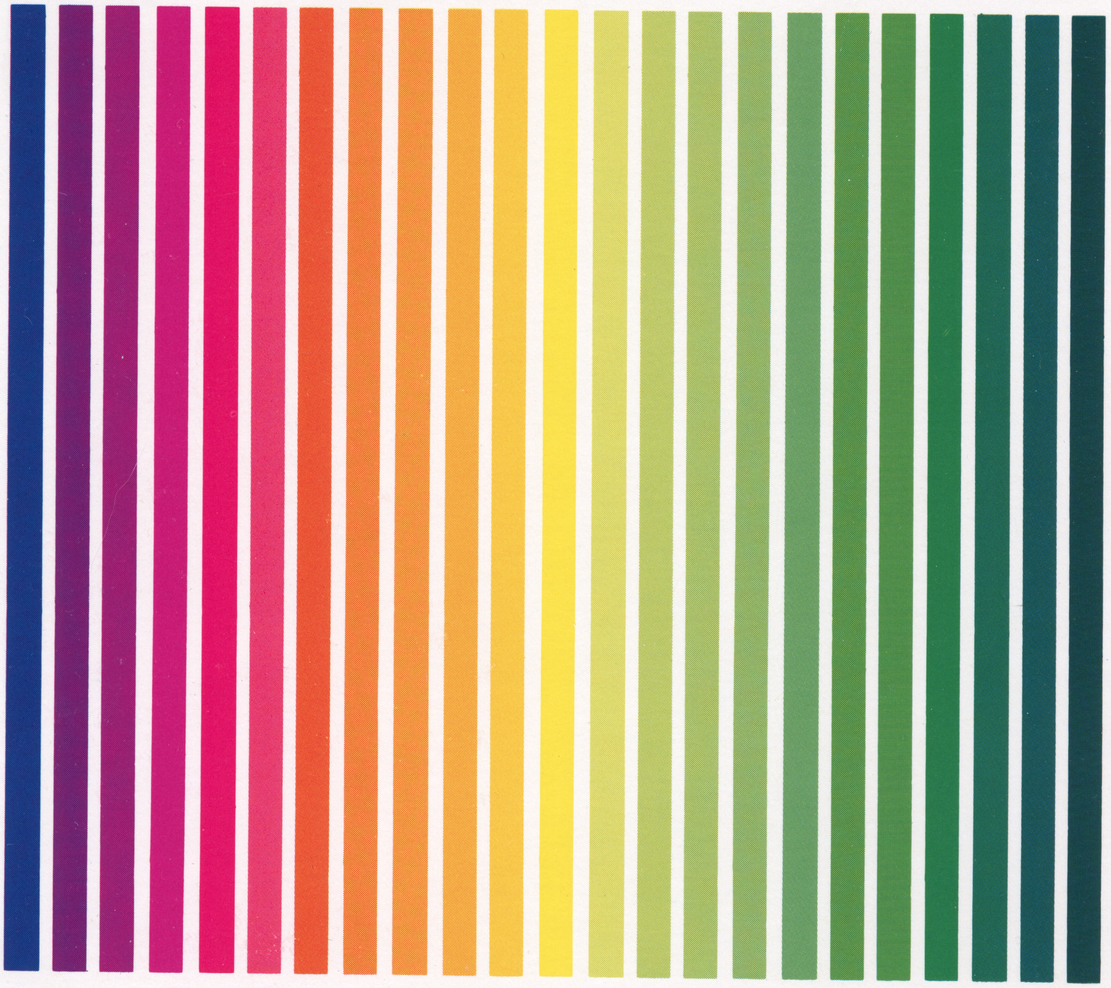
APX ATARI® PROGRAM EXCHANGE



MARCH 1982

SCREEN DUMP UTILITY

DISKETTE (APX-20045)
REQUIRES: 24K RAM

User-Written Software for ATARI Home Computers

MARCH 1982

SCREEN DUMP UTILITY

DISKETTE (APX-20045)
REQUIRES: 24K RAM

SCREEN DUMP UTILITY

TRADEMARKS OF ATARI

The following are trademarks of Atari, Inc.

ATARI®
ATARI 400™ Home Computer
ATARI 800™ Home Computer
ATARI 410™ Program Recorder
ATARI 810™ Disk Drive
ATARI 820™ 40-Column Printer
ATARI 822™ Thermal Printer
ATARI 825™ 80-Column Printer
ATARI 830™ Acoustic Modem
ATARI 850™ Interface Module

****

Distributed by

The ATARI Program Exchange
P. O. Box 427
155 Moffett Park Drive, B-1
Sunnyvale, CA 94086

To request an APX Software Catalog, write to the address above, or call toll-free:

800/538-1862 (outside California)
800/672-1850 (within California)

Or call our Sales number, 408/745-5535.

****

TABLE OF CONTENTS

INTRODUCTION

The Screen Dump Utility is a collection of three programs supplied on disk which allow the user to obtain hardcopy of the video display. The first program in the utility (SD1) allows transferring any text on the screen to any Atari printer. The remaining programs (SD2 and SD3) allow transferring both text and graphic displays to the Atari 822 printer . The user is assumed to be an applications programmer who wishes to use the utility as a debugging tool or to imbed one of the utilities in his program.

This manual is organized into two parts. Part I will allow the user to quickly start using the utility in its simplest form. Part II is a detailed presentation of the capabilities and options of each program in the utility. Please note that this is NOT a Programmer's Manual and will not present either an explanation of the source code or information on how to modify these programs.

PART I. GETTING STARTED

The Screen Dump Utility disk contains three programs:

SD1 Allows dumping text only.
SD2 Dumps text and graphics (excluding players and missiles).
SD3 Dumps text and graphics (including players and missiles).

The essential difference between Programs SD2 and SD3 is size. SD3 requires more memory space than SD2. For the purpose of this part of the manual, we will assume that SD2 and SD3 are the same and discuss only SD3.

MAKING A BACKUP COPY

Before proceeding, it would be prudent to store the master utility disk in a safe place after making a backup copy. Here's how to make a backup copy:

1. Go to DOS.

2. Format a new disk using menu item I.

3. Duplicate the utility disk using menu item J.

4. Save the original disk in a safe place and work only with the backup copy.

TO USE SD1 - TEXT DUMP ONLY

1. Under DOS, copy file SD1 to AUTORUN.SYS using menu item C

2. Be sure that an Atari printer is properly attached to the computer and "on-line."

3. Be sure that a programming ROM cartridge (BASIC, Assembler, PILOT, etc.) is inserted.

4. Reboot the disk: Turn off the computer, wait about 5 seconds, then turn the computer on again.

5. SD1 is now initialized and will remain in RAM until the computer power is turned off.

6. To use SD1, enter or load an application program which puts something onto the video display. For example, under BASIC type:

```
FOR I = 32 TO 90 : PRINT CHR$(I); : NEXT I <ENTER>
```

You should see:

```
READY
FOR I = 32 TO 90 : PRINT CHR$(I); : NEXT I
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKL
MNOPQRSTUVWXYZ
READY
```

7. To obtain immediate hard copy of what is currently displayed, press

```
CTRL-P
```

Note how ALL text on the screen is printed, including "READY," etc.

8. To obtain hardcopy of what is displayed UNDER PROGRAM CONTROL, change RAM address "RUNFLG" (address 7523 decimal) to non-zero. For example, enter and run the following program:

```
  10 PRINT"(ESC)(CTRL)(CLEAR)" :  REM CLEARS SCREEN
  20 FOR I = 32 TO 90 : PRINT CHR$(I);
  30 IF I = 81  THEN POKE 7523, 1  : REM SCREEN DUMP AT "Q"
  40 NEXT I
```

Note that the printout occurs as soon as the letter "Q" is displayed, then execution resumes and the remaining characters are displayed (but not printed).

TO USE SD3 (or SD2) - GRAPHICS DUMP

1. Under DOS, copy file SD3 to AUTORUN.SYS using menu item C

2. Be sure that an Atari printer with graphics capability (822) is properly attached to the computer and "on-line."

> NOTE: An attempt to use SD2 or SD3 to print graphics on any printer other than the Atari 822 will result in meaningless characters being printed.

3. Be sure that a programming ROM cartridge (BASIC, Assembler, PILOT, etc.) is inserted.

4. Reboot the disk: Turn off the computer, wait about 5 seconds, then turn the computer on again.

5. SD3 is now initialized and will remain in RAM until the computer power is turned off.

6. To use SD3, enter or load an applications program which puts something onto the video display. For example, under BASIC, enter and RUN the following program:

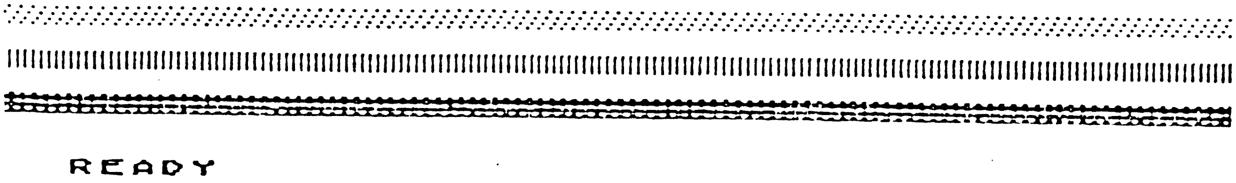
```
10 GRAPHICS 3
20 SETCOLOR 0,8,4  : REM Register 1 -  Hue 8, Luminance 4
30 SETCOLOR 1,8,8  : REM Register 2 -  Hue 8, Luminance 8
40 SETCOLOR 2,8,12 : REM Register 3 -  Hue 8, Luminance 12
50 COLOR 1 : PLOT 0,14 : DRAWTO 39,14
60 COLOR 2 : PLOT 0,16 : DRAWTO 39,16
70 COLOR 3 : PLOT 0,18 : DRAWTO 39,18
```

The screen will display 3 luminance levels of horizontal blue lines.

7. To obtain immediate hardcopy of what is currently on the video display, press

    CTRL-P

When both keys are released the printer will print:



Note how everything on the screen, including the "READY".was printed. Note also how the three luminance levels were printed in different grey scale patterns.

8. To obtain hardcopy of what is displayed under program control, change RAM address "RUNFLG" (address 7523 decimal) to nonzero.

For example, add line 65 to your sample program as shown below:

```
10 GRAPHICS 3
20 SETCOLOR 0,8,4
30 SETCOLOR 1,8,8
40 SETCOLOR 2,8,12
50 COLOR 1 : PLOT 0,14 : DRAWTO 39,14
60 COLOR 2 : PLOT 0,16 : DRAWTO 39,16
65 POKE 7523 , 1       : REM RUNFLG
70 COLOR 3 : PLOT 0,18 : DRAWTO 39,18
```

and then RUN it. Note how the hardcopy was obtained after the first two horizontal lines were displayed and that execution resumed after the printer was finished. Since the third line was not yet on the screen when the hardcopy was requested, it was not printed.

READY

PART II - GETTING DOWN TO DETAILS

If you have read Part I of this manual, you should now be able to obtain hardcopy of screen displays using programs SD1, SD2 or SD3. (SD2 can be run using the same procedure given for SD3.) To use these programs to their full potential, however, you should learn more about them. This part of the manual will explain the options of each program and how to make full use of them.

COMMON CHARACTERISTICS

All three programs are written in 6502 machine code and designed to load in and initialize under AUTORUN.SYS. The amount of RAM required is substantially different, but all three programs begin at memory location $1D50 and build up in memory. The lowest available memory pointer, "MEMLO", is automatically moved upward to allow room for the utility program just below it.

You are provided with a relocatable object module and a relocatable load facility. If you use this facility to relocate the utility, you must protect the RAM space and also initialize the utility by doing a Jump to Subroutine at relative location ORIGIN + 6. By entering at this location, the utility will NOT move MEMLO.

Imbedding the Utility in User Code

If you wish to add your own code, just start at "MEMLO", build up in memory, and reinitialize "MEMLO" above your code.

If you wish to append your program to the utility and then AUTORUN it, you MUST:

1. Change RUN ADDRESS ($2E0) to point to your program entry.

2. Jump Subroutine to the utility AUTORUN initialization entry at location $1D50.

Using this procedure, when AUTORUN is booted, control is transferred to the user program, the utility gets properly initialized, and control returns to the user program.

Executing the Utility

After initialization, the utilities may be executed in any of three ways:

1. Immediate keyboard control. Press CTRL-P when you wish to print what is on the screen.

2. Under program control using "RUNFLG".

The first byte in each utility program is labeled "RUNFLG". Once the program has been initialized, each 1/60th second interrupt generates a jump into the utility to test RUNFLG. If RUNFLG is zero, it is ignored, and a return (RTS) takes place. If RUNFLG is non-zero, the screen dump to printer takes place before the return. In this way, you can initiate a screen dump by changing one RAM location. This is convenient in the higher level languages with a POKE or in machine code with an STA. RUNFLG will be zeroed upon completion of the screen dump.

One word of caution in using RUNFLG. The start of the screen dump to printer will not take place IMMEDIATELY upon execution of the POKE to RUNFLG. It might take as long as 1/60th second after the POKE until the next interrupt occurs and the screen dump takes place. A lot of code can be executed in 1/60th second! If you want to be absolutely certain that the screen does not change before the screen dump takes place, put in a test of RUNFLG before proceeding with your code. For example, in BASIC:

```
 5 RUNFLG = 7523
10 PRINT "Anything"
20 POKE RUNFLG, 1
30 IF PEEK (RUNFLG) <> 0 THEN 30
40 REM Continue your program.
```

3. Under program control using a Jump to Subroutine.

An alternative to a POKE to RUNFLG is a direct jump into the utility at the appropriate entry point.

BASIC language users may execute the utility with the 'USR' command to location 7513 decimal ($1D59 Hex).

```
Example:      X = USR (7513)
```

If another argument is used, it will be used as the decimal address of a player missile. This will be explained in the next section. Any additional arguments will be ignored.

PILOT language users may execute the utility with a 'CALL' command to location 7516 ($1D5C Hex).

```
Example:      CALL:7516
```

Assembly and machine language users may also execute the utility with a Jump to Subroutine at location $1D5C Hex.

```
Example:      JSR $1D5C
```

This entry point assumes that the code was previously initialized either under AUTORUN.SYS or initialized with a call to the "relocatable initialization entry point" ORIGIN + 6.

In machine code, the Jump to Subroutine is actually simpler than changing RUNFLG. It also has the advantage of an IMMEDIATE screen dump to printer and the RUNFLG test to resume execution would not be necessary.

Aborting

If you wish to abort a screen dump once it has started, press the escape (ESC) key. The BREAK key will occasionally abort the dump, but it is unpredictable and should be avoided. The SYSTEM RESET key will always abort the dump, but it has been found to occasionally mess up the stack and should be avoided.

Conflict with DOS

The present location of the utility programs is such that a call to DOS may destroy vital entry point jump vectors. Consequently, the user has two options:

1. When the user completes his activity with the DOS Menu, he must reboot the disk to AUTORUN the utility.

2. The user may use the following procedure to avoid having to reboot the disk:

   a. Before starting the programming session, a MEM.SAV file should be created on the working disk as follows:
      i) Go to DOS
      ii) Create a MEM.SAV file with Menu item 'N'.

   b. Turn off the computer, wait five seconds, then turn it on again to boot the utility disk.

   c. Resume your programming session. Now when DOS is called, the utility will be saved in the MEM.SAV file. When the user completes his activity with DOS by using Menu item 'B', the utility will be automatically restored by the MEM.SAV file.

   d. The utility needs to be reinitialized to restore needed Jump vectors before it can be executed from the keyboard (CTRL-P) or by poking RUNFLG. Reinitialize as follows:

      i) BASIC language users initialize the utility with the USR command to location 7507 decimal ($1D53 Hex).

      Example: X = USR (7507)

      All other arguments will be ignored.

ii) PILOT language users initialize the utility with the 'CALL' command to location 7504 decimal ($1D50 Hex).

Example: CALL:7504

iii) Assembly language users initialize the program with a Jump to Subroutine at location $1D50 Hex.

iv) If the utility has been relocated and is imbedded in a user's program and located where the DOS DUP.SYS program will overwrite it, then the user will have to reinitialize the utility at location ORIGIN + 6.

The utility will now respond to the keyboard (CTRL-P) and RUNFLAG initiated execution.

SD1 TEXT ONLY

Screen Dump Utility SD1 will print any text on the screen when the program is invoked. Any graphics on the screen will be ignored. In BASIC graphics modes 1 and 2, double size characters will be approximated on the printer by spacing between character print positions.

Program SD1 requires approximately 1.3K bytes of RAM and resides in locations $1D50 to $226E.

When using the SD1 screen text dump utility with a 40-column printer such as the Atari 820 or 822, you will probably want to suppress the automatic carriage return and line feed option. Since the printer does a carriage return and line feed itself on line overflow, you will frequently get double line spacing with this option left on. To turn off the auto CR/LF, change location $1D67 Hex from 0 to 1. In BASIC or PILOT, this can be accomplished with a 'POKE 7527,1'. This procedure is not necessary with any 80 column printer, such as the Atari 825 , since there is no line overflow.

SD2 AND SD3 TEXT AND GRAPHICS

Screen Dump Utilities SD2 and SD3 will print both text and graphics on an Atari 822 printer. Printing is done under direct control of the print head on a dot-by-dot basis. The printer character font table is NOT used, so that printouts of text may look considerably different than conventional printing. Instead, printouts of text using SD2 or SD3 will approximate the font used in the Atari video generator. In other words, you will get a hardcopy of what you see on the screen.

SD2 and SD3 are quite similar, except that SD2 ignores players and missiles, while SD3 can print them if they are handled properly by the user program. SD2 is approximately 3.7K bytes long and resides in RAM locations $1D50 to $2A9B. SD3 is approximately 4.6K bytes long and resides in RAM locations $1D50 to $2E33.

Considerable flexibility in the printout is available with SD2 and SD3. In particular, you may wish to:

1. Increase the size or proportion of the printed image.

2. Print black on white or white on black.

3. Print in "grey scale" or black/white.

4. Determine grey scale from luminance or color.

5. Print data which has been "fine scrolled."

6. Print players and missiles (SD3 only)

These options are preset with default values to accommodate typical requirements for a graphics screen dump, but may be changed by the user if desired.

SCALING

The size of the printed image may be varied independently in both horizontal and vertical axes with two variables: VMODS and HMODS.

VMODS at $1D65 is the variable used to set the vertical scale. It must be an integer less than or equal to 16 and greater than 0. VMODS is the denominator of the fraction:

Vertical length = 16 / VMODS.

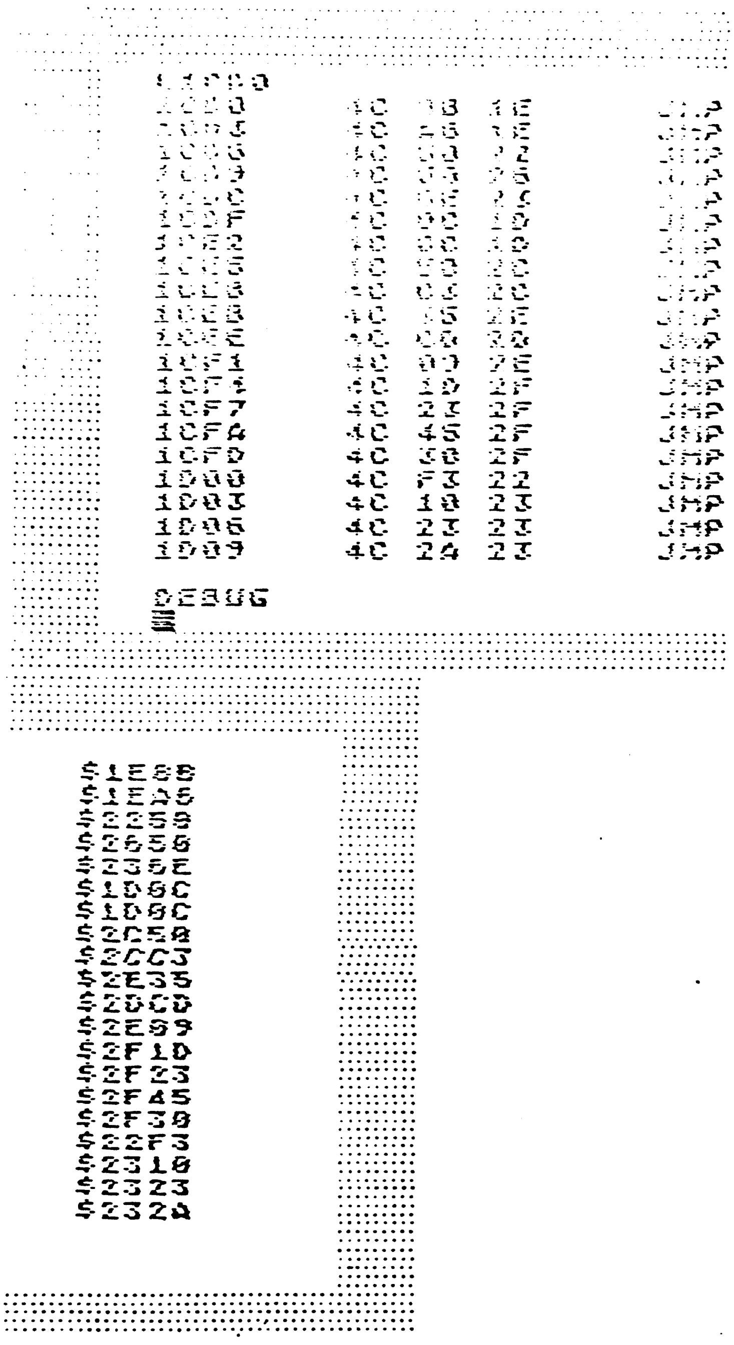
Therefore, as VMODS increases, vertical length decreases. This variable is initialized a preassigned value of 16 when the program initializes the printer.

HMODS at $1D66 is the variable used to set the horizontal scale. It must be an integer less than or equal to 16 and greater than 0. HMODS is the denominator in the fraction:

Horizontal width = 16 / HMODS.

Therefore, as HMODS increases, horizontal width decreases. This variable is also set to 16 when the program initilizes the printer.

As the horizontal scale increases to the point that the image will no longer fit on a single strip of paper, the remaining unprinted portion will carry over and print in a second strip below the first strip. In fact, the Atari 822 printer requires a minimum of two strips at the smallest scale. This is illustrated in Figure 7. These strips may be aligned and taped together to give an enlargement wider than the printer paper. As the scale increases still further, more than two strips of paper may be required to print the entire screen. Using this technique, it is possible to enlarge the screen using an Atari 822tm to approximately eight feet wide by five feet high with VMODS = HMODS = 1. This will require approximately 22 strips, 64 hours, and 110 feet of thermal paper!



```
L1CD0
1CD0          4C  8B  1E          JMP
1CD3          4C  A6  1E          JMP
1CD6          4C  59  22          JMP
1CD9          4C  56  26          JMP
1CDC          4C  5E  23          JMP
1CDF          4C  8C  1D          JMP
1CE2          4C  8C  1D          JMP
1CE5          4C  5A  2C          JMP
1CE8          4C  C3  2C          JMP
1CEB          4C  35  2E          JMP
1CEE          4C  CD  2D          JMP
1CF1          4C  89  2E          JMP
1CF4          4C  1D  2F          JMP
1CF7          4C  23  2F          JMP
1CFA          4C  45  2F          JMP
1CFD          4C  3B  2F          JMP
1D00          4C  F3  22          JMP
1D03          4C  10  23          JMP
1D06          4C  23  23          JMP
1D09          4C  24  23          JMP

DEBUG
```

```
$1E8B
$1EA6
$2259
$2656
$235E
$1D8C
$1D8C
$2C5A
$2CC3
$2E35
$2DCD
$2E89
$2F1D
$2F23
$2F45
$2F3B
$22F3
$2310
$2323
$2324
```

Figure 7. Graphics dump on Atari 822.

REVERSE IMAGE

Normally the printer prints black images on a white background. You can reverse the image to print white on a black background by simply pressing the Atari ( ) key on the keyboard. This key "toggles," so press it again to return to "normal." Figure 8 illustrates "normal" and "reverse" images.

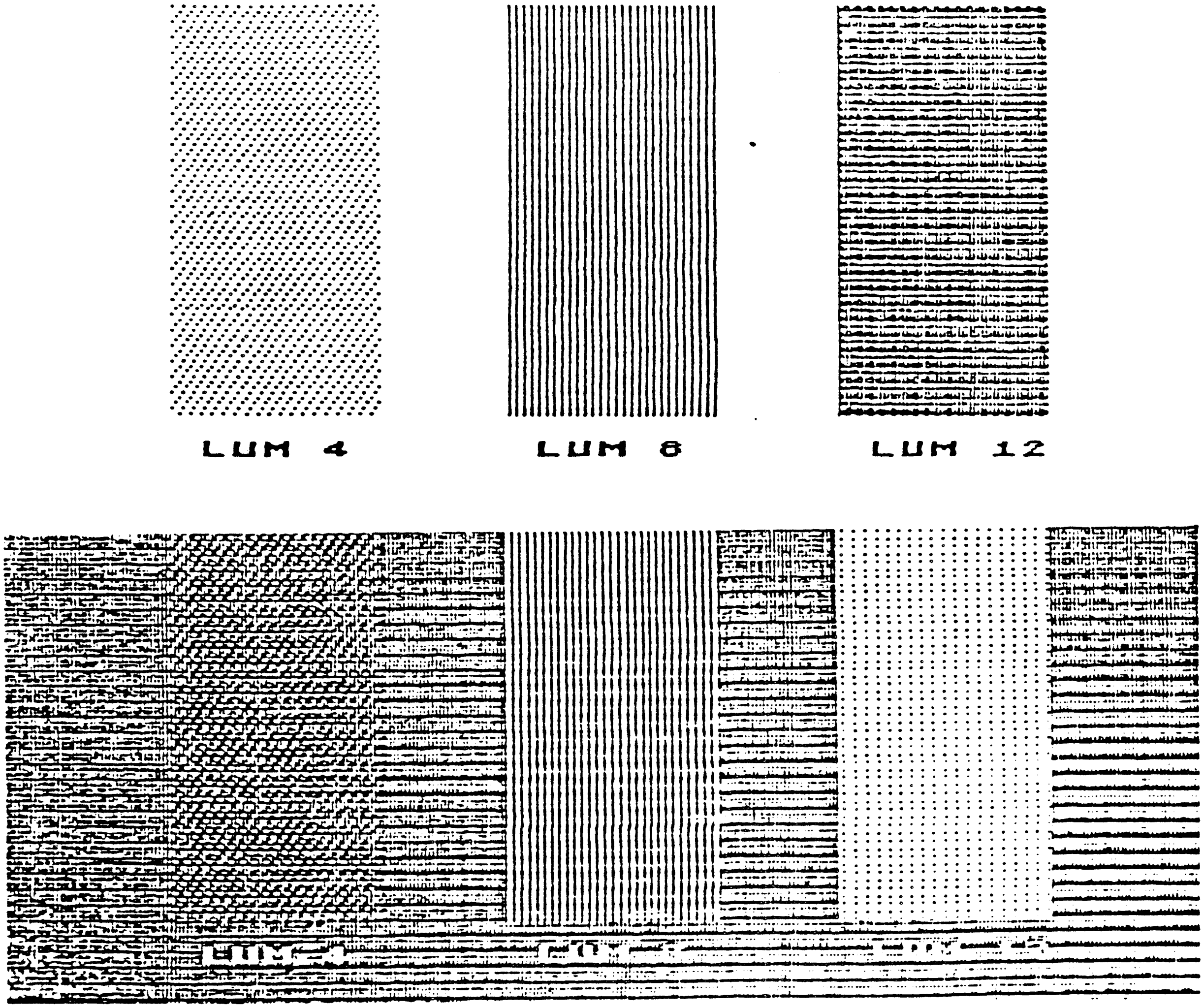


Figure 8. Normal and Reverse images

GREY SCALE

Obviously, the screen dump utility program cannot reproduce a color video display EXACTLY on a printer which has only black and white capability. An approximation of some sort is required. The program uses eight different dot patterns to approximate a grey scale. But how to represent various "colors" in the grey scale is itself a problem, since "color" consists of both hue and luminance. Should a low intensity red print the same as a very bright yellow? You can select whether the grey scale should correspond to luminance OR hue (but not both). The default is that grey scale corresponds to luminance. What this means is that two areas on the screen may look different because they differ in hue, but they will print exactly the same if they have the same luminance (assuming the default condition). Conversely, if grey scale is set according to hue, two areas on the screen may look different because of a difference in luminance, but print the same because the hue is the same. Figure 9 illustrates the eight grey scale patterns corresponding to either luminance or hue levels. (Remember, you can select grey scale on luminance OR hue, but not both.)

The user may select luminance or hue for grey scale by setting or clearing bit 6 in STATUS, a one byte variable located at $1D67 Hex (7527 decimal). Specifically, grey scale will correlate with luminance if bit 6 is zero, or correlate with hue if bit 6 is one.

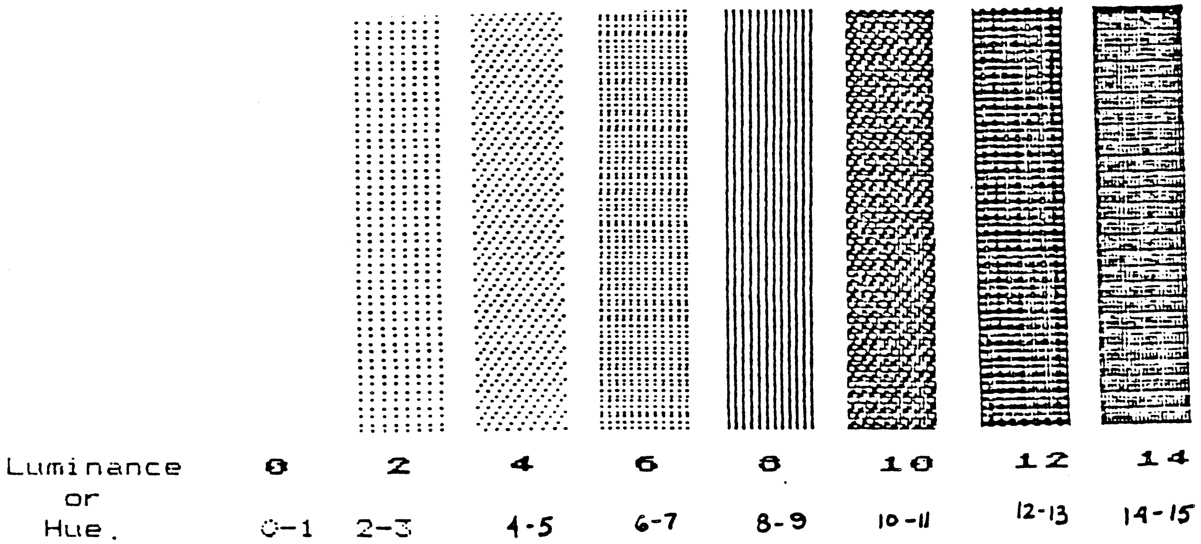


Figure 9. Eight Grey Scale Patterns

STATUS BYTE

The STATUS byte allows user selection of several options in addition to grey scale. Table 1 briefly describes these. The STATUS byte is located at $1D67 Hex (7527 decimal).

Table 1. - STATUS Byte bit assignments

BIT	DESCRIPTION OF OPTION (Default = CLR)
7	CLR = Preset "random" color registers are used SET = System color registers are used
6	CLR = Use Luminance for grey scale SET = Use Hue for grey scale
5	CLR = Text prints in Black & White SET = Text prints in grey scale
4	CLR = Players print in Black & White SET = Players print in grey scale
3	CLR = Missiles print in Black & White SET = Missiles print in grey scale
2	NOT USED
1	NOT USED
0	Used only in SD1 to determine CR/LF function. CLR = Carriage Return/Line Feed sent at end of each line. SET = Inhibit end of line CR/LF if exactly 40 characters are sent in a line.

"Random" Patterns

If the various hues are close to the same value and the various luminances are nearly the same brightness, then the grey scale patterns printed will be very similar or perhaps identical. This would make it difficult to distinguish the different areas of the image. In this case, the user may wish to print the screen using a different set of patterns. A set of random patterns has been provided and may be used by setting "STATUS" Bit 7 on. Now the picture will be printed using a random selection of grey scale pattern for each color register. This frequently results in a better looking and more easily discernable printout of the image. A different set of random patterns is used for hue and for luminance, so that the printout should look different depending upon the state of Bit 6. The user may change the "RANDOM" values of the registers in the same way the system registers are changed. The preset registers are organized in the same order using the same scheme as the system registers found at Hex location $2C3 through $2C8. The preset registers are located at $1D78 Hex (7544 decimal) through $1D80 Hex (7552 decimal).

Text Grey Scale

If Bit 5 is set, text will be printed in the grey scale pattern corresponding to the appropriate color register setting. Text usually looks best printed in a "solid" pattern, however, so the default for Bit 5 ignores grey scale on text characters.

Player/Missile Grey Scale

Bits 4 and 3 are similar to Bit 5 except that they select grey scale or "solid" patterns for Players and Missiles, respectively. This is used only in SD3.

FINE SCROLLING

Lines on the screen that have been fine scrolled horizontally or vertically may be accurately printed, provided that the user updates the program's scrolling registers whenever he changes the CTIA registers. Since the Atari computer system stores fine scroll data in "write-only" registers. it is necessary for the user to make this information available to the screen dump program. The Horizontal Scroll Register is located at $1D68 Hex (7528 Decimal) and must contain the same value written to the corresponding CTIA Register at address $D404 Hex. Similarly, the Vertical Scroll Register is located at $1D69 Hex (7529 decimal) and must contain the same value written to the CTIA Register at $D405 Hex.

PRINTING PLAYERS/MISSILES

Programs SD1 and SD2 will ignore all Player-Missile Displays. SD3 will accurately print Players and Missiles provided that the user updates 14 bytes of data relating to Player Missile size and location. Since the system stores Player/Missile data in "write-only" registers, it is necessary for the user to make this information available to the screen dump program. The purpose and location of the needed data is as follows:

	Purpose	CTIA Location	Program Location
1.	Player-Missile Vertical Data Byte Address (high order byte only).	$D407	$1D6A
2.	Horiz. Position of Player 0	$D000	$1D6B
3.	Horiz. Position of Player 1	$D001	$1D6C
4.	Horiz. Position of Player 2	$D002	$1D6D
5.	Horiz. Position of Player 3	$D003	$1D6E
6.	Horiz. Position of Missile 0	$D004	$1D6F
7.	Horiz. Position of Missile 1	$D005	$1D70
8.	Horiz. Position of Missile 2	$D006	$1D71
9.	Horiz. Position of Missile 3	$D007	$1D72
10.	Horiz. Size of Player 0	$D008	$1D73
11.	Horiz. Size of Player 1	$D009	$1D74
12.	Horiz. Size of Player 2	$D00A	$1D75
13.	Horiz. Size of Player 3	$D00B	$1D76
14.	Horiz. Size of All Missiles	$D00C	$1D77

If the user prefers, he may keep these bytes, the Horizontal Scroll Byte and Vertical Scroll Byte, in his program locally and provide the starting address of these bytes in one of two ways:

1. Store the starting address in locations $3E8 and $3E9 (1000 and 1001 decimal).

2. BASIC language users may provide the address as the first argument in the USR statement.

```
     Example:     X = USR (7513,16384)
```

The utility will get the needed 16 bytes of data starting at location 16384 decimal. All other arguments will be ignored by the utility.

The user must keep the bytes in the following order:

1. Horiz. Scroll Value
2. Vertical Scroll Value
3. Player-Missile Vertical Data Pointer
4. Horiz. Position of Player 0
5. Horiz. Position of Player 1
6. Horiz. Position of Player 2
7. Horiz. Position of Player 3
8. Horiz. Position of Missile 0
9. Horiz. Position of Missile 1
10. Horiz. Position of Missile 2
11. Horiz. Position of Missile 3
12. Horiz. Size of Player 0
13. Horiz. Size of Player 1
14. Horiz. Size of Player 2
15. Horiz. Size of Player 3
16. Horiz. Size of All Missiles

These 16 bytes will be written into the LOCAL Player missile RAM area in the utility program each time the screen print program is executed. If the user pointer at $3E9 is set to zero, and if no argument is supplied in a BASIC USR(7513) statement, no data will be moved. Please note that the high byte of the user pointer ($3E9) must be zero if the user enters the data directly into the local utility program player missile RAM area; otherwise, this area will be written over.

PRINTER CONSIDERATIONS

The Atari 820 is a 40- column dot matrix printer which plugs into the serial I/O bus. It has no graphics capability and will work only with SD1.

The Atari 825 is an 80-column dot matrix printer which plugs into the Atari 850 Interface Module. It has no graphics capability and will work only with SD1.

The Atari 822 is a 40-column thermal printer which plugs into the serial I/O bus. In the graphics mode it will print a single row of 280 dots on each horizontal pass. This implies two limitations:

1. Since it would require a minimum of 320 dots horizontally to print all the information on the screen, it will require more than one paper width to print the entire image, even at the minimum scaling.

2. Even though the program uses bidirectional printing with the Atari 822, it will still take a lot of time to print graphics. At the minimum scaling, it will take approximately 15 minutes to print the entire screen.

APPENDIX A - ERROR CODES

The screen print program may abort execution for a variety of reasons. The cause is indicated in an error code byte located at $1D64 Hex (7524 decimal). The error codes and corresponding error conditions are as follows:

Error Code	Cause
1	No error - the screen print program ran successfully.
128	The BREAK key was pressed during program execution.
240	The ESCAPE key was pressed during program execution.
241	The RESET key was pressed during program execution.
250	GTIA Mode - Text print program SD1 cannot print GTIA data.
251	No DMA - Video DMA disabled.
252	The printer failed to open properly.
253	The Serial I/O Bus is active. The program will not execute unless the Serial Bus is free.
254	The System Print buffer is not empty, indicating that the printer is in use. The program will not execute unless the printer is not busy at the time the program is called.
255	The screen print program has not been initialized yet. It must be initialized prior to program execution.

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.

ATARI PROGRAM EXCHANGE

REVIEW FORM

We're interested in your experiences with APX programs and documentation, both favorable and unfavorable. Many software authors are willing and eager to improve their programs if they know what users want. And, of course, we want to know about any bugs that slipped by us, so that the software author can fix them. We also want to know whether our documentation is meeting your needs. You are our best source for suggesting improvements! Please help us by taking a moment to fill in this review sheet. Fold the sheet in thirds and seal it so that the address on the bottom of the back becomes the envelope front. Thank you for helping us!

1. Name and APX number of program ____

2. If you have problems using the program, please describe them here.

____

____

____

3. What do you especially like about this program?

____

____

____

4. What do you think the program's weaknesses are?

____

____

____

5. How can the catalog description be more accurate and/or comprehensive?

____

____

6. On a scale of 1 to 10, 1 being "poor" and 10 being "excellent", please rate the following aspects of this program?

_____ Easy to use
_____ User-oriented (e.g., menus, prompts, clear language)
_____ Enjoyable
_____ Self-instructive
_____ Useful (non-game software)
_____ Imaginative graphics and sound

7. Describe any technical errors you found in the user instructions (please give page numbers).

8. What did you especially like about the user instructions?

9. What revisions or additions would improve these instructions?

10. On a scale of 1 to 10, 1 representing "poor" and 10 representing "excellent", how would you rate the user instructions and why?

11. Other comments about the software or user instructions:

```
 -----
|     |
|STAMP|
|     |
 -----
```

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]