

WELCOME TO SIGNWRITER

SIGNWRITER is a computer software package for producing large, neat characters. It is useful wherever people need to read information from a distance -- in schools, offices, shops, exhibitions, laboratories, travel bureaux, etc. It is also valuable as a low-cost alternative to conventional typesetting and as a means of creating and refining graphic designs -- logos, technical symbols, advertisements, etc.

SIGNWRITER can print onto any flexible material that will pass through a printer, using any type of ribbon that is available. In addition to black lettering on white paper, it can produce coloured signs, adhesive labels, overhead transparencies, and lithographic or duplicator stencils. Various processes exist for transferring images onto other surfaces, notably T-shirts and sheets of metal or plastic.

The version of SIGNWRITER you have here, running on a BBC microcomputer, is actually the baby of a whole family. The programs started life on 16-bit Personal Computers with at least 128K of usable memory and had to be squeezed hard to fit into 32K memory, which is the lowest common denominator of all BBCs. Under its surface there is far more maths than in other character-drawing programs and it concentrates on one simple goal -- giving the highest possible quality of printout on large characters -- and will not win any prizes for speed!

Once you are past the "Getting Started" pages, you should not need to refer to these instructions while running the programs; they are intended mainly as background reading. Many commands mentioned here are for Acorn DFS; the commands may be slightly different for other disk filing systems.



f8
Enter
Sign

f1
Print
Sign

f2
Alter
Font

f3
Process
Font

f4
Toolkit

Signwriter

(C) Wight Scientific 1987

Rewritten for the BBC micro
by Steve Greenley

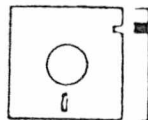
This is the Signwriter main menu

Press a function key

Press Escape to leave Signwriter

GETTING STARTED

Don't run the programs from your master disk. Stick a write-protect tab over the square notch in the side of it. *FORMAT a new disk and then *BACKUP your master onto it. Your DFS manual explains all. Put the master disk away safely. You need it if your working disk gets damaged or to claim upgrades.



Put your SIGNWRITER working disk in drive 0. Type *CAT and look at the disk's directory for a file called README, which explains any changes since this booklet was last reprinted. If there is one, turn on your printer, type *TYPE README followed by control-B and then the Return key. Some files on your master disk are not needed while the programs run and must be deleted from a working disk to leave room in its directory for the program's temporary files. Type *DELETE README <Return>, then *DELETE MOREFNT <Return>.

Now we are ready to begin. Do a SHIFT-BREAK (i.e. hold down the Shift key, press Break, count to three, and release). You should see the screen light up with BBC SIGNWRITER's main menu, as on the opposite page.

BBC SIGNWRITER comes set up for any dot-matrix printer compatible with a recent-model Epson (RX,FX,LX), connected to the BBC's parallel printer port, and with a DIP switch set so that it automatically executes a line feed after each carriage return. If any of these conditions is not fulfilled, please turn to page 10 about the SETTING file.

SIGNWRITER is designed to run on all variants of BBC micro, and the programs need to be told what configuration of disk drive(s) you have. On a BBC B, which treats each surface of a disk as a distinct drive, SIGNWRITER uses two surfaces, one for programs, the other for a processed form of its font. If you have one or more double-sided drives you can skip the next page, because the master disk is already set up for you: it uses drive 0 for programs and drive 2 (the other side of the same disk) for processed font files.

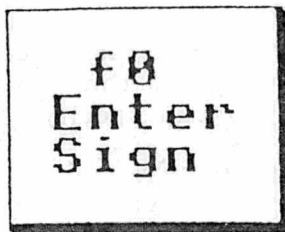
Using single-sided disk drives

If you have two single-sided drives the processed font files should go on drive 1, and if you have just one single-sided drive you will have to use two distinct disks in drive 0, shuttling them in and out when prompted by the programs. In either case you must notify SIGNWRITER. This is a once-only operation for a given set of drive(s); if you fail to do it the programs will crash as soon as they try to look for a font. At the main menu, press key f4 for the "Toolkit", then press key 1, then answer one question with the number 0 or 1 of the disk drive for your processed font.

If you have single-sided drive(s) and cannot read side 2 of your master disk you will have to generate the processed font files from the storage form of the font on side 0. At the main menu press key f3 for "Process font", and follow the program's instructions. The main font supplied on your master disk is called SFONT. Processing a font takes at least ten minutes and can occasionally run into hardware problems, so you may need to read page 20.

In short, SIGNWRITER allows you to put its files wherever is best for your particular configuration of hardware. Just take care that there is always disk and directory space free for the programs to write their temporary files in.

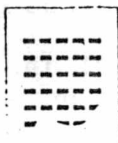
When you press "f0 Enter Sign", the inputting program starts running and immediately reads in two files: SETTING, to tell it about your printer etc, and SPACING, so it knows how to space out the characters in your current font.



The program asks a lot of questions, but many of them have safe default answers you can accept by pressing plain <Return>. If you don't understand something, just plough on and learn by experience. You can't do any harm, so long as your master disk is tucked away safely.

First, there are some preliminary questions about the size and position of the whole sign.

A "horizontal" sign
goes across the paper,
like ordinary text:



A "vertical" sign
goes down the paper
like a banner:



The default side-to-side
position of a sign is in
the middle of the paper,

but you can put
it any distance
in mm from the
left edge, e.g.
0 mm like this,

or further over.

Usually you don't need to worry about the width of a sign. The program checks that you are not trying to exceed the width of the paper but if you want a particular size of sign you can specify its width in mm.

SIGNWRITER can draw a rectangular box to enclose your sign. Three extra questions let you adjust it. This box cuts down the space left for text, of course. It is deceptively easy to use too little white space as a border inside the box or to make the box lines too thick. Also, most printers can print only a limited range of line thicknesses the same in both directions. The box is calculated to miss descenders in g j p q and y on the bottom line, but you must check it with unusual fonts.

Inputting the text of the sign: Once you confirm that your sign's overall dimensions are OK, text input begins. This phase of the program is like a miniature word-processing program. As far as SIGNWRITER is concerned, all characters in a sign are just like ordinary text, no matter how exotic their shapes. All characters on one line are to the same scale and have the same print enhancement and features.

To edit the sign you can move about with the arrow keys. The Delete key deletes a character to the left of the cursor and control-A deletes to the right. Control-U deletes all of the current line. Control-left or control-right arrow keys move by one word left or right, while shift-left or -right jump to line ends. COPY stores the current line for later use on another line and SHIFT-COPY retrieves it.

Any key on the keyboard with a corresponding character in the current font will be accepted. If you type a character that is not in your font, the computer will beep in protest. For a character not on the keyboard, type f6; you will be asked to type in that character's number in the font (see page 15).

Type <Return> to make the program register the current line and adjust the line spacings below it if necessary. If you move around with the arrow keys and edit an earlier line, be sure to press <Return> after editing it.

Function key f5 A box at the top of the screen shows the character height, etc of the line the cursor is on. To alter these features, press key f5. A pop-down menu will appear, with options for changing character height, the position of a line across or down the sign, underlining, spacing, etc. Use the up/down cursor keys to move the illuminated bar to select an item in any pop-down menu, and press <Return> to accept it. To leave a menu, press <Escape> or move to the CANCEL line and press <Return>.

Height of characters sets the size of the characters (so it sets their width too). It actually refers to the height of a typical capital E or H. Typing plain RETURN in response to this question will cause the maximum height of characters that will fit across the sign to be calculated and selected. If you do not specify a height, the default is 12 mm.

Position down the sign refers to the baseline that the characters sit on; the descenders of p, q, and y dangle below. The default distance puts each line 1.5 times its own height below the previous line. This distance looks fairly reasonable if all the lines have the same character height, but is usually wrong if they do not.

Slide-to-side position inside the sign allows you to center a line (the default),

or left-justify or right-justify it, or indent it by a chosen amount. Use this feature, for example, to mix character sizes on one level in a sign by putting a line zero vertical distance down from the line above.

Underlining is drawn 1/2 the capitals' height below a line and 1/8 capital height thick.

Character spacing is normally proportional, i.e. each character in the font has an amount of leading and trailing space that usually (but not always) produces a visually pleasing effect. You can overrule the default spacing in various ways. Regular spacing can be useful for tabulating figures. Adjusting the spacing character by character is fiddly work.

Ordinary characters include any printing style that your printer can normally produce. See the section on SETTING.

When you have finished inputting, press ESCAPE to exit from the editor. Be sure to answer Y to "Save this sign (Y/N)?", unless you really want your sign to disappear for ever. Details of your sign will be stored in a file called SIGN, whose format is described in the next section.

You will be warned if a sign may overflow the paper. If you print such a sign, characters will be cut off at the edge of the paper, but otherwise the printing will be quite normal. This can be used to build up truly enormous characters.

Always begin a complicated sign by drawing it roughly on paper first. Check how it will look by using the "f1 Print Sign" option for a quick draft scaled-down print. To improve the spacing or do anything clever, do the spadework with "f0 Input Sign", then load the SIGN file into into your word-processing program for fine adjustments.

THE STORED SIGN

The SIGN file is an ordinary text file, stored on a disk, which acts as a messenger between the input and the output programs. You can TYPE it. You can load it into a word processing program and alter it or join it to another file. You can even write your own special program to produce it. But if you want to print your SIGN afterwards, take care to leave it strictly in SIGNWRITER's special format, as an ordinary text file, not in some funny word-processor code!

Line 1 of SIGN says HRZ for a horizontal sign, or VRT for vertical. Each line after that represents one character that you typed in (except the spaces), in a special format:

Number in font, Height, Distance across, Distance down,
Name of character

like this: 87,8.00,63.80,38.00,W
or this: 1,12.00,14.42,49.00,smile

Commas separate the items. Spaces are not allowed. Height describes a typical capital letter drawn to the same scale. Character positions refer to the bottom left corner of the character (or a rectangle around it) and are measured across and down the sign, just like a crossword puzzle, regardless of whether it is vertical or horizontal on the paper.

block is an instruction to print a rectangle. A box round a sign consists of 4 long thin blocks. Each block is specified like a character, except that it has no character number:

Width, Depth, Distance across, Distance down, "block"

HRZ
83, 12.0, 5.42, 17.42, S
73, 12.0, 16.97, 17.42, I
71, 12.0, 20.87, 17.42, G
78, 12.0, 32.87, 17.42, N
47.9, 0.42, 0.0, 0.42,block
47.9, 0.42, 0.0, 22.85,block
0.42, 22.0, 0.0, 22.42,block
0.42, 22.0, 47.47, 22.42,block

Here is a simple sign and
the SIGN file that produced
it, padded with spaces for
legibility:



SIGN

SETTING UP FOR YOUR BBC

SIGNWRITER comes set up to drive an 80-column Epson dot-matrix printer (or its equivalents, sold by other companies) connected to the parallel port of a BBC micro. Most details that distinguish one printer from another are in the file SETTING, which the programs read for key pieces of information. You might like to print it out with `*TYPE SETTING <Control-B> <Return>` (followed by `Control-C` to turn off the printer), before reading further.

SETTING is an ordinary text file (what WORDWISE calls a "spooled" file). Most lines are self-explanatory, made up of a number, then a comma, then some text. You can edit it with a word-processing program, so long as you scrupulously retain the structure of the original file. Some lines consist of the word "blank" to maintain compatibility with versions of SIGNWRITER running on other machines.

Though the parent 16-bit version of SIGNWRITER can drive almost all common printers, memory limitations allow the BBC version to drive only Epson-type 9-pin printers with an 8-inch-wide carriage.

Most BBC owners set their printer up to print a Line Feed (ASCII character 10) automatically whenever it receives a Carriage Return (ASCII 13), usually by means of a DIP switch. The SETTING file on your master disk assumes that your printer is set up like this. If it does not auto line feed you must append ",10" to the appropriate lines in SETTING and add one to the byte count on the previous line.

If your printer is connected to the serial port, make the very first number in SETTING negative.

Some dot-matrix printers are Epson-almost-compatible and need small changes to the number of dots per inch across or down the page in SETTING, and occasionally to the number of interleave passes, too. Notable examples are early Star and Panasonic models.

Lines 26 to 29 contain the numbers (27,71,27,69 and 27,72,27,70) to turn on and off ordinary printer characters. For a printer that has an NLQ facility, these should be replaced by the codes found in the manual for that purpose.

f1 Print Sign

To print a sign, select f1 from the main menu. This produces a menu of five options. As usual, you can just press <Return>, but:

- 1) Sends all the bytes that normally go to the printer to a file instead. You can *EXEC them to a printer later.
- 2) If your sign file is called something other than SIGN, this is how to print it.
- 3) Prints a mirror image of the sign, useful for making iron-on transfers for T-shirts, for example.
- 4) Does a faster, low-resolution print-out; useful to give a quick idea of what a sign will look like.
- 5) Scaling down the size of the sign also helps to give a quick preview. Scaling up is not recommended.

After you press <Return> the BBC will philosophize for a minute or so. If it waits twenty minutes, check that the printer is connected and turned on! Then the printer will do a little skip 37/216 of an inch down the page to take up any slack in the paper. (Be prepared to allow for this when positioning a sign precisely on paper.)

Printing an A4-sized sign of average complexity takes at least fifteen minutes, and it will not run any faster if you watch it happening! Printing also uses every byte of memory it can scrounge on a minimal 32K BBC, so messages and screen display are pared right down. You may get a message "Line too complex", which means there are just too many character outlines to be held in memory at that pass of the print-head. All you can do is make the sign simpler, or print it vertically. During printing, if you press any key, the program will pause and allow you to exit.

UNFINISHED BUSINESS

One of the sacrifices that had to be made in adapting 16-bit SIGNWRITER for the BBC is that you cannot read back a sign into "f0 Input Sign" and revise it. So it is worth making an effort to understand the structure of the SIGN file and the file manipulations required to print a sign repeatedly. For example, take care to *RENAME and *COPY to another disk any file that might one day need reprinting.

The SIGNWRITER family of programs and fonts evolves all the time and feedback from customers is a vital part of this process. Here are some examples:

The key to putting signs on T-shirts is finding moderately priced polyester shirts. Contacting Trevor Dean of Image Promotions (01-767-0785) for details.

A short program that can print out a SIGNWRITER-generated headline in the middle of a WORDWISE text is available from Cottage Micro Services (0252-22539) for a small fee.

SIGNWRITER will run on an Electron with a disk drive, except that the screen display may sometimes become messy.

Both of Wight Scientific's best font designers started out as customers. Maybe you too would like to contribute fonts, or even a few characters for a modest royalty. Don't waste your best signs on the National Gallery -- send us a copy!



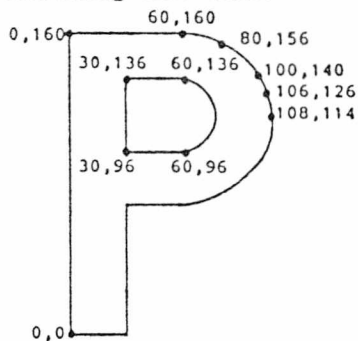
HOW CHARACTERS ARE STORED

While you are new to SIGNWRITER, stick to printing out signs with the default font supplied on your master disk. Once you are ready to use other fonts and create new characters or modify existing ones, read on. Designing neat, well-proportioned characters is actually quite hard work.

Each character in a SIGNWRITER font is defined by a series of X and Y coordinates of points around its outline. These coordinates are measured in arbitrary "grid units". Most characters fit inside an 160 x 160 grid, though they can extend outside these limits.

Here is a diagram of the letter P. Each point around its outline can be reached from the zero point or "origin" (bottom left) by going a distance X across to the right and then a distance Y up. The (X,Y) coordinates of some points in its outline are marked here. If you wished to describe this outline in words you might say something like this:

"Start at position (0,0). Draw a straight line to position (0,160), then another straight line to (60,160). Now draw a smooth circle arc through (60,160), (80,156), (100,140), (106,126), (108,114)... Take your pencil off the paper and go to (60,96); draw a series of straight lines through points (30,96), (30,136), (60,136)..."



This process, like the child's game of joining dots, is essentially how SIGNWRITER handles character outlines. There are serious limitations on where points can go, notably that the values of X and Y must be whole numbers. Large characters are greedy for memory and disk space.

The entry in SFONT for any character consists of two lines. The first line's format is:

Number in font, Space before, Space after, Name of character

A character's number in the font is from 1 to 255 and the space before and after is measured in grid units. The "name" of a character is normally just the character itself, but it can be anything you like, such as "pentagon".

The second line looks like gibberish, for example:

`%YuZu$LUUeeUUE&SLNMNR$UY&\[W_`

Within this gibberish, three characters have special meanings for the dot-joining process just described:

\$ means "Join up the following points with straight lines."

& means "Join up the following points with circle arcs."

% means "Skip over the following points."

After each introductory character (\$ & %), the subsequent characters are taken in pairs, referring respectively to the X and Y coordinates of the next point. Each of these characters represents a number from -40 to +40, calculated by taking its position in the ASCII code (see the end of the booklet) and subtracting 85, and means "number to add to X" then "number to add to Y" -- what mathematicians call delta-X and delta-Y. So, starting from (0,0), the (X,Y) coordinates of successive points are calculated by adding delta-X and delta-Y for each position in succession.

This format keeps SFONT reasonably compact, yet made up purely of characters that can be printed on paper or shown on a screen. So SFONT can be examined or altered with most word-processing programs.

Now for the mathematical constraints on the dot positions. Straight lines and skips are no problem. Curiously enough, nor are the absolute size of a character, the position of its origin, or a hole in the middle. Curves are the hard part. Three points define each circle arc, and the tricky part of designing characters is to find whole-number values of X and Y on which to put points for a smooth change in curvature around a complex character, while keeping the number of points as small as possible.

The printing calculations proceed correctly only if the points around a curved outline obey three strict rules:

- a) A point must be positioned precisely wherever the outline passes through a local maximum or minimum value of X or Y
- b) The second point in any circle-arc sequence must not be at a local maximum or minimum of X or Y.
- c) No three successive points within a circle-arc sequence may lie in a straight line.

It does not matter which character has which number in a font, so long as the numbers are strictly in ascending numerical order. Ultimately the assignment of characters to positions in your font depends on you. SFONT on your master disk contains a core character set of letters, numbers, and punctuation marks, at their normal places in the ASCII code (see the table at the back of this booklet). The nearest approach to an accepted convention where to put less common characters (accented letters, currency symbols, mathematical symbols, etc.) is the "extended ASCII" code of the IBM PC and SIGNWRITER fonts from Wight Scientific follow this convention whenever possible. Really exotic characters obviously have no natural place to go at all.

The commonest character to give trouble is the pound sign. In most fonts you will find it at the the IBM PC position of 156. However, in the main SFONT on BBC disks you should find it at the BBC's allocated position, 96.

To see which characters are at which positions in any font file, just *TYPE it or copy it to the printer. The SPACING file offers a conveniently brief list of the positions of all the characters in any font. Remember that to input a character with no corresponding key on the keyboard you press key f6 and then type in the relevant number.

DESIGNING CHARACTERS

When designing a character from scratch, especially a complex one such as a company logo, it is best to rough out a sketch on squared paper first. You might even want to go the whole way on paper, since the BBC micro does not have enough memory to give you really powerful on-screen design. Here is how to do it:

(1) Mark each point around the character outline with its X and Y coordinates. (2) Choose the order in which to enter the points. (3) Go around the character outline, converting absolute X and Y values into differences from the preceding values. (4) Change these difference figures into the actual characters to go into the second line of the character's SIGNFONT entry (see the ASCII-code table at the end of the booklet). (5) Use a word-processing program to add the new character to SIGNFONT in the appropriate place. (6) Process the font (see page 20) and do a trial print-out. (7) Back to the drawing board to refine it, and so on until you are satisfied. Once you understand the arithmetic involved you can design quite a complex character in a couple of hours, and it can be very satisfying to do so.

On other computers running SIGNWRITER a small Basic program called FONTEDIT is provided as a help in this process, and as a starting point for programmers who want to write their own programs for creating characters. When this booklet was printed it had not been adapted into BBC Basic, but you are welcome to have a copy of the Microsoft Basic program if you send a stamped addressed envelope and a formatted disk.

If you design any new characters, Wight Scientific would be most grateful to receive details so that they can be made available to other users. Be sure to send a printout of the characters and their (X,Y) coordinates. Small royalty payments may be possible for extra fonts or special characters that we succeed in selling separately.

The "f2 Alter Font" option from the main menu can show you any character on the screen. It lets you move around and add or subtract parts of the outline to see how the character changes. It also takes care of the book-keeping of editing characters, storing them in the font, moving them, deleting them, etc.



Answer the program's first question with the name of a font file to work on -- usually just the default, SFONT on the currently logged drive. Do not proceed any further unless you have a backup copy of the font file stored somewhere safe; don't risk messing up your only copy of a valuable file. Practise on existing characters before you launch into creating new ones.

Information about the whole font file is read into memory, and then a menu of options is displayed. Each time you edit a character, the new details are placed in a "session file", which is not incorporated into the font until you opt to leave the program.

The first menu lets you edit an existing character, create a new one, or delete one. To edit an existing character you choose the character by positioning a cursor in the font window over it. When creating a new character you must first supply a name and number for it.

Characters should normally be positioned at the bottom left of the grid (the grid is a scaling guide revealed by pressing f2). The outline window automatically accommodates the whole of a character; this can cause disconcerting changes in apparent size on-screen. The scale is reflected in the size of the cross-hairs and the grid (when active). After the character is drawn, the flashing cross-hair cursor will appear on the first point. The cursor keys allow you to move the current point. When you press RETURN or another command key the display of the character outline will be updated, taking the new point position into account.

Function key f0 moves the cursor from point to point around the character outline (auto-repeating if held down). Notice that there is a point wherever the outline changes direction or changes from a straight line to a curve. There are also points part-way along very long lines, which must be treated as several lines. (You need not calculate them; the program does it before storing a character.) f0 moves from point to point in the order that they are stored in SFONT, which may not be obvious on the screen. Two points are superimposed where one sequence ends and another begins, so the cursor will sometimes appear not to move when you press f0. Function key f1 moves the cursor in the opposite direction.

It is easy to put a curve-point in an illegal position (see the rules on page 15), but DESIGN will protest with a brief diagnosis of the problem in the command window and/or will refuse to draw the faulty section of curve until you move the point to a legal position.

To add a new point, press the COPY key. The new point is added on top of the existing one at the cursor position, and can then be moved to wherever you want it. Similarly, to delete a point, position the cursor on it and press the DELETE key; the cursor then moves to the next point.

To create a new sequence of straight lines, press function key f3 and follow the prompts, pressing RETURN to mark each point. As new points are input the sequence appears on the screen, as if you were stretching string over pegs sticking up from a board. To form a closed loop (i.e. a complete outline) you must bring the cursor all the way back to the first point and put a final point there too. To mark the final point, press the Space Bar instead of RETURN.

Creating a new sequence of circle-arcs (function key f4) is tricky. A new sequence cannot be drawn until you are on the third point (because three points are needed to define a unique circle). Even thereafter you will probably get some temporarily illegal, or very odd outlines. Do not obsess about getting points positioned right first time; aim for a good number of points, to be moved into place later.

Function key f5 reveals a sub-menu of "sequence operations", when one sequence of straight lines or circle arcs in the character outline is highlighted like a string of beads. F0 and F1 now jump from sequence to sequence, the arrow keys move a whole sequence bodily, DELETE deletes all of it, COPY duplicates it, and so on. You can also reverse the order of points within one sequence, to help satisfy the constraints on points in circle arcs or to simplify the storage string.

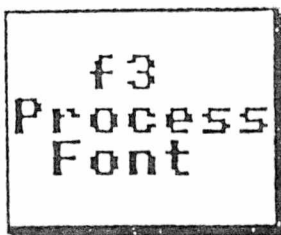
A sub-menu of whole-character operations can be accessed via function key f6. These include rotation by 90 degrees clockwise or anticlockwise, reflection up/down or side-to-side as in a mirror, expansion (or contraction) by 10%, etc. Expansion usually distorts a character, because of rounding errors when X and Y coordinates are made into whole numbers.

Press f8 to alter the spacing either side of a character between -128 and +128 grid units. The outline of one character can contain up to 75 points in up to 30 sequences.

To stop editing a particular character press ESCAPE. You will then be given the option to abandon the amendments just made, and returned to the first edit menu.

Make sure, if you buy another font, that you receive and read the accompanying general fonts instruction sheet and any specific instructions for that particular font. There are many ways in which whole fonts or individual characters designed on 16-bit computers can bust, or sail very close to, the limits imposed by the BBC's memory, disk capacity, and directory size. At the time of printing, SIGNWRITER fonts are almost fully interchangeable among all computers, but in due course the IBM version will develop to a point where characters that are legal on a PC may become illegal on a BBC.

SIGNWRITER's font storage format is designed for compactness and for humans to read, not computers. Each time you modify a font, it must be processed into the actual files that are used, FASTFNT and SPACING. Do this with the "Process Font" option in the main menu.



This needs a lot of file space. If you have 80-track (or double-density) drives you can probably fit the files onto the same disk as the SIGNWRITER programs; on a single 40-track drive you must first copy the font file onto a blank disk, which is to receive the FASTFNT files, and insert this disk when prompted after selecting "Process Font".

Any versions of FASTFNT and SPACING already on the disk will be over-written by the new versions, so make sure that copies of your existing versions (or the SFONT from which they were produced) are safely backed up on another disk.

About 15 minutes of hard calculations will now follow (so don't stay watching the screen), converting the X and Y coordinates of points around the outline of each character into mathematical parameters that the sign-printing program can find and use as fast as possible. (Warning: Tired old disk drives or bugged DFSs can systematically corrupt disks during this operation; they attempt to write to a disk before it is running at the correct speed.)

During this process the font is checked for any problems that would foul up the calculations involved in printing. Any problems found are reported on screen and in the file FAULTS. Many reported faults are non-fatal, giving a usable font, but the faulty character may look a bit odd or may generate a streak across the page. Not all faults in a character outline are detected: for example gaps or overlaps may first show up as streaks across the output, like this:

faulty p

COPYING SIGNWRITER AND CLAIMING UPDATES

SIGNWRITER is sold for use on one computer by a single individual or a small group of people, but the disk is not copy-protected in any of the usual ways with misformatted tracks, etc. If you give away or sell a copy for use on another computer you are infringing the program authors' copyright and breaking the law; pirate copies of programs have a habit of diffusing away from their original source, yet still being traceable (from internal markers in the programs) back to the original purchaser!

The biggest incentive for buying a legitimate copy of the program is that all programs improve with time, as bugs are removed and further features added. You can obtain an update to the latest version of BBC SIGNWRITER in exchange for your master disk (or alternatively a discount on the price of one of the 16-bit versions if you trade up to a more powerful computer). The first such exchange is free and subsequent ones cost £5. However, the BBC version is unlikely to develop as far as the parent 16-bit version, both because of the limitations of the BBC computer and because this version was produced outside the offices of Wight Scientific. To claim an update, send a brief letter bearing your name, full postal address, and details of where and when you bought your copy of SIGNWRITER to:

BBC SIGNWRITER updates
Wight Scientific
44 Roan Street
LONDON SE10 9JT

Please describe any problems you encountered or suggestions you have for improving the programs. Enclose in that letter a large self-adhesive label with your name and full postal address, legible and understandable by any postal sorting clerk. Also enclose your old master disk. The new version will be put on this same disk, so be sure to keep a copy of the old version (in case you don't like the changes). The best way to send a floppy disk through the post is taped between two pieces of corrugated cardboard (cut from an old box) with their grains running at right angles, then inside an ordinary brown envelope. Special disk mailers or registered post are a waste of money.

PROBLEMS?

All programs contain bugs. If you find one in SIGNWRITER, please report it promptly. However, before you blame the programs, please check carefully that you are not in some way making a mistake. Is your equipment set up correctly, particularly the computer-to-printer cable? Have you accidentally corrupted one of the files? You may also find the following comments about problems helpful:

Stripes across your output can arise from a worn printer ribbon or poor alignment in the pins of the print-head (the cure in both cases is to buy a new one). With a BBC micro the most common cause is failure to set the auto-line-feed switch (or change SETTING). For the very best quality your printer can produce, at the cost of a nearly 8-fold reduction in speed, you can modify SETTING to use only one wire in a dot-matrix print-head on each pass across the paper. Some makes of printer are inherently faulty in their built-in logic or mechanical movement and you may then have to play with the numbers in SETTING.

Using the wrong filename Be sure to distinguish these two files: (a) the messenger from sign-input to sign-output, usually called SIGN, and described on page 9; (b) the stream of graphics data that normally goes straight to the printer but can be redirected into a file by selecting option 1 of the print-out menu.

Worn-out ribbons Printing signs with uses up ribbons even faster than word-processing. The cost can be reduced by buying many ribbons at a time, but be careful: there are many inferior grades of ribbon on the market and ribbons have a limited shelf life (the ink evaporates). We recommend the re-inking service offered by Aladdin, 4 Hurkur Crescent, Eyemouth, Berwickshire, TD14 5AP (telephone 08907-50965).

Uneven line thicknesses SIGNWRITER produces its best results on really large characters. Below 6 mm in character height the number of dots required to build up a bar in a letter is low enough that the rounding errors involved in using rows of printable dots can make the characters look less good, especially m, n, and h. In this size range it is usually better to use your printer's own character font.

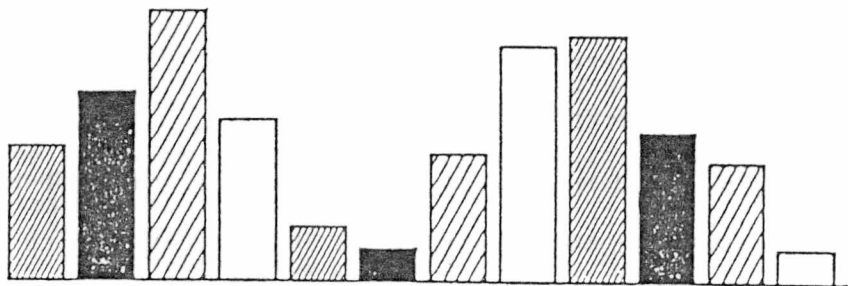
Total failure to print Certain ROMs mess up the BBC's memory map and foul up well-behaved programs, such as SIGNWRITER. If in doubt, take 'em out!

Disk full This error message occurs most often with large fonts and small disk capacities. Your disk may not actually be full (owing to inefficiencies, small directory size, and outright bugs in various BBC DFSSs) but the answer is the same: delete all unnecessary files, *COMPACT the disk, and if all else fails, cut down that font.

Process-font crashes If your BBC almost finishes processing a large font, and then gives some unhelpful error message like "Channel", don't give up -- press BREAK. FASTFNT and SPACING are good, as far as they go, except that SPACING has a ragged ending that you can chop off with a word-processor.

WANT TO KNOW MORE?

SIGNWRITER was developed from an earlier program called HEADLINE, published by Wight Scientific in 1984 as part of "PLOT PLOT PLOT - computer programs to draw figures and graphs". This package was aimed at scientists engaged in writing up research results for publication who need to produce camera-ready graphics at minimum cost. PLOT PLOT PLOT is sold as a booklet with the full source code (in Microsoft Basic) and thorough explanation of the programs, together with a discussion of how to send graphics data to printers, plus an adaptation into BBC Basic. The booklet costs £10 to a single individual with a single computer. A disk with all the programs costs a further £25+VAT. Here is a histogram produced on an Epson printer by these programs:



The ASCII Code

0	32	space	64	@	96	`
1	33	!	65	A	97	a
2	34	"	66	B	98	b
3	35	#	67	C	99	c
4	36	\$	68	D	100	d
5	37	%	69	E	101	e
6	38	&	70	F	102	f
7	39	'	71	G	103	g
8	40	(72	H	104	h
9	41)	73	I	105	i
10	42	*	74	J	106	j
11	43	+	75	K	107	k
12	44	,	76	L	108	l
13	45	-	77	M	109	m
14	46	.	78	N	110	n
15	47	/	79	O	111	o
16	48	0	80	P	112	p
17	49	1	81	Q	113	q
18	50	2	82	R	114	r
19	51	3	83	S	115	s
20	52	4	84	T	116	t
21	53	5	85	U	117	u
22	54	6	86	V	118	v
23	55	7	87	W	119	w
24	56	8	88	X	120	x
25	57	9	89	Y	121	y
26	58	:	90	Z	122	z
27	59	;	91	[123	{
28	60	<	92	\	124	
29	61	=	93]	125	}
30	62	>	94	^	126	~
31	63	?	95	_	127	delete

128-255 IBM PC extended-ASCII code: accented letters, box outlines, Greek letters, etc, including 156 Pound sign £

16-bit SIGNWRITER was written by Anthony Durham, Edward Bishop, Robert Insall, and Suzanne Miller. The BBC version was produced by Steve Greenley, in Stamford and Newcastle.

Copyright Wight Scientific, 44 Roan Street, London SE10 9JT
These instructions were printed in July 1987.