

SINC-LINK

Vol. 7 No. 4 July-Aug '89

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.

SEND CORRESPONDANCE TO:

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

ZX80/ZX81
TS1000/1500
PC8300
TS2068
SPECTRUM
QL
LARKIN I/F



TORONTO TIMEX-SINCLAIR
USERS CLUB

FABULOUS SUMMER ISSUE

MORE MATERIAL THAN YOU CAN POSSIBLY READ IN A SINGLE SITTING. REALLY!
THERE'S SO MUCH INFORMATION BETWEEN THESE PAGES THAT IT CAN'T ALL BE INDEXED. SO I DIDN'T TRY.
LOT'S OF GOOD STUFF FOR QL, 2068 & ZX81. SO READ ON!

EXECUTIVE OFFICERS:

PRESIDENT:	JEFF TAYLOR (244-8583)
TREASURER:	BILL LAWSON (444-8772)
SECRETARY:	GEORGE CHAMBERS (751-7559)
ACTIVITIES:	RENE BRUNEAU (531-9749)
TAPE LIBRARIAN ZX81:	" " "
TAPE LIBRARIAN 2068:	RENATO ZANNESE (635-6536)
TAPE LIBRARIAN QL:	HUGH HOWIE (634-4929)
PAPER LIBRARIAN:	GREG ROBINS (920-7747)
NEWSLETTER:	JEFF TAYLOR (244-8583)
LIAISON OFFICER:	GEORGE CHAMBERS, 14 RICHOME COURT,
(Out-of-town members)	SCARBOROUGH, ONTARIO, M1K 2Y1
	(416-751-7559)

TORONTO TIMEX-SINCLAIR USERS CLUB

14 RICHOME COURT, SCARBOROUGH, ONTARIO, CANADA M1K 2Y1

Editorial

I was sitting here trying to think of something intelligent to write when I realized what a mess I had sticking out of the back of my trusty old 2068. 2 printer interfaces, a plotter i/f, a modem connector, a graphics tablet i/f and the disk i/f. It's going to get worse, too. I still have to stick on my RS-232 i/f, my Ramdisk board (when assembled) and my digitizer board (if it ever arrives). That'll be 9 items hanging off the back! Does anyone have a suggestion as to how I can hide or remote all this stuff?

We also have a whole mess of good reading material in this issue. Once again, we have another large issue for your reading enjoyment. The articles keep rolling in. Writers, keep up the good work. It's a pleasure being the editor when I don't have to pester people to produce.

Stan, don't stop writing!

I was dismayed to read that Stan Lemke of Lemke Software Development was getting out of the TS2068 software business. Stan is best known for his Pixel Print series of programs but he has contributed many other programs to various publications over the years. 2 that come to mind are his Bit Image Graphics program and his RLE encoder/decoder program, both of which I use in the assembly of our newsletters. Like many former vendors of Timex-Sinclair products, Stan has found that due to poor response to his ads he cannot justify staying in business.

I can only hope that he will continue to offer his smaller programs to publications like this one so that users can benefit from his expertise. Stan, thanks for your support over the years and thanks for making Pixel Print public domain. That was a nice gesture.

Club Business

One of the problems of running a club during the summer is that holidays tend to get in the way. My latest attempt at arranging an Executive meeting failed for just that reason. The same goes for the QL users meeting at Hugh's place. The regular club meetings will continue even though I know that 2 or even 3 of the Executive will not be attending the July 5th get-together. My point is this; just because turnouts to the meetings drop a bit during the summer doesn't mean interest is waning. On the contrary, if article contributions are any indicator, then interest in the club has taken a fairly dramatic upturn in the past few months. Keep attending and keep sending in material.

Club Executive nominations will take place at the September meeting and elections will take place at the October meeting. Plan to attend or at least send in your nominations and votes.

Toronto BBS Has TS File Area!

As of Sunday, July 2/89, Ray Dyer's TIBM WIZARD BBS will now support a Timex-Sinclair file area. I cannot stress enough that this is an opportunity we can't let slip by.

I first contacted Ray about his TI-99/4A BBS because he was offering over 300 RLE files for downloading. Now he has provided space for us to upload and download in file area 21. Let's use it! Call Ray's 24 hour BBS at (416)-743-6703 and leave a message or a program on file.

J.T.



LISTING

BOB'S NOTEBOOK

Dot Matrix Printer Copy of SCREEN\$
(all 24 lines)

Providing your TS2068 is equipped with the LKDOS version 3 EPROM, the program listed below will do the following:

1. make a copy of any SCREEN\$ on a dot-matrix printer in normal size, ie, about the same size as a TS2040 copy;
2. copy all 24 lines;
3. store the machine code in the LKDOS RAM at address 16100, where it will park until the computer is switched off;
4. indent the left margin five spaces.

WARNING!

You MUST reset the printer each time before calling the routine, ie, you MUST switch the printer OFF and then ON. This clears the printer buffer; if you don't do this the first line of the graphics image will be corrupted.

The code can be activated at any time after it is first used, by pressing the NMI button on the disk drive interface board and then pressing the F key. In other words, after using this program the first time, you may dispense with its BASIC and load any other program and copy screen displays at will. (You will get a row of dots and dashes on the bottom row of pixels in the last line; this is due to the way Larken handles the SCREEN\$ save via the NMI button.)

The code is set up for the Smith-Corona Fastext 80 printer and you may have to substitute values to suit your own printer. I am including the Zeus source code (without line numbers) for those who need to do this. In this source code, esca=commands to set the line spacing so there will be no gaps; escm=five spaces for left margin indent; escr=line feed and carriage return; esck=commands to set the bit imaging graphics. If you change the length of any of these source code lines, you must alter the value to be then loaded into the B register, eg, if you want to indent the left margin eight spaces, you will add three more 32's against escm and then you must change the value from 5 to 8; ie, LD B,8 after LD HL,escm. I have put a note against the crucial spots in the source code listing. The

```

5 RESTORE 1000
10 FOR i=0 TO 184 STEP 2
15 READ a: READ b
20 RANDOMIZE USR 100: POKE
(i+16100),a+256*b
30 NEXT i
40 CLS
50 INPUT "drive? 0-3 ";dr: RANDOMIZE
USR 100: GO TO dr
60 INPUT "Insert correct disk and
enter full screen name. "; LINE n$
65 PRINT AT 10,0;"When screen
loaded,""restart printer and
press""NMI button + F"
70 RANDOMIZE USR 100: LOAD n$SCREEN$
100 RANDOMIZE USR 100: POKE 8214,16100
120 PAUSE 0: STOP
200 RANDOMIZE USR 100: SAVE "grafLK.Bn"
LINE 5
1000 DATA 205,129,63,175,33,152,63,6
1001 DATA 2,205,135,63,205,61,63,33
1002 DATA 0,64,6,3,14,0,197,34
1003 DATA 30,63,17,32,63,6,8,126
1004 DATA 18,36,19,16,250,197,205,40
1005 DATA 63,193,42,30,63,35,13,32
1006 DATA 230,193,5,200,17,0,7,25
1007 DATA 24,218,0,0,0,0,0,0
1008 DATA 0,0,0,0,14,8,6,8
1009 DATA 33,32,63,175,203,6,23,35
1010 DATA 16,250,205,90,63,13,32,238
1011 DATA 201,205,118,63,175,50,128,63
1012 DATA 33,85,63,6,5,126,205,90
1013 DATA 63,35,16,249,175,50,128,63
1014 DATA 201,27,42,5,0,1,245,219
1015 DATA 127,254,237,32,250,241,211,127
1016 DATA 58,128,63,61,50,128,63,192
1017 DATA 33,152,63,6,2,205,135,63
1018 DATA 24,199,175,33,147,63,6,5
1019 DATA 205,135,63,201,0,175,33,144
1020 DATA 63,6,3,126,205,90,63,35
1021 DATA 16,249,175,201,27,65,8,32
1022 DATA 32,32,32,32,13,10,0,0
1023 DATA 0,0,0,0,0,0,0,0

```

ESC A-B = paper feed 8/72

code assembles at address 55000 for loading and running at 16100 in the LKDOS RAM.

see 4-16 in printer book

SW1=ON

cont.

ZEUS Source Code
Based on article by Syd Wyncoop
written for CCATS MC SIG.

```

ORG 16100
DISP 38900
ENT
CALL narrow
XOR A
LD HL,esck
LD B,2
CALL grloop2
CALL grafic
LD HL,16384
LD B,3
enter2 LD C,0
PUSH BC
third LD (store),HL
LD DE,buffer
LD B,8
getlp LD A,(HL)
LD (DE),A
INC H
INC DE
DJNZ getlp
PUSH BC
CALL proces
POP BC
LD HL,(store)
INC HL
DEC C
JR NZ,third
POP BC
DEC B
RET Z
LD DE,2048-256
ADD HL,DE
JR enter2
store DEFW 0000
buffer DEFS 8
proces LD C,8
prlp1 LD B,8
LD HL,buffer
XOR A
prlp2 RLC (HL)
RLA
INC HL
DJNZ prlp2
CALL print
DEC C
JR NZ,prlp1
RET
grafic CALL margin
XOR A

```

254,237

6 - 2
205, ,

```

LD HL,esck
LD B,5 ; CHANGE VALUE (see text)
grloop LD A,(HL)
CALL print
INC HL
DJNZ grloop
XOR A
LD (lncnt),A
RET
esck DEFB 27,42,5,0,1
print PUSH AF
prtlp1 IN A,(127)
CP 237 ; is LP on 229
JR NZ,prtlp1
POP AF
OUT (127),A ; PRINT
LD A,(lncnt)
DEC A
LD (lncnt),A ; end of line?
RET NZ
LD HL,esck ;
LD B,2
CALL grloop2
JR grafic
margin XOR A
LD HL,escm
LD B,5 ; CHANGE VALUE (see text)
CALL grloop2
RET
lncnt DEFB 00
narrow XOR A
LD HL,esca
LD B,3 ; CHANGE VALUE (see text)
grloop2 LD A,(HL)
CALL print
INC HL
DJNZ grloop2
XOR A
RET
esca DEFB 27,65,8
escm DEFB 32,32,32,32,32
esck DEFB 13,10

```

R. H. MITCHELL
20 WILD BRIARWAY
WILLOWDALE, ONT.
M2J 2L2

890610

LARKEN INDEX.B1
A Printer Fix
by G. Chambers

SAVING MAGAZINES

el Richardson

Magazines accumulate relentlessly and we soon have no idea what's in them or even why they sit there in a pile daring us to toss them. There may be a favorite that is easily stored away each month but many have only a page or an article that makes it difficult to discard. I have found a way to deal with these.

Folded plastic sleeves called "REPORT COVERS" that have a separate spine that presses on hold the pages I want to keep. They come in several colours and the larger spine will accommodate 25 - 30 pages. Take magazines that can be cut up and a sharp knife. Carefully cut out the pages you want and fit them into a plastic cover.

I have also cut off an appropriate magazine cover to use as the first page. A sticky label on the outside identifies them as I wish. A number of these "Custom Mags" can be kept in a cardboard storage box for magazines available at stationery stores. I also have some marked with stick-on plastic tabs like those on binder dividers.

This has reduced the sheer bulk of material many times and made it all very available. One could even catalogue the stuff with PROFILE or the like.

FOR SALE! FOR SALE! FOR SALE!

- 1 - TS2068 Computer
- 1 - Fastext 80 Printer
w/Tractor feed
- 1 - Printer Interface (Tasman)
- Various books and programs
All for \$150
- 1 - TS1000 in Gladstone keyboard
w/16K RAMpack
- 1 - TS2040 Printer
- 1 - JT1115 CompuDeck Recorder
- 1 - Memopack RS232 Serial interface
- 1 - Byteback modem
- 1 - Hunter board (needs work)
- Various chips, books, and software
All TS1000 for \$75

Contact Art Johnson, Winstanly Cresc.
Toronto, Ont. M1B 1N3
Tel (416) 282 8430

On the Club Larken Disk #2 (OMNIBUS) there is a program called "index0.Cm". It is a TIME MACHINE, compiled version of a Basic program. 6 Two places in the program have a call for a printer. The program checks to see if the printer is ON. If the printer is not in place or is not turned on the program flashes a reminder to turn the printer on.

This is a neat feature, however it does cause problems. It presumes the printer will return values of 58 (small printer ON) and 237 (large printer on). My printer does not return these values. Instead the responses from my printer are 56 and 108.

The solution is to POKE new values into eight addresses in the "index0.Cm" code. The POKES in my case were:

POKE 32325,56	(old 58) small printer	58
POKE 32357,108	(old 237) large printer	229
POKE 32425,56		
POKE 32457,108		
POKE 34344,56	et cetera	
POKE 34376,108		
POKE 34430,56		
POKE 34462,108		

To find out the correct code numbers for your setup run the following lines:

```
10 LET n = IN 251
20 PRINT AT 10,10;" ";AT 10,10;n
30 GOTO 10
```

Experiment by turning the TS2040 printer on and off. Then change the program: LET n = 127 and do the same thing for the large printer. The code numbers will appear on the centre of the screen.

Verify the addresses have the old numbers shown above (you might have different vintage copy); then POKE in the numbers appropriate for your printer, and re-SAVE the program. This will need to be done with each copy of the program you happen to have.

INSOMNIAC ELECTRONICS UPDATE

"Gert and I have been able to contact Jim Horne and are helping Jim straighten things out. Jim was not keeping very good records for a while and has lost track of some of the orders sent him. If anyone has sent an order to Jim but has not heard back they should contact myself at the address on the back cover and describe their order and the manner of remittance made. Gert and I will try to have their money refunded or order filled. We'll try to help but we can't take any responsibility for any lack of performance by Jim."

The above note was printed in the Vancouver Sinclair Users Group newsletter, the ZX-APPEAL. The author of the note appears to be:

ROD HUMPHREYS
2006 Highview Place
Port Moody, B.C. V3H 1N5
CANADA

Anyone who has experienced difficulty with an order to Insomniac Electronics take note.

G.F.C.

SINC-LINK

RAMDISK to DISK BACKUP
A Larken 2068 utility

THE LATEST FROM S.N.U.G.

The national meeting was held at 7pm on Saturday May 6th 1989. About 35 SNUG members attended. Mel Nathanson was sick and unable to attend so the very able Frank Davis served as the spokesperson for SNUG.

The following is a synopsis of what went on.

Election Results - Sixty-seven ballots were mailed with forty-three returned by the deadline. The vote count was as follows:

President - Mel Nathanson
First VP - Paul Holmgren
Second VP - Basil Wentworth
Secretary - Audeey Curnutt
Treasurer - John Cushran

Congratulations to all those who were elected and many thanks to those who ran but weren't elected. Better luck next time. Thank you for your dedication to making SNUG and the whole Sinclair community a viable entity. If you are unaware of who was running here are all the candidates, with the names of all who withdrew removed.

President - Joan Kelly, Gary Lessenberry, and Mel Nathanson.

First VP - Alex Burr, David Hartman, Ted Hechman, Paul Holmgren, and Donald Lambert.

Second VP - Frank Davis, Ruth Fegly, Laurie Futrell, Frank Orosz, and Basil Wentworth.

Secretary - Mike Bowers, Audrey Curnutt, Andrew Hradesky, Gary Lessenberry, and William Wood.

Treasurer - Chris Crawford, John Cushran, and Ken Frankel.

Thanks again from SNUG.

After the election results were announced we discussed: What is SNUG? What seemed to be the driving idea was that SNUG must be an information clearing house and be able to disseminate Sinclair information to all members of the Sinclair community. SNUG should help with getting our annual fest in shape, spreading the Sinclair news (i.e., new programs, books, and hardware), and offering help to any group or individual needing it.

Finally Paul Holmgren is keeping a cross reference of all newsletters in the Sinclair community. If you are interested in it contact Paul via the Indiana User Group, ISUG, 513 E. Main St., Peru IN 46970.

If you want to contact anyone in SNUG their address is SNUG, 7515 Arbordale Dr., Port Richy, FL 34668.

Retyped from the May 1989 issue of SMUG Bytes, the Sinclair Milwaukee User Group newsletter.
G.F.C.

Larry Kenny has sent me a listing which will be of interest to RAMdisk owners. Usually we sort of store programs permanently on the RAMdisk and never think of other possibilities. With a program which makes it easy to save it's contents, the RAMdisk can readily be used for other things.

One thing I can envision is it's use with the Larken Sequential filing program. One could store a complete file of 22 tracks onto the RAMdisk (temporarily), in order to carry out rapid searches, etc. on the Sequential file program, without the delay of loading another track from a disk drive.

Anyway, here's Larry's listing.

-7001

G. Chambers

```

7010 REM  RAMDISK to DISK BACKUP          by L
ARKEN ELECTRONICS
20 BORDER 7: INK 0: PAPER 7: CLS : PRINT AT
5,5;"LARKEN RAMDISK BACKUP";AT 8,3;"SAVE RAMDI
SK to DISK (S)";AT 10,3;"LOAD RAMDISK from
DISK (L)";AT 13,8;"Press S or L"
30 PLOT 0,50: DRAW 0,100: DRAW 255,0: DRAW 0
,100: DRAW 255,0
35 PLOT 2,52: DRAW 0,96: DRAW 251,0: DRAW 0
,96: DRAW 251,0:
40 LET a$=INKEY$: IF a$="L" OR a$="1" THEN
GO TO 100
50 IF a$="S" OR a$="s" THEN GO TO 200
60 GO TO 40
80 BEEP .1,20: INPUT "NUMBER of Ramchips ? "
;nc: IF nc>8 THEN GO TO 80
90 RETURN
100 PRINT AT 10,3: FLASH 1;"LOAD": GO SUB 80
105 PRINT #0;"Move Ramdisk switch to UP, ok?"
: PAUSE 0: INPUT :
110 RESTORE 500: FOR a=1 TO nc
115 READ x: OUT 7,(x+64): OUT 244,240
120 LET n$="R,Bank.C"+STR$ a: PRINT AT 18,3;"
LOADING "; FLASH 1;n$: RANDOMIZE USR 100: LOAD
n$CODE 32768,32768
140 NEXT a
150 RANDOMIZE USR 100: GO TO 4: RANDOMIZE USR
100: CAT ""
170 GO TO 300
200 PRINT AT 8,3: FLASH 1;"SAVE": GO SUB 80
210 PRINT #0;" Press any key when ready": PAU
SE 0: INPUT :
220 RESTORE 500: FOR a=1 TO nc
230 READ x: OUT 7,x: OUT 244,240
250 LET n$="R,Bank.C"+STR$ a: PRINT AT 18,3;"
SAVING "; FLASH 1;n$:
260 RANDOMIZE USR 100: SAVE n$CODE 32768,3276
8
280 NEXT a
300 OUT 244,0: PRINT #0:TAB 6: FLASH 1;">> AL
L DONE <<"
310 BEEP .4,10: BEEP .4,1: IF INKEY$="" THEN
GO TO 310
320 STOP
400 RANDOMIZE USR 100: SAVE "RBAKUP.B1" LINE
1
500 DATA 7,3,5,1,6,2,4,0

```

QLIPS

It has happened to me so often, as I am sure it has happened to you. You are working away and your concentration is all to the task in hand, you make an error and for some reason you have a mental block.

No matter what-ever it is, you decide that 'LIST' is what you should enter, instead you enter 'RUN'. Now this may not be too bad in many cases, but say you are making a Back-up of a cartridge or disc, this can be fatal, as the first thing your computer wants to do, is format something, and if you should happen to have the cartridge or disc in the wrong device, bang goes a whole stack of work.

Another thing that can happen is that instead of entering 'List' or 'CLS' or 'NEW', your hand drifts to the right of your key-board, and as soon as it gets there, your finger is so close to the 'RESET' button, your fingers curl round the edge of your key-board, and a little pressure and away goes something else.

One of my various hobbies is working with WOOD. Now, when you work with wood, it is the same as cutting hair. The old saying goes that they are not cutting hair any longer now-a-days. NO SIR. They're cutting it SHORTER. Same as wood, you can not cut it longer, and it is rather difficult to join two pieces of wood together so that the join is concealed, so there is an old saying in carpentry to cover this. 'Measure twice---cut once'. A very good motto for all computerists to follow.

Keep typing. When there is the least element of doubt, take a second look at what you would like, but before you act, take your hands away from that key-board, and take a second look at what you THINK you would like, or what you SHOULD do.

Nine times out of ten, this little action of taking your hands away from the board, will prevent you from making a costly error.

THINK TWICE ----- ACT ONCE

Come to think of it, if I followed my own advice, I would think twice and speak once, might save a lot of hassle at times !

H. H. H.

FOR THE "QLERS" USING SPREADSHEET

When using the Psion Spreadsheet, after each entry, you have to ENTER and then use the arrow cursor keys to go to the next entry. Real Gagnon in the issue #8 of QL_DOC, Feb and Mar 1989, has placed in the public domain a program to reduce the necessary keypresses to ALT and the arrow cursor. I have translated the program, listed below, and I have used it to enter some data into the SPREADSHEET. After entering the program, you have to run it and indicate where you desire to save the end result. e.g. the "ABACLAV.exe" in my case, I used FLPI_, but you may choose to save it to MDV1_, or whatever. Then the boot program must be amended to load this option. I entered line" 8 EXEC flpi_ABACLAV_EXE " AFTER moving line 9 to 10, and line 8 to 9. The purpose of course is to make room for the new line 8.

The listing is shown ^{above} ~~below~~, it is also available on a cartridge from Hugh Howie.

Louis Laferriere

```

100 REMark Creating a file ABACLAV_EXE
110 REMark for QL_DOC #8 by Real Gagnon Mtl Que. 89
120 REMark This program is public domain (Translated by Louis Laferriere)
130 :
140 CLS:prog$="ABACLAV_exe"
150 INPUT "Where to save "&prog$ (e.g. flpi_)?":save$
160 IF LEN(save$)=0:save$="flpi_"
170 PRINT "One moment s.v.p."
180 :
190 RESTORE 340:a=RESPR(1000):basr=a
200 count=0
210 REPEAT 1
220 IF EOF:EXIT 1
230 AT 3,0:PRINT"> "
240 READ x:POKE a,x:a=a+1
250 AT 3,0:PRINT " <";
260 count=count+1
270 END REPEAT 1
280 :
290 PRINT "One keypress to save "&save$&prog$("&count$ bytes)"
300 PAUSE
310 SEEXEC save$&prog$,basr,count,128
320 PRINT "OK!"
330 :
340 DATA 96,14,0,0,0,0,74,251,0,6
350 DATA 65,66,65,67,76,65,112,11,114,255
360 DATA 116,1,78,65,145,200,67,250,0,150
370 DATA 54,120,0,208,78,147,112,0,78,65
380 DATA 54,40,0,138,12,67,0,201,103,0
390 DATA 0,74,12,67,0,217,103,0,0,34
400 DATA 12,67,0,193,103,0,0,42,12,67
410 DATA 0,209,102,0,0,94,97,0,0,68
420 DATA 18,68,0,208,97,0,0,70,96,0
430 DATA 0,46,97,0,0,52,18,68,0,216
440 DATA 97,0,0,54,96,0,0,30,97,0
450 DATA 0,36,18,68,0,192,97,0,0,38
460 DATA 96,0,0,14,97,0,0,20,18,68
470 DATA 0,200,97,0,0,22,49,124,0,0
480 DATA 0,138,96,0,255,152,18,68,0,10
490 DATA 97,0,0,4,78,117,36,104,0,76
500 DATA 54,120,0,224,78,147,78,117,47,8
510 DATA 112,8,114,255,118,5,147,201,78,65
520 DATA 32,95,96,0,255,112,0,42,67,108
530 DATA 97,118,105,101,114,32,65,66,65,67
540 DATA 85,83,32,131,99,114,105,116,32,112
550 DATA 97,114,32,82,131,97,108,32,71,97
560 DATA 103,110,111,110,32,81,99,56,57,10
570 DATA 0,0

```

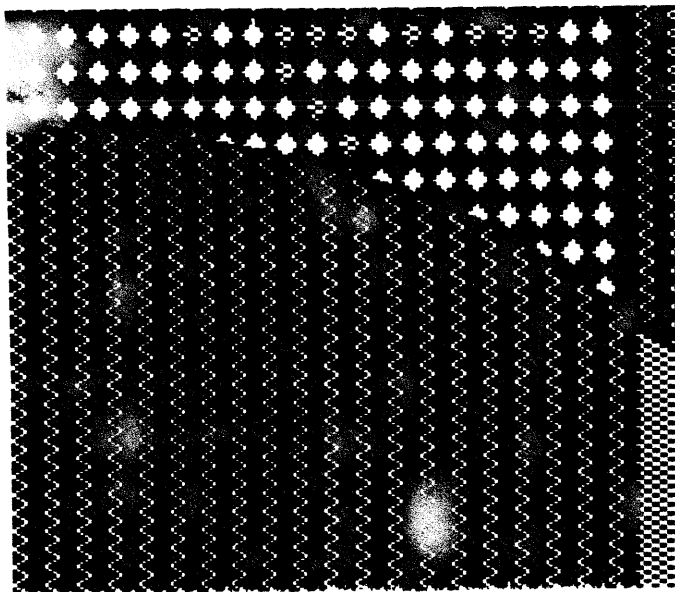
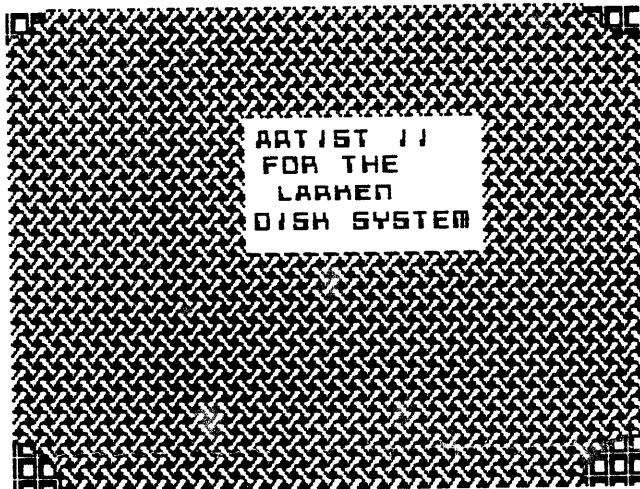
Lcopy is a public domain program that will enable to make copies using LKDOS. This program first appeared in a issue of Zx Monthly and was slightly modified to be used with LKDOS.

The program is all in basic, a 93 byte section of machine code is poked into the memory starting at location 60245. This program is very fast and prints out great pictures.

The program has been set up to use 2 disk drives. You can delete line 12 and then edit line 6, if you have only one disk drive.

With this program you can select double strike, seven different widths and four different heights. To save this program enter GOTO 9000.

See sample printout. (photo-reduced)

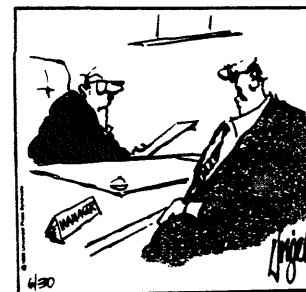


```

1 REM Larken Screen
Dump
2 REM a program by T. Andrews
3 REM modified for LKDOS by
4 REM Renato Zannese 06/89
5 REM
6 CLS : PRINT ""Insert Graphic Disk
into DRIVE 1"
8 PRINT ""Press a Key to Start"
10 PAUSE 0
12 RANDOMIZE USR 100: GO TO 1
15 RANDOMIZE USR 100: CAT
18 INPUT "FILENAME(+EXT):";AS
20 IF LEN AS>9 THEN GO TO 15
25 IF AS="" THEN STOP
35 RANDOMIZE USR 100: LOAD ASSCREENS
40 RANDOMIZE USR 100: POKE 16093,32: RE
M turn off tokens
50 LET is=CHRS 27+"J"+CHRS 20: REM line
feed = 20/216"
60 INPUT "Double Strike? (Y/N)";SS
70 LET st=1+(ss="Y")+(ss="y")
80 INPUT "Width (1 to 7)";w
90 INPUT "Height (1,2,4,8)";h
100 POKE 23259,h
110 POKE 23258,w
120 FOR f=175 TO 0 STEP -8/h
130 LPRINT is+CHRS 13: REM linefeed plus
carriage return
140 FOR t=1 TO st
150 LPRINT CHRS 13
160 POKE 23261,f
170 LPRINT CHRS 27+"Z"+CHRS 0+CHRS w;: R
EM quad density
180 RANDOMIZE USR 60245
190 NEXT t
200 NEXT f
210 STOP
9000 RANDOMIZE USR 100: SAVE "Lcopy.B1" L
INE 9100
9005 STOP
9100 RESTORE 9900
9105 CLEAR 60000
9110 FOR n=60245 TO 60336
9115 READ a: POKE n,a
9120 NEXT n
9900 DATA 33,220,90,54,0,33,222,90
9905 DATA 54,0,62,1,33,219,90,70
9910 DATA 14,16,203,9,184,40,4,203
9915 DATA 8,24,247,65,197,33,221,90
9920 DATA 126,144,71,33,220,90,78,205
9925 DATA 170,34,71,4,126,7,16,253
9930 DATA 230,1,33,219,90,70,33,222
9935 DATA 90,245,134,119,203,14,241,16
9940 DATA 248,193,16,216,62,3,205,1
9945 DATA 22,33,218,90,70,197,58,222
9950 DATA 90,215,193,16,248,33,220,90
9955 DATA 52,32,170,201
9960 RUN

```

HERMAN



"I can start work Monday if
I don't win the lottery."

THE AMAZING DIGI₂DUBBER
by Fred Nachbaur

ZX81

I'm sure that the VSUG finds itself in a bit of a predicament. I loaned my entire collection of ZX81 programs, pictures and other files. How would one librarian make copies of well over 500 files without growing old before the project is completed?

In these days after the goldrush, we could all benefit from a quick and painless way of making copies of library tapes. There are several options:

- 1) Use an attenuator cable (available from Radio Shack) to copy from one deck to another.
- 2) Use a home stereo or "boombox" double cassette deck to make an audio dub.
- 3) Load each program into the computer, then save it back to another tape.

Option #3 is by far the most reliable. The reason is that audio dubs don't really do justice to digital signals. If you scope the output of a typical cassette deck, you'll find that the amplitude of the pulse train is by no means constant. This "sagging" effect becomes more pronounced when making an audio copy. Furthermore, dropouts tend to be aggravated, often to the point where the copy simply doesn't work.

What we need is an "attenuator cable" that emulates the behavior of the computer in saving/loading tape signals. DIGI₂DUBBER does exactly that. The signal on copy made using DIGI₂DUBBER is indistinguishable from a signal generated by the ZX81 directly.

Figure 1 shows the schematic diagram of DIGI₂DUBBER. Here's how it works.

Resistor R1 limits the current into the input transformer T1, and makes the input to the DIGI₂DUBBER look mostly resistive (omitting it caused bizarre spurious responses, presumably ..(faulty n/1 copy here)

Transformer T1 does several things. Most importantly, it isolates the two cassette decks, avoiding any possibility of 'ground loop' problems. It also boosts the signal voltage, helping to ensure that we can process the analogue signal into a good approximation of a (digital) square wave. Further, it's limited frequency response is used to advantage, to help prevent both low-frequency and high-frequency noise from reaching the recording deck.

The signal from the T1 is then half-wave rectified using diode D1 and resistor R2 (or D2 and R3). The use of a rectifier on both halves of T1's secondary keeps the current (and therefore the current reflected back into the primary) symmetrical, reducing distortion due to non-linearity. Switch S1 selects the 'input phase'; in other words, whether we are selecting the positive or negative excursions of the signal.

The reason for this switch is that most decks will falsely give an extra pulse in each pulse-train in one phase (usually but not always the negative phase). This corruption

leads to later difficulty in loading. I'll describe later how you determine the proper setting for your decks.

The rectified signal is then clipped by the resistor R4 and diode D3. Across D3 we now have a constant-amplitude square wave (or reasonable facsimile thereof).

Finally, capacitors C1 and C2 and resistors R5 and R6 provide a two-pole filter that matches the transfer characteristics of the similar network in the ZX81's SAVE circuitry. Going to the dubbing deck we thereby have a signal that 'looks' as if it were being sent by a stock ZX81.

Note that the DIGI₂DUBBER is completely passive; it does not need a source of DC power, as did my earlier 'load conditioner' designs.

BUILDING DIGI₂DUBBER

Construction is straight-forward. I built mine on a piece of perf board. Transformer T1 is readily available from Radio Shack and elsewhere. It is a 500 ohm (centre-tapped) to 8 ohm "transistor audio output transformer", used in reverse. I used a sub-mini version rated at 0.1 watt, but just about any size should work fine.

The diodes are the ubiquitous 1N914A/1N4148 glass silicon types. Radio Shack sells them at 50 for \$3 (this is even better than the prices at most wholesale jobbers). The resistors and capacitors are similarly 'garden variety'.

Switch S1 can be virtually any kind of SPDT switch. I used a push-button switch salvaged from a 'dead body' in the junk room of the stereo shop where I work.

Note that the "MONITOR" cable is simply an extension of the input cable. This is so that you can simultaneously load the program, or use a "header reader" program like PNR to make a listing using DIGI₂DUBBER.

DIGI₂DUBBER is intended for copying tapes. It is not meant as a load conditioner, or a panacea for lousy tapes. However, if the original tape loads normally into a stock ZX81 computer, DIGI₂DUBBER will insure that the copy will also.

Connect the INPUT plug to the EAR jack of the source recorder. Connect the OUTPUT plug to the MIC jack of the deck being used to record the copy. Optionally, connect the monitor jack to a stock 2K TS1000 (running PNR) to get a listing of program names and SAVE/LOAD times.

Set the playback deck's volume to about the same level as you normally use to load tapes. When using DIGI₂DUBBER, it is better to err on the direction of less signal. I found settings of between 5 and 7 (on a JIL JT1115) to be ideal. Too much clipping (i.e. high volume settings) may compromise the integrity of the copy by generating spurious pulses.

Start the playback deck with the source tape, and the recording deck with an object tape of the same or greater length, and away you go.

The first time you use DIGI₂DUBBER, you will have to determine which of the two switch

cont.

positions are appropriate for your setup. The best way to do this is with an oscilloscope. First make an original tape from a stock ZX81. SAVE a large blank array; e.g. DIM A\$(12000), then SAVE "array". This will give a large section of tape with only 0's on it, making it easier to see on the 'scope.

Connect the 'scope's input across diode D3. Play back the portion of the tape with the steady-sounding array data, and adjust the 'scope's vertical gain and horizontal frequency for a steady image.

Experiment with the switch in both positions. On most decks you'll find that in one position you will get four pulses in each pulse train, with a level that's largely independent of the input level. This is the correct setting. In the other position you'll count five pulses per train, or you will count four pulses over only a narrow volume range.

If you don't have a 'scope, set up the system as described above and connect an earphone to the EAR jack of the recording deck. Put the recording deck into the RECORD mode, and play a tape on the playback deck. Flip the switch while listening to the signal. In one position the signal will seem louder and may appear to have a 'lower' pitch; this is the wrong position. In the other position, the signal will seem clearer but a little quieter; this is the proper position. Make a note of this position.

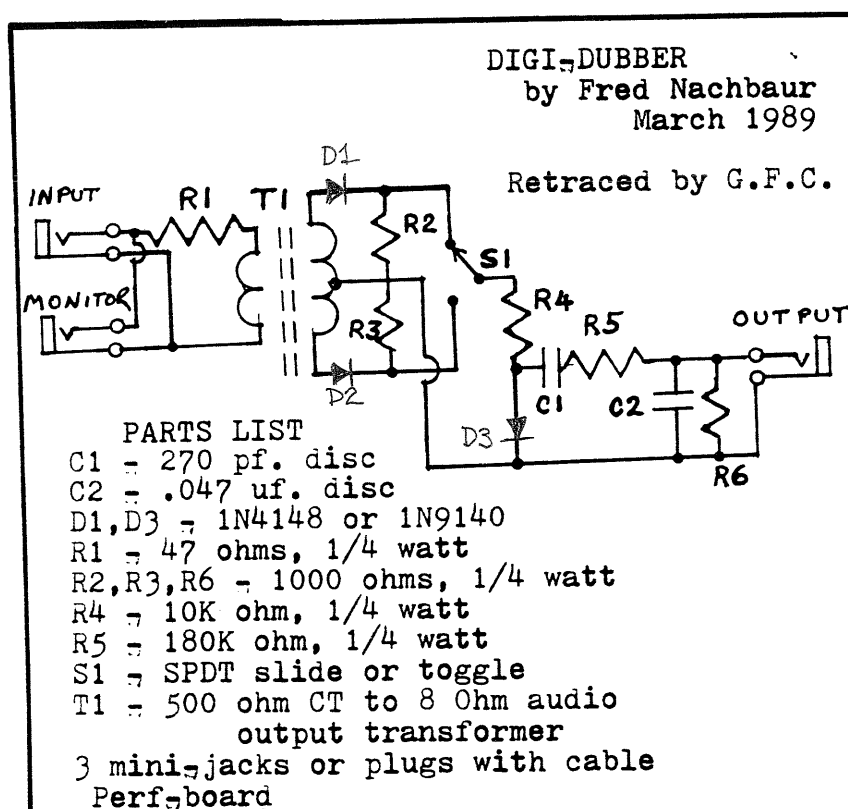
Setting the optimum playback level can also be done using the earphone/monitor approach (or by turning the monitor switch ON on a deck a deck that allows monitoring through the speaker). As you increase the volume from zero, there will be a relatively large range (typically from about 4 to 7) over which the signal sounds essentially the same. Beyond this, it will sound noticeably "harsher", due to clipping. Again, you could use a scope for a more scientific approach. Just below the point where this clipping starts is the ideal setting. Your recordings at this setting will be undistorted, and relatively immune to dropouts.

One final note: When making copies of library tapes, or any tape for that matter, be careful with head azimuth alignment. Align the recording deck to a standard test tape, or failing that, to a GOOD QUALITY pre-recorded tape. The noisier the better; heavy metal is great for setting azimuth because of all the high-frequency noise.

Adjust the playback deck's head to the tape being copied. It is therefore good practice to always use the same decks for play and record; the recode deck's azimuth alignment will be "known good", and the playback deck's setting will "slop" as required to match the source tape.

Taken from the ZXAPPEAL newsletter

Retyped by G.F.C.



Use DIGI-DUBBER to make high quality duplicates of ZX81/TS1000/TS1500 tapes. Wire it up on a small piece of perf-board or other support. MONITOR jack optional.

QL

QL CURES Bill Lawson

REGULAR CRASHES

1. TRY THE QL ON IT'S OWN. THAT IS WITH NOTHING, AND I DO MEAN NOTHING, PLUGGED IN. IF IT CRASHES, THEN PUSH ALL THE PLUG-IN CHIPS DOWN IN THEIR SOCKETS. IF YOU CAN MANAGE IT, EASE EACH ONE UP AND RESEAT.
2. FIT A NEW 7805 REGULATOR, PREFERABLY THE 78S05 OR THE LM340T WITH HEATSINK COMPOUND.
3. MOVE THE QL AWAY FROM THE TV/MONITOR ON BAKING FOIL WITH IT UP THE SIDES. DON'T PUT IT ACROSS THE SCREEN!

CRASHES WITH ADD-ONS

1. ADD-ON MEMORYBOARDS HAVE THE CHIPS IN SOCKETS SO ANY POOR CONTACT FROM POOR SOCKETS MEANS A LOCKED UP 200 PAGE THESIS IN QUILL. PUSH ALL THE CHIPS DOWN IN THEIR SOCKETS TO CURE IT. THIS APPLIES EQUALLY TO BRAND NEW VIRGINAL BOARDS. DO IT BEFORE YOU HAVE A CRASH, NOT AFTERWARDS.
2. LOOK AT THE VOLTAGE REGULATOR AND HOW IT IS COOLED. ON SOME BOARDS THE REGULATOR IS NOT IN GOOD CONTACT WITH THE HEATSINK OR CASING AND HAS NO THERMAL COMPOUND EASE THE REGULATOR UP, INSERT THE NASTY GUNGE AND SCREW DOWN. THE REGULATOR IS DESIGNED TO SHUT DOWN IF THE INTERNAL CHIP TEMPERATURE EXCEEDS A CERTAIN VALUE. THIS IS ONE ASPECT IT DOES VERY WELL WHEN IT ALSO SHUTS DOWN YOUR SOFTWARE. FRAYED TEMPERES ALL ROUND.
3. EARLY VERSIONS OF EPROMS WERE NOT WIHOUT THEIR LESS APPEALING ATTRIBUTES AND PERHAPS TONY TEBBY MAY AT SOME TIME STEP US THROUGH THEM. I KNOW MIRACLE DOES A 'MANUAL' FOR THE TRUMP CARD BUT WHERE DO I BUY THE STRONG GLASSES TO READ IT.
4. IF YOU HAVE AN EXPANSION PLUGGED IN WITH ANOTHER EXPANSION PLUGGED IN TO THAT THEN ARRANGE IT SO THAT THE DANGLEY BITS ARE SUPPORTED. THE CONNECTORS ARE JUST CONNECTORS, NOT SUPPORTS FOR PART OF THE FORTH BRIDGE.
5. LOOK AT YOUR EXPANSION CARD TO SEE IF IT HAS A PAL CHIP ON IT. IT IS PROBABLE THAT THIS PAL CHIP IS NOT WHOLLY COMPATIBLE WITH THE PARTICULAR 8301 IN YOUR QL. THERE ARE AT LEAST FOUR 8301'S, ALL HAVING DIFFERENT CHARACTERISTICS. THERE IS ONLY ONE ZX8301 AROUND NOW, SO IF YOUR PAL IS NOT PALLY WITH THE LATEST 8301, HARD CHEESE.
6. TAKE ALL THE GUTS OUT OF THE QL CASING AND SPRAY THE INSIDE WITH NICKEL SCREENING SPRAY, TOP AND BOTTOM. DO THE SAME ON THE POWER SUPPLY CASE AND ON ADD-ON CASES. WHEN DRY, COAT WITH LACQUER BEFORE RE-ASSEMBLY. IT STOPS THE RANDOM ZAP FROM STAR WARS EXPERIMENTS.
7. USE A POWER SUPPLY THAT FEEDS A SMOOTH LOW VOLTAGE TO THE QL WHICH KEEPS THE WHOLE THING COOL. IT CURES 99% OF ALL KNOWN ILLS.

QL DON'T WORK

SYMPTOM. PLUG THE POWER SUPPLY IN THE QL AND NOTHING.

cont.

1. CHECK YOU HAVE THE POWER AT THE SOCKET, THE POWER SUPPLY IS PLUGGED IN TO THE MAINS AND IT IS PLUGGED IN TO THE QL. IS THE POWER ON LED LIT? IS THE POWER SUPPLY HUMMING OR BUZZING? IF NO THEN CHECK THE MAINS PLUG CONNECTIONS. THE THREE PIN-MAINS PLUGS WERE VERY WELL DESIGNED TO ENSURE IT UNSCREWED ITSELF WHILST NO ONE WAS NEAR IT. GET THE SCREW OUT AND SCREW IT UP. BETTER STILL SOLDER IT UP AND BE SURE.

2. MAINS TO THE POWER SUPPLY BUT NO BUZZ. UNPLUG, I REPEAT UNPLUG AND OPEN THE CASE. IT WILL SMELL OF OVERHEATED NEARLY CHARRED THROUGH ELECTRICS BECAUSE IT GETS DAMNED HOT. IF YOU LOOK CAREFULLY YOU WILL FIND TWO THERMAL FUSES. MAPLIN STOCK THESE BUT YOU CANNOT SOLDER THEM IN BECAUSE THE HEAT FROM THE SOLDERING IRON BLOWS THEM. YES YOU CAN IF YOU ARE CAREFUL. CLAMP ON A BIG HEATSINK AND SOLDER ON THE OUTSIDE. IN PRACTICAL TERMS, POP A PAIR OF PLIERS IN THE DEEP FREEZE OVERNIGHT TO BE USED AS A HEAT SHUNT, SPRAY THE FUSE WITH FREEZER SPRAY AND BE QUICK AT SOLDERING. ONE THERMAL FUSE IS IN THE SO CALLED 9 VOLT LINE WHILST THE OTHER IS IN THE AC LINE. NOW YOU KNOW WHY IT IS FAIRLY SAFE TO LEAVE THE QL ON FOR WEEKS AT A TIME. IF IT GETS TOO HOT THE REGULATOR SHUTS DOWN AND THE TWO THERMAL FUSES BLOW.

3. WITH THE POWER SUPPLY BUZZING, CHECK THE LEAD TO THE QL BY WRIGGLING IT. BETTER STILL PUT A VOLTMETER BETWEEN THE CENTER PIN AND THE OUTSIDE. OK SO YOU HAVEN'T GOT A VOLTMETER. STICK A SMALL 6 VOLT BULB ACROSS AND SEE IF IT LIGHTS UP. IT DOES, GO TO THE NEXT STAGE.

4. LIFT THE TOP OF THE QL, LIFT OUT THE MEMBRANE TAILS AND MAKE SURE THE SIX LED TAILS ARE IN PLACE. LOOK AT THE 7805 REGULATOR BEHIND THE MICRODRIVES, REMOVE THE SCREW, TOSS THE 7805 IN THE BIN, FIT A NEW ONE WITH THE GUNGE AND POWER UP. BINGO, THE YELLOW LED IS ON BUT THE SCREEN IS BLACK OR WHITE BUT CERTAINLY NOT 'TWEED'. OH, IT IS 'TWEED'! YOU HAVE CRACKED IT. ALL THAT IS REQUIRED IS THAT YOU FIT THE MEMBRANE TAILS BACK READY FOR THE SCREWS TO BE FITTED. IT IS ABSOLUTELY CERTAIN THAT IF YOU PUT THE SCREWS BACK BEFORE A FINAL CHECK, THE KEYBOARD MEMBRANE WILL FAIL AT THE FIRST KEY PRESS. IT WILL FAIL ANYWAY DUE TO YOU DISTURBING IT AND YOU WILL HAVE TO REPEAT THE EXERCISE.

5. THE SCREEN IS BLACK, WHITE, SPOTTED OR OTHERWISE NON 'TWEED'.

IF IT IS ON A COLOUR MONITOR, LOOK TO SEE IF THE WHITE SCREEN APPEARS AT THE VERY INSTANT YOU PLUG IN, OR IF THE BLACK SCREEN NEVER BRIGHTENS SLIGHTLY OR THERE IS ANY VISIBLE CHANGE. REMOVE THE ZX8301 OR WHATEVER YOUR NUMBER IS AND FIT A NEW ONE.

CLEVER CLOGS CAN POKE A 15M Ω OSCILLATOR ON TO THE APPROPRIATE PIN TO REVIVE THE CHIP. IT WORKS SOMETIMES. THOMAS BENT CLAIMS A 100% CURE. CAN WE ALL HEAR IT THOMAS?

6. THE POWER LEAD IS PLUGGED IN THE QL WHEN AFTER A SECOND OR SO A WHITE OR GREEN SCREEN APPEARS. FAULTY RAM. THIS CANNOT BE FIXED ON THE KITCHEN TABLE.

7. THE 'TWEED' START UP PATTERN ALSO F1/F2 APPEARS BUT THE FRAME WILL NOT LOCK. FRAME SYNC COMES FROM THE MC 1377P BEHIND THE MICRODRIVES. IT NEEDS 12 VOLTS TO WORK SO TRACE THE SUPPLY THROUGH THE SMALL 7812 REGULATOR NO POWER, NO WORK. NEITHER DOES THE SERIAL PORT AS THEY NEED +12 VOLTS AND -12 VOLTS.

8. SOME SNOW ON THE SCREEN WHEN USING A TV BUT NO PICTURE. OCCASIONALLY THE MODULATOR IN THE METAL CAN FAILS, AS DOES THE MC1377P WHICH IS FED LARGE DOSES OF HEAT REGULARLY. IT IS A SIMPLE PLUG IN CHIP.

9. MONO MONITORS ARE FED THROUGH A SINGLE TRANSISTOR. IT FAILS OCCASIONALLY. TRY THE TV TEST FIRST.

SWITCHING REGULATORS CREATES LARGE AMOUNTS OF RADIO FREQUENCY INTERFERENCE. THEY USUALLY STOP MICRODRIVE 2 FROM WORKING AND CORRUPT DATA FROM MICRODRIVE 1. I REPLACE THEM WITH A 78S05 EVEN IF THEY ARE NEW. IT HAS CURED OVER TEN QL'S SO FAR, SO THERE MUST BE SOMETHING IN IT.

RUBBISH THROUGH THE SERIAL PORTS OR PART RUBBISH HAS BEEN TRACED ON SEVERAL OCCASIONS TO ANTI-BOUNCE CHIPS. I HAVE HAD SEVERAL THAT WORK OK BUT IT IS A POINT WORTH REMEMBERING IF YOUR PRINTER/MODEM GOES A BIT FUNNY.

DENNIS BRIGGS (11/11/88)

QUANTA VOLUME 5 ISSUE 12, JANUARY 1989 W K (BILL) LAWSON

QLips
by H. Howie

It is astounding where all the dust and lint comes from, from the air, from your clothes, even from your hair, call it Dan Druff. That is, if you have any hair left. One thing for sure; where you have electricity, you have static, and where you have static you have a magnet which draws dust like a son of a gun. And where you have computers and Hi-Fi and such, then you and I both know, you have trouble, sooner or later.

Question...Does a bald-headed bearded man have dandruff?

The QL being black, shows up this dust so prominently it is hard to avoid seeing it. Monitor screens pick it up and look fuzzy if they are not kept clean. Same with TV.

An excellent way to cut down on this fuzz is to take a soft cloth, spray a little Armor-All on it and gently wipe over your equipment. Make sure you switch it all off before you start, one can never tell when a little static can cause something to blow. Wiping can cause this static, and so can dust. This is one good reason to use something to reduce static, and silicon based compounds are good for this.

Don't forget to go over the printer. A little bottle of Air from a camera shop, will blow all the dust away which is gathering around the head area...Keep those moving parts free of dust.

Keyboards are more difficult. I use a 1 inch paint brush with soft bristles, spray a little Armor-All on the brush, and whisk away the dust, wipe with a soft cloth after.

Keyboards are a very sensitive area, and it is not unknown for them to become 'sticky'; once again cleanliness is a must. I am not certain of this, but I have heard it said, (or did I read it?) that Tom Bent, of QL fame, says that a little WD 40 can work wonders for a keyboard. I have not tried it myself, but I should imagine it would work pretty good.

And on that note, after you clean up your equipment that is just how it looks.

Pretty Good
H.H.H.

More RLE Graphics

This lovely lady was downloaded from the RLE file area of the TIBM WIZARD BBS (416)-743-6703. The file was then decoded using Stan Lemke's RLE Decoder program. This program is available in the club disk library to club members. This copy was printed on a TS2040, the picture inside the front cover was done with a Commodore 1520 plotter, a McMichael interface and a TS2068. Aren't graphics neat? There are lots more where she came from, so dust off that modem and get BBSing.

J.T.



Catherine
Deneuve

Peanuts



Provided to Sinc-Link by Bill Jones "UPDATE"

LKDOS TOOLS by the Half Dozen

```
100 REM "LKDOS Print Tools #1-5"
102 BORDER 0: PAPER 0: INK 7: CLS
104 CLEAR : LET a=0: LET b=0: LET u$=""
```

```
9800 REM #1. Initialization Routine
9804 RANDOMIZE USR 100: OPEN #3,"lp": PR
INT #0:"Input Type of Interface" "<1> Aæ
rco-Oilger"" "<2> Tasman"" "<3> A & J": PA
USE 0: LET zz=CODE INKEY$-49: IF zz>2 TH
EN CLS : GO TO 9804
9806 RANDOMIZE USR 100: POKE 16096,zz
9808 INPUT "Line Width?";Lw: RANDOMIZE U
SR 100: POKE 16090,Lw-1
9809 INPUT "Left Margin Tab?";Lm: RANDOM
IZE USR 100: POKE 16094,Lm
9810 INPUT "<1> for Line Feed - <2> No L
f";Lf: IF Lf<1 OR Lf>2 THEN GO TO 9810
9812 RANDOMIZE USR 100: POKE 16092,(10 A
ND Lf=1)+(0 AND Lf=2)
9814 PRINT #0:"Turn OFF PRINTER, Re-Set
Printerswitches, then TURN Back ON. E
NTER when Done": PAUSE 0: LPRINT : CLS
9818 FOR n=1 TO 150 STEP 5: LPRINT "xxxx
";: NEXT n: LPRINT : REM Test print lin
es
9820 STOP
9822 REM #2. CAT PRINT * Print a Disk C
atalog
9824 CLS : INPUT "Drive #";dn: RANDOMIZE
USR 100: GO TO dn
9826 RANDOMIZE USR 100: POKE 16090,31: R
EM * Set line to 32 Chrs
9850 RANDOMIZE USR 100: OPEN #2,"lp": RA
NDOMIZE USR 100: CAT "",: PRINT ": RANDO
MIZE USR 100: CLOSE #2
9860 PRINT AT 20,0:"Install new Diskette
, -OR- ""Q"" to Quit, Any Key for N
ext": PAUSE 0: CLS : LET z$=INKEY$: IF z
$="Q" OR z$="q" THEN GO TO 9864
9862 GO TO 9826
9864 STOP
9866
9868 REM ** Sending String Data to Disk
9869 REM #3. Create the U$ Data first
* Use Caps Sym Shf to QUIT * Use ENTER f
or starting new line
9870 LET u$="": CLS : PRINT AT 20,0:"Tur
n Disk ON, Enter a file nbr <1 to 5 onl
y)": INPUT zz: GO SUB 9970+zz: CLS : LET
u$="": LET a=0: LET b=0: PRINT AT a,b;"
";:
9872 POKE 23611,220
9874 IF PEEK 23611<221 THEN GO TO 9874
9876 LET c=PEEK 23560: POKE 23611,220
9878 IF c=12 THEN GO TO 9890
9880 IF c=13 THEN GO SUB 9940: GO SUB 9
902: GO TO 9872
9884 IF c=14 THEN GO SUB 9902: GO TO 99
06
9885 PRINT AT 21,29;LEN u$;AT a,b;CHR$ c
;: LET b=b+1: IF b=32 THEN GO SUB 9940
9886 PRINT AT a,b,"_";: LET u$=u$+CHR$ c
9888 GO TO 9872
9890 LET u$=u$( TO LEN u$-1): LET b=b-1:
IF b=-1 THEN LET b=31: LET a=a-1
9892 PRINT AT a,b," ";: GO TO 9872
9902 PRINT #5;u$: LET u$="": PRINT AT 21
,29," ": RETURN
9906 CLS : PRINT AT 20,0:"<1> Create Mor
e Text"" "<2> QUIT": INPUT xx: IF xx=1 TH
EN GO SUB 9940: GO TO 9872
```

A Fist Full of LKDOS TOOLS

As more TS-2068 users become LKDOS users the questions roll in. Some are working "Max DOS" (with Oilger Dos), but most are just getting their feet wetter with the rich LKDOS functions. Sometimes the process becomes a bit of a head scratching event. Update does not sell hair restoring lotion, but perhaps these utilities will have even better results.

The LLISTING for "LKDOS TOOLS" contains a group of utilities, each headed by REM statements. Each of these utilities may be plucked out to make individual sub-routines to MERGE to any BASIC program for their specific TOOL. There are no "ride-in Variables" and the utilities are independant. But there are a few vars that are re-used by the follow on tools.

The first few lines are simple Screen set ups. Then Line 104 sets three vars that need "Fast Finding" by the utilities. When initialized FIRST in the vars file, each subsequent use makes the programs run faster than if a large file must be searched for the vars values.

TOOL #1, The LKDOS INITIALIZATION, begins at line 9800. When the computer is turned ON, LKDOS is asleep. The "Wake-Up" slap is <RANDOMIZE USR 100:> followed by a LK System Command. The LKDOS manual tells us that we can use <RANDOMIZE USR 100:OPEN#4,"dd"> and from then on we can use <PRINT#4:> followed by a System Command. However, as we get into "MAX DOS", the PRINT#4 causes some conflicts. Actually, I find that it is better to always just use RANDOMIZE USR 100 before the System Commands, and never <OPEN#4,"dd">. Later, we will see that having channel 4 OPEN for the PRINT#4 commands can result in "too many channels open" and cause some confusion.

So, as this programmers personal preference, these utilities wont use PRINT#4 at all! If you thought that PRINT#4 was an essential ingredient for the soup then this should convince you, as there are many many LKDOS system commands and all are preceded by RANDOMIZE USR 100. AND, channel #4 remains closed throughout. You can corrolate the POKES in line 9804 through 9898 with the LKDOS "Version 3 EPROM" manual. <OPEN#3,lp> wakes up the PRINTER CODE in the LKDOS EPROM. At that time the key addresses in the LKDOS RAM chip are set to numbers that may or MAY NOT be suitable for your printer, or your desires. Beginning at line 9800 and ending at line 9814 is an Initialization routine that will fit into your needs for all Basic Program.

The next TOOL begins at line 9818 and is simply a "Proof Test" of the Initialization routine. But note the LPRINT at the end of line 9814. This is necessary after the Initialization tools to reset the printer's carriage for the new LEFT MARGIN. Most printers requires anOFF and back ON for any switch settings that changes print style or margins. Line 9818 simply prints groups of five characters for you to easily check to see that your desired TAB and LINE WIDTH are correct. You ALWAYS POKE 16090 with ONE Number LESS than your desired line width. Also, I might mention that all of these system POKES require either RANDOMIZE USR 100 or PRINT #4 AHEAD of the POKE.

In fact, all LKDOS commands require the RANDOMIZE or PRINT #4. This is because the LKDOS system is operating in an EXTRA MEMORY BANK, "the DOS BANK". So, RANDOMIZE USR: or PRINT#4: causes a BANK SWITCHING caper in order to carry out the following LKDOS command. The LKDOS system opens up another bank of memory and uses 16K of that new bank of memory addresses. Later we may see more "chunks" of the LKDOS Bank used, but for now thats all of the noodles in that bowl of soup (16K). Half is the EPROM and the other half is RAM.

cont.

```

9908>RANDOMIZE USR 100:CLOSE #5:STOP
9918 STOP : REM #4. LPrint or View disk
file
9920 CLS : PRINT AT 20,0;"<1> View File
<2> LPRINT File": INPUT xx: IF xx=1 THEN
GO TO 9928
9924 CLS : PRINT AT 20,0;"Set Printer ON
, Then ENTER": PAUSE 0: CLS : GO SUB 996
0: REM * Send data to printer instead of
screen
9928 CLS : LET u$="": GO SUB 9980+zz: RE
M * Open disk file for INPUT to screen o
r printer
9930 ON ERR GO TO 9946: INPUT #5;u$: RE
M * Bring in record from disk
9932 IF xx=1 THEN PRINT u$: REM * Pr
int this record and get next record from
disk.
9934 IF xx=2 THEN LPRINT u$: LET u$=" "
9936 GO TO 9930
9940 LET a=a+1: LET b=0: IF a>21 THEN P
OKE 23692,255: LET a=21
9942 PRINT : PRINT " ";: RETURN
9946 ON ERR RESET :RANDOMIZE USR 100:
CLOSE #5
9950 STOP
9952 GO TO 9920
9956 REM #5. Gsub Routine for quick for
mat changes
9960 CLS : INPUT "Input TAB;";Lm: INPUT
"Line Width";Lw: RANDOMIZE USR 100: POKE
16094,Lm: RANDOMIZE USR 100: POKE 16090
,Lw-1: LPRINT : RETURN
9970 REM * Tell me how to Concatenate a
number to OUT File?
9971 RANDOMIZE USR 100: OPEN #5,"Dta1.A$
OUT ": RETURN
9972 RANDOMIZE USR 100: OPEN #5,"Dta2.A$
OUT ": RETURN
9973 RANDOMIZE USR 100: OPEN #5,"Dta3.A$
OUT ": RETURN
9974 RANDOMIZE USR 100: OPEN #5,"Dta4.A$
OUT ": RETURN
9975 RANDOMIZE USR 100: OPEN #5,"Dta5.A$
OUT ": RETURN
9981 RANDOMIZE USR 100: OPEN #5,"Dta1.A$
IN ": RETURN
9982 RANDOMIZE USR 100: OPEN #5,"Dta2.A$
IN ": RETURN
9983 RANDOMIZE USR 100: OPEN #5,"Dta3.A$
IN ": RETURN
9984 RANDOMIZE USR 100: OPEN #5,"Dta4.A$
IN ": RETURN
9985 RANDOMIZE USR 100: OPEN #5,"Dta5.A$
IN ": RETURN

```

5 REM "CODE FINDER" ** Adaptable to all
TS-2068 systems ** Finds start and end a
ddress, and byte length of any CODE progr
am.

```

10 BORDER 0: PAPER 0: INK 7: CLS : INPU
T "Data Disk #?";dd: RANDOMIZE USR 100: G
O TO dd: RANDOMIZE USR 100: CAT " ",: INPU
T "File name";z$: RANDOMIZE USR 100: LOAD
z$CODE

```

```

20 CLS : PRINT "Searching for Starting
Code Adr""": FOR z=30000 TO 65535

```

```

30 IF PEEK z<>0 THEN PRINT "Code Start
is ";z: LET aa=z: GO TO 50

```

```

40 NEXT z

```

```

50 PRINT ""Searching for END Code Add
ress": FOR z=65367 TO z STEP -1

```

```

60 IF PEEK z<>0 THEN PRINT "End address
s=";z: GO TO 80

```

```

70 NEXT z

```

```

80 PRINT ""The byte length is ";z-aa:
STOP

```

The next LKDOS TOOL is at line 9822 and is a "CAT PRINTER". This lets you LPRINT a Disk Directory, another, and as many more as you would like to do, and from any drive in your system. You can get hard copies of all of your disks in the library with this CAT PRINT utility. Also, if you want to you can add <GOSUB 9960> just after the line number of line 9826. This would let you set up TAB for the Cat Print in order to print double column catalogs.

Tools #3 is a very versatile group of functions. First it uses some of the "TS-2068 DATA INPUT" program's techniques. That's the neat program that was presented in the April 89 issue. Here it is abbreviated to provide a fairly fast input typing to a "Sequential File SAVER". Actually this set of functions is a "mini data base manager and printer", jerked together to demonstrate a method of creating manageable SEQUENTIAL FILES in disk. The File Saver operates so slick that you don't even realize that the typed data is going into the disk file, and you won't believe that it is until proven by the next TOOL.

So, let's discuss what's happening between line 9869 and 9908. First, the DATA INPUT section operates to INPUT data from the keyboard. A "Character Counter" is printed at the bottom right corner of the screen. You can create "LINES" of data by watching the counter as you type. For example, if you plan to print lines of 64 characters, then as the counter reaches near 64, touch ENTER. The cursor springs down to the start of the next line--- BUT !! What you won't believe is that each time you touch ENTER, the previous line of data is socked away in a Sequential Disk File!! Actually ALL data ABOVE the current line being typed are "DEAD LINES". They have all been saved to disk, and are retained on screen just for your orientation.

Back at line 9890 a "File # is entered". Then a <GO SUB 9970 "OPENS a Sequential Disk File" with the name of "Dta" plus the file number. Example: file #1 will have the title of "Dta1.A\$ OUT ". Thereafter each time ENTER is touched, the data in U\$, the typing buffer, is saved to that file in disk. Lines 9872, 9874, and 9876 comprise the (now familiar) "Peek K State" INPUT typing function for reading the keyboard and assigning the key struck code to var c. If the key is DELETE, then line 9878 goes SUB to the DELETE Character routine. If the key struck is ENTER, then line 9880 goes SUB to the Sequential File SAVER routine. After the line of data is SAVED the typing buffer U\$ is zeroed and its back to the K-State for more characters.

Continuing- Then at line 9884, if CAPS-SYM SHIFT is the key, we have decided to QUIT, and the last line is SAVED at 9906. During input typing none of these escape routes are taken and line 9985 executes to do several things. First, the character counter prints the current length to keep you up-dated. Then the last character is printed to screen. Then the "print position variables a and b are up-dated. Then line 9986 prints the cursor and then Concatenates the last character to U\$. At line 9888 the GO TO 9872 gets Peek K again for the next character. Now let's look back at lines 9880 and 9885. Both use <GO SUB 9940> which is to check to see if the screen is full.

If so, then at line 9940 we POKE 23692,255. This kills SCROLL and any subsequent line ENTERED will push up from the screen bottom, a SCROLL UP if you will. Line 9906 is the END ESCAPE routine. But it also gives you a choice of continuing input to the disk file. If you decide to quit then line 9908 does a V E R Y IMPORTANT function. It CLOSES the FILE with <RANDOMIZE USR 100:CLOSE#5>. This is absolutely essential when you finish inputting data to an open disk file. ANY LKDOS FILE is OPENED with <RANDOMIZE USR 100> and must be CLOSED in the same manner. Remember that LKDOS is operating from the "DOS BANK of memory" and

cont.

2068

```

9990 REM *** LKDOS AUTO SAVE
      ROUTINE ***
9992 POKE 23730,255: POKE 23731,PEEK 2364
2+3
9994 RANDOMIZE USR 102: RUN

```

Two More TOOLS

The above two utilities are very useful to me. The LKDOS disk catalog does not contain the Start Address and END address or byte length of CODE FILES (Extension .C1). So, one is left fumbling in the dark when one needs that information to peek a code file. If you can peek a code file you can convert the peeks to Characters (if the code represents character data). Or, if it is a CODE Program, the code can be disassembled. "CODE FINDER" does that for you.

Next, we have a nice little utility to AUTO SAVE a program. "Hack, that's easy to do with just RANDOMIZE USR 102, eh"? It sure is, but the SAVE takes about 7 cylinders of disk space! This is because an Auto Save SAVES every bit of RAM from PROG clear up to Ram Top. So an auto save actually saves 38K of stuff to disk, when the actual program may be only 2K in length and will fit in just ONE CYLINDER of the disk.

The "LKDOS AUTO SAVE" utility given above reduces RAM TOP to about 750 bytes above the basic program and its variable file. Then it saves the program so that it auto runs by holding the ENTER key as the computer is turned on.

There is some rationale about Auto Running programs with LKDOS. We will attack it this way. We reduce RAM TOP to make the program shorter, thus taking up less disk space. BUT- when the program is reloaded, RAM TOP is STILL REDUCED. The result is that the program will Auto Boot O.K., but then may have a severe shortage FREE MEMORY for operation. If we reduce FREE (by setting Ram Top DOWN before the save) then it could be that the program won't run after auto loading. The way to fix this is to "prepare the program before saving it". We do that by installing a bit of programming in the FIRST convenient program line of the program to be saved.

There are two addresses, the contents of which controls Ram Top. They are addresses 23730 and 23731. Upon start up these two addresses have these values: 23730=87 and 23731=255. So, we should POKE these values to the two addresses EARLY in the auto run program. We do that by adding to the first operating line of the program:
 * <POKE 23731,255:POKE 23730,87>. Then when the program auto loads Ram Top will be set to 65367, its normal setting.

won't operate without RANDOMIZE USR 100.

TOOL #4, is the "View-PRINT" for Sequential Disk Files. These functions begin at line 9918 and end at line 9946. First, at line 9918, the STOP is to keep this package of functions from operating after the previous routine is finished. So, if you are operating both together, the STOP will be reported. Then to enter the routines type <CONT ENTER>. Line 9920 gives the choices of Viewing the Disk File or Printing it. We will skip the rationale about screen viewing and work with PRINTING. Line 9924 goes SUB to a neat little "PRINTER FORMAT" routine at line 9960, where you can format the TAB and Line width.

Then line 9928 does a little shenanagin of selectively OPENING a previously closed file in disk. Variable ZZ is carried over from the INPUT TOOL section (line 9870). Its value is between 1 and 5. So the proper line for the GO SUB is already fixed. This group of lines are the "OPEN#5 "IN" group". The objective is to OPEN the disk file and read the file back to the printer. A little parenthetical comment is appropriate. I couldn't find a key to Concatenate the titles with a number for either the OUT titles or the IN titles. This leaves the door open for someone to feed back the solution.

At line 9930 the <INPUT #5;u\$> "reads the disk file that has been opened. The disk file contains as many "RECORDS" as were previously separated by "ENTER". Saying that in a different way; Each time you SEND data to a OPEN FILE, it forms a new RECORD. Remember that earlier we sent the "lines of data" to disk each time we touched ENTER. So, each line became a separate RECORD, and now we are READING the records back from disk, one at a time. MOVING on to line 9932 and 9934, variable xx=2 for PRINTING, so line 9934 executes to LPRINT U\$, which has the current RECORD from disk.

Then Line 9936 goes back to line 9930 to get the next record. There is one problem to overcome. We are in a LOOP here, returning to line 9930 to keep on reading in records from the disk file. When there are no more records, the system will cause a report code "no more records" and stop. So, we place <ON ERR GO TO 9946> at the beginning of line 9930. Then when there is an attempt to read beyond the last record, the ON ERR will trip and line 9946 will execute.

When line 9946 executes, every record of the disk file has been printed. The ON ERR is RESET, and the disk file is CLOSED. And that is the end of this haranguing group of epistles (except for a few closing comments).

Sequential Files seemed to be a very complicated part of LKDOS to me. So, I pushed it aside until I could get rountult. Actually, once one gets used to "OPENING" and CLOSING FILES, which is the principal bug-aboo, sequential files offer great flexibility of managing data. The file handling is much faster than I thought possible. The principal advantage is the TINY amount of computer memory that is used. One can just keep on building a disk file to an extremely large size, much larger than can be handled within FREE RAM memory. These utilities points out just one method of data management with sequential files using short "line length" records. A record can be from a single character in length to several thousand characters.

Each individual save to disk is a separate RECORD. So, to build large records, one just keeps on typing and when finished, do the SAVE with <PRINT #5;"data" or U\$>. Don't forget to CLOSE the file when finished, either sending or retrieving data. Have fun! -BJ_

TRIP TO ENGLAND
by G. Chambers

As some of you are aware I recently took a 3-week trip to England. I shall take a moment to tell you what I noticed on the Sinclair scene. I need to caution you that I covered only places in Southern England. Places like Gatwick, Reading, Plymouth, and points in Cornwall such as Truro, Penzance, Helston.

Firstly, there is a dearth of good Spectrum computer magazines. I was able to pick up copies of SINCLAIR USER, YOUR SINCLAIR, and CRASH. The first two magazines are oriented toward 12 to 15 year-olds, I would judge. Very juvenile in language and tone. CRASH is not much better. An interesting thing though. Where I have copies of CRASH which contain upwards of 150 pages, the current issue contains only 34 pages. It is definitely in decline. The others have maybe 90-100 pages.

These three magazines all include a games tape stuck to the cover of the magazine. There's about two games per tape. I'm not sure what the game quality is; probably budget games.

There are other magazines which touch briefly on Sinclair computers. POPULAR COMPUTING Weekly (2 pages to Sinclair); COMPUTER SHOPPER (2 1/2 pages to Sinclair); COMPUTER EXPRESS, a weekly (half-page column);

There's also a magazine for the QL. Called QL World, it contains possibly 50 pages. An overseas subscription to QL WORLD (12 issues) would cost 30 pounds for surface mail. The cost for airmail is a bit confusing. It's either 10 pounds or 15 pounds extra.

Where once Boot's (chemists) and W. H. Smith (booksellers) used to be the popular Spectrum dealers, now Boots has nothing in computers, and Smith's confine themselves to selling games tapes and joysticks. Smith's don't even have many Sinclair computer books. I saw two books on the Z88, and one for the Spectrum. Several for the Amstrad, though. Where other electronic stores used to carry various computers, including Sinclair's, they seem to be out of the business. Hobby computers seem to definitely have lost their gloss.

I did come across one hobby computer store in Helston, Cornwall. It redeemed my faith a bit. Here they had Spectrum +2's and +3's; also Amstrads, C-64's, and Ataris. They had a couple of computers set up so that for 20 cents you could play any game for 5 minutes (and a further 5 minutes for another 20 cents, if there was no one waiting in line) Sort of a try-before-you-buy. They had 3-inch disks for sale, and all the goodies that go with hobby computers. Lots and lots of games tapes and disks.

You could also come across Spectrum games cassettes in odd places. Places that had music cassettes for sale might very well have a similar rack of games tapes. They would be budget tapes costing between \$4 & \$6.

Amstrad is selling dish antennas to pick up satellite TV. Where TV dishes in North America seem to be maybe 5-6 feet across, these are about 24 inches in diameter and cost about 200 pounds. Yet they pick up transmissions from a satellite that is 22,300 miles in the sky, about the same distance as ours. Howcum?

Not too much to report on new additions to the library. We are up to disk #9. There is a disk #10, which will contain PIXEL PRINT PLUS by Stan Lemke, but I am still working on it.

There is one development, shall we call it. I have been working on a program for saving disks to tape. By this I mean a program which will automatically save a disk track by track to an audio tape, then turn around and make a new disk copy from that tape. Or as many disk copies as you want.

Actually, I had the program written quite some time ago but I found that the drive was being exercised unduly. My coding had the drive head going back and forth to track 0 every time it read the next track. I have solved that shortcoming. And incidentally that exercise will provide the material for another newsletter article!

What all this is leading up to is this. I am going to try sending the Larken library disks out to members in the form of a tape that you can input into your system to create your own disk. You would then return the tape to me.

Why do this. Well, it is to simplify my life a bit. I think it will be easier on me to send out a tape cassette in the same manner as we are presently doing with the 2068 tape library. Sort of transferring some of the work-load onto you, the members! Well, we'll see how it works out.

In the meantime, if you have any ideas for Larken library disks, let me know.

NEWSLETTER COMMENTS
G. Chambers

Our newsletter has been receiving favourable comments lately from our membership. This is very satisfying to our Editor I am sure. However if the newsletter is good it has to reflect the effort that our members have made in providing the material to go in to it. To quote from our members:

"The May/June issue was great, should satisfy all since there was something for all computers, or so it seemed to me. Really lot's of information this time." D.L.

"By the way, keep up the excellent technical articles in the newsletter like the SNODGITS and TECHNICIAN TED pieces. I also love all the hints and tips you give on the Larken interface. PLEASE keep up the EXCELLENT work you do on the newsletter. It is appreciated." S.S.

"The latest copy of SINC-LINK is outstanding! Really well produced with meaty articles. Everyone involved deserves a big hand. I think it is possibly the best issue yet" M.R.

If these comments echo your sentiments then why not help us to keep it up. Write up a contribution to our newsletter. Don't put it off. We would like it now!

BATTERY BACKED CLOCK

Before I get into the battery backed clock for the QL, let me just mention something I found in an electronics magazine. BABANI is offering two books for the QL. Here are the extracts of the advertisements.

BP:150 AN INTRO TO PROGRAMMING THE SINCLAIR QL \$7.80
Helps the reader make the best use of the Sinclair QL's almost unlimited range of features. Compliments the manufacturer's handbook.

BP:156 AN INTRODUCTION TO QL MACHINE CODE \$10.00
The powerful Sinclair QL microcomputer has some outstanding capabilities in terms of its internal architecture, the QL has a large address range, advanced instructions which include multiplication and division. These features give the budding machine code programmer a good start at advanced programming methods. This book assumes no previous knowledge of either the 68008 or machine code programming.

These books may be ordered by telephone (416) 445-5600, or write to: MOOSEHEAD PUBLICATIONS, 1300 DON MILLS RD., NORTH YORK, ON., M3B 3M8. That's right in town! (and not too expensive either).

On to the clock. The following is NOT original. It is a compilation of two other articles. The battery-backed part was written by Dennis Briggs, and the clock write protect was written by Steve Papierowski. Each appeared in Quanta about a year ago.

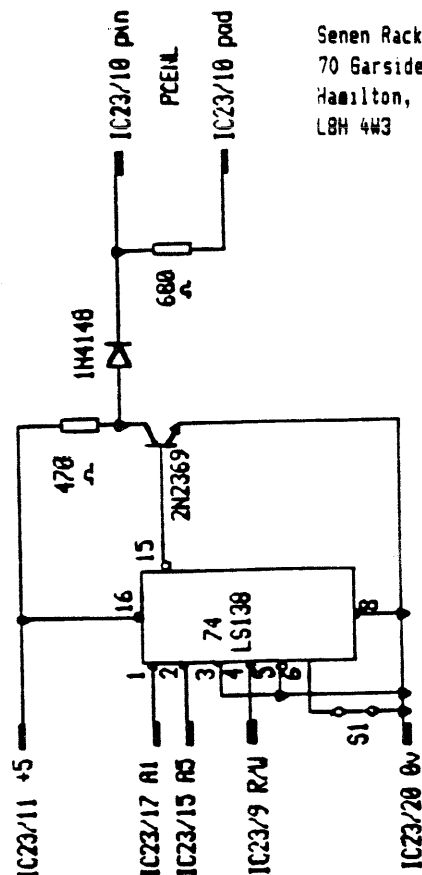
Most changes involved in running your clock continuously are minor. One of the many jobs of the ZX8302 is to provide the clock for the QL. If this chip can be powered by battery, then the clock would retain it's (not always proper - I'll get into that later) time. Fortunately, the 8302 has two 5V power pins, pin 40, and pin 11, and only one has to be kept active to work. For issue 5 boards, replace the link in position D22 with a small diode together with a further diode in D23. A small 3.8 volt battery connected to the points marked BATTERY (under the heat-sink) and fastened with sticky pads under microdrive 2 does the trick. For issue 6 boards, this simple approach is not possible due to tracing changes. A practical solution is to bend pin 40 out of it's pad. A 3.8 volt battery can be connected by soldering POS(+) to pin 40, and NEG(-) to regulator common. If possible check that the current is less than 2 microamp with the main power OFF. Keep in mind that the clock will be corrupted when a system crash occurs!

After you get the battery backed clock to work, you may find that it is sometimes reset on power-up. This is because the computer has wrote to the 8302's clock set line. You may have also noticed that when running some programs, the clock will lose alot of time. All these problems are because the write line to the clock in the 8302 has been triggered. The solution to these problems is found in the second article by Steve.

Steve has found that the only way to ensure TOTAL reliability of the real-time clock under ALL circumstances, is to provide a switchable write-protect function to the ZX8302. This will even eliminate the potential of corruption during system crashes!

What follows is essentially a re-write.

The real-time clock write registers are found at address \$18000/1 (\$10000-\$1FFFF) depending on board issue. Anyway, the important address lines are A1 & A5, which are both active low during write access to the clock registers. The 74LS138 chip uses A1, A5, and R/W to provide a disable signal (hold high) via a fast switching transistor to the 8302 enable line (PCENL) during all potential writes to the clock. Switch S1 allows the user to set the clock when necessary. All connections are made to the 8302 only, pin 10 of which must be lifted, and a 680 ohm resistor carefully added in series between pad and pin. For S1, a two pole DIP switch can be brought out to the rear panel. See circuit diagram for details. He states, "The above mod. works perfectly on my previously unreliable (40% failure rate) QL, which uses the battery backup detailed in QUANTA (Sep 86)." The editor added the last line of the article, "I tried this on one of my QL's - it certainly write protects the clock but it was not a 100% cure for the clock failing on power up. It looks better than just adding a battery and will probably help in many cases." I've got the battery working, and I am currently working on the write protect circuit. If anyone has any other articles of this sort, please get in touch with me.



Senen Racki
70 Garside Ave. N.
Hamilton, Ontario
L8H 4W3 (416) 549-6863

QL Clock Write Protect

THE QL and the RGB MONITOR

Translation of excerpts from an article
by Pierre GOUDET published by QL_DOC in
issue No 3, April-May 1988.

Translated by Louis Laferriere

The first time I tried to send the
output of my QL to a colour screen I
didn't realize the difficulties ahead.

My first purchase was a Radio Shack
monitor CM 4 . The DIN 8 pins of the
computer didn't match the monitor so I
replaced it with a DB-9, so as not to
void the warranty on the monitor.
After several attempts to obtain the
pinout information of the monitor I
returned it to RADIO SHACK.

Next I bought a TV/Monitor 14 " RGB. What
a surprise, no cord to interconnect with
the computer. #2 the plug on the monitor
is EIAJ 8 conductors, #3 once connected
to the computer the monitor stops
working. The display flashes across the
screen so fast it is impossible to read
it. Then everything disappears after
about 10 seconds. Finally a GENIUS stated
that the display stopped because the
signal from the QL was too strong. Just
before returning the monitor I called
Mr Real Gagnon the editor of QL_DOC who
recommended installing diodes in series
with the RGB leads. I installed one diode
1N1418 or 1N914 and miracle it worked
beautifully.

The monitor I used was Model #
C539-14544 and it is also a TV set . Text
is quite legible at 80 characters per
line with the QL, therefore it is also
perfect for word processing with my 2068.

Pierre Goudet

HACKER'S HAVEN PRESENTS:

PART 2 -- ZX-81 (AND 2068) MIDI INTERFACE PROJECT by LOU CHAMPAGNE

In the next few issues I will be describing the design and construction of a ZX-81 computer interface to connect to MIDI equipped musical equipment. Hopefully some members will become involved in developing software and applications for this project.

In the last article I described the development of a MIDI interface using the Motorola 6850 ACIA chip. Since then, after some experimentation with the circuit concepts, and discussion with other Timex users some necessary changes become apparent.

Do not use the TIL-111 for the opto-isolator as the max frequency (and therefore the data throughput rate) varies from component to component too much and in some units was found to be too slow resulting in a poor waveform at the 6850 RXD input. I have switched to a 6N136 and now have excellent results.

The clock frequency of 2 Mhz is too much for the average 6850. I was using a 6850 "B" model which functioned properly at first but degraded somewhat after long periods of use. I believe this was thermal but never-the-less I figured wrongly that I could "push" the quoted spec. of 1.5 Mhz. So now the circuit contains a "divide by 4" on the oscillator output made out of a 74LS74 chip. It is also necessary to send a 55hex to the ACIA as the second setup byte in order to use the onboard "divide by 4" in the ACIA. The resultant clock frequency in the ACIA of 125Khz provides the necessary clock for 31.25 Kbaud which is MIDI.

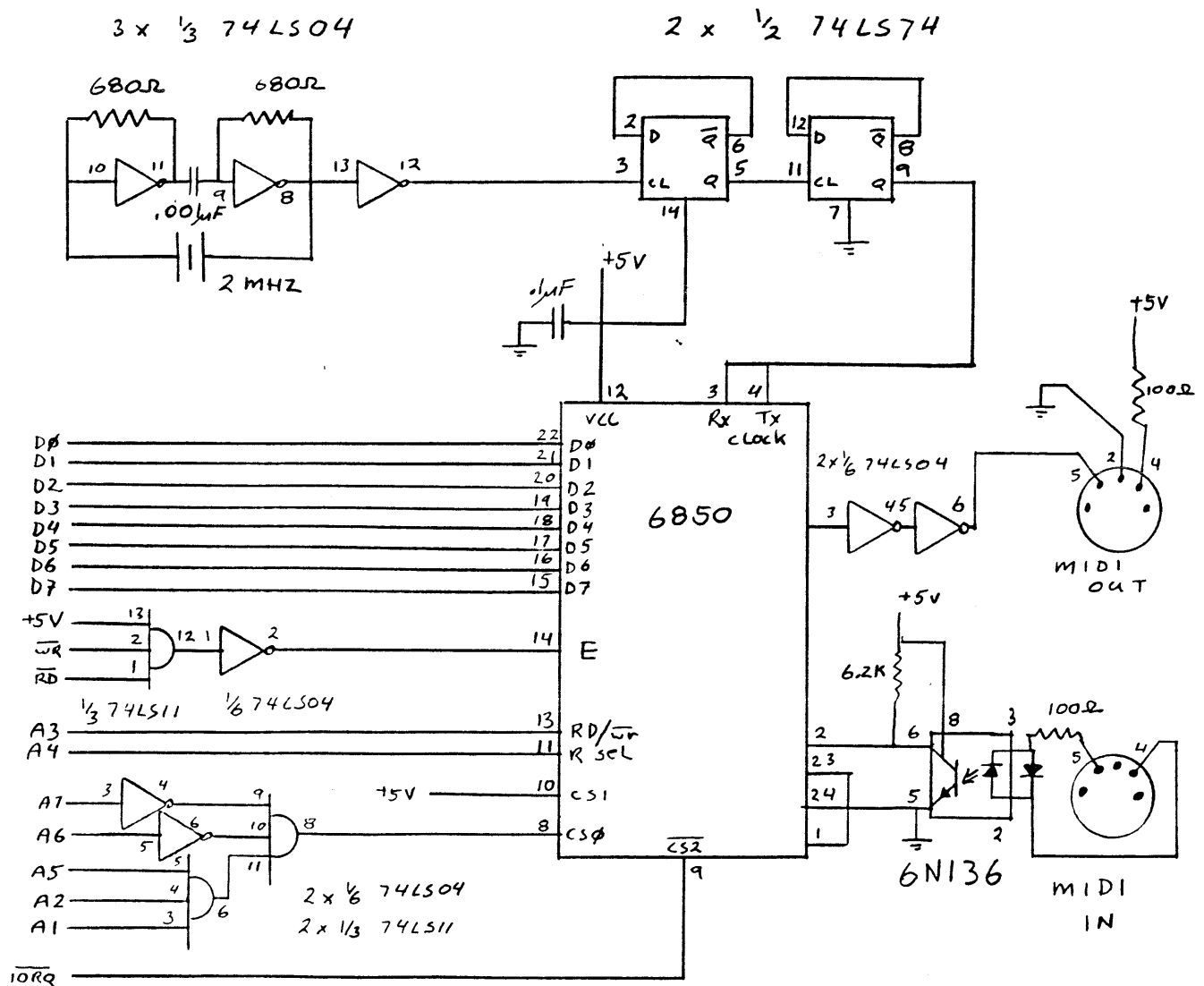
The addressing must be changed for proper operation as well. To work with the Larken disk drive interface as well as to solve the 2068 interfacing problem. In addition early circuit tests revealed an interaction between the keyboard and the interface which has been fixed. To anyone who has experimented in building the interface I suggest the following become standard.

A0 is NOT to be used, this means the board can be accessed in the FAST and SLOW modes (ZX-81 users will have to subtract 1 from the address for SLOW mode M/L). I consulted Larry Kenny on the disk drive address interaction problem and have since decided to use these addresses...

BINARY	6850	DECIMAL	HEX
0010 0111	CONT	39	27
0010 1111	STAT	47	2F
0011 0111	TXD	55	37
0011 1111	RXD	57	3F

ZX81

Since A3 now controls RD/W and A4 operates the register select lines on the 6850 the schematic that has now evolved is this....



cont.

Incidentally, Larry Kenny's LDOS eeprom has a routine on it to output data to any odd # port. From Basic POKE 12300, port # in decimal; POKE 12301, data byte value ; and then RAND USR 16374 to output the data.

The following are excerpts from the MIDI spec. 1.0

MIDI... serial interface at 31.25 Kbaud

8 bit data 1 start bit 1 stop bit no parity bit

multi byte format - most messages use at least 3 bytes

These messages are structured like this...

example of NOTE ON message

byte 1	byte 2	byte 3
1001 xxxx	Okkk kkkk	0vvv vvvv
xxxx= MIDI CHANNEL 0-15 (represents 1-16)	kkk kkkk=KEY NUMBER (0-127)	vvv vvvv=VELOCITY (0-127)

example of NOTE OFF message

byte 1	byte 2	byte 3
1000 xxxx	Okkk kkkk	0vvv vvvv

VELOCITY usually set to
0000 0000 except when
"RELEASE VELOCITY" is used

example of PITCH BENDER message

byte 1	byte 2	byte 3
1110 xxxx	Oppp pppp	Oppp pppp
	ppp pppp= PITCH BEND (0 - 127)	this second optional byte provides higher resolution on some synth modules but is ignored if not used
	where 64 is centre(normal)	

example of PROGRAM CHANGE message

byte 1	byte 2	
1100 xxxx	Onnn nnnn	no 3rd byte
	nnn nnnn=PRESET NUMBER for tone changing (0-127)	

also there are some important messages that are only one byte long

cont.

1111 1000 MIDI CLOCK - received or transmitted to synch up to
 ----- another devices time base - one of these
 is sent every 24th of a beat of music
 so in 4/4 time 96 are sent every bar
 so at 120 beats per minute 48 of these
 must be sent or received and processed
 every second.

1111 1010 START - used to tell other sequencers to start
 ----- sent before MIDI CLOCK starts

1111 1100 STOP - used to stop all MIDI CLOCK controlled
 ----- devices

There are plenty more messages in the MIDI spec. 1.0 the curious should consider contacting THE INTERNATIONAL MIDI ASSOCIATION , 5316 WEST 57th Street, Los Angeles, California USA 90056.










HEX	DEC	ADDRESS	ASSEMBLER	COMMENT

3E	62	16514	LD A,	SETUP BYTE
03	03	5	03	-RESET
D3	211	6	OUT-A,	OUTPUT IT
27	39	7	27	PORT 27H(CONT)
3E	62	8	LD A,	#2 SETUP BYTE
55	85	9	55	#BITS CLOCKDIV PARITY
D3	211	20	OUT-A,	OUTPUT IT
27	39	1	27	PORT 27H(CONT)
DB	219	2	IN A,	INPUT FROM
2F	47	3	2F	2FH (STAT)
CB	203	4	BIT 0,	BIT CHECK ON
47	71	5	A	STATUS TO FLAG
28	40	6	JRZ,	LOOP IF
FA	250	7	FA	NO BYTE IN RXD
DB	219	8	IN A,	INPUT FROM
3F	57	9	3F	3FH (RXD)
D3	211	30	OUT-A,	OUTPUT TO
37	55	1	37	37H (TXD)
C3	195	2	JMP	JUMP TO
8A	138	3	8A-	408AH (16522)
40	64	4	40	TEST RXD

This is some machine code that tests out the interface on a ZX-81. It is a illustration of how the ports are used as well as the decoding of the status byte. At any rate it tests a completed interface by passing MIDI IN data out through the MIDI OUT jack. Connect the MIDI OUT of the source device (keyboard, sequencer, or drum pad etc.) to MIDI IN of the interface. The MIDI OUT of the interface should go to MIDI IN of a controlled device such as a synth voice module.

TILL NEXT TIME HAPPY MIDI HACKING -- HACKERS HAVEN LOGGING OFF BYE

FASTEXT 80 CRIB SHEET

CONTROL CODES					
CAPS	27;87;1 49	15	27;45;1	27;77	12
NONE	27;87;0 48	18	27;45;0	27;80	
Symb.	1	2	3	7	8
CAPS	 ON	 132 ON	 ON	 96 Elit.	
NONE	 OFF	 OFF	 OFF	 80 Pica	
Mode	Elong.	Cond.	Under line		Form Feed

This chart is for those who have a Smith-Corona Fastext 80 printer. I made the chart using my "Sketcher" program which I have recently adapted to use the 40 character-per-line proportionally-spaced code which I call p40. Then I printed it in pseudo-NLQ mode using my graphics print program which I call "grafLQ.B4". I have supplied copies of these programs to the club library but if anyone wants a copy from me, I'll provide them on a disk for Larken; please send \$10.00.

Bob Mitchell 20 Wild Briarway Willowdale Ont M2J 2L2

```

1 REM Data Creator from 2/86 ZX Computing modified for T/S 2068 by Steven V.
Gunhouse
2 CLEAR 64999: LET x=65000
3 CLS : PRINT AT 10,5:"Loading M/L ..."
4 LET a=10: LET b=11: LET c=12: LET d=13: LET e=14: LET f=15
5 READ a$: IF a$="END" THEN GO TO 10
6 FOR n=1 TO 15 STEP 2: LET w=16*VAL a$(n)+VAL a$(n+1): POKE x,w: LET x=x+1:
NEXT n
7 GO TO 5
10 CLEAR : REM Your program should start after this line
20 DELETE 9000,9998: REM The program will load itself, create the proper data
, then delete everything not needed in the new program.
25 REM Including lines 20-100
30 INPUT "Start address? ";s
35 PRINT "Start Address: ";s
40 POKE 65000,s-256*INT (s/256): POKE 65001,INT (s/256)
50 INPUT "End address? ";e
55 PRINT "End Address: ";e
60 POKE 65002,e-256*INT (e/256): POKE 65003,INT (e/256)
70 RANDOMIZE USR 65004
80 PRINT "Remember: ""You must add a line: "" 2 CLEAR ";s-1;": LET X="";s
100 DELETE ,2: DELETE 20,100
9972 DATA "0000000021270F22"
9973 DATA "ATFE454CCDD81622"
9974 DATA "9FFE2AEAFDE5BE8"
9975 DATA "FDB7ED527DE60728"
9976 DATA "0E2AEAFD7DE6F86F"
9977 DATA "1108001922EAFD2A"
9978 DATA "EAFD2B22EAFDE5B"
9979 DATA "E9FDB7ED522A1FE"
9980 DATA "21BCFE22A5FE3E08"
9981 DATA "32A3FE2AEAFD2B22"
9982 DATA "EAFD237E218EFE77"
9983 DATA "AFED6F47ED8F4FED"
9984 DATA "6F79FE0A3803C607"
9985 DATA "4F78FE0A3803C607"
9986 DATA "472AA5FE79C63077"
9987 DATA "2B78C630772B22A5"
9988 DATA "FE21A3FE3520C42A"
9989 DATA "9FFE011800CDBB12"
9990 DATA "21A7FE56235E1B73"
9991 DATA "2B72ED5B9FFE0118"
9992 DATA "00EDB02AA1FE1108"
9993 DATA "00B7ED52D225FEC9"
9994 DATA "0000000000000000"
9995 DATA "0000000000000000"
9996 DATA "001400E422000000"
9997 DATA "0000000000000000"
9998 DATA "0000000000220D00"
9999 DATA "END"

```


IN

May/June 1989

May 22, 1989

Dear Out-of-town members

We delayed the publishing of this newsletter until after the Washington Computer Fest, so that we could bring a first-hand report while it was still current news. With a bi-monthly newsletter things can become quite stale by the next issue.

The report is in the newsletter so I won't comment much about our trip. Except to mention that the drive back was as bad as the drive down. We ran into rain and a heavy snowstorm which made driving very treacherous. I have to admit that I was fortunate in doing my share driving during the fine weather part of the trip. Well, I must confess to maybe not doing my fair share of the driving, but I'm grateful that my companions did not complain about it!!

Bob Warner, one of our members who is also a CATS group (ComputerFest sponsors) has just written to tell me that I won a package of Dyson disks at the Fest. The draw took place after we had departed. Hey, that's great. I told you the Fest was a success!!

In the newsletter there is an advert for some computer stuff by Orin Zelenak. I'm sort of interested in one of the computers, and I wondered if there was enough interest among the members to warrant buying the whole lot and selling at cost to members who might be interested only in parts of the offering. If you are interested drop me a line.

The ZXAPPEAL newsletter has a couple of items in it which I shall try to get into the next issue of the newsletter. One is a passive device to aid in making ZX81 tape copies. I shall have to construct one myself to help in making club copies of ZX81 tapes. I have always made original copies, and it is a tedious process.

The other item is an improved software routine for making TS2068 tape copies. This one is similar to other copy programs which use the TS2068 as an intermediary. The program is written by Eric Boisvert of BYTEPOWER magazine, and appears to be superior to any others. He makes the claim that it will make a better copy than a poor original. Well, we'll see. Seems almost too good to be true! I'm looking around for an unloadable tape to try it out on!

I have a number of interesting items that some of you will be interested in. One is a set of schematics for the QL. There are two sheets, about 17 by 22 inches. Any QL owners want a copy? Cost is nominal, i.e. photocopying cost, whatever that turns out to be (and postage, of course). Also, the same member, Senen Racki, a QL enthusiast, has sent me a copy of a QL Technical Guide (170 pages), and a QL Service manual (62 pages). Same thing applies. I may have mentioned this in the last newsletter.

From another newsletter I have a copy of the schematics for the Larken ZX81 interface. If any Larken ZX81 owner is interested I can send a copy out. It is just three sheets. There is no documentation for it.

But talking about documentation, Bill Harmer has been expanding his ZX18 Larken manual considerably. It is a draft, and

rather wordy in my estimation. But for any Larken ZX81 owner I'm sure that it will be seen as a godsend. If you want a full copy (about 50 pages), or if you have partial copy and just want the revised/supplementary sheets (25 sheets) the same as above applies. Just ask for them.

What else have we got. Well, Stan Lemke has released most, if not all, his software into the public domain, and he has sent me a copy of PIXELPRINT PLUS. This is a Desk Top Publisher type of program. Stan sent it to me on tape, with the suggestion that I make it available to the club membership. I shall put it in our tape library, and make it tape 66. Better ask for it by name until I get it into the catalogue. I have also converted it to the Larken Disk format, and shall be putting it into the Larken Disk Library. Tentatively as TTSUC Libr. Disk #10. It will be in the form of two DS 40 TPS disks, or one DS 80 TPS disk. The Larken library is growing, though several of the disks are not really ready to my satisfaction yet. I shall send out a catalogue to all known (to me) 2068 Larken owners, probably with the next newsletter. As a reminder club library disks are \$2 for members, \$5 for non-members.

I think that I'm just about up-to-date with all club correspondence. If you are waiting for something do drop me a line. I shall be going to England for a short while, and I shall look around for things Spectrum and QL-wise. I don't expect to see much in the way of ZX81. Probably saw more at the ComputerFest than I shall see overseas. I hope to be able to pick up some magazines, in any event.

Bob Mitchell has been working on a program which will copy a screen to the printer. He has made the first use of the Larken user-NMI feature. That puts some user code in the Larken LKDOS RAM, where it can be called by the NMI button and pressing the "F" key. The routine then copies the screen to a dot-matrix printer. One of our club disks will contain the necessary programs. It is not ready just yet. Needs some polishing up, with a menu and instructions.

Hugh Howie has taken on the role of section leader for the QL users in our club and doing what can only be called a bang-up job. He is very enthusiastic. He is building up a club QL library on micro-cassettes, and I believe also on 5 1/4 inch disks. If you have any questions, etc., I suggest that you drop him a line. Ask what he has in the QL library. Hugh's address is as follows:

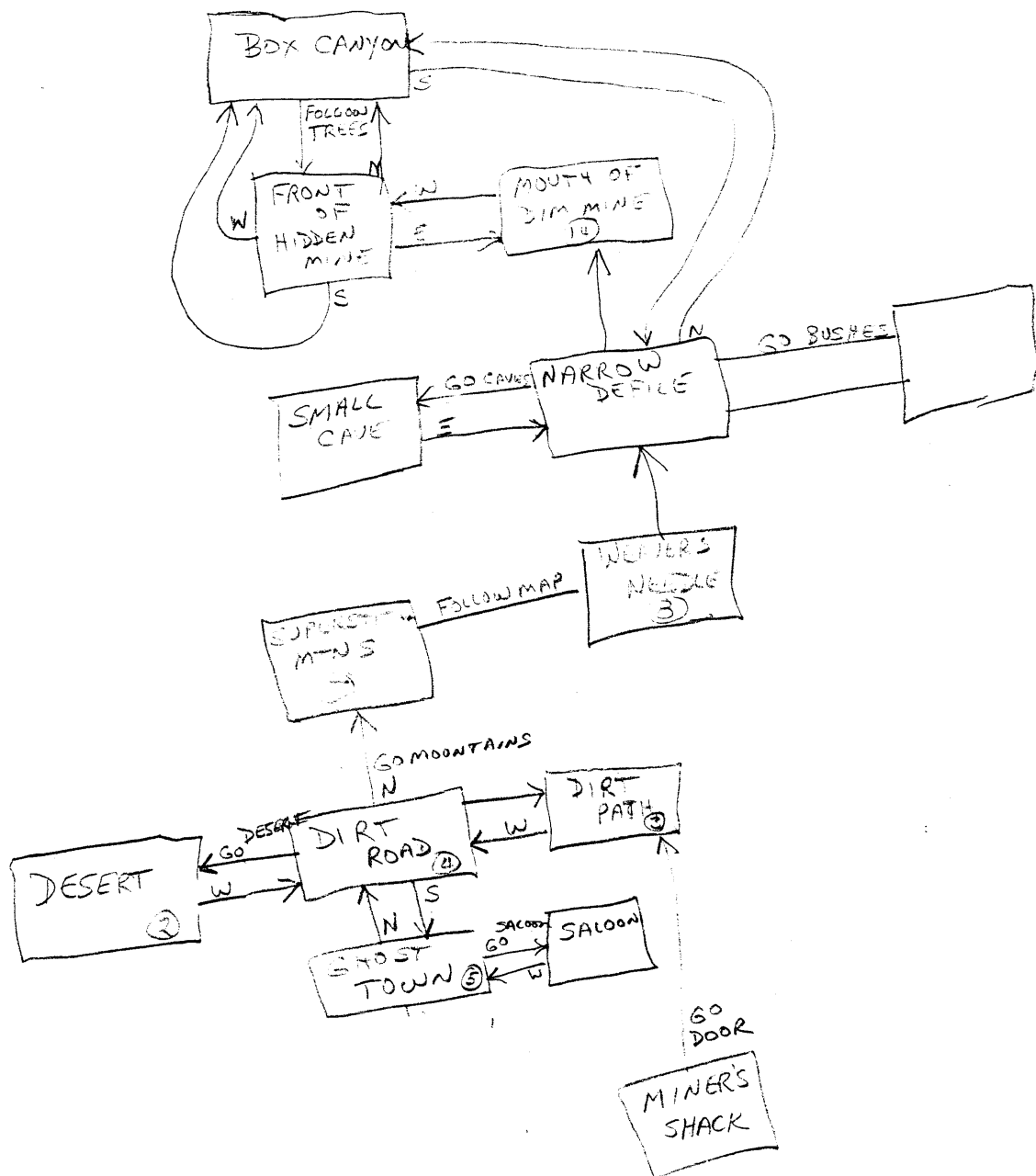
Hugh Howie
586 Oneida Drive,
Burlington, Ont. L7T 3V3

tel (416) 634 4929

Of course I'll still be here to look after all Out-of-town members needs, so you can drop me a line also. It's just that if you contact Hugh initially, you should get faster service.

I shall have to close now. Seem to have run out of topics!!

Sincerely,
George Chambers



DUTCHMAN'S MINE.

Somewhat complete.

TORONTO TIMEX-SINCLAIR USERS CLUB
May 22, 1989

JUN 1 '89

14 Richome Court
Scarborough, Ont.
M1K 2Y1

Les Cottrell
108 River Heights Drive
Cocoa, FL 32922

Dear Les,

Thank you for the revised Club disk #6. I'm sure there were a number of conflicts between the menu and the programs on the disk. Re the Dutchman's Mine, I converted it from another computer program which had the industry standard way of handling strings. I found it quite a task to modify it for the TS2068, so there may very well be some bugs in it. I enclose a map which I have made of the program. I never seem to have time to work my way through the game properly. However, as I recall the mule can be caught if you use a carrot that is to be found in the cabin. Also the mule can only carry so much, and one or so articles are too big or too long or something. If you let the mule go you cannot catch him again. You can find things by digging in the sand at some places. Things that you need on the hunt. I forget what else there is about the program. As I recall I did work my way through the whole thing once; seems to me that when I came out of the last tunnel/cave(?) that I was in the miner's cabin! What do the changes that you made, do?

Re the program "index.B1" I know why your printer locks up. It is because Bob Mitchell, when he wrote the program arranged for it to look for a printer ready signal from the printer. Trouble is, everyones printer seems to send a different "printer ready" signal back to the computer. The problem is that the check is made in the TIMACHINE-compiled code "indexO.C1". I have suggested that he put the Basic version, ready to be compiled, on the disk, so that other members can make the change and recompile it themselves. For example Bob's Fastext 80 printer sends back the code "237" as a ready signal. My Fastext 80 (the same printer) returns a "108". So I modified the line to suit myself and recompiled it. Do you have TIMACHINE?

To find what your printer "ready" signal is, enter the following lines:

```
10 LET n = IN 127: PRINT AT 10,10;" "; AT 10,10; n
20 PAUSE 50: GOTO 10
```

You will get different numbers depending on the printer state. To try out the TS2040 printer use the value "IN 251"

I'll search the code and see if I can locate the value 237; maybe it can be fixed by POKEing in your printer value. It would be worth a try.

I have a copy of BETA BASIC, but I simply have never used it. Too many things on my plate, I think!

I'll enclose the Games Tape "A" for you as you requested. And I'll close this letter off while on this sheet! Sincerely,
George Chambers