

EDITOR'S NOTES

I apologize for the tardiness of this issue of the BoSTUG newsletter. Part is due to delays in receiving contributions; part is due to my moving, which has scattered my life in three directions and explains some sloppiness in the layout.

Again we have a range of offerings, but from only a few contributors.

The good news is that we have halted the decline in ST user group membership - up one to 157. The bad news is that less than five percent of the membership participates in the activities.

Over a hundred of us live within Interstate 495. We would like to know what we are doing right and what we can do to be righter. A call from you to me at 889-0830 would be a big help. We don't want to lose you so long as you retain an interest in the Sinclair computer of your choice.

Another bit of news in the way of support for Sinclair Timex users is the announcement from the gossip mill that a number of dedicated ST users are banding together to form a National or even North American S/T user group.

Editorially we applaud this move, although it is about three years overdue.

Some advice: my experience with the ST User Group here suggests that about 20% of a membership will participate in an activity, but only 5% will become actively involved.

We wish the National Sinclair Timex User Group early success and encourage all BoSTUG members to join.

DIRECTORY

Co-director	Peter Hale	889-0830
Co-director	John Kemany	263-3347
General Questions	John Kemany	263-3347
Machine Language	John Kemany	263-3347
Hardware Library	Lee Ball	Lee, what's your PHONE?
Software Library	Jim Rodlin	481-2155
Editor (<u>pro tem</u>)	Peter Hale	889-0830
QL sub-Group	Peter Hale	889-0830
Flexi-BBS		648-7651
Sysop:	Bob Cutter	646-4425
TIMEVARP (BBS)		481-0555
Sysop:	Jim Rodlin	481-2155
Member Services		
Liason	Will Stackman	547-0655

The Sinclair/Timex Newsletter is the official publication of the Boston Computer Society's Sinclair/Timex User Group (BoSTUG).

Membership is \$35.00 per year from the Society at 1 Center Plaza, Boston, MA 02108.

The newsletter maintains an exchange policy with Sinclair/Timex User groups willing to reciprocate. Please forward exchange issues to the editor, Peter Hale, at P.O. Box 8763, Boston, MA 02114.

Unless copywritten by the author, articles are in the public domain. Attribution is requested.

TABLE of CONTENTS

	page
Editor's Notes	1
Directory	1
Letter to Editor	2
2068 Windows, Part II	3
Psion Productivity Tip	5
Text87 - A Review	6
RamDisk for QL (program)	6
2068 Telecommunications Notes	10
SuperBASIC for BEGINNERS PRINT	11
Query on ABACUS	12
Going Online, T/S Style	12
Calendar Notes	13
Sing a Song of Silliness	14
Sinclair Fests Abound	15
QL Sold Out?	15
One Meg Expansion for QL	15

LETTER TO THE EDITOR

Congratulations for one of the finest (if not THE finest), and most informative User's Group Newsletter, that I have seen. I am a connoisseur of newsletters; therefore I should know whereof I speak. I get four different User's Groups newsletters from the B.C.S., one from a dealer, two from a National organization one from a publisher in New Hampshire, and recently I bought seven issues of QUANTA, which I haven't been able to read yet.

The reason I applaud your latest effort is because of its clarity. As you know, I am housebound and cannot attend any of the B.C.S. meetings, although I have been a member for several years. I started PC-ing with the ZX-80; then the ZX-81, when the ZX-80 destroyed itself.

I should have stuck with the ZX-81 because after that, things got complicated.

I graduated to a CP/M machine which couldn't do anything

unless a couple of disks were loaded into the maws of its drives. The instruction manuals were numerous and not written for the layman. It could employ four different Basic languages, which relied on disk-loaded operating systems. I never became conversant with any one of them because a veritable hail of programs on disks became available through dealers who bombarded me with their catalogues and I obtained many from the Public Domain.

Then Sinclair cozened me into buying a QL. I succumbed because of my experience with the ZX machines, which I had learned to regret abandoning.

I soon regretted my purchase, however, chiefly because of the weight, awkward bulk and writing style of the User's Manual. Terms were used that I couldn't understand and there was a dearth of examples like your "e.g.s" in the article on using SuperBASIC 'COPY'.

The only fault with your article was that you did not emphasize the need to read the article with the computer ON. It all seemed so clear when I read it, but I didn't teach my hands to use the new knowledge until much later.

I am reminded of Sophocles' ancient dictum: "Although you may think you know a thing, you can never be certain until you actually do it."

Mike Mitchell's (no relation) experience with his QL is just the opposite of mine. I've had mine for two years - going on three now - and have only partially mastered Quill and if it hadn't been for your Tax-i-QL, I wouldn't even have tried to use Abacus.

I host a QL subgroup meeting

on a bi-monthly basis, and have benefitted greatly by getting to know you and Henry April and the others.

However, I don't learn much at the QL sub-group meetings here because of my poor eyesight and deafness, but I look forward to them as a high point in my dull existence.

I keep comparing the QL to the CP/M computer that I have and the QL comes off "second best".

A disturbing slowness exists in Quill which causes some funny effects when following a fast typist. Furthermore there are several features in my CP/M Perfect Writer word processor program that I wish were in Quill. For instance there is no Global Replace ability - which takes less than two seconds in a 14,000 byte letter in Perfect Writer.

(You may want to learn about TRIT87, an review of which is elsewhere. Ed.)

Likewise, I can move a block of text marked at its begining and its end with two key strokes and cursor movement. I agree that the same can be done in Quill, but you have to wait until each letter is blocked out in white at the rate of about 1 per second.

I haven't tried to move large blocks around in a single file, but I have merged a portion of one file into another, by deleting what I didn't want to move, naming the portion differently and merging. In the Perfect Writer program this is done with two windows and is very fast.

Keep up the GOOD work.

- John Mitchell
Westwood, MA

Theory of Windowing

Windowing is a display technique for dividing a computer screen into separate, independently controlled areas for text, graphics or other forms of information. This article will explore techniques that make windowing possible.

For windowing displays, a number of problems must be dealt with. First, screen contents must be preserved when a window is opened (Not all windows do this). Screen preservation is called 'Non-destructive windowing'.

One non-destructive technique, discussed in the January, 1988, S/T newsletter involves copying the entire screen into memory before opening a window. This method is inefficient if more than one window is involved or windows are more sophisticated than simple 'pop-up' messages. For real windowing, we must find some other method.

Other problems a window driver must handle are: text from a window must be confined within its window (no 'spillage'), text must be directed to its proper window or the main screen, overlap must be handled without distorting the screen or losing text, and it all must be done QUICKLY.

The idea proposed involves storing data in what I call a 'window file'. It only works with text, and it depends on the computer's ability to identify a character printed on its own screen (eg. SCREEN\$(Y,X) for the IS 2088).

The structure of a window file is basically a string of display characters which constitute the text in that window, plus the text that the

window is overlapping or 'shadowing':

Byte Definition	Notes
1 ROW /--	Upper left hand
2 COLUMN \--	corner of window
3 WIDTH /---	Actual text area < is 2 less than W,D
4 DEPTH \----	to take border chars into account.
5 RPP.....	row print position
6 CPP...	column print position

WxD

```
-----
|                                     |
|           Window                   |
|                                     |
|           Text                     |
|                                     |
|-----
```

WxD

```
-----
|                                     |
|           Shadow                   |
|                                     |
|           Text                     |
|                                     |
|-----
```

A window file is created internally whenever a window is opened and purged when the window is closed. To print to a window, text is placed directly in the window file. A separate display routine handles the actual printing later by copying everything it finds in the window file onto the screen.

A window file's header contains six bytes signifying window position, dimensions, and current print position within the window. Print position is updated by incrementing COL for each character until the right border is reached or a CHR\$(13) (ENTER) is encountered. In that case, COL = left margin and ROW = ROW + 1.

Following the header is the

information that the window holds. It is stored as strings of text in the order that they appear in the window: Row 1 (Top line), Row 2, Row 3 etc. (Note: NO Return characters are stored in the window file -- only spaces and letters; the last character in row 3 is followed immediately by the first character in row 4).

Immediately following the Window text file is another file, the Shadow text file. The shadow text file keeps track of the text that was underneath the window when the window was opened, in order to create windows without corrupting the main display. When a new window is opened, one of the first things it does is copy that part of the screen (including all text or even another window) into the new window's shadow file. Thus, when the window is closed, it knows what to put back on the screen.

If you keep your window print routine separate from your screen update routine, you can do some interesting tricks. One idea is to effect 'hidden windows' by reprinting the shadow file, causing the window to disappear while maintaining the window's content. Hidden windows can be scrolled, printed to, or cleared without effecting the main screen this way. To get the window back it is only necessary to call the screen updating routine, and the window will reappear instantly! This is useful for viewing an underlying screen display without closing the window. Just be sure to update the shadow file before reopening the window.

Scrolling a window is straight forward. Copy each row in the window file onto the one above (or below) it and insert a blank line in the last row.

Afterwards, call the screen update routine to refresh the display. Just scroll the window one line at a time and it's a snap. Bi-directional scrolling is possible this way. Scrolling a window does not affect the shadow file.

Occasionally the programmer may wish to scroll the display UNDERNEATH an open window. This is where life gets complicated and the shadow file gets put to real use. The trick is to handle the four areas around the window (above, left, right, below) separately, and to pay special attention to the part that scrolls behind the window. For that, the top row of the Shadow file is copied onto the line above the window, the Shadow file is scrolled internally and then the part of the line just below the window is copied into the bottom row of the Shadow file. The sight of a stationary window floating above a scrolling background is impressive and worth the effort.

Finally, to close a window just copy the shadow file back onto the screen and purge the window file.

Theoretically, the only limit to the number of windows in use at once is set by the amount of spare RAM. Since window files store text only, they are very memory efficient (A 10 by 30 character window uses only 606 bytes including the shadow. An equal portion of the display file requires 2400 bytes, not counting color attributes!). However, window handling becomes very slow if more than 2 or 3 windows are open at once.

I have not yet tried to implement my window files concept with actual code. This may change by the time the next issue of the newsletter

comes out. If it does, I will include a listing with my next article. I hope that this discussion inspires a few of you to try to write a windowing routine along these lines. I suppose that a window file could be stored as an array or long string variable in BASIC (note: an array takes SIX bytes for every element stored!) but I think a window driver would have to be compiled or written in machine code to be acceptable. Whatever your preference, good luck and Keep On Timexing...

- Jim Rodlin

PSION PRODUCTIVITY TIPS

A limitation to Psion programs is that file names may not exceed 8 characters. Right?

Wrong! We know that Quill adds the extension "_doc" to any file name saved from it. This is useful in recognising a file type by its extension.

However, you may give any three character extension (including numerals and symbols in any order) to a Quill file and provided you enter it as well as the rest of the file name, the file will load.

The advantage in having more characters to describe the file will be obvious to anyone with extensive correspondence.

JJones_304 might save a letter to John Jones on March 4th.

The followup on April 15 might be JJ415_etc or JJ304_415 or JJ415_&&& but never JJ&&&_415.

Caution! You are on your own indexing your correspondence. Any index system must be consistent. Don't use extensions which can be confused with other file types.

- Peter Hale

Rarely does software arrive on the QL scene with a potential for dramatic changes in the way the computer is used, but TEXT87 is such a program.

I only recently received my copy and am not yet familiar with all its features. Otherwise I would be using it to write this newsletter.

Briefly it is a What-you-see-is-what-you-get text editor. It offers all the features that would have made Quill unsurpassed, and gives up little without paying the price of needing to reference dozens of control codes.

It has on-screen prompts and a heirarchical menu, much like the Psion programs. It is happy under Taskmaster but can be multitasked with Control C.

First, you move around a large document and move or erase blocks of text very rapidly.

Second, instead of the Quill's Design feature that commits you to certain parameters for the whole document, TEXT87 uses 'rulers' that set margins, tabs, justification and line-feeds. Rulers can be saved and recalled for later use in different parts of documents or other documents.

Third, you may have multiple lines in headers and footers.

Fourth, you may 'Go to' any part of the document directly by specifying a line number, and the range of cursor controls is wider than for Quill.

Fifth, (I am always in favor of a fifth) Quill documents may be easily imported through a special routine. Other text files can also be imported in a general routine. (CONT p9)

RAMdisking capabilities can increase the versatility of a computer by providing temporary file storage.

RAMdisk reserves a portion of memory for files. Thereafter you access the file the same way you access it on disk or on microcartridge. There are two differences: The device is called ram as in 'ram_fred'. The other difference is that access to the file is very much quicker than to files stored on magnetic media.

A practical application is in exporting and importing files between PSION programs.

QDOS anticipated ram diskling and allows for up to eight ram disks to be addressed.

Of the two types of ramdisking the first is called dynamic ramdisking and creates eight zero sectorred ramdisks. As files are stored in a ramdisk its size expands up to the available free memory in ram; a 'DIR ram1_' might show -22/0 for a file of 22 sectors. The advantage is that only as much free memory is allocated for ramdisking as is required.

The other type, published here, is a static ramdisk. You determine how many sectors you want when formatting the ramdisk. A certain amount of guessing is necessary.

Key-in the following Super-BASIC program. There is a lot of hex data but it would be 50% greater in decimal code.

When done, RUN the program with a formatted cartridge in mdv1. Two things happen: You create "ramdisk_bin", a binary file, and a SuperBASIC file called ram_boot that loads the binary code and generates a sample of ramdisking speed.

```

1000 REMark Program reads in hex codes from data lines & outputs them to a file
1010 RESTORE
1020 OPEN_NEW #4,mdvl_ramdisk_bin
1030 REPEAT readloop
1040   IF EOF THEN EXIT readloop
1050   READ a$
1060   pos=1 : l=LEN(a$)
1070   REPEAT spacestripper
1080     same=pos
1090     IF pos>l THEN NEXT readloop
1100     pos=pos+ (a$(pos)=" ")
1110     IF pos<>same THEN NEXT spacestripper
1120     b$=a$(pos TO pos+1)
1130     IF b$(2)=" " THEN b$="0"&b$(1)
1140     pos=pos+2
1150     value%=0
1160     FOR n=1,2
1170       cd=CODE(b$(n))
1180       value%=value%+(3-n)^4*(cd-48-7*(cd>64 AND cd<71)-39*(cd>96 AND cd<103))
1190     END FOR n
1200     PRINT #4, CHR$(value%);
1210   END REPEAT spacestripper
1220 END REPEAT readloop
1230 CLOSE #4
1240 OPEN_NEW #4,mdvl_ram_boot
1250 PRINT #4,"100 addr=RESPR(2048)"
1260 PRINT #4,"110 LBYTES mdvl_ramdisk_bin,addr"
1270 PRINT #4,"120 CALL addr"
1280 PRINT #4,"130 REMark In the next line, change the value '100' to suit"
1290 PRINT #4,"140 FORMAT ram1_100"
1295 PRINT #4,"150 REMark Setting up example filenames to demo DIR RAM1_"
1297 PRINT #4,"160 FOR n=1 TO 10"
1298 PRINT #4,"170 OPEN_NEW #4,'RAM1_EXAMPLE FILENAME NUMBER: ' & n"
1299 PRINT #4,"180 END FOR n"
1300 CLOSE #4
1310 :
10000 DATA " 61 20 43 FA 00 0C 34 78 01 10 4E 92 70 00 4E 75 00 01 06 2E 07 52 "
10010 DATA " 41 4D 5F 55 53 45 00 00 00 00 00 00 70 18 72 62 74 00 48 41 47 E8 "
10020 DATA " 00 1C 45 FA 00 F4 26 CA 45 FA 03 64 26 CA 45 FA 04 E4 26 CA D6 FC "
10030 DATA " 00 0C 45 FA 05 48 26 CA 26 FC 00 00 00 24 36 FC 00 03 26 FC 52 41 "
10040 DATA " 4D 30 41 E8 00 18 70 22 4E 41 4E 75 70 03 2E 28 00 24 EF 4F EE 8F "
10050 DATA " 74 C0 D4 87 60 00 01 DA 61 00 01 BE 66 26 38 19 5B 44 83 1E 0C 44 "
10060 DATA " 00 29 62 18 20 3C DF DF DF FF C0 99 90 2A 00 14 B0 AB 00 3E 66 06 "
10070 DATA " 0C 19 00 5F 67 04 70 F4 4E 75 4B E8 00 68 2E 09 3A 28 00 1E 42 68 "
10080 DATA " 00 1E 52 68 00 1E 70 47 74 40 43 E8 00 58 61 4A 67 08 0C 40 FF F6 "
10090 DATA " 67 14 60 0C 36 04 22 47 61 00 03 E2 66 DE 70 F8 31 45 00 1E 4E 75 "
10100 DATA " 45 E8 00 58 70 12 42 62 51 C8 FF FC 22 4A 2A 47 34 C4 14 DD 53 44 "
10110 DATA " 8E FA 31 45 00 1E 72 0E 74 26 1F 28 00 2C 61 06 11 5F 00 2C 4E 75 "
10120 DATA " 70 FF 48 E7 BF 0C 4C D7 00 05 76 01 61 0E 52 80 67 F4 53 80 58 8F "
10130 DATA " 4C DF 30 FC 4E 75 1C 28 00 1D 61 00 04 4A 4A 80 6D 00 00 BA 0C 00 "
10140 DATA " 00 40 65 00 01 12 0C 00 00 4B 62 22 D0 40 30 3B 00 86 4E FB 00 02 "
10150 DATA " 00 1C 00 1C 00 20 00 24 00 18 00 66 00 A4 00 86 FF 1A 00 82 FF 2C "
10160 DATA " 03 96 70 F1 4E 75 70 00 4E 75 70 00 60 14 70 00 24 28 00 20 EF 4A "
10170 DATA " EE 8A 04 82 00 00 00 40 D2 82 69 1A 24 01 6B 20 06 82 00 00 00 40 "
10180 DATA " 69 0E ED 82 69 0A D4 82 EE 4A B4 A8 00 24 6F 0E 72 00 24 28 00 24 "
10190 DATA " 70 F6 60 CE 74 40 72 00 21 42 00 20 4E 75 20 3C 52 41 4D 30 D0 2A "
10200 DATA " 00 14 22 C0 22 FC 5F 20 20 20 32 FC 20 20 22 14 70 00 4E 75 70 07 "
10210 DATA " 60 7E 7A 03 0C 42 00 40 6E 0E 2F 09 61 16 24 5F 04 92 00 00 00 40 "
10220 DATA " 4E 75 70 FC 4E 75 7A 07 60 06 7A 07 74 0E 72 00 78 00 38 28 00 1E "
10230 DATA " 67 00 FF 66 3F 04 2F 28 00 24 2F 28 00 20 42 68 00 1E 21 6C 00 04 "

```

```

10240 DATA " 00 24 53 44 ED 8C D2 84 2F 02 61 00 FF 66 24 1F 20 05 72 00 61 24 "
10250 DATA " 21 5F 00 20 21 5F 00 24 31 5F 00 1E 4A 80 4E 75 16 28 00 1C 53 03 "
10260 DATA " 67 08 57 03 67 04 B0 00 4E 75 70 EC 4E 75 48 C1 48 C2 0C 00 00 07 "
10270 DATA " 62 00 FF 0E 7E 00 4A 83 6C 02 9E 81 59 00 67 00 FF 00 6D 10 61 CE "
10280 DATA " 66 DE 76 FF 55 00 67 00 FE F2 6D 22 6E 10 76 00 58 00 67 1A 76 0A "
10290 DATA " 55 00 67 04 6D 10 E1 4B DE 89 2F 07 DE 82 61 18 22 09 92 9F 4E 75 "
10300 DATA " E1 4B 2F 01 43 EF 00 03 2E 09 52 87 61 02 22 1F 4E 75 2A 28 00 1E "
10310 DATA " 28 28 00 20 B8 A8 00 24 6D 0E 6E 04 4A 03 6D 04 70 F6 4E 75 4A 44 "
10320 DATA " 67 08 61 00 00 A0 66 F4 60 0C BE 89 63 00 00 8A 61 00 00 A8 66 E6 "
10330 DATA " 4A 43 67 7E 20 0D 90 8C 51 80 EF 88 4B F4 08 08 30 2C 00 02 E5 48 "
10340 DATA " DA C0 DA C4 4A 43 6E 26 BE 89 63 12 1A D9 52 44 08 04 00 09 67 F2 "
10350 DATA " 52 45 06 84 00 00 FE 00 50 E8 00 2C B8 A8 00 24 6D 2C 21 44 00 24 "
10360 DATA " 60 26 70 00 BE 89 63 20 B8 A8 00 24 6C 36 10 1D 12 C0 B6 40 66 02 "
10370 DATA " 2E 09 52 44 08 04 00 09 67 E4 52 45 06 84 00 00 FE 00 21 44 00 20 "
10380 DATA " BE 89 62 00 FF 5E 0C 43 00 0A 66 08 B6 00 67 04 70 FB 4E 75 70 00 "
10390 DATA " 4E 75 21 44 00 20 60 00 FF 58 4B EC 00 08 32 2C 00 02 53 41 70 F0 "
10400 DATA " BA 9D 57 C9 FF FC 59 8D 4E 75 4B EC 00 08 32 2C 00 02 53 41 70 F5 "
10410 DATA " 0C 15 00 FD 58 8D 57 C9 FF F8 66 06 2B 05 53 54 70 00 4E 75 1C 28 "
10420 DATA " 00 1D 61 00 01 D4 67 46 4B E8 00 58 74 40 21 42 00 20 21 6C 00 04 "
10430 DATA " 00 24 0C 28 00 04 00 1C 67 00 00 BE 78 00 7A 00 52 45 61 00 00 DE "
10440 DATA " 66 28 4A A8 00 58 67 12 43 E8 00 32 36 19 4B E8 00 68 61 00 00 D6 "
10450 DATA " 66 E2 60 6E 4A 44 66 DC 38 05 60 D8 70 F9 4E 75 70 F5 4E 75 0C 80 "
10460 DATA " FF FF FF F6 66 7A 10 28 00 1C 6D 72 55 00 6D E6 7C 00 4A 54 67 E4 "
10470 DATA " 4A 44 67 0C 3A 04 ED 8C EF 8C EE 4C 21 44 00 20 4B E8 00 58 2A C2 "
10480 DATA " 42 5D 42 9D 42 9D 70 12 43 E8 00 32 3A D9 51 C8 FF FC 42 9D 42 9D "
10490 DATA " 42 9D 61 6E 29 68 00 24 00 04 42 A8 00 20 21 46 00 24 31 45 00 1E "
10500 DATA " 61 5A 60 24 10 28 00 1C 6D 32 0C 00 00 02 67 1A 6E 22 22 28 00 58 "
10510 DATA " EF 89 EE 49 31 45 00 1E 21 42 00 20 21 41 00 24 70 00 4E 75 70 F8 "
10520 DATA " 4E 75 42 68 00 32 60 F2 38 05 7C 40 61 88 60 5E 78 00 61 6C ED 8D "
10530 DATA " EF 8D EE 4D 21 45 00 20 70 40 4B E8 00 98 42 A5 59 40 6E FA 60 04 "
10540 DATA " 70 03 60 02 70 07 43 E8 00 58 60 00 FC 5A B6 6D FF FE 66 0E 60 08 "
10550 DATA " 61 0C 10 01 61 08 B0 01 56 CB FF F6 4E 75 C3 4D 12 35 30 00 0C 01 "
10560 DATA " 00 61 6D 0A 0C 01 00 7A 6E 04 04 01 00 20 4E 75 61 00 FD 54 66 E0 "
10570 DATA " 28 28 00 20 21 44 00 24 53 84 48 44 52 44 3A 28 00 1E 4B EC 00 08 "
10580 DATA " 30 2C 00 02 53 40 BA 5D 66 0C B8 55 62 08 1B 7C 00 FD FF FE 52 54 "
10590 DATA " 54 8D 51 C8 FF EC 50 E8 00 2C 70 00 4E 75 48 E7 0F 8C 1C 28 00 1D "
10600 DATA " 61 4A 53 2A 00 22 4A 28 00 2C 67 28 20 28 00 24 EF 48 EE 88 43 E8 "
10610 DATA " 00 58 22 80 72 00 74 04 61 00 FC A2 2F 08 70 13 4E 41 20 5F 22 81 "
10620 DATA " 72 34 74 04 61 00 FC 90 41 E8 00 18 43 EE 01 40 34 78 00 D4 4E 92 "
10630 DATA " 4C DF 31 F0 34 78 00 C2 4E D2 48 86 E5 0E 45 EE 01 00 24 72 60 00 "
10640 DATA " 1C 2A 00 14 E5 0E 2C 33 60 3E 28 46 4E 75 2A 4B 49 ED 00 18 41 EE "
10650 DATA " 01 00 70 0F 24 18 67 18 24 42 B9 EA 00 10 66 10 B2 2A 00 14 66 0A "
10660 DATA " 4A 2A 00 22 67 08 70 F7 4E 75 51 C8 FF E2 7C 00 1C 01 E5 0E 20 35 "
10670 DATA " 60 3E 67 12 42 B5 60 3E 2F 09 2F 0D 20 40 70 19 4E 41 2A 5F 22 5F "
10680 DATA " 72 00 32 19 5B 41 6F 56 41 E9 00 05 2E 08 DE 81 22 4F 55 8F 2F 0E "
10690 DATA " 9D CE 34 78 01 02 4E 92 2C 5F 32 1F BE 88 66 3E 4A 80 66 3A 3E 01 "
10700 DATA " 67 30 C2 FC 02 04 50 81 70 18 74 00 4E 41 4A 80 66 24 32 07 2B 88 "
10710 DATA " 60 3E 30 07 53 40 30 C0 30 C7 20 FC 00 00 00 40 42 98 20 FC FD 00 "
10720 DATA " 00 00 53 40 6E F6 34 01 70 00 4E 75 70 F4 4E 75 61 00 00 3A 66 2E "
10730 DATA " 57 76 98 00 66 26 2C 36 98 02 02 86 5F 5F 5F 00 06 00 30 70 00 "
10740 DATA " 4E 41 20 68 00 48 45 FA FA C0 B5 E8 00 04 67 0A 20 50 22 08 66 F4 "
10750 DATA " 70 F1 4E 75 21 46 00 26 4E 75 70 0F C0 36 B8 01 53 00 66 20 2F 0D "
10760 DATA " 4B EB 00 08 34 78 01 16 4E 92 2A 5F 66 5C 72 03 D2 76 98 00 08 81 "
10770 DATA " 00 00 D3 AE 00 58 60 4A 70 F1 72 00 32 36 B8 02 6B 42 E7 89 D2 AE "
10780 DATA " 00 18 7C 00 3C 36 18 02 DC AE 00 20 72 00 12 36 68 00 52 81 08 81 "
10790 DATA " 00 00 38 01 54 81 34 78 01 1A 4E 92 22 6E 00 58 DC 44 53 89 1D B6 "
10800 DATA " 68 00 98 00 53 86 51 CC FF F4 53 89 42 36 98 00 70 00 4E 75 "

```

- Mike Mitchell

So how is this any better than a decent editor?

One. Where Quill lets you visualise bold, normal, underlined and high or low script on screen, TEXT87 lets you visualise all type faces including proportional, elite, italics and double-wide through extensive use of red white and green as well as on-screen character spacing.

Two. Numerous fonts are available in the deluxe edition of TEXT87 and there is the facility to design your own fonts. These show on screen and may be expanded up to quadruple height.

Three. Normal printing uses only the fonts available in the printer, but TEXT87 also has a screen dump to print out in graphic mode to capture overhigh characters and fancy type faces.

Four. TEXT87 comes with a number of printer_dat files covering a range of printer types. You select one to be loaded on start-up after configuring a working copy of TEXT87 for your own use. But if you change printers, you can change printer_data from within the program.

Five. Up to six 'frames' on a page are possible. For comparison, Quill has three frames - header, footer and text body. TEXT87 can have four columns in the text body.

You enter the text as a single column as wide as one column on the finished page. At print time, TEXT87 calculates which lines are to be printed and they are printed in columns as in a newspaper.

Thus if the page is 60 lines long, to be divided into 2 columns, lines 1 and 61 are

printed in a single pass with the column space between. Next lines 2 and 62 are printed and so on to the end of the page. Page two has lines 121 and 181 on the first line. And so on to the end of the file.

Not everything is peaches and cream. The 20 page documentation is sketchy on details and doesn't always cover features that suddenly are offered in the prompts.

TEXT87 is rich in defaults but offers no overwrite protection. No file handling is offered within the program. You must revert to SuperBASIC (with CTRL-C) to run a directory or copy or delete files. This is a minor annoyance only because I am so used to the ease of using these features in Quill. In truth leaving out such features lets you access the full potential of any front ends or Toolkits that you have installed.

TEXT87 is not yet easy to use. It has many marvelous features I do not yet understand and have not mentioned here, including the choice of setting parameters in either inches or the metric system! It is tedious to set up your working copy with the limited instructions, and the prompts are sometimes elegant, sometimes maddeningly obscure.

Yet, I have realised that with practice it will become my wordprocessor of choice and I will not begrudge (in time) the six months delay it took to arrive after I had sent my £40.00 (no US\$, no VISA) to Software87, London. UK. £10 more for the multiple fonts and the font modifier. (£1.00 = @ US\$ 1.90).

- Peter Hale

2068 TELECOMMUNICATIONS NOTES

The following is an overview of telecommunications hardware and software available for the T/S 2068 and how it all works.

For modems, the 2068 user has several options. The Westridge 2050 300 baud modem is still available through several T/S dealers, but anyone who hasn't yet purchased a modem should consider a standard 300/1200 baud modem now that software is available to support 1200 baud. The best buy is the Avatex 1200 'smart modem', from Megatronics for about \$85.00. They also have the 1200HC, a Hayes compatible version with the full Hayes command set, for just over \$100.00. Call 1-800-232-6342 with a VISA card or write Megatronics at PO BOX 3660, Logan, UT 84321.

Connecting a 2068 to a 'standard' modem like the Avatex requires an RS-232 interface such as the Z-SIO card, a 3.5 x 5" circuit board that plugs into the rear expansion bus of the 2068. A feed-through connector lets you piggyback other peripherals onto the Z-SIO. The right edge of the card has a standard 25 pin RS-232 connector.

The Z-SIO comes in kit from Ed Grey Enterprises for \$29.95. The number is 213-759-7406 or 213-516-6648, or write PO Box 2186, Inglewood, CA 90305.

The last piece of hardware you need is the cable. Any IBM compatible male-to-male cable will do (for \$12.00 extra, it comes with the Avatex modem).

The software to drive this hardware is a terminal program called Specterm-64 Ver 4.1. There are versions for tape, JLO Safe Disk (V2.3 or higher), AERCO FD-68 disk, and Larken LKDOS system. All work

at 300/1200 baud with the Z-SIO (There is also a version for the WC 2050 300 baud modem). The software features a 64 column display, X-Modem & ASCII file transfer protocols, 31+ K buffer, and a 7K Basic area that may be customized to personal needs. All in all it's a very fast and elegant terminal program.

Another terminal program, ZTerm-64, works in conjunction with the OS-64 cartridge and the WC 2050 modem. It uses the 2068's high resolution video mode so text is displayed with greater clarity than with Specterm-64 (which uses four-pixel wide characters to fit 64 columns onto a 32 column screen).

ZTerm-64 supports full-size printers via a AERCO, Tasman, or A+J printer interface, and it has X-Modem protocol. I am not certain about 1200 baud operation, but it is incompatible with the Z-SIO interface. Zebra Systems was carrying ZTerm-64 for \$29.95. Call (718) 296-2385 or write to 78-06 Jamaica Ave, Woodhaven, NY 11421 for information.

(Note- Larry Kenny of Larken Electronics is working on a version of ZTerm to work directly with the LKDOS System. ZTerm-64 is available for most other disk systems.)

These two terminal programs have made MTERM and MTERM II obsolete. Anyone still using them should consider upgrading to one of the 64 column packages. Since Telenet has begun phasing out most of their 300 baud outdialing modems in favor of 1200 baud, I would recommend Specterm-64 over ZTerm if you use (or plan to use) PC-Pursuit. The benefits are well worth it.

-Jim Rodlin

SuperBASIC for BEGINNERS

PRINT

Continuing from last month's column, we move on to PRINT.

Most of us have a pretty clear idea of what the KEYWORD PRINT does.

E.g. PRINT "HELLO"

sends the word HELLO to the upper left hand corner of the right hand screen.

Correspondingly, a=10: PRINT a will send the value ten to the position just under HELLO.

The use of the word PRINT is sometimes confusing because it does not "print" to a printer. The history of the evolution of the term to apply to screen displays is interesting but not helpful here. At the end of the tutorial we will demonstrate printing to a printer.

Many other keywords are disguised PRINT procedures so we will discuss nine of them.

AT x,y moves the print position to the co-ordinate x,y. Unlike other Sinclair BASICs, AT cannot be combined with PRINT, as in PRINT AT x,y; "HELLO". It must be written:

```
AT x,y: PRINT "HELLO"
```

CLS "prints" a screen of blank spaces and moves the cursor to position 0,0.

DIR prints the name of the medium, the number of free sectors, the total available sectors and the names of all the files on the medium.

INK var prints all subsequent characters in a chosen color where var = 0 to 7.

INKKEY\$ (var) prints the character pressed on the keyboard.

INPUT a\$ prints the string (or a prints the number) entered at the keyboard and assigns it to the variable a\$ (a).

LIST prints the SuperBASIC program currently in memory to the left screen.

PAPER var "prints" the background of the screen a chosen color if followed by CLS.

TO x; when following the word PRINT, moves the cursor to the xth column.

None of these PRINT commands would work if QDOS did not open channels to the screen.

On powerup QDOS opens three channels, numbered zero (#0), one (#1) and two (#2), on the screen and defines them as three windows, the wide, narrow console at the bottom, and the white and red screens side by side above.

All the PRINTing Keywords, except one, are assigned a default channel #1, called the execution window, the right hand one in monitor MODE 4.

The exception for defaults is LIST which has channel #2 as its default. Thus the program can be listed on the left and its output viewed on the right when run.

E.g. PRINT "HELLO" is the same as PRINT#1, "HELLO". PRINT#2, "HELLO" behaves differently.

CLS clears only the HELLO in the righthand window. CLS#2 clears the lefthand HELLO.

Channel #0 defaults to the console window on the bottom and is connected to the keyboard.

When printing to other than a default channel, it is necessary to specify the channel with the "#" followed by the

channel number, followed by a comma. SuperBASIC ignores any spaces you may insert for visual clarity in a listing.

Some Practical Applications

You may view the directories of both drives simultaneously at follows:

E.g. DIR#2, mdv1_ : DIR mdv2_

will display the directory of drive 1 in the left screen and the directory of drive 2 in the right screen.

You may make paper printouts of directories or listings by opening a channel to the printer:

E.g. OPEN#3, ser1 (the printer port)

```
DIR#3, mdv2_  
LIST#3  
CLOSE#3
```

will print the directory of drive 2 to the printer and then will print the current SuperBASIC listing in memory.

Also, if you saved the program, you may

```
COPY mdv2_fred to ser1
```

to print a listing of a SuperBASIC program called fred.

Some Exercises

```
#1 PAPER 4: CLS: PRINT "HELLO"  
#2 PAPER 2: PRINT "HELLO"  
#3 PAPER 6: INK 2: PRINT "HELLO"  
#4 FOR n=0 TO 6 STEP 2  
  PAPER n: INK n+2  
  PRINT n; TO n; "HELLO"  
END FOR n
```

Don't forget to actually practice these few examples so that your hands learn what your brain has just learned.

- Peter Hale

QUERY ON ABACUS

When using ABACUS on the QL I can ORDER the rows on a spreadsheet by selecting a column on which to Order, but I cannot order the columns by selecting a row.

How can I get around this situation? Alternatively, is there any way to rotate the spreadsheet 90° so that the rows become columns and the columns rows?

- Sherm Waterman
S. Yarmouthport, MA

GOING ONLINE, T/S STYLE

Okay, so you have purchased a modem and terminal program for your computer, read the user manuals, and are ready to go. What next? Or perhaps you're wondering what all the fuss is about and you're not sure if you want to go online. Let me describe what you're missing.

The most notable aspect of the world of telecommunications is the extraordinary number of computer BBSes 'out there'. It is estimated that there are over 1,000 BBSes in the United States and Canada and most are privately operated.

Typically a BBS is a small computer equipped with a modem, disk drives and usually a printer. Although features vary between BBSes, most store messages, bulletins, and sometimes programs, that any computer user can access. Most BBSes are also free.

In the Sinclair/TimeX realm, there is no more dynamic, informative, and interesting source of support for your computer than on the various BBSes and online services. They are a continuous source of free software, news, and friendly advice from the other T/S users who already use

modems with their computers.

I recently called the North-Shore Sinclair BBS in Chicago where I downloaded several files including back copies of their newsletter, Nite-Times, and an updated list of Timex/Sinclair BBSes. I read messages left by other users and left a couple of my own including questions on rumors about plans to form a national T/S user group.

I expect someone will leave replies to my messages over the next few days. Meanwhile, I will be calling other BBSes to see what's going on in the rest of the T/S world.

You can reach a lot of people who share your special interests through bulletin boards. I am involved in an effort to organize a special interest group on Artificial Intelligence for Sinclair users. By leaving messages on BBSes to bring attention to the subject, I located others who were interested in joining the new group. By taking a similar approach, you can find information about nearly any interest. BBSes put you in contact with people from all over the country.

"Wait a minute! I can't afford long distance calls to Chicago or Los Angeles every night! What am I supposed to do?"

There is a low cost option, called PC-Pursuit, that lets modem users make UNLIMITED long-distance calls to major cities during off-peak hours, for a flat \$25.00 per month. The one-time registration fee of \$25.00 is waived for BCS members. (Weekdays, between 7 am and 6 pm, PC-Pursuit is \$10.50 to \$14.00 per hour.)

I regularly use PC-Pursuit to call favorite Bulletin Boards in cities around the country.

I keep in touch with people and ideas that interest me for less than the price of dinner (for one) once a month.

For information on PC-Pursuit, call 1-800-835-3638 (689-5700 in Virginia) or write Telenet Communications Corp, 12490 Sunrise Valley Drive, Reston, VA 22096. The PC-Pursuit bulletin board is 1-800-835-3001 (689-2987 in Virginia).

Bulletin boards are one aspect of telecommunications although for many (myself included), they were the reason to have a modem in the FIRST place.

A full description of commercial online services such as Comuserve is beyond the scope of this column, but perhaps another reader will submit an article about CIS or The Source and Sinclair computers (hint! hint! hint!). Till next issue, Keep On Timexing!

-Jim Rodlin
Sysop, BCS TIMEWARP BBS

CALENDAR NOTES

The next General meeting of the Sinclair Timex User Group will be at 7:00 pm on Wednesday, June 15, at the UMass Harborside Campus, room 1-061 Wheatly Building.

There will be no July meeting.

The next QL subgroup meeting will be on Saturday, July 2, at the home of John Mitchell in Westwood. John's number is 326-5420, to call for directions.

FLASH! SIR CLIVE SINCLAIR Will be the featured speaker at the June 22 General meeting of the Boston Computer Society.

See the new 288 laptop in person at the New England Life Hall starting at 7:30 pm.

SING A SONG OF SILLINESS

(From the D-FW DATA Expansion, the journal of the Dallas Fort Worth T/S User Group, citing 'The Prairie Home Companion' as the source)

To the tune of 'A Modern Major General' from 'The Pirates of Penzance'

I've built a better model than
the one at Data General
For data bases animal and
vegetable and mineral.
My OS handles CPUs of multi-
plex duality.
My PL-1 compiler shows impres-
sive functionality.

My storage system's better
than magnetic co-polarity.
You needn't even bother check-
ing out a bit for parity.
There isn't any reason to
install non-static floor
matting.
My disk drive has capacity for
variable formatting.

I feel compelled to mention
what I know to be a gloating
point;
There's lots of space in
memory for variables floating
point.
In short, for input vegetable,
animal and mineral
I built a better model than
the one at Data General.

The IBM new home computer's
nothing more than germinal.
At Prime they still have
problems with an interactive
terminal.
While Tandy's done a lousy job
with operations boolean,
At Wang the byte capacity's
too small to fit a Coolie in.

Intel's major finances are
something of the troubled
sort.

The Timex/Sinclair crashes
when you implement a bubble
sort.

All DEC investors soon will
find they haven't spent their
money well.

And need I even mention Nix-
dorf, Univac and Honeywell?

By striving to eliminate all
source code that's repetitive
I've brought my benchmark
standings to results that are
competitive.

In short, for input vegetable,
animal and mineral
I built a better model than
the one at Data General.

In fact I've a Winchester of
minimum diameter,
When I can call a subroutine
of infinite parameter,
When I can point to registers
and keep their current map
around,
And when I can prevent the
need for mystifying
wrap-around.

When I can update record
blocks with minimum of
suffering,
And when I can afford to use
100k for buffering,
When I've performed a matrix
sort and tested the addition
rate,
You'll marvel at the speed of
my asynchronous transmission
rate.

Though all my better programs
that self-reference
recursively,
Have only been obtained
through expert spying done
subversively.
Still for input vegetable,
animal and mineral
I built a better model than
the one at Data General.