used with the PRINT USING or DISP USING statement to specify format. Kangaroo allows either upper case or lower case specifications. The permissible IMAGE formats are:

Image Symbol	Symbol Replication	Purpose	Comments
X	y	Blanks	Go anywhere
" "	n	Text	Go anywhere
' '	n	Text	Go anywhere
D	y	Digit	Fill = blanks
Z	y	Digit	Fill = zeros
*	y	Digit	Fill = zeros
S	n	Sign	+ or -
M	n	Sign	Blank or -
E	n	Sci Notation	Format = ESDDD
.	n	Radix	Radix is .
C	n	Comma	Number seperator
R	n	Radix	Radix is ,
P	n	Period	Number seperator
A	y	Characters	Strings
()	y	Replicate	For specifiers only
K	n	Compact	General purpose
,	n	Delimiter	
/	y	Delimiter	Output CR/LF

INITIALIZE - INITIALIZES A MASS STORAGE UNIT'S MEDIUM.

INPUT <input list> - allows values to be assigned to variables from the keyboard during program execution.

INPUT "userprompt"; <input list> - allows the user to change the normal input prompt (a question mark) to a user defined prompt. This can be any string.

INPUT "userprompt",<initial string>;<input list> - allows user to specify an initial value string for input. The user prompt is protected, but the editing keys are active over the initial string.

<HP Private>

INTEGER <var list> - used to dimension and reserve storage space for INTEGER precision variables - simple and array.

LET - used to assign a value to a variable or variables. The word "LET" is optional in the LET statement.

LET FN - ASSIGNS VALUE TO BE RETURNED FROM A DEF FN.

LIST [<filename>][<begin line>[,<end line>]] - output a listing to the display of all or part of the (optionally named) program in order from the lowest numbered to the highest numbered line. If one line identifier is specified, only that line is listed. (A screen's worth.) If two line identifiers are specified, that block of lines is listed.

LISTIO - lists the mnemonic assignments of the PIL loop.

LOCK <string> - locks the keyboard so that the user must enter the password to have access. This is an absolute secure.

MARGIN - SETS COLUMN OF "END OF LINE" BEEP.

MERGE <file name>[,<beg line>[,<end line>]] - merges named file into current workfile. Merge acts as a LIST into the current file at the start merge line (or current line if defaulted) between the beg line and end line (or the whole file if defaulted).

NAME <string> - names the current workfile and gives the user a new, unnamed workfile.

NEXT - used with the FOR statement, defines the last statement of a for-next loop and causes the loop counter to be incremented and tested.

OFF ERROR - cancels any ON ERROR condition currently active.

OFFIO - deactivates the PIL loop, but saves the current configuration in system memory for later reactivation. See RESTOREIO.

OFF TIMER# - deactivates the corresponding ON TIMER#. No further interrupts from that timer will occur until re-activated.

ON ERROR <legal command> - used to prevent some recoverable program execution errors from halting execution by causing branching when an error occurs and suppressing the normal error process. ON ERROR overrides DEFAULT. Any statement or statements legal after THEN are permissible. If the command is a GOTO, control is passed to that line. If there are more commands on the line, they will be ignored in this case. Otherwise, the command(s) are executed and control is automatically returned to the line after the error causing line.

ON <expression> GOSUB <linelist> - allows any one of one or more

<Kangaroo E.R.S.>
<08/06/81>

subroutines in the current program to be accessed based on the value of the numeric expression. A value of one corresponds to the first line identifier in the list, two to the second, etc. Return is to the statement following the last line identifier in the list.

ON <expression> GOTO <linelist> - allows program control to be transferred to one or more statements in the current program based on the value of the numeric expression. A value of one corresponds to the first line identifier in the list, two to the second, etc.

ON TIMER# <number>,<secs> <legal command> - allows user-specified interrupt intervals for different timers. When the interrupt occurs, the specified action occurs. Control returns to the line following the one which was executing when the interrupt occurred unless the command is a GOTO.

OPTION ANGLE DEGREES | RADIANS - sets trigonometric mode. Default is radians. Option angle transcends programs.

OPTION BASE - allows the default lower bound of arrays to be specified as one rather than zero. OPTION BASE 0 is the default, but can be included for documentation purposes. The OPTION BASE statement must be before any DIM, REAL, SHORT, and INTEGER statements, and only one OPTION BASE statement is allowed in a program.

PACK - PACKS THE CONTENTS OF A MASS STORAGE DEVICE.

PLIST [<filename>][<beg line>[,<end line>]] - same as LIST except that the listing appears on the current print device. Text files are listed without line numbers.

POP - pops GOSUB returns off the stack. Used in exception handling routines.

PRINT [<print list>] - causes the items specified in the print list to be output to the current print device. The items can be variables, array identifiers, expressions (excluding multiple line user-defined functions), or TAB separated by commas or semi-colons. Two commas in a row cause a field to be skipped. A CR-LF is output if no print list is included. A semicolon at the end of the list suppresses the normal CR-LF.

PRINT USING -

PRINT# - used to record values onto the specified file. In serial access mode, recording starts at the beginning of the file or after the last data item accessed. In random access mode, with the defined record number specified, recording starts at the beginning of the defined record. The data list can include variables, constants, and literals, separated by commas.

PRINTER IS ":dev" - redefines which device is used for PRINT

<HP Private>

statements.

PRINTER IS * - defaults print device to LCD.

PROTECT - card reader command to write a "write protect" onto the card to prevent inadvertent copying to the card.

PURGE [<filename>] - removes named file from memory. If no file name is given, the current workfile is emptied.

PUT - "PRESSED" A KEY (UNVERSE "GET")

PWIDTH <expression> - sets width of printed line (number of characters output) before a CR/LF is output.

RANDOMIZE [<seed>] - re-evaluates the random number seed. With no parameter, the seed is generated using the system timer. With the parameter specified, the parameter is used to compute the seed, and the "random" number sequence will be repeatable. A seed of 0 gives the sequence of all zeros.

READ <list> - specifies variables for which values are to be assigned from a DATA statement. Cannot be executed from the keyboard.

READ# - retrieves values for variables from the specified file. In serial access mode, reading starts at the beginning of the file or after the last data item accessed. In random access mode, with the defined record number specified, reading starts at the beginning of the defined record. READ# can be used to reposition the data pointer by omitting the variable list in random mode. The variables in the variable list must be separated by commas.

REAL <var list> - used to dimension and reserve storage space for non-subscripted and array variables and declares them as full precision.

REM - allows insertion of non-executable remarks into the listing of a program to provide documentation and make the program easier to follow. The ! character may be used in place of the word REM.

RENUMBER - allows the program to be renumbered. If no parameters are specified, numbering begins with ten and is incremented by ten. Complete syntax given in Chapter 5, Section 3.

RENAME [<filename>] TO <filename> - allows any file to be given a new name. Default is current workfile.

RESTORE [<line>] - repositions the data pointer to the beginning of the lowest numbered DATA statement in the current program, or

<Kangaroo E.R.S.>
<08/06/81>

to the given line.

RESTORE#<number> [,<line>] - restores the data pointer in the numbered data file to the optional line or beginning of the file.

RESTOREIO - reactivates the PIL loop using the previous mnemonics defined by the user. If the loop configuration has changed, an error is reported.

RETURN - the last line of a subroutine, and transfers control back to the statement following the GOSUB statement.

RUN [<line>] - used to begin execution of a program at either the specified line or the lowest numbered line in memory.

RUN <filename>[,<line>] - same as RUN, but runs the program "filename" stored in RAM, rather than the current workspace file.

SHORT <var list> - used to dimension and reserve storage space for simple and array variables and declare them as short precision.

STANDBY OFF - returns to normal timeout after 5 minutes of inactivity.

STANDBY ON - keeps Kangaroo from going into deep sleep. Used when slow devices are on the PIL.

STOP - terminates the program execution.

TRACE FLOW - used to document program flow. Any branching causes a trace output to be displayed which designates where the branching was from, and which line it was to.

TRACE OFF - turns off the tracing operation.

TRACE-VARS - reports when variables are changed in a running program.

TRANSFORM - CHANGES ONE TYPE OF FILE TO ANOTHER (BASIC → TEXT, ETC)

UNPROTECT - card reader command to erase the protect flag on a card.

WAIT <secs> - causes program execution to be delayed the appropriate number of seconds before it continues.

WIDTH <numeric exp> - width of displayed line.

~~REMOVE?~~ WIO (<numeric exp>,<numeric exp>) - writes second parameter to chip register given in first parameter.

<HP Private>

11.7 Numeric Functions

Name and Parameter	Meaning
ABS(X)	Absolute value of X
ACOS(X)	The arccosine of X
ANGLE(X,Y)	The angle between x-axis and the point (x,y); $-\pi \leq \text{ang} \leq \pi$
ASIN(X)	The arcsin of X
ATN(X)	The arctangent of X, in 1st or 4th quadrant
CEIL(X)	Smallest integer \geq X
COS(X)	The cosine of X
COT(X)	The cotangent of X
CSC(X)	The cosecant of X
DATE	Returns current Julian date, YYDDD
DEG(X)	Radians to degrees conversion
EPS	Smallest machine number ($1E-499$)
ERRL	Line number of latest error
ERRN	Number of latest error
EXP(X)	E^X
FLOOR(X)	Same as INT(X) (Relates to CEIL)
FP(X)	Fractional part of X
INF	Machine infinity $=+9.999999999999E499$
INT(X)	Largest integer \leq X
IP(X)	Integer part of X
LEN(S\$)	Length of string S\$
LOG(X)	Natural logarithm, $X > 0$
LOG10(X)	Log to the base 10 of X, $X > 0$
MAX(X,Y)	If $X > Y$ then X else Y
MEN	Returns number of bytes of RAM unused
MIN(X,Y)	If $X < Y$ then X else Y
MOD(X,Y)	$x - y * \text{int}(x/y)$
NUM(S\$)	ASCII value of first character of S\$
POS(S1\$,S2\$)	Searches string S1\$ for the first occurrence of string S2\$. Returns starting index if found; otherwise, returns 0.
RAD(X)	Degree to radians conversion
RIO(X)	Returns value of PIL register X
RMD(X,Y)	Remainder of X/Y : $X - Y * \text{IP}(X/Y)$
RND	The next pseudo-random number in a sequence of numbers $0 \leq X < 1$
SEC(X)	Secant of X
SGN(X)	The sign of X, -1 if $X < 0$, 0 if $X = 0$, and +1 if $X > 0$
SIN(X)	Sine(X)
SQR(X)	The positive square root of X

<Kangaroo E.R.S.>
<08/06/81>

TAN(X)
TIME
VAL(S\$)

PI
RES

The tangent of X
Time in seconds since midnight.
Returns the numeric equivalent of
string S\$.

3.14159265357

THE LAST NUMERIC RESULT TO BE DISPLAYED
OR PRINTED.

11.8 String Functions

Name and Parameter	Meaning
CAT\$(X)	Returns a string containing catalog of file number X
CHR\$(X)	ASCII character whose code is X $0 \leq X \leq 255$
DATE\$	Returns a string containing the current date in DDMMYY format
KEY\$	Returns the ASCII string code of the currently depressed key.
STR\$(X)	Returns ASCII string equivalent of number X
TIME\$	Returns the current time as an ASCII string in HHMMSS format (24 hour).
UPRC\$(A\$)	Returns string with all uppercase letters
VER\$	A SIX CHARACTER STRING INDICATING THE OPERATING SYSTEM VERSION.

11.9 Print Functions

Name and Parameter	Meaning
TAB(N)	Skips to the column specified

Compatability with 85A	APPENDIX A
------------------------	------------

The following table summarizes the differences between Kangaroo and the HP85A mainframe commands and functions. The notation used is:

x	same or extended syntax
n	new feature not on 85
s	syntactically different command, same semantics
d	different semantics, same syntax
r	removed or not implemented on Kangaroo

The Remarks column describes the differences.

COMMAND		Remarks
ALARM ON/OFF	n	
ASSIGN IO	n	
ASSIGN#	x	75 uses RAM filename, 85 requires a CREATED tape file
AUTO	x	75 starts dynamically at current line 85 defaults to line 10
BASIC	n	
BEEP ON/OFF	n	
BEEP	s	75 uses "[freq [,sec]]" 85 uses "tone, number cycles"
BYE	n	
CALL	n	
CAT	s	75 has CAT ALL 'filename' CARD 85 refers to tape only
CHAIN		r
CLEAR	d	75 has CLEAR LOOP ":dev" 85 refers to CRT
COM		r
CONT	x	
COPY	d	75 refers to file duplication into another filespecifier 85 refers to screen dump
CREATE		r
CRT IS		r
CTAPE		r

<Kangaroo E.R.S.>
<08/06/91>

DATA	x	
DEFAULT ON/OFF	x	
DEF FN	x	75 permits parameters
DEF KEY	n	
DEG		r 75 chaged to OPTION ANGLE (Level 1)
DELAY	n	
DELETE	x	75 will delete current line if no line number given
DIM	x	
DISP	x	75 does not buffer the output
DISPLAY IS	n	
EDIT	n	
END	x	not required by 75
END DEF	n	75 conforms to Level 1
ENDLINE	n	
ERASETAPE		r
FETCH KEY	n	
FETCH	n	
FLIP		r
FN END		r
FOR-TO-STEP	x	
GOSUB	x	
GOTO	x	
GRAD		r
IF-THEN-ELSE	x	
IMAGE	x	75 allows lower case specifications
INIT		r not needed on 75
INPUT	x	75 permits prompt and initial string
INTEGER	x	
KEY LABEL		r
LET	x	
LET FN	x	
LIST	x	75 permits file designator and defaults to entire file 85 lists only current file, 32 lines at a time. At end of file available memory is displayed.
LISTIO	n	
LOAD		r
LOADBIN		r
LOCK	n	
MEM	n	
MERGE	n	

<Kangaroo E.R.S.>
<08/06/81>

NAME	n	
NORMAL		r ATTN does same things
OFF KEY#		r
OFF ERROR	x	
OFFIO	n	
OFF TIMER#	x	
ON ERROR	x	75 permits anything legal after THEN
ON-GOSUB	x	
ON-GOTO	x	
ON KEY#		r 75 uses DEF KEY
ON TIMER#	x	75 uses any number of timers, seconds and any command legal after THEN 85 limited to 3 timers, milliseconds and GOTO or GOSUB only
OPTION ANGLE	n	75 conforms to Level 1
OPTION BASE	x	
PAUSE		r
PLIST	x	75 strips off line numbers if plisting a text file, can PLIST by file name
POP	n	
PRINT	x	75 defaults to LCD if no PRINTER IS and does not buffer the output
PRINT#	x	75 has variable length records
PRINT ALL		r
PRINTER IS	x	75 refers to external printers
PRINT USING	x	75 permits lower case specification
PROTECT	n	
PWIDTH	n	
PURGE		s 75 deletes named file from RAM 85 deletes a tape file
RAD	x	
RANDOMIZE	x	75 uses real time clock for seed
READ	x	
READ#	x	75 does not have fixed length records
REAL	x	
REN	x	
RENAME-TO	x	
REN	x	75 permits optional line range
RESTORE	x	
RESTORE#	n	
RESTORE IO	n	
RETURN	x	
REWIND		r
RUN	x	75 allows filename after RUN
SCRATCH		r
SECURE		r
SETTIME		r

<HP Private>

<Kangaroo E.R.S.>
 <08/06/81>

SHORT	x	
STANDBY ON/OFF	n	
STOP	x	75 does not reset program counter
STORE		r
STORE BIN		r
TEXT	n	
TRACE FLOW	n	
TRACE OFF	n	
TRACE VARS	s	traces all variables 85 uses TRACE VAR <variable list>
UNPROTECT	n	
UNSECURE		r
WAIT	s	75 in seconds 85 in milliseconds
WIDTH	n	
WIO	n	85 IO rom has same capability

All graphics commands are also removed.

BASIC predefined functions:

ABS(x)	x	
ACOS(x)	n	Level 1 version of arc cosine
ACS(x)		r
ANGLE(x,y)	n	Level 1 version of angle of vector
ASIN(x)	n	Level 1 version of arc sine
ASN(x)		r
ATN(x)	x	
ATN2(y,x)		r
CEIL(x)	x	
CAT\$(x)	n	
CHR\$(x)	x	
COS(x)	x	
COT(x)	x	
CSC(x)	x	
DATE	x	75 uses real time clock 85 uses system timer
DATE\$	n	
DEG(x)	n	Level 1 radians to degrees
DTR(x)		r
EPS	x	
ERRL	x	
ERRN	x	
EXP(x)	x	

<Kangaroo E.R.S.>
 <08/06/91>

FLOOR(x)	x		
FP(x)	x		
INF	x		
INT(x)	x		
IP(x)	x		
KEY\$	n		
LEN(s\$)	x		
LGT(x)		r	
LOG(x)	x		
LOG10(x)	n		Level 1 log base ten
MAX(x,y)	x		
MEM	n		
MIN(x,y)	x		
MOD(x,y)	n		Level 1 implementation of MOD
NUM(s\$)	x		
PI	x		
POS(s1\$,s2\$)	x		
RAD(x)	n		Level 1 degrees to radians
RES	n		
RND(x,y)	x		
RND	x		
RIO(x)	n		
RTD(x)		r	
SEC(x)	x		
SGN(x)	x		
SIN(x)	x		
SQR(x)	x		
STR\$(x)	n		Level 1 string of number
TAB(x)	x		
TAN(x)	x		
TIME	x		
TIME\$	n		
UPC\$(s\$)		r	
UPRC\$(s\$)	n		Level 1 uppercase implementation
VAL(s\$)	x		
VAL\$(x)		r	

<HP Private>

Command Abbreviation	APPENDIX B
----------------------	------------

Kangaroo allows the user to abbreviate input of key words. For example, a common command is AUTO. The user could enter

AUTO 10

or use the abbreviation:

A. 10

This feature is implemented in the 'cheapest' way possible, and as such some abbreviations may not seem optimal. The abbreviations are parsed alphabetically and what is listed are the shortest meaningful abbreviations. For example, A. is the same as AU.

but the table only lists the shortest distinct abbreviation. The abbreviations can occur anywhere in a line that the word can occur. Warning: a period cannot replace the last character in the word.

Abbreviation	Command
a.	AUTO
al.on	ALARM ON
al.off	ALARM OFF
as.#	ASSIGN#
b.	BASIC
be.	BEEP
be.on	BEEP ON
be.off	BEEP OFF
c.	CAT\$
c.a.	CAT ALL
cl.l.	CLEAR LOOP
cl.":dev"	CLEAR ":dev"
co.	COPY
d.	DELAY
da.	DATA
defa.on	DEFAULT ON
defa.off	DEFAULT OFF
dele.	DELETE
di.	DISP
di..	DISPLAY IS

<Kangaroo E.R.S.>
<08/06/91>

e.	EDIT
en.	ENDLINE
end d.	END DEF
f.	FETCH
f.k.	FETCH KEY
g.	GOTO
gos.	GOSUB
i.	INPUT
im.	IMAGE
int.	INTEGER
k.	KEY\$
l.	LOCK
li.	LIST
li..	LIST IO
m.	MERGE
n.	NEXT
na.	NAME
o.e.	ON ERROR
o.t.	ON TINTER
of.e.	OFF ERROR
of.t.	OFF TINTER
op.a.	OPTION ANGLE
op.b.	OPTION BASE
p.	PROTECT
pl.	PLIST
pri.	PRINT
pri.#	PRINT#
pu.	PURGE
pw.	PWIDTH
r.	RUN
ran.	RANDOMIZE
re.	READ
re.#	READ#
ren.	RENAME
renu.	RENUMBER
res.	RESTORE
res.#	RESTORE#
ret.	RETURN
s.on	STANDBY ON

<Kangaroo E.R.S.>
<08/06/S1>

s.off	STANDBY OFF
sh.	SHORT
t.	TEXT
ti.	TIME\$
tr.	TRACE FLOW
tr.o.	TRACE OFF
tr.v.	TRACE VARS
u.	UNPROTECT
w.	WAIT
wi.	WIDTH

<HP Private>

Key Codes	APPENDIX C
-----------	------------

This appendix contains the reference data for the LCD character codes. Note:

Underlined 7,8,9 are not available from the keyboard.
Control 7,8,9 are used for accent grave, checkerboard, and lazy-T.

Character translation:

The LCD hardware has the following configuration:

08=tilde character	88=underlined tilde
0A=lazy-T character	8A=underlined lazy-T
7E=insert cursor	FE=underlined insert cursor
7F=replace cursor	FF=underlined replace cursor

The locations 7E and 7F are fixed because the display generator displays 7E or 7F when it wants an insert or replace cursor. However, ANSI dictates that 7E is the tilde, and we want 7F to be the lazy T. There is a software device driver that translates 7E->08 and 7F->0A. Nothing knows about this but the lowest level of software. If the user sends a 08 or 0A, these are BS and LF and get processed specially anyway.

A parallel mapping goes on with FE->88 and FF->8A, for these pairs are the previous characters with their high bits set.

<Kangaroo E.R.S.>
<08/06/81>

The Non-ASCII Keys:

	unmodified	shift	control	control-shift
key 0: ATTN	80	A0	C0	E0
key 1: TIME	81	A1	C1	E1
key 2: APPT	82	A2	C2	E2
key 3: EDIT	83	A3	C3	E3
key 4: up	84	A4	C4	E4
key 5: down	85	A5	C5	E5
key 6: <-	86	A6	C6	E6
key 7: ->	87	A7	C7	E7
key 8: I/R	88	A8	C8	E8
key 9: FET	89	A9	C9	E9
key A: DEL	8A	AA	CA	EA
key B: CLR	8B	AB	CB	EB
key C: LOCK	8C	AC	CC	EC
key D: RUN	8D	AD	CD	ED
key E: TAB	8E	AE	CE	EE

The format of the non-ASCII keys is

1CSnnnnn where C=1 if control is pressed, 0 if not,
 S=1 if shift is pressed, 0 if not,
 nnnnn=the key number (0-1F)

Currently, there are only 15 non-ASCII keys so bit 4 of the
keys is always zero.

<Kangaroo E.R.S.>
 <08/06/81>

Keystrokes and generated characters

dec	hex	char	keys	dec	hex	ch	keys	dec	hex	ch	keys	dec	hex	ch	keys
0	00	DELTA	cl-sp	32	20	sp	bar	64	40	sh-+		96	60	`	cl-7
1	01	degree	cl-a	33	21	!	sh-1	65	41	A	A	97	61	a	a
2	02	x-bar	cl-b	34	22	"	sh-2	66	42	B	B	98	62	b	b
3	03	<-	cl-c	35	23		sh-3	67	43	C	C	99	63	c	c
4	04	alpha	cl-d	36	24	\$	sh-4	68	44	D	D	100	64	d	d
5	05	beta	cl-e	37	25	%	sh-5	69	45	E	E	101	65	e	e
6	06	GAMMA	cl-f	38	26	&	sh-6	70	46	F	F	102	66	f	f
7	07	bell	cl-g	39	27	'	sh-7	71	47	G	G	103	67	g	g
8	08	<BS>	cl-h	40	28	(sh-8	72	48	H	H	104	68	h	h
9	09	sigma	cl-i	41	29)	sh-9	73	49	I	I	105	69	i	i
10	0A	<LF>	cl-j	42	2A	*	*	74	4A	J	J	106	6A	j	j
11	0B	lambda	cl-k	43	2B	+	+	75	4B	K	K	107	6B	k	k
12	0C	mu	cl-l	44	2C	,	,	76	4C	L	L	108	6C	l	l
13	0D	<CR>	cl-m	45	2D	-	-	77	4D	M	M	109	6D	m	m
14	0E	tau	cl-n	46	2E	.	.	78	4E	N	N	110	6E	n	n
15	0F	PHI	cl-o	47	2F	/	/	79	4F	O	O	111	6F	o	o
16	10	THETA	cl-p	48	30	0	0	80	50	P	P	112	70	p	p
17	11	OMEGA	cl-q	49	31	1	1	81	51	Q	Q	113	71	q	q
18	12	delta	cl-r	50	32	2	2	82	52	R	R	114	72	r	r
19	13	epsilon	cl-s	51	33	3	3	83	53	S	S	115	73	s	s
20	14	pi	cl-t	52	34	4	4	84	54	T	T	116	74	t	t
21	15	umlat-A	cl-u	53	35	5	5	85	55	U	U	117	75	u	u
22	16	umlat-a	cl-v	54	36	6	6	86	56	V	V	118	76	v	v
23	17	umlat-O	cl-w	55	37	7	7	87	57	W	W	119	77	w	w
24	18	umlat-o	cl-x	56	38	8	8	88	58	X	X	120	78	x	x
25	19	umlat-U	cl-y	57	39	9	9	89	59	Y	Y	121	79	y	y
26	1A	umlat-u	cl-z	58	3A	:	sh-;	90	5A	Z	Z	122	7A	z	z
27	1B	<ESC>	cl-BACK	59	3B	;	;	91	5B	[sh-0	123	7B	{	cl-,
28	1C	SIGMA	cl-+	60	3C	<	sh-,	92	5C	¥	cl-/	124	7C		sh-=
29	1D	unequal	cl-=	61	3D	=	=	93	5D]	sh--	125	7D	}	cl-.
30	1E	pound	cl-;	62	3E	>	sh-,	94	5E	^	sh-^	126	7E	~	cl-*
31	1F	checker	cl-8	63	3F	?	sh-/	95	5F		cl--	127	7F		cl-9

Error Messages	APPENDIX D
----------------	------------

This appendix contains the error numbers and their meanings generated by the operating system. Unused numbers are not reported at this time.

Mathematical errors:

- 1 underflow |res| < eps
- 2 overflow
 - |real res| > inf
 - |integer res| > 99999
 - |short res| > 9.9999e99
- 3 COT,CSC (n*pi), n integer
- 4 TAN,SEC ((2k-1)*pi), k integer
- 5 0^neg
- 6 0^0
- 7 null data, string or numeric
- 8 division by zero
- 9 neg number^non-integer
- 10 SQR(neg number)
- 11 arg out of range
 - ATN2(0,0)
 - ACS, ASN(x), |x| > 1
 - ON expr GOSUB,GOTO
- 12 LOG(0)
- 13 LOG(-)

System errors:

- 15 system error
- 16 CONTINUE before RUN
- 17 more than 255 nested FORs
- 18 more than 255 nested GOSUBs
- 19 out of memory
- 20 RAM is invalid (on wake up only)
- 21 ROM missing

Card reader errors:

- 23 read/write error on card
- 26 filename mismatch on card operation
- 27 write protected card
- 28 card not part of file
- 29 verify failed

Data typing errors:

- 30 OPTION BASE error

<Kangaroo E.R.S.>

<08/06/81>

- OPTION BASE after array declaration
- OPTION BASE not 0 or 1
- duplicate OPTION BASE in program
- 33 DATA type and READ or READ# type mismatch
- 34 no DATA to READ
- 35 DIM existing variable
- 36 illegal DIM
- 37 duplicate FN
- 38 FN inside FN
- 39 nonexistent FN reference
 - END DEF w/o DEF FN
 - jump inside multiline FN
- 40 FN parameter type mismatch
- 41 missing FN=
- 42 recursive FN definition
- 43 numeric input, expecting literal
- 44 too few inputs
- 45 too many inputs

Miscellaneous program errors:

- 46 missing NEXT
- 47 missing FOR
- 49 null data
- 50 tried to PRINT# or READ# without ASSIGN#
- 51 RETURN w/o GOSUB
- 52 IMAGE error (runtime only)
- 53 PRINT USING error
 - overflow IMAGE
 - numeric into string IMAGE
 - string into numeric IMAGE
- 54 illegal TAB argument
- 55 subscript out of range
- 56 string too long
- 57 missing line number
- 59 Record overflow

File errors:

- 60 invalid filename
- 61 file already exists
- 62 file does not exist
- 63 not enough memory
- 65 can't copy this
- 66 bad password
- 67 bad line length
- 68 invalid file type
- 69 workfile not named

PIL errors:

- 72 loop timeout during ASSIGNIO
- 73 bad loop command
- 74 loop has changed

<HP Private>

<Kangaroo E.R.S.>

<08/06/81>

75 no loop response during send
76 bad transmission
77 wrong status
78 loop not assigned
79 loop was not saved

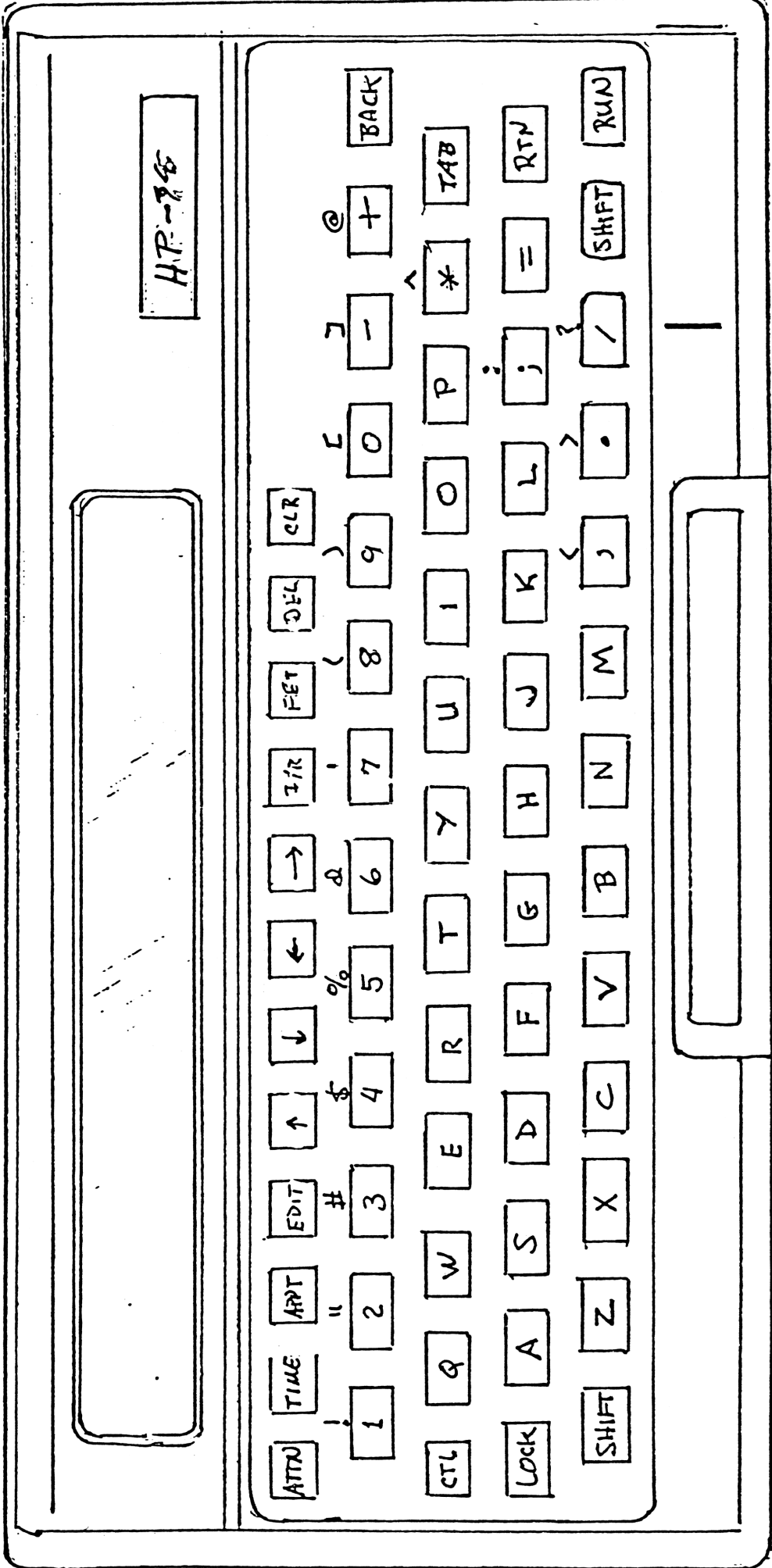
Syntax errors:

80) missing
81 Bad expression
82 String error
83 Missing comma
84 Excess characters
85 Expression too big
86 Illegal after THEN or ON
TIMER, or ON ERROR
87 Bad DIM statement
88 Bad statement
89 Invalid parameter
90 Line number>9999
91 Missing parameter
92 Syntax error
93 ; expected
94 Missing TO

Time and appointment errors:

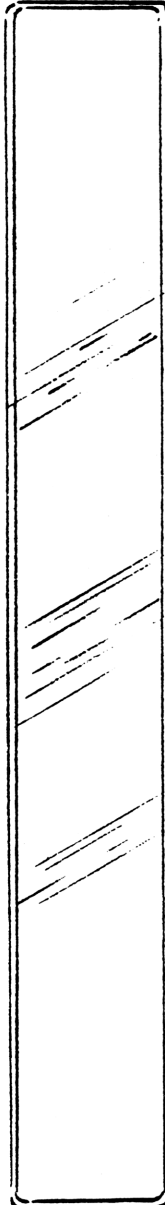
101 adjust factor error
102 invalid APPT delete
103 invalid day field
104 invalid date field
105 day/date mismatch
106 invalid time field
107 invalid alarm specifier
108 duplicate APPT
109 invalid repetition field

<HP Private>



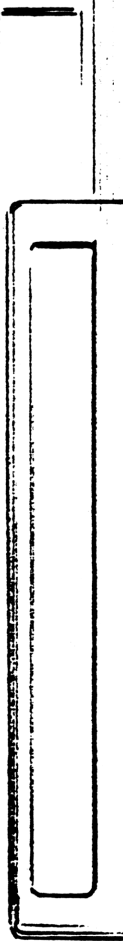
KANGAROO KEYBOARD

12-15-80



14112-75CX

ATTN	TIME	APPT	EDIT	↑	↓	←	→	1/R	FET	DEL	CLR	
!	1	"	#	\$	%	&	'	()	^	@	+
	2	3	4	5	6	7	8	9	0	-		BACK
CTL	Q	W	E	R	T	Y	U	I	O	P	*	TAB
LOCK	A	S	D	F	G	H	J	K	L	;	=	RTN
SHIFT	Z	X	C	Y	B	N	M	,	.	/	SHIFT	RUN



Files	CHAPTER 4
-------	-----------

4.1 Kangaroo memory

The essence of the Kangaroo computer is its file handling system. A file is nothing more than a contiguous area of memory which may be manipulated by the user. The operating system is designed to make the actual file operation transparent to the user. Below is a map of the internal memory configuration.

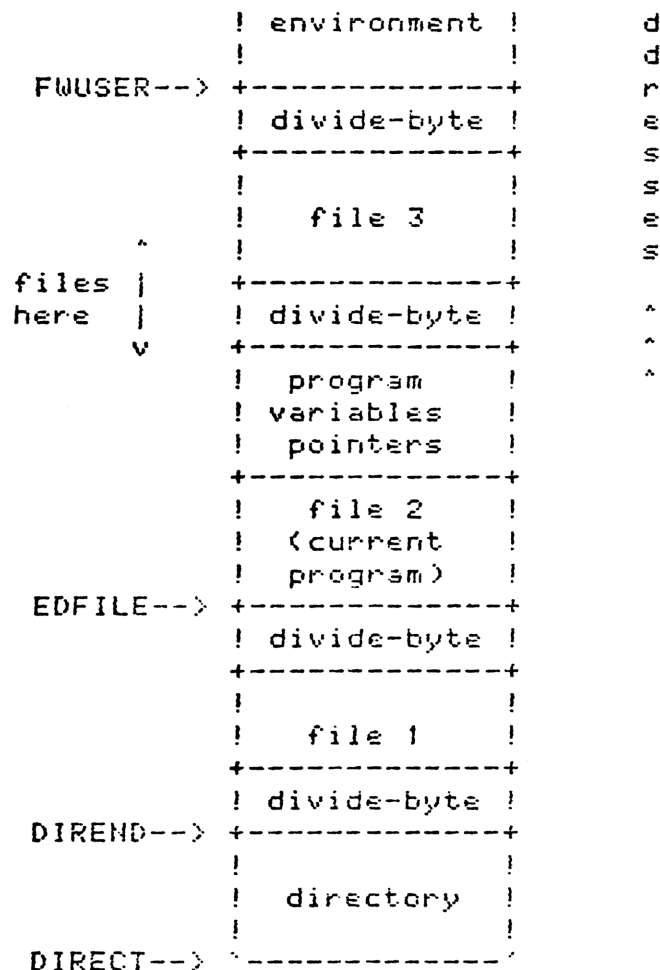
MEMORY LAYOUT:

```

LWAMEN--> ,-----,
           !         !
           !  gosub/for  !
           !   stack   !
           +-----+
           !  reserved  !
           !   memory   !
LAVAIL--> +-----+
           !//////////!
           !/         /!
           !/  unused  /!
           !/   space  /!
           !/         /!
           !//////////!
R12--> +-----+
TOS ==> !         !
           !  R12 stack  !
NXTMEN--> +-----+
           ! divide-byte !
           +-----+
           !         !
           ! environment !
           !         !
FWVARS--> +-----+
           ! divide-byte !
FWLVAR==> +-----+
           !         !
           ! calculator  !

```

^
^
^
i
n
c
r
e
a
s
i
n
g
a



Notes:

1. There is a byte called a divider byte between after each file and after the directory so that if an address for an insertion is given there is no trouble in determining into which file you mean to insert.
2. The important system pointers are as follows:
 - LWAMEM - The highest location in RAM that the system has.
 - CALVRB - FOR/NEXT falls between here and the return stack.
 - RTHSTK - top of the GOSUB/RETURN stack which grows down.
 - NXTRTN - the next available slot for a GOSUB/RETURN entry.
 - LAVAIL - The bottom of the RESMEM area which grows down from the top
 - R12 - points to the top of the R12 stack

TOS - remembers where the bottom of the current R12 stack is. (TOS stood for Top Of Stack in Capricorn.)
STSIZE - points to the size byte of the statement being currently processed
NXTNEM - points to the absolute bottom of the R12 stack and the absolute top of the environment stack.
FWVARS - points to the currently active environment block (see below).
It also acts as the base pointer for referencing remote variables.
FWLVAR - acts as the base pointer for for local variable referencing.
FWUSER - points to the bottom of environment stack which is the calculator environment.
It also points to the top of the file area.
PCR - points to the current executing line in a program.
R10 - points to the next token to be executed

EDFILE - points to the current program

DIREND - points to the end of the directory

3. All system files should have names that are illegal input. This convention prevents a 'file already exists' error from occurring for a file name the user can type in but does not appear in a catalog.

4. A file directory entry looks like:

EQU FOR	BYTE	DESCRIPTION
OFFSET	NO.	
DR.LOC	0/1	location of the file: It is the absolute address of the file. It is maintained by ADJUST.
DR.SIZ	2/3	size of file: is the total size of the file including the PCB but not any divider bytes
DR.TYP	4	type of file (access permission) It is the type of file. Even file types are decompiled as if basic tokens. Odd file types are decompiled as literal text. (see notes below)
DR.NAM	5	type of file (name of file for CAT)

Currently the last 3 bits are a code for the name of the file type.

DR.DAT 6/9 date of creation:
It is in strange internal form (see specific routines).

DR.NAM 10/17 name of file right filled with blanks
legitimacy of a file name is determined by FGETNM.

Total length: 18 bytes equated to DRENSZ

5. An environment control block (ECB)
looks like this:

EQU FOR OFFSET	BYTE NO.	DESCRIPTION
E.LEN	0/1	total length of block
E.PREV	2/3	length of previous ECB
E.RMEM	4/5	amount of RESMEM
E.FCNT	6	size of FOR loop info
E.GCNT	7	size of GOSUB info
E.DATL	8/9	pointer to current DATA line
E.DATA	10/11	where in current DATA line
E.ONER	12/13	the ON ERRCLR address
E.RTNR	14	rom number of the return file
E.RTNM	15/22	which file to return to
E.RTN	23/24	offset into file for return

Total length: 25 bytes equated to ECBLEN

6. A program control block looks like this:

EQU FOR OFFSET	BYTE NO.	DESCRIPTION
P.OPTB	0	option base set
P.LEN	1/2	length of pgm and PCB
P.PLEN	3/4	length of var pointer area
P.CLEN	5/6	length of common area?
P.ELEN	7/8	length of environment area

Total length: 9 bytes equated to PCBLLEN

7. Insertions are made at the following places

environments: at NXTNEM
files: at FWUSER
directory entries: at DIREND

8. The type is stored in two bytes. In the first byte

each bit is intended as an access permission flag or to indicate some other property of the file. The second byte is used to store the name that is to be used by the CAT program.

bits in the access byte:

bit 7 - file in ROM
bit 6 - runnable
bit 5 - editable
bit 4 - listable
bit 3 - purgable
bit 2 - copyable
bit 1 - file format is lined file
bit 0 - token file

examples of typical file types:

type name	types		equates
	access	name	
general system	00000000	00000000	TYSYTM
TEXT	00111110	00000001	TYTEXT
BASIC	01111110	00000010	TYBASC
APPT	00001100	00000011	TYAPPT

The above description is intended only to give the reader a passing familiarity with the internal data structure. It is not intended as a tutorial of the Kangaroo data structure. More complete internal documentation will be available on request.

4.2 Editfiles

The diagram above indicates that Kangaroo allows multiple files and file types in RAM. The user can manipulate these files in various ways and the purpose of this section is to illustrate the user/machine interface for handling files. The goals of the software interface are:

- a) to encourage economy of RAM space
- b) to keep number of commands small
- c) to not sacrifice power or mnemonic value.

So, how does the user create a file and use it? After the initial setting of the system clock (when the batteries are first

File Name: TR"PIL
Last Update: <810817.1415>

This file contains HPIL commands for Kangaroo

The HPIL built-in functions for Kangaroo are in two classes: those routines that are high level oriented and those that are directed at the actual PIL registers. The combination of the two forms allows Kangaroo to address the loop, talk to printers and display devices, and receive data from the loop. The high level routines can be done by the low level ones (except for LISTIO), but in "normal" useage the high level ones are preferable. Loop integrity and operation are easier to maintain via the high level routines, and input off the loop can be done via the low level routines.

High level (BASIC) loop commands:

```
ASSIGNIO
CLEAR LOOP
CLEAR ":device"
DISPLAY IS *
DISPLAY IS ":device"
LISTIO
OFFIO
PRINTER IS *
PRINTER IS ":device"
RESTOREIO
STANDBY ON/OFF
```

Low level loop commands:

Register useage:

#0	contains status of instrument
bit	meaning
7	system controller
6	controller active
5	talker active
4	listener active
3	send service request
2	ready for command received (if receiver) set local ready (if sender)
1	clear "interface clear received" flag
0	master clear

#1 contains the 3 bit frame category

#2 contains the 8 bits of data in the frame

(not to be mistaken with a DATA FRAME)

#3-#7 are current loop status and are automatic

(later documentation will describe their actions)

To send commands and ready frames:

UIO 0,(number)	
160	sets Kangaroo to be CA and LA
UIO 1, 128	command frame coming
UIO 2,(number)	actual data for a command
128	send device clear
160+address	device dependent listener
192+address	device dependent talker

20	device clear
24	enable asynchronous requests
144	interface clear
154	auto address unconfigure
155	loop power down
32+address	listener address
63	unlisten all listeners
64+address	talker address
95	untalk active talker
WIO 1, 160	ready frame coming
WIO 2,<number>	
64	end of transmission
65	end of transmission with error
66	not ready
96	send data (read from register 2 later)
97	send status (read from register 2 later)
99	send accessory id (same)
100	take control
128+initial address	auto address starting at initial address

To send data:

WIO 0, 16	sets Kangaroo to LA (disables error checking)
WIO 1, 0	send data
WIO 2,<data byte>	in decimal

For example, to send the string 'Hello, Dolly' to current listeners:

```

5 STANDBY ON
10 A$='Hello, Dolly'
20 WIO 0,80           ! listener active
30 WIO 1,0            ! data follows
40 FOR I= 1 TO LEN(A$)
50 WIO 2,NUM(A$[I,I]) ! data byte send
60 IF MOD(RIO(1),7)<4 THEN 60 ! wait for return of data sent
70 X=RIO(2)           ! clear for next send (optional)
80 NEXT I

```

To read data you need to use the following BASIC loop:

```

10 IF MOD(RIO(1),4) THEN X=RIO(2) ELSE 10

```