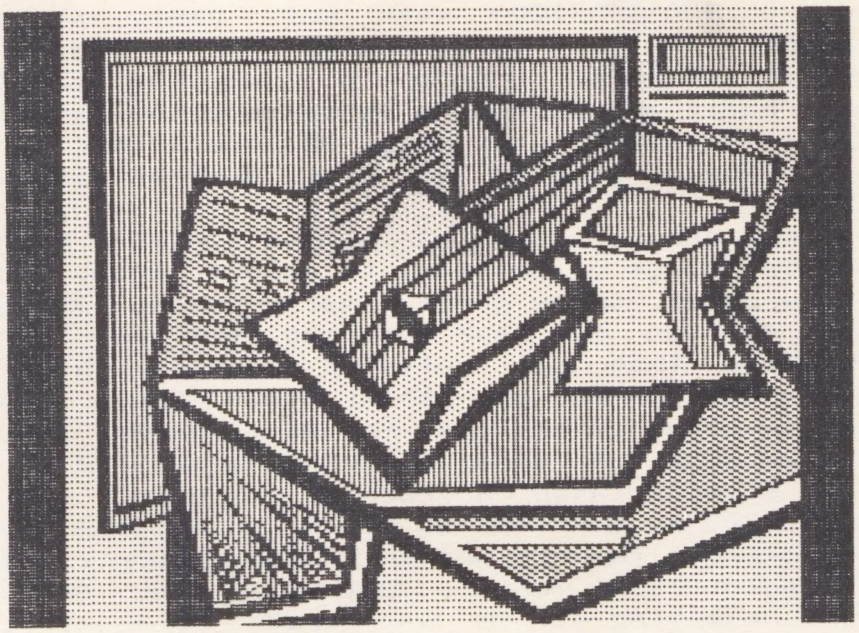
DAL-ACE

DALLAS ATARI COMPUTER ENTHUSIASTS

VOLUME 4 ISSUE 5 MAY 1983



I'm sure by now that you have seen the new keyboard for the 2600 VCS. Looks very nice doesn't it. For $90 you can convert the VCS into a REAL computer. You notice I said REAL. More on that later.

It seems that Atari is going after the home market with a vengeance. For that price you get 16K of ROM and 8K of RAM. The later is expandable to 32K. The main processor is the old standby, the 6502. The keyboard is somewhat like the old TI-99/4 (not the 4A). The basic is close to MICROSOFT basic in the way it handles strings. Atari is planning to release a whole new line of drives, printers, etc for the new computer.

With the new computer comming out we, DAL-ACE, are faced with a new problem:do we support the users of the new computer? The way I see it is one, it is a computer, not a game machine. Two, it does have the name ATARI on it and we are here to support Atari owners. But with the above reasons, we have to look at the negatives. One, can you imagine the number of users we will have! I have joked about having to get Reunion Arena to hold our meetings but it could come true. Second, this means that we will have to create another library to support the 2600.

I would like to here your opinion on this matter. Please contact one of the members at large or myself.

Now, as for that name 'computer'. It seems that Atari uses it for everything they sell. If you notice in the letters VCS, the C stands for computer. Now I ask you is the 2600 by its self a computer. Of course not. By the same token a 400/800/1200 with out basic is not a computer either, its just a big 2600! There should be a law against it. Atari is trying to sell the 1200XL for $800 with no basic and Commodore is selling the 64 for as low as $260 complete. Now let me go on record that I think the 64 is a piece of garbage, But thats another story.

As you can see by shopping around today, the Atari user has a wide range of software to select from. You can buy adventure games, space games, maze games, shoot-em-up games, etc. As you my have noticed the key word was 'game'. It seems that all software writers think Atari users want is GAMES. Do you think that Apple would be where they are today if all they had were games? No, they are there because of application programs, data base programs and other related software. To bring the point home, I was asked by a friend to write a program for his company to instruct technicians on a new product. It was to have an audio-visual format. Just what the Atari does best. After presenting it to his upper management, they said they didn't want a 'game machine'.

It all comes down to this:if all you can get is games, then the Atari will always be called a game computer.

Happy 'GAME' computing,
Gary

ATARI® BASIC Reference Manual Update

This product update contains a number of corrections and additions to the *ATARI BASIC Reference Manual.*

Page 1. This definition is missing from the **TERMINOLOGY** list:

Floating Point Number: A number containing an integer part, a decimal point, and a fractional part. The total number of significant digits in a floating point number, excluding the exponent, may be either nine or ten. This depends on whether the exponent is an even or odd multiple of 10.

Page 6. This information pertains to the **ARITHMETIC OPERATORS** subtraction and exponentiation:

Note: Avoid negating zero, as this will produce an invalid number. For example, if you type

```
PRINT -0
```

the result will be

```
-0E-<8
```

Note: Since the algorithm used to generate exponents (^) is only an approximation, you cannot obtain integer results with it—for example, 2^2=3.99999996. To correct this, use the following technique:

```
X=2^2
PRINT INT (X+.5)
4
```

Page 7. This Note regards the use of the **LOGICAL OPERATORS:**

Note: Avoid using the statement PRINT A=NOT B, as the results are not predictable. Essentially, any PRINT statement with a NOT operator will be unpredictable.

Page 13. This Note is in reference to **SCREEN EDITING:**

Note: Large amounts of editing may lock up the system. It's recommended that programs under development be stored to cassette or diskette periodically (every 30 or 40 edits) with the SAVE or CSAVE command.

Page 20. This Note regards **ON/GOSUB** statements:

Note: If an ON/GOSUB expression evaluates to a number greater than the number of subroutine entries, then a POP statement will be necessary to clear the stack (see POP command, Section 4).

Page 22. Further information on **RESTORE (RES.):**

The RESTORE statement will not generate an error if the line number referenced does not exist. Instead it will RESTORE to the next larger line number in the program. Care should be taken to update RESTORE statements when renumbering a BASIC program.

Page 25. Some additional information on using the **INPUT (I.)** statement:

When executing an INPUT from the screen, avoid moving the cursor away from and then back to the same line; otherwise, the wrong data may be input. Specifically, the INPUT prompt will be included in the INPUT string.

If a string of 128-255 characters is INPUT, then RAM locations 1536-1664 will be overwritten. This area is normally reserved for storage of programs or data. (See the *ATARI Tech Reference Notes.*)

To INPUT strings of more than 127 characters, use the GET command and store the values into a string (see Section 5, OPEN/CLOSE and PUT/GET commands).

Note: The maximum number of characters that can be INPUT from the screen is 120. The maximum for other devices is 255.

Note: Make sure that every INPUT statement has a variable after it; otherwise, unpredictable results may occur.

Page 26. This regards the use of the **LOAD (LO.)** command:

Note: If a program is loaded that is too large for the available memory space, it may give unpredictable results without an error message.

This Note should follow the **LPRINT (LP.)** command description:

Note: An LPRINT command with a semicolon at the end will cause the following LPRINT statement to print on the next 40-column tab. A 40-column printer will move to the next line in such a case. To use the semicolon effectively, use the OPEN statement for the printer, then write to the printer with a PRINT statement (see OPEN/CLOSE and PRINT commands, Section 5).

Page 27. This information pertains to the filespec definition:

Note: Be sure to include the closing quotation marks on a filespec parameter, especially when putting multiple statements on one line. For example,

```
OPEN #1, 4, 0, "D:TEST":STOP
```

will work, but

```
OPEN #1, 4, 0, "D:TEST:STOP
```

will not function correctly.

Page 28. This is an addition to the **POINT (P.)** section:

Note: To update a file, you must open it with a 12 in aexp1.

In the last paragraph under **PRINT (PR. or ?)**, the first sentence should read:

A comma tabs every 10 spaces.

The following sentence should conclude the final paragraph on **PRINT (PR. or ?)**:

However, if the last character to be printed (as in a string with quotation marks) is a CTRL R or CTRL U, then the next PRINT will begin at the end of the current line.

This note should then conclude this section on **PRINT (PR. or ?)**:

Note: In rare circumstances data printed to a diskette may have part of the BASIC program embedded in it. If this occurs, retry the operation.

This Note regards the **PUT (PU.)/GET (GE.)** section:

Note: In certain circumstances the GET function may modify other variables within the program. To avoid this, PRINT any number to the screen between each GET.

Page 30. Here is a corrected version of the table—note in particular the correction on cmdno 32:

cmdno	OPERATION	EXAMPLE
3	OPEN	Same as BASIC OPEN
12	CLOSE	Same as BASIC CLOSE
13	STATUS REQUEST	Same as BASIC STATUS
17	DRAW LINE	Same as BASIC DRAWTO
18	FILL	See Section 9
32	RENAME	XIO 32,#1,0,0,"D:TEMP,CAROL"
33	DELETE	XIO 33,#1,0,0,"D:TEMP.BAS"
35	LOCK FILE	XIO 35,#1,0,0,"D:TEMP.BAS"
36	UNLOCK FILE	XIO 36,#1,0,0,"D:TEMP.BAS"
37	POINT	X10 37,#1,A,B
38	NOTE	X10 38,#1,X,Y
254	FORMAT	XIO 254,#1,0,0,"D2:"

Page 33. The last sentence in the paragraph about the **CLOG** function should read:

CLOG(0) through CLOG(1) are inaccurate and should not be used.

Page 34. The last sentence in the paragraph about the **LOG** function should read:

LOG(0) through LOG(1) are inaccurate and should not be used.

Page 38. The last line in the first paragraph should read:

was stored there previously.

Page 39. The first sentence should read:

Upon execution, the screen displays THE SQUARE ROOT OF 10000 IS 100.

In the second paragraph, the last line should read:

number 1000000000.

This is additional information on the **VAL** function:

Only the numeric field will be translated, while the text will be ignored. For example:

VAL("5SUM") = 5

This information pertains to **String Concatenation:**

Note: BASIC cannot move strings of 256-character multiples correctly. String lengths should be checked; if any string contains a multiple of 256 characters, add or subtract one character from the amount to be moved.

In *Figure 7-6,* the correct result of the program on the left is:

BCD#

Page 42. Some additional information on using the **DIM (DI.)** statement:

Make sure that the DIM statement does not contain a space between the string or array name and the left parenthesis of the dimensioned amount; otherwise, the following will happen—

DIM L (10) becomes DIM L10)

—and this variable can no longer be referenced.

Note: The command COM is identical to DIM and may be used in its place.

Page 43. This is an additional Note for the **DIM (DI.)** section:

Note: Due to a discrepancy in boundary checking, arrays of up to 32766 by 32766 in size can be dimensioned. The programmer should size the array ahead of time to ensure that there is no "virtual" storage space.

Additional information on using the **CLR** command:

The second sentence in the last paragraph, beginning "It also clears ...," should be deleted.

The CLR command will not initialize the values in strings and arrays.

Page 45. Here is a corrected version of **TABLE 9.1**:

TABLE 9.1—TABLE OF MODES AND SCREEN FORMATS

Gr Mode	Mode Type	SCREEN FORMAT Horiz. (Columns)	Vert. (Rows) Split Screen	Vert. (Rows) Full Screen	Number Of Color Registers	Split Screen	RAM Required (Bytes) Full Screen
0	TEXT	40	—	24	1½	—	992
1	TEXT	20	20	24	5	674	672
2	TEXT	20	10	12	5	424	420
3	GRAPHICS	40	20	24	4	434	432
4	GRAPHICS	80	40	48	2	694	696
5	GRAPHICS	80	40	48	4	1174	1176
6	GRAPHICS	160	80	96	2	2174	2184
7	GRAPHICS	160	80	96	4	4190	4200
8	GRAPHICS	320	160	192	1½	8112	8138

Page 49. The last sentence under **PLOT (PL.)** should read:

"The range of points begins at 0 and extends. . ."

Page 50.

In **TABLE 9.3**, the color PURPLE should be inserted after PINK in the first column, and the number 5 should be inserted after 4 in the second column.

Page 51. The sentence directly under **TABLE 9.4** should read:

"DEFAULT" occurs if no SETCOLOR statement is used.

Page 53. Here is a corrected version of **TABLE 9.5**:

MODE, SETCOLOR, COLOR TABLE

Default Colors	Mode or Condition	SETCOLOR (aexp1) Color Register No.	Color (aexp)	DESCRIPTION AND COMMENTS
		0	Color data	—
LIGHT BLUE	Mode 0 and	1	actually	Character luminance
	all text		determines	(same color as background)
DARK BLUE	windows	2	character to	background
		3	be printed.	—
BLACK		4		Border
ORANGE		0	Color data	Character
LIGHT GREEN	Modes 1	1	actually determines	Character
DARK BLUE	and	2	character to be	Character
RED	2	3	printed.	Character
BLACK	(text modes)	4		Background, border
ORANGE		0	1	Graphics point
LIGHT GREEN	Modes 3, 5,	1	2	Graphics point
DARK BLUE	and 7	2	3	Graphics point
	(four-color	—	—	—
BLACK	modes)	4	0	Graphics point (background default), border
ORANGE	Modes 4	0	1	Graphics point
	and 6	—	—	—
	(two-color	—	—	—
	modes)	—	—	—
BLACK		4	0	Graphics point (background default), border
		—	—	—
LIGHT BLUE		1	1	Graphics point luminance (same color as background)
DARK BLUE	Mode 8	2	0	Graphics point (background default)
	(1 color,			
	2 luminances)	—	—	
BLACK		4	—	Border

Page 54. In *Figure 9-4,* line 80 should read:

80 XIO 18, #6, 12, 0, "S:"

Page 55. This information pertains to **TABLE 9.6:**

In Column 1, # 14, a period, not a bar, shows on the screen.

In Column 3, #'s 92-95 should show a superscripted circled 1 next to their characters.

Page 56. Here is a corrected version of **TABLE 9.7:**

TABLE 9.7—CHARACTER/COLOR ASSIGNMENT

		Column 1 Conversion	Column 2 Conversion	Column 3 Conversion	Column 4 Conversion
MODE 0	[2] SETCOLOR 2	#+32	#+32	#-64	NONE
		POKE 756,224		POKE 756,226	
MODE 1	SETCOLOR 0	#+32	#+32	#-32	#-32
OR	SETCOLOR 1	NONE	#+64	#-64	NONE
MODE 2	SETCOLOR 2	#+160	#+160	#+96	#-96
	SETCOLOR 3	#+128	#+192	#+64	#+128

[2] Luminance controlled by SETCOLOR 1, 0, LUM.

Page 58. The last paragraph should read as follows:

Note that the DATA statement in line 90 ends with 256, which is outside of the designated range. The 256 is.....

In **TABLE 10.1:**

The PITCH VALUE of 193 should have a musical note of "E," not "D."

Page 63. The last line in item 9 should read:

precedence will save a few bytes.

Page 67. In *Figure 11-2,* line 0260 under Data should be:

#2

Page E-1.

The right parentheses are missing after the word "CONSTANT" in Atari Functions of Inverse Cosine, Inverse Secant, and Inverse Cosecant.

Page H-7. Line 160 in the program should read:

160 IF K=125 OR K=155 THEN 180

Page H-8. Line 50 in the program should read:

50 PLOT 0,0:DRAWTO 159, DR

Page 117.

Following COM, "(see DIM)" should be deleted and replaced with "A-1."

Page 118.

Under "Input/Output Devices," Line Printer should be followed by "(P:)," not "(L:)."

Page 119.

"NOTE, 26" is missing from the listing.

```
1 REM SNAKE BY HARRY HAFELE
2 REM 4-15-83
3 REM A LITTLE SNAKE RUNS AROUND EATING BUGS
4 REM WHEN IT EATS A BUG IT GROWS
5 REM WHEN IT BITES ITSELF IT DIES
6 REM SEES HOW BIG YOU CAN MAKE IT BEFORE YOU DIE
80 REM SMAX SET THE MAXIMUM SIZE OF THE SNAKE
81 REM IT ALSO USES ALOT OF MEMORY
82 REM ADJUST IT AS YOU NEED IT
90 SMAX=1000
100 DIM X(SMAX),Y(SMAX):REM SAVES LOCATION OF SNAKE
101 DIM XDIR(8),YDIR(8):REM STORES INCREMENTS BASED ON DIRECTION
102 DIM STKDIR(15):REM USED TO CHANGE STICK VALUE TO DIRECTION
110 FOR I=0 TO 100:X(I)=0:Y(I)=0:NEXT I
120 FOR I=0 TO 8
122 READ A:XDIR(I)=A
124 READ A:YDIR(I)=A
126 NEXT I
128 DATA 0,0,1,0,1,-1,0,-1,-1,-1,-1,0,-1,1,0,1,1,1
130 FOR I=0 TO 15
132 READ A:STKDIR(I)=A
134 NEXT I
136 DATA 0,0,0,0,0,8,2,1,0,6,4,5,0,7,3,0
140 GRAPHICS 5
150 SETCOLOR 0,13,10:REM SNAKE
160 SETCOLOR 1,8,8:REM BUG
165 SETCOLOR 4,0,0:REM BACKGROUND
200 SBEG=2:SEND=1:BUGS=-1:SSIZE=1
210 SDIR=1
230 XMAX=79:YMAX=39
240 X(1)=RND(0)*XMAX:Y(1)=RND(0)*YMAX
260 GOSUB 2000
280 GOSUB 5000
300 REM DRIVER LOOP
304 REM LINE 305 MAKES SNAKE GROW BY NOT ERASING ITS TAIL
305 IF COUNT>0 THEN COUNT=COUNT-1:GOTO 319
310 COLOR 0:PLOT X(SEND),Y(SEND)
315 SEND=SEND+1:IF SEND>SMAX THEN SEND=1
319 TRAP 4110
320 COLOR 1:PLOT X(SBEG),Y(SBEG)
321 TRAP 40000
330 SCUR=SBEG+1:IF SCUR>SMAX THEN SCUR=1
340 X(SCUR)=X(SBEG)+XDIR(SDIR)
360 Y(SCUR)=Y(SBEG)+YDIR(SDIR)
370 GOSUB 2200
380 SBEG=SCUR

395 REM TEST IF MOVE WILL EAT BUG OR SELF
399 TRAP 4100
400 LOCATE X(SBEG),Y(SBEG),A:IF A=2 THEN GOSUB 2000:GOTO 300
401 TRAP 40000
410 IF A THEN 500
```

```
420 STK=STICK(0)
430 IF STKDIR(STK)<>0 THEN SDIR=STKDIR(STK)
440 GOTO 300
495 REM OOPS - YOU JUST BIT YOURSELF
500 GRAPHICS 2
510 GOSUB 2045
520 POSITION 6,3
530 ? #6;"OUCH"
535 FOR A=1 TO 3
540 FOR I=100 TO 20 STEP -1:SOUND 1,I,10,6:NEXT I:SOUND 1,0,0,0
550 NEXT A
600 IF STRIG(0)=1 THEN 600
610 RUN

1995 REM PUT A BUG SOMEWHERE ON THE SCREEN
2000 BUGX=RND(0)*XMAX:BUGY=RND(0)*YMAX
2005 LOCATE BUGX,BUGY,A:REM DONT PUT BUG ON SNAKE
2006 IF A THEN 2000
2008 GOSUB 2100:REM MAKE EAT BUG SOUND
2010 COLOR 2:PLOT BUGX,BUGY:REM PLOT NEW BUG
2020 COUNT=COUNT+SCOUNT:REM MAKE SNAKE GROW
2040 BUGS=BUGS+1:SSIZE=SSIZE+SCOUNT
2045 ? "YOU ATE ";BUGS;" BUGS "
2046 ? "AND HAVE GROWN TO SIZE ";SSIZE
2050 RETURN
2095 REM MAKE LITTLE SNAKEY SOUNDS
2100 FOR I=10 TO 1 STEP -1
2110 SOUND 0,100+I,10,8
2120 NEXT I
2140 SOUND 0,0,0,0
2150 RETURN
2200 HISS=HISS+1:IF HISS<>10 THEN SOUND 1,0,0,0:RETURN
2210 HISS=1
2220 SOUND 1,0,0,1
2230 RETURN
2245 ? "YOU ATE ";BUGS;" BUGS "
2246 ? "AND HAVE GROWN TO SIZE ";SSIZE
2250 RETURN
3000 FOR I=0 TO 100:? X(I):NEXT I
3995 REM OOPS WE JUST WENT OFF THE SCREEN
3996 REM DO A WRAP
4000 IF X(SBEG)>XMAX THEN X(SBEG)=XMIN
4010 IF X(SBEG)<XMIN THEN X(SBEG)=XMAX
4020 IF Y(SBEG)<YMIN THEN Y(SBEG)=YMAX
4030 IF Y(SBEG)>YMAX THEN Y(SBEG)=YMIN
4050 RETURN
4100 GOSUB 4000:GOTO 400
4110 GOSUB 4000:GOTO 320
5000 ? "ENTER YOUR LEVEL (1-100) ";
5010 TRAP 5000:INPUT A
5020 IF A<1 THEN 5000
5025 IF A>100 THEN 5000
5030 SCOUNT=A
5040 RETURN
```

EDUCATIONAL SOFTWARE REVIEW

KINDER COMP
by SPINNAKER SOFTWARE CORP

REQUIRES 48K,BASIC,DISK,JOYSTICK
PRICE $28.00
REVIEWED BY HARVEY COBB

This program consists of 6 educational games in one.
1. Draw - allows child to make colorful drawings by using the joystick.
2. Scribble - child touches a key and the character repeats itself for a full line.
3. Names - child types in a name or word (up to 15 characters) and the computer makes interesting graphic pattern it.
4. Sequence - five numbers in a numerical sequence are displayed to the child. The child must type in the next number in the sequence. A correct response is rewarded with a smiling face. An incorrect response is followed by a sad face. After five correct responses are made, the child is rewarded with a brief treat of colorful animation and sound.
5. Letters - the child is presented with a lower case letter that must be matched by pressing the upper case letter on the keyboard. The reward system is similar to game #4.
6. Match - Three numbered graphic patterns must be compared to a pattern in a red box. One of the three patterns matches the pattern in the box. The child types the number that represents the right answer. The reward system is similar to game #4.
The six different games combine to make an excellent educational package that retains the child's interest. Once the instructions are explained and demonstrated, the child can easily move from one game to another.

****

DISCLAIMER

SURVIVOR
by Synapse Software
Written by Richard Carr

This article is reprinted from The I/O Connector the Newsletter of the San Diego ACE

Synapse Software has issued yet another interesting program in SURVIVOR, a space shoot-em-up that offers a challenge for up to our players.
The scenario is a familiar one: you are the lone survivor of a fleet of starcruisers whose mission is to destroy four heavily guarded space forts. The enemy Xenogryphs continuously hurl trackers and fighters at you while you work on destroying the forts. The forts are heavily armed with gunners that await you after their barricades are penetrated. Seven skill levels guarantee a challenge for even the skilled trooper. (The seventh skill level is near impossible to beat!).
You can fly your Starwedge Cruiser solo, with one or two gunners, and/or a propulsion engineer, through a scrolling galaxy complete with asteroids and other dangers. (Sorry 1200 XL'ers, you can only use two of the players.) As you destroy all of the gunners of one fort, the fort disintegrates and you move on to the next.
SURVIVOR is a very well designed space game that offers many game options to keep the game interesting. Along with the skill level and player numbers options. you can switch back and forth between a manual and automatic firing mode by simply depressing the "A" key. You can regulate your propulsion to accelerate and decelerate gradually or instantaneously by using the "t" key on your keyboard. If you find yourself desperately in trouble, you can activate one of the seven "smart bombs" which instantly destroys all enemy ships on the screen (and you WILL need them!)
The Starwedge Cruiser resembles the triangular spaceship that we all have seen from the famous Asteroids game, but the firing speed is tremendously faster. When using the two-gunner mode, your cruiser is equipped with two 360 degree cannons mounted on the front and back of the ship. Playing this game with four players is very challenging for all.
The screen scrolling is very smooth and adds to the dimension of the game. The cruiser rotation is very clean for the up-down, left-right positions, but is less responsive for the diagonal positions. Perhaps Mr. Carr designed the cruisers this way intentionally to call to the difficulty .
If you are not completely bored with space games, SURVIVOR should be another welcome game to the Atari owner's library. With the many options available on this game, as your skill level grows, you will still be offered the challenge of SURVIVOR.

The Sensitive Atari
by Richard Q. Fox

This article is reprinted from The Atari Information Digest of the Atari Boosters League of Winter Park, Florida.

The Atari 400/800 can measure temperature, light level, and humidity, You can add this capability to your computer for just a few dollars, and without making any modifications. The PADDLE ports of the Atari are actually specialized analog input ports.They can read the value of resistances in the range of 100 to 100,1000 ohms with a resolution of one part in 228. The Atari reads the value 60 times a second and puts them in memory locations which can be accessed by BASIC PEEK statements. This article includes a simple program which will read the inputs, convert them to useful units of measure, and display them on the screen.

The temperature sensor is a thermistor. Its resistance decreases with temperature. I use a Fenwal model GA45P2 which has a resistance of 50,000 ohms at 25 degrees Centigrade. You can use this temperature sensor to measure room temperature, body temperature, and many other temperatures.

The light sensor is a GE X-6 photocell. It's resistance decreases with increasing light. When the photocell is plugged into the Atari, it can be used in the darkroom to control exposures of prints, or it can be used to measure biological parameters. I have used it to measure the breathing rate of a mouse by shining a focused light beam at the mouse's stomach, and pointing the photocell at the edge of the circle of light. When the mouse breathes in, the light reflects into the photocell, When the mouse breathes out, the light reflection moves out of view of the photocell. Similarly, NASA uses photocells to measure the heartbeats of astronauts. They put a small lamp on one side of the ear lobe and the photocell on the back side. Each time the heart beats, the pumped blood darkens the ear lobe enough to change the light level received by the photocell.

The humidity sensor is a DEVRY Industires HYGROPAK model HA. It is actually a particle sensor, which is most sensitive to humidity, The higher the humidity, the lower the resistance of the sensor. None of the above sensors casts more than $5.00.

The program listing which follows is intended as a demonstration of the capabilities of the Atari analog input system. Lines 10 through 62 OPEN device 2 as the keyboard input, put the screen editor in GRAPHICS 2 mode without a text window, and prints an opening menu. Line 75 reads the keyboard. It looks for any one key to be struck. Lines 80 through 90 display the selected key on the screen for a fraction of a second, before line 95 selects the next procedure to execute. If the key was a 1, then the temperature procedure at lines 100 to 170 is executed. If the key was a 2, then the photocell procedure at lines 200 to 260 is executed. If the key was a 3, then the humidity procedure at lines 300 to 360 is executed. If the key was not a 1, 2, or 3, then the closing screen at lines 999 to 1040 is displayed.

The temperature procedure starts by setting up GRAPHICS 2 without text window, and turning off the cursor. Line 120 is the important line, It reads the analog input as an 8 bit value between 0 and 228 by doing a PEEK(624). Location 624 is the place where the computer executive stores the analog input from the left pot of jack 1 (there are a total of 8 pot inputs; 2 pots per jack - the shadow locations of these 8 inputs are locations 624 through 631). Line 130 is also significant. It converts the value read to a temperature in Fahrenheit. The values for this equation were obtained by exposing the sensor to 2 know temperatures, measuring the input values, and using some algebra to solve two simultaneous equations for two unknowns (the slope and intercept of the linear conversion equation). Lines 140 and 150 print the result. Lines 160 and 170 check to see if a key was pressed while the temperature was being read and displayed. If any key was pressed, then the value is read and thrown away, and the program goes back to the main menu. The light sensor and the humidity sensor are handled in a similar fashion. Only the conversion equations are different.
The hardware needed to connect the sensors to the Atari is very simple. A 9 pin female connector is used to plug into the paddle jacks. The connector, which is of the "DB" variety, is available for $.75. Solder wires to pins 7 and 9 of the connector, and to the two leads of the sensor. Plug the sensor into jack 1, and you are in business. This demonstration shows the versatility of the Atari analog inputs. The applications are only limited by your imagination.

For sale or trade. $15/per game (On disk) Sultan's Palace, Castle, Anthill, Caverns of Mars, Shamus, Sands of Egypt, (On cass), Megalegs, Cripts of Terror, Pacific Coast Hiway, Lords of Karma, Star Raiders(Cart) call John Canedy 253-2883.

FORTH MEETING

The next FORTH MEETING will be held 05-12-83 at 7:30 the home of Eric Weeren. 12920 Audelia, Apt 256, Riverwalk Apartments phone 699-7770

DAL-ACE EDUCATION SIG

The next meeting of the DAL-ACE Education SIG will be held the Wed. prior to the regular Saturday meeting at 7:30 at Software etc.,14400 Dallas Parkway (across from Ewing Buick).

```
10 REM ANALOG INPUT DEMO
15 OPEN #2,4,0,"K:"
20 GRAPHICS 2+16
30 PRINT #6;"ANALOG INPUT DEMO"
40 PRINT #6;"1.TEMPERATURE"
50 PRINT #6;"2.LIGHT LEVEL"
60 PRINT #6;"3.HUMIDITY"
62 PRINT #6:PRINT #6;"PLUG SENSOR INTO PINS 7 & 9 OF JACK 1"
75 GET #2,A
80 PRINT #6;A-48
90 FOR TIME=1 TO 80:NEXT TIME
95 ON A-48 GOTO 100,200,300
98 GOTO 999
100 GRAPHICS 2+16
110 POKE 752,1
120 P=PEEK(624)
125 POSITION 0,0
130 T=(38-P)*18/13+74
140 PRINT #6;"THE TEMPERATURE IS "
150 PRINT #6;INT(T);"DEGREES  "
160 IF PEEK(764)<>255 THEN GET #2,A:GOTO 20
170 GOTO 120
200 GRAPHICS 2+16
210 POKE 752,1
220 P=PEEK(624)
225 POSITION 0,0
230 L=100-P*100/228
240 PRINT #6;"THE LIGHT LEVEL IS "
250 PRINT #6;INT(L);"    "
260 IF PEEK(764)<>255 THEN GET #2,A:GOTO 20
270 GOTO 220
300 GRAPHICS 2+16
310 POKE 752,1
320 P=PEEK(624)
325 POSITION 0,0
330 H=100-P*100/228
340 PRINT #6;"THE HUMIDITY IS "
350 PRINT #6;INT(H);"    "
360 IF PEEK(764)<>255 THEN GET #2,A:GOTO 20
370 GOTO 320
999 GRAPHICS 1+16
1000 PRINT #6;"PARTS LIST"
1005 PRINT #6;"TEMPERATURE SENSOR:FENWAL THERMISTOR GA45P2,50000 OHMS"
1010 PRINT #6;"AT 25 DEGREES C."
1020 PRINT #6
1025 PRINT #6;"LIGHT SENSOR:GETNRAL ELECTRIC X-6 PHOTO CELL"
1030 PRINT #6
1035 PRINT #6;"HUMIDITY SENSOR:DEVRY INDUSTRIES HYGROPAK MODEL HA"
1040 GOTO 1040
```

NEXT MEETING -

HOLIDAY INN - 1735 N. 35E FRWY
TAKE VALWOOD EXIT OFF 35E NORTH.
CONFERENCE RM 1
TIME: 2:00 PM DATE: JUNE 4,1983

MEETING AGENDA

1:00 TO 2:00 SALES
2:00 TO 2:30 BUSINESS MEETING
2:30 TO 2:45 GENERAL QUESTIONS
2:45 TO 3:15 TECHNICAL QUESTIONS
3:15 TO 3:45 BREAK,NEWSLETTERS AND SALES
3:45 TO 5:00 DEMOS ETC.

ADVERTISING RATES

Personal Classified Ads: FREE
Commercial Classified Ads: Half Page - $25.00
(Must Be Camera Ready Copy!) Full Page - $35.00
Deadline: 15 th of each Month Mail to Editor, c/o DAL-ACE

****

Purchasing :	Jack Withrow	680-8947
	Ed Kobus	492-2922
Education :	Jeff Rutherford	1-727-2945
BBS :		214-589-1254
Library :	Ernie Runyon	817-595-2243

The clubs Tecnical Reference Manuals are available for $5.00 + $1.50 shipping or pick up at club meeting. A real good deal for a 104 pages that are nicely bound (spiral spine).

*** New prices on club disks !!! Club library disks are now $5.00 each. Take advantage of a package deal and buy 3 - get one free.

FUTURE MEETING AGENDA

MAY - REVIEW OF TEXT EDITORS TO INCLUDE TEXT WIZARD, LETTER PERFECT, WORDMAN II, AND ATARI WORD PROCESSOR.

JUNE - OPEN QUESTION AND ANSWER SESSION.

JULY - MISC. BUSINESS PROGRAMS TO INCLUDE VISACALC AND THE BOOKKEEPER.

AUGUST - DATA BASES FEATURING FILEMANGER 800, DATA PERFECT, AND THE DATA BASE PROGRAM IN THE LIBRARY.

DALLAS ATARI COMPUTER ENTHUSIAST
(DAL-ACE)

is an independent user education group not affiliated with ATARI, Inc.. This is the official newsletter of the DAL-ACE and is used for the education of its members as well as dissemination of information concerning ATARI computer products.

Subscriptions are $16.00 per fiscal year, beginning in January and prorated each quarter. Other user groups may obtain free subscriptions with the exchange of newsletters.

DALLAS ATARI COMPUTER ENTHUSIASTS

OFFICERS

DAL-ACE
10451 CLARY DR.
DALLAS, TX. 75218

U.S. POSTAGE PAID
BULK RATE
PERMIT NO. 57
ALLEN, TEXAS 75002