Micro Technology Unlimited

K-1009-1C AIM-65 PRINTER ENHANCEMENT PACKAGE

GRAPHIC QUICK PRINT

GRAPHIC QUALITY PRINT

TEXT PRINT

APRIL 1980

Micro Technology Unlimited
841 Galaxy Way
PO Box 4596
Manchester, NH 03108

TABLE OF CONTENTS

K-1009-1 TEXT/GRAPHICS PRINTOUT PACKAGE USER'S MANUAL

This software package greatly enhances the capabilities of the AIM-65 on-board printer without any modifications whatsoever to a properly functioning printer. Two different printout programs are provided, one which can print the contents of our Visible Memory (or any contiguous 8K block of memory) as an accurate dot-for-dot image on a 320 by 200 grid, and the other which can print the contents of the text buffer maintained by the AIM-65 Text Editor as full lines of text without breaks every 20 characters. This is accomplished by printing the text sideways as strips of 10 lines. The standard line length is 60 characters, which is limited by the AIM editor, but the program is capable of line lengths up to 127 characters if the text is formatted correctly in memory.

Both programs are written in AIM-65 machine language and are provided on the enclosed cassette in binary object code form at several different memory locations. Listed in the appendix are the addresses of major routines and data tables in each program. Source listings, custom modifications, and licenses for these programs are individually negotiated. Please contact MTU for details.

CASSETTE FILE LISTING

The following programs are recorded on the enclosed cassette in standard AIM-65 format with a gap length of $08.

1.	SCPHI	0CD8-0FFF	Screen print for images created by SWIRL and VLIFE in the K-1008-5 package
2.	SCP04	0456-0771	Screen print for use with BAS04 in the K-1008-5 package
3.	SCP20	4456-4771	Screen print for use with BAS20 in the K-1008-5 package
4.	SCPGR	0366-0681	Screen print for standalone use with GRAPH in the K-1008-5 package
5.	TXPTL	0200-07FB	Text print in low AIM-65 memory
6.	TXPTH	0A04-0FFF	Text print in high AIM-65 memory

Since the high speed AIM-65 tape format is somewhat more sensitive to cassette recorder differences than KIM-1 format, it may be necessary to try more than one cassette recorder or experiment with the head alignment. It is suggested that a local copy be made of each program the first time it is successfully loaded to protect against loss and tape degradation. Place the enclosed copyright label on the backup tape. Unreadable cassettes will be replaced by MTU but since they will be recorded on the same machine, the only variable will be tape dropouts.

SCREEN PRINT USAGE WITH THE SWIRL AND VLIFE DEMONSTRATION PROGRAMS

When using the screen print program it is of course necessary to form the desired image in graphic memory. This is normally done with the various programs in the K-1008-5C software package. Although it is very desirable to use a Visible Memory to hold the image pattern for previewing before printing, any contiguous 8K block of memory can be used for that function. This section will describe methods by which the output of the SWIRL and VLIFE demonstration programs in the K-1008-5C software package can be printed on the AIM-65 printer.

Assuming that a suitable image has been created in graphic memory, follow these steps to print the image:

1. Place the K-1009-1C cassette in the recorder and instruct the AIM-65 to load the file named SCPHI. The program will load into the topmost locations in a 4K AIM thus preserving the demonstration program that created the image.

2. The only information the screen print program needs is the page number of the beginning of graphic memory. This must be stored in location 0107. Normally it would be $80 if the Visible Memory is jumpered for addressing from $8000 to $9FFF.

3. Normally Screen Print is a subroutine but SCPHI has had call and return instructions assembled into it. With the AIM monitor, start execution at 0CD8 for a quick print of the screen contents.

4. The printer should energize and feed about one inch of paper before starting the actual screen print. The screen print itself should be about 5 3/8 inch long and then the printer should stop. Except for being stretched horizontally about two-to-one, the printout should exactly match what was on the screen. If the image is scrambled or starts out OK and later becomes scrambled, see the troubleshooting section of this manual. The print samples on page 9 show the print quality that can be expected from a properly adjusted AIM printer.

5. For a quality print start execution at 0CDE. In this case the upper half of the screen is printed first, about 1 inch of space, and the lower half is printed. The two image halves may now be taped together for a reasonably well proportioned image with improved dot definition. Use Scotch Highlander cellophane packing tape; other brands will eventually fade the image.

IMPORTANT USAGE NOTES

Although the screen print program is very easy to use, there are some potential pitfalls. If it is necessary to halt the printout before it is complete, the user **MUST** use the Reset button on the AIM, not the single-step switch. Using Reset is necessary to insure that the thermal printhead elements are turned off; otherwise they will overheat and be damaged.

Please note that the program swaps zero page locations 02-1F with some of the invisible locations in the Visible Memory in order to make integration into user software as easy as possible. Page zero is normally restored at the end of the print operation but if printing is interrupted, page 0 will NOT be restored.

Finally, the print program is extremely time sensitive so it disables interrupts while it is executing. The initial state of the interrupt disable flag is restored when it returns.

SCREEN PRINT USAGE WITH THE K-1008-5C BASIC SUPPORT PACKAGE

There are two versions of the screen print program on the cassette for use with the AIM-65 BASIC Graphics Support routines in the K-1008-5C package. After loading and linking the appropriate screen print program with BAS04 or BAS20, one can initiate screen prints from BASIC without using the AIM-65 monitor. Use the following procedure to link Screen Print with BASIC Graphics Support:

1. Follow the loading procedure for the BASIC interface program (BAS20 or BAS04) as given in the K-1008-5C manual up through step 2.

2. Load the appropriate file from the K-1009-1C tape; SCP20 with BAS20 or SCP04 with BAS04.

3. Link the two programs together by modifying the following locations with the AIM monitor:

BAS20	BAS04	
4827 55	0827 55	Linkage to quick print from USR(10)
4828 44	0828 04	
4829 59	0829 59	Linkage to quality print from USR(11)
482A 44	082A 04	
4BE4 20	0BE4 20	Linkage to auto quick print if location 0106=1
4BE5 56	0BE5 56	(see discussion below)
4BE6 44	0BE6 04	
4BEE 20	0BEE 20	Linkage to auto quality print if location 0106=2
4BEF 5A	0BEF 5A	(see discussion below)
4BF0 44	0BF0 04	

4. If desired the two linked programs may now be written to tape as one file for easier loading in the future.

5. Follow steps 3 through 5 in the K-1008-5C manual.

6. Answer MEMSIZ= with 1109 for a 4K AIM or 17493 for a 20K AIM.

7. Complete the procedure in the K-1008-5C starting with step 7. In step 8 BASIC will reply with 579 bytes free for a 4K AIM or 16963 bytes free for a 20K AIM.

PRINTING THE SCREEN

With the screen print program linked in, there are two ways to cause a screen print from BASIC. One is by executing the statement Z=USR(10) for a quick print or Z=USR(11) for a quality print. (Be sure that the USR link in locations 0004 and 0005 is set to A9 (169 decimal) and 47 (71 decimal) for 20K AIM's or A9 and 07 for 4K AIM's.) The statements may either be typed in and executed immediately or issued by a BASIC program. The other way is by setting location 0106 (262 decimal) to a 1 for quick printing or a 2 for quality printing. Then whenever the screen fills with text and a scroll would normally take place, a print takes place automatically instead and the screen is cleared. To disable this feature and restore normal screen scrolling, set 0106 to anything else.

Please note that the screen print program should not be interrupted during execution. If it is absolutely necessary to interrupt, press the RESET button on the AIM-65. While this will stop the printer safely, page zero will be lost and BASIC will have to be reinitialized thus loosing the BASIC program in memory at the time.

SCREEN PRINT USAGE FROM MACHINE LANGUAGE

The last version of the screen print program on the cassette is called SCPGR and is intended for use with the GRAPH program in the K-1008-5C package. GRAPH and SCPGR are intented to be used with user written application programs in machine language. SCPGR loads right below the Tier 3 routines of GRAPH and leaves about 357 bytes for the user's application program in a 4K AIM (exclusive of page 0 and the stack).

Usage of SCPGR is quite simple. First the page number of the Visible Memory must have been stored in location 0107. This location is normally used by the fourth breakpoint address in the AIM debug program. In order to preserve the value stored there, the user must not define the fourth breakpoint address. This still leaves 3 breakpoints available; adequate for all but the most stubborn bugs.

After setting the VM address in 0107, the user simply does a JSR to 0366 for a quick print of the screen or a JSR 036A for a quality print. The program will print the screen in the corresponding format and return with all registers and all status preserved except the carry flag. This includes the state of the interrupt disable and decimal mode flags. Note however that interrupts are disabled during execution of the program. Page 0 is also completely restored so the user need not worry about page zero conflicts. In operation, two page zero locations are saved on the stack and an additional 30 (decimal) are saved in the first 30 locations immediately following the visible portion of the Visible Memory (locations 9F40-9F5D id the Visible Memory is addressed at 8000).

Note that if the program is relocated by hand or reassembled that certain sections of code must not cross a page boundary. These are critical timing routines and being split accross a page boundary will throw off the timing by as much as 20% which will turn the printout into a garbled mess. See the Theory of Operation and Troubleshooting section for additional information.

TEXT PRINT USAGE WITH THE AIM-65 TEXT EDITOR

One of the unique features of the AIM-65 is its built-in text editor. Although the editor does an excellent job with programs and other text, listings of the text on the 20 column AIM printer are difficult to read because of line fragmentation. The Text Print program will print the contents of the text editor buffer sideways on the paper, thus line length is theorectically unlimited. This is particularly suitable for final documentation of a program or even word processing since the text print program can print lower case characters (note however that the standard AIM-65 keyboard routine does not allow the input of lower case characters). The primary advantage of the text print program in text-only applications is that a Visible Memory is not required, it is smaller than a combination of DTEXT and the screen print program, and lines longer than 53 characters may be accomodated.

Two versions of the Text Print program are supplied on the included cassette. TXPTL starts at 0200 and extends up to 07FB. This is a convenient location since the text buffer may then start at 07FC and extend up as far as the installed memory goes. TXPTH is the same program assembled into the upper part of a 4K AIM's memory. This version can be used when it is necessary that the text buffer start at 0200 for some reason.

When entered, the text print program looks at locations 00E3 and 00E4 to find the address of the text buffer. These locations are used for this purpose by the AIM text editor so they should normally already have this information stored there. The program will then print "pages" of 10 lines of text each until a line with a zero byte (ASCII NULL) is found at which point it stops. If the text length is not a multiple of 10 lines, the last page will be only partially filled.

The width of each page is normally set to 60 decimal (3C hex) which is the maximum line length that the AIM editor can handle. This value can be changed however if it is known that all lines are less than 60 or that text prepared with some other program contains lines longer than 60. The alternate line length can be stored in location 0201 for TXPTL or location 0A05 for TXPTH. Lengths of zero or greater than 127 are illegal. If a line is encountered that is longer than the page width, the excess characters are truncated and lost.

LOADING INSTRUCTIONS

1. Decide which version of the program to use and load it from the K-1009-1C tape with the AIM monitor. The low memory version, TXPTL, is recommended for most uses.
2. Prepare the text to be printed with the AIM text editor or another program that stores its text in a compatible manner (carriage returns at the end of each line with a null stored after the final carriage return). When using TXPTL be sure to answer the FROM= prompt with an address greater than 07FB (0800 is recommended) When using TXPTH, be sure to answer the TO= prompt with an address less than 0A04 (0800 is recommended).
3. Be sure the address of the text is stored in locations 00E3 and 00E4. The AIM text editor automatically does this if it is used.
4. If a line length other than 60 is desired, store it in location 0201 (TXPTL) or location 0A05 (TXPTH).
5. With the AIM monitor, start execution at 0200 (TXPTL) or 0A04 (TXPTH). The paper should start feeding and after about an inch, text should appear. If more than 10 lines are in the text buffer, additional pages separated by an inch will be printed until the terminal null is seen in the text buffer. Printing should only be interrupted with the RESET button, otherwise the printer motor and print elements may be left on which will damage the printer.

Note that Text Print is a program rather than a subroutine. When printing is complete, a jump is taken to the AIM-65 monitor. Page zero from 0000 to 0030 is used and is not swapped out. Although no assumptions are made about what is stored in these locations upon entry, they will be permantly changed during the course of Text Print operation. This usage of page zero does not bother the AIM text editor since it uses the upper end of page 0 for storage but it could affect a user text preparation program.

THEORY OF OPERATION

The nicest thing about the AIM-65 printer is that it is totally under software control. This not only reduces parts count and production costs but makes sophisticated but unforseen applications (such as embodied in this package) possible with nothing more than a software change. Seldom does a design concept have so many important advantages without any significant disadvantages.

Basically the AIM printer consists of a moving head for horizontal scanning of the paper and a means of advancing the paper at the end of each scan. The print head itself has 10 tiny thermal elements that may be independently heated or cooled under the control of 10 output port bits. The thermal elements are designed so that they can heat and cool in less than 2 milliseconds. While an element is hot, it leaves a blue trail on the paper about 1/50 of an inch wide. Brief (1.4 milliseconds) application of current will cause a nearly round dot to be printed. Because of the response speed of the elements, they can easily overheat in just a few milliseconds if not operated properly.

In normal operation the head scans back-and-forth with a throw of about .2 inches. Since the 10 elements are themselves spaced by .2 inches, an area 2 inches wide is scanned by the head. Note that printing is done both left-to-right and right-to-left thus doubling the print speed but also introducing a potential alignment problem. Printing characters, or for that matter any dot pattern, then is simply a matter of turning on the proper elements at the proper time while the head is scanning.

To simplify the standard AIM print program, the printer generates two sets of timing pulses appropriate for printing 20 columns of 5 by 7 dot matrix characters. across the paper. One of these is the Start pulse which marks the beginning of a left-right scan pair and serves to synchronize the program to the direction of head travel. The other signal is a square wave-like pulse train that changes state whenever a character dot should be printed. The timing of this pulse train is set by the printer manufacturer for two characters 5 dots wide separated by 2 dots to be covered by each thermal element. Since there are 10 thermal elements, there are actually 10 characters being printed at any point in time. This timing signal also sets the horizontal character spacing by incorporating a 2 dot gap between the two sets of 5 transitions that occur during a half-scan. Both of these timing signals are generated by a reed switch/magnet assembly in the base of the printer, thus the signals are somewhat noisy as well.

Since the dot timing signal includes inter-character spacing, it is not possible to use this signal in a graphics print application. What is actually necessary is to split up the time between Start pulses into a number of equal intervals and thus generate the timing strobes internally with software. Thus it is necessary to first <u>measure</u> the printer speed to calculate the software strobe timing and then <u>track</u> variations in that speed making appropriate adjustments in the software strobes while printing is taking place. If speed tracking is not done, the software will gradually get out of sync with the print head movement.

SCREEN PRINT OPERATION

When Screen Print is entered the first operation is saving the registers and page zero followed by setting pointers, increments, and flags according to whether a quick print or a quality print is desired. Following this the printer motor is turned on and 5 forward and back print cycles are allowed to occurr in order for the printer speed to stabilize. For the next 16 print cycles the average cycle time accurate to one microsecond is determined by measuring the total in 16 micro-second increments. During the seventeenth cycle, the proportionate times allowed for various tasks during succeeding print cycles are calculated from the cycle time using the 16 bit multiply and divide routines that are part of the program. After all of the timing is determined, a wait for the end of the 17th cycle is entered and when the Start timing pulse is seen, the print routine proper is entered.

The actual printing part of each half of the scan cycle is divided into 20 parts which correspond to the 20 dots that each thermal element prints. For a quick print each dot position would receive a single bit from the Visible Memory. For quality print each Visible Memory bit controls two dot positions. Printing of the entire screen width requires 160 left-right print cycles to complete. After this is done, the quality print flag is tested. If it is on, it is turned off, some pointers are adjusted, and the program is re-entered at the speed measurement point. Remeasurement of the speed provides space between the two screen halves and compensates for long-term motor heating effects. When the printout is complete, page zero, the registers, and the status are restored before returning to the calling program.

TEXT PRINT OPERATION

Much of the text print program is the same as the screen print program. The main difference is that the dot timing programming is different so that only 10 dot slots with appropriate interline spacing are generated. Also the program has its own character shape table so that lower case characters can be printed. In operation the program maintains 10 text pointers which correspond to the 10 lines being printed at any given time. These pointers scan the 10 lines and the characters are translated into dot patterns on the fly for printing. Since Text Print is a program rather than a subroutine, there is no saving of page zero or the registers.

TROUBLESHOOTING

Since the screen print and text print programs work by interpolating the timing signals provided by the AIM printer, certain printer maladjustments that have little effect on normal printer operation can adversely affect the operation of these programs.

Basically the programs depend on the printer motor speed being reasonably constant (within 2%) after 5 warmup cycles. A speed tracking function in the software takes care of variations less than 2% but larger variations can cause the program to loose synchronization with the print cycle. When this occurrs the dot patterns break up entirely. This would be particularly evident on horizontal lines printed by the screen print program. The cause of excessive motor speed variation is almost always binding of the drive motor gear or occasionally insufficient mesh which causes an occasional slipped cog. Follow the instructions on the AIM user's manual for adjusting the motor gear mesh.

The actual speed of the printer motor is not very important but if it is too fast, the dots may be smeared somewhat. On the other hand, two slow a speed will increase speed variation. The optimum printer speed is 120 milliseconds per left-right print cycle. On some printers at certain speeds the program may be unable to attain synchronization with the print cycle and thus produce a garbled result. In such cases, try different print speeds (between print runs, not during them) to see if the problem clears up. The potentiometer marked VR3 on the AIM board is the print speed adjustment.

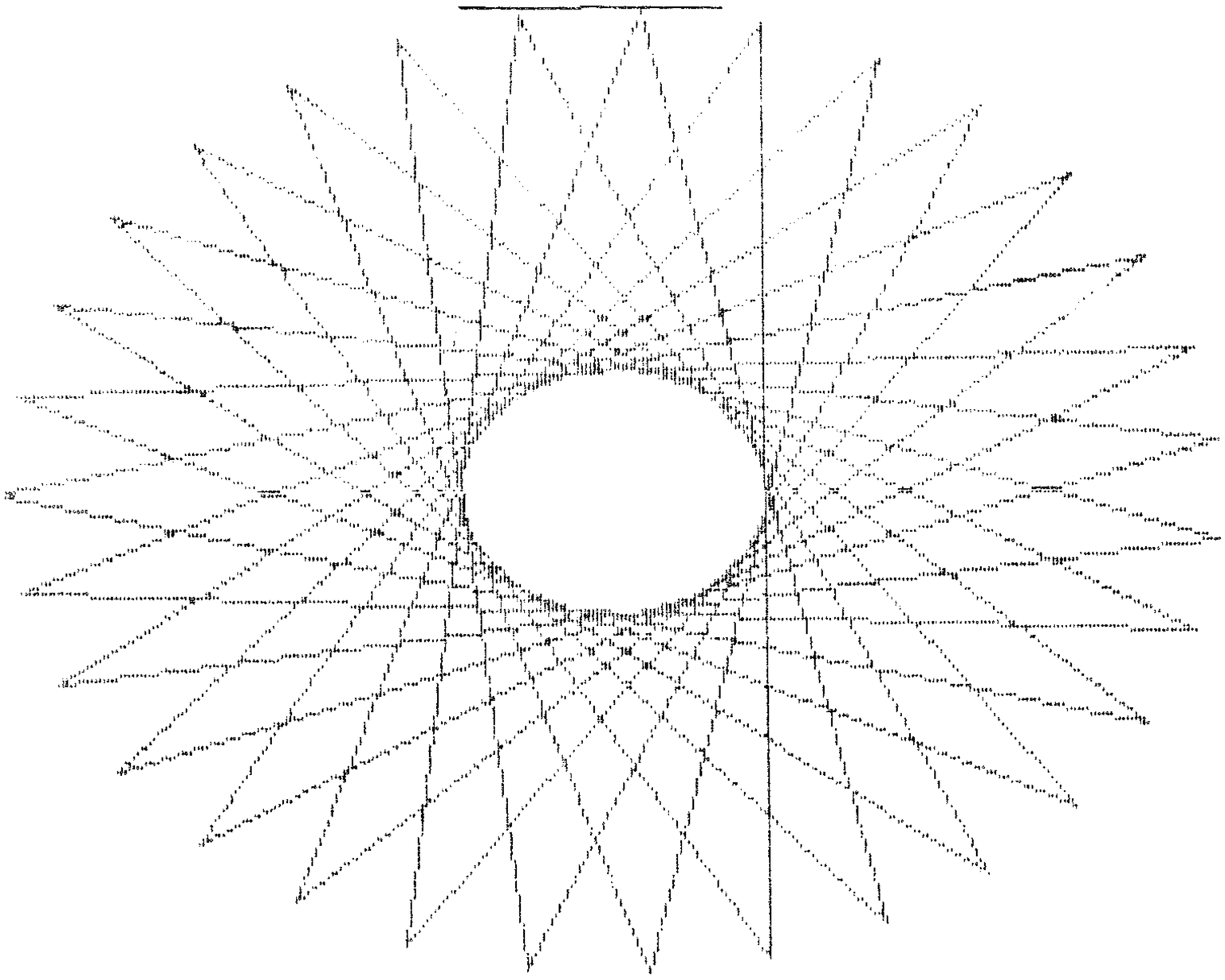
There is also a print density adjustment on the AIM. Best results with the screen print program in quick print mode are usually obtained with this control set for maximum density. Also some brands of paper have a lower "development temperature" and print more uniformly. Photographs of AIM printout for reproduction are best done with panchromatic film and a red filter.

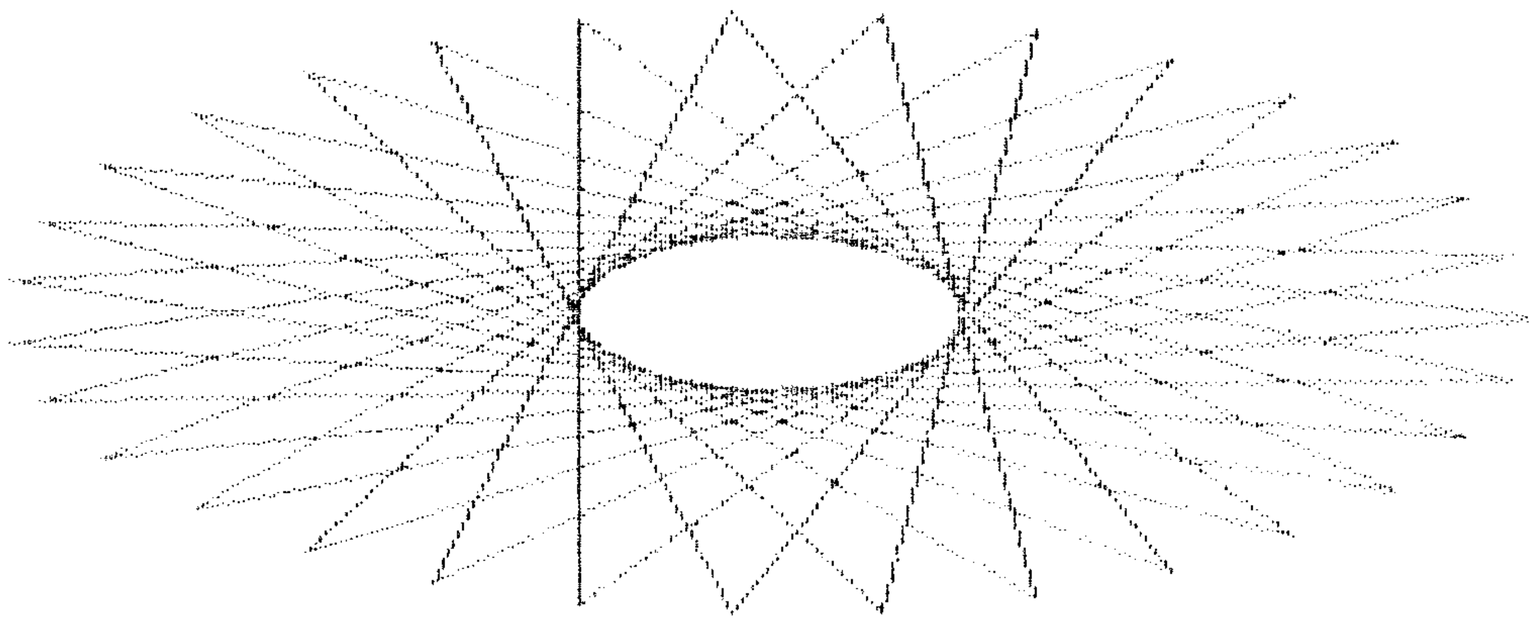
Because the printer prints in both directions, it is necessary that the left-going and right-going portions of the cycle be precisely balanced so that horizontal lines of dots (vertical on the paper) line up properly. Generally it is not possible to do this perfectly but results equal to the print samples on page 9 should be obtainable. If the dots consistently fail to line up, there is a mechanical adjustment on the printer that will make them line up. See the AIM user's manual for details on making this adjustment. Optimum adjustment for normal AIM printing should also be optimum for the programs in this package.

If proper operation of these programs cannot be obtained and the user has reassembled them, the problem might be that a critical timing loop has been split accross a page boundary. This is a problem because a conditional jump on the 6502 requires an extra cycle to jump accross a page boundary. The following locations in the various programs have critical timing:

SCPHI	SCP04	SCP20	SCPGR	TXPTL	TXPTH
0D90-0DA3	0502-0515	4502-4515	0412-0425	0296-02AC	0A9A-0AB0
0DF3-0DFC	0565-056E	4565-456E	0475-047E	0303-030C	0B07-0B10
0E05-0E0B	0577-057C	4577-457C	0487-048D	0315-031B	0B19-0B1F
0F03-0F05	0675-0678	4675-4678	0585-0588	03D0-03D2	0BD4-0BD6
0F27-0F3B	0699-06AE	4699-46AE	05A9-05BE	0766-077B	0F6A-0F7F

PRINT SAMPLES





THIS PROGRAM IS DESIGNED TO PRINT THE CONTENTS OF THE AIM-65 TEXT EDITOR BUFFER AS FULL 80 CHARACTERS LINES. THE TEXT IS PRINTED SIDEWAYS AS STRIPS OF 10 LINES EACH. THE FULL 96 CHARACTER ASCII FONT WITH LOWER CASE DECENDERS IS SUPPORTED. THE NUMBER OF CHARACTERS PER LINE (UP TO 127 CHARS/LINE) MAY BE ALTERED BY ENTERING AT A SECOND ENTRY POINT.
THIS PROGRAM HAS TWO ENTRY POINTS:
TXPRT IS THE ENTRY POINT FOR PRINTING 80 CHARACTERS PER LINE AS

DEFINED BY THE AIM-65 TEXT EDITOR. TXPRTI IS THE ENTR

ADDRESSES OF CONSTANTS, VARIABLES, AND MAJOR ROUTINES

SCREEN PRINT PAGE 0

Name	Page 0	Visible Memory	Description
ADPT1	0000	on stack	Address pointer number 1
MODE	0002	1F40	Print mode flag, bit 7=0=Quality print, =1=Quick print
HALF	0003	1F41	Screen half counter for quality print
SKIP	0004	1F42	Number of VM bytes covered by a single thermal element
BACKSL	0006	1F44	Number of VM bytes covered by all 10 thermal elements
THR8	0008	1F46	Image of right 8 thermal elements
THL2	0009	1F47	Image of left 2 thermal elements
PTHR8	000A	1F48	Previous image of right 8 thermal elements
PTHL2	000B	1F49	Previous image of left 2 thermal elements
SCNCNT	000C	1F4A	Used to count dots printed per thermal element
BMASK	000D	1F4B	Bit mask used to select bits from the Visible Memory
DMYCYC	000E	1F4C	Used to count dummy print cycles during speed track stabilize
SPLSTM	000F	1F4D	Speed error in current print cycle
LPFDLA	0010	1F4E	Used for speed tracking
LPFOUT	0011	1F4F	Speed correction in current print cycle
TPHASH	0012	1F50	High bytes of 4 calculated phase times
TPHASL	0016	1F54	Low bytes of 4 calculated phase times
MPCD	001A	1F58	Multiplicand/divisor for arithmetic routines
PROD	001C	1F5A	Multiplier/product/dividend/quotient for math routines

SCREEN PRINT MAIN PROGRAM (note that addresses are given for SCPHI only)

QKP	0CD8	Quick print convenience entry from AIM monitor
QLP	0CDE	Quality print convenience entry from AIM monitor
QKPRNT	0CE4	Normal subroutine entry to quick print
QLPRNT	0CE8	Normal subroutine entry to quality print
	0D52	Turn on printer motor and wait for 5 print cycles
NXTHLF	0D65	Time 16 print cycles
	0DA7	Calculate phase times from measured printer speed
WNDOW0	0DF1	Determine current speed error
PHAS1	0E2E	Start of left going half of print cycle
	0E66	Start of right going half of print cycle
PRTEND	0EA8	Test print mode and then repeat or shutdown
PRSTOP	0EB8	Turn printer motor off and return
THPRT	0ECD	Set up thermal elements and print for one dot time
LDELAY	0F27	Precision delay subroutine
WAPCYC	0F3D	Wait for printer start pulse subroutine
P0SWAP	0F4A	Swap page zero with end of Visible Memory
SBACK	0F6A	Subtract BACKSL and Add constant from K table to ADP1
ADADP1	0F77	Add constant from K table to ADP1
UNSMPY	0F87	Unsigned 16 bit multiply
SRQA	0FAD	Quad shift PROD right with sign extend
SRQL	0FB0	Quad shift PROD right logical
SLQL	0FB9	Quad shift PROD left with zero shifted in
RLQL	0FBA	Quad shift PROD left with carry shifted in
UNSDIV	0FC3	Unsigned 32/16 divide
TPPHI	0FE6	High bytes of phase percentages
TPPLO	0FEA	Low bytes of phase percentages
TOVHHI	0FEE	High bytes if phase overhead times
TOVHLO	0FF2	Low bytes of phase overhead times
K	0FF6	Table of offsets for manipulating ADP1

TEXT PRINT PAGE 0

Name	Address	Description
CHPLN	0000	Maximum number of characters per line
THR8	0001	Image of right 8 thermal elements
THL2	0002	Image of left 2 thermal elements
SCNCNT	0003	Used to count dot rows per character
LNCNT	0004	Used to count character dot lines
CHCNT	0005	Used to count number of characters on a line
BMASK	0006	Bit mask used to select bits from font table
SPLSTM	0007	Speed error in current print cycle
LPFDLA	0008	Used for speed tracking
LPFOUT	0009	Speed correction in current print cycle
TPHASH	000A	High bytes of 5 calculated phase times
TPHASL	000F	Low bytes of 5 calculated phase times
MPCD	0014	Multiplicand/divisor for arithmetic routines
PROD	0016	Multiplier/product/dividend/quotient for math routines
CHADR	001A	Character address list pointers
NXADR	002E	First character of first next line
PRTDIR	0030	Print direction flag, +=left, -=right, 0=dummy delay

TEXT PRINT MAIN PROGRAM (note that addresses are given for TXPTH only)

TXPRT	0A04	Program entry point
	0A05	Line length, 3C hex (60 decimal) is normal
MNLP	0A12	Beginning of main loop, find 10 lines
PRSCN1	0A3A	Set up hardware and turn on printer motor
NXTHLF	0A6F	Time 16 print cycles
	0AB1	Calculate phase times from measured printer speed
WNDOW	0B07	Determine current speed error
PHAS1	0B40	Start of left going half of print cycle
	0B7A	Start of right going half of print cycle
PRTEND	0EA8	Test print mode and then repeat or shutdown
THPRT	0BAB	Set up thermal elements and print for one dot time
LDELAY	0F27	Precision delay subroutine
UPDATE	0BF3	Set pointers for next column of dots in the 10 characters being printed
CHROW	0C46	Get specific dot row of character from font table
CHTB	0CCA	Font table for 96 ASCII characters, 7 bytes per char.
LDELAY	0F6A	Precision delay subroutine
WAPCYC	0F3D	Wait for printer start pulse subroutine
UNSMPY	0F8D	Unsigned 16 bit multiply
SRQA	0FB3	Quad shift PROD right with sign extend
SRQL	0FB6	Quad shift PROD right logical
SLQL	0FBF	Quad shift PROD left with zero shifted in
RLQL	0FC0	Quad shift PROD left with carry shifted in
UNSDIV	0FC9	Unsigned 32/16 divide
TPPHI	0FEC	High bytes of phase percentages
TPPLO	0FF1	Low bytes of phase percentages
TOVHHI	0FE6	High bytes if phase overhead times
TOVHLO	0FFB	Low bytes of phase overhead times