

ZGRAPH

TABLE OF CONTENTS

GENERAL	1
ZGRAPH	2
BINCONV	14
XxBINCAT	16
BINPLAY	17
DOSAVE	18
BINPRINT	19
ZGRAPH FILE FORMATS and EXAMPLES	20

Authored and copyrighted (C) 1982/1983 by Karl A. Hessinger.
ZGRAPH is published by **MISOSYS**, Alexandria, VA.

LDOS is a trademark of Logical Systems, Inc.
TRS-80 and TRSDOS are trademarks of Tandy Corp.

GENERAL

ZGRAPH is a powerful graphics editor package that gives you the tools to construct screen images using your computer's block graphics capabilities. These images may be saved to disk and converted into forms usable by BASIC and machine language programs. Besides the ZGRAPH editor, the package includes five utility programs you can use to create, display, and manage ZGRAPH screens. ZGRAPH does it all: rapidly, totally, and economically!

The ZGRAPH package is provided on a 35-track single density data diskette for LDOS Version 5.1. The PRO-ZGRAPH package is provided on a 40-track single density data diskette for LDOS/TRSDOS Version 6.

Z G R A P H

Z G R A P H

The ZGRAPH graphic utility package permits utilization of all of the TRS-80's capabilities in the creation of graphics screens. It will function with those computers supporting TRS-80 block graphics. ZGRAPH is invoked with:

ZGRAPH	Invoke the GRAPHIC editor
ZGRAPH *	Re-enter ZGRAPH and
<F><A><Y>	abort screen clearing

ZGRAPH Editor

The ZGRAPH/CMD file comprises the graphics editor that allows creation of graphic images. When "ZGRAPH" is typed from DOS Ready, the machine language program will load and take control. A graphic logo and a copyright message are displayed during initialization to inform you that ZGRAPH is loading. If you inadvertently exit from ZGRAPH without saving your screen images to disk, you may recover your images by re-entering via "ZGRAPH *" then issuing the commands, <F><A><Y> to recover the screens.

When ZGRAPH is ready for use, the screen will clear and a flashing graphic cursor will be displayed in the upper left corner of the screen. This is the primary cursor and indicates that ZGRAPH is in the graphics mode. This mode is also the command mode. ZGRAPH possesses two sets of commands, primary and secondary. Primary commands are available anytime the flashing graphic cursor is displayed. Secondary functions are invoked by first depressing <F> and then the appropriate function code. A 'help' list of commands at both levels is available by typing <H> for primary commands or <F><H> for secondary functions. The respective keystrokes will display the complete list of primary or secondary commands available. The <BREAK> key can be pressed to abort most commands.

Primary Command List

<C>ursor home	<H>elp	<R>everse
<D>raw mode	<I>nsert text	<S>et marker
<E>rase mode	<L>ocate marker	<X>-flip
<F>unction	<M>ove mode	<Y>-flip
	<P>osition	

Secondary Function List

<A>bort	<F>ill	<M>erge	<R>ectangle	<W>indow
lank	<G>et	<O>utput	<S>ave	<X>change
<C>ircle	<H>elp	<Q>uery	<T>ranslate	<Z>ero
<D>uplicate	<I>nput		<U>sage	<+> Magnify
<E>xit	<L>ine		<V>iew	<-> Reduce

Many ZGRAPH commands and functions (such as <H>elp) overlay the lower portion of the video display screen. Rest assured that your images are not affected. At the point of execution of the command or function, the lower portion of the screen will be restored. The screen may be restored after the <H>elp command by depressing <ENTER>.

The TRS-80 Graphics Screen

The video display screen of the Model I or Model III TRS-80 consists of 1024 bytes of memory arranged as 16 rows of 64 columns. A Model 4 TRS-80 compatible machine has a display screen of 1920 character cells arranged as 24 rows of 80 columns. Each memory location is capable of displaying one ASCII or special character or any combination of the six (2 wide by 3 high) graphic dots [Note: A Model 4 displays the two lower graphic blocks as 2 wide by 1 high]. These graphic 'dots' will be referred to as pixels (picture elements) in these instructions. When considering the screen as "M" rows of "n" columns of text, the rows are numbered <0 to m-1> starting with the top row and the columns are numbered <0 to n-1> from left to right. In the graphics mode, the pixels are numbered 0 to 2n-1 [127/1591, from left to right along the X-axis and 0 to 3m-1 [471711, from top to bottom. ZGRAPH allows any of the 160 (224 on the Model 111/4) possible characters (ASCII, graphic and special) to be displayed at any point on the screen.

PRIMARY COMMANDS

The following paragraphs describe the operation of the primary commands.

Cursor Movement

Cursor movement depends on the mode that ZGRAPH is in. In the graphics mode (when ZGRAPH is first entered), movement is achieved using the number keys 1-4 and 6-9 or the four ARROW keys. This allows convenient movement of the cursor using the numeric keypad as shown in Figure I (the regular number keys will also work). The keypad arrangement of:

Z G R A P H

Z G R A P H

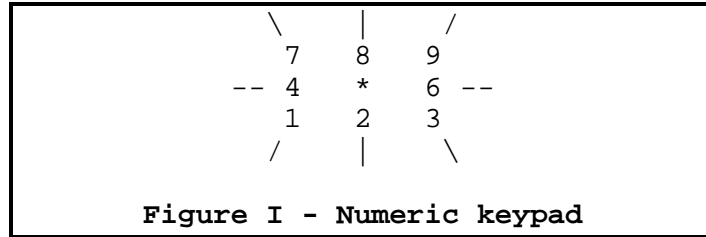


Figure I - Numeric keypad

is directly related to the movement of the cursor. Pressing <4> will move the cursor to the left and <8> will move it up. The corner keys <7>, <9>, <1> and <3> will move the cursor diagonally. All keys will auto-repeat after a short delay. The repeat rate may be set to fast or normal. On the Model I/III, the faster repeat is activated by simultaneously depressing the <SHIFT> key along with the movement key. Under DOS Version 6, the fast or normal rate is set by using the <F1> key to invoke the fast rate and the <F2> key for the normal rate. [On the Model I/III, the keys are also additive; depressing <6> and <9> simultaneously will move the cursor in a direction between right and diagonally to the upper right. Pressing <7>, <4> and <1> simultaneously will move the cursor to the left more rapidly than <4> alone, but pixels will be skipped.] The screen wraps around on all edges. If you go off the screen to the left, you will reappear on the right. The same is true of the top and bottom. The <C>ursor Home command will return the cursor to the upper left corner of the screen. Now that you know how to move the graphic cursor let's examine the three modes that the cursor may be in.

<D>raw Mode

In this mode, the cursor will leave a trail of bright graphic pixels everywhere it goes. [If you are using the additive feature of the cursor movement keys, the effect will be to produce a dotted line.]

<E>rase Mode

This mode is the reverse of <D>raw. Everywhere the cursor is moved, the graphic pixels will be turned off (dark). Be sure to cancel this mode by selecting <D>raw or <M>ove as soon as you no longer need it to avoid accidentally erasing pixels.

<M>ove Mode

This is a non-destructive means of moving the cursor. When in this mode, the cursor may be moved through existing graphic or text locations without disturbing the contents.

<S>et Marker

Certain functions such as line, rectangle and duplicate require two points of reference to accomplish their job. One reference point is always the current cursor location. The other point is established with the <S>et command. When <S> is depressed, an invisible marker is placed at the current cursor location. The cursor may then be moved to establish the second point of reference. Only one marker may be <S>et at any one time. If you <S>et a new point, the old one will be lost. To view the location of the marker use the <L>ocate or <P>osition commands.

<L>ocate Marker

This command will cause the <S>et marker to flash for a short interval. When the marker stops flashing, other commands or cursor movement can be executed.

<P>osition

This command will display the row,column and x,y pixel positions of the cursor and marker. It also displays the current mode. <ENTER> will terminate the display of the positions. Information similar to the following will be displayed when <P> is depressed.

-- Mode --		X - coor	Y - coor	Column	Row
=====		=====	=====	=====	===
Move	Cursor :	87	48	43	16
	Marker :	29	22	14	7

<I>nsert Text

This mode is distinguished by an underline cursor. The mode will remain active until <ENTER> is depressed. While in the text <I>nsert mode, cursor movement is via the arrow keys. The cursor is non-destructive of both graphics and text. Simply move the cursor to the desired position and start typing text. The cursor will advance after each letter is entered. The screen wrap-around in the text mode is identical to the graphics mode when the arrow keys are used; however, when text is entered, the line will be advanced if you go off the right edge. The cursor will not move past the bottom edge of the screen. Remember when you see the effect of entering text on the surrounding text, that each text character occupies the same space as six graphic pixels (2 x 3). Three special entries are available in the text mode:

<CLEAR><SPACE> - will enter a graphic blank instead of the normal ASCII blank entered with the space bar. This is important if you <R>everse.

Z G R A P H

<CLEAR><@> - will enter a full graphic block (chr\$(191); all pixels on).

<CLEAR><-> - on the Model III/4 only, will display a reverse video question mark. Two hexadecimal digits may then be typed (they will not be displayed) and will be interpreted as the character they represent. This allows access to the Model III special character set. For example, depress **<CLEAR><->** then **<E><F>** and the copyright symbol will be displayed. You will always get the special characters and not the alternate space compression characters.

<R>everse

This command will reverse the screen video. All bright graphic areas will become dark and vice versa. Text will not (and can not) be reversed. If you want spaces between text to remain dark when the screen is **<R>everse**d, use the space bar when entering text otherwise use **<CLEAR><SPACE>** as explained under **<I>** above.

<X>-Flip

This command will create a mirror image of the screen about the Y-axis. The graphics will be a true mirror image and the order of text characters will be reversed but, of course, the individual text characters cannot be reversed. A second **<X>-Flip** will restore the screen to its original configuration.

<Y>-Flip

The same as **<X>-Flip** as described above except about the X-axis.

SECONDARY FUNCTIONS

The secondary functions of ZGRAPH are obtained by depressing **<F>**. At the prompt,

Function?

the following functions are available (if you decide not to enter a function, depress **<ENTER>** to return to the command mode).

ZGRAPH Data Transfer Functions

ZGRAPH has a number of in-memory screen buffers in addition to the video display screen. The number of buffers varies with the amount of memory available. All but one of these buffers are general purpose buffers and are available to the user to store displays. This is useful when creating a large graphic consisting of several ZGRAPH images or in creating those images using the <M>erge function. ZGRAPH can also load and save images to disk files. All data moving to and from the disk passes through the primary video display. The last internal display buffer is best described as the error recovery or auxiliary buffer. When any function is executed that destroys an existing screen display (lank, <G>et, <I>nsert text, <L>oad, <M>erge, etc.) the current display is automatically saved to the auxiliary buffer prior to the function being executed. If you discover that you made an error (<M>erged the wrong display for example), you may recover the original display using the <A>bort command. Figure II below shows how data is moved within ZGRAPH.

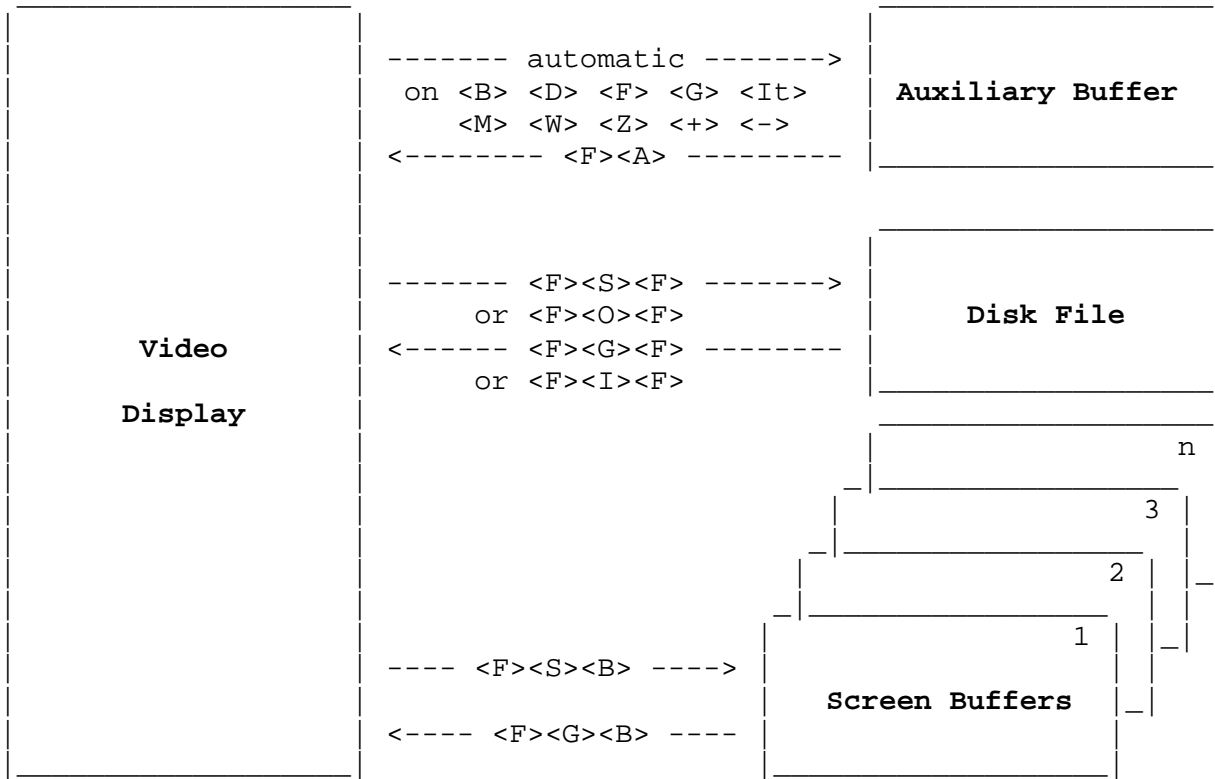


Figure II - Data movement within ZGRAPH

<G>et

Get is the function for loading the video display screen. Pressing <G>et will prompt the question,

Get from uffer or <F>ile?

If uffer is requested, you will be prompted to select a buffer with:

Buffer number (1-n)?

The current contents of the display buffer will be replaced by the contents of the buffer that you specify. If you request the <F>ile form of get, you will be prompted to enter a filespec via the message:

Filespec w/o ext for get >

Up to ten characters may be entered; eight for the name and two for the drivespec (i.e., testpict:2). An extension of "/BIN" (binary) is assumed for all ZGRAPH binary files. If a drivespec is not entered, all drives will be searched. If no filespec is entered, ZGRAPH will use the last filespec entered with either the <G>et or <S>ave commands. If you accidentally destroy a display by <G>etting another image, the original display may be recovered with <A>bort. The <G>et command may be cancelled prior to execution by depressing <BREAK>.

<I>nput

Reads a Multiple Binary File (/MBF type) from the disk and loads the screen images into the in-memory buffers. You will be prompted to enter the filespec as in the <G><F> subcommand functions except that the file extension used will be "/MBF". Note that any memory buffers currently in use but not contained in the /MBF file will be left unaltered. If the /MBF file contains any buffer number that exceeds the highest buffer currently available, that buffer image will not be loaded.

<O>utput

Writes all of the in-memory buffers that have data in them to a disk file. The file will be written in Multiple Binary File (/MBF) format which can be reloaded with the <I>nput subcommand function or used in the BINPLAY/CMD program.

<S>ave

This function will save the screen image to a disk file or a memory buffer. The disk save will use a rectangular area of the screen described by the cursor home position and the current cursor position. Saving to a buffer will use the entire screen. To save a screen image, first position the cursor to the lower right-hand corner of the area that you wish to save. When <S>ave is depressed at the "Function?" prompt, you will be queried:

Save to uffer or <F>ile?

Z G R A P H

The and <F> options are the same as under <G>et. The next prompts will indicate the size of the screen that ZGRAPH is set to save. If you forgot to position the cursor to the lower right corner of the area that you wish to save, answering 'N' to these prompts will return you to the command mode without saving the image. Like <G>et, if no filespec is entered, ZGRAPH will use the last filespec entered with either the <G>et or <S>ave commands. <S>aving does not affect the contents of the screen.

<M>erge

<M>erge allows you to superimpose one image over another. You will be prompted for:

Merge from uffer or <F>ile?

The responses and remaining queries are the same as <G>et. Unlike <G>et, however, the current display is not cleared but rather, the new display is superimposed on top of it. In the merge process, ZGRAPH uses the following rules to establish the merging of a "source" buffer/file to the video display:

1. Any source byte will replace a video graphic blank (X'80').
2. If both video and source bytes are graphic, the source graphic byte will be logically ORed (pixel by pixel) with the video byte.
3. If the video byte is graphic and the source byte is non-graphic, the video will retain its current pixel configuration.
4. If the video byte is non-graphic, the video will retain its non-graphic value.

<X>change

This function completes the data manipulation capabilities of ZGRAPH. <X>change will prompt with:

<X>change Screen with Buffer (1-n)?

Enter a buffer number from 1 to n. The <X>change function swaps the contents of the display with the specified buffer (the contents of the screen are placed into the buffer while the old contents of the buffer are placed on the screen). The auxiliary buffer retains a copy of the previous screen image.

<A>bort

The <A>bort command allows you to recover a screen display that was inadvertently destroyed. For instance, if you <G>et a new image without <S>aving the old one, you can recover with <A>bort. Depressing <A> will yield the query:

OK to load auxiliary buffer?

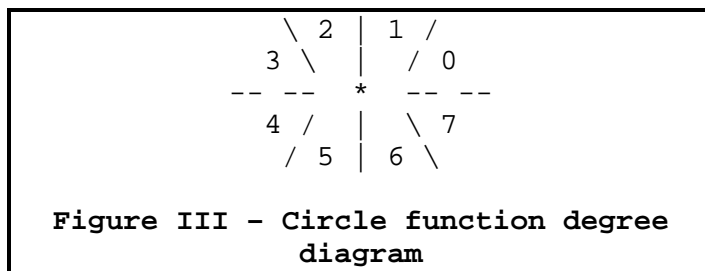
Answering <Y>es will cause the auxiliary buffer to load into the display. The reason for the question is that while <A>bort will allow you to recover from other catastrophic errors there is no recovery from <A>bort; that is, invoking <A>bort will cause a permanent loss of the current screen display. If you want to preserve the current screen, do a <S>ave just prior to issuing the <A>bort function.

ZGRAPH Graphic Generation Functions

The following functions are used to generate various images.

<C>ircle

The <C>ircle function will draw a circle or an arc around the current location of the cursor. You will be prompted for "Radius?" which is in units of pixels along the Y-axis. The next prompt is for "Starting arc (0-7)>". This value represents the number of the arc as illustrated in figure III. The last prompt is for "Ending arc (0-7)?". For instance, an arc from zero degrees to ninety degrees is 0-1. To draw a full circle, specify arc 0 to arc 7. A left half-moon would be arc 2-5. If ZGRAPH is in the <E>rase mode, the arc(s) will be reset rather than set.

**<D>uplicate**

The <D>uplicate command makes use of the marker created with the <S>et command. First <S>et the marker in the upper left corner of the section of the screen that you wish to duplicate. Move the cursor to the lower right hand corner and depress <D>. The block to be duplicated has now been defined. The graphic cursor is replaced with the underline cursor. Position this cursor (using the arrow keys the

Z G R A P H

same as the insert-text mode with <BREAK> to abort) at the upper left corner of the area where you wish to place the duplicate block and press <ENTER>. As much of the block will be duplicated to the bottom of the screen. If necessary, the duplicated block will wrap around the right edge of the screen. The image contained in the block may be repeatedly duplicated by depressing <D>.

<F>ill

This function can be used to change pixels enclosed within a boundary. In DRAW and MOVE modes, all pixels will be set until a boundary of set pixels is reached. In ERASE mode, all pixels will be reset until a boundary of reset pixels is reached. If no such boundary exists, the display screen edges will be considered to be the boundary.

<L>ine

<L>ine will establish the best fitting (straightest) line between the marker <S>et and the current cursor position. In ZGRAPH "move" or "draw" mode, the line will be constructed with set pixels. In "erase" mode, the line will be constructed with reset pixels. The marker position will be updated to the current cursor position after each <L>ine is drawn. The automatic re-setting of the marker will provide an easy way to construct lines connected end-to-end.

<R>ectangle

To create a rectangle of any size, first <S>et the marker at the upper left corner of the desired rectangle. Move the cursor to the lower right corner and depress <R>ectangle. The four sides of the rectangle will be constructed as "set" or "reset" pixels depending on the mode as in <L>ine.

<Z>ero

This function is used to fill the rectangle formed by the cursor and the marker with either all pixels ON or all pixels OFF. It is useful for clearing a large block of the screen (or "whiting" a large block).

<+>Magnify

This function can be used to enlarge a particular rectangular area of the screen display. It will magnify the graphics in the rectangle formed by the marker and the cursor. You will be prompted to enter a magnification ratio in the range <2-5>. For a magnification ratio of "2", each pixel is mapped to a 2x2 pixel group; a magnification ratio of "3" maps each pixel to a 3x3 pixel group; etc.

<->Reduce

This function can be used to shrink a particular rectangular area of the screen display. It will reduce the graphics in the rectangle formed by the marker and the cursor from a 2x2 pixel group to a 1x1 pixel group depending on the reduction mode. You will be prompted to enter the reduction mode <1,2>. Mode 1 will set a corresponding pixel if any one of the four pixels in the 2x2 group is set. Mode 2 requires any two of the four pixels to be set before setting the corresponding pixel in the reduction.

lank

lank gives you the options of clearing the display screen or any buffer. It will display the prompt:

Blank <S>creen, uffer or <A>ll ?

If blank <A>ll is selected, you will be asked, "OK to blank all?" as a double check of your intent. If <S>creen blank is selected, ZGRAPH clears the video display screen. The screen is saved to the auxiliary buffer and may be recovered if the blanking was inadvertent. Note that the blanking function fills the target screen/buffer with graphic blanks (X'80'). If you select the uffer blanking, you will be prompted to select a specific buffer.

<T>ranslate

This function will translate all occurrences of a character to another character. After typing <T>, respond with the decimal value to translate and the decimal value to result after the translation. The "find" and "replace" character values may also be entered as their single key entry in addition to their decimal ASCII value. For example, the letter "A" may be entered as either "65" or "A" (without the quotes). Note that <CLEAR-SPACE> and <CLEAR-@> (graphic block 191) are equally acceptable as "single key entries". You can rapidly change all text blanks to graphic blanks with this command.

<W>indow

The first thing you will notice when you depress <W>indow is that the cursor disappears. While in the window mode, the entire screen display will move in response to the arrow keys. Any part of the image moved off of the edges of the screen is lost. To terminate the window mode, press <ENTER>. This command is very useful to reposition an entire image on the screen.

Z G R A P H

<E>xit

This function provides a graceful exit from ZGRAPH to DOS Ready.

<U>sage

This function is used to obtain the status of the in-memory buffers. It displays a graphic block next to all buffers which have data in them. The graphic block "flag" is set by <X>change and <S>ave, and is reset by Iank.

<V>iew

This function allows the rapid display of the memory buffers contents to the video display screen. You will be permitted to specify the buffer range, and a relative speed at which the buffers will be displayed, <1-9>. <V>iew may be aborted by depressing <BREAK>.

<Q>uery

<Q>uery is used to question the operating system. On the Models I/III, it obtains directory information. <Q>uery will issue two prompts as follows:

Directory of which drive ?
Display /IN or /<M>BF files ?

The first prompt is used to specify the drive number for the directory information. The second is to specify whether you want to see the names of the binary or multiple-binary files. It will then list a directory of all files on the specified drive that possess the specified extension. The drive volume name and available space in "K" will also be displayed.

Under DOS Version 6, the <Q>uery command is used to access any DOS library command. <Q>uery will issue one prompt as follows:

Command?

You can then enter any library command such as "DIR /BIN:2". When the DOS command completes, you can return to the ZGRAPH screen by depressing the <ENTER> key.

BINCONV: ZGRAPH File Conversion Program

The post-processing program, BINCONV/CMD, has been provided to allow ZGRAPH created displays to be used in other applications. BINCONV will translate the ZGRAPH binary file format to other formats. It is invoked by entering the command:

BINCONV	Run conversion program
----------------	------------------------

ZGRAPH's standard file format is a pure binary representation of the screen display. Each line of the screen memory is <S>aved as the values of the memory bytes terminated by a carriage return (x'0D'). A screen saved with ZGRAPH would thus occupy a number of bytes equal to the number of rows multiplied by one greater than the number of columns.

The BINCONV program will display the following menu of choices:

- ```

* * ZGRAPH file conversion utility * *

<1> - ZGRAPH to Load Module
<2> - ZGRAPH to Packed BASIC
<3> - ZGRAPH to BASIC Data
<4> - ZGRAPH to EDAS
<5> - Disk directory
<6> - Exit to DOS

```

Depress the appropriate number key for the format you desire. The formats are as follows:

**ZGRAPH to Load Module**

The following prompts must be answered under this format mode:

```

Starting address { 15360/12288 } ?
[For 5.1.3; Transfer address { 73 } ?]
Filespec to convert (w/o ext) ?

```

For the Model I/III, the starting address defaults to the start of the video display memory and the transfer address defaults to the system vector, @KEY. The effect of choosing the default values (depressing <ENTER> in response to the prompts) is to create an executable /CMD file that will place your image on the screen and pause until a key is depressed, at which time it will return to DOS. Under DOS Version 6, the file is constructed as a core image which loads at the start of user RAM and returns to DOS Ready.

**ZGRAPH to Packed BASIC**

This format mode will prompt for the following:

```

 Array name { ZG } ?
 Starting index { 0 } ?
 Starting line number { 100 } ?
 Line number increment { 10 } ?
 Input filespec (w/o ext) ?
 Output filespec (w/o ext) ?

```

The default values will create a file with the extension of "/BAS". The file will consist of packed graphics strings with each line consisting of the string (ZG\$(#)="packed value of one line of your image") starting with an index (#) of 0, line number of 100 and line number increment of 10.

**ZGRAPH to BASIC Data**

The following prompts must be answered:

```

 Starting line number { 100 } ?
 Line number increment { 10 } ?
 Input filespec (w/o ext) ?
 Output filespec (w/o ext) ?

```

This option will create BASIC data statements starting with line 100 (if the default is used). Each statement will consist of 16 decimal numbers representing the sequential values of your screen image. The file will be saved with an extension of /BAS to allow merging into your BASIC program.

**ZGRAPH to EDAS**

Prompts similar to "ZGRAPH to BASIC Data" will appear. The file that will be created will have an extension of /ASM and will be in the EDAS editor/assembler format. Each line of the file will consist of a DEFB statement and 16 decimal values representing the values of the bytes of your image. This file may then be merged into an assembler program. Under LDOS Version 5, the file will be headered and line numbered. Under LDOS/TRSDOS 6, the header and line numbers will be omitted.

**Miscellaneous**

The fifth menu option gives you the capability of displaying a disk directory [under LDOS Version 5] or of executing a DOS command [under LDOS/TRSDOS Version 6]. This is identical to the <Q> function of ZGRAPH.

To exit BINCONV, press <6> from the main menu

**xxBINCAT/CMD**

These programs are used to print a single binary file or a concatenation of single files to a printer. The RSBINCAT program supports the Radio Shack DMP printers. The EPBINCAT program supports the Epson printers. xxBINCAT is invoked with the command:

|                                                                |                                                                                 |
|----------------------------------------------------------------|---------------------------------------------------------------------------------|
| <b>xxBINCAT (ADDLF,DENSE,RS2100)</b> Run the printing program. |                                                                                 |
| <b>ADDLF</b>                                                   | An EPBINCAT parameter used to force a line feed after a carriage return.        |
| <b>DENSE</b>                                                   | If entered, the entire graphic will be printed in boldface via overstrike.      |
| <b>RS2100</b>                                                  | A parameter to be entered if the DMP2100 printer is being used (RSBINCAT only). |
| Abbreviations: A=ADDLF, D=DENSE                                |                                                                                 |

The BINCAT program will print the graphics cells stored in one or more ZGRAPH binary (/BIN) files. If more than one input file is specified, BINCAT will concatenate the images so they are printed left to right. Therefore, it is possible to combine two or more screen images to make a larger "picture". For instance, a large picture made up of six screens in a three across by two high "picture" can be printed by first concatenating and printing the three top images then concatenating and printing the three lower images.

When BINCAT is first invoked, it will display the prompt:

**ZGRAPH filespec w/o ext ?**

Enter the file specification of a ZGRAPH binary file. BINCAT will continue to prompt for additional file specifications until you depress <ENTER> by itself to end the input. Each file identified will be concatenated for printing across the page. If you depress the <BREAK> key, BINCAT will exit to DOS Ready. Once all file(s) are entered, ZGRAPH will prompt with:

**Enter magnification (1-9) ?**

The magnification value will cause printing to use as many dots vertically and horizontally for printing a pixel as specified by the value. For example, a magnification of "3" will print each screen pixel in a 3-dot by 3-dot impression. When the printing is complete, BINCAT will give you the opportunity to print another copy of the concatenated image. It does this by returning to the "Enter magnification" prompt thus allowing you to specify any desired magnification. If you respond with <ENTER> by itself, BINCAT will return to the "Input filespec" prompt. If you enter a <BREAK>, BINCAT will exit to DOS Ready.



**BINPLAY/CMD**

This program will perform a video display sequence of all buffers saved in a ZGRAPH Multiple Binary File (/MBF). It is invoked via the command:

|                                                              |                                                                                                                                     |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>BINPLAY filespec (PAUSE,DELAY,REPEAT)</b> Invoke MBF play |                                                                                                                                     |
| <b>filespec</b>                                              | Specifies the file containing the screen images. The default extension is "/MBF".                                                   |
| <b>DELAY=val</b>                                             | "Val" is specified as the relative length of time a screen is displayed. The range of values acceptable is <0-255>. Default is 128. |
| <b>PAUSE=sw</b>                                              | If "sw" is specified as <ON>, then BINPLAY will wait for you to depress <ENTER> before displaying the next frame. Default is OFF.   |
| <b>REPEAT</b>                                                | Is specified to invoke a repeating play of all screens until <BREAK> is pressed.                                                    |
| Abbreviations: D=DELAY, P=PAUSE, R=REPEAT                    |                                                                                                                                     |

The BINPLAY program is useful for displaying a small series of graphic screens. It will display the sequence of screens saved from the ZGRAPH in-memory buffers that are in use at the time that the ZGRAPH <O>utput function is specified.

The DELAY parameter allows you to set the "viewing" time of each screen. The PAUSE parameter is used to suppress the automatic advance to each screen. This mode is useful when previewing a Multiple Binary File. If the screens are designed as a repeating "slide" show, then the REPEAT parameter will cause the play to automatically repeat the image sequence. It will play continuously until <BREAK> is pressed.

**DOSAVE Screen Saving Filter**

DOSAVE is a keyboard filter that is similar to the DOS screen print function. However, where the screen print directs an image of the screen to the printer, DOSAVE will direct the screen image to a disk file specified by the user. The screen saving filter, DOSAVE, is established by entering the command(s):

|                                |                      |
|--------------------------------|----------------------|
| For LDOS 5.1                   |                      |
| <b>FILTER *KI using DOSAVE</b> | Enable screen saver  |
| For LDOS 6.x                   |                      |
| <b>SET *DS to DOSAVE</b>       | Reside filter module |
| <b>FILTER *KI using *DS</b>    | Enable screen saver  |

Once established, depressing <CLEAR><SHIFT><S> will activate the filter. The prompt,

**Filespec?**

will appear on the screen. Enter the desired file specification (a default extension of "/BIN" will be applied). The contents of the screen will be saved in the standard ZGRAPH binary format. These screen files may be loaded into ZGRAPH for further operations.

# B I N P R I N T

## BINPRINT File Printing Program

This program provides the capability of printing a binary graphic file to a printer that supports compatible block graphics (i.e. MX-80 with GRAFTRAX or other MX printers with the ALTCHAR printer driver that is part of the GRASP package). Printing is invoked with the command:

**BINPRINT filespec (OFFSET,STRIP=value)**

**OFFSET**                   Used for non-GRAFTRAX MX-80 printers.

**STRIP=value**            Will convert any character above "value" to a blank (X'20')

Abbreviations: O=OFFSET, S=STRIP

Typing "BINPRINT filespec" from DOS Ready will cause the specified file to be sent to the printer. The default extension, /BIN, will be used if none is specified. Note that the DOS screen print function <CONTROL><\*> remains active in ZGRAPH and may also be used if this feature has been selected using the SYSTEM(GRAPHIC) command [and the screen print (JKL) option of KI/DVR for the Model I/III LDOS 5.1 user].

The parameter, OFFSET, is used to add the decimal value 32 to graphics codes in order to place the code value into the range proper for those printers supporting the TRS-80 graphics but at a CHAR+32 value.

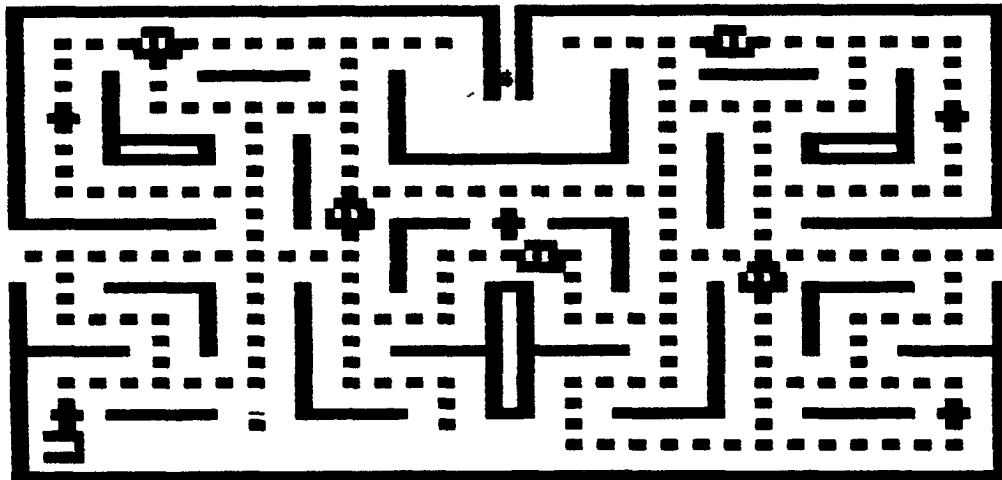
**ZGRAPH FILE FORMATS**

The "/BIN" binary format file is composed of each video row of characters terminated by a carriage return, <ENTER>. It will contain from one to n rows of data based on the location of the cursor when the file was saved.

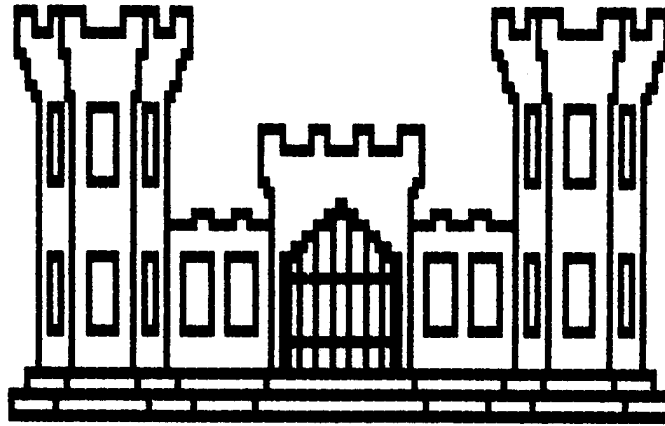
The "/MBF" multiple binary format file stores a number of memory-buffer images. It uses the first sector as a flag field to indicate which buffers are saved in the file. Relative byte 0 stores the length of the flag field <1-255>. Relative bytes 1-255 will contain an X'01' if the next image in the file stores the corresponding buffer. An X'00' indicates that the buffer is unused and no image exists for it in the /MBF file. Each image is contained in a 1K (Model I/III) or 2K (Version 6) block [i.e. 4 sectors or 8 sectors]. The first image corresponds to the first flag containing an X'01'; the second image corresponds to the second flag containing an X'01'; and so forth.

**EXAMPLES**

The following examples were generated using ZGRAPH. The printing was accomplished using both Epson MX-80 GRAFTRAX and MX-100 printers. The Graphic Support Package (GRASP) available from **MISOSYS was** used to implement pixel graphic printing on the MX-100 for both the 10-pica and 12-pica graphics. The /BIN files for these graphics are included on your ZGRAPH diskette.

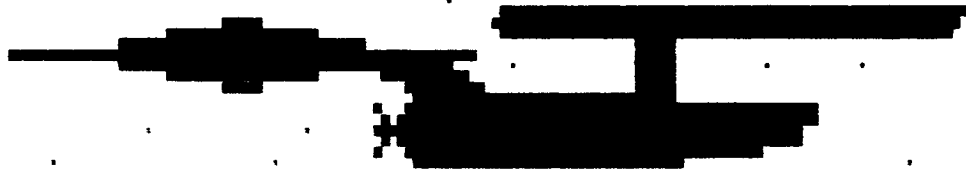


PACMAN: MX-100/ALTCHAR-STD12



CASTLE: MX-80 GRAFTRAX/Condensed-double strike

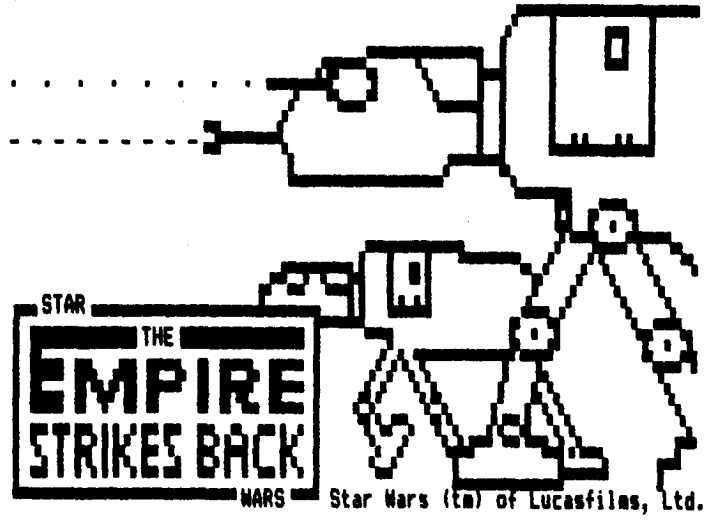
STAR TREK



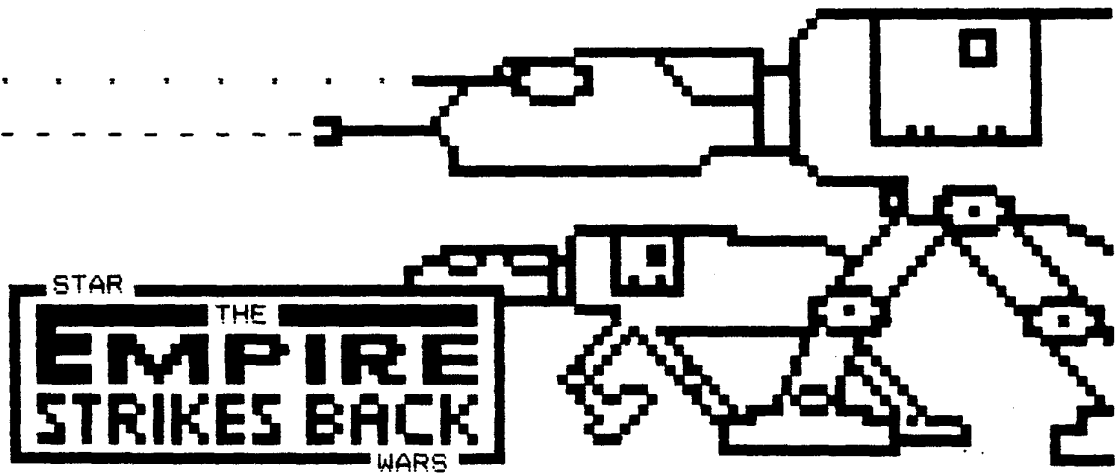
Our Mission..... To go where no man has ever gone before.....@

STARTREK: MX-100/ALTCHAR-STD12

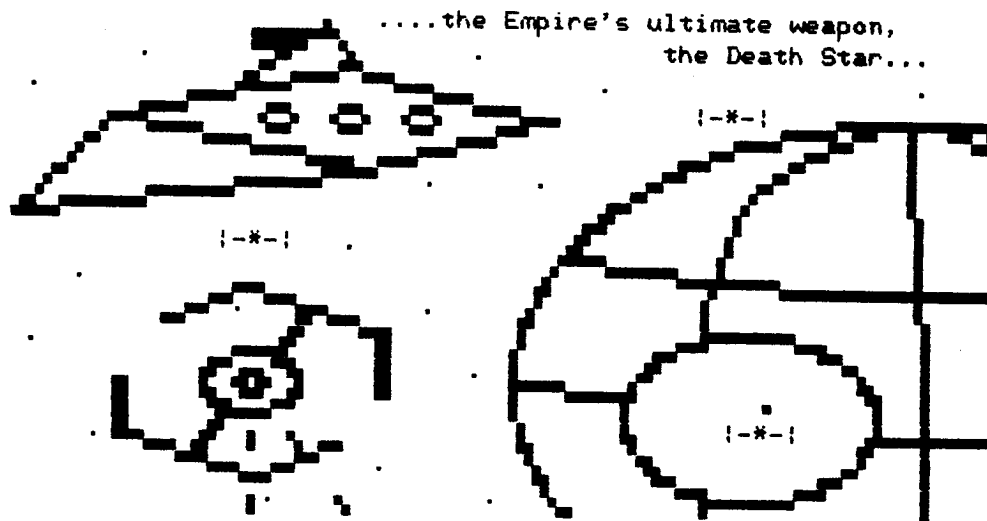
FILE FORMATS and EXAMPLES



EMPIRE: MX-80 GRAFTRAX/Condensed-double strike



EMPIRE: MX-100/ALTCHAR-STD10



DEATHSTR: MX-100/ALTCHAR-STD12

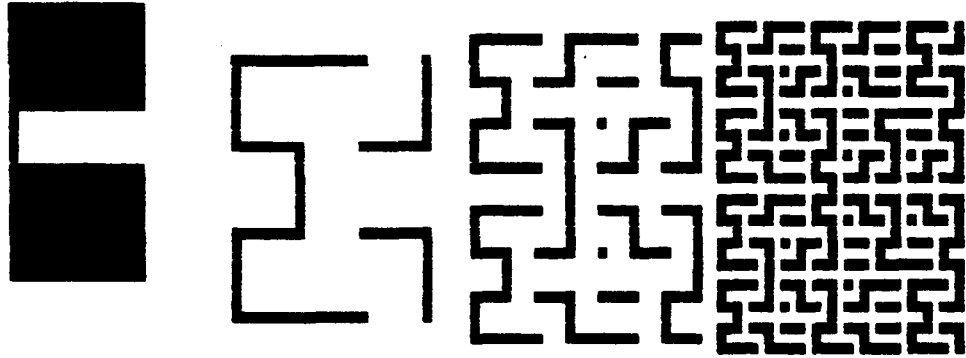
A long time ago in a galaxy far, far away....

**STAR  
WARS**

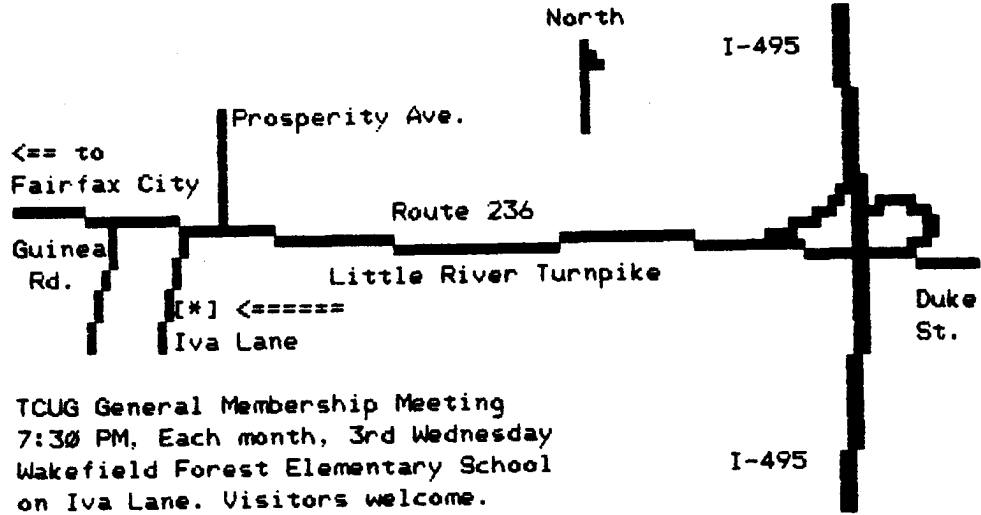
Star Wars (tm) of Lucasfilms, Ltd.

STARWARS: MX-100/ALTCHAR-STD12

HILBERT CURVES



HILBERT: MX-100/ALTCHAR-ST012



TCUGMAP: MX-100/ALTCHAR-STD12