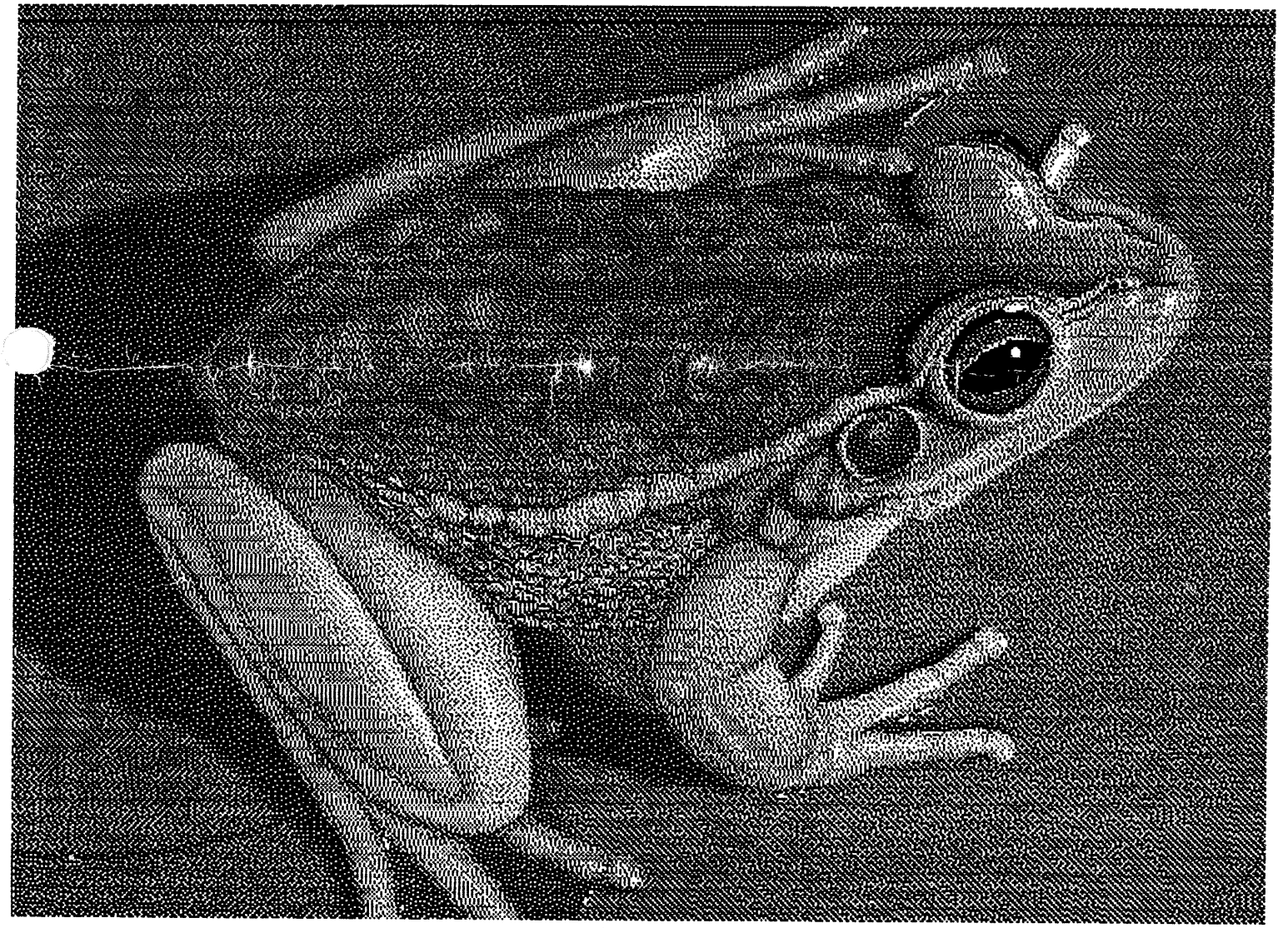
SINC-LINK

MAY-JUN '91 VOL.9 NO.3



TORONTO TIMEX-SINCLAIR USERS CLUB

SINC-LINK VOL.9 NO.3

MAY-JUN '91

SINC-LINK IS A PUBLICATION OF THE TORONTO TIMEX-SINCLAIR USERS CLUB AND IS ISSUED 6 TIMES A YEAR. COPIES OF THE NEWSLETTER ARE $1.50 EACH FOR NON-MEMBERS. CLUB MEMBERS RECEIVE FREE COPIES AS PART OF THE $20.00 ANNUAL MEMBERSHIP FEE. A NEWSLETTER SUBSCRIPTION ONLY IS AVAILABLE FOR $12.00.

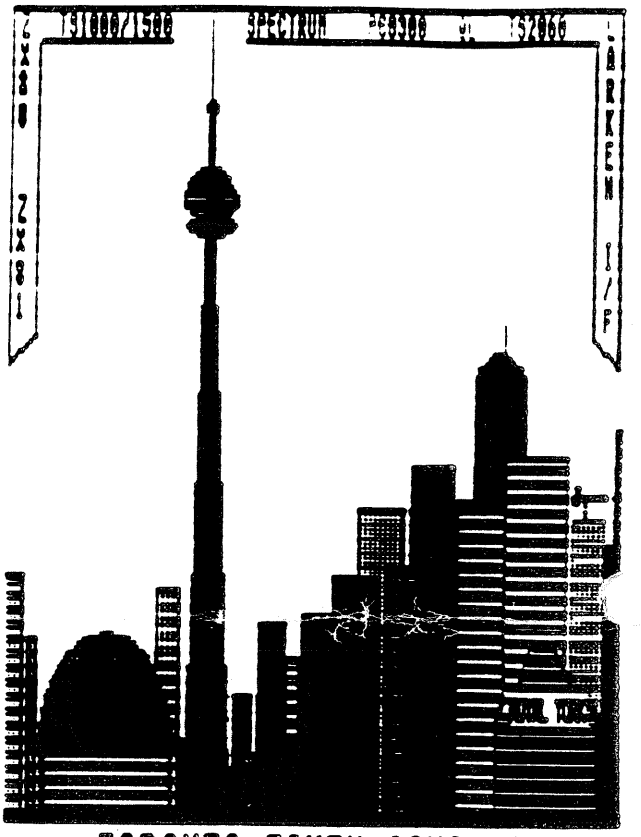
NEWSLETTERS ARE EXCHANGED, FREE OF CHARGE, WITH OTHER TIMEX-SINCLAIR USERS GROUPS.

PLEASE CREDIT THIS PUBLICATION AND THE AUTHOR IF YOU COPY MATERIAL.

THE CLUB MEETS ON THE FIRST WEDNESDAY OF EACH MONTH AT FOREST HILLS COLLEGIATE INSTITUTE, 730 EGLINTON AVE. W., TORONTO. START TIME: 7:00 PM.

SEND CORRESPONDANCE TO:

Attention: SINC-LINK EDITOR
TORONTO TIMEX-SINCLAIR USERS CLUB, 14 RICHOME COURT, SCARBOROUGH, ONTARIO, CANADA M1K 2Y1



TORONTO TIMEX-SINCLAIR USERS CLUB

EXECUTIVE OFFICERS:

	(Area Code 416)
PRESIDENT:	RENE BRUNEAU (531-9749)
TREASURER:	BILL LAWSON (444-8772)
SECRETARY:	GEORGE CHAMBERS (751-7559)
ACTIVITIES:	RENE BRUNEAU (531-9749)
TAPE LIBRARIAN ZX81:	LYMAN PAQUETTE (482-4479)
TAPE LIBRARIAN 2068:	RENATO ZANNESE (635-6536)
TAPE/DISK LIBRARIAN QL:	HUGH HOWIE (634-4929)
PAPER LIBRARIAN:	JEFF TAYLOR (244-8583)
NEWSLETTER:	JEFF TAYLOR (244-8583)
LIAISON OFFICER: (Out-of-town members)	GEORGE CHAMBERS. 14 RICHOME COURT. SCARBOROUGH. ONTARIO. M1K 2Y1 (416-751-7559)

TORONTO TIMEX-SINCLAIR USERS CLUB

Editorial

I really must thank a few of the club members for their kind words about my third anniversary as newsletter editor. It's very gratifying to hear from our brothers and sisters within the T/S community and I can assure you that your letters and cards have certainly bolstered my resolve to continue to try to put out a quality product for you.

So, to Joan K., Les C., Donald L., Larry C. and all the members who were so complimentary, my heartfelt thanks.

I can't take all the praise for Sinc-Link's success, though. While I do the assembling, the cover and occasionally an article, it is the continual high-quality writing of you active members that makes this newsletter the fine publication it is today. All you members who have contributed over the last three years and all you writers whose work we have culled from other newsletters because we felt it was worth repeating, take a bow. Alright, enough mushy stuff.

MSCRIPT 5.5 & TASWORD 22M

I recently received the MSCRIPT update from Jack Dohany (no E) and a TASWORD 2 variation from Larry Crawford and am now trying to learn all the features these two prolific programmers have built into their versions. This editorial is a combination of both programs. Can you spot where they start and stop?

More TOS Info

In the last couple of issues I'd written about my exploits in repairing one of our TOS/ZEBRA/English Micro Disk sytems and a request for information on this disk system. Well, I'd like to thank Joan Kealy for offering to send me her backup and in particular I'd also like to send my thanks to James Wilson who took it upon himself to assemble and send me a great package of material on this subject. If more material comes in I'd be glad to exchange what I've received with interested members. Drop me a line and let's keep supporting another T/S utility.

ZX81/TS1000 Hi-Res

Last issue I wrote about the newest additions to the ZX81 library, namely two tapes full of hi-res programs courtesy of Greg Harder. The demand for these tapes has been quite surprising. Members that we didn't even know had ZX81s have been asking for copies. Bear with us and we'll get those tapes and manuals to you. It just takes time to make good quality copies.

Digitised Pictures

In last issue there were a few digitised pictures of yours truly. I am now running a closed circuit tv camera through my SMUG digitiser and using John Mcmichael's VIDEOTEX software. Watch for a review of his updated program and more digitised pictures next issue.

That's all for now...

J.T.

BOB'S NOTEBOOK - Programming Tips

AN ALTERNATIVE TO NMI BUTTON PLUS THE F KEY

Here is an example of how to use a rather obscure LKDOS Extended BASIC Command. It concerns that pesky line of dots that appear at bottom of the screen whenever the NMI push button is pressed. Those dots, which actually represent the registers and stack pointers involved, clutter any graphics copies which print the bottom two lines. However, close reading of the LKDOS Operating Manual turns up an interesting item: Advanced machine code programmers can add an extra command to LKDOS by using the PRINT #4: DATA a,b,c...command. Moreover, more reading reveals another clue from the manual: LKDOS address 8214 "Userad" holds the address of the user NMI function *OR the address of the User command (DATA)*.

If this could be used to trigger a graphics-copy machine-code routine without pressing the NMI button and the F key, then it should be possible to make an uncluttered copy on a wide printer, that is, without those dots. Graphics 24 is one such routine which copies all 24 lines of a SCREEN$ and is found in the Omnibus Collection where it loads into the LKDOS at address 16100. Thus, it ought to be possible to substitute the LKDOS DATA command for the NMI button. On investigation, this turned out to be the case. The following lines, which appear in the Omnibus program, were involved in making the DATA command work:

```
1020 INPUT AT oo,oo;(u$'"drive? (0-4) ");drv: PRINT #od: GO TO
drv: RETURN
1050 PRINT #od: POKE oz,16100: LET b=USR oy: RETURN
2570 PRINT AT oo,oo;"Standby...": GO SUB 1050: IF b<>205 THEN
FOR i=24605 TO 24788: PRINT #od: POKE i-8505,PEEK i: NEXT i
2580 PRINT #od: POKE 8214,16100
2590 CLS: PRINT AT oo,oo;"CHECK Printhead at left          each
time.": PAUSE 100:RETURN
5380 GO SUB 5410: ON ERR GO TO 5390: INPUT AT oo,oo;"  COPY
SCREEN$  "'" name? (<=9) "' LINE s$: GO SUB 1020: GO SUB 5500:
PRINT #od: LOAD s$SCREEN$: PRINT #od: DATA 0
5390 ON ERR RESET: GO TO m3
5410 CLS: GO SUB 1050: IF b<>205 THEN GO SUB 2570: RETURN
5420 RETURN
```

Explanations:

Line 1020 is a sub routine to choose the drive. (u$ is used throughout the program to hold the name of a chosen option).
Line 1050 does a double poke into LKDOS RAM 8200 (variable oz) to get the value in LKDOS address 16100.
Lines 2570 to 2590 check that value and if it is not 205 then it reloads the contents of system variables free space 24605 to 24788 into the LKDOS RAM at address 16100 (24605 minus 8505); 2580 puts the value 16100 into LKDOS address 16100 and finally 2590 puts a reminder on-screen before returning.
Line 5380 checks that all is ready via 5410; if there is an ERR message (see below), the action jumps to line 5390 and thence back to the menu.
Don't leave out 5420.

The crux of all this is the part at the end of line 5380: PRINT #od: DATA 0. Not knowing what to add after the DATA token, I stuck in the zero. However, no matter what was put in there, an error message resulted each time so ON ERR commands were added to overcome the error. Result: the routine worked and the line of dots was history. By the way, the NMI button + F still does its job when these changes are made, so it's always there as an alternative.

One postscript to this story: I later had a go at making the autoboot code, stored at 16300, work in a similar fashion. During a lot of experimenting a funny thing came to light: the command line had to be <PRINT #4: DATA 0>; the computer crashed (locked up) if <RANDOMIZE USR 100: DATA 0> was used. (I haven't the foggiest notion why. Can anyone help explain this?) So this means that to quit any program and get back to Omnibus, the PRINT #4 switch has to be operative in that program.

SOME MORE PROGRAMMING TIPS

PROFILE: When altering the BASIC program, be careful not to declare any variable ahead of d$. This string must be the first variable in the VARS area or the program simply will not work.

PRINTER CONTROL CODES: There are two distinct methods of sending these to your printer. The FIRST one often used in programming, uses OUT commands to do the job. For example, to switch to ELITE typeface requires OUT 127,27: OUT 127,77. BUT, there must be a check on printer readiness before the commands and this can vary from printer to printer and even from one TS2068 to another. Thus it is necessary to make the following:<GO SUB line#> prior to each OUT command. The "line#" is the line containing a routine such as <IF IN 127<>237 THEN GO TO line#>; line# +1 must be <RETURN>.

The SECOND one, perhaps not much used, is really more straight forward: This assumes you are using the LKDOS printer driver and depends on first switching off the detokenizing and filtering (by POKING PPAS at LKDOS RAM address 16093 to 32) which then causes all subsequent characters to be sent directly to the printer. PPAS is POKEd with a zero at the close of sending these printer commands to return to detokenizing and filtering. Thus the subroutine to turn on ELITE typeface becomes:

```
RANDOMIZE USR 100: POKE 16093,32
LPRINT CHRS 27; CHRS 77
RANDOMIZE USR 100: POKE 16093,0
RANDOMIZE USR 100: POKE 16094,marg: LPRINT: RETURN
```

No printer ready check is needed but note that the margin setting must be repoked since the double poke to 16093 upsets it. So if you find some of the first style printer commands cause you problems with your printer and/or TS2068, switch to the second kind.

MORE ON PRINTER CONTROL CODES

I've recently added the following to both my Tasword and Mscript lists.

First TASWORD: This assumes you have Graphics editing capabilities in your copy of Tasword. British Font ON (27 82 2) OFF (27 82 0): these could be different for your printer; check your manual. I put these on key 6 (137 and 134 respectively). The main use for this font is to get the pound sign to print instead of a # sign. But the application has other possibilities: If you write in French or Spanish and these fonts are included in your printer's memory, substitute them to get the necessary accented letters.

Another addition (this time on the 5 key -- 138 and 133)) was a quick switch from standard line spacing to narrow line spacing and vice versa. The commands will vary from printer to printer but for the Fastext 80 they were Narrow ON (27 65 8) OFF (27 50). This allows for packing more text into a page or part of a page.

Now MSCRIPT: You will need instructions for LCODES provided with Mscript version 5.5 to do this. The codes above or your equivalents will be added to the LCODE text. I put the line spacing against capital A for ON and small a for OFF. British font ON against capital B and OFF against small b. I see that the French font could go into capital F and small f. Mscript has lots of spare letters available for additional commands whereas Tasword is quite limited.

KEYBOARD IDIOSYNCRASIES

Using INKEY$ for keyboard input to any program invites some troubles since the ROM Debounce and Error Detection routines are not used. Debouncing is the conversion of mechanical contact bounce into a clean transition between two logic states. It involves the disabling of the software's ability to accept a character for a few milliseconds after receiving one. This prevents false input or extra signals from a keyboard due the bouncing of the keys. Keys sometimes tend to bounce due to inherent design or the keyboard being old or heavily used.

If you have trouble with INKEY$, it may be wise to substitute some BASIC lines that call on the ROM LAST K routine which does

use the ROM debounce and error detection routines. These will involve PEEKing system variable 23560 (LAST K) and will require some lines like the following (found in OMNIBUS):

```
3140 PAUSE oo: LET lk=PEEK 23560
3150 IF lk=13 THEN GO TO 3120
3160 IF lk>=48 AND lk<=52 THEN LET drv=lk-48:
PRINT #4: GO TO drv: PRINT AT 18,29: FLASH 1;
drv: GO SUB bp
3170 IF lk<97 OR lk>122 THEN GO TO 3140
3180 GO TO 3180+(lk-96)*10
```

To explain:

3140 assigns the value in address 23560 to variable lk.
3150 to 3170 put some conditions on lk: if it is 13 (ENTER), then go back to start of the input routine; if it is between 48 and 52 (keys 0-4) then select drive and show this number on-screen (bp is the variable in Omnibus for the BEEP subroutine; if lk is outside the values for keys a to z, then try again.
3180 if the input survives all these conditions, then go to the line number indicated by the formula.

There's more to be read about this subject in Update Apr 89 p.12 and Apr 88 page 20.

Bob Mitchell Willowdale Ont
Dated: 910223

WOOD AND WIND COMPUTING

A NEW SOFTWARE RELEASE !!!

QLERK : A powerful accounting program for the QL $75.00 on 2 disks

QLERK is an ARCHIVE based accounting program for home and business. It includes payable, receivable, and general ledger modules integrated with my address database program, DBAddres. QLERK has check writing and invoicing built in. It even keeps track of your VISA, Master Charge, or other charge cards. Bank balancing is easy with QLERK's help. You can display all your checks and deposits and compare them with your bank statements. You don't have to use a predefined account coding system. You can use the one you want. When you run the general ledger, each account is individually tallied making income tax time a snap. It makes a general ledger export file for ABACUS.

I have developed QLERK over the past 3 years working with several businesses. I have many,many hundreds of hours of development work in this program. I have been careful not to release it too soon but now feel it is ready. At $75.00 QLERK is a pretty expensive piece of software but for those who want to keep track of their finances it has no equal on the QL and will quickly pay for itself. It has many features that more expensive PC accounting software like Peachtree and DAC Easy fail to include like adding new vendors, customers, accounts or products on the fly. The address database features are second to none for being able to target subgroups for mailing labels or other purposes. A PC version of QLERK using PC ARCHIVE is also in the works.

Don't send money for QLERK. To order QLERK first send a request for a more detailed description and special order form that will ask for necessary information. Each purchaser of QLERK will get a custom copy of the software with their home or business embedded in the code.

Bill Cable : RR3 Box 92 : Cornish, NH 03745
603 675 2218

Q L I P S

By Hugh H Howie

GARBAGE: ---------- GETTING RID OF IT
One night recently I had a phone call from a QL'er, and during the process of yakking, he mentioned he was having a small problem with loading some programs with long names. The problem was not so much that he could not load the program, but that he hated having to type in all those letters in front of the File Name,

I was able to tell him there was a program in the QL Library called DIR_EDIT_BAS, by Howard Clase, that would do an excellent job for him, I also said I could remember giving him that program some time back. Later that week my friend said the program was really wonderful, and had saved him a lot of time in getting rid of a lot of garbage.

I had a couple disks given to me some weeks ago but had never done much with them, as each and every title started off with 'List1' or 'List2' so that every time I started to work with them I had to type in those extra letters. Too much work. So last night I thought of trying this program on them, but the thought came to me that perhaps there could be an even easier way to my job. There is! Most of us have it right at our fingertips. ---- TK2.

Section 5 of TK2 tells you how it is done. But is somewhat obscured as it is tagged on at the end of the section, tucked into half-a-dozen lines at the very end. All it says is that "Renaming is a process similar to COPYing a file" So I went after the COPY. And those extra bits tagged on at the front of a file name, can be swept away in the blink of an eye. I mean that. Very quickly. Like FAST man!

Say you have a disk on which every title starts off with "LIST1". It will look like this:-
Listl_Garbage
Listl_MoreGarbage.
Listl_etc

Put the disk in shall we say, FLP1, and type:- WREN FLP1_LIST1_FLP1
(note comma) ↗
Just look at that and remember it, as one day you are going to use it.

Press [ENTER], and watch the "LIST1" go.

§

TURTLE-GRAPHICS----------(TERRAPIN?)
I keep looking for little things for you to punch in for yourselves, to give you something to do, and also give an insight into the various aspects of programming.

This time it is a small fancy little snippet of a Turtle Graphic. Not quite as slow as a turtle, but it sure has more twists aad turns than a turtle. A bit faster too, but the patterns are interesting. So why not sit down with me and have bit of a rest from all that hard stuff.

This came to me with a lot of other stuff, some good, some not so good. Some I don't know whether it is good or bad as it does not work for me. I recently got Minerva, and I understand that some of the earlier material does not work on it. I still am trying to learn about Minerva. But she sure is one COOL lady. The QL really is a lot cooler.

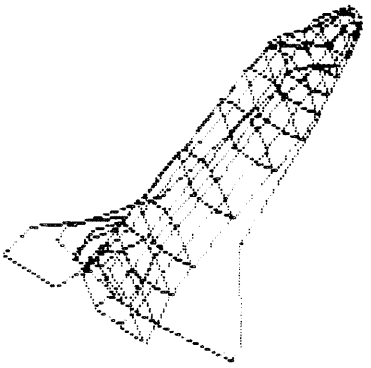
Here is the little listing for you to type in. I can send it to you, but it is cheaper this way.

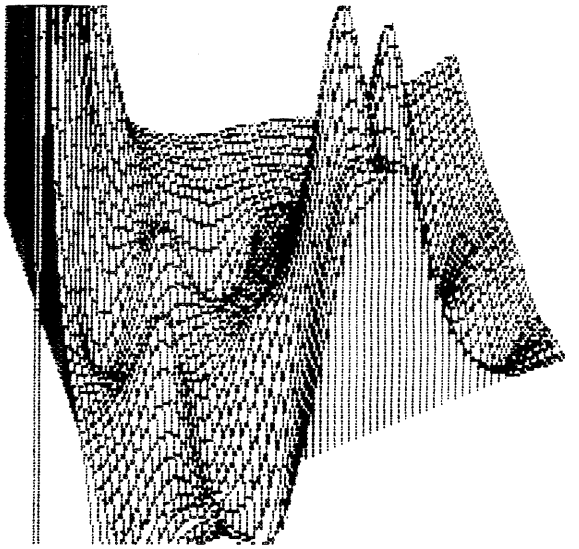
```
100 REMark Turtle graphics
110 WINDOW 512,256,0,0
120 PAPER 0:CLS
130 PENDOWN
140 REPeat logo
150   IF RND>.68 THEN CLS
160   note=RND
170 POINT 60,60
180   INK RND(1 TO 7)
190   turtle=RND(3 TO 70)
200   terrapin=RND(25 TO 340)
210    FOR k=1 TO turtle
220     MOVE k/.7
230     TURN terrapin
240     BEEP 3200,k/note
250   END FOR k
260 IF INKEY$<>"" THEN STOP
270 END REPeat logo
```

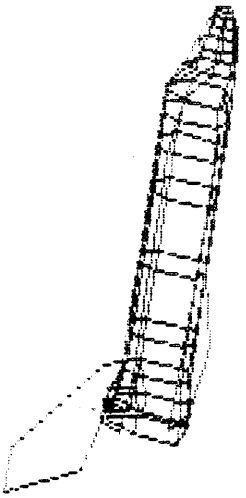
Now was that not fun? Would you like another? Well space is getting a wee bit short so we will leave it off for another time.

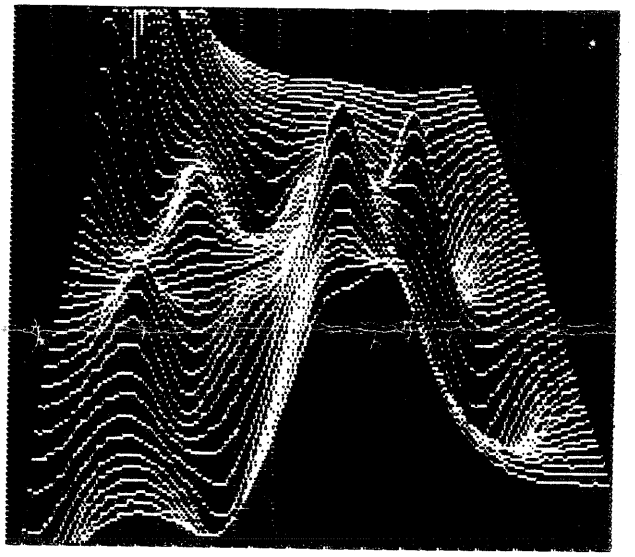
In the meantime, 'night all, and have pleasant dreams of circles flashing round in front of your eyes. 'Bye §

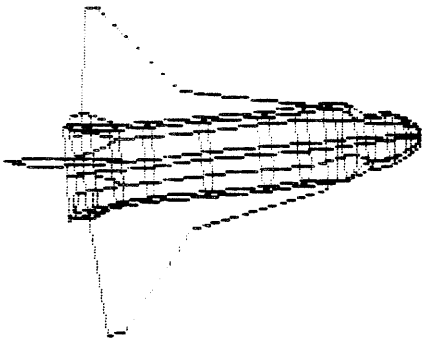
MORE ZX81/TS1000 HI-RES PICTURES

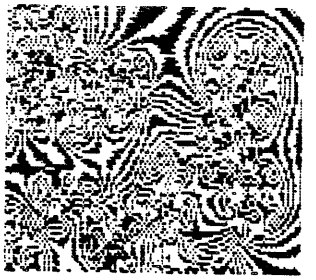












NOTE: YOU MUST HAVE NON-VOLATILE MEMORY LIKE THE MODIFIED HUNTER BOARD, THE SCRAM BOARD OR THE ON-BOARD STATIC RAM BOARD PLUS EXTENDED MEMORY TO RUN THESE PROGRAMS. CONTACT THE CLUB FOR MORE INFO.

PUBLIC NOTICE REVISED

It's true that I'm trying to get away from the retail business side of things, but my intention is to make things EASIER for folks, not harder! When it comes to filling orders, I'm often the slowest of the slow. In theory at least, a dealer has nothing to do except fill orders, whereas there are often long periods of time when I cannot fill orders due to the press of other things. However, if a person is unable or unwilling to get something from a dealer, that person is welcome to try me. Here's what I'd like:

1. Please state clearly what it is that you want.
2. Please describe your hardware completely.
3. Please include SASE with your order.
4. Please write concisely and don't ask a lot of questions.
5. Send no money. I will include bill with shipment.
6. Be prepared to wait; I am sometimes VERY slow.

I will use your SASE to acknowledge your order. I'll let you know if your order will take longer than 30 days, or if I'm unable to fill your order. If you don't like what you get, or find the price too high, you can return it. Returns are always welcome. If what you ordered exceeds your budget limits, you can pay for it over a period of time to suit your budget.

Here's what I'm currently working on:

1. A dealer price list... very complex, trying to simplify.

2. A CONFIG program which will enable the user to easily configure WORD MASTER and TYPELINER for just about any dot-matrix printer. Adaptable to other software which is fussy about printers.

3. TECH DRAW V3 (now completed) for any disk system. For mouse, trackball, joystick or keyboard.

4. MOUSE DRIVER for use with Zebra A/D + Radio Shack Color Mouse. Nearly done. Will be quite nifty.

5. A LOADME program which will enable the user to easily load and print documentation files on disk in Mscript or Word Master format, without owning or knowing either word processor. Will be provided free on all disks containing docs in MS/WM format.

6. DEVPAC documentation, long overdue.

7. An improved version of SPECTERM64 which will include support for the AERCO dual-channel RS232 interface.

8. A new and improved version of the AERCO RS232 interface.

Jack Dohany
390 Rutherford
Redwood City, CA 94061

April 5, 1991

SuperBasic Tutorial Part 2
Printing in Columns

In the first episode of this exciting serial I showed you how to structure a simple program so that it could easily be expanded and improved without loosing track of where you are. All we have to do is to consider a statement, think how it might be improved, write the improvement as a FuNction or PROCedure and call it by name from the original statement. The program was one to print the directory of a storage device (disk or mdv cartridge) onto paper that had appeared in an earlier issue of Sinc-Link. The first modification we made was to give the user the option of either accepting program's suggestion for the source of the directory or putting in one of your own without having to stop the program and EDIT a line.

I left you with the problem of altering the program so that the default offered doesn't always revert to the one built into the program in the PROCedure "Set_up", but uses whatever device you used last time. There are two ways to do this, one is to modify line 22Ø (not 21Ø - my mistake) as follows:

```
22Ø dev$=Get_dev$(dev$):DIR#3,dev$
```

The first statement can change the value of the global variable dev$, which couldn't happen in the earlier version. The other way would be to change the dv$ variable to df$ throughout the FuNction Get_Dev$, this means that df$, an alias (formal variable) for dev$ within the function, could be altered and dev$ along with it. But I don't favour this way of doing it, it's a bit too mysterious!

Another point struck me while re-reading what I wrote last time: some beginners might be puzzled about the use of CHR$(1Ø) and CHR(12) in the program. These are ASCII control codes which cannot usually be typed in from the keyboard; CHR$(1Ø) is an exception, it's the <ENTER> key, also known as line feed (LF), since that's what it does both on the screen and on the printer. (You can also generate it with <CTRL J> - try it!). CHR$(12) is form feed (FF) and does nothing on the QL screen, but has a noticeable effect on the printer - it moves to a new sheet of paper at the end of the run. (On some micros it cle the screen - but not this one!) One other code worth mentioning here is CHR$(13), carriage return (CR), which on the QL screen is implied in the LF, and probably on your printer if you have set the DIP switches to their most useful settings.

If you have tried our program as it stands you will probably have noticed that it only prints one name on each line, and since file names are generally rather short this is a shocking waste of trees. Wouldn't it be nice to print it in several columns across the page, like, for example, Quill does on the screen when you ask for a list of your _docs? That is what we are going to look at in this episode. The first thing we have to do is to identify the line that we need to alter, i.e. the one that actually instructs the computer to send the directory from the mdv tape or floppy to the printer. We have to look no further than line 22Ø; it is the DIR#3 command that does it. We have intercept the stream of file names tha this command spews straight down the serial port. This is where the QL's flexible handling of channels come in handy.

Aside on channels.

A computer would be useless unless you could put information into it and get something back out - it needs to be able to communicate. Information can be put into the computer from the keyboard, from a file on tape or disk, via the net etc. Output can be obtained on the screen, on a printer or it can be put into a file. Some computers have different commands for dealing with different kinds of devices (e.g. PRINT to the screen, but LPRINT to a printer), but to the QL they are all just numbered channels. The commands you use are the same irrespective of the device at the other end of the channel. (Although some are inappropriate - like CLS to a printer channel.) When you open a channel you have to tell QDOS what kin of input/output device it is, and it

sets up the appropriate "driver" to look after the details for you.

Back to our columns.

The DIR command directs the list of file names to a numbered channel, so we can intercept it if we make the channel a file rather the printer. Then we can read it back from the file into an array, and get full control of the list of file names as we print them out. Printing them out at regular intervals across the page is easy, but proper printing in columns involves going down each columns in turn from left to right; and that is what we shall do - phone book order is what I am aiming at.

The listing contains all the lines you will have to alter or add to Prt_dir2 to upgrade it to Prt_dir3. It is not a complete program in itself; it will not run by itself. Type in the listing, and save it under some temporary name (e.g. mdv1_Prt_dir_UG). Then load Prt_dir2 and MERGE in the upgrade. You must do it in this order since lines with the same line number are overwritten during a MERGEr, and we want the later versions to stand. This assumes you have the same line numbers as the listing published in part one; if you have RENUMbered, then you will have to be very careful and make the alterations by hand.

Upgrade Listing

When merged with Prt_dir2 these lines convert it to version 3, do not try to RUN them on their own!

```
100 nm$ = "Prt_dir3"
110 REMark hjc 1991.03.13  Ver 0.33

162   temp$='ram1_temp_dir'
164   chars%=80:    REMark output line
166   cols%=4

220     Col_dir 3,Get_Dev$(dev$)

630 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
640 DEFine PROCedure  Col_dir(ch%,dv$)
650 LOCal a$,nl%,i,n%,st$(2,30),rows%
660 n%=0: nl%=chars% DIV cols%-1
670 Make_array dv$
680 PRINT#ch%,st$
690 PRINT#ch%: FOR i=1 TO rows%
700   FOR j=0TO cols%-1
710     a$=ddir$(i+j*rows%)
720     PRINT#ch%,!Procrust$(a$,nl%);
730 END FOR j: PRINT#ch%: END FOR i
740 END DEFine
750 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
760 DEFine PROCedure   Make_array(dv$)
770 LOCal a$,c%,i: c%=ch%+1
780 REMark Put directory into file
790 DELETE temp$: OPEN_NEW#c%,temp$
800 DIR#c%,dv$: CLOSE#c%
810 n%=Line_count(c%,temp$)-2
820 rows%=n%/cols%+.499
830 n%=cols%*rows%
840 DIM ddir$(n%,nl%)
850 OPEN_IN#c%,temp$
860 FOR i =1,2:INPUT#c%,st$(i)
870 FOR i=1 TO n%
880   IF EOF(#c%): EXIT i
890   INPUT#c%,ddir$(i)
900 END FOR i
910 CLOSE#c%: DELETE temp$
920 END DEFine
930 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
940 DEFine FuNction Line_count(ch%,f$)
950 LOCal a$,ct,n%:n%=0
960 REMark Count no. of lines in file
970 OPEN_IN#ch%,f$
980 REPeat ct
990   IF EOF(#ch%): EXIT ct
1000  INPUT#ch%,a$: n%=n%+1
1010 END REPeat ct
1020 CLOSE#ch%
1030 RETurn n%: END DEFine
1040 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
1050 DEFine FuNction  Procrust$(r$,p%)
1060 LOCal l%: l%=LEN(r$)
1070 IF l%>p%
1080   r$=r$(1TO p%)
1090 ELSE r$=r$&FILL$(" ",p%-l%)
1100 END IF
1110 RETurn r$: END DEFine
```

(This series will be continued in the next issue)

(c) 1991 H.J.Clase

Howard Clase

(709) 753-6415
Box 9947, Station B,
St John's, Nfld, A1A 4L4

email hclase@munucs.ucs.mun.ca (Note

Printing in Columns

First we need to bring the version info up to date (100 & 110). Before we get to the main additions it is necessary to define a few more global variables, and the best place to put these is into the PROCedure Set_up (lines 162, 164 & 166). Temp$ is the temporary file we shall have to set up to hold the directory. If you have ramdisk use it here, if not replace ram1_ with any device (flp or mdv) that will have a bit of space on it. Using a variable in this way means that you only have to alter the name once to change it thoughout the program. Chars% is the number of characters in the line of the output device; this is normally 80, but to save paper while I am testing the program I use a screen for channel #3 which is only 37 characters wide. (This is an example of the usefulness of the QL's channels; all I have to do is to alter the definition of channel #3 in line 160 to switch from screen to printer.) Cols% is the number of columns required, you can set it to whatever suits you. Later we might add a procedure to enable the user to chose the number required while the program is running.

I've designed my new PROCedure to look as much like the QDOS PROCedure it replaces as possible, so it requires the same two parameters, a channel number and a device name (or, in this case, a function that returns one). (You cannot use the # notation for your own superbasic PROCedures; it has to be replaced with a space.) The main change to the original listing occurs in line 220. (You can use either version - see above) This change leads into the sequence of two new PROCedures and two new FuNctions that follows.

Col_dir is the PROCedure that deals with printing in columns, but there are a few preliminaries to go through first. Make_array puts the directory into an array, ddir$. Lines 790 & 800 OPEN a new file as channel #c%, and puts the directory of your chosen device into it. Not surprisingly QDOS objects if you try to OPEN_NEW a file with a name that alredy exists, and the program would stop with a QDOS error message if it were not for the protective DELETE statement - luckily QDOS does not object to deleting a file that doesn't exist! DIR produces a series of ASCII text 'lines' each followed by a LF characte. (CHR$(10)), so that is what the temporary file consists of. The first two lines of the directory of a formated device are always the device name and the sector statistics. (Well not quite always, I have a version (2.02) of Toolkit 2 with a bug in it so that it fails to produce these two lines from an empty mdv tape, and the program as written stops here if there are no files on the tape. It will, of course always stop if the tape is not formatted.) The rest of the file is a list of the names of the files on the tape or disk. Since there may be anything from 0 to about 450 we have to count them. This is done at line 810, n% is the actual number of files - two less than the number of lines found by function Line_count, which I'll explain later.

Then follows a bit of fudging. Since I want to print in proper columns the array has to be exactly the product of the numbers of rows and columns. Suppose there were 98 files and we ask... for four columns, we can only print rows, so the first line would have to consist of files number 1,26,51 and 76 and so on, but when we get to the last row but one we should run out in the last column if we had only dimensioned our array to 98, generating an "out of range" error. Line 820 calculates the number of rows - the .499 makes sure it rounds up not down - 830 adjusts the value of n%, and 840 dimensions the array ddir$() to n% strings of maximum length nl%. This length was calculated at line 660 - the -1 ensuring at least one space between the columns.

Dimensioning Arrays

Because of the way it was dimensioned ddir$ is a global array. I should have preferred to make it LOCal to Col_dir, but this couldn't be done since we don't know how big it has to be when we enter the procedure. Variable arrays have to be dimensioned to help QDOS organise t... memory. Since it is a string array yo... have to tell it how many elements (file

names) and the maximum number of characters you want in each - if you try to assign a longer name then only the first nl% will go into the array. A LOCal array is declared by putting the dimensions in brackets after the variable name in the LOCal statement, which must be the first line in the array or function. The variable st$ declared in line 65Ø is an example of a LOCal array, this is to hold the device name and the sector statistics. Since we know in advance what size this has to be there is no problem, but we cannot dimension ddir$ until we know how many file names there are, and it has to be done with a DIM statement. SuperBASIC is more flexible than most BASICs in that you can re-dimension an array to be larger than it was before, so there is no problem with the fact that it will be re-dimensioned each time you print a directory.

Having dimensioned our array we have to fill it with the file names. Line 85Ø re-OPENs our file for INPUT to the computer as channel #c% again. (The OPEN command also allows you to alter or add to the file, but we don't need to do that.) The short form FOR loop at line 86Ø puts the drive name and stats into st$. I could have just used the first two elements of ddir$, but I didn't want them to be truncated. Note that the INPUT command - with the appropriate channel number - is used, the same command you use to get input from the keyboard. Then we move to a long FOR loop which reads the file names from the temporary file and puts them into the array. Because we may have increased the value of n% to be greater than the actual number of names we have to allow for getting to the end of the file before we get to the end of the loop; this is done at line 88Ø. The last few arrayelements may remain empty. Then we've finished with the file and can dispose of it (91Ø).

We now return to line 68Ø, which demonstrates a useful trick. If you want to print a whole array you do not have to write your own loops, PRINT array_name will print out the whole thing. If you follow the array_name with one of the formatting "punctuation marks" (, ; !) it is printed as though each printed element is followed by the mark; a comma will tab each element - try the effect of the others for yourself. Since there is nothing in line 68Ø the two entries are printed as separate lines. Then we move to the main double loop (69Ø - 72Ø). The outer (i) loop controls the rows and the inner (j) loop picks out the entry for each column as we go across the row. The ! puts a space before each column except the first, and the procrustean FuNction ensures that each is exactly the right length. (The formatting mark ! is not used often enough by amateur programmers, as well as the above role it will not print at the end of a line if there is not enough room for the whole string, but starts a new line - no split words.) The PRINT statement between the two END FORs makes sure that there is a new line after each row, it is redundant if you have got the characters per line right in Set_up, but helps keep things tidy if you haven't.

There are still two FuNctions left unexplained. Line_count just returns the number of lines in an ASCII text file, you can use it, for example on a basic file - try:

```
PRINT Line_count(4,"mdv2_Prt_dir3")
```

(you must have prt_dir3 loaded at the time - or Line_count at least, and don't forget the quotes!. It won't work on a Quill file they are not ASCII.) Note in particular line 99Ø. Whenever you use a loop to run through a file, since you do not usually know the length in advance you have to use a REPeat loop rather than a FOR loop, and you must test for end-of-file right at the beginning - it might be an empty file. Nothing happens to a$ here, it's just somewhere to put each line while you count it. You could easily adapt this FuNction to find the length of the longest file name in the list as well as count them couldn't you? (This month's assignment, you might also try to adapt it to find the length of a file in bytes - hint use INKEY$). The other FuNction, Procrust$, either truncates (1Ø7Ø/8Ø) or pads out with spaces (1Ø9Ø)

the string r$ to make its length exactly p%.

If there is anything I have not explained in enough detail please write to me. Next time Afferbeck Lauder.

(c) 1991 H.J.Clase

Howard Clase

(709) 753-6415
Box 9947, Station B,
St John's, Nfld, A1A 4L4

email hclase@munucs.ucs.mun.ca (Note change.)

Put some **punch** in your
PRINT FACTORY© graphics!!

Collection#1

Miscellaneous
Men/Boys
Women/Girls
Digitized Misc.

Collection#2

Animals
Funny
Sports
Transportation
Computers
Food
Holidays
Xmas

Over 150 large & small graphics in each collection!

Only $12.95 ppd. each or both collections for $19.95 ppd. [Check or money order.]
Available on tape or 5 1/4 DSDD LARKEN or OLIGER disk - please specify.
Send order to:
John McMichael, 1710 Palmer Dr., Laramie, WY 82070

UPDATE MAGAZINE
513 E. Main Street
Peru, IN 46970

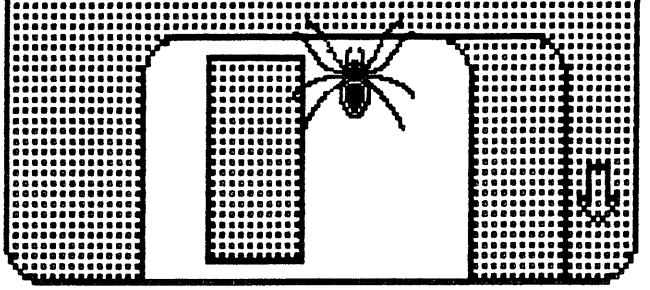
NEWS AND

NEW

PRODUCTS

Bug Alerts & Updates

by Richard Hurd



Here are a few updates and corrections to some of my articles and programs.

NEC 1035 Disk Drive Overview

From SINC-LINK Vol.7 No.6 NEC 1035 3.5" floppy disk drive overview.

In the article I mentioned having problems with formatting disks from MEI/Micro Center. After moving the drives power supply further away from the drive, the problem vanished. The no-name disks have held up as well as the name brand Maxell, 3M, and FUJI disks I've also tried. The price of the no-name disks has gone down to $.39 ea. in lots of 25. See their ad in Computer Shopper magazine.

LARKEN Disk Utility Updates.

An update for Copy Machine which is a modified version of Larry Kenny's Move utility. It allows for batch copy, copy all, and batch erase. The improvement simply makes it easier to reLOAD Copy Machine.

A bug alert for LK2cpy. Which is a modified version of Larry Kenny's 2 drive copy utility. It uses a search routine to look for the highest numbered used track on the disk. It then copies all tracks up to and including the last used track. The fix was suggested by Bob Mitchell, in regards to a full disk. Thanks Bob. Note that for a full disk copy, I suggest using Byte Power's COPY 3 utility program, available from the group's disk exchange. Which due to its' compacting of track data, gives a faster copy. This is the purpose of LK2cpy, but it is defeated with a full disk. LK2cpy is at its best with 75% or less of used disk space.

An update for my RDBKUP (RAMDISK backup) utility. The original saved all tracks of the RAMDISK to a double sided floppy. The improvement now saves only up to the last used track.

These changes can be found on my LARKEN UTILITIES DISK version 2.3. Along with updates on Double to Quad (d2q), a utiltity for 5.25" quad density drive users. It copies 5.25" double density disks to quad density disks using just the quad density drive. I've also added a Quad to Double (q2d) utility, which copies tracks from a quad density disk to a preformatted double density disk. For example you get a double density disk from a friend, use d2q to convert to the quad drive, you then have a preformatted double density disk to load up with something using q2d to return to your friend.

Copy Machine

Important Note: Do not CLEAR or RUN while making changes. After making changes, GO TO 9999 to SAVE.

```
  20 RANDOMIZE USR 60016
 200 RANDOMIZE USR 60016
8095 IF INKEY$="1" THEN  GO TO 8200
8200 CLS : PRINT """"Insert CopyMachine Disk In
Drive""TAB 8;"Which Drive? (0-4)"
8210 IF INKEY$<>"" THEN  GO TO 8210
8220 IF INKEY$="" THEN  GO TO 8220
8230 LET d=CODE INKEY$-48: IF d<0 OR d>4 THEN  BEEP
.1,0: GO TO 8210
```

LK2cpy

```
 615 IF sid=1 THEN LET c=(x/2)-1
 635 IF a=blx AND sid=2 THEN GO TO 740
2065 LET blx=sid*numtrks

Optional
 660 REM after line number, eliminates auto verify,
speeds up copy.
```

RDBKUP (HURD'S VERSION)

```
 210 GO SUB 2000: PRINT "TAB 9;"Copy Count:";stks
2000 REM Get Last Used Track
2010 LET stks=PEEK (mem+21)
2020 LET x=2
2030 FOR f=((mem+24)+(2*stks-1)) TO (mem+24) STEP
-1
2040 IF PEEK f=245 THEN  GO TO 2060
2050 NEXT f
2060 LET x=f-(mem+23)
2070 LET stks=x
2080 RETURN
```

This page was produced with Word Master, Typeliner & Headliner. Printed with a Star NX-1000.

1301 Kiblinger Place
Auburn, IN 46706
Tele: (219) 925-1372
01 27 1991

Dear George,

This is to announce that I am the official SNUG NEWSLETTER EDITOR. Amongst many weeks of wait this has been accomplished. Most of the delay was occassioned by a sort of computer type glitch. Paul Holgren was waiting for material from me and I was waiting for material from him. Now that that has been reconciled we have also reconciled the disk compatibility between us since I now have an Oliger disk interface in addition to my Larken. Paul will not go to another disk I. F. since he is so pleased with Oliger. I have not had my Oliger long enough to really learn much about it but I am beginning to feel that he has a point. I plan to get setup so that I will have both Larken and Oliger operating at the same time, which system gets used will depend upon the disk commands used.

That was one reason to get an Oliger so that I was compatible to Paul and another reason is that there are approximately 1700 disk interfaces for the T/S 2068 sold and of that there are 300 plus Larken and 600 plus Oliger so if I can handle both Oliger and Larken I am compatible with 59% of the known disk interfaces. However, there is the matter of drive size, I am using 5 1/4 both 40 track and 80 track so that should take care of at least a great majority of the Larken or Oliger users.

I am looking for Hen's Teeth, in other words something quite rare. That is a version of MSCRIPT that will support both the Larken and the Oliger from the menu and hopefully cassette too. I would like to have version V5.0 or later but would accept an earlier version if available. Right now on Oliger I am using Mscript V5.0 and when I use Larken I use V5.5. Presently when I transfer files from one the the other I have to SAVE to cassette an(LOAD back to whatever disk system I want to store the material on. V5.0 tha. I have was originally cassette but I have managed to convert it to disk.

Also, I don't know how many know that TASWORD II, which I don't like nor does Paul, files can be LOADed into MSCRIPT but the format will not be clean. But the text is there and can be cleaned up much more easily than typing in the whole text into MSCRIPT. There is a program to clean it up that I will have to try, it could save much time.

As EDITOR I am asking for input. Anything from I don't like this to submission of material to be published. Hints, shorscuts, one liners to long articles on anything pertaining to the T/S computers and even some that aren't if the articles are related. Is there something that you are looking for, write and space permitting it will be published, got something to sell, again send it in and it will get published. Looking for articles on specific topics, write in or call. I have a filing cabinent full of material, I can't afford to supply free copies but the fee will be quite reasonable as long as where I go is reasonable. I will get the submitted material into format for publishing and send it to Paul on a Oliger disk and he will make it ready for the copy machine after picking what should be published. The newsletter will be five sheets of paper and will have a minimum of graphics for front page to increase the amount of text presented. The outside page will be a mailer so actually there will be 9 pages of text. The newsletter will only be as good as the submissions.

Remember that postage cost money, and it has gone up, if you expect a reply send a LSASE please. My mail box almost always has an outgoing letter waiting for the post man so I know postage costs. I have to pay for my hobby from my allowance and when it is spent I have to wait till next month befo I can do anything. Luckily, it has not came to that yet. Also, as much as I like to make phone calls that is one area that my wife watches like an accountant. So I have to limit my calls.

Material submitted should be on 5 1/4 40 or 80 track disk to ease the typing LOAD and should be either Oliger or Larken, perferrably MSCRIPT or if not that TASWORD II. I have never used any other wordprocessor for the T/S 2068. No SPECTRUM versions please since I am not set up for SPECTRUM. If you do not have Oliger or Larken disk or the wrong size then cassette is acceptable. Please use a good quality tape and if postage is sent the material will be returned. If you do not have a disk system and don't want to use cassette hardcopy will be accepted, preferrably typed but long hand will be accepted as a last resort. Anything I type may have errors that I have introduced. Don't worry about misspellings or writing errors, the content is more important than correct English.

Submissions should be sent to me at the following address:

DONALD S. LAMBERT
SNUG EDITOR
1301 KIBLINGER PLACE
AUBURN, IN 46706

Let's see how far we can take our wonderfull little computers!

Sinclairly yours,

Donald S. Lambert

TS2068 SERIAL CABLE

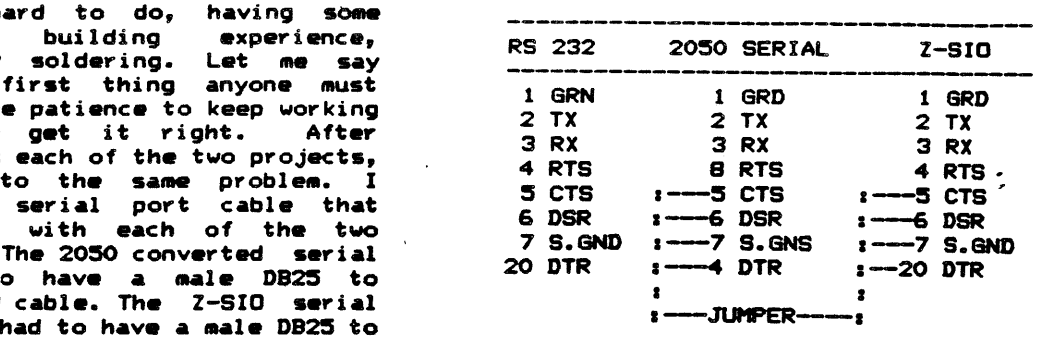
Gallen Bench

Fellow TS 2068 users. I have found over the last few months that some of the information needed to complete some projects is not always clear with the instructions you may receive with the project. During the last couple of months I have started two such projects, namely the conversion of the 2050 modem to a modified serial port, and the building of a Z-SIO port card for my 2068.

The building of both projects was not that hard to do, having some prior kit building experience, especially soldering. Let me say that the first thing anyone must have is the patience to keep working at it to get it right. After completing each of the two projects, I ran into the same problem. I needed a serial port cable that would work with each of the two projects. The 2050 converted serial port had to have a male DB25 to female DB9 cable. The Z-SIO serial port card had to have a male DB25 to male DB25 cable (DB9 and DB25 refers to the connector types).

The instructions for each project did not give a clear pinout for either cable. They did give the signal available at each pin, but how these should be configured with the signals needed for the external modem was not given. After some study and just plain trial and error, I was able to find the correct configuration for each cable. The information was there, but for a novice, it took a little work and some added advice from some friends to put it together right.

Listed below is the serial cable configuration that I have found will work on the Z-SIO serial port card, and the modified 2050 serial port. I hope this information is helpful.



I hope the above listing of how I constructed the cables for either of these serial ports will be helpful.

From The PLOTTER n/1

SCROLL TEXT in a WINDOW

The accompanying program is a demonstration of scrolling text in a window. By adjusting the values of a few variables, one can make the window any size and locate it anywhere on the screen. It is usually located near the bottom or along the sides. The procedure is simple and the action relatively fast for BASIC. As a matter of fact, there is a PAUSE 5 in line 7040 to slow things down.

Window scrolls are just another way of producing a flashy, eye-catching effect. But, used with discretion, they do liven up many an otherwise dull program. This demonstration is arranged to make it easy to modify and incorporate into a program of your own.

The window in this illustration has a useable width of 24 characters, and provides for 3 lines of print. Hence the dimensioning in lines 1000 and 1005, and the frame generated in lines 8010 to 8030. You can use other ways to frame and even use paper color alone to set it off.

The message to be scrolled is in lines 3010 to3070. Notice that the message is divided up into strings of 24 characters maximum, to conform to the window size. Each string is sent to the subroutine that scrolls the window's contents and prints the new message segment, on its bottom line. At the end of the message, lines 3080 to 3090 clear out the window and line 3100 prints the sign-off. In a working program, one would go, instead, to the next step at this point.

The action of the printing in the window is such that the message segments move progressively from c$ to b$ to a$ in lines 7010 and 7015 with d$ just providing spaces. As they move in this fashion, they are printed in ascending rows by line 7020. This produces the scroll effect. Line 7040 prints the message segments one character at a time - typewriter style - on the bottom line of the window. We could have printed the line in one fell blow, but then the action might be too fast and require a larger PAUSE value to offset this. One gains but little in so doing, and loses the BEEPing which adds such a nice touch.....

```
  5 REM ** "E-41", 9-24-88, WF
 10 REM ** SCROLL TEXT IN A
           WINDOW
 15
1000 DIM a$(24): DIM b$(24)
1005 DIM c$(24): DIM d$(24)
1010 PRINT AT 3,0;"  The screen
itself does not    scroll.  Any
other action can goon here, alte
rnating with the   display in th
e window.  It's up to you."
1020 GO SUB 8000
1030
3000 REM ** WINDOW MESSAGE
3010 LET m$="In the window size
sel-": GO SUB 7000
3020 LET m$="ected we are limite
d to": GO SUB 7000
3030 LET m$="a maximum of 24 cha
ract-": GO SUB 7000
3040 LET m$="ers to a line.  Les
s": GO SUB 7000
3050 LET m$="than this is O.K.
More": GO SUB 7000
3060 LET m$="than this will be t
opped": GO SUB 7000
3070 LET m$="off....": GO SUB 70
00
3080 FOR m=1 TO 3: LET m$=d$
3090 GO SUB 7000: NEXT m
3100 LET m$="END OF DISPLAY...."
: GO SUB 7000
3110 STOP
3120
7000 REM ** PRINT IN WINDOW
7010 LET a$=b$: LET b$=c$
7015 LET c$=m$
7020 PRINT AT 16,4;a$;AT 17,4;b$
;AT 18,4;d$
7030 FOR n=1 TO LEN c$
7040 PRINT AT 18,3+n;c$(n): IF
$(n)<>" " THEN BEEP .005,20: PAU
SE 5
7050 NEXT n: RETURN
7060
8000 REM ** DRAW WINDOW FRAME
8010 PRINT AT 15,3;"▛▀▀▀
```

TASWORD

EDIT-------help page
CAPS LOCK--capitals lock
TRUE VIDEO-cursor to word left
INV. VIDEO--cursor to word rt.
ARROWS-----cursor movement
GRAPHICS---printer control chr.
DELETE-----delete character
<= ---------move line left
<> ---------center line
>= ---------move line right
AND--------insert line/char.
OR---------go to end of text
AT---------go to start of tex
STOP-------load/save/print
NOT--------delete line
STEP-------reform to para end
TO---------scroll down
THEN-------scroll up
ENTER------start of next line

Print Control Characters
On Epson FX80 Off

- enlarged
- condensed
- underline
- italics
- emphasised
- double strike
- elite
- proportional

EXTENDED MODE CONTROL KEYS

SCROLLING

F - fast scroll up
G - fast scroll down

ZX PRINTER

P - print text
L - large printer on marker
K - large printer off marker

MISCELLANEOUS

C - change window on text
X - clear text
R - replace of find text
I - insert mode on/off
EDIT - help page(this info)
ARROWS - cursor movement

FORMATTING

E - right justify on/off
W - word-wrap
J - justify line
H - unjustify line

MARGINS

A - set left margin at cursor
S - reset margins to normal
D - set right margin at cursor

BLOCK COMMANDS

B - mark beginning of block
V - mark end of block of text
N - copy marked block to cursor
M - move marked block to cursor

TEXT FILE AT 33280

SH	HELP	CAPS LOCK	WORD ←	WORD →	CURSOR ←	CURSOR ↓	CURSOR ↑	CURSOR →		DELETE
	1 1	2 2	3 3	4 4	5 5	6 6	7 7	8 8	9 9	0 0

F_M		W. WRAP	JUSTIFY	FIND				INSERT		
	Q	W	E	R	T	Y	U	I	O	P
S_S	←LINE	CENTER	LINE →			INSERT	END	HOME		

F_M	L MARG	NORM MARG	R MARG	FAST ↓↓	FAST ↑↑	UNJUST	JUSTIFY			
	A	S	D	F	G	H	J	K	L	Enter
S_S	FILES	DEL LINE	REFORMT	SCROLL↓	SCROLL↑					

			CLEAR	64/32	END	START	COPY	MOVE	EXTMODE			
Shift	Z	X	C	V	B	N	M	Symbol Shift	Break (Space)	Shift		
				BLOCK	BLOCK	BLOCK	BLOCK	F_M=SH+S_S				

Kansas Area Timex Sinclair Users Group

KEYBOARD OVERLAY

MSCRIPT

FORMAT LINE COMMANDS

> in first column plus:
JU=Y/N JUstify left
CE=Y/N CEnter
FR=Y/N Flush Right
HI=nmbr Hanging Indent
SS=Y/N Single Sheet
LM=nmbr Left Margin
LL=nmbr Line Length
LS=nmbr Line Spacing
PL=nmbr Page Length
PS=nmbr Page Spacing
PN=nmbr Page Number
TT=text Top Title
ET=text Even-pn TT
OT=text Odd-pn TT
BT=text Bottom Title
EB=text Even-pn BT
OB=text Odd-pn BT
HM=nmbr Header Margin
FM=nmbr Footer Margin

HOME MENU

Add text
Backup
Catalog
Drive select
Erase file on disk
Give name to file
Interface select
Kill text
Load text file
Number of copies
Paper/ink color
Quit (run to resume)
Restore killed text
Save text
To text
Use: Disk or Cassette
View memory
X = Block print
Y = Eject (formfeed)
Z = Linefeed

SUB MENUS

COMMAND MENU

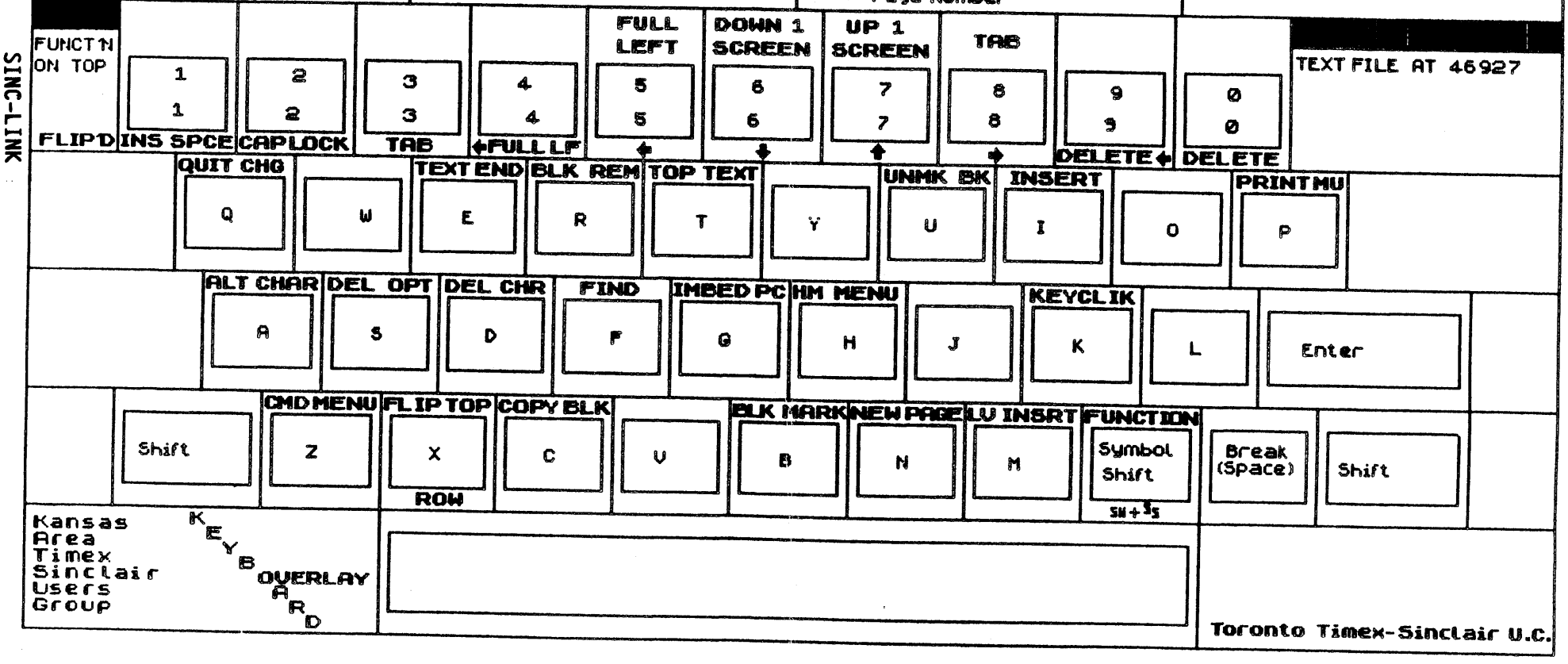
A- Append document (A name)
C- Change find string (C string)
F- Find text string (F string)
H- Home menu (H)
L- Load new document (L name)
S- Save document (S name)
T- Set Tab positions (T x,x,x)
W- Window width adjust (W x)

PRINT MENU

JU - Justification
SS - Single Sheet
PL - Page Length
PS - Page Spacing
LM - Left Margin
LL - Line Length
LS - Line Spacing
PN - Page Number

MISCELLANEOUS

POKES



Kansas Area Timex Sinclair Users Group

KEYBOARD OVERLAY CARD

Toronto Timex-Sinclair U.C.

Q L I P S

by Hugh H. Howie

Recently we made an exchange of our QL Library with the NESQLUG in Boston, and when I was going through their Library, I came across this item of interest, at least I was interested in it, so perhaps you also will be able to glean a few gems from it.

It was submitted to the NESQLUG Library by Peter Hale; written by an unknown author. This author although unknown to us, is worthy of our recognition.

********** QLHardw_doc **********

Author unknown . . . . . .Contributed by Peter Hale

The QL has undergone only two design changes during its varied fortunes from a hardware point of view leading to the conclusion that either it was nearly right from the word go, or it is so awful that very little can be done with it. Events taking place susequent to its launch confirm that it is a good machine that has many advanced features as well a microdrives which are OK but not everyones cup of tea. It is significant that a fairly major design change took place at build no.14 moving from Issue 5 to Issue 6 boards. Issue Ø,1,2,3 & 4 boards have not been encountered leaving one to assume they were R & D material.

Reference will simply be made to Issue 5 or Issue 6 boards as build numbers such as DØ5, D11, D16 etc. also AH,JM,JS ROM's are not significant from the hardware aspect. The only certain way to ascertain what is inside the black box is to open it up and examine the board. This is achieved by turning the QL over to gain access to eight cross-head screws. There being four short ones under the front lip and four long ones in line with the legs. Do not remove screws from under the microdrives at this stage. Use a new screwdriver with the point covered in cling film to obviate an amateur chewed screw look.

Lift the keyboard up gently noting two wide flat strips which are the membrane connector tails. Grasp these between finger and thumb pulling gently but firmly vertically upwards. There is no need to disconnect anything else at this stage.

Re-assembly is simple, with the membrane tails being carefully inserted by a gentle downward movement ensuring that they are not creased which could break the internal track. Also check that the tails are not folded over in the socket as this will prevent some keys from functioning.

The issue number is on the board in three places, but the two different boards are easily identified by looking to the right of the ROM chips. If it contains a HAL 16L8 chip, it is Issue 6. Issue 5 boards employed a 74LSØ3 chip.

The various ic's (chips) will be dealt with purely from a servicing aspect laying particular stress on practicality. No attempt will be made to discuss the technical merits of a particular ic or to propose extensive modification to the circuit invloving spaghetti wiring.

Purchase of the QL Service Manual is strongly advised and at least some practice made on soldering small components before embarking on any updates or projects. It is essential that correct means be employed in extracting ic's as levering with screwdrivers etc. will damage the board as well as the ic. Pcb's are not available as replacement parts consequently damage of a serious nature will scrap the machine. As always it is advisable to work on baking foil but most of the ic's appear to be fairly robust.

Two ULA's are used on the main board and one on each microdrive. The microdrive ULA is the same as the SPECTRUM microdrive ULA. It goes off pop occasionally probably due to it being fed large doses of heat from the 5 volt regulator heatsink, its own heat and heat from the motor. It is not easy to

replace on mdv1 and a socket cannot be used as the ULA is jammed up to an ic on the board with a insulated screen in between. Leaving the screen off produces more heat, smoke, damage and anguish.
The microdrive pcb is not available as a replacement part which is a pity as chip failure can damage the board.
Fastened to the board is also the microdrive READ/WRITE head. Not unknown to fail but expensive to replace. It is only available as a 'sub-assembly' being a bog standard stereo tape head on a plastic mounting. Thus two parts become a sub-assembly !!!.
Microdrive ULA's arrive with adorned with a variety of coloured spots, white ones, pink ones etc. They must mean something but do not appear to have any performance significance.
Rattles heard whilst microdrive are spinning indicate the rubber drive rollers are not perfectly true or more probably the roller in the cartridge is not true. Replacement of the microdrive rubber roller is easy with a bit of glue to make sure they stay in position.
Microdrive 2 has a useful space under it which will be later utilised when the battery back-up is explained.
Turning to the main board, one of the most frequent replacements is the ROM chips. There are two which must be replaced as a pair. You can't get away with updating the one chip. Usually owners want to update to a JS ROM which is easy even if you have an AH. Just extract the two chips and insert the new ones the correct way round. Have a look at the JS ROM chips as the colour spots again appear (blue ones usually) and a leg disappears from the 128k chip. It is one of the 5 volt supplies. Don't worry it works OK. Whilst the top is off the QL it is worth considering how much keyboard bashing you do and the consequences of a keyboard failure at a critical time. Murphy's law will cause it to fail on something important. Replacement of the switch contacts is by means of fitting a new keyboard membrane being both cheap and easy, after the aluminium plate has been removed. A bubble mat provides the 'springs' for the keys so there may be good reason for fitting a new one of these at the same time. The ultimate I suppose is to invest in a proper PC type serial keyboard plugged into SER2. Be careful when refitting the upper case to the lower, as any poor contact on the six LED connections stops the appropriate microdrive.
Back to the main board to have a look at the ULA which controls the serial ports and the microdrives. It is a ZX 8302 having a 5 volt supply on pin 40 & pin 11 with pin 40 being fed through a link.
Links in the QL are zero ohm resistors. That is, they look exactly like a resistor making it easy for automatic mechanical insertion but with a single black band to indicate the value. Even though the circuit diagram shows a link for both Issue 5 & Issue 6 boards, in practice, only Issue 5 has this present at the side of SER1 port marked D22. That's right. Issue 5 board has a link which looks like a resistor, in a position marked for a diode, with another position at the side for another diode D23.
For some reason Sinclair decided to redesign this area on Issue 6 boards.
Part of the ZX 8302 provides the clock for the QL with all the hassle of having to reset it frequently, but if the chip can be permanently powered then the date and time would remain correct.
To implement this change, replace the link in position D22 with a small diode together with a further diode in D23. A small 3.8 volt rechargeable battery connected to the points marked BATTERY (under the heatsink) and fastened with sticky pads under microdrive 2 does the trick. For purists, something like a 270 ohm resistor in the positive lead will restrict the charging current.
On Issue 6 boards this simple approach is not possible due to changes in the track layout. The only practical solutions being to isolate pin 40 either by bending it out, cutting it or the track, after which the same type of battery can be connected by soldering. POS (+) to pin 40, NEG (-) to regulator common.
Staying with the ZX 8302 provides a further item of interest in that colour spots are not used but marking codes are.

Example:-
ZX 83Ø2 NCRO371275 USA F812119 F8416
ZX 83Ø2 NCRO371275 USA F812877 F8431

In my experience, a change of chip significantly improves microdrive handling, but it pure conjecture as to whether the ZX 83Ø2 is faulty or if chips of later manufacture are of an improved quality.
Moving on to the next ULA, namely the ZX 83Ø1, a similar situation arises in that different markings appear to coincide with differing performance.

ZX 83Ø1 markings:-
PS P Ø2 CLA231Ø Y84Ø8B
PS P B CLA2345 8544C ZX83Ø1 Sinclair 1983

Jittery monitor displays usually respond to the replacement of the ZX 83Ø1 with a loss of colour requiring a new MC 1377p. Squirt the MC 1377p with freezer to restore your QL to full colour until it warms up again. This trick also produces some amazing results on other chips as does running the machine with the top off. Indications therefore are that it gets a bit hot inside and some ventilation would help in conditons of high ambient temperature.
The reported machine crashes can be reduced by cooling the machine but for real improvement the 78Ø5 5 volt regulator requires attention both in the QL and in any add-ons. A 1ØØ nf capacitor must be soldered between input and common on the chip not some distance away. It is part of the specification and is chearfully ignored by some manufacturers of computers to the detriment of the performance. The knittng of three platefuls of spaghetti hanging out the back is asking for trouble. Keep wires neat and well away from each other, especially the power lead, mains lead and VDU lead. I sugest you use a curly VDU lead and run the power lead to the right of the machine. Running the TV or monitor slap up against the QL is bad practice not only for the QL but it is in a poor position to be viewed without strain. If possible the interior of the QL casing should be sprayed with elecrically conductive paint and two 78Ø5 regulators paralleled up with the use of liberal doses of heat-sink compound. Any regulators in add-on boards must be checked to make sure they are contacting the heat-sink and again use heat-sink gung. Do not forget the capacitor across the input to the regs to remove any possibility of oscillation.
The two remaining large chips are the 68ØØ8 CPU and the MAB 8Ø49 co-processor both of which appear to be reliable. The 8Ø49 receives the RS 232 interface signals,monitors the keyboard, controls the loudspeaker & co trols the joystick. A simple but effective test is to plug SER1 into SER2 then get the computer to talk to itself by some BASIC software.
A joystick test can also be incorporated as well as a colour and sound test.
This pretty-well proves the system. A RAM test requires a m/c routine but if there is a RAM failure it is usually visible as a black line on the tweed start-up screen or with a completely green or white screen.

MODIFICATIONS
1. The work involved in having your clock running continuously is quite small and fairly easy. If possible check the current consumption which should be less than 2 microamp with the mains power OFF. If the consumption is much greater then examine the back of the board for a resistor which should not be there.
If the clock does not work from a battery, check for continuity between the battery and the chip. It is not unknown for tracks having been cut.
2. Some QL's have 33K resistors soldered to pin 21 & 19 of the 83Ø2 then connected to -12 volts. Some machines don't have them and some have different value resistors. It is a nifty piece of soldering but the manual states it is mandatory. It looks to me that the factory have never read the manual.
3. Own up all those who kept the dongle !. Now is the time to put it to good use by opening it up to reveal a 16K EPROM. Have the EPROM reprogrammed with what you need then insert it in the socket you have soldered into the vacant

location at the side. If a switch is used to pull the appropriate chip select line low, then two EPROMS can be used but only one at any one time.

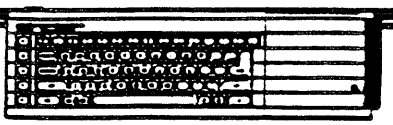
Semiconductor	Function	Issue 5	Issue 6
74LSØØ	2-in NAND Gates	*	*
74LSØ3	3-in NAND Gates	*	-
74LS245	Bus-transceiver	*	*
74LS257	Address Multiplexer	*	*
78Ø5 5v	Regulator	*	*
7812 12v+	Regulator	*	*
7912 12v-	Regulator	*	*
HAL 16L8	Hard Array Logic	-	*
MA 8Ø49	Peripheral controler	*	*
MC 1377P	PAL/TV Signal Generator	*	*
MC 68ØØ8	Central processor	*	*
SN 1488	RS 232 Buffer	*	*
SN 1489A	RS 232 Buffer	*	*
ZX 83Ø1 ULA	Controls display	*	*
ZX 83Ø2 ULA	Controls MDV's, SER & clock	*	*

Test Program

The test program on the cartridge SER1 and SER2 to be linked so that the QL can talk to itself which is the most comprehensive of the sytems tests. If the ports are not linked then type GOTO 448Ø to carry on.

The memory test is very detailed running as a job but it takes a very long time to complete. A QL with the full 512K of extra memory will take about two and a half hours. Watching it is as interesting as watching paint dry but it is great as a demonstration for non-computer people. If you put a tiny bit of something in memory it indicate it in the BAD window.

If free_mem from Toolkit II or better still Free_memory from Supercharge or the memory test indicate faulty chips then put a good memory chip in a test clip and put it over each chip in turn. Power down between each test or the results will not be meaningful.



We carry **ALL** hardware and software lines for the Sinclair QL.

We also carry the Z88.

WRITE FOR FREE CATALOG

Sharp's, Inc.

Rt. 10, Box 459
Mechanicsville, VA 23111
(804) 746-1664 or 730-9697

IT'S HERE!!

QL TRIVIA-QLUE

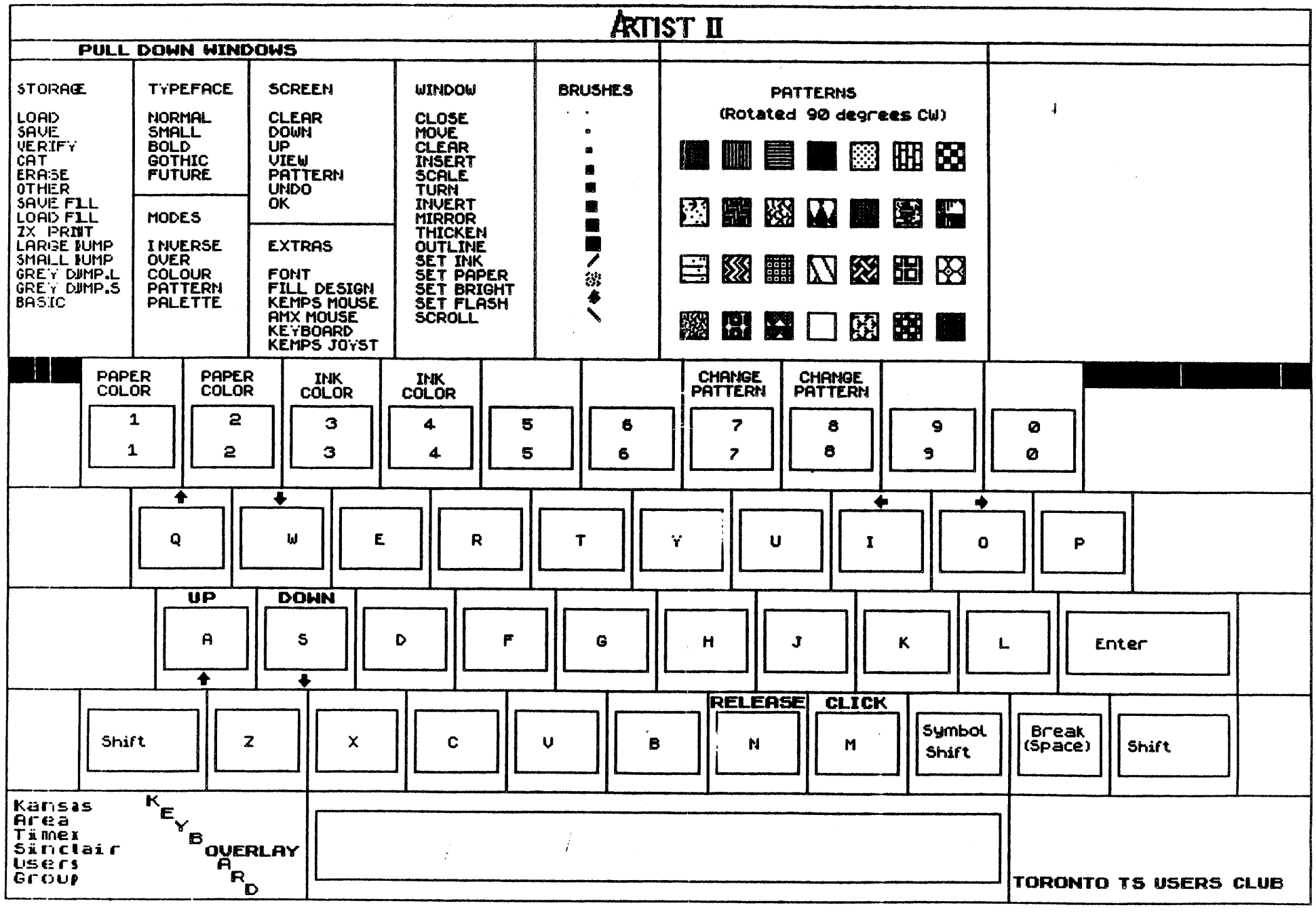
HERE IS GREAT NEW COMBINATION GAME!
IF YOU LIKED QLUE AND TRIVIA, THEN THIS IS FOR YOU!
WE HAVE COMBINED THESE TWO GREAT THINKING MANS' GAMES
IN ONE GREAT PACKAGE!!

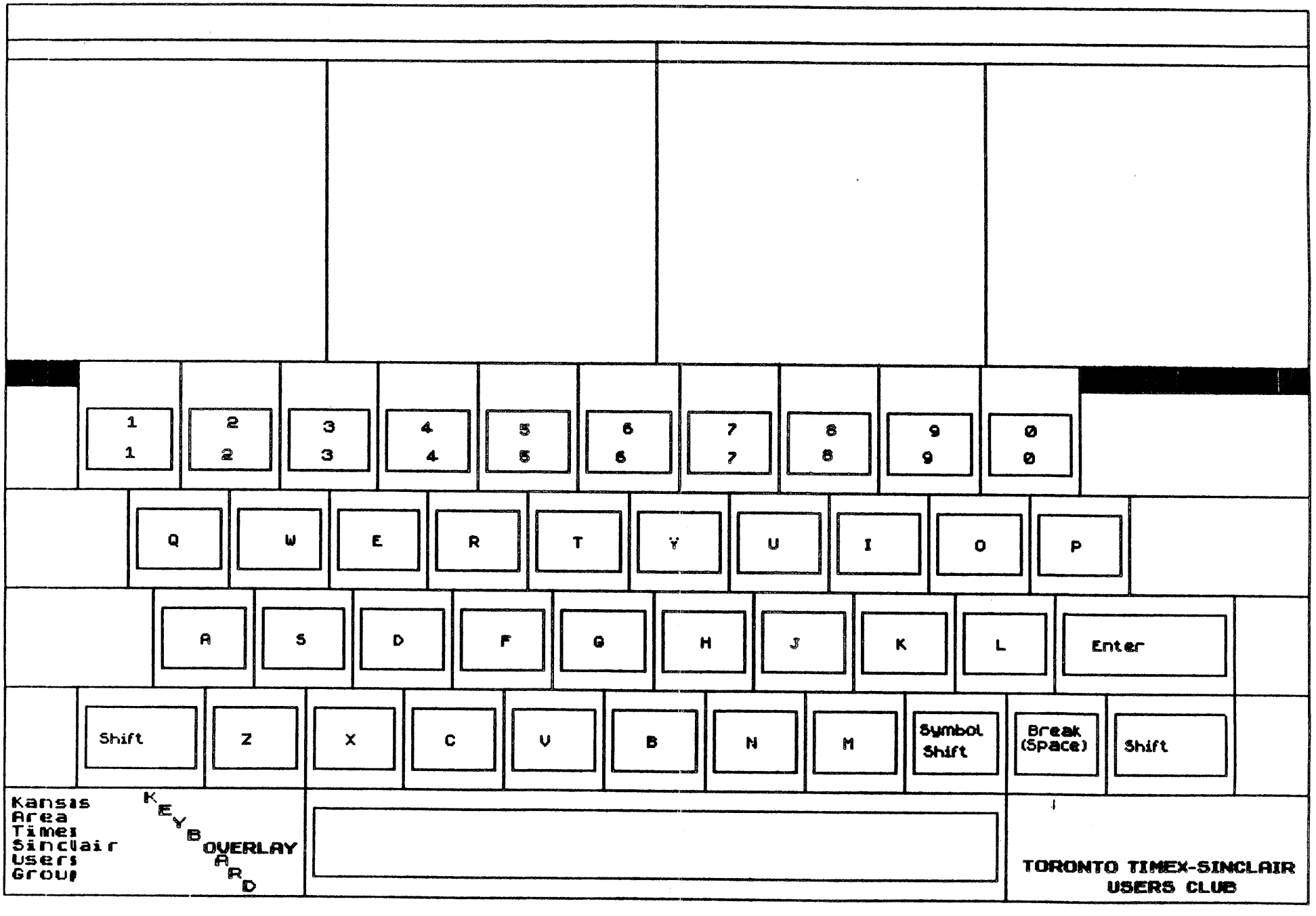
13 ROOMS - 13 WEAPONS - 13 SUSPECTS
THAT'S OVER 2100 POSSIBLE OUTCOMES!
AND NOW - OVER 750 TRIVIA QUESTIONS!
QUESTIONS COVER EVERY CONCEIVABLE SUBJECT!
RELIGION - HISTORY - SCIENCES - GEOGRAPHY - ASTRONOMY
YOU'VE NEVER SEEN QUESTIONS LIKE THESE!

Also included, at no extra cost, your choice of
FOOTBALL FORECASTER or LIFE EXPECTANCY FORECASTER
(Specify which you would like when you order.)

WE THINK YOU'LL LIKE THIS ONE!
OUR PRICE? ONLY $19.95 +$2ph

AVAILABLE EXCLUSIVELY FROM:
RMG ENTERPRISES


ARTIST II
PULL DOWN WINDOWS
STORAGE
LOAD
SAVE
VERIFY
CAT
ERASE
OTHER
SAVE FILL
LOAD FILL
ZX PRINT
LARGE DUMP
SMALL DUMP
GREY DUMP.L
GREY DUMP.S
BASIC
TYPEFACE
NORMAL
SMALL
BOLD
GOTHIC
FUTURE
MODES
INVERSE
OVER
COLOUR
PATTERN
PALETTE
SCREEN
CLEAR
DOWN
UP
VIEW
PATTERN
UNDO
OK
EXTRAS
FONT
FILL DESIGN
KEMPS MOUSE
AMX MOUSE
KEYBOARD
KEMPS JOYST
WINDOW
CLOSE
MOVE
CLEAR
INSERT
SCALE
TURN
INVERT
MIRROR
THICKEN
OUTLINE
SET INK
SET PAPER
SET BRIGHT
SET FLASH
SCROLL
BRUSHES
PATTERNS
(Rotated 90 degrees CW)
PAPER COLOR 1 1
PAPER COLOR 2 2
INK COLOR 3 3
INK COLOR 4 4
5 5
6 6
CHANGE PATTERN 7 7
CHANGE PATTERN 8 8
9 9
0 0
Q
W
E
R
T
Y
U
I
O
P
UP
DOWN
A
S
D
F
G
H
J
K
L
Enter
Shift
Z
X
C
V
B
RELEASE
CLICK
N
M
Symbol Shift
Break (Space)
Shift
Kansas Area Timex Sinclair Users Group
KEYBOARD OVERLAY
TORONTO TS USERS CLUB


1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
0 0
Q
W
E
R
T
Y
U
I
O
P
A
S
D
F
G
H
J
K
L
Enter
Shift
Z
X
C
V
B
N
M
Symbol Shift
Break (Space)
Shift
Kansas Area Timex Sinclair Users Group
KEYBOARD OVERLAY
TORONTO TIMEX-SINCLAIR USERS CLUB

****
*** INTERRUPT MODE 2 ***
*** by Bob Mitchell ***
*** 910403 ***
****

INTRODUCTION

The purpose of this paper is to bring together many of the details on the Interrupt Mode 2 (IM2) that I have gathered over the past several years. References are included in brackets, eg [1], to indicate source material detailed at the end of the text.

BACKGROUND

An interrupt is a signal sent to the microprocessor which generally suspends the execution of the current program, without the program being aware of it. The processor resumes its original work when the interrupt service routine finishes. These interrupts take place at the rate of 60 per second (50 per second for Spectrum).

Interrupts are used for two main purposes: [1]

1. To deal with hardware events such as a key being pressed or a character arriving through a serial port. These events cannot be ignored; the incoming data must be either processed immediately or stored in a buffer.

2. To call subroutines that are provided by the hardware or operating system.

SPECTRUM

In the Sinclair Spectrum and indeed in the TS2068, such an interrupt forces it to jump to 38 hex or 56 decimal where it increments the system variable FRAMES and scans the keyboard to determine what is being keyed in, if anything. If this interrupt were diverted elsewhere to a user routine it would provide a very powerful facility with unlimited potential uses. This is precisely what the use of IM2 is all about.

There are actually three interrupt modes in the Z80 processor but only two of them apply here: IM1 and IM2. When the Z80 receives an interrupt, it expects the interrupting device to place one byte of data on the data bus which acts as the low order of an address.

The interrupting device is the ULA (Uncommitted Logic Array) which implements necessary timing and hardware control. It is used to select chips and decode addresses for RAM and ROM; for timing and control of the Z80A microprocessor; for keyboard operation sensing; to control signal conditioning and transfer of input and output information. [2]

The high order of the address comes from the I register. The Z80 looks at this address which gives it A SECOND ADDRESS that it then jumps to. In the Spectrum, no such low order byte is provided and by default the Z80 thinks it sees FFh or 255d. Apparently this does not always work properly ; Toni Baker warns against using IM2 on page 102 of her book on machine code in that after some time a system hangup appears to occur. In any case, it certainly does not work with the TS2068, at all.

Still, it is worth examining the Spectrum method more closely, the better to understand what has to be done with the TS2068.

The I register needs to be changed to a value so that the pair of addresses <I*256+255> and <I*256+256> point to a user defined routine. But be warned: the I register cannot be altered to hold a value between 64 (40h) and 127 (7Fh). If

the I register is loaded with addresses within the forbidden area, there will be some screen interference. [3]

An inspection of the addresses so formed is made and a jump is made to the address that it holds; the latter is known as a VECTORed address. Therefore a vector must be found somewhere in RAM and above 32768 (8000h).

A search of the Spectrum ROM turns up a suitable value for I which will cause a jump to 65129. This value is 09h. Putting this into the address formulae in the previous paragraph gives the pair of addresses 09FF and 0A00 (2599 and 2600 dec); examining the ROM disassembly for Spectrum shows these two addresses contain 69FE or FE69 taking the bytes in the proper order which equates to 65129 dec. This is the start address of the user routine that IM2 will activate.

Two added notes: [3]

1. If Interface 1 is fitted to the Spectrum, the vectored address may be taken from the Shadow ROM causing a crash.
2. Prior to using IM2, all registers should be saved and then restored on returning to the normal IM1 status.

TS2068

The TS2068 does not supply a fixed low order byte but rather one that may vary from 0 to 255. This could be a problem but there is a way out. One technique to overcome this problem is to create a block of 257 addresses each filled with the same value. The I register is altered to point to this block and IM2 is called. No matter what the lower order byte turns out to be, the block will produce the same address which will vector control to the start of the user-generated routine. Why 257 addresses? Remember that two addresses are inspected and if the low order byte were 255 (ie, the last of 256 bytes), there would have to be one more address with the same value as all the rest.

The vectored address can be a value which is a multiple of 256; or it can be any address, if a specific one is defined in the set-up routine. If the vectored address finds an instruction to jump to a specific address, then that is where it will go.

Examine listing 1 (a MOTHER PROGRAM used to generate the block of bytes and set up the vectoring to a specified address).

LISTING 1

```
FE00 C5        PUSH BC          :Save all
FE01 D5        PUSH DE          :the
FE02 E5        PUSH HL          :registers
FE03 F5        PUSH AF
FE04 2100FB    LD   HL,FB00     :point to the block start.
FE07 0600      LD   B,00        :set reg B for loop;becomes
                                 FF when loop starts.
FE09 36FC      LD   (HL),FC     :put FC into HL
FE0B 23        INC  HL          :point to next byte.
FE0C 10FB      DJNZ FE09        :repeat until B=00 again
FE0E 36FC      LD   (HL),FC     :put FC into 257th address
FE10 3EC3      LD   A,C3        :put C3 into A register
FE12 32FCFC    LD   (FCFC),A    :load FCFC with C3 (JP)
FE15 2169FE    LD   HL,FE69     :load HL with addr 65129
FE18 22FDFC    LD   (FCFD),HL   :poke this into next two addrs
FE1B 3EFB      LD   A,FB        :load A register with FB
FE1D ED47      LD   I,A         :load I register with FB
FE1F F1        POP  AF          :restore all registers
FE20 E1        POP  HL
FE21 D1        POP  DE
FE22 C1        POP  BC
FE23 ED5E      IM2              :turn on IM2 & jump to 65129
```

```
FE25 C9        RET             :return
```

However, this complexity is not needed if a jump is to an address which is an exact multiple of 256. Listing 2 below gives an example of this version. [4]

LISTING 2

```
FE00 C5        PUSH BC
FE01 D5        PUSH DE
FE02 E5        PUSH HL
FE03 F5        PUSH AF
FE04 2100FE    LD   HL,FE00
FE07 0600      LD   B,00
FE09 36FD      LD   (HL),FD
FE0B 23        INC  HL
FE0C 10FB      DJNZ FE09
FE0E 36FD      LD   (HL),FD
FE10 3EFE      LD   A,FE       :put FE into A register
FE1D ED47      LD   I,A        :load I register with FE (65024)
FE1F F1        POP  AF
FE20 E1        POP  HL
FE21 D1        POP  DE
FE22 C1        POP  BC
FE23 ED5E      IM2             :turn on IM2
FE25 C9        RET             :return
```

In this version the value loaded into the A register at address FE10 determines where the user defined routine should start.

Some final thoughts:

When this technique (the 257 byte table) was installed in the TOOLKIT utility TS2068 version (tstk.C1 on the Omnibus Disk), it of course made all the options using IM2 operational. Later when the TS2068 version was converted back to Spectrum, the technique still did its job, at least with the Spectrum Emulator. Whether it works on a Spectrum machine has not been confirmed.

A concerted effort has been made here to collect accurate information as a ready reference next time use of IM2 is to be attempted. There may be errors and I would be grateful if any reader would let me know of any that are found.

Bob Mitchell 20 Wild Briarway Willowdale Ont M2J 212

****

Footnotes (References used in the text).

[1] Barron's Dictionary of Computer Terms
[2] ZX Spectrum User's Handbook; by Simpson and Terrell
[3] Howard Allen: Popular Computing Weekly 850704
[4] Floyd Chrysler: Time Designs Sep/Oct 87.

****

Q L I P S

By Hugh H Howie

Doing some reading recently and it came to me that we should be holding some kind of a big birthday bash for some of those folks who have done so much for us, or should I say, so much to us?

It is not so long ago that a computer was some kind of nut who computed things for us. He had to be for we were not capable of doing it for ourselves, until a guy called Sinclair (and others) came along about ten years ago, (or so) when he came out with his little black boxes which we have all learned to love. (Did someone say hate?)

Anyway we have come a long way since then, perhaps too far. We have all kinds of little black boxes, and white ones too. Some a bit off-white, some just plain colored according to the type of language we are using to describe them. That language is a bit colored at times!

Anyway, Sinclair got Knighted by the Queen of the British Empire for what he did to us. Just imagine, he got to be called a SIR, and allowed to ride around on a White Charger just like a fairy tale Prince, just for giving us a black box to play with. Now was that fair?

He did not do this all on his own, Sinclair didn't, he had the help of a man called Charles Babbage who with the help of the Government tried to build a mechanical machine to do all the work of the Government so that they could collect taxes a lot easier in the future. The government of the day put up a handsome £17000, and Babbage himself put in £6000.

That was way back about 1830 or somewhere about then. Who cares about the exact date? they did not have computers to do the work for them. In fact there is doubt as to the exact date of his birth. Some of my sources say 1791, and some 1792. But I needed something to write about so my contention is he was born in 1791. One source says he was born in Devon in 1792, where his father was a banker. The most exact date I can get he was born on December 26th 1791, in Walworth in London. Just a quarter mile from the Elephant and Castle Pub. A couple hundred miles from Devon and a year earlier.

I guess he got his flair for figures from his banker/father, also how he got his idea of compiling a table of logarithms. (Cor bless 'im)

He at one time even tried ghost hunting, and there are aplenty of them over there, and ever since we have been blessed with our own versions of ghosts on the silver (or whatever) screen. Quite a guy this, he, hated organ-grinders.

He went to Trinity College, and about 1810 tried his hand at constructing a machine to do his calculations. He got so far as to ask the Government of the day for a grant, but the Prime Minister at that time, one Robert Peel, dismissed his ideas as "worthless to science" Even Disraeli thought the counting machine's only use might be how to calculate how much money had been wasted on it.

A few years later Sir Robert Peel changed his thoughts, and offered Charles Babbage a knighthood, which our Charles of course turned down. Thereby losing a nice little stipend. Ah So! I guess his computer was down that day!

A few years pass, Kings and Courtiers come and go, but the works of Charles Babbage go on forever. So much so that without his pioneering we would not have so many of the things we have to-day. We even used his ideas to get to the Moon.

Now, to-day, all those years later, Charles Babbage is being honored. In Britain a Postage Stamp is being brought out in his honour. Not only that, but they are going to build a full size working model of his "Second Difference Engine" (It was later they called it a computer.)

This contraption which he wanted to build way back then, is going to be built this year, and it is going to be displayed in the Museum of Science and Industry. Now aint that something?

The man who invented the computer is being honoured TODAY. All those years after he could not get money to do the job.

By the way, the model being built this year is going to cost the Computer Industry about £500,000. thats half a million, (pounds that is.) On top of that, it is going to weigh in at about three ton, and stand about six feet high. Got a spare room in your home?

To finalise the story. Charles Babbage was not buried in one piece. They reckoned he was such a genuis, his brain should be preserved for posterity. It is kept in the Hunterian Museum, at the Royal College of Surgeons. His body was buried in Kensal Green Cemetery in 1871.

So there you have it. When you are alive, and working with computers, you are told you are out of your mind. When you die, they take the mind out of you! and store it away for folks to gawk at.

Sometimes you just can't win..

Lets have a BIG BIRTHDAY BASH for
CHARLES BABBAGE, the father of all Computer Battles.

Happy 200 Years Mr Babbage. §

MIKE'S NOTEBOOK

By: Michael J. Di Rienzo

(NOTE: REPRINTING OR REPRODUCING THIS COLUMN WITHOUT THE EXPRESSED WRITTEN PERMISSION OF THE AUTHOR IS HEREBY PROHIBITED. FOR PERMISSION, WRITE THE AUTHOR IN CARE OF THIS PUBLICATION.)

Here's another easy-to-use command utility which quickly PRINTs a string of text at any size and at any pixel coordinate location. You will be able to use this utility in your own programs. The syntax follows: PRINT USR large;AT X,Y;W,H;"txt" The text at the end of the line can either by within quotes or use a string variable (A$...Z$). The variables "X" & "Y" are the column (0-255) and line (0-191) pixel location where you want the text to commence PRINTing. The coordinate 0,0 is at the upper left corner of the screen. The "W" variable is the letter width in terms of characters (1-32), and "H" is the height of the letter (1-24). Be sure to note the commas and semicolons in the command syntax. If you make an error, you will get a "Nonsense in Basic" error. Note that the PAPER and INK command can be used within the PRINT statement in their usual way. FLASH also works. If your text runs off the screen, it will wrap around to the opposite side, so you'll have to plan for text length and size. Be sure to RUN line 100 to initialize the CODE before RUNning for the first time, then use RUN thereafter. The CODE is not relocatable and is 350 bytes long. Have fun!

Happy TIMEXing...

"LARGE"

By Michael J. Di Rienzo

```
  10 BORDER 0: PAPER 0: INK 7: C
LS : OUT 254,1: LET large=60400
  20 LET d$="2068 LARGE print"
  30 RANDOMIZE
  40 LET w=INT (1+RND*2)
  50 LET h=INT (1+RND*12)
  60 LET x=INT (RND*(256-(w*8*16
)))
  70 LET y=INT (RND*(192-(h*8)))
  80 LET i=INT (1+RND*6)
  90 PRINT INK i;USR large;AT x,
y;w,h;d$: PAUSE 100: CLS : GO TO
 40
 100 CLEAR 59399: LET t=0
 110 FOR n=60400 TO 60749
 120 READ a: POKE n,a: LET t=t+a
 130 NEXT n: IF t<>31728 THEN PR
INT FLASH 1;"Data Error!": PRINT
 "Recheck DATA lines": STOP
 140 RUN
 150 DATA 17,117,33,225,167,237,
82,32,250,213,223,254,59,32,40,2
31,254,172,32,35,205,220,27,205,
96,38,237,67,10,91,223,254,59,32
,20,205,220,27,205,96,38,237,67,
12,91,62,8,50,14,91,223,254,59,4
0,2,207,11,231,205,239,27,205,17
5,47,33,15,91,235,237,176,62,255
,18
 160 DATA 33,15,91,126,35,34,0,9
1,111,60,200,38,0,41,41,41,237,7
5,54,92,9,62,8,50,4,91,58,11,91,
50,9,91,58,10,91,50,8,91,62,9,50
,5,91,126,35,34,2,91,7,50,6,91,5
8,5,91,61,32,50,58,4,91,61,32,24
,58,14
 170 DATA 91,71,58,12,91,79,58,1
0,91,129,5,32,252,50,10,91,42,0,
91,195,60,236,50,4,91,58,13,91,7
1,58,9,91,128,50,9,91,42,2,91,19
5,89,236,50,5,91,58,12,91,71,58,
9,91,50,7,91,58,13,91,79,197,205
,221,236,193,58,7
 180 DATA 91,60,50,7,91,13,32,24
1,58,8,91,60,50,8,91,5,32,221,58
,6,91,195,105,236,128,64,32,16,8
,4,2,1,58,142,92,238,255,71,58,1
43,92,160,71,58,8,91,230,248,111
,58,7,91,254,192,208,31,31,31,23
0,31,103,203,28,203,29,203,28,20
3,29,203,28,203,29,62,88,180
 190 DATA 103,58,142,92,166,176,
119,58,7,91,71,230,7,246,64,103,
120,31,31,31,230,24,180,103,120,
23,23,230,224,111,58,8,91,71,31,
31,31,230,31,181,111,235,33,213,
236,120,230,7,79,6,0,9,70,26,33,
6,91,203,70,40,3,176,18,201,47,1
76,47,18,201
```

THE QUEBEC LINK

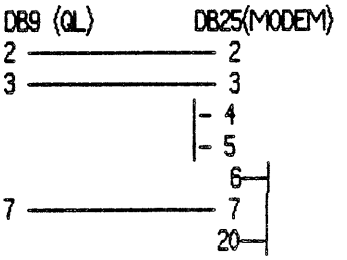
by: Réal Gagnon, 8286 St-Hubert, Montréal (Québec) H2W 2R4

ON-LINE WITH THE QL

First you need a modem. You should look for 1200 or 2400 bps. The 300 bps modem are very cheap but very,very slow. Then you need a cable. Since the QL serial port is not standard, you cannot buy a modem cable off the shelf. It is easy to build to right cable, you need:

1 DB25 male connector with casing
1 DB9 male connector with casing
1 cable with at least 3 wires

The connections are:



On the DB25 side, 4-5 and 6-20 connections are optionals. But designed this way your cable is in fact a double cable. To connect a modem, you plug the DB9 in QL's SER2 and the DB25 to the modem. But if we plug the DB9 in SER1, we have a NULL-MODEM cable that can be used to transfert from a QL to a PC (you will need a GENDER CHANGER because the serial port on the PC is MALE too, but that's another story_).

Next, you have to find a communication software. We will look at Q_LINK and QL52.

Q_LINK is a canadian software written by META MEDIA PRODUCTION from B.C. It is a shareware program, so it can copied and distributed freely. You try it, if you like it, you register directly to the company for a small price.

Q_LINK supports XMODEM protocol which is essential to download programs or files from a BBS. There is a dialing directory and an option to redial a number if it is busy, plus macro keys, on-line editor, etc_ There is no terminal emulation, this is a pure ASCII terminal! With the shareware version you get no documentation, you have to register to get a manual.

QL52 is from the QUANTA library. It is a public domain program very easy to use but has less power than Q_LINK. QL52 supports XMODEM too, but only the checksum version, Q_LINK has the CRC and the checksum (XMODEM CRC is more reliable than the checksum version). There is no dialing directory, you have to send the command directly to your modem or define an ALT KEY to do it. QL52 emulates a VT52 terminal and can display a special hebrew character set if you want.

Ok, you have your modem, cable, software and some BBS telephone numbers. Since you are a QL user, you are a "distinct computer user" from say the vast majority of PC users. You won't find any QL programs on BBS. You can write/read letters to/from other users on the BBS, join international conferences on any subject, download text files or graphics GIF files.

Habitually, files available for download are PACKED into ZIP or ARC files. This is done to combine many files into one and reduce the transmission time. Once the download is made. The file is not ready to be read or used, you have to unARC or unZIP it first! Hopefully, software to do that exists for the QL.

For ARC file, UNARC from METAMEDIA will do the job. The ZIP are handled by QL_UNZIP, written by Jan Bredenbeek of Holland.

GIF files are picture stored in a format so that a screen created on a AMIGA can be viewed on a QL screen, colors included with no problem. There is at least 1000 GIF pictures in the public domain, on any subjects from astronomy to XXX-rated picture. To view those pictures you need again a special program called SHOWGIF (COMPUSERVE). Results are stunning, you will find that your QL can show very good graphics especially in colors.

Q_LINK:
META MEDIA PRODUCTIONS, 726 West 17th
Vancouver BC, V5Z 1T9

QL52:
QUANTA LIBRARIAN, Paul Holmgren
5231 Wilton Wood Court
Indianapolis IN 46251 USA

SHAREWARE & PUBLIC DOMAIN for QL (1$ catalog on disk):
HIC LOGIQUE, PO Box 1393, Place du Parc
Montreal (Quebec) H2W 2R4

FOR THE TIMEX/SINCLAIR 2068

Entering and Recalling Text in Machine Code

By Dennis Jurries

This subroutine will allow the user to enter up to 277 characters into machine code and recover them at will. The routine consists of four parts. The first part is the machine code loader program.

```
 10 FOR x=55501 TO 55533
 20 READ z: POKE x,z
 30 NEXT x
 40 DATA 62,0,205,48,18,225,70,
35,229,33,160,215,126,35,254,128
,56,250,16,248,126,35,254,128,20
0,215,24,248,201,205,205,216,1,2
01
```

RUN the above program then DELETE it. The second part consists of six lines that allow the text you input into machine code, to be recovered.

```
 10 FOR x=1 TO z
 20 POKE 55503,x
 30 RANDOMIZE USR 55530
 40 PAUSE 0
 50 NEXT x
 60 STOP
```

The third part of the routine allows the user to put text into machine code. Once the user puts the text in and tests it, then this part may be deleted.

```
 100 LET z=1: LET j=55224: LET a
$=""
 110 PRINT "Input text (limit 32
characters  if you want to print
text on the 24th line only).  Pr
ess enter  when finished with t
ext."
 120 PRINT
 130 GO SUB 300
 140 INPUT a$
 150 IF LEN a$>=55500-j THEN PRI
NT AT 0,0;"Text too long.  Input
text.": CLS : LET j=j-LEN a$: LE
T z=z-1: GO TO 110
 160 FOR i=1 TO LEN a$
 170 POKE j,CODE a$(i)
 180 IF j<55500 AND i=LEN a$ THE
N POKE j+1,128
 190 IF j<55500 AND i=LEN a$ THE
N LET j=j+1
 200 IF j<55500 THEN LET j=j+1
 210 NEXT i
 220 LET z=z+1
 230 GO SUB 400
 240 LET z=z-1
 250 STOP
 300 PRINT "TEXT #";z;"  ";277
-LEN a$;" characters remaining."
 310 RETURN
 400 PRINT
 410 PRINT "More text? (y/n)"
 420 PAUSE 0
 430 IF INKEY$="y" OR INKEY$="Y"
THEN CLS : GO TO 110
 440 RETURN
```

The above is set up to print on the 24th line. Several different messages can be run if you add a line, 45 CLS. If you wish to print text from the top of the page, then POKE 55502,2. To change back to the 24th line, POKE 55502,0.

The following is the OP CODE for the machine code routine, that was entered in part one above. This is the fourth part of the subroutine.

```
55501 LD A,0
55503 CALL 4656
55506 POP HL
55507 LD B,(HL)
55508 INC HL
55509 PUSH HL
55510 LD HL,55223
55513 LD A,(HL)
55514 INC HL
55515 CP 128
55517 JR C,55513
55519 DJNZ B,55513
55521 LD A,(HL)
55522 INC HL
55523 CP 128
55525 RET Z
55526 RST 16
55527 JR 55521
55529 RET
55530 CALL 55501
55533 01
55534 RET
```

To save the machine code routine, type SAVE "mssg" CODE 55224,350.

PROGRAMMERS PLEASE TAKE NOTE

Would I be out of order if I were to request that ALL programmers take a thought as to where their programs might be put to use?

So many excellent programs, especially Data-Base/Address style, are of little or no value to the Canadian user. Why ?

Because our Postal Code is in the format A5A 5A5, whereas the USA Zip Code is all numbers.

Make the Address Code universally acceptable. Please? H.H.H.

MINERVA

Recently I bought MINERVA, and I like it, but only up to a point. I find that when I first switch on the QL I have to wait quite a few minutes to give it time to warm up, else I get so many varieties of the tweed pattern, and the clock crashes all over the century, I cuss Minerva all over the place.

Another thing I do not like about it, and that is the fact so many of the earlier programmes are no good any more.

Yes, it does have some good points. I like the Alt/Ctrl/Shift/Tab reset. Really great.

And I just love how much cooler the machine runs. And it is nice to have one little bit less sticking out the back, (TK2)

My version is 1.66. How do other folks get along with her?

Would I marry her again? Well-------
H.H.H.

Compare the PSION MOBIL COMPUTER to any other portable computer now in production.

Dollar for Dollar, feature for feature, it can't be beat.

Model	MC-200	MC-400	MC-600
CPU	**** 80C86 16 bit CMOS ****		
Speed (Mhz)	7.68	7.68	7.68
BIOS mfg	Phoenix	Phoenix	Phoenix
RAM Standard	128K	256K	768K
RAM Maximum	512K	512K	768K
Screen diag	10"	10"	10"
type (LCD)	SupTwist	Retardation	Film
protocol	CGA	DCGA	DCGA
resolution	640x200	640x400	640x400
col x line	85 x 14	85 x 14	???
No. of Keys	69	69	79
Detach keybd	no	no	no
Slots Stand	0	0	0
Prop	2	2	2
Battery type	8xAA Alk/Opt Rech	NiCad Pak	
life	75 hr	60 hr	40 hr
replace	yes	yes	yes
Ports Ser/Par	1/1	1/1	1/1
Video Port	no	no	yes
ROM size	256K	256K	256K
Software in ROM	GUI	GUI	MS-DOS 3.2
Touch pad	yes	yes	no
Floppy Stand	no	no	no
Floppy Ext.	no	no	1.44Mb 3.5"
Hard Disk	no	no	no
RAM Disk	OK	OK	1 Mb intern
Maximum	2 Mb	2 Mb	8 Mb
Slots	4	4	4
Fast serial transfer	***** External Option *****		
Modem	***** External Option *****		
Microphone	yes	yes	no
Speaker	yes	yes	yes
Programing language	OPL	OPL	n/a
Dimension	*** 12.4" x 8.9" x 1.9" ***		
Weight (lbs)	4.3	4.3	4.3
Warranty	******** 12 months ********		
Price	$1199	$1699	$2999

Call for current quote from
EMSoft
Box 8763, Boston, MA 02114
(617) 889-0830

Mar13, 1991

Dear Out-of-Town Members,

Before I go any further, I'd like you to take a look at the newsletter. Isn't it great! Seems to me this issue has something for just about everybody. Now, most of the credit goes to our members who have contributed all this material. But I was looking back in our old n/l's, and find that Jeff Taylor has been the Editor for exactly three years now. Seems like that was about when our newsletter took off. I wonder now.....could all of you who appreciate the n/l, drop a line to Jeff and let him know personally how much you appreciate his efforts. You can imagine that a person sometimes has need of such a lift after three years of effort!! Please do drop him a line, all of you.-Do it NOW! His address is:

335 - 75 Lemonwood Drive
Islington, Ontario
CANADA M9A 4L3

In order to get this letter out I have had to put aside some interesting programming exercises I was doing.

One of the things was replacing the "pico" menu screen compression routine in my Larken "OMNIBUS" program with the "cmpres.Bl" picture compression routine that you will see in this issue of the newsletter. More about that later.

Another project has been to try and help a member having difficulty with managing the printer on the 2068.

Then I have been experimenting with a disk management system from Bob Swoger, called Logicall.

I also did some experimenting with displaying "screen$" a la library disk #15, using the before-mentioned "cmpres.Bl" routine. And while I was about it I did some more refining of the "cmpres.Bl" program itself, to suit my fancy.

So there has been a lot going on.

When working on the printer problem I had to re-learn printer drivers. Since having the Larken system on the 2068 I had pretty well forgotten all about Tasman, Aerco, and Hacksel driver software, and had to get back into it. I have always felt that the matter of controlling a printer on the 2068 (and other computers) was somewhat akin to a "black art", more than an understood thing. I am still of that feeling, but I think it is about time we shed some light on the topic of printer control. The material I have on hand does not seem to offer much understanding into the principles. I think that with more members getting printers now, we should try to get some articles into the newsletter on the topic. Anyone interested in producing an article, or interested enough to write about what you know, for me to shape into articles? For instance, the differences between the three,(or four, if you include the Larken) printer drivers. The principles behind the need to "tokenize" or not to "tokenize" when sending out printer control characters. Why must we "point" (in addresses 26703/4) to the Aerco and Hacksel driver code, and apparently not need to with Tasman. Is it possible to use Tasman code with a Hacksel I/F? And lots more questions. Any responses?

Enough on that subject.

I mentioned getting a copy of Logicall, earlier in this letter. Logicall is Bob Swoger's effort and is designed to make it easier to manage the Larken disk system. It is not fair of me to critique this program suite; I am too biased toward Bob Mitchell's OMNIBUS to make a fair judgement. My comment would be that Logicall is somewhat elementary. Talk about damning with faint praise! Bob Swoger is trying to get the program out to all Larken users. I can send you a copy if you'd like.

Bob Mitchell has incorporated Kris Boisvert's (BYTEPOWER) "HELLO" program into OMNIBUS. I'm afraid we also modified it somewhat to suit the OMNIBUS application better. It's a super program idea, and I'd recommend every Larken owner borrow the Boisvert disk from our library, that is, disk #32.

There is also a new version of OMNIBUS, in the disk suite #2, a version suited for those users who do not have a RAMdisk. Ask for it.

Jeff Taylor wished to thank all the members who wrote and phoned, with suggestions re his Portuguese TOS disk system. About 6 members who were familiar with this system gave suggestions. I can report that it is up and running now, as a result of these suggestions.

I have been a little remiss in answering some letters from members. Too busy getting our Neighbourhood Watch organization up and running, I guess. I have been back at the computer for a while, and am getting caught up. But I would like any member who is waiting for something from me, to drop a line and ask for it again. I might have overlooked your request. I should have the outstanding letters answered by the time you get this newsletter. I would

like any member who is still waitng for something from me to drop me another line; I might have overlooked your request.

Joan Kealy asks whether the animal on the last issue was a cheetah, or was it one of our "Lynx"! She goes on to apologise, sayng, "I need a resistor in my pun line". We forgive her, but only because of the kind things she went on to say about our newsletter!

Ian Robertson, one of our in-town members (we mentioned him in the last newsletter) unpacked a Larken RAMdisk that had been sitting idle for a coule of years. Plugged it in and the programs were still intact on it. It has 4 chips on it, so that might give other owners an idea how long the batteries will last on their RAMdisks. My own RAMdisk, with 8 chips on it has the same set of batteries since December 1988. I have three cells on mine.

Back in '88 I was having problems with improper chips, and I put the third cell on to see if that would make a difference (It didn't, but I left it there anyway. I replaced the chips.

I read in past issues of NITE TIME NEWS, the Chicago area club newsletter, that some people, including members of our club, had trouble with Larken disk sytems. Seems that their old, really old TEAC drives could not read disks that were formatted at the 6 ms. head setting speed.
I cannot go into it in this letter, but if anyone is interested, I'll send you the clippings from newsletter on the nature of the problem, and the solutions.

Something else I read in other club newsletters. There was a little brou-ha-ha over Michael De Rienzo's articles. Nothing to do with us. Seems that his prefatory statement asking 'no copying without permission' riled some folks.

We are publishing them with permission. Well, Michael asked that we send him copies of our newsletters that carried his articles, which we are doing. I felt the articles were so interesting that we had to carry them. I hope you agree. There's a very interesting one in this issue, "cmpres.Bl". Try it, it's neat. It's the one I mentioned using earlier.

I have been compressing games screens onto a single Larken track using "cmpres.Bl". I then have been working on a program which would step the drive one track at a time, and load/display the compressed screen from each track in turn. A problem though. If the drive stops spinning before I request the next image, the drive will not pick up that track. Seems the drive must be spinning a moment or so before the 'track read' takes place. I solved this by having the Drive take two steps. This gave the drive time to get up to speed. What happens is that I "read" each track successively on one side of the disk, then go back and read each track on the other side of the disk. The advantage of the whole thing is that the drive does not have to go back to track 0 every time a new file (screen) is to be loaded. I shall write up an article for the next n/l.

Speaking of Michael Di Rienzo, I see in the current issue of the Seattle Area T/S UG n/l an advert of his offering a lot of TS2068 equipment for sale. Makes me think he is giving up on Timex.

One of our members, Larry Crawford, has been working on Tasword, making it more responsive to the Larken environment. Seems to have completely reworked the Basic. Looks pretty good. Anyone interested, ask me about it. I use Tasword a lot to put the HELP sections in the club library.

Some time ago I mentioned that the club had bought Peter Hacksel's printe interface remainders. There were about 10 printer interface boards. We still have three boards of the type that plug into the cartridge dock. Two of them are populated. They are not as useful as the boards which plug into the rear of the computer, since if you have a Larken system the cartridge dock is already occupied. No doubt the board could be adapted to rear use, though I'm not familiar with what's required. Anyway, you can have one of these boards for $10. You would need to make up your own ribbon cable for it.

I've run dry. Can't think of anything more, computer-wise. Shall have to close this off now. But one last reminder. Do drop a line to Jeff. And if you seem to have been overlooked by me in some request, I apologise, and do remind me.

Sincerely,

George Chambers