

POVERSOFT
Products from Breeze/QSD, Inc.



POWERDOT

LOADING INSTRUCTIONS

The distribution disk you have received does not have an operating system on it. It cannot be backed up, but it has a special file transfer system which can be used to move the program files onto a diskette of your choice. Once you have moved the files onto your diskette, that diskette then becomes your master disk. Make several backups of the master disk, then put it AND the original distribution disk away in a safe place against that inevitable day when all your working disks are bombed.

To use the file transfer system, follow these steps:

1. Model I users: Format a single density diskette using TRSDOS.
Model III users: Format a double-density diskette using TRSDOS. If you have more than one drive, place this diskette in drive 1.

If you have only one drive, create a TRSDOS system diskette with as much free space as you can. Kill off all unnecessary files.

2. Boot the distribution diskette. You will then be presented with a list of the files on the diskette, and the transfer system will ask you for the destination drive. Enter the drive number where your formatted diskette is. Press <ENTER>.
3. The file transfer system will move the files onto your diskette one by one. If you have only one drive, you must enter 0 for the destination drive, and you will be told when to swap diskettes.
4. After all the files have been transferred, take the distribution disk out of drive 0. Boot up a system diskette. Make several backups of the disk which now contains the new files.
5. Put the distribution disk and one copy of your file master away in a safe place. Don't touch them again unless you absolutely have to.

If you are using the LDOS operating system (RS catalog # 26-2213 or 26-2214): This diskette can be treated as an ordinary data diskette by the LDOS system. You can simply use the LDOS COPY library command to move the files onto your own diskette. However, the distribution diskette cannot be backed up using mirror-image backup procedures.

If you are using LDOS and wish to use the file transfer system, you must format the destination disk using TRSDOS. The

file transfer system only recognizes TRSDOS formats on the destination disks. If you are using a Model I, do not create a double-density TRSDOS 2.7DD disk for your destination disk! Use single-density TRSDOS 2.3 ONLY.

NOTICE

This software is sold on an AS-IS basis only, and Breeze/QSD Inc. shall not be liable for any loss or damage, whether real or alleged, arising from the use of this software. No warranties of merchantability or fitness for a particular purpose are made, expressed, implied, or imagined. Installation, and determination of suitability for a particular purpose are the sole responsibility of the user.

PowerDOT: An Overview

PowerDOT is a modular program subtitled Part1, Part2, etc. Parts 1 and 2 are included in this package and are the Dot Editor (PDOT1/CMD) and the Dot Addressor (PDOT2/CMD, the PRINT Module). Collectively they allow you to create, review, edit, and print full page text, TRS-80 block graphics, and Dot graphics on a variety of printers, including the EPSON MX-series and the C. Itoh Prowriter 8510A.

Other uses than those just mentioned have been taken into consideration. A special disk routine has been added to allow loading of word processor, BASIC language, and command files. With a little special planning PowerDOT can be used to add graphics to these files and copy them back to disk.

PowerDOT can be broken into three major routines.

1. DISK I/O: All disk operations with the exception of the special ASCII I/O routine are done in "blocks". A block is defined as the smallest number of bytes that can be saved or loaded to/from disk files. One block will be equal to sixteen lines of text times whatever width is selected. PowerDOT allows you to go beyond the limitations of your screen. You are no longer limited to 64 character line lengths. Your screen is sixteen lines tall so you can consider a block to be a screen in depth.

If you set the screen width to 100, then one block would be equal to 1600 bytes (16 lines * width, which is 100 in this case). If you set the width to 200 then one block would be equal to 3200 bytes. If you set the width to 64 (the minimum allowed) one block would equal 1024 bytes. When creating a new graphic, the file to hold it must be created, and the width selected first. You should read the section of this manual entitled DISK OPERATIONS for further information. Information on how wide to set your graphic is found in the section entitled "Part2: The Dot Addressor."

2. KEYBOARD EDITOR: PowerDOT utilizes a "live" keyboard. The four arrow keys move the screen up, down, left and right, while the numeric keypad is used for pixel drawing. In the character mode the cursor is controlled by the four arrow keys only. All characters are supported, even Mod III special characters and printer command characters.

A number of special keys have been incorporated to assist you in creating your graphics. See the section entitled KEYBOARD INPUT. Other than the character mode all special keys are only in

effect while the key is being pressed. When the CLEAR key, for example, is held down at the same time as the numeric keypad is being used, erase mode is enabled. When you let go of the CLEAR key you automatically return to the draw mode. This style of keyboard use may take a little while to become accustomed to, but it was designed this way to minimize switching in and out of different modes and remembering which mode you are in at a given time.

There are also insert and delete functions to help mostly in centering (left & right) your graphics by enabling you to insert spaces as needed.

3. PRINTER DRIVER: PowerDOT's printer driver is a separate command file. See the section entitled "Part2: The Dot Addressor" for full details. Text and graphics are printed directly from disk files only, and the files must have been created by one of PowerDOT's other modules, in this case "PDOT1". PowerDOT utilizes its own printer driver and is not compatible with any other printer drivers, so make sure no other printer drivers are loaded when you are running PowerDOT.

NOTE TO LDOS USERS: You should clear out all unnecessary routines, drivers, etc. from high memory before running PowerDOT. This includes KI/DVR and PR/FLT, which will severely interfere with PowerDOT's functioning.

Text, Block, and Dot graphics are supported. Block graphics may be mixed with text. However, if printed in dot mode the text will be ignored and only the graphics will be printed. Block graphics are created using the dot addressable capabilities of your printer so even if your printer doesn't know how to print block graphics (e.g., GRAFTRAX+), PowerDOT will teach it.

There are a variety of print options available and it is recommended that you read about PDOT2 before attempting any printing. There are two example files included with this program called "SUPERDOT" and "SCREEN". SUPERDOT is the original (first) graphic created on the prototype version of PowerDOT. It is a normal-res Dot graphic and can be printed in styles <3> through <6>. Screen is the same graphic that is displayed when you execute PowerDOT and it can be printed in all six print styles.

Before you create a graphic, you should have an idea of what mode you intend to print it in. The same graphic will look totally different when printed under each of the different print options. The high-res mode will look the closest to what you see on the screen. However, Normal-res and Ultra-res will probably be the options used most often. (You'll find out after some experience.)

**** PRE-FLIGHT CHECK LIST ****

1) PowerDOT Parts 1 & 2 ready for execution on your DOS? Follow the instructions supplied to transfer the PowerDOT /CMD files "PDOT1/CMD" & "PDOT2/CMD" and graphic files "SUPERDOT" & "SCREEN" onto your destination disk. LDOS users, make sure all unnecessary routines have been cleared out of high memory.

2) Lots of data space available? Single drive owners should prepare a minimum configuration disk with around 60k of free space available. PowerDOT is capable of cramming more than one million five hundred thousand user programmable dots on a single sixty-four line by eighty column page. To store that much data on a disk file would require some two hundred and forty kilobytes of free space. DON'T panic! That's possible, but not required. The average PowerDOT file will be around 60K, but again, it depends on width and how many blocks you use.

3) Printer armed and ready? Your printer should be ready to go utilizing "normal" switch settings (if any) and the proper manufacturers cable. PowerDOT will, whenever possible, work around any routines that would require changing "normal" switch settings.

KEYBOARD INPUT

The following is a complete description of the keys utilized by PowerDOT. Some keys perform different functions depending whether you are in the character/edit mode or the normal draw mode. The character/edit mode will be discussed afterwards and the following descriptions assume that you are in the normal draw mode.

NORMAL DRAW MODE FUNCTIONS

NUMERIC KEYPAD: The numeric keypad is used to draw with a blinking pixel for a cursor. When looking at the numeric keypad, the direction the pixel will draw is represented by the position of the keys. The <5> key in the center of the keypad is the reference point. In other words, If you press the <9> key, the pixel will draw up and to the right. Notice that the <9> key is up and to the right of the <5> key. Keys <4>, <6>, <8>, and <2> draw horizontal and vertical and the <1>, <3>, <7>, and <9> keys draw diagonally.

- < 1 > Draws a diagonal line down and to the left.
- < 2 > Draws a vertical line down.
- < 3 > Draws a diagonal line down and to the right.
- < 4 > Draws a horizontal line left.
- < 6 > Draws a horizontal line right.
- < 7 > Draws a diagonal line up and to the left.
- < 8 > Draws a vertical line straight up.
- < 9 > Draws a diagonal line up and to the right.

KEY MODIFIERS:

These same key functions are changed by pressing the following keys at the same time.

< CLEAR > = Erase Mode

When held while drawing will enable ERASE mode. Pixels will be reset instead of set. You can use this function to erase a line or to move across a blank area of the screen (a blank line is fully reset anyway). This is the fastest way to position the pixel. Any characters passed over by the pixel will be erased.

< M > = Set Angle #1

When held while drawing or erasing will set a different angle for the <1>, <3>, <7>, and <9> keys. Horizontal and vertical lines will be dotted (every other pixel set).

< N > = Set Angle #2

When held down while drawing or erasing will set still another angle to the <1>, <3>, <7>, and <9> keys. Horizontal and vertical lines will be dotted.

< N > and < M > together = Set Angle #3

When both <N> and <M> are pressed while drawing or erasing will cause the <1>, <3>, <7>, and <9> to remain at their normal angle, but the lines they draw will be dotted (every other pixel off).

< P > = The PIXEL Mode

While pressed, the "P" puts you in the PIXEL mode. When the number keys are used to draw while in this mode, only the pixel will move. No line will be drawn. In the upper left hand corner of the screen there will be three sets of numbers displayed. They are the pixel's (X,Y) co-ordinates (0-47 & 0-127) on the screen

The third number is the distance the pixel moved during "auto repeat".

This routine has a number of uses. It can be used to move the cursor without changing the pixels that are already lit (NOTE: text characters will be erased). It can also be used to measure, locate and match up drawings such as letters, etc. When you let go of the <P> key you will return to the normal draw mode. The present pixel location when you press the key will either be lit or dark.

This key toggles whether or not the present pixel is left on. If you are on a blank portion of the screen and after you hold the <P> key the pixel blinks off and then stays on, you can release it and press it again and this time it will blink on and then stay off. Whatever state the pixel stays will switch everytime you press the <P> key. To relocate the pixel without destroying the characters on the screen use the CHARACTER/EDIT mode (see below).

The four arrow keys scroll the screen up, down, left, and right. If the outer limits of the "page" are met, these keys will have no effect. You should be sure to let go of all number keys before pressing any of the arrow keys. The graphics you have drawn on the screen are on the screen only. The screen is saved to the memory buffer before the screen moves but "slurring" the keys (pressing certain combinations of keys pressed at the same time) can fool the program and what you had on the screen will be lost. The present pixel location will have to be set or reset before the screen moves and the rule of what will happen is this. If the pixel location, when an arrow key is pressed, is different from the last time an arrow was pressed then the pixel will be reset (turned off). In simpler terms, if you move the pixel the next time you move the screen the pixel will go off before the screen moves.

If however, you scroll the screen, stop, then scroll the screen again, as long as you didn't move the pixel it will not be reset nor will it affect the drawing whatsoever. If you did move the cursor and moved it back to the very same location, the above will happen just as if you didn't move the pixel at all.

< LEFT ARROW > Scrolls toward the left side of the page.

< RIGHT ARROW > Scrolls toward the right side of the page.

< DOWN ARROW > Scrolls towards the bottom of the page.

< UP ARROW > Scrolls towards the top of the page.

< B > = Display current BLOCK

This key will display the current block number and offset from the left of the page to the center of the screen of the current

location of memory the screen is displaying. If held at the same time as the arrow keys are being pressed, it will interrupt the screen scroll as long as it is held. When released the screen scroll will continue.

***** BLOCK 00001 ***** OFFSET 00032 *****

The above line indicates that your present position is the upper left most portion of the graphic page. Block 1 is the very first block (Remember 1 block = 16 lines * width?) available. Offset 32 is half the distance across the screen. The reason for displaying the distance to the center of the screen from the left of the page instead of to the left of the screen is that most of the time you will be wanting to know when you are in the center of the page. If the width is set to 100 then when offset equals 50 you are in the center. When you are all the way to the right of the page the offset will equal the width-32.

The block display line will be displayed in a position equal to the top line of the block displayed. If you ever feel lost, just press .

< S > = Save location of Pixel

Pressing this key will save the present location of the pixel. It is used as a starting point for the AutoDraw(tm) routine.

< D > = DRAW

Pressing this key will activate the AutoDraw(tm) line draw routine.

The AutoDraw(tm) Line Draw Routine

The line draw routine makes use of the <S> key and the <D> keys. Pressing the <S> (source or start) key marks the starting point of the line that will be drawn. It will stay in effect until the <S> key is pressed again in a different location. The new location then becomes the starting point. When the <D> key is pressed a line will be drawn as close as possible from the starting point to the destination point. All angles are supported and the line is drawn as closely as possible to a straight line.

Lines drawn may span screens. In other words, a line can be drawn from the first block clear through to the last block in memory. If the <E> key is held after the creation of the line with the <D> (destination draw) key, the last line drawn with AutoDraw will be automatically erased regardless of the present pixel cursor location.

Care should be observed when using this routine as the line will be drawn immediately from the source point to the destination point without any notice as soon as the <D> key is pressed.

The main purpose of this routine is to draw long lines at angles which would be difficult to accomplish because of the inability to view the entire drawing area on the screen. Please note that some angles (those closest to 0, 45, 90, 135, 180, 225, 270, 315 and 360 degrees) may produce a slight error in precision.

CHARACTER EDIT MODE

<@> = Enable Character Edit Mode

This key activates the CHARACTER/EDIT mode. When this mode is entered the arrow keys will no longer scroll the screen. The numeric keypad will no longer draw with the pixel. The pixel will be replaced with a cursor. The arrow keys will now move the cursor. All other keys will now be live and the corresponding character will be printed if pressed. Pressing <ENTER> will exit back to the normal draw mode. The pixel location will be where the cursor was when you exited. You can switch into this mode to move to a new pixel location without destroying other pixel lines or characters.

The following keys have special functions while in this mode.

< LEFT ARROW > Moves cursor to left.

< RIGHT ARROW > Moves cursor to the right.

< UP ARROW> Moves cursor up.

< DOWN ARROW > Moves cursor down.

<(SHIFT) LEFT ARROW >

Deletes one space at the cursor. The entire memory buffer is affected. All succeeding characters are moved up one place toward the cursor.

<(SHIFT) RIGHT ARROW >

Inserts one space at cursor. Actually, the character under the cursor is not converted to a space. Instead, the entire memory buffer is affected. succeeding bytes are moved up one place to make room for the new character.

The main use for the insert/delete routine is to move portions of the graphic left or right on the page. If however, you hold the keys long enough you will eventually delete or insert an entire line.

To relocate a graphic up or down on the graphic page see the section on disk I/O.

<(SHIFT) @ > Character INSERT Mode

Pressing these keys will enter the "character insert mode". After entering this mode you may type in any three digit ASCII code. The character represented by that code will then be displayed at the cursor location. For example, to put a carriage return in your text, you would press <(SHIFT) @> followed by three numbers. A carriage return is 13 so you would type 013. Notice the leading 0. You may type any number from 0 to 255. This is a good way to insert printer commands or special characters in BASIC programs. With BASIC programs you will have to use a Chr\$(21) with some special characters. See your BASIC language manual. The following keys are usable under this mode:

< CLEAR > Clear Screen

In this mode the <CLEAR> key will clear the screen. Be Careful!

< ENTER > Exit back to normal draw mode.**< BREAK > Display the COMMAND MENU**

The command menu is displayed when you first execute the program. It can be redisplayed by pressing <BREAK> except in the CHARACTER/EDIT mode and during disk operations.

DISK OPERATIONS

PowerDOT disk I/O is controlled by the command menu. These are the commands available:

<0> = Return to DOS Ready

This key will exit the program and return you to DOS Ready. You may exit this program and then re-execute it without destroying the information in the memory buffer. You may have to re-define memory, however.

<1> = The Dot Editor

This key will enter the main body of the program. All drawing, editing, etc. are done in this mode.

<2> = Initialize file

This should be the first thing you do when you execute PowerDOT. When selected you will be asked for the filespec of this graphic. If your file is found it will be loaded into memory automatically. If the file does not fill the entire memory buffer the remainder of the buffer will be cleared. The width of the graphic will automatically be set and you will be returned to the command menu.

If the file specified is not found, you will be notified and will have the opportunity to abort the routine by pressing <BREAK>. If you press <ENTER>, a new file will be created using the filespec given. If no extension was given "/DOT" will be added to the filespec.

You will then be asked for the width of the graphic and then for a message. You may enter any message up to 200 characters to further identify the file. The message will be displayed whenever that file is loaded; however it does not become part of the graphic proper. The file will then be created and the memory buffer will be cleared. You will then be returned to the command menu.

<3> = Extend file

This is the fastest way to obtain more space for your graphic if you run out of memory space. When selected, the lowest numbered block in the memory buffer will be saved to the disk file. Then the rest of the blocks will be relocated downward to provide room for a new block at the high end. The new area is cleared and you will return to the command menu.

It should be understood that the entire memory buffer is not saved to disk, only the lowest single block. If you are only interested in saving one block to disk then pressing this key would do the job also.

<4> = Save/Load Memory Block

This routine is for saving or loading the entire memory buffer or just part of it. After you select save or load memory you will have the opportunity to alter the routine. You may enter a different filespec to save to a different disk file. Note that under this routine, no default extension is supplied. You can also lower the total number of blocks that will be loaded or saved. Regardless of how many blocks are loaded or saved the lowest numbered block is still the first block in the memory buffer.

You may also change the starting block number. This refers to the block number of the disk file, not the memory buffer. For instance, if you had a graphic in block one, and then saved onto

disk one block with "1" as the starting block number and then changed the starting block number to "2" and saved one block again. Your graphic located in block one of the memory buffer will have been copied to block one and two of the disk file.

<5> = Save/Load Screen

This routine is used to save what is on the screen to a separate disk file and add it back to another screen location. the extension "/SRN" will be added to the file if no extension is supplied. It should be understood that the file is added back to you graphic, not loaded back. In other words only the actual space that contained graphics or alphanumeric characters will be returned to the new location. The rest of the area will remain unchanged.

For example, if you had a picture of a transistor in a screen file, you could load it back to any part of the graphic (for instance, a schematic diagram) without destroying the graphic. Only the transistor image will be added back.

<6> = ASCII I/O

This routine will load and save any unprotected file into the memory buffer. There is further information about this routine located at the back of this manual under the title SPECIAL TRICKS.

<7> = Define Memory

This routine will allow you to reset the width of the graphic. It is the same routine that is used when you initialize a new file, only there are no disk operations involved. There are further suggestions on the use of this routine in the section on "Special Tricks".

<BREAK>

The break key will exit to the command menu during all submenus and prompts. If you break out of a filespec prompt you will have to re-enter the filespec under option <4> before using any disk I/O routines other than <3> = Extend File.

For PDOT2/CMD
EPSON users, use "EPDOT/CMD"
PROWRITER users, use
"PRODOT/CMD"

PowerDOT User's Manual

13

**** Part2 The Dot Addressor ****

PowerDOT Part 2 (PDOT2/CMD) will print your graphic on your printer. Part2 prints directly from disk files. The file must have been created by the "Initialize File" routine of Part1. Your printer manual will have information on how many dots are possible per line of text. The width of the dot graphic printed may not exceed one half of the total number of dots possible per line. A 9x9 dot matrix printer such as the EPSON MX-80 will have a maximum number of 960 dots per line. That means that the maximum width graphic that can be printed is 480. When printing character & block graphics the maximum width must not exceed the column width of the printer. For the MX-80 that would be 80. For the MX-100 that would be 132.

After you select the print mode of your choice you will be prompted for the filespec of the graphic. You may omit the extension if you wish. You will then be able to select one of three centering modes: center, left, or right. Your graphic will then be printed directly from the disk file to your printer. The disk containing the file to be printed must remain in the drive until the graphic is printed. When the entire file has been printed the "End of file" message will be displayed. At this time you may return to the menu and continue with another filespec or do a form feed. If you do not use PowerDOT to formfeed the paper at least once the printer will be left in a non-standard line feed mode. You will have to turn it off and then back to restore normal settings. Using this program's line feed command at least once will return the linefeed spacing of your printer to normal.

If Your printer has static switch settings they should be set to the most flexible condition (Probably the setting they are in now) with automatic linefeed on carriage returns. You should also be using the proper manufacturers cable. We are not talking about the computer's manufacturer but the printer's manufacturer. For instance, If you are using an MX-80 with a Radio Shack cable, this program will not operate correctly. It WILL if it is either an EPSON cable, an EPSON "type" cable, or a modified for EPSON Radio Shack cable.

Finally if you somehow reserve too much high memory and this program does not have enough memory to create the print buffers it needs it will notify you. You should have just enough memory to print the widest possible graphic with a 32k machine if you don't you will need to narrow the graphic or install the additional memory chips to update to a full 48k. You need the extra memory anyway!

The following is a complete description of each print mode.

<1> = Print Block Graphics

This routine will manufacture its own block graphic set so that even if your printer doesn't have a block graphics set it can still print them. Any characters that are mixed with the graphics will also be printed. Printer commands may be used, but your total line length in the style of type you have selected must not cause a wrap around by your printer. For instance, if you use the expanded print mode, you must use the character insert mode to place a carriage return in the line so that it will terminate before the line wraps around.

Printer commands will not affect the graphics printed. You should also avoid any double strike printing or distortion of the block graphics will occur.

<2> = Print Block Graphics (Bold)

This routine is the same as the one above except both the characters and the block graphics will be double struck twice, therefore filling in the space between all dots with additional dots. You may not want to use this mode with a brand new ribbon, but you would with the average "worn out" one.

<3> = Print Dot Graphics

This routine will print your graphic in dot mode. Each block graphic that is in the file will be converted to one dot on the paper. All characters that are in the file will be ignored. The dot spacing will be one dot per dot space in all directions. The maximum width that can be printed will be 1/4th of the maximum allowed by your printer. For the MX-80 this would be 240.

<4> = Print Dot Graphics (Bold)

This routine is the same as the one above except that four dots will be printed for every one block graphic in the file. The additional three dots will be placed one half space to the left and below the first, effectively filling in the spaces between dots. The outcome will be little squares instead of tiny dots. You may not want to use this routine with a brand new ribbon but you certainly would with an old one.

<5> = Print Dot Graphics (High-res)

This routine is the same as the one above except that only one extra dot will be supplied by the program. It will be placed one half dot width down from the one you programmed. Dot spacing across the width of the page is still one half dot so you may print a graphic that is twice as wide in the same amount of space. The maximum width of the graphic may be equal to 1/2 of the total number of dots allowed by your printer. For the MX-80 that would be 480.

<6> = Print dot graphics (Ultra-res)

This routine produces the highest possible resolution. Dot spacing in all directions is one-half dot. That means that you can program each dot every half dot space. Creating a full page

graphic to run on this routine would require multiple files or a hard drive system. Or you can use it to produce a miniature graphic.

NOTICE: PowerDOT will push your printer to the maximum resolution it is capable of. At times your printer will be required to perform one dot line feeds and multiple line feeds accurate to one third dot space. We have noticed that some printers (not related to any specific model) will occasionally misfire on the position of a dot or a line of dots. This is not the fault of the program and may not (probably won't) occur on your printer. If it does, experimentation will tell you how to correct for it. If we knew exactly what your printer would do under all circumstances we would have corrected for it. We have already corrected for bugs that are common to all of a specific model of printer. To minimize any undetectable errors, you should adjust your tractor feeds as accurately as possible and make sure your paper can be fed freely and straight. We also recommend that you leave the top (removable or hinged) cover open while printing intricate graphics. We have read nothing about the heat that can be created by the print head during long printing sessions in the manuals of the printers supported, but, it would seem to us that the cooler you keep the print head the longer it will last. We have done MANY printouts on our MX-80's and Prowriters with no problems or noticeable wear at all.

Finally, if you have questions concerning the way your printer should be set up to print dot graphics, please contact the store you purchased the printer from or the manufacturer. We cannot and will not accept inquiries concerning your printer's operation. If you feel that only we can answer your question please inquire by mail with a self addressed, stamped envelope included. It's not that we don't want to help as much as we can, rather most problems have nothing to do with the program and tie up phone lines and people that need to accomplish other things.

Send your letters and questions concerning PowerDot to:

PowerSOFT
11500 Stemmons Expressway, Suite 125
Dallas, Tx 75229
attn: PowerDOT Support

****** SPECIAL TRICKS ******

Shading in areas: Shading can be done probably a lot easier than you think. A shaded area could be created by filling in an area with alternating dots and spaces. The way to shade is simply to draw a series of diagonal lines in the area to be shaded. As long as you don't press any of the horizontal or vertical direction keys on the keypad you will never have two dots side by side. Try this and you'll see how easy it is. The effect will be alternating dots and spaces throughout the shaded area.

Clearing memory: The best way to facilitate clearing memory is to attempt to load a file with the initialize file routine that doesn't exist. Initializing a file only writes one sector. If there are no additional records to the file the memory buffer the rest of memory will be cleared when it is read in. I personally have a file named "NONO" (don't ask me why I chose that name) that I use to clear memory with. The file consists of record 0 only. It also sets the width for me.

Where to start drawing: This is a valid question, and is simply answered. You should start with the main part of the program located as near as possible to the correct portion of the page. If you need help deciding where that should be I would suggest that you create a file similar to a piece of graph paper. This can be done simply by using the line draw routine. Your grid should basically be the size of the screen per square. Then print the file in the mode you wish. You can then pencil in your idea on the paper and use it as a reference. Drawing with this program gets easier and easier as you use it and become accustomed to its features.

You can also leave notes throughout the dot graphic. In the dot mode of printing all non-graphics are ignored. So you don't have to worry about erasing them later.

Adding graphics to BASIC programs: Although programs like PowerDRAW are better suited to this kind of thing, You can do it with PowerDOT. First create a BASIC program with a series (15 per screen) of print statements or A\$= type statements. An example would be:

```
10 PRINT "  -sixty four spaces- ";
11 PRINT "  -sixty four spaces- ";
16 PRINT "  -sixty THREE spaces-";
```

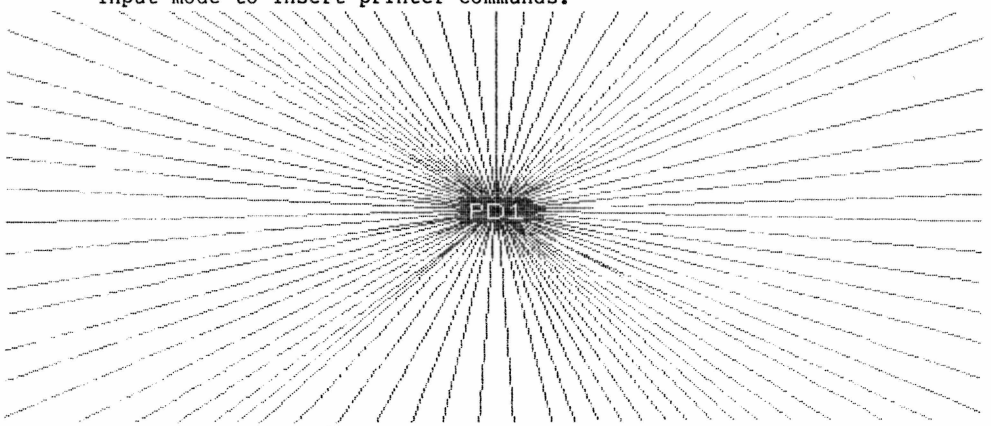
Notice that the last line had only 63 spaces inside the quotes. That is so the final screen position is left black to prevent the screen from scrolling. You can do the same thing with A\$=" - sixty four spaces -". Notice also that the line numbers

take up the same amount of space. Each is two digits wide. Now save the program to disk. Exit BASIC and execute PDOT1. Use the clear memory file technique I discussed above to clear the memory buffer. Use the ASCII I/O routine to load the BASIC file you just saved into the buffer. Press <1> from the menu to view the program. Don't worry about the stuff that doesn't make sense. We're interested in the spaces between the quotes. Use the define memory routine to set the width to 256. Now go back to viewing the BASIC program. Move the arrow keys to line up the first line of the program so that you are centered between the first and second quotes. When you have centered the top line, reset the width to 71. Return to see the BASIC program. the quotes should now be all lined up. If not, experiment with the width until they are. Now you can use PowerDOT to draw in the screen the way you want it.

After you are done save it back to disk using the ASCII I/O routine. Don't forget to add the extension "/BAS" to the filespec. You might also want to save it under a different filespec in order to keep the first program available to create additional screens for other BASIC programs.

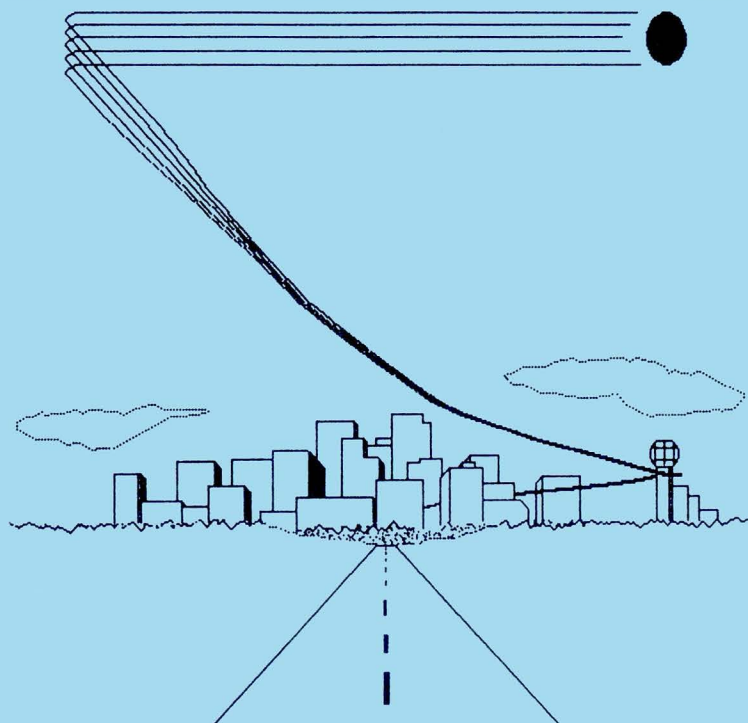
It is important to clear memory before loading an ASCII file. PowerDOT utilizes its own end of file marker in the memory buffer. When writing out in ASCII it looks from the top to the bottom of the memory buffer for the marker. It writes out everything before it. The end of file marker PowerDOT uses are the CHR\$ codes 244, 245, and 246. On a Model III those characters look like the "fickle finger of fate" special Mod III character.

Adding graphics to word processor files: Similarly to the explanation above you can load the file, adjust the width and draw in the graphics and write the file back out. Not all word processors can print graphics. You can also use the Character input mode to insert printer commands.



ITS A BIRD ----- ITS A PLANE ----- ITS -----

POWERDOT



FASTER THAN A <=> ABLE TO CONVERT
"s" TO SINGLE "s"

NOTICE: The enclosed software is provided for the sole use of the registered owner. Please read the non-disclosure agreement below.

REGISTRATION and Non-Disclosure of the Below Stated Product:

The person whose name appears below recognizes that he/she has received a copy of the above stated software, and that this program's only intended use IS for the below stated person for his or her own computer. Further, purchaser agrees that all copies of this program are licensed to him/her for their exclusive use, therefore ANY copy of this program that is given away, loaned, or sold to another person, without the written permission of Breeze/QSD, Inc., will be in violation of the United States Copyright laws, and will result in great damage to Breeze/QSD, Inc. that is far in excess of the copies involved. To continue to receive support, SEND THIS card in! Please fill out completely and LEGIBLY (print or type)

Date _____

Product Name _____ Where purchased _____

Date purchased _____ Version _____

Serial # _____ Type Computer? _____

Your Name _____

Address _____

City _____ State _____ Zip _____

Telephone _____

I have read this agreement, understand it, and agree to its terms.

Signed _____ Date _____

PLACE
STAMP
HERE



A DIVISION OF BREEZE/QSD, INC.

11500 STEMMONS FWY.
SUITE 125
DALLAS, TEXAS 75229