B L U E R A M O P E R A T I O N S

GENERAL: The BLUE RAM contains the following features:

1. 4096 bytes of read-write random access memory (RAM) which can be write protected either manually or by program control. This memory is in addition to the 4K screen memory in the BALLY ARCADE.
2. 128 bytes of additional scratch pad memory which cannot be write protected.
3. A RANGE switch to allocate the add-on memory to either 6000H for normal use or 2000H for game cassette emulation. In this position, the RAM acts as if it were a cassette plugged into the front of the BALLY ARCADE and running a machine code game program.
4. Two 8-bit bi-directional I/O ports (PORT A and PORT B) where any bit or combination of bits in either port may be configured to be an input or output under control of the program. Bits may be reconfigured at will by output to several pseudo-ports within the BLUE RAM.
5. External power and ground (+5V and 0V) available for powering additional low-power applications.
6. Special connections to AUDIO-MIX, CLOCK, $\overline{\text{IOREQ}}$, and $\overline{\text{WAIT}}$ for expanded sound and connection/sync to slower (future) devices such as floppy disks, etc.
7. All connections to the BLUE RAM are via a 24 pin Zero-insertion-force (ZIF) socket for easy access. This type of socket can accommodate 24 pin plugs, component leads, and bare wire ends directly without the risk of bending.

EXTENDED STRINGS: Of the many ways to use the BLUE RAM in conjunction with BASIC, probably the easiest is in the form of extended strings. As you are aware, BASIC would access up to 900 string entries (with no program) using the @(n) symbol. If you wrote a 1000 byte program then only 400 string entries were available. However, with the BLUE RAM 2112 string entries are always available, regardless of the size of your BASIC program. What's more, these string entries do not "move around" or otherwise change when you modify your BASIC program or load another program. These extended strings are accessed using the %(2n+24574) form where 2n is twice the number of the string entry desired. The first string entry, therefore, is at %(24576) and the second is at %(24578), the third at %(24580), etc. The MODE switch should either be in the RAM position or AUTO-"enabled". Executing a &(192) will enable RAM in the AUTO position. Try this short example:

```
10 FOR N=24576TO 24600STEP 2
20 %(N)=(N-24574)+2;NEXT N
30 FOR N=24576TO 24600STEP 2;PRINT N,%(N);NEXT N
```

B L U E R A M U T I L I T Y

LOADING: Connect the BLUE RAM and place the RANGE switch in the 6K-RAM position and the MODE switch in the AUTO position. Load the tape in the usual way using :INPUT. Program will auto-start showing BLUE RAM UTILITY at the top followed by 6000 on the third line. Stop the tape, it is now ready to use. Disregard the pause after ...TV=82;TV=13.

ENTRIES: 00 through FF enters that hexidecimal byte into memory at the specified address and sequences to the next address. NOTE: Up to seven entries can be shown on a line. After the seventh entry, the program will show the current address on the next line, ready for another entry.

ERASE erases the previous entry from the screen but not from memory. The cursor is backed up such that another entry can be made to replace the existing memory contents if desired. An erase after the first digit continues at the current memory address. Otherwise, the memory address is backed up by one to maintain its relative position on the screen. Erasure of a non-displayed entry is an error.

SPACE or x or NEXT or STEP displays the contents of the current memory address in hexidecimal and advances to the next address. Similarily, after seven contents have been displayed, the program will show the current address on the next line, ready for another display. This entry can be mixed with normal entries (00 through FF) as desired.

+ or GOTO allows the entry of a new address in hexidecimal. Any address may be specified from 0000 through 7FFF. Four digits are required.

- or LIST allows a hexidecimal listing of a number of addresses (their contents) without repetitively pressing the SPACE key. Enter two four digit addresses in hexidecimal and receive a listing of their contents inclusively.

= or PRINT shifts to the dump mode where entries are made in the format: BBBB EEEE O where BBBB is a four digit beginning address, EEEE is a four digit ending address, and O is an option from the set: GO, R, C, or L. GO indicates that another block of memory is to be dumped (to a maximum of four). R indicates that once the program being dumped is loaded, the loader (bootstrap) will run the BASIC portion of the program-if any. C indicates that the loader will auto-call the loaded program (in RAM) vice the BASIC portion. L indicates that nothing should be done upon loading, that more program segments follow. Once an option other than GO (or the fourth GO which is interpreted as option L) is entered, the program prompts for tape recording to begin. After depressing GO, the dumping process begins. The data being dumped is in two parts: the reload "bootstrap" which begins: &(192)=0;A= ...followed by the data to be dumped from memory which usually takes the form of mostly ??????'s. The ? occurs wherever the memory content is not a printable

character. This is nothing to worry about. Incidently, the bootstrap loader does not show what it is loading so patience is required during the load process. A valid load will finish with: 00 STOP TAPE ... . A bad load will end with ERROR. The bootstrap loader only checks for the right number of bytes on a read from the tape (the character count of the load must be the same as was written). If a character is simply mis-read there is no way to know. BASIC is too slow to allow for an absolute verification during the load process.

GOSUB calls a machine code routine whose address is entered as a four digit hexidecimal address. The routine may be one which was entered or one of the three included with the utility. They are as follows:

6C00 - Copy a game cartridge
6C40 - Enable multi-color screen
6C80 - Go to a full machine code program which may or may not return to BASIC.

INTRINSIC ROUTINES: 6C00 is used to copy a game cassette to tape. The procedure is as follows: 1) enter GOSUB 6C00; 2) remove the BASIC cassette; 3) insert the cassette to be copied 4) press GO; 5) remove the cassette and reinsert the BASIC cassette. At this point the game instructions reside at 6000 through 67FF for a 2K ($19.95) cassette and 6000 through 6FFF for a 4K ($24.95) cassette. The program (utility) is ready for further instructions which may consist of a PRINT entry to write the game onto tape or one of the other entries to modify the game before playing it or writing it to tape. To play the game place the MODE switch to ROM and the RANGE switch to 2K-CASS. and press RESET. NOTE: the copy of the game destroyed all intrinsic routines so that a reload of the utility is required to use them further.

6C40 enables the multi-color screen in accordance with a table of data located at 7000. The table is logically divided into up to 12 items of 10 bytes each. The first 8 bytes of each item specify the settings of the eight color registers (4 left and 4 right). The 9th byte specifies the screen outline color register in the upper 2 bits and the LEFT/RIGHT boundary in the lower 6 bits. A value of 0 means that the boundary is full left. 28 means full right. 14 splits the screen in half, etc. The tenth byte specifies at what line this color setting ends. The last active item should specify C8 as its tenth byte. EXAMPLE:

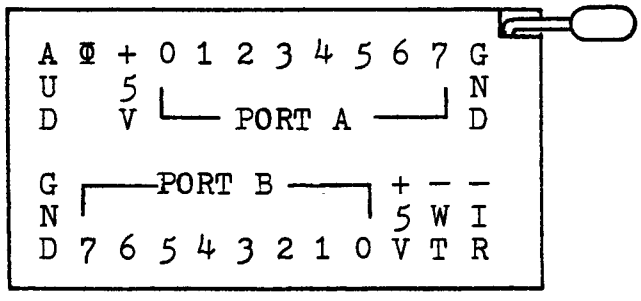
```
7000 00 00 07 07 30 30 37
7007 37 0A 30 58 58 57 57
700E 90 90 97 97 1E 60 A0
7015 A0 C7 C7 C7 C7 A0 A0
701C 14 C8
```

Play with these values (except for the final C8) to see the effect.

6C80 switches control to a machine code program or routine which begins at 6000. Watch the ARCADIAN for special applications of this routine...(wild graphics and sound effects).

You will note that each of the first 25 string entries was set to its own number (n) and the printout shows the relationship between the value placed in the %() statement and the string number.

INPUT/OUTPUT: The BLUE RAM contains two 8-bit bi-directional I/O ports. The first 8 bits are refered to as port A while the second 8 are refered to as port B. Either port may be configured with any bit or combination of bits being in the input mode or the output mode. The following diagram shows the position of each bit of each port on the 24 pin ZIF socket:



PIN	FUNCTION
1	Ground return for port A
2	Bit 7 of port A
3	Bit 6 of port A
4	Bit 5 of port A
5	Bit 4 of port A
6	Bit 3 of port A
7	Bit 2 of port A
8	Bit 1 of port A
9	Bit 0 of port A
10	+5 volt source for port A
11	Z80 clock used for external synchronization
12	AUDIO-MIXER used to add or extract sound
13	Ground return for port B
14	Bit 7 of port B
15	Bit 6 of port B
16	Bit 5 of port B
17	Bit 4 of port B
18	Bit 3 of port B

19 Bit 2 of port B

20 Bit 1 of port B

21 Bit 0 of port B

22 +5 volts source for port B

23 $\overline{\text{WAIT}}$ used for external synchronization

24 $\overline{\text{IORQ}}$ responding to ports 80H through 0BFH

Access to the I/O ports is via the &(n) symbol where n is taken from the following table:

write &(n)=	read =&(n)	FUNCTION
162		Configures port A bits to be input bits or output bits according to the value sent to &(162). Each bit is an input bit unless it is programmed as an output bit by setting its corresponding bit value into &(162). 1=bit 0; 2=bit 1; 4= bit 2; 8=bit 3; 16=bit 4; 32=bit 5; 64= bit 6; 128=bit 7; 40=bits 3 and 5; etc. as outputs. Any position not set will become an input even if it was set to an output by the previous value.
163		Configures port B bits to be input bits or output bits according to the value sent to &(163) as in port 162 above.
160	160	Accesses port A for data transfers to and from the BLUE RAM. Input bits set high during a read result in corresponding values from the read. Output bits set high during a write result in high logic states at those bit positions on the ZIF socket. Those not set result in low logic states at those bit positions.
161	161	Accesses port B for data transfers to and from the BLUE RAM as in port 160 above.
128	128	Accesses bit 0 of port a individually. If bit 0 is an input bit and is being held high during a read, a value of 128 will be returned; otherwise the value is 0. Any value sent will set the output for this bit low if it is an output bit.

write &(n)=	read =&(n)	FUNCTION
129 thru 135	129 thru 135	Same as &(128) except bit 1 through 7 of port A respectively.
136 thru 143	136 thru 143	Same as &(128) except bit 0 through 7 of port B respectively.
144	144	Same as &(128) except that any value written to &(144) will set the output for this bit high.
145 thru 151	145 thru 151	Same as &(144) except bit 1 through 7 of port A respectively.
152 thru 159	152 thru 159	Same as &(144) except bit 0 through 7 of port B respectively.
64	64	Any access to this port sets the mode to ROM when the MODE switch is in the AUTO position.
192	192	Any access to this port sets the mode to RAM when the MODE switch is in the AUTO position.

NOTE: Output pins will only supply 2 ma. of current. Do not expect them to drive heavy loads. Use small relays or other amplifiers where more power is required.

WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING
Do not connect A.C. appliances directly to the BLUE RAM I/O ports. NEVER CONNECT A.C. FROM THE WALL (120V) DIRECTLY TO ANY PART OF YOUR BALLY ARCADE OR THE BLUE RAM ADD-ON OR SEVERE DAMAGE AND SHOCK HAZARD WILL RESULT!!! Relays or optical isolators with at least 1500V isolation should be used for controlling 120V A.C. devices.

Open the ZIF socket (handle in the up position) and insert the remaining 4.7K ohm resistor between pins 10 and 9 of the socket and close it (handle on the down position). Try this simple program:

```
10 &(162)=0;&(163)=0        Set ports A & B to all inputs.
20 A=&(160);B=&(161)        Reads both ports
30 IF A#0PRINT "A",A        Prints port A if #0
40 IF B#0PRINT "B",B        Prints port B if #0
50 GOTO 20                  Recycle to try again
```

As long as the resistor is there, you should receive a

```
A    1
```

printout. Now move the resistor end from pin 9 among pins 8 through 2 and over to 14 through 21 and watch the printout change as the program continually monitors the connection of the resistor to the various input pins. Switches and other sensors may be similarly connected to the I/O pins in order to hook up to some real applications. Watch the ARCADIAN for further sample applications and tutorials.

MACHINE LANGUAGE PROGRAMMING: This is far too complex and wide ranging a subject to adequately treat here. There are many books written on the subject and we expect to run tutorials in serial form in the ARCADIAN concerning how to use machine code routines in conjunction with BASIC programs and the built-in routines of the BALLY ARCADE itself. Similarly, additional data will be distributed relating to the "extra" pins on the ZIF socket for external synchronization to devices such as floppy disks, etc.

SWITCH SETTINGS: There are two switches located on the BLUE RAM for control of the memory range and memory protection:

RANGE: has two positions. The 6K-RAM position is the normal position for this switch. This position is used for all memory uses except cassette emulation. It allocates memory to 6000H (or 24576 decimal). The 2K-CASS. position allocates memory such that it displaces the cassette plugged into the cassette slot in the front of the BALLY ARCADE. The only use of this position is in emulating a game cassette by running a machine code program similar those in the game cassettes. Future issues of the ARCADIAN will explore this capability further. Also, see the COPY A GAME CARTRIDGE feature of the BLUE RAM UTILITY program.

MODE: has three positions. The AUTO position is the normal position for this switch. In this position, the setting can effectively be switched internally between RAM and ROM by the special I/O ports &(64) and &(192). (See the table under INPUT/OUTPUT). The center position of the MODE switch is the RAM mode which forces the first 4096 bytes of memory into the read/write mode (no memory protection).

The ROM position forces the first 4096 bytes of BLUE RAM memory into the read-only mode where its contents cannot be altered as long as power is applied to the BLUE RAM. This position is helpful whenever it is important to preserve the contents of memory, say prior to writting it to tape. Remember, however, that in this position, memory cannot be changed even if you want to change it. Strings cannot be altered, at least not the first 2096 entries. The switch must be moved to the RAM position or the AUTO position with write enabled (&(192)=0).