

SINC - LINK

Vol. 6 No.5

Sept - Oct '88

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 A FREE COPY AS PART OF THE \$20.00 ANNUAL MEMBERSHIP FEE.

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

ALL MATERIAL IS PUBLIC DOMAIN AND CAN BE REPRINTED. PLEASE CREDIT THIS PUBLICATION AND THE AUTHOR IF YOU COPY MATERIAL.

SEND CORRESPONDANCE TO:

Attention: SINC-LINK EDITORS,
TORONTO TIMEX-SINCLAIR USERS
CLUB, P.O. BOX 7274 STN. A,
TORONTO, ONTARIO, CANADA, M5W 1X9

TS2068 ISSUE

Page 2	Editorial
Page 3	Bob's Notebook
Page 4	Larken EPROM
Page 5,6,7	Big Copy
Page 8	Simple Sound Port
Page 9	Label Maker
Page 10	LKDOS 3 Review
Page 11	Pascal on Larken
Page 12	Larken Game Save
Page 13	File Recovery
Page 14	The Last Page

EXECUTIVE OFFICERS:

PRESIDENT:	ED MAYBEE (743-8141)
VICE PRESIDENT:	JEFF TAYLOR (244-8583)
TREASURER:	BILL LAWSON (444-8772)
SECRETARY:	GEORGE CHAMBERS (751-7559)
ACTIVITIES DIRECTOR:	RENE BRUNEAU (531-9749)
TAPE LIBRARIAN ZX81:	
TAPE LIBRARIAN 2068:	RENATO ZANNESE (635-6536)
TAPE LIBRARIAN QL :	REG COTTLE (785-3987)
PAPER LIBRARIAN :	TOM AROELLA (653-6978)
NEWSLETTER :	R. BRUNEAU, J. TAYLOR, R. ZANNESE
LIAISON OFFICER (Out-of-town members):	GEORGE CHAMBERS, 14 RICHOME COURT, SCARBOROUGH, ONTARIO, M1K 2Y1 (416-751-7559)

TORONTO TIMEX - SINCLAIR USERS CLUB

P. O. Box 7274 Stn. A Toronto, Ont., M5W 1X9
Canada

EDITORIAL

It is with swelled heart (and slightly swelled head) that I am proud to announce that Sinc-Link placed third out of twenty-two entries in Time Designs' "Top Ten Newsletters" survey!

This survey, published in the May/June '88 issue of TD, judged entries from all over North America on the basis of content, originality, layout and appearance. They also published my name as editor so I'm particularly happy!

Of course, the editor is only responsible for the layout and presentation. It is the contributors who must supply the all-important originality and content. Without them, I wouldn't have anything to present! So I think that a hearty round of congratulations are due to the writers of the May/June '88 (Vol 6-3) issue. Heartfelt thanks to George Chambers for submitting that issue to TD and for taking the time to fill out their questionnaire in winning style.

On to business. For those of you who aspire to greatness or at least political office of a kind, nominations for the new club Executive will be taken at the September meeting. Can you think of someone you'd like as president, v-p, treasurer etc.? Submit his/her name, submit your name! Voting will take place at the October meeting. Plan to be there and cast your vote.

In case you're wondering, the Sinc-Link blurb on the front-left cover is changing print style every month as I experiment with a program called "1st Class Fonts" from Byte Power. I'll review it in the next newsletter.

Keep those cards, comments and articles coming and maybe we can grab first place in TD's next newsletter survey!

J.T.

Toronto Computes! Presents



COMPUTER FEST '88

Fall Edition

October 14, 15 & 16

Friday Noon-9pm Sat. & Sun. 10am-6pm

Exhibition Place

Arts Crafts & Hobbies Building

Keep the Toronto Computer Fest '88 in mind (See above). Because of staffing problems and the cost of a booth our club will not be present at the show. However our club has been asked to provide support at the experts' table.

The way it works is like this: Computer Fest folks publish a schedule of when the expert for each system will be present. This way, the public can know when to come to the show to ask questions etc., about their particular system. At the moment we have been assigned periods as follows. Fri Oct 14th, 4 to 8pm, Sat 10 to 2pm, Sunday unassigned.

Rene Bruneau and myself have volunteered to sit in as Timex/Sinclair experts. Probably we will have a bit of Timex stuff there as a talking point.

If you are interested in a role please contact Rene or myself. Look at the cover of this newsletter for phone numbers.

George Chambers

OPEN #, CLOSE #, CHANNELS & STREAMS - A TUTORIAL

Let's delve into the mysterious realm of channels and streams and those two odd tokens OPEN # and CLOSE #. These two commands are extremely powerful and have many uses and yet there is nothing at all about them in the TS2068 User Manual except to say that "these and other commands like FORMAT, MOVE, CAT etc will be used with future peripherals". Well, by now we know that many of these are indeed used with the disk drive interfaces. But, long before DOS's came along, it was apparent that OPEN # and CLOSE # did have a role to play with the basic machine. Much of this has to do with the opening and closing of channels and streams with respect to the TS2040 printer and the TS2068 keyboard and screen.

The TS2068 communicates with peripheral devices with the aid of the channel information area which lies between the system variables and the BASIC program. This area starts at address 26688 and ends at 26708, just short of the program area which starts, normally, at 26710. A short program given in the listing below will print out all the information on channels and streams. Part of its output is shown below as Table 1 (with some explanatory notes); the code has been divided into 5-byte blocks, each block representing one channel.

TABLE 1 - CHANNEL INFORMATION AREA

Address	Chan	Value	Notes
26688	K	0	1st 2 bytes = ROM add.1280d
26689		5	(5*256+0); ROM PRINT routine.
26690		14	2nd 2 bytes = ROM add.3086d
26691		12	(12*256+14); get input key.
26692		75	Code for letter "K".
26693	S	0	1st 2 bytes same as for "K".
26694		5	
26695		191	2nd 2 bytes = add. 4543;
26696		17	input routine called.
26697		83	Code for letter "S".
26698	R	231	This block is reserved for
26699		10	use by the computer and
26700		191	is not available to the
26701		17	user.
26702		82	Code for letter "R".
26703	P	0	1st 2 bytes also point to
26704		5	1280d; ROM Print routine.
26705		191	Input routine called;
26706		17	(see explanation below).
26707		80	Code for letter "P".
26708		128	Stop byte.

When the TS2068 is switched on, 21 bytes are copied from address 4522d in the ROM to the channel info area where they may be altered by the user.

In the fourth block (Channel "P"), addresses 26703 and 26704 point to the ROM PRINT routine at 1280d (0500h). These are often changed to point elsewhere (eg. in the Aerco prcode, they are changed to [5] and [251] respectively which equates to 64261 (the start of the prcode routine). Once these two bytes have been changed, all printing will go via Channel "P" to the <prcode> routine. To get back to the TS2040, these two bytes must be restored to [0] and [5] respectively.

In Spectrum these 21 start-up bytes are in ROM 5551d and are copied to the Spectrum channel info area which resides at 23734 to 23754. Addresses 23749 and 23750 equate to 26703 and 26704 in the TS2068.

Summing up, each block represents a channel and contains five bytes; the first two represent the address of the called output routine (normally in the ROM); bytes 3 and 4 represent the address of the called input routine; finally, byte 5 is the code of the character representing the channel.

Channel "S" stands for "screen" and handles the output to the monitor or TV display. Channel "P" stands for the "printer" and handles output to this peripheral. Channel "K" stands for the keyboard but handles both output and input; it inputs from the keyboard but also outputs to the lower part of the screen display.

When the PRINT command is used, the output goes through channel "S" to the main (upper) part of the screen. By careful use of the OPEN # command, the output can be made to go through any other channel.

Streams are associated with a specified channel and there are 16 streams available (0 to 15). Using the PRINT command normally sends the output through stream "2" to channel "S". To make the PRINT command go to channel "P", it is only necessary to open stream 2 to channel "P". This is done by the command [OPEN #2,"P"]. Note that this instruction has two arguments, the former first tells the computer which stream is being altered and the latter identifies the single character of the channel to which the stream is to be directed.

If you type [OPEN #2,"P"] then [PRINT "test";PAUSE 0], the PAUSE 0 command prevents the input routine being called which would print an error report (bytes at addresses 26705 and 26706 call the ROM routine at address 4543). Without the PAUSE 0, the printing would be immediately erased.

At switch on, only four of the 16 streams are opened.

- Stream 0 points to channel "K"
- Stream 1 points to channel "K"
- Stream 2 points to channel "S"
- Stream 3 points to channel "P"

When a stream has been opened to a different channel, it will continue to do what it has been directed to do until told otherwise. A channel may be closed by the command [CLOSE #n] where n is any number from 0 to 15. Also, a channel may be closed by pointing the stream to its start up channel. For example, if stream 3 has been opened to channel "S", it may be closed by the command [CLOSE #3,"P"]. This example would send all LPRINT and LLIST commands to the screen until it is told to revert to normal. This is particularly useful to the programmer during the debugging process to look at the results of the LPRINT commands on the screen rather than waste printer paper.

When a stream has been opened to the printer, the edit line (lower screen area) is unaffected; in fact, any PRINT #0 or PRINT #1 commands will continue to display on the lower screen. Use of the hash symbol [#] followed by a [0] or [1] changes the print stream to the lower screen. PRINT, LIST, INPUT and INKEY\$ may all have their output affected this way; their default value is stream 2 (the upper screen); that is to say, PRINT #2 is the same as PRINT and PRINT #3 is the same as LPRINT, etc.

Type in the following listing and save it; then use it anytime you want to check the current status of channels and streams.

```

3 REM This program works in
  either TS2068 or Spectrum
5 REM =====
6 REM Use GO TO 10 for info
  on channels & streams.
7 REM Use GO TO 200 to PEEK
  stream info in system
  variables. Use GO TO 220 to
  PEEK channel info area up to
  start of program area.
9 REM =====
10 LET a=PEEK 23631+256*PEEK 2 :start of CHANS
3632: LET b=a :from sys variables
20 IF PEEK a=128 THEN GO TO 70 :last byte in table
30 PRINT "CHANNEL ";CHR$(PEEK :channel letter
(a+4))
40 PRINT "Output Address=";PEE :output address
K a+256*PEEK (a+1)
50 PRINT "Input Address=";PEEK :input address.
(a+2)+256*PEEK (a+3)
60 PRINT : LET a=a+5; GO TO 20
70 FOR a=23574 TO 23605 STEP 2
80 PRINT "Stream ";(a-23574)/2 :stream number.
;TAB 10;
90 IF PEEK a+256*PEEK (a+1)=0 :closed channel
THEN PRINT "CLOSED"; NEXT a; STO
P
100 LET c=(PEEK a+256*PEEK (a+1 :channel letter
))+b+3
110 PRINT "points to Channel ";
CHR$(PEEK c); NEXT a
120 STOP
130 RANDOMIZE USR 100; SAVE "ch :Larken SAVE routine
ans.B1" LINE 10
140 STOP
200 FOR i=23568 TO 23605; PRINT :addresses of
i,PEEK i; NEXT i :channels attached
210 STOP :to streams.
220 LET prog=PEEK 23635+256*PEE :start of prog area
K 23636

```

Bob's Notebook con't

eprom3.ct

```

230 FOR i=b TO prog=1: PRINT i, :examines chan
PEEK i;TAB 20;CHR$ PEEK i AND PE info area up to
EK i->32 AND PEEK i<=90: NEXT i start of prog area

```

LARKEN EPROM DOS Version 3

by G. Chambers

The FOR...NEXT loop in line 200 will display a copy of 38 bytes in the system variables area containing the address of channels attached to streams. Each stream holds two bytes which accounts for 32 bytes and the extra 6 are used by the internal editor and automatic listing routines. Each two bytes contains a relative displacement, from the byte before the channel info area, to the channel that is pointed to. A zero means that the stream is closed.

For Larken users, note the extensive use of OPEN # in the extended BASIC commands available and particularly note that when any or all windows or large printer channels are opened, 38 bytes (total) of memory are added to the channel information area. This changes the location of the start of the program area and can cause a problem if there is machine code in a REM statement in the first program line. (This problem can be avoided by PEEKING the start of the prog area and then using this value plus 5 as the call address for the machine code routine in the REM statement; eg, [LET x=PEEK 23635+256*PEEK 23636: LET x=x+5: RANDOMIZE USR x]). In these 38 bytes which you can examine by using line 220 of the listing, you will find that new channels have been set up at the beginning of what was the start of the program area.

An interesting and illuminating exercise is suggested for Larken users:

```

.[LOAD "chans.B1"], ie, the listing above.
.Open some of the Larken extended BASIC channels.
.Try [RANDOMIZE USR 100: OPEN #4,"dd"].
.Then [PRINT #4: OPEN #5,"w0"]. (These will
.Then [PRINT #4: OPEN #6,"w1"]. (open three
.Then [PRINT #4: OPEN #7,"w2"]. (windows.
.Now use the listing above to look at what has
happened. Try doing the same thing with the Spectrum
emulator in place, if you have one.

```

I'll leave the heavy thinking to you at this point. What I have written will pose some other questions and I look forward to someone else expanding on this theme in some future issue.

by Bob Mitchell

n30n

TIMEX 2068 COMPONENTS FOR SALE

1. Complete System
 - TS2068 w/Romswitch
 - TS2050 modem+ Specterm Software
 - Ollger DD Interface
 - (Includes Motherboard w/RGB 1'face)
 - Ollger Printer Interface
 - Smith Corona Fastext 80 Printer w/cable
 - Disk Drive with Power Supply (all cased)
 - Lots of software on disks (free)

Total package (excluding shipping costs) \$600 Can.

2. - TS2068 without Romswitch \$65 Can.
3. - TS1000 (new in carton) \$30
4. - TS1000 RAMPack \$10
5. - ByteBack modem w/software \$30
 - (2068 or 1000)
6. - TS2040 Printer \$25

If any one is interested in this equipment drop John Matheson a line. Address is 1852 Appleford St Gloucester, Ont. K1J 6T4 CANADA. His phone number is (613) 746-7869

Larry kenny has come out with a new EPROM DOS for his Larken system. We have only had tthis version for a few days, but Bob Mitchell has done a short review of it else where in this newsletter. Larry is offering this latest version of the EPROM for \$5 plus S&H, plus your old EPROM. I happened to have five spare EPROMS around so I sent them to Larry along with \$26 and asked him to send a set so that we cold upgrade some of our members systems. This has been done.

An interesting thing however. Two of the EPROMS that I sent to Larry were made by TI and were the type that have a "penthouse" (as Bill Jones calls it) on top of them. Sort of a raised area where the window is. Anyway, turns out that they would not work. At least they would not work until I connected a wire between pins 14 and 20, and bent pin 20 so that it did not enter the socket. Something like we experienced with the very first Larken cartridges. I see, also, reference to this sort of a problem in the Current issue of TS UPDATE where there is a note to the effect that "...apparently the EPROM with the "penthouse" on top cannot be used with the Larken cartridge."

It's not really a problem, just an interest' quirk. I'm sure that we will hear the whys wherefores shortly.

Bob Mitchell mentions a small 'bug' in the PRINT routine. I have drawn Larry's attention to it. It is not a real bug, but here's how it comes about. In the Aercio printer drive (and maybe others) you have to poke in a number that is one less than the number of characters you want to be printed out on a line. That is to say, if you want a line with 64 characters in it, TASWORD style, you must POKE in a 63. It would seem that the new EPROM defaults to 64, instead of to the needed 63. It is still operational; you simply have to POKE a 63 into the DOS. When I spoke to Larry, he was unaware that the wrong number was in the EPROM. I'm sure it will be corrected promptly.

As I mentioned earlier, we have upgraded 5 systems in the club. By sending the five EPROMs out we were able to do this without anyone having to shut down their system, or to buy an EPROM they did not really have a use for. I mention this because if any other members are interested in upgrading by this process they should contact me forthwith and I will return my EPROMs for another go around. It will cost you \$5.40. Plus your old EPROM, of course.

Big Copy

Here is a program that can be used to make a large shaded copy of a screen using the 2040 printer in the spectrum mode.

It prints out in four sections and these sections have to be taped together.

To use the program you would first set your computer in the spectrum mode. Enter CLEAR 49999:LOAD"B_C"CODE . Once this loaded you can load a screen by entering LOAD"SCREEN"SCREENS : RANDOMIZE USR 50000.

This program is presented by Jean-Pierre Overbeek of Holland. Entered by Renato Zannese 8/30/88.

```

1 REM BIG COPY
2 REM saved as "B_C" CODE
10 CLEAR 49999: LET adr=50000
20 LET a=10: LET b=11: LET c=1
30 LET d=13: LET e=14: LET f=1
40 FOR q=100 TO 200 STEP 10: R
LOAD a$,tot
40 LET w=16*VAL a$(1)+VAL a$(2
50 POKE adr,w: LET adr=adr+1:
LET tot=tot-1
60 LET a$=a$(3 TO ): IF a$<>""
THEN GO TO 40
70 IF tot<>0 THEN PRINT "Erro
r in line ":q: STOP
80 NEXT q
85 STOP
90 PRINT "All is OK!!! Press a
key to SAVE": PAUSE 0
95 CLS : SAVE "B_C"CODE 50000,
520: BEEP .1,.1: PRINT "Prepare
to VERIFY": VERIFY "B_C"CODE :
STOP
100 DATA "3e003259c53a57c53258c53
a58c54f3a59c547c5f5cd89c4f1c13e08
325dc5e57e325cc52f773a59c5473a58c
54fc5f5cd4b",6155
110 DATA "c4f1c17ee607325ac57ee63
8cb3fcb3fcb3f325bc5c53a5cc54f0608
cb40cce7c3cb79cd0ac4cb2110f2c179c
3258c5e1",6721
120 DATA "233a5dc53d325dc5c26fc33
a59c53c3259c5fec0c255c3060af5c5cd
cd0ec1f110f73a57c5c6403257c5fe00c
250c3cdc8c4",6785
    
```

```

130 DATA "c9f5c53a5ec53cfe21cafdc
3fealca02c4325ec5c1flc93e81c3f7c3
cdcd0e3e01c3f7c33a5bc5ca13c43a5ac
5c5e5dd2169",7625
140 DATA "c447fe00ca26c4110400dd1
910fc21ff5a3a5ec506004f0906047ecb
27cb27cb27cb27dd5600b277dd2311200
01910eaelc1",5019
150 DATA "c9cb39cb39cb3978cb3fcb3
fcb3f21005811200047fe00ca67c41910
fd09c90f0f0f0f0a050a050c010403010
40208080208",3542
160 DATA "00080002000004000000000
000210040dd21b8c4cb39cb39cb391600
5919780608cb7fc2a9c4dd23dd23c3b4c
4dd5600dd23",4330
170 DATA "dd5e0019dd231710e6c9100
008000080004000200400020001001608
010018210040cb4lcadac4cb3ec3dcc4c
b26230b78b1",3817
180 DATA "c2d0c415c2cac4060021ff5
778cb3fcb3f772310f721005936472310
fb21005a78cb3fcb3f77230478fe00c20
0c521ff5836",5338
190 DATA "00dd2137c52100590e20dd5
600dd230608cd7aca2ac5367823cb2210
f40d79fe00c21cc5c9774400005444000
05654000054",4324
200 DATA "2a252e572a252400022aa40
00228a40003a8a4",1076
    
```

Addr	Hex	OP	Operand/Notes
C350	3E00	LD	A,00
C352	3259C5	LD	(C559),A
C355	3A57C5	LD	A,(C557)
C358	3258C5	LD	(C558),A
C35B	3A58C5	LD	A,(C558)
C35E	4F	LD	C,A
C35F	3A59C5	LD	A,(C559)
C362	47	LD	B,A
C363	C5	PUSH	BC
C364	F5	PUSH	AF
C365	CD89C4	CALL	C489
C368	F1	POP	AF
C369	C1	POP	BC
C36A	3E08	LD	A,08
C36C	325DC5	LD	(C55D),A
C36F	E5	PUSH	HL
C370	7E	LD	A,(HL)
C371	325CC5	LD	(C55C),A
C374	2F	CPL	
C375	77	LD	(HL),A
C376	3A59C5	LD	A,(C559)
C379	47	LD	B,A
C37A	3A58C5	LD	A,(C558)
C37D	4F	LD	C,A
C37E	C5	PUSH	BC
C37F	F5	PUSH	AF
C380	CD4BC4	CALL	C44B

C383	F1	POP	AF	C3FD	3E81	LD	A,81
C384	C1	POP	BC	C3FF	C3F7C3	JP	C3F7
C385	7E	LD	A, (HL)	C402	CDCD0E	CALL	0ECD
C386	E607	AND	07	C405	3E01	LD	A,01
C388	325AC5	LD	(C55A), A	C407	C3F7C3	JP	C3F7
C38B	7E	LD	A, (HL)	C40A	3A5BC5	LD	A, (C55B)
C38C	E638	AND	38	C40D	CA13C4	JP	Z,C413
C38E	CB3F	SRL	A	C410	3A5AC5	LD	A, (C55A)
C390	CB3F	SRL	A	C413	C5	PUSH	BC
C392	CB3F	SRL	A	C414	E5	PUSH	HL
C394	325EC5	LD	(C55B), A	C415	DD2169C4	LD	IX,C469
C397	C5	PUSH	BC	C419	47	LD	B,A
C398	3A5CC5	LD	A, (C55C)	C41A	FE00	CP	00
C39B	4F	LD	C,A	C41C	CA26C4	JP	Z,C426
C39C	0608	LD	B,08	C41F	110400	LD	DE,0004
C39E	CB40	BIT	0,B	C422	DD19	ADD	IX,DE
C3A0	CCE7C3	CALL	Z,C3E7	C424	10FC	DJNZ	-4 >C422
C3A3	CB79	BIT	7,C	C426	21FF5A	LD	HL,5AFF
C3A5	CD0AC4	CALL	C40A	C429	3A5EC5	LD	A, (C55E)
C3A8	CB21	SLA	C	C42C	0600	LD	B,00
C3AA	10F2	DJNZ	-14 >C39E	C42E	4F	LD	C,A
C3AC	11	POP	BC	C42F	09	ADD	HL,BC
C3AD	79	LD	A,C	C430	0604	LD	B,04
C3AE	C608	ADD	A,08	C432	7E	LD	A, (HL)
C3B0	3258C5	LD	(C558), A	C433	CB27	SLA	A
C3B3	E1	POP	HL	C435	CB27	SLA	A
C3B4	23	INC	HL	C437	CB27	SLA	A
C3B5	3A5DC5	LD	A, (C55D)	C439	CB27	SLA	A
C3B8	3D	DEC	A	C43B	DD5600	LD	D, (IX+0)
C3B9	325DC5	LD	(C55D), A	C43E	B2	OR	D
C3BC	C26FC3	JP	NZ,C36F	C43F	77	LD	(HL), A
C3BF	3A59C5	LD	A, (C559)	C440	DD23	INC	IX
C3C2	3C	INC	A	C442	112000	LD	DE,0020
C3C3	3259C5	LD	(C559), A	C445	19	ADD	HL,DE
C3C6	FEC0	CP	C0	C446	10EA	DJNZ	-22 >C432
C3C8	C255C3	JP	NZ,C355	C448	E1	POP	HL
C3CB	060A	LD	B,0A	C449	C1	POP	BC
C3CD	F5	PUSH	AF	C44A	C9	RET	
C3CE	C5	PUSH	BC	C44B	CB39	SRL	C
C3CF	CDCD0E	CALL	0ECD	C44D	CB39	SRL	C
C3D2	C1	POP	BC	C44F	CB39	SRL	C
C3D3	F1	POP	AF	C451	78	LD	A,B
C3D4	10F7	DJNZ	-9 >C3CD	C452	CB3F	SRL	A
C3D6	3A57C5	LD	A, (C557)	C454	CB3F	SRL	A
C3D9	C640	ADD	A,40	C456	CB3F	SRL	A
C3DB	3257C5	LD	(C557), A	C458	210058	LD	HL,5800
C3DE	FE00	CP	00	C45B	112000	LD	DE,0020
C3E0	C250C3	JP	NZ,C350	C45E	47	LD	B,A
C3E3	CDC8C4	CALL	C4C8	C45F	FE00	CP	00
C3E6	C9	RET		C461	CA67C4	JP	Z,C467
C3E7	F5	PUSH	AF	C464	19	ADD	HL,DE
C3E8	C5	PUSH	BC	C465	10FD	DJNZ	-3 >C464
C3E9	3A5EC5	LD	A, (C55E)	C467	09	ADD	HL,BC
C3EC	3C	INC	A	C468	C9	RET	
C3ED	FE21	CP	21	C469	0F	RRCA	
C3EF	CAFDC3	JP	Z,C3FD	C46A	0F	RRCA	
C3F2	FEA1	CP	A1	C46B	0F	RRCA	
C3F4	CA02C4	JP	Z,C402	C46C	0F	RRCA	
C3F7	325EC5	LD	(C55E), A	C46D	0A	LD	A, (BC)
C3FA	C1	POP	BC	C46E	05	DEC	B
C3FB	F1	POP	AF	C46F	0A	LD	A, (BC)
C3FC	C9	RET		C470	05	DEC	B

C471 0C	INC	C			
C472 010403	LD	BC,0304		C4C5 00	NOP
C475 010402	LD	BC,0204		C4C6 010016	LD BC,1600
C478 08	EX	AF,AF		C4C9 08	EX AF,AF
C479 08	EX	AF,AF		C4CA 010018	LD BC,1800
C47A 02	LD	(BC),A		C4CD 210040	LD HL,4000
C47B 08	EX	AF,AF		C4D0 CB41	BIT 0,C
C47C 00	NOP			C4D2 CADAC4	JP Z,C4DA
C47D 08	EX	AF,AF		C4D5 CB3E	SRL (HL)
C47E 00	NOP			C4D7 C3DCC4	JP C4DC
C47F 02	LD	(BC),A		C4DA CB26	SLA (HL)
C480 00	NOP			C4DC 23	INC HL
C481 00	NOP			C4DD 0B	DEC BC
C482 04	INC	B		C4DE 78	LD A,B
C483 00	NOP			C4DF B1	OR C
C484 00	NOP			C4E0 C2D0C4	JP NZ,C4D0
C485 00	NOP			C4E3 15	DEC D
C486 00	NOP			C4E4 C2CAC4	JP NZ,C4CA
C487 00	NOP			C4E7 0600	LD B,00
C488 00	NOP			C4E9 21FF57	LD HL,57FF
C489 210040	LD	HL,4000		C4EC 78	LD A,B
C48C DD21B8C4	LD	IX,C4B8		C4ED CB3F	SRL A
C490 CB39	SRL	C		C4EF CB3F	SRL A
C492 CB39	SRL	C		C4F1 77	LD (HL),A
C494 CB39	SRL	C		C4F2 23	INC HL
C496 1600	LD	D,00		C4F3 10F7	DJNZ -9 >C4EC
C498 59	LD	E,C		C4F5 210059	LD HL,5900
C499 19	ADD	HL,DE		C4F8 3647	LD (HL),47
C49A 78	LD	A,B		C4FA 23	INC HL
C49B 0608	LD	B,08		C4FB 10FB	DJNZ -5 >C4F8
C49D CB7F	BIT	7,A		C4FD 21005A	LD HL,5A00
C49F C2A9C4	JP	NZ,C4A9		C500 78	LD A,B
C4A2 DD23	INC	IX		C501 CB3F	SRL A
C4A4 DD23	INC	IX		C503 CB3F	SRL A
C4A6 C3B4C4	JP	C4B4		C505 77	LD (HL),A
C4A9 DD5600	LD	D,(IX+0)		C506 23	INC HL
C4AC DD23	INC	IX		C507 04	INC B
C4AE DD5E00	LD	E,(IX+0)		C508 78	LD A,B
C4B1 19	ADD	HL,DE		C509 FE00	CP 00
C4B2 DD23	INC	IX		C50B C200C5	JP NZ,C500
C4B4 17	RLA			C50E 21FF58	LD HL,58FF
C4B5 10E6	DJNZ	-26 >C49D		C511 3600	LD (HL),00
C4B7 C9	RET			C513 DD2137C5	LD IX,C537
C4B8 1000	DJNZ	+0 >C4BA		C517 210059	LD HL,5900
C4BA 08	EX	AF,AF		C51A 0E20	LD C,20
C4BB 00	NOP			C51C DD5600	LD D,(IX+0)
C4BC 00	NOP			C51F DD23	INC IX
C4BD 80	ADD	A,B		C521 0608	LD B,08
C4BE 00	NOP			C523 CB7A	BIT 7,D
C4BF 40	LD	B,B		C525 CA2AC5	JP Z,C52A
C4C0 00	NOP			C528 3678	LD (HL),78
C4C1 2004	JR	NZ,+4 >C4C7		C52A 23	INC HL
C4C3 00	NOP			C52B CB22	SLA D
C4C4 02	LD	(BC),A			

```

C52D 10F4    DJNZ  -12 >C523
C52F 0D      DEC   C
C530 79      LD    A,C
C531 FE00    CP    00
C533 C21CC5  JP    NZ,C51C
C536 C9      RET
C537 77      LD    (HL),A
C538 44      LD    B,H
C539 00      NOP
C53A 00      NOP
C53B 54      LD    D,H
C53C 44      LD    B,H
C53D 00      NOP
C53E 00      NOP
C53F 56      LD    D,(HL)
C540 54      LD    D,H
C541 00      NOP
C542 00      NOP
C543 54      LD    D,H
C544 2A253E  LD    HL,(2E25)
C547 57      LD    D,A
C548 2A2524  LD    HL,(2425)
C54B 00      NOP
C54C 02      LD    (BC),A
C54D 2AA400  LD    HL,(00A4)
C550 02      LD    (BC),A
C551 28A4    JR    Z,-92 >C4F7
C553 00      NOP
C554 03      INC   BC
C555 A8      XOR   B
C556 A4      AND   H
C557 00      NOP
C558 00      NOP
C559 00      NOP
C55A 00      NOP
C55B 00      NOP

```

First thing you have to do is carefully solder the 22k resistor to a thin piece of wire. Solder the end of the resistor to the pin called sound. Cover the resistor and wire with thin heat shrink tubing. Solder the other end of the wire to the center of the RCA jack and solder the outside of the RCA jack to the big ground trace on the rear of the Larken interface.

The second thing you have to do is solder one end of the 100k resistor to the pin called beep. Solder the other end of the resistor to the end the wire from the beginning.

The last part you have to do is solder the 10k resistor between the ground and the wire that goes to the center of the RCA jack. Refer to diagram if you are having any trouble.

Make sure all visible solder joints/wire are covered up by the heat shrink tubing. The long wire can glued to the rear board on the Larken interface.

If you don't have a Larken interface, this modification can be done inside the computer. Be careful when working inside the computer.

If you don't have a monitor with an audio input you can use the Mini Audio Amplifier made by Radio Shack # 277-1008

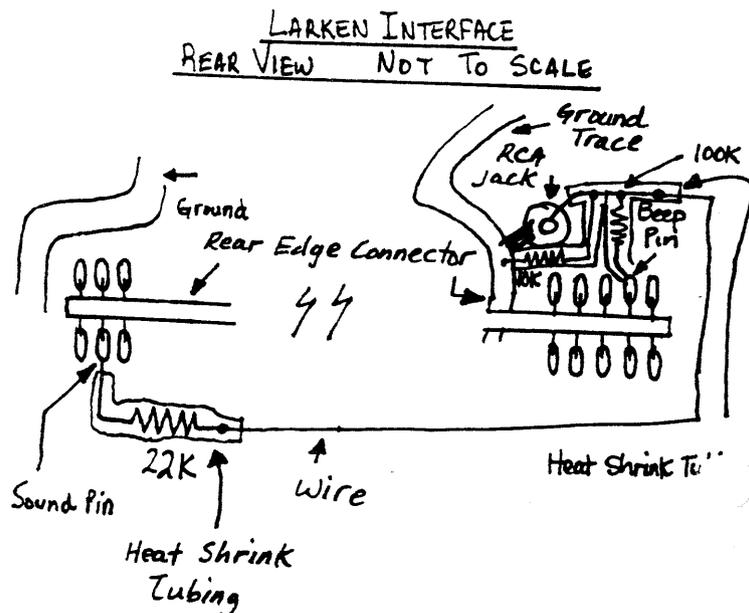
Simple Sound Port
By Renato Zannese 5/31/88

A simple sound port can be added to the rear of the Larken Disk Interface so your monitor / amplifier can control the volume coming from the speaker on the TS 2068.

You will need to get the following items to make a external sound port:

- 22k resistor
- 10k resistor
- 100k resistor
- an RCA jack

You have to open you computer so you can cut one of wires that goes to the speaker inside the computer.



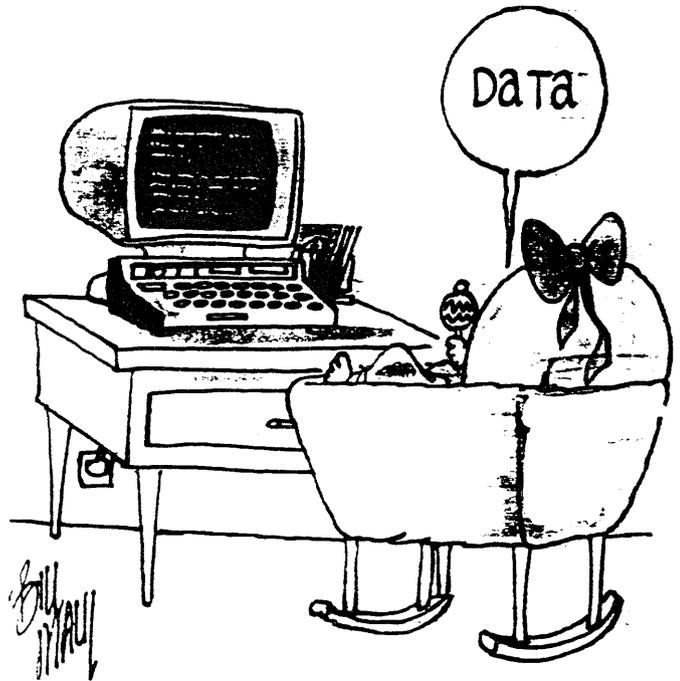
AN IMPROVED LABEL MAKER FOR LARKEN SYSTEM
by G. Chambers

Writing programs to create labels is a time honoured pastime among the computer crowd. But I think this program offers just a bit more.

It is designed to produce labels of the contents of your Larken system disks. It offers the option of printing to the 2040 printer or a large printer. It provides the option of sorting the files alphabetically. It also provides the option of printing the program names in one, two or three columns.

Although the program makes use of the Hacksel printer software "lprint.C1", another driver software could be used by modifying lines 160 and 350.

The program operates on the basis of reading the directory (Track 0), placing the names in a (d\$) array, then manipulating them as required.



program names are held in d\$(n)

```

100 REM File Label Maker v1.1
    for the LARKEN system
110 REM By G. Chambers
    14 Richome Court
    Scarborough, Ont.
    CANADA M1K 2Y1
120 REM Use to make a printout
    of disk files
130 REM Data statements contain
    Drive Control Code.
140 REM For use with the DSK400
    system of 5090 bytes/track
    using a DD drive i.e.80 tracks
150 RANDOMIZE USR 100; OPEN #4,
"dd"
160 PRINT #4: LOAD "lprint.C1" C
ODE
170 LET c$=""
180 RESTORE VAL "210"
190 FOR n=VAL "63000" TO VAL "6
3047"
200 READ a: POKE n,a
210 NEXT n
220 DATA 195,43,246,0,0,0,0,0,0
,243,205,98,0,201,58
230 DATA 100,0,251,201,205,33,2
46,58,176,92,50,29,32,205,126
240 DATA 0,205,123,0,33,112,32,
17,156,224,1,0,20,237,176
250 DATA 195,38,246
260 DIM d$(90,11)
270 LET trac=VAL "23728": LET 1
oadbuf=VAL "63000"
280 LET name=VAL "57688"
290 BORDER PI/PI: PAPER PI/PI:
CLS

```

```

300 PRINT AT VAL "2",VAL "6": I
NK VAL "2": PAPER VAL "6": LARK
EN DISK UTILITY ": PAPER PI/PI:
,TAB VAL "10": PAPER VAL "6": L
abel-Maker ": PAPER VAL "1",TAB
VAL "6": PAPER VAL "6": By Ge
orge Chambers "
310 INK VAL "3": PLOT NOT PI,VA
L "108": DRAW NOT PI,VAL "62": D
RAW VAL "255",NOT PI: DRAW NOT P
I,VAL "-62": DRAW VAL "-255",NOT
PI
320 INK VAL "6": PLOT VAL "8",V
AL "114": DRAW NOT PI,VAL "50":
DRAW VAL "238",NOT PI: DRAW NOT
PI,VAL "-50": DRAW VAL "-238",NO
T PI: INK VAL "7"
330 PRINT ".....Install disk ne
eding a label".... then Pres
s a key": PAUSE NOT PI
340 POKE 23658,0: PRINT AT 12,0
: OVER 0;c$:c$:AT 14,0: INK 7:"L
arge or small printer? (L/S) ":
PAUSE 0
350 LET n$=INKEY$: POKE 26703,0
: POKE 26704,5: IF n$="1" THEN
POKE 26704,254
360 PRINT AT 14,0: OVER 0;c$: P
AUSE 50: PRINT AT 16,0:"Print Nu
mber of columns (1-3) ": PAUSE
0: LET p=CODE INKEY$:48
370 PRINT AT 16,0: OVER 0;c$: P
AUSE 50
380 INPUT "Sort (Y/N)? ":s$
390 INPUT "Enter Disk Title":n$
400 LPRINT
410 REM Picking pgm names
    from buffer(Track 0)
420 POKE trac,NOT PI: RANDOMIZE
USR loadbuf
430 LPRINT "Disk Name: ":n$: LP
RINT
440 FOR n= PI/PI TO VAL "90"

```

```

450 IF PEEK (name+3)=NOT PI THE
N LET e=n-1: LET n=VAL "90": GO
TO VAL "520"
460 IF PEEK (name+PI/PI)=VAL "2
54" THEN LET name=name+VAL "34"
: GO TO VAL "490"
470 FOR m=PI/PI TO VAL "9"
480 LET d$(n,m)=CHR$ PEEK (name
+m)
490 NEXT m
500 LET name=name+VAL "34"
510 PRINT AT 11,11:d$(n); "

```

```

520 NEXT N
530 IF s$>"y" THEN GO TO 590
540 PRINT AT 11,11: FLASH 1:" S
orting ": FLASH 0
550 FOR i=1 TO (e-1): FOR j=i T
O e
560 IF d$(i)>d$(j) THEN GO SUB
580
570 NEXT j: NEXT i: GO TO 590
580 LET q$=d$(j): LET d$(j)=d$(
i): LET d$(i)=q$: RETURN
590 PRINT AT 11,11:"

```

```

600 FOR n=1 TO e
610 IF INT (n/p)=(n/p) THEN LP
RINT d$(n, TO 10): GO TO 630
620 LPRINT d$(n):
630 NEXT n
640 LPRINT : LPRINT : LPRINT
650 DIM f$(VAL "9")
660 PRINT AT VAL "20",VAL "6":
Label finished""Press a key for
another start"
670 PAUSE NOT PI: GO TO VAL "28
0"

```

```

680 STOP
8000 INK 0: PAPER 7: BORDER 7: C
LS : LIST : STOP
9900 REM ** Save to Disk **
9910 CLEAR : PRINT USR 100: SAVE
"label2.B1" LINE 100
9930 STOP

```

LKDOS VERSION 3 EPROM
HIGHLIGHTS & FIRST IMPRESSIONS by Bob Mitchell

Version 3 came hot on the heels of version 2; now the LKDOS cartridge contains some major additions and improvements. This article is a summary of the highlights I have used in the few days I have had version 3, some applications I have already identified and used and some of my overall impressions. The new version 3 eprom comes with new instructions and you MUST REPEAT MUST read these very closely to glean all the information LARKEN has included.

NOTE: [PRINT #4] below may be replaced by [RANDOMIZE USR 100] OR [PRINT USR 100] if stream #4 has not been OPENED.

HIGHLIGHTS

1. PRINT #4; LPRINT "file" differs from the PRINT command by sending the file to the printer. Like PRINT this does a good job on text but line numbers will be garbled in program listings.

2. PRINT #4; NEW (reloads an AUTOSTART program in the current drive.) This can be very useful in two ways:

(1) Used from the keyboard, it does a "warm start" which lets you avoid switching the computer off and on for a fresh start. A "warm" start is one that does not change any of the settings in the cartridge or any selected banks.

(2) Used in a program line, it reloads the AUTOSTART program in the current drive. (If there isn't one in the current drive, it does a cold start.) This avoids having a duplicate of the AUTOSTART program such as "menu.B1" to return to, as was the case in the first version.

3. PRINT #4; MOVE is a [rename file] command. It does not copy programs as can be done by the LARKEN "move" program supplied by LARKEN on disk. To rename a file follow this command with "oldname", "newname". If you try to use a newname that is already on the disk, you will get an error report [Z - Name Exists]. This error trap applies only to this command; you can still SAVE to an existing file name (eg, when you are saving a file with new listings or data.) MOVE is particularly useful for renaming an NMI save.

4. NMI SAVES. LARKEN has given us a choice of saving the SCREEN along with the entire program. Pressing the push button on the disk interface and then any key from 1 to 5 will do the NMI save with the SCREEN too. Holding the CAPS SHIFT key down while pressing any key 1 to 5 will skip the SCREEN save. Pressing the [a] key will usually stop the program. This can often get you out of an apparent "crash" when dabbling with machine code. The [s] key can still be used to do a separate SCREEN save. After pressing the NMI button, you can always press [ENTER] if you change your mind and then you return to your program. The [f] key can be used to run a User Defined NMI routine. I have not tried this yet.

5. PRINT #4; POKE 8200, address to be PEEKed. Use this to PEEK into the Cartridge eprom or RAM. Then use USR 110 (eg, LET b=USR 110) to print the PEEKed value. This does a single address PEEK. You are warned not to POKE or PEEK addresses 96 to 111 or the cartridge will crash.

6. LARGE PRINTER DRIVER. Vastly improved, it now does all the essentials: change line length, left margin settings, line feed as well as handle TAB and comma [,]. It will print any GRAPHICS characters in a listing as underlined ASCII characters. (With my printer this turns out to be the ASCII character followed by the underscore character on the [0] key probably because there is no backspace feature included in my printer's specifications.) In my eprom the line length is set improperly at 65. This will probably be corrected in eproms shipped after LARKEN was told about this one. Change any of the settings by PRINT #4; POKE address, value. These new parameters open up all sorts of possibilities: For example, if you like to make copies on your wide printer of your CAT outputs, you can change the line length to say 48 by POKE 16090,47 and print the copy with three columns of file names and blocks used. This done after using OUT 127,15 prints a neat CAT in condensed letters that fits nicely on the diskette jacket. Incidentally, there is a new glue stick called Tack a Note by Dennison that makes paper self-sticking and removable; just perfect for this job.

7. SEQUENTIAL FILES. Further to all the good things mentioned above comes the piece de resistance: sequential file handling. I first tried this by listing a file to a sequential file; here is how it can be done.

8. Type in both listings below and SAVE them. The line numbers in listing #1 are intended to allow it to be MERGED with Tasword. Those in listing #2 are intended to allow it to be MERGED with any listing you wish to send to a sequential file, assuming the latter does not have conflicting line numbers. Now, LOAD listing 2 and MERGE the required program and GO TO 9940 to do the save to a sequential file. Restart the computer and LOAD Tasword. MERGE listing 1 into Tasword assuming again that there is no conflict of line numbers. GO TO 2190 to LOAD the seq. file you just saved. This will be slow (if there is a better way I have yet to find it). When the LOAD is completed, GO TO 10 and the ASCII listing should be there. You will also have the lines from listing 2 but it is no problem to erase them.

LISTING #1

```
2190 PRINT #4; CLOSE #5
2195 INPUT "name? "; LINE n$;
LET n$=n$+" IN "
2200 PRINT #4; OPEN #5,n$
2205 FOR i=33280 TO 49280 STEP
64
2207 LET f=0; LET j=0
2210 LET a$=INKEY$#5
2215 IF CODE a$=255 THEN GO TO
2240
2220 PRINT a$;
2221 IF CODE a$=13 THEN LET f=1;
LET a$=CHR$ 32
2225 POKE i+j, CODE a$
2227 IF f AND j<64 THEN LET
a$=CHR$ 32; LET j=j+1; GO TO
2225
2229 IF NOT f THEN LET j=j+1; IF
j=64 THEN GO TO 2235
2230 IF NOT f THEN GO TO 2210
2235 NEXT i
2240 STOP
```

NOTES

Start here to close stream #5
Enter name of file to be read
IN, ie, loaded.
Open up stream #5.
File is input as code into
memory starting at address
33280. 49280 allows for 250
lines of 64 characters.
255 is the end of the file
sent by LKDOS when saved.
Each character is read in.
When end of line is reached,
set flag f and make 13=32.
POKE this value into memory.
If flag is set and J is less
than 64, pad out the rest of
the line with spaces.
If flag is still not set, then
keep reading in characters.

LISTING #2

```
9940 RANDOMIZE USR 100; OPEN #
4,"dd"
9950 PRINT #4; CLOSE #5
9960 INPUT "name? "; LINE n$;
LET n$=n$+" OUT "
9965 PRINT #4; OPEN #5,n$
9970 LIST #5
9980 PRINT #4; CLOSE #5
9990 STOP
```

NOTES

Initialize use of PRINT #4.
Start here to Save seq. file
Enter name for file save.
Open up stream #5 and SAVE
the listing.
Close the stream.

FIRST IMPRESSIONS

9. This version has just about all that one could wish for. I would have liked to see a Double PEEK or DEEK included but maybe that is where I could use the User Defined Command [DATA] feature.

10. Sequential filing will have to be studied very closely to exploit its capabilities to the full. I'm sure that what I have done so far it but the tip of the iceberg. Other readers should be able to supply some better examples for utilizing this new feature.

11. My congratulations to Larry Kenny for this and all his other innovations that have opened up new horizons for us Timex Sinclair survivors; I heartily recommend you LARKEN owners order the new version 3.

-30-

Bob Mitchell 880805

PASCAL DISK HANDLER
for the
LARKEN DISK OPERATING SYSTEM

One of our club members, David Solly, has a strong interest in the Pascal language (HiSoft Pascal for the Spectrum, cum TS2068), and has been working on PROCedures to allow access to the Larken system.

David Solly is interested in hearing from everyone with an interest in Pascal. His address is 1545 Alta Vista Drive, Apt. 1402, Ottawa, Ont. K1G 3P4.

David has supplied me with an 11-page dissertation on this subject. Much too lengthy to put in a newsletter, but I can send you a copy if you are interested. What follows is an introductory paragraph from it, and also a couple of paragraphs from David's letter to me on the subject.

"Access to the Larken Disk Drive System from within a compiled HiSoft (TM) Pascal program is now possible thanks to the procedures developed by David Solly and Larry Kenny. This article demonstrates how to instal these procedures and gives a practical demonstration on how they may be used within a simple directory program."

Dear George,

I hope by now you have received the copy of my article on accessing the Larken Disk Operating System from within compiled Pascal. I sent a copy of it to HiSoft in England and received a letter from a Mr. Nutkin saying that he found it very interesting. He also sent me the printer patches for TS 2068 Pascal and also for the programs Mon and Gen which are in the 2068 version of Devpac 3. I have included them with this letter on a separate sheet. I have tried the patch for Pascal and it works although I should warn you that the printer driver in Larry's DOS does not seem to be able to handle the left and right curly brackets, therefore; if you wish the document to print out correctly use the option for curly brackets which are "(*" and "*)". Another way to get around the problem is to load in another short printer driver at about address 23320 without opening #3 to the line printer before loading in Pascal. One advantage of Larry's new system when working with Pascal and Basic listings it that you can now spit them out in a detokenized form to the disk and then reload them into M-Script for inclusion in an article. Anyway, I hope to be doing some serious programing in Pascal and to be able to send some useful programs to you and to the Toronto User Group.

Along the same token, Ken Schieman and I are working on a disk handler for Abersoft Forth. I am not so much into Forth as I use to be but I figure that if I can make the Larken system more useful by making it accessible in other computer languages then I figure it will help make the TS 2068 itself more useful and hopefully this will get others interested in programing with the TS 2068.

by George Chambers

VERA CRUZ on the Larken Disk System
by G. Chambers

A number of Spectrum programs, mainly the games variety, are arranged that further blocks of data are loaded within the program, during it's operation. This article describes the method used to convert one of these programs to the Larken system.

VERA CRUZ consists of two program sections. The first section relates to the murder scene, while the setting for the second section is the police station. The LOAD routine that we are interested in is in the Basic portion of the program, so the modification is relatively easy.

Each of the two sections of VERA CRUZ is comprised of a short BASIC loader, followed by several blocks of code, then ended with a larger BASIC program. We shall handle each section separately. In this particular application we have no need of the initial Basic loader, and it will be ignored.

The following details a step by step procedure for each of the two sections.

1. Enter as a single direct command: LOAD "" CODE:
LOAD "" CODE: LOAD "" CODE
2. Load the program from tape. The command in item 1 should load in the three blocks of code in the first section.
3. Save the code to disk with the following commands:
RANDOMIZE USR 100: SAVE "vera.C1" CODE 32768,11150
RANDOMIZE USR 100: SAVE "vera.C2" CODE 64256,205
4. MERGE the longer BASIC program which follows the three blocks of code on the tape.
5. Revise several program lines as follows:
Change: 10 PRINT #4: LOAD "cruz.B1"
Change: 30 PRINT USR 100: SAVE "vera.B1"
Add: 35 RANDOMIZE USR 100: OPEN #4,"dd"
Add: 36 PRINT #4: LOAD "vera.C1" CODE
Add: 37 PRINT #4: LOAD "vera.C2" CODE
Delete: 810
Change 820 Remove part of line 820
(AT 12,10;"AND START TAPE":)
6. Save the program to disk by GOTO 30

Saving the second section to disk is pretty well the same as was done on the first one. There are 5 blocks of code in the second section.
7. Enter CLEAR 49999
8. Enter as a direct command the following:
LOAD "" CODE: LOAD "" CODE: LOAD "" CODE: LOAD "" CODE: LOAD "" CODE
9. Save the code to disk using the following commands
RANDOMIZE USR 100: SAVE "cruz.C1" CODE 50000,6338
RANDOMIZE USR 100: SAVE "cruz.C2" CODE 64256,1112
10. MERGE the Basic program following the code blocks, and modify it as follows:
Change: 30 RANDOMIZE USE 100: SAVE "cruz.B1"
Add: 32 PRINT #4: LOAD "cruz.C1" CODE
Add: 34 PRINT #4: LOAD "cruz.C2" CODE
11. Save the Basic to disk with the command GOTO 30

This completes the conversion. In our next article we will deal with a program in which the LOAD command is contained in the M/C section of the program.

LARKEN RAMDISK
G. Chambers

In the last newsletter I mentioned that I had purchased a RAMdisk from Larken Electronics. I shall give you an update of my experieces with it.

Really, all pretty routine. Mostly fine-tuning my menus and being a bit more selective as to what programs I place on my OMNIBUS disk. I have put Tasword onto the RAMdisk. It seemed to be the program I called up more than anything else. Mostly because of club business, I suppose.

Well, one thing though. I decided that this was a great way to go, but I needed a quad-density drive to hold all the programs that I would like to have at my fingertips! Imagine, twice as many program options!!

One of our members, Richard Hurd, tells me that he has bought a TANDON TM101-4A drive from JB Technologies, and has it up and running. This is a quad drive. I spoke to Larry K., and he says that the TANDON is the only American-made drive I should consider, and further, that I might do well to buy it from Ed Grey Enterprises. So I have sent a money order off for one of them.

Ed Grey is one of the two people in a partnership, Clifford and Grey, that supplied so many of the 2050 modem boards to Timex owners. His address is P.O. Box 2186, Inglewood, CA 90305. His phone No. is 213 759 7406. Call him between 6pm and 9pm. I'll keep you posted.

A TS2068/Larken Program
FILE RECOVERY
by George Chambers

It sometimes happen that a user deletes a disk file, only to realize later that file is still needed. This program provides a simple means of recovering these files.

The file recovery process is based on the fact that even though the disk Directory shows no record of the file it is nevertheless still on the disk. A disk file ERASE command simply places a marker at the start of the file name in the directory track (0). The marker is the value '254'. If you were to do an inspection of file names on the directory track, using the utility 'doctor.B1', you would see that unused name cells and 'erased file' cells have this '254' marker.

This program loads the directory track into a buffer area in the computer, searches the name cells for the desired file, and removes the 'erased' marker. The now-modified directory track 0 is then copied back onto the disk.

You should note that the success of this program depends on the erased file not having been over-written by a subsequently program SAVE. Being overwritten means that the tracks containing the actual program have been reused, and the file is most certianly lost. Sometimes a file may not be overwritten because there were lower-numbered tracks available for storage, in which case file recovery is still possible. The only way to know is to try.

see next page for listing:

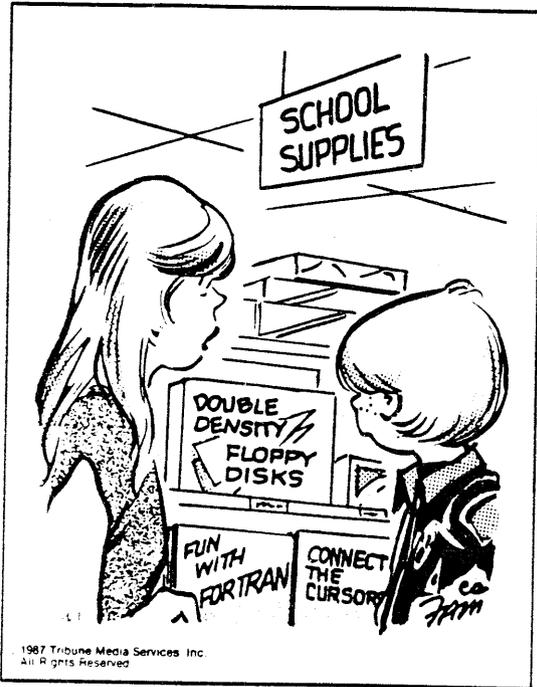
```

100 REM File Recovery Utility for
the LARKEN system
110 REM By G. Chambers
14 Richome Court Scarborough,
Ont. CANADA M1K 2Y1

120 REM Use to recover an accid
entally erased file
130 REM Data statements contain the
"RHMDOS.C2" Code.
140 REM For use with the DSK400 system o
f 5090 bytes/track. Single-,double-,or Q
uad-.
150 GO TO 170
160 CLEAR : RANDOMIZE USR 100: SAVE "recovr.
B1" LINE 170
170 RESTORE 210
180 FOR n=63000 TO 63104
190 READ a: POKE n,a
200 NEXT n
210 DATA 195,43,246,195,72,246,195,104,246,2
43,205,98,0,201,58
220 DATA 100,0,251,201,205,33,246,58,176,92,
50,29,32,205,126
230 DATA 0,205,123,0,33,112,32,17,156,224,1,
0,20,237,176
240 DATA 195,38,246,205,33,246,58,176,92,50,
29,32,33,156,224
250 DATA 17,112,32,1,0,20,237,176,205,150,0,
205,126,0,205
260 DATA 120,0,195,38,246,205,33,246,205,129
,0,205,123,0,33
270 DATA 112,32,17,156,224,1,0,20,237,176,19
5,38,246,201,0
280 DIM e$(9): DIM a$(9): DIM c$(1): DIM b$(
9): DIM d$(256)
290 LET trac=23728: LET nexttrack=63006: LET
loadbuf=63000: LET savebuf=63003
300 BORDER 1: PAPER 1: CLS
310 LET a$=" RETURN "
320 PRINT AT 2,6; INK 2; PAPER 6;"LARKEN DIS
K UTILITY"; PAPER 1;,,TAB 5; PAPER 6;"Delete
d File Recovery"; PAPER 1,,TAB 6; PAPER 6;"B
y George Chambers"
330 INK 3: PLOT 0,108: DRAW 0,62: DRAW 255,0
: DRAW 0,-62: DRAW -255,0
340 INK 6: PLOT 8,114: DRAW 0,50: DRAW 238,0
: DRAW 0,-50: DRAW -238,0: INK 7
350 PRINT AT 10,0;d$;AT 14,1;"Insert disk wi
th deleted file""TAB 2;"and Press any key t
o start": PAUSE 0
360 REM Search of tracks for
erased file
370 INPUT "Enter the full name of the delete
d file ";b$
380 POKE trac,0: RANDOMIZE USR loadbuf
390 LET f=PEEK 57520*PEEK 57521-1
400 POKE trac,1: RANDOMIZE USR loadbuf
410 PRINT AT 10,0;d$;AT 10,10;b$;AT 10,2;"Se
arching tracks for program";AT 13,8;"Track
Program"
420 PRINT AT 14,9;"No. Name"
430 FOR n=1 TO F
440 IF n>1 AND PEEK 57500=0 AND PEEK 57501=0
THEN LET n=79: PRINT AT 10,0;d$;AT 11,1; P
APER 7; INK 2;"File name not found on tracks
": PAUSE 0: STOP
450 LET d=n
460 FOR m=1 TO g
470 LET a$(m)=CHR$ PEEK (57501+m)
490 NEXT m
500 PRINT AT 16,9;n;TAB 14;a$
510 IF a$=b$ THEN LET n=f
520 IF n<f THEN RANDOMIZE USR nexttrack
530 NEXT n
540 REM Search for deleted name in D
irectory
550 POKE trac,0: RANDOMIZE USR loadbuf
560 LET g=57524+d
570 POKE g,245
580 LET c$=b$(1): LET e$=b$(2 TO ): LET e$="
RETURN "+e$
590 FOR n=57688 TO 61088 STEP 34
600 IF PEEK (n+4)=0 AND PEEK (n+5)=0 THEN P
RINT AT 20,3; PAPER 7; INK 2;"Erased Filenam
e not found";TAB 9;"in Directory": PAUSE 0:
STOP
610 FOR m=1 TO g
620 LET a$(m)=CHR$ PEEK (n+m)
625 NEXT m
630 IF e$=a$ THEN LET a=n: LET n=61088: POK
E (a+1),CODE c$: GO TO 640
640 NEXT n
650 LET m=a+11
660 POKE m,d
670 FOR n=1 TO 22
680 IF PEEK (m+n)=249 THEN LET n=22: GO TO
700
690 POKE g+n,245
700 NEXT n
710 PRINT AT 9,0;d$;AT 10,3;"Program has bee
n restored"""" PAPER 5; INK 2;"Remove WRITE
-PROTECT tab from defective disk and insta
ll. """" Press a key to save to track 0
": PAUSE 0
720 POKE trac,0: RANDOMIZE USR savebuf
730 PRINT AT 11,0;d$;AT 13,9;"JOB COMPLETED"
;AT 15,10;"Press a key","to display recovere
d file(only)": PAUSE 0
740 PRINT USR 100: CAT b$,
750 STOP
1000: SAVE "recovr" LINE 100
1010 STOP

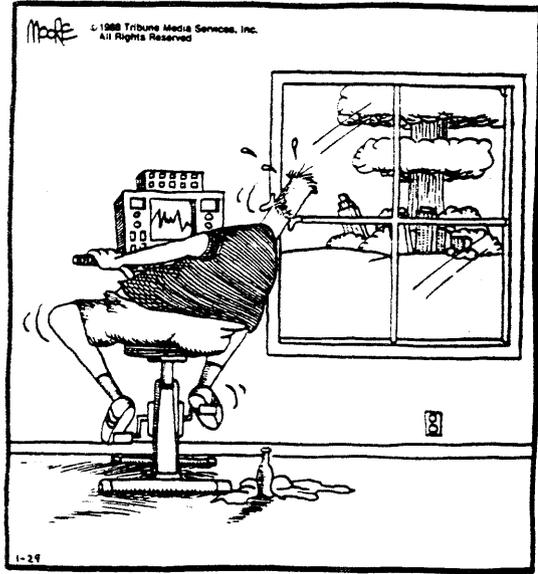
```

On Sale Today



"What happened to the cigar box filled with scissors, paste and crayons?"

In The Bleachers



Bob enters the wrong data into his computerized exercise bike.

Postmaster, if Undelivered Return to :

Toronto Times - Sinclair Users Club
P. O. Box 7274 Stn. A
Toronto, Ont., M5W 1X9
Canada

