A Brief Description of
Blue RAM Super-Extended BASIC (1.0)

Perkins Engineering
1004 Pleasant Avenue
Boyne City, MI 49712

A Brief description of Blue RAM Super-Extended BASIC (1.0)
Version 1.1 - Released Oct 15, 2000

This document has been retyped and converted to PDF format courtesy of the *Bally Alley* newsletter. For other reprints and more information visit: http://www.ballyalley.com Corrections? Suggestions? Email Adam Trionfo at: ballyalley@hotmail.com

BLUE RAM SUPER-EXTENDED BASIC (1.0)

INTRODUCTION. BLUE RAM Super-Extended BASIC (1.0) is the direct result of the efforts of Jay Fenton and Perkins Engineering. It contains virtually all of the features of Bally BASIC, plus many, many more. These instructions are not intended to teach programming or cover the Bally BASIC features. Rather, it is a brief description of the additional features provided by this language.

GENERAL IMPROVEMENTS. There are many improvements which do not directly reflect in the language such as faster program execution. Others take the form of new commands, new syntax, or new variations on old commands. A list of general improvements follows:

* Built-in keyboard driver
* Faster multiply / divide
* Four color screen
* Versatile program editor
* Program "bomb" recovery
* Faster overall execution
* Additional graphic commands
* Four new data types
* More versatile math forms
* Changeable print number base
* Boolean operations
* Windowed graphics and text
* 300 and 2000 BAUD tape interface
* Two character fonts
* Eight mode flags
* User extensibility
* Full sound effects driver
* Parallel printer driver
* "Trace" program debug aid
* Larger program area

GENERAL OPERATIONS. To operate this cartridge, the Blue Ram must be connected and the switches placed in the A range and the RAM or AUTO mode. The Blue Ram keyboard will operate if connected but in either case the Bally keypad will operate. When using the keyboard, the keys have the following effect:

BREAK- Halt operations
ESC- Halt operations
LEFT BLANK- NEXT
LINE FEED- GO+10
TAB- INPUT
RIGHT BLANK- CIRCLE

In addition, the following control keys (hold down CNTL and press a letter key) are implemented:

A	RND	H	(backup)	O	CIRCLE	V	DEFAULT
B	BOX	I	INPUT	P	PRINT	W	SHOW
C	CLEAR	J	GOTO	Q	SNAP	X	RUN
D	DATA	K	IF	R	RETURN	Y	SCROLL
E	(edit)	L	LIST	S	STEP	Z	ZERO
F	FOR	M	GO	T	TO		
G	GOSUB	N	NEXT	U	POINT		

All key words may be entered using the shifted letter, the control key, it may be spelled out (eg. S C R O L L), or it may be abbreviated (eg. S C .).

A program "bomb" recovery procedure has been implemented. If the program should bomb for some reason (garbage on the screen and/or keypad/keypad lockup) you may recover by pressing and holding RESET, pressing and holding the + key on the keypad, releasing the RESET button, then releasing the + key.

A total of 3100 bytes are available for programs, strings, machine language routines, etc., beginning at %(24576)

By holding down the + (list) key on the keypad, a running program will be traced, line-by-line, on the screen.

NEW VARIATIONS ON OLD COMMANDS. Several commands from Bally BASIC now have different parameters associated with them:

```
:PRINT                Dumps program and string memory to tape
                      at 2000 BAUD using the 2000 BAUD cable.

:PRINT %(AAAAA),NNNN  Dumps nnnn words (2x nnnn bytes) beginning
                      at address aaaaa to tape at 2000 BAUD using
                      the 2000 BAUD cable.

:PRINT 300            Arms the Bally BASIC serial port to dump to
                      tape using the Bally tape interface.

:INPUT                Loads program and string memory from tape
                      at 2000 BAUD using the 2000 BAUD cable.

:INPUT %(aaaaa)       Loads a block of data from tape into memory
                      beginning at address aaaaa, at 2000 BAUD
                      using the 2000 BAUD cable.

:INPUT 300            Arms the Bally BASIC serial port to load
                      from tape using the Bally tape interface.

:LIST                 Checkreads a 2000 BKUD tape using the 2000
                      BAUD cable.

:LIST 300             Reads the Bally BASIC serial port from tape
                      to the screen for visual verification.
```

SPECIAL NOTES ON 2000 BAUD- The special cable that comes with the cartridge is a 2000 BAUD interface which allows loading and dumping to tape at more than 6 times faster than with the standard Bally 300 BAUD interface. The box end of the cable should be locked into the Blue Ram ZIF socket in the front-most row of pins with the cable extending to the left, in a similar manner to the keyboard cable. Both may be used at the same time. The light on the cable will come on when dumping and when data is detected on tape during tape reading. The phone plug end of the cable connects to the tape recorder at the EAR jack for loading and the MIC jack for dumping.

```
GOSUB llll,v,nl,n2,...  Similar to a standard GOSUB or GOTO
GOTO llll,v,nl,n2,...   except that variable v will first be
                        loaded with Q, v+1 with n2, etc. This is
                        equivalent to DATA v,nl,n2.... ;GOSUB
                        1111.

BOX x,y,w,h,m           The values for the mode m have increase
LINE x,y,m              to the following values:
```

0 - Nothing
1 - XOR FA
2 - XOR FB
3 - XOR FC
4 - Overlay BC
5 - Overlay FA
6 - Overlay FB
7 - Overlay FC

where: BC= Background color
FA= First foreground color
FB= Second foreground color
FC= Third foreground color

PX(x,y)
This is the operand for determining the color of a pixel at x,y. Its responses can be: O=BC 1=FA 2=FB 3=FC

NEW COMNANDS. The following new commands have been implemented:

POINT x,y,m
Same as BOX x,y,1,1,m.

CIRCLE x,y,r,m
Draws a circle of radius r on the screen at x,y using mode m (same as BOX).

SCROLL x,y,w,h,n
Scrolls a window w,h at x,y n lines up or -n lines down.

SNAP x,y,w,h,loc
Copies a multicolor field w,h at x,y into memory at loc for a subsequent SHOW. The amount of memory needed is (w+4+(RM#0))xh+4 eg. 6,4 = 12

SHOW x,y,sm,loc
Displays a previously SNAPped field at x,y from memory loc using the showmode sm: O=Overlay, 1=OR, 2=XOR, 3=blank

DATA v,nl,n2,
Loads a secession of variables beginning with v with the trailing operands nl,n2, This is equivalent to v=nl;v+l=n2;

ZERO
Sets all one-letter variables to 0.

DEFAULT
Sets all two-letter variables to their preset values as follows:

BC	239	CC	7
FA	165	LC	0
FB	91	CF=LARGE	
FC	233	CR	80
XR	80	CL	-79
XL	-79	CT	51
YT	51	CB	-48
YB	-48	NB	10
NT	3	XY	0
CX	-79	RM	0
CY	51		

RPL llll/oldtext/newtext
Replaces oldtext with newtext in line llll.

RPL llll//nnnn
Renumbers line llll to nnnn and re-sequences it to its proper position as line nnnn.

PLAY %(aaaaa) Plays a sound string in the background mode while the program continues. With the proper sound string (at aaaaa) this can play three-part harmony, explosions, or any other sound effects the Bally can make.

OP ..... This is a user extensibility command. When the language encounters this command for execution, a branch (jump) is taken to a user provided interpreter routine via a jump vector at address 6DCCH.

llll(edit)(edit)... .. The edit key (CNTL E or PAUSE) is used to step through an existing program line llll one character at a time. Characters may be deleted or new ones inserted as you go. When the end of line occurs, the line will have been changed to reflect what remains showing on the screen.

NEW DATA TYPES. Four new data types have been implemented as follows:

BYTE(v,b) Accesses a single byte of a variable v. b is 0 for the lower byte and 1 for the upper.

! When an exclamation point precedes a number that number is taken as hexadecimal.

> Provides the address of the line number immediately following the right-angle bracket symbol.

← Provides the address of the variable immediately following the left arrow.

NEW OPERATORS. Five new operators have been provided as follows:

- The negative sign negates the value immediately following it. (eg. 5x-7=-35)

↑ This is the Boolean operator AND (7↑5=5)

↓ This is the Boolean operator OR (9↓7=15)

← This is the Boolean operator XOR (3←5=6)

→ This causes the preceding value to be shifted right (sign extending) the number of places in the following term or left (circularly) that many places if the following term is negative.
(24→2=6) (8→-3=64)

NEW VARIABLES. Fourteen new two-letter variables have been provided:

CF Character Font. This variable is set to LARGE for the regular 50 character font or SMALL for a new 3x5 set.

CC Character Color. This variable sets the mode of character screen printing. Its values are the same as the mode values of the BOX command.

LC	Last Character.	Contains the ASCII value of the last text character printed.
CL	Character Left.	The text printed on the screen is constrained to a window controlled by these variables. The window is preset to the full screen.
CR	Character Right.	
CT	Character Top.	
CB	Character Bottom.	
XL	Graphic X Left.	Similarly, the graphics symbols are also constrained to a window as controlled by these variables. Again the window is preset to full screen.
XR	Graphic X Right.	
YT	Graphic Y Top.	
YB	Graphic Y Bottom.	
FA	Foreground Color A.	Since there are four simultaneous colors available on the screen at once, these variables round out the set with BC and FC.
FB	Foreground Color B.	

NB Number Base. This variable controls the number base in which numeric values are printed. It is normally set to ten but may be set to 2 for binary, 8 for octal, 16 for hexadecimal, etc.

MODE FLAGS. The upper 8 bits of the note timer (NT) have been implemented as mode flags since only the lower byte is used as the note interval time. Each bit, when set, has its own meaning as follows:

Bit 7 This bit is used by the program to indicate when a full keyboard scan is to be done as opposed to only a scan of the BREAK and ESC keys. You can generally ignore this bit.

Bit 6 When this bit is set the regular arcade background processor will operate off of the screen interrupt. The PLAY command makes use of this bit. Also, there are some counters and timers supported by this processor.

Bit 5 When this bit is set by the programmer, it informs the system that another user defined background processor has been established. A call will be made to this processor from the screen interrupt via the vector at 6DCFH.

Bit 4 This bit disables printing to the screen when it is set. Sound associated with screen printing is also inhibited.

Bit 3 When this bit is set, it arms the printer driver for printing all characters meant for the screen. This printer driver is for the printer interface provided with the Blue Ram MODEM interface.

Bit 2 This bit set, along with some additional software linked via vector at 6DD8H, will call that software with the ASCII character code in A for all characters meant for the screen.

Bit 1 With this bit set the printer will print lower case characters as opposed to "words" such as one character commands. For example a "t" would print instead of "PRINT"

Bit 0 This bit disables the CNTL "words" and makes the keyboard yield the actual ASCII CNTL characters.

SAMPLE PROGRAM STATEMENTS. The following programs are provided as a basis for experimentation. Try modifying them to see the effects.

```
10    CLEAR;BOX 0,0,15,15,6;CIRCLE -5,3,5,7;CIRCLE 5,3,5,7;CIRCLE 0,-5,
      5,7;CIRCLE 0,0,12,5;SNAP 0,0,24,24,@(0);. DRAW THREE COLOR PATTERN
20    FOR N=0TO 50;SHOW RND (148)-74,RND (88)-44,0,@(0);NEXT N;. PUT 'EM
      ALL OVER THE SCREEN
30    FOR N=1TO 50;SCROLL 0,N-25,N,N,25-N;NEXT N;. SCROLL CENTER OF SCREEN
40    FOR X=80TO -79STEP -1;F=(80-X)÷5;FOR D=0TO 20STEP 10;SHOW X+D,Y,0,
      >(100+RM);NEXT D;FOR N=0TO 20;NEXT N;NEXT X;. NOTE THAT THIS LINE
      REQUIRES THAT LINES 100 THROUGH 104 BE ENTERED AND "POKED" WITH THE
      DATA STATEMENTS BELOW
50    GOTO 10
100   ABCDEFGHIJKLMNOPQRSTUV;. ALPHAS ARE SPACE RESERVERS FOR POKES
101   ABCDEFGHIJKLMNOPQRSTUV
102   ABCDEFGHIJKLMNOPQRSTUV
103   ABCDEFGHIJKLMNOPQRSTUV
104   ABCDEFGHIJKLMNOPQRSTUV

DATA  >100,8,9,0,12291,-4096,-4096,-4093,-16384,12291,3276,15408
DATA  >101,8,9,0,12291,-4096,-4096,-4093,-16384,12291,3084,12348
DATA  >102,8,9,12291,-4096,-4096,-4093,-16384,-16381,15363,3075,15
DATA  >103,8,9,12291,-4096,-4096,-4093,-16384,-16381,-4096,-16384,-16381
DATA  >104,8,9,12291,-4096,-4096,-4093,-16369,-16372,12300,12348,-4096
```

NOTE: Once these data have been poked into lines 100 through 104 these lines cannot be listed! They also cannot be edited since they are no longer printable characters. They essentially represent the values stored as a result of a SNAP command. For example, with the proper picture on the screen, SNAP 0,0,8,9,100 would have the same effect as the DATA statement. The advantage of SNAPping pictures into lines of a program is that it will not change as it will when the storage location is the @() string.

```
NT=!1800;LIST ;NT=3     (lists the program to the printer via the
                         parallel printer port in the MODEM interface)
NT=!lA00                (enables the printer for printing lower case
                         characters instead of token words)
```