SINC-LINK

VOL.8 NO.4

JULY - AUG '90

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

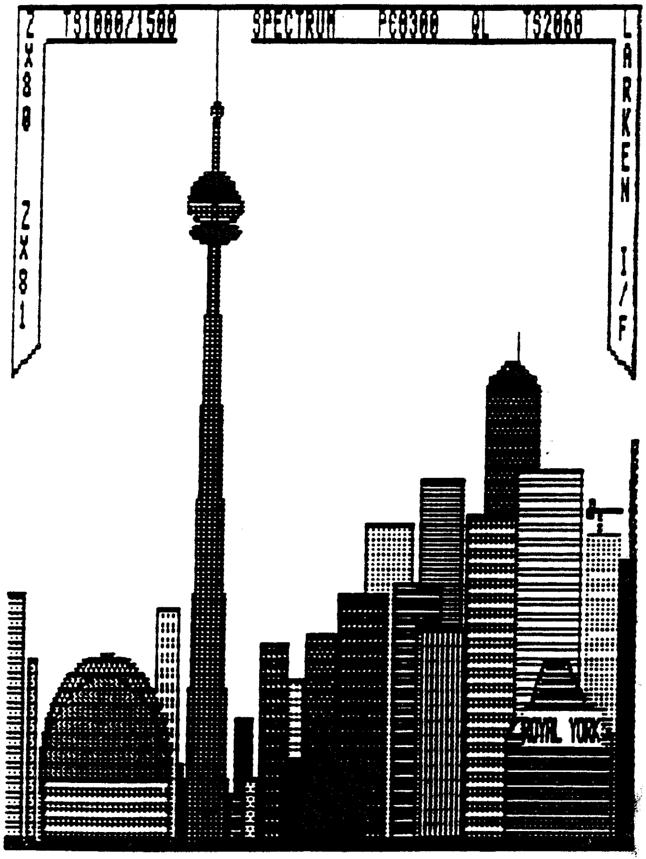
NEWSLETTERS ARE EXCHANGED, FREE OF CHARGE, WITH OTHER TIMEX-SINCLAIR USERS GROUPS.

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

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

SEND CORRESPONDANCE TO:

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



TORONTO TIMEX-SINCLAIR USERS CLUB

EXECUTIVE OFFICERS:

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

TORONTO TIMEX-SINCLAIR USERS CLUB

Editorial

About a year and a half ago I wrote that I had been saved from the frustration of having to rely on tapes to load the 2068 when I purchased a Larken disk drive interface. I was so happy at getting consistent loads and saves in a few seconds that I thought I had reached the ultimate in mass storage technology.

Well folks, I was wrong. Last week I completed my latest hardware project, popped on the chips, plugged it in and turned on the power. Presto! Instant loading and saving. No whirring drives, no external power supply, no worrying about protecting disks, no moving parts.

Of course, I'm talking about the Larken RAMdisk. The same load and save commands are used but the speed just has to be seen. One moment the computer's RAM is empty. Simply request a program from the RAMdisk and it's installed literally in the blink of an eye. For someone like me who is used to chattering disk drives, the silent and speedy execution of this marvel is a little eerie.

One of the other useful features of the RAMdisk is its battery back-up. When the 2068 is powered down, the programs stored in the the RAMdisk remain intact thanks to the onboard 3 volt battery. In fact, I could turn off my 2068, remove the RAMdisk from the edge connector, take the unit across town (or across the country), plug in into someone elses' 2068 and all the data stored on it would still be there for use.

I intend to install my word processing program, a modem program and a suite of utilities on the RAMdisk. With a capacity of up to 256K, several large programs can be put in place for use at any time.

Thanks to Ian Robertson and Rene Bruneau for lending me SRAM chips on short notice. I now intend to utilise the many programs George Chambers and Bob Mitchell have written for the RAMdisk and which are available in the club disk library.

The RAMdisk is not going to obsolete my disk drives but it is going to eliminate some of the drive usage and cut down on my reliance on disks as a storage medium.

Apology

Last issue I promised a review of Byte Power's *"The Print Factory"*. Well, I haven't learned enough of this excellent packege to do it justice, there are so many features. So you'll just have to wait at least one more issue before I produce the review. Sorry.

Did anyone notice the errors in the last issue? The one on the front cover was the most glaring, considering that I had mentioned another group with the same name but no hyphen. Some of the newsletters had their last few pages out of order. Again, sorry. Production problems. We have received some very nice letters commending the May - June issue. To those readers, thanks for taking the time to write.

That's all for now...

J.T.

BOB'S NOTEBOOK

This program was written to keep track of several bank account balances. Note that it is intended only to track current balances. It is written to fit into one block using the LKDOS system; thus it can be stored in RAMDISK without wasting precious track space. It is entirely in BASIC and cannot be compiled as written due the use of the VAL tokens during calculations. This use of VAL strings is not recognized by Timachine but VAL numbers is.

Here is the listing with some comments and additional information at the end. In the listing and comments, the word TOTAL refers to a computation of balance amounts using designated formulae.

```
 100 REM Bank Balances

 110 GO TO 260
 120 IF s$(oa)="." THEN LET s$="
0"+s$
 130 FOR j=oa TO LEN s$
 140 IF s$(j)="." THEN GO TO 170
 150 NEXT j
 160 LET s$=s$+"."
 170 IF s$(LEN s$-oa)="." OR s$(
LEN s$)="." THEN LET s$=s$+"0"
 180 IF s$(LEN s$-oa)="." THEN L
ET s$=s$+"0"
 190 IF LEN s$<>oi THEN LET s$="
"+s$: GO TO 190
 200 RETURN
 210 PRINT #oo;"Calculating..."
 220 FOR i=oa TO a2
 230 LET bt=VAL f$(i): LET bt=IN
T (bt*100+.5)/100: LET s$=STR$ b
t: GO SUB 120: LET t$(i)=s$
 240 NEXT i
 250 RETURN
 260 RANDOMIZE USR 100: OPEN #od
,"dd"
 270 ON ERR GO TO 9005
 280 PRINT #od: CLOSE #oc
 290 IF s=ob THEN CLOSE #ob
 300 IF s=oc THEN OPEN #oc,"p"
 310 IF s=oe THEN PRINT #od: OPE
N #oe,"lp": PRINT #od: POKE 1609
0,47: PRINT #od: POKE 16094,oj:
LPRINT #oe
 320 CLS : PRINT ''"   BANK BALAN
CES: MAIN MENU"'u$;"   Options",
"Status"'u$
 330 PRINT "1> Set Date",d$'"2>
Name File",y$'"3> Save File"''"4
> See Balances"'"5> CHANGE MENU"
'"6> Reset File"''"7> Set Displa
y",("SCREEN" AND s=ob)+("TS2040"
 AND s=oc)+("WIDE" AND s=oe)''"0
> Quit"
 340 PRINT u$
 350 PRINT 'TAB of;"Press choice
 (0-7)": PAUSE oo: LET k=CODE IN
KEY$-48
 360 IF k=oa THEN INPUT "date? "
; LINE d$
 370 IF k=ob THEN INPUT "name? (
max 6) "; LINE y$
 380 IF k=oh THEN ON ERR RESET :
 CLS : LIST : STOP
 390 IF k=oc THEN GO TO 1020
 400 IF k=od THEN GO TO 680
 410 IF k=oe THEN GO TO 570
 420 IF k=of THEN GO TO 460
 430 IF k=og THEN INPUT ("displa
y mode?"'"2=Screen  3=TS2040  5=
Wide ");s: IF s=oc THEN PRINT #s
 440 IF k=oo THEN PRINT #od: GO
TO od: PRINT #od: NEW
 450 GO TO 260
 460 CLS : CLEAR 65535: INPUT "h
ow many accts? (max 10)";a1: INP
UT "how many totals? (max 7)";a2
: IF a1>10 OR a2>7 THEN GO TO 46
0
 470 LET oo=NOT PI: LET oa=SGN P
I: LET ob=oa+oa: LET oc=ob+oa: L
ET od=ob+ob: LET oe=od+oa: LET o
f=oc+oc: LET og=of+oa: LET oh=od
+od: LET oi=oh+oa: LET oj=oe+oe
 480 DIM c$(a1,21): DIM b(a1): D
IM b$(a1,oi): DIM t$(a2,oi): DIM
 g$(a2,16)
 490 DIM f$(a2,50): LET u$="====
=============================="
 500 DIM m$(oe,of): LET m$(ob)="
SCREEN": LET m$(oc)="TS2040": LE
T m$(oe)="WIDE"
 510 LET s=ob: LET y$="...nil":
LET d$="setdate": GO TO 520
 520 CLS : FOR i=oa TO a1: PRINT
 i;TAB 2;c$(i): NEXT i
 530 FOR i=oa TO a1
 540 INPUT ("Enter name/acct id
for rec ";i)' LINE h$: IF CODE
 h$=oo THEN GO TO 560
 550 LET c$(i)=h$: PRINT AT i-oa
,ob;c$(i)
 560 NEXT i: PAUSE 100
 570 CLS : PRINT ''"BANK BALANCE
S"'"CHANGE MENU"'u$
 580 PRINT "1> Main MENU"'"2> Ac
ct Names"
 590 PRINT "3> TOTAL Names"'"4>
TOTAL Formulae"'"5> Balances"
 600 PRINT u$''"Press choice (1-
5)": PAUSE oo: LET k=CODE INKEY$
-48
 610 IF k=oa THEN GO TO 320
 620 IF k=ob THEN GO TO 520
 630 IF k=oc THEN GO TO 960
 640 IF k=od THEN GO TO 890
 650 IF k=oe THEN GO TO 810
 670 GO TO 570
 680 LET tb=23: CLS : PRINT #s;T
AB of;"BANK BALANCES ";D$: PRINT
 #s'
 690 FOR i=oa TO a1
 700 IF c$(i,oa)=" " THEN PRINT
#s: GO TO 730
 710 IF s<>oe THEN PRINT #s; INV
ERSE oa;i; INVERSE oo;TAB ob;c$(
i);TAB tb;b$(i)
 720 IF s=oe THEN LET tb=27: PRI
NT #s;i;TAB od;c$(i);TAB tb;b$(i
)
 730 NEXT i
 740 PRINT #s;TAB of+od;"TOTALS"
: FOR i=oa TO a2
 750 PRINT #s;g$(i);TAB tb;t$(i)
 760 NEXT i
 770 IF s<>ob THEN PRINT #s''''
 780 PRINT #oo;AT oo,oo;"Press a
 key": PAUSE oo: GO TO 320
 810 CLS : LET tb=23: PRINT : FO
R i=oa TO a1: PRINT i;TAB ob;c$(
i);TAB tb;b$(i): NEXT i: FOR i=o
a TO a1: INPUT ("Enter balance
of Acct ";i), LINE h$: IF CODE
 h$=oo THEN GO TO 850
 820 LET b(i)=VAL h$
 830 LET s$=STR$ b(i): GO SUB 12
0: LET b$(i)=s$
 840 PRINT AT i,tb;b$(i)
 850 NEXT i
 860 GO SUB 210
 870 GO TO 320
```

(bank balances - cont.)

```
 890 CLS : PRINT "Enter all elem
ents as b(x)      where x is the
 acct number."'': FOR i=oa TO a2
: PRINT i;TAB oc;f$(i): NEXT i
 900 FOR i=oa TO a2
 910 INPUT AT oo,oo;("Enter TOTA
L formula# ";i)' LINE h$: IF CO
DE h$=oo THEN GO TO 940
 920 LET f$(i)=h$
 930 PRINT AT oa+(i*ob),oc;f$(i)
 940 NEXT i
 950 GO SUB 210: GO TO 320
 960 CLS : FOR i=oa TO a2: PRINT
 i;TAB ob;g$(i): NEXT i
 970 FOR i=oa TO a2
 980 INPUT AT oo,oo;("Enter TOTA
L name# ";i)' LINE h$: IF CODE
h$=oo THEN GO TO 1000
 990 LET g$(i)=h$
1000 PRINT AT i-oa,ob;g$(i)
1010 NEXT i: PAUSE 100: GO TO 57
0
1020 INPUT "drv? (0-4) ";drv: PR
INT #od: GO TO drv
1030 ON ERR GO TO 9005: PRINT #o
d: SAVE y$+".Bi"
9005 ON ERR RESET : PAUSE 60: ON
 ERR GO TO 9010
9010 CLS : PRINT AT 21,oa;"Enter
 password "
9015 POKE 23658,oo: POKE 23607,2
50
9017 INPUT LINE p$
9018 POKE 23607,60
9020>IF p$="password" OR p$="unl
ock" THEN CLS : ON ERR RESET : G
O TO 260
9030 ON ERR GO TO 9005
9040 STOP
```

Once entered into the 2068 and saved, reset the computer, reload the file and initialize it using option 6 on the MAIN MENU to erase all data and variables in the file.

Enter the number of bank accounts (up to ten) that you wish to track. These could include bonds, chequing and savings accounts, RSP accounts, etc. There is room for 21 characters when describing each account (bank, acct number, type of account, etc). See example a.

Next, enter the number of TOTAL's you want to calculate (up to seven). Establish the names of these TOTAL groupings (option 3) and then develop the formulae for these (Option 4). See example b.

The balance for each account is stored in numerical array b( ); a grouping might be b(1)+b(3) or b(1)*1.2. There is room for 50 characters in these formulae. See example c.

Finally, enter the balances (option 5 on the CHANGE MENU). DO NOT ENTER BALANCES UNTIL OTHER OPTIONS HAVE BEEN ENTERED. You can view your balance sheet on the screen, TS2040 or Wide Printer. (Option 4 on the MAIN MENU).

The program uses ON ERR which has to be handled gingerly to avoid a lock-up, so I have built in two escape routes; this is particularly useful during altering or de-bugging the BASIC. One is at the MAIN MENU where pressing <8> will do an ON ERR RESET and LIST the file; the other is at the end of the listing (lines 9005 to 9040) which allows you to hold down the BREAK key to get back to BASIC. Even so, make sure you save your balance sheet before playing around with the program. If you want to take out these escapes, delete line 380 and delete <PAUSE 60> in line 9005.

While you are looking at those lines, note line 9020 which contains the password privacy codes. You can replace the two codes with anything you like. This does not provide full security for your information because anyone familiar with the LKDOS system can break in during the LOAD function. You could also read the data using Disk Doctor. However, these codes offer a reasonable degree of privacy.

When changes are made to any data, the current data are listed on the screen and a prompt is shown at the bottom to enter new data. If an item is correct as displayed, simply press the [ENTER] key to step to the next item. This technique is used several times, ie, at line 520 for name and acct i.d.; at line 890 for TOTAL formulae; at line 810 for balances and at line 960 for TOTAL names.

The following techniques are used to display data on the screen as quick.. as possible:

1) Justification to express amounts in dollars and cents is performed at line 120 with decimal points being aligned and any necessary trailing zeros added; formulae are evaluated at line 210 to 250; these subroutines are located early in the listing to cause the least delay. Note line 230 which in evaluating the string f$ ensures that only two decimal places will be displayed. Remember, at every GO SUB command, the TS2068 starts at the first line and looks for the GO SUB line number so the earlier it finds it, the quicker the results.

2) all calculations are performed at the time the changes are entered; leaving them to the time of display would speed up entry time but slow down the display.

2068

Now, some more notes on specific lines:

270 ON ERR GO TO de-activates BREAK key; ON ERR RESET re-activates it.

290-310 LKDOS printer codes are set up depending on value of <s> determined in line 430.

320-450 MAIN MENU: Includes current status information (date, file name, display device. Line 440 as written reloads an autostart program in RAMDISK and may be changed to suit (or ON ERR RESET: STOP).

460-560 RESET FILE: This lets you start a fresh file. Line 470 sets variables for numbers 1 to 10 to save bytes. Arrays are established: c$ (acct names); b (numeric balances for computation) b$ (balances derived from b( ) and justified for display); f$ (TOTAL formulae for computation; g$ (TOTAL names); t$ (TOTAL values derived from f$( ) and justified for display).

680-780 DISPLAY BALANCES: Lines 710 and 720 change display for wide printer. The LKDOS printer driver does not like INVERSE in LPRINT statements or indeed any attribute commands such as PAPER, INK etc. Was this an oversight, Larry Kenny?

910 ENTER TOTAL FORMULAE: You may use all the arithmetic functions here, ie, add <+>; subtract <->; multiply <*>, etc.

```
1 TD100 Chequing
2 TS200 Savings
3 CT $US
4 CT Chequing
5 CT GIC
6 CT RSP

Example a
```

```
Cash on hand
Investments
$US
$US in $CAN
Net Worth $CAN

Example b
```

```
1 b(1)+b(2)+b(3)*1.2+b(4)

2 b(5)+b(6)

3 b(3)

4 b(3)*1.2

5 b(1)+b(2)+b(3)*1.2+b(4)+b(5)+b
(6)

Example c
```

QL "TRY-IT"

This ONE-LINER is rather peculiar.

'copy con to ser'

The cursor will disappear, and anything you type will go straight to the printer when you press <ENTER> provided your printer is attached to ser1. ***WITHOUT APPEARING ON THE SCREEN***

While you are still in this mode, try 'lrun xxxx_something', and the command goes to the printer !

Using SCR instead of CON will give you 'Bad name'.

As I understand it, SCReen works as a display device, whereas CONsole, acts as a separate terminal.

To break out, press CTRL-SPACE.

Try it !

H.H.H.

SOME POPULAR MISCONCEPTIONS (Mostly about filenames).

From time to time I read in an article or letter about the QL statements that I know to be untrue or at least misunderstood. Sometimes the same misconceptions occur over and over again. The following comments arise from a recent (Jan 90) issue of QL World and the final issue of a US newsletter called Quantum Levels, which started with high hopes as a bimonthly in August 1986 but only managed 12 issues in three years.

Basic Filenames.

Many people seem not to realise that there are two ways of presenting QDOS commands with filenames, and that the rules for allowable characters differ between them. In each case there is a limit of 36 characters in addition to the five which define the device (e.g. mdv2_).

1. The normal method of supplying a "parameter" consisting of unadorned ascii characters when only normal letters, numbers and the underscore character "_" are accepted, e.g. LOAD mdv1_my_prog1.

2. QDOS will also accept strings, otherwise it would be almost impossible to write file handling programs. (I think early versions of the QL only worked this way.) By a string I mean

i. a set of characters in quotes e.g. SAVE "flp1_!@#$",

ii. a string variable to which a string of characters has been assigned e.g. COPY a$,b$

iii. a string function e.g.

```
                    MERGE dev$&"_"&p$&CHR$(233).
```

In these cases the only restriction is that the first five characters must be a legitimate device name followed by an underscore, the rest can be anything you can type in at the keyboard, and even unprintable characters (use CHR$() as above). Thus you could create files with unprintable names that would be almost impossible to delete without reformating the medium - if there were any point to this! Some commercial programmers have used filenames consisting entirely of spaces, which are invisible in a normal listing - I once discovered one on a disk someone sent me that read "FORMAT flp1_:FORMAT flp2_" that was named "   "; luckily I discovered it using my Ftidy program before it was activated.

As far as I can see these rules apply to all commands that take filenames in the QL ROMs and in Toolkit II, such as RENAME.

FORMAT also works in the same way as far as medium name is concerned; that's how they got the date in the form 9\9\85 on the early mdv cartridges.

You can even put one totally nameless file onto each medium; e.g. SAVE flp1_ creates such a file. The file can be BASIC, text or machine code, and is available to any of the commands referring to a single file name. However, DIR, and any commands or programs which make use of the directory, fail to recognise the file at all. If you suspect such a file may be lurking about on one of your disks/cartridges you can reveal it by COPY flp1_,mdv2_secret (or something similar - COPY flp1_,scr_ also works). If a file called "secret" appears on your target device, then it is secret no more and now available to all the normal manipulations.

Psion Filenames.

It is fairly common to read that Quill files must have the extension _doc, export files must have the extension _exp etc., This is not so, these are merely the defaults that the Psion suite uses if provided with nothing else. While you are restricted to a maximum of eight normal letters and numbers for the main part of the name and there must be a thre character extension you can use any three characters that can be obtained from the keyboard for the extension. You must, however, type them all in when you want to save or load the file; "letter_@#$" is quite acceptable as a quill file name for example. You can import any ascii text file into quill too, but it must also fit these rules, so you may have to COPY it to a regulation filename (or rename if you have the facility.) It is the file header itself that distinguishes quill files from abacus files etc. and not the extension as is often thought. I use a three character date code on all my letters e.g. Michael_B12 means a letter to Michael written on November 12th (to me at any rate!)

More Psion tricks

Shift + F5 refreshes the screen in all the programs, but for some reason is undocumented. (Much better than F2 twice!)

Have you discovered that CTRL + down-arrow deletes from the cursor to the end of the line, and CTRL + up-arrow from the cursor to the beginning of the line?

Howard Clase,
Box 9947, Station B,
St John´s,
Newfoundland,
CANADA, A1A 4L4.

e-mail: hclase@kean.ucs.mun.ca.bitnet
Tel: (N. Am.) (709) 753-6415
(U.K.) 0101 709 753-6415

1990.04.10

323 1/2 N. Church Street
Bowling Green, OH 43402
March 20, 1990

Dear George,

Being a mathematician, I tend to do a lot of fooling around with DEF FN and FN. You may have seen my letter on some of the stranger examples in ZX Computing about 3 years ago. Those weren't exceedingly practical - they took too long to return an answer. But they were intended to illustrate just how much you could actually do with the DEF FN command. If written in the normal way, they would have taken several lines.

If you want a simpler (and more useful) function, try this one out:

```
                    10 spaces
   1 DEF FN l$(a)=("          "+STR$ a)(LEN STR$ a TO)
```

This function produces a string of 11 characters which ends in the value of the variable a. In other words, it could be used for left-justifying numbers, if you want them to line up in nice columns. If you will be using relatively small numbers, you could drop a few of the spaces - just make sure there is one less than the final length you want.

I have in various places collections of dozens of DEF FN functions, though as I said most of them are not necessarily practical. Oh, if you are trying to figure out why a DEF FN might take a bit of time and can't find the letter in question, it is because the functions involved something called recursion, in which a function may call itself several times in order to obtain an answer. Naturally, the more times the function has to call itself, the longer it takes. It is a sophisticated approach which lets you put the equivalent of a FOR/NEXT loop in a function, but it is somewhat tricky as well. Among other things, you have to design the function very carefully to avoid being caught in an endless loop.

I should give you an example. Some of the more sophisticated computers have a built-in function called INSTR. This function takes as parameters 2 strings and a number, and returns a number which is the first occurence of the second string in the first string, after the position corresponding to the number. If it isn't found, a value of 0 is returned.

As a subroutine, such a function would look like the program below. Here, a$ is the first string, b$ is the second string, and s is the number input, and pos is the result.

```
1000 FOR i = s TO LEN a$-LEN b$+1
1010 IF a$(i TO i+LEN b$-1)=b$ THEN LET pos=i: GOTO 1030
1020 NEXT i: LET pos=0
1030 RETURN
```

That is actually quite a lot going on there. If it were written on another computer, I would have had to include one more IF to be sure s<=LEN a$-LEN b$+1, I would have to be sure a$ was long enough to possibly have b$ in it. On our computers, if s is too big, it will automatically skip the loop.

To do the same thing in a single DEF FN statement is complicated, of course, but it can be done. I have to specifically check all the possibilities. And I have to also use the VAL function to make sure I can get out of the loop. If I referred to the function directly, there would be no way out of the recursion.

So, without further ado, here it is:

```
5 DEF FN p(a$,b$,s)=VAL (VAL$ (("(""s"" AND a$(s TO s+LEN b$-1)=
b$)+(""FN p(a$,b$,s+1)"" AND a$(s TO s+LEN b$-1)<>b$)") AND s<=
LEN a$-LEN b$+1)+"+"""""")+"+0")
```

As I said, it is extremely complicated. One thing that makes it especially complicated is that I have to make sure that if s>LEN a$-LEN b$+1 it will not attempt to evaluate any of the subscripted strings. The "+"""""" and "+0" are there to avoid errors trying to evaluate an empty string. Let me try and walk through it for you.

There are 3 cases to consider. If s is too big, the whole thing collapses to VAL (VAL$("+"""""")+"+0"). The VAL$ produces an empty string - it evaluates the string +"" - so that the VAL only evaluates +0 to get 0. That is as it is supposed to be. If s is small enough, then the VAL$ recieves the string "(""s"" AND a$(s TO s+LEN b$-1)=b$)+(""FN p(a$,b$,s)"" AND a$(s TO s+LEN b$-1)=b$)+"""""". Of course, the first and last quote are removed, and all the double quotes are converted to single quotes.

This is where the other two cases come in. We already know that s is small enough. If at that location, we can find b$ in a$, the VAL$ simply produces "s", which has 0 added to it and is evaluated to give the answer. If b$ is not at that location, the VAL$ produces "FN p(a$,b$,s+1)", which the VAL then evalutes to determine if b$ can be found at the next position (and the next, and the next, until we hit one of the other cases).

I admit, this is a rather extreme example, and hence will evaluate slowly if it can't find the answer the place it checks. I presume if a$ was too long, it might actually produce an Expression too complicated error. Needless to say, that is not a problem with using an actual subroutine.

Anyway, enough fun for one day. That particular function I think I would never try and put to any practical use, but would instead use the subroutine. There are examples where the subroutine is less practical or almost impossible where this could be done, but they are unusual mathematical functions which might take too long to explain. Oh, needless to say, even the more sophisticated computers would have trouble handling.

I have not recieved the March issue of Sinc-Link yet, but I presume it is about as slow as usual and I will get it in the next couple of days. I'll be mailing this from Windsor, where I am spending my March Break, so I won't really know if I got it for a little while anyway. Take care, and Peace!

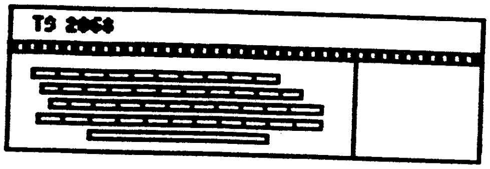
P.S., I just finished checking it out. I originally had one too many "(" in the DEF FN, but I have corrected it. And it performs as billed. If you were to use it to search a string of about 150 characters for the last character, it would take roughly a minute. Actually, that isn't quite as bad as I expected, but it seems longer than it actually is.

Sincerely,

Steven V. Grinhouse

WORD-MASTER - A REVIEW

By Lionel M. Keeping



This page was printed with the 2068 in SPECTRUM mode on a Panasonic KX-P1180 9-pin printer, using the WORD-MASTER suite of programs from Britain. If this is the first time you have seen such output, then, like me, I am sure you are impressed. If word processing or Desk Top Publishing is your requirement, this review is of interest to you.

In my humble opinion, this is the best word processing package available for the SPECTRUM emulated 2068 today. I have used MSCRIPT and have had a fleeting try at TASWORD II. MSCRIPT is a very good program and, as a word processor may be slightly better than WORD-MASTER, but with the graphics capability and the other features available through the extension programs, WM wins hands down.

First of all I will discuss WM itself. Its major difference from the other programs is the file handling capability. It allows more than one file in RAM at once (it can be many more, depending on size, up to the available 29k). These may be other text files (created with other word processors like MSCRIPT), graphic files (may be created with artist type programs like ART STUDIO, which was used to create the graphic of our favourite computer), fonts (user defined or otherwise), or other applications programs that operate with WM (to allow its amazing output). It also allows several text files to be linked to form one larger file.

As a word processor, WM has all the required amenities, or is lacking in only minor points. The features include:

1. File Handling:

a. create a file
b. load a file (including SCREEN$) from disk or tape
c. change drives
d. get a file already in RAM
e. delete a file from RAM
f. erase a file from disk
g. catalogue the drive
h. link files

2. File Options:

a. alter screen columns from 16 to 64
b. alter screen colours
c. delete all but ASCII codes in a file
d. find a selected page in the file
e. save or rename the current file
f. exit to main file handling screen
g. exit to print options
h. begin writing

3. Edit Options:

a. insert and typeover modes
b. insert command lines (or non printing comments) to control special options (see 4 below)
c. fast scroll through text
d. imbed control characters into text (much easier than MSCRIPT)
e. underline words (visible on screen)
f. delete / undelete word or line
g. block move, copy, delete, save
h. search / replace with or without auto replace and case sensitivity and with a "smart" feature that will search on a lowercase string but capitalize the first letter at the begining of a sentence!

4. Command Lines:

a. place non printing comments in text
b. change pitch by typing "elite", "pica", "condensed", "large", or "normal"
c. change justification by simply typing "fill", "centre", "right", or "left"
d. change margins ("margin nn", nn in 10ths of an inch) or columns ("column nn", nn between 16 and 255)
e. define up to 7 special characters which may be used to print otherwise unavailable characters
f. send printer control codes using "lprint n,n...." (up to 16 codes per line)
g. select "wide" (double spacing, fill justification)
h. "reset" the printer
i. force a page break using "form" for form feed
j. select "draft" or "nlq"

5. Graphic Printing:

a. print a graphic, from RAM, with the text
b. vary the width and height of the graphic when printed
c. print text to left or right of the graphic
d. print graphic with shading to approximate colours

6. Headers and Footers:

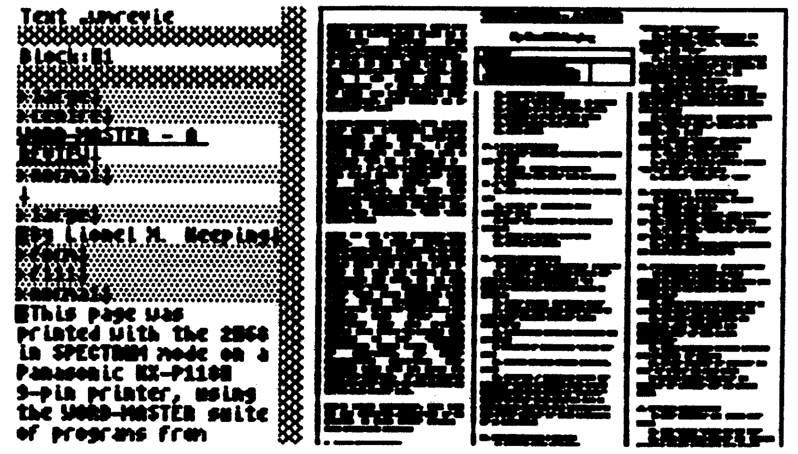
a. create headers and footers which may be called up from RAM to print in the document
b. move the printhead to a selected line before printing
c. styles and pitches in headers and footers are independent of the main document
d. send control codes within the header or footer
e. print header or footer to left or right of the page
f. print page number in header or footer (or in main text)

7. Print Options:

a. set number of lines per page
b. set form feed on or off
c. use fanfold paper or single sheets (with prompt for next sheet)
d. select page number start
e. print selected pages

f. set line spacing.

As you can see, the features are numerous. To help you keep track, WM prints two or three lines at the bottom of the screen to remind you which mode you are in and which keys perform the functions. For example, while writing, the information lines tell you if Caps Lock is on, whether you are in insert mode, how many columns are displayed, if fast scroll is on, that [Shift 9 - GRAPHICS] gives printer control options, that [Shift 2 - EDIT] calls up the delete options, that [Sym Sh W - DRAW] enables the block function and that [Sym Sh E - REM] allows the search function. It also tells you the number of words in the document (at the last save), the number of characters, and the amount of free RAM or other information, depending on the feature selected.



Text .wmrevie

Block:01

WORD-MASTER - A REVIEW

Mrs Lionel M. Keeping

This page was printed with the 2068 in SPECTRUM mode on a Panasonic KX-P1180 9-pin printer, using the WORD-MASTER suite of programs from

PAGE LAYOUT SCREEN

There are similar help lines for the other options, including the PRINT mode, FILE HANDLING mode, FILE OPTIONS mode, DELETE options, and BLOCK options.

All in all, the features are very extensive and the program is relatively easy to use, but, as with any software with so many features, it takes a bit of getting used to (ever try using WORDPERFECT on an IBM without something to tell you what all the function keys mean?!).

If the foregoing were the only benefits of WM, you would probably be as well off with MSCRIPT, except for its lack of graphics capability. However, it is the extension programs that make this whole thing worthwhile.

There are two major programs that make this package work. They are TYPELINER and HEADLINER. TYPELINER is the desktop publishing program that produced this page after it was typed with WM. HEADLINER is a graphic creation program that allows use of large fonts and graphics to create graphic files.

The first of these, TYPELINER, is the showcase of the package. It takes the "raw" text produced on WM and allows you to manipulate it virtually any way you wish. It does this by means of user defined "blocks" for the page layout.

There are three types of blocks - Text, Graphic or Box. The first, the Text block, must be used to contain all text on the page. This is not as restrictive as it sounds. Rectangular blocks may be created and/or deleted very easily, resized, renumbered, and moved around the page.

The page is an area of about half the screen, set up on the right of the monitor. All blocks are visible on the page when being created or modified but disappear when the page is "Viewed" prior to printing. To "View", V is pressed and the text automatically "pours" into the text block and fills it, stopping when the block is "full". Text may be forced into another block by using the "FORM" command on a command line in the text. This is all visible in the preview. (See the illustration below.)

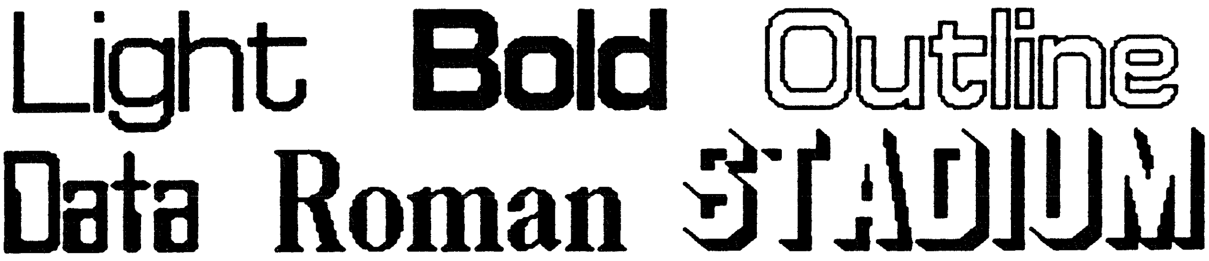
Graphic blocks are created by pressing SHIFT 9 (GRAPHICS). When the name of a graphic in memory is entered, it fills the block created. These may also be moved and resized. The restriction here is that the graphic must be a file created with HEADLINER. This is also quite simple, and when a file is created from a screen, it actually compresses the number of bytes used to allow you to store more screens (or parts of screens) than would normally be possible. For example, the graphic of the page layout screen you see above is stored in only 5139 bytes with HEADLINER after being created from an NMI SCREEN$ save using LARKEN DOS. Unfortunately, HEADLINER loses the bottom 2 lines of the SCREEN$ when it loads it from disk.

Boxes may also be created. There are six styles of Box available - horizontal line, vertical line, rectangle, thick line rectangle, shaded line rectangle and double line rectangle. These are used to place borders around text or whatever other use may come to mind (for example, separating columns of text).

While using TYPELINER, the text may be edited at any time simply by pressing "E". You may also move to any block of text before editing by stepping through the blocks using the "N" and "M" keys. Each text block may have its own justification, either fill, left, right or centred, by using command lines in the text. Fonts may be selected (12 are provided with the package) using number codes corresponding to their place in the fonts listing. The fonts used may be varied at will and may also be modified using the font editor so you can create your own personal font (on a 24 by 24 point grid).

HEADLINER, as you might expect, may be used to create headlines. It has a limited graphic capability, allowing straight lines, rectangles, arcs and circles to be drawn. When printed with TYPELINER, each graphic is printed with two passes of the print head (text is printed with three passes!) so the quality is quite good. Shading is possible and the user may define his own patterns for shading. There are 6 large fonts (6x6 characters) supplied - Light, Bold, Outline, Data, Roman and Stadium (see the illustration below). Each may be varied in height and width to two sizes and may be printed in Bold and Italics. A program is also provided to allow conversion of other fonts (1x1 character only) like those used with PIXEL PRINT by Stan Lemke.

I could go on and on about these spectacular programs but the best way to find out about them is to try them yourself (your printer must be capable of ESC "K" - 60 dpi and ESC "L" - 120 dpi graphics and ESC "J" - n/216ths line feed). They are now for sale in North America by Jack Dohany who, apparently, has made some modifications to add extra features. If anyone would like to contact him, his address is - Jack Dohany, 435 Woodward Way, Athens GA 30606. He also supplies extra fonts.



HEADLINER FONTS

File Header Reader
Larken TS2068
by G. Chambers

Here is yet another header reader. But this one has some interesting features about it that will interest you. Even if you have a header reader which satisfies you this program has a number of programming techniques which are worth more than a passing glance.

The program POKEs code into a part of the memory which we seldom use, namely just below the system variable area. This was selected purposely. If you have a Larken RAMdisk this block of code can be stored as an integral part of the OMNIBUS program. In fact the code can be stored in any AUTOSTART program, and used independantly of this Basic program. When stored there, a simple PRINT USR 24800 will produce a near-instant screen listing of the disk in the drive the Larken system is currently pointing to. If necessary, change the drive by prefacing the command with a PRINT USR 100: GOTO x (where x = the selected drive number).

Put the code into an AUTOSTART program by first running this program to get the code into the 24800+ area of memory. Then do a SAVE with the command 'PRINT USR 100: SAVE "code.Cl" CODE 24800,178'. Load your AUTOSTART program, and then load the just-saved "code.Cl" file. Now reSAVE the AUTOSTART program (with the NMI-button and "D" key, of course). It's done. Test it by breaking out of the program, and doing a PRINT USR 24800. Or if the AUTOSTART is a menu program you could introduce this function as one of the menu options.

If you are using the OMNIBUS AUTOSTART menu program and have stored the code as described above you can "REM" LINE 7 of this program, since you have no need to move the code. This will speed program initialisation.

The address 24800 was purposely selected. The AUTOSTART program on the OMNIBUS disk (TTSUC Larken library disk #2) has a number of other m/c utilities stored in the memory area 24311/24989, and this was the last remaining space! If you wish to store the code in another location you may do so by using the m/c program reloc8.Cc, found on the OMNIBUS disk also.

This program can print to the screen, to the TS2040 printer, or to a large printer. It does this by pointing channel #2 to the desired output device. Normally channel #2 points to the screen, and that is where you see listings, etc. In the program LINE 330 causes the output to appear on the 2040 printer, while LINE 340 points it to the large printer. After the disk has been read LINE 370 points the computer back to the screen.

This program has been set up to work with an SCM Fastext 80 printer. You will probably need to customise it to suit your printer. The arrangement allows you to selct a normal printout, or one in compressed text form. Look at lines 290, 440, 460, and 470. LINE 290 first does a GOSUB to 460 (to restore the large printer to normal operation), then does a GOSUB 440 (if compressed text mode has been selected). The lines 440, 460, and 470 are essential to ensure the printer functions are properly set. Line 470 is used to delay computer operation until the printer indicates it is ready for the next printer instruction. Without it, the computer sends the instructions quicker than the printer can accept them, and they are missed. You will probably need to modify the value of 108 shown in line 470. You need to determine what code your printer returns to the computer as a "ready" signal. To do this, do the following test. With an empty computer, enter the following program line:

```
 10 PRINT AT 10,10;"    "; AT 10,10;
 IN 127: PAUSE 50: GOTO 10
```

Note the numbers that appear when you printer is turned on and off. The number you should put into you program is the one that appears while your printer is on. Some printers require that paper be present before they are really ready to go. Put paper in!

LINE 300 provides for printing two labels side by side on a large sheet of paper, as a paper-saving measure. Print labels on the left side of the sheet, then roll the paper back and print on the right-hand side. The values shown in the line (36 and 68) can be changed to print at the desired point on the paper.

LINE 420 will need to be modified to suit your particular system arrangements.

```
 110 BORDER 1: PAPER 1: CLS
 120 PRINT AT 2,2; INK 2; PAPER
7;" LARKEN DISK UTILITY   v3.0 "
;AT 4,7; PAPER 2; INK 7;"File He
ader Reader";AT 6,3; PAPER 2; IN
K 7;"  1990  George Chambers "
 130 INK 7: PLOT 0,108: DRAW 0,6
2: DRAW 255,0: DRAW 0,-62: DRAW
-255,0
 140 INK 7: PLOT 8,114: DRAW 0,5
0: DRAW 238,0: DRAW 0,-50: DRAW
-238,0
 150 RESTORE 160: FOR n=24800 TO
 24978: READ a: POKE n,a: NEXT n
 160 DATA 205,134,97,175,50,29,3
2,205,126,0,205,123,0,33,136,32,
17,112,52,213
 170 DATA 175,18,35,126,254,250,
40,14,254,253,32,246,35,126,254,
249,40,240,18,19
 180 DATA 24,234,209,26,167,202,
186,0,33,29,32,190,56,9,40,11,24
5,205,129,0
 190 DATA 241,24,241,119,205,126
,0,19,213,205,123,0,33,114,32,6,
6,126,254,46
 200 DATA 40,6,35,215,16,247,24,
5,62,32,215,16,251,6,3,126,35,21
5,16,251
 210 DATA 33,124,32,205,94,97,19
5,82,97,24,187,1,1,0,33,134,32,2
05,94,97
 220 DATA 62,13,215,195,10,97,62
,32,215,78,35,70,33,9,0,205,138,
97,33,99
 230 DATA 0,205,138,97,33,231,3,
205,138,97,33,15,39,205,138,97,5
8,100,0,251
 240 DATA 205,233,48,205,161,49,
243,195,98,0,167,237,66,216,62,3
```

```
2,215,201,0,0
 250 RANDOMIZE USR 100: OPEN #4,
"dd"
 260 LET drv=0: PRINT #4: GO TO
drv
 270 POKE 23658,8: INPUT "Select
  2)Screen 3)2040  5)Large ";ptr

 280 IF ptr=5 THEN  PRINT #4: PO
KE 16094,8: INPUT "Choose Column
: L)eft  R)ight ";b$
 290 IF ptr=5 THEN  INPUT "N)orm
al or C)ompressed ";c$: GO SUB 4
60: IF c$="C" THEN  GO SUB 440
 300 IF ptr=5 THEN  IF b$="R" TH
EN  PRINT #4: POKE 16094,(68 AND
 c$="C")+(36 AND c$="N")
 310 PRINT AT 20,0;"Drive Now>";
drv: INPUT "Select Drive ";drv
 320 IF drv<0 OR drv>4 THEN  GO
TO 310
 325 PRINT #4: GO TO drv
 330 IF ptr=3 THEN  OPEN #2,"p"
 340 IF ptr=5 THEN  PRINT #4: OP
EN #2,"lp"
 350 CLS : PRINT : POKE 23658,0:
 INPUT "DISK TITLE ";C$: PRINT "
Disk Name: "; C$
 360 PRINT AT 2,0: RANDOMIZE USR
 24800
 370 PRINT #4: CLOSE #2
 380 IF ptr<>2 THEN  PRINT AT 16
,5
 390 POKE 23658,8: PRINT ''TAB 1
0;"End of file"'"    Press a key
 to continue": INPUT "another? Y
/N ";b$
 400 IF b$<>"Y" AND b$<>"N" THEN
  GO TO 380
 410 IF b$="Y" THEN  CLS : GO TO
 270
 420 PRINT '"Going to RAMdisk in
 10 seconds": PAUSE 500: PRINT #
4: GO TO 130: PRINT #4: NEW
 430 GO TO 380
 440 GO SUB 470: OUT 127,15: GO
SUB 470: OUT 127,27: GO SUB 470:
 OUT 127,48: GO SUB 470: RETURN
 450 STOP
 460 GO SUB 470: OUT 127,18: GO
SUB 470: OUT 127,27: GO SUB 470:
 OUT 127,50: GO SUB 470: RETURN
 470 IF IN 127<>108 THEN  GO TO
470
 480 RETURN
 490 PRINT
8000 INK 0: PAPER 7: BORDER 7: C
LS : LIST
8010 STOP
9000 PRINT USR 100:SAVE "header.
B2"LINE 100
```

```
************************************************
******************** CORRECTION *******************
*********** OTHER USES FOR LARKEN RAMDISK *********
**************** ARTICLE MAR-APR ' 9Ø *************
************************************************
```

Most humble apologies to anyone who tried to create a BASIC AROS by following my instructions. The following corrections will remove some of the frustration: 1. The LOAD statements in two of the sections of the article will need to have RANDOMIZE USR 1ØØ added. 2. In the "CREATING AN AROS" section: line 3Ø should have FOR X= 32768 not 62768. The instruction in para 6 should read: OUT 244, 24Ø: RANDOMIZE USR 1ØØ: LOAD "test.Cl" CODE 32776: OUT 244,Ø. Once again, sorry!! Larry C

****

In addition to George Chambers Modifications to Pixel Print + I have found one more which is quite important:

In Line 8420 change the

"GO TO" to a "GOSUB".

I noticed this error when I went to LOAD an Icon and the 2068 gave me a "RETURN WITHOUT GOSUB" error.

I recieved my copy quite recently so I assume that most of you might still have this Problem.

Andrew Zettel

PIXEL PRINT PLUS

Note: The Pixel Print Plus suite of programs includes an Icon Library. The instructions for this library, called the 'Icon Library Manager User Guide' were originally on paper. One of our members, Les Cottrell, has recently converted them to PP+ format. This 'page' has been added to our PP+ library disk #10. We are including print-out of this page in the newsletter for the benefit of members who may have received a copy of PP+ (minus this sheet) from our disk library.

SEE NEXT PAGE

Icon Library Manager User Guide

The ICON LIBRARY MANAGER is, as it's name implies,an ICON librarian. With this program, you are able to browse thru a file of up to 115 ICONS: view each one, enlarge it, place a box around it, print it (TS 2040 printer only), save it out in SCREEN$ or the PIXEL PRINT DESKTOP PUBLISHER format

When LOADed, the main library screen will be displayed. Once you have loaded in a library, ICON 1 will be shown, as well as the first 10 ICON titles. If you do not load a library the ICON screen and title blocks will be empty. Near the bottom of the screen a MENU line will be displayed (when it is available..... ICON not being drawn). To select any item, just press the key associated with the first letter of the option.

ICON (I)

ICON prompts you to select a new ICON from the library. INPUT a number between 1 and 115. Your selection will be drawn and the title displayed (above the ICON). Note: The 'librarian' keeps track of the next new ICON number that you select (to reduce the SAVE/LOAD time associated with the library), so you must LOAD ICONS sequentialy.

LOAD (L)

LOAD allows you to LOAD a new ICON into the library. Use the ICON option to set the library pointer to the position you wish to store this ICON. This can be at the bottom of your library, or as a replacement for an existing ICON. Then select LOAD by pressing 'L'. You will be asked to INPUT an ICON name/title of 6 characters or less. Type in the desired name.

After the name/title is INPUT, you will be asked to verify that you really want to LOAD an ICON...PRESS ENTER to continue with the LOAD. At this point, if you press any other key, the LOAD function is cancelled, but the name/title will be placed in the file library. This is an effective way of changing an ICON name without changing the ICON.

SAVE (S) -- Pixel Print format

Save allows you to SAVE the bytes of the selected ICON. The ICON will be saved using the existing ICON name/title in the library.
NOTE: This is the same ICON format used by the Pixel Print DTP.

COPY (C)

COPY provides you with a variety of functions in addition to sending a graphics screen-dump to the TS 2040 printer. You can select the desired size (1 to 3 times the standard size), place the ICON anywhere on the screen, and draw a box/frame around it if you desire. You can also choose to SAVE the SCREEN$.

ICON SIZE?

INPUT the desired ICON size (1 to 3 times normal). SIZE 1 is 32 pixels wide, 58 pixels high. SIZE 2 is 64 wide and 116 pixels high. SIZE 3 is 96 pixels wide and 174 pixels high.

INITIAL COLUMN?

INPUT the pixel column (0 for the left screen edge, 255 for the right screen edge) that you want to be the left-most column for your ICON. If you specify a column that will place part of the ICON off the screen, it will be adjusted as needed automatically.

INITIAL ROW?

INPUT the pixel row (175 for the top of the screen, 0 for the bottom of the screen) that you want to be the top-most row of your ICON. Again, if you specify a row that places your ICON off the screen, it will be adjusted for you.

BOX BORDER? (Y/N)

Press Y to place a 'BOX' around your ICON. Press any other key for NO BOX

When the ICON is completed as you defined it.....you will be given another 3 item menu...'SCREEN$ COPY RETURN'. Select SCREEN$ by pressing 'S' to SAVE the SCREEN$. You will be asked for the SCREEN$ name.... INPUT a name 6 characters or less. Select COPY by pressing 'C'. This sends a screen-dump to the TS 2040 printer. Select RETURN by pressing 'R'. This will RETURN you to the library screen/menu.

PRINT (P)

PRINT will give you a TS2040 printer listing of the ICON library names in your file. This is a handy reference to print and keep on hand.

QUIT (Q)

Quit will take you to Basic. You will be asked to PRESS ENTER to QUIT. Press any other key to RETURN to the library. If you PRESS ENTER you will be asked if you want to SAVE your LIBRARY before 'stopping'. Press 'Y' to SAVE your ICON library. Press any other key to STOP.

NOTES:

At any LOAD/VERIFY error (or after 'quitting'), you can re-enter the library simply with RUN (ENTER).

You can make a personal copy of the ICON LIBRARY MANAGER after using the QUIT command with: RUN 9999 (ENTER).

Improving the TS1000

R. Bruneau

Many articles have been done about improving Uncle Clive's "Door Stop" ever since he introduced the ZX-81 way back in the early 1980's. Some improvements, particularily the commercial products, require little or no modifications to the basic computer. Others, such as the John Oliger Video Upgrade, the Bent Rom, and internal 64K memory reqiure some trace cuts and soldering.

In general, improvements to the TS1000 can be divided into three categories:

1. Basic Modifications
2. Memory and Display
3. Peripheral Devices

All of the following improvements are covered in magazine or news letter articles or technical books available from the club book library or from club members.

Basic Modifications:

- Stronger power supply (1 amp. or greater) with a power switch.
- Replace the heat sink on the voltage regulator with a larger one to run cooler.
- Install a reset button to recover from crashes.
- Change the components of the LOAD and SAVE circuitry to improve the loading and saving of programes.
- Mount the computer and external add-ons on a rigid base.
- Install a real keyboard by hardwiring it to the CPU board. Improve performance by replacing the keyboard diodes with a LS245 IC to buffer the keyboard.

Memory and Display:

- Install a composite video output and purchase a composite monitor ($50.00 or less from surplus).
- Install 8k static ram internally in the 8-16k area of the TS1000 memory map. Added advantage: Hi-rez graphics are now possible.
- Install up to 192k of static ram and eproms internally. Hi- rez graphics.
- Replace the ROM with an EPROM (the BENT rom is an example).

Peripheral Devices:

- Printer interface for big printers
- Keyboard interface for those who don't want to modify their computers.
- Programable interfaces for process control

  Imagine a TS1000 monitoring the heating and air-condtioning in your house, activating shades, attic fans, solar heating.......
- Convert the TS1000 into a colour computer (the John Oliger Upgrade)

The C Page,

By Tim Swenson

Hopefully last month's column perked your interest in C. This month we'll take our first look at the language.

C is a structured language similar to Pascal. Many of the same ideas and contructs are used. In C, all variables must be declared before used. The idea of functions encompasses SuperBasic's ideas of procedures and functions.

C has FOR, WHILE, and REPEAT UNTIL loops. IF statements are treated the same as in SuperBasic. C also uses begin and end marks the same way that Pascal uses BEGIN and END. C uses { and } for this.

C is a very terse language that is not at all as easy to read as english. Where some Pascal programs can almost be read as straight english.

Let's take a look at program 1. This is a short program that will print out the string "Hello World" on the screen.

The /* and */ mark the begin and end of comments. The include statement loads in a standard file that is used by C. Any file that ends in _h is usually a header file that includes things like constant definitions.

The heart of the program starts with main. Every C program must have a main function. The compiler starts execution with the function main.

After the begin mark ({) comes the single line of code. Printf is a C library function that prints data to the screen. C is not very powerfull and is made mostly of functions. A lot of necessary functions are in the standard C library.

At the end of the program comes the end mark. As is, this program will compile and run (I've tried it).

Now that you have a better grasp of the structure of C, next month we'll try a bigger program.

```
/* Program 1. */

#include <stdio_h>

   main()
      {
         printf("Hello World\n");
      }
```

The **QL LIBRARY** continues to grow.

We have had a number of significant additions in recent months, some of more or less general interest, others of more specialised interest. All are welcome be they small or large. So please send in your latest creations, if you like it, you can be sure someone else will like it.

From Real Gagnon in Montreal who was the publisher/editor of QL-DOC, we have a nice little Archive programme which will allow you to keep track of your Video Collection We have a number of his productions in the library already. Thank you Real.

Howard Clase in Newfoundland has sent in a nice little programme which gives you the power to design your own windows in SuperBasic without having to go through a whole stack of co-ordinates. This can be merged into your own progrwmmes. He calls it 'GLAZIER'. Nice name, nice to use, and Oh so easy. Makes a tough job interesting.

Howard has also given us permission to distribute his programmes which are in the QUANTA library, this is a real honour. One is a calendar program, then we have something to test your chemistry knowledge. A routine to close an Archive file left open. Fount designers. A set of procedures useful to beginners. Lots more. This should let you see how diversified Howard is. Welcome to the club Howard.

From Arnold Nieuwennoff in Sutton, MA. USA we have a couple of adaptations from 2068 to QL. One shows Standard Time and Star (or Sidereal Time) running to-gether. The other will calculate the distance in miles or kilometers of any two points on earth, a real must for the armchair traveller, and does that not just about cover us all?

Then Timothy Swenson in Washington who has contributed in the past, has let us have a compiler, 'Small-C' which should be of special interest to many. Tim has spent a lot of time on this. Fills a 40 track 720 disk.

I can handle 5 1/4 40 or 80 track, or 3 1/2 disks. For cartridges you can work it out for yourselves. Don't forget the postage. (USA please send cash or money order.) Don't forget formatted media.

The clubs thanks go to all contributers. If I have missed anyone or anything, my apologies. Blast me for it. Please---

Hugh H. Howie. QL Librarian.

QL News Chaos Buster

A little while ago I purchased Chaos Buster from Bill Cable, Wood and Wind Computing. A really worthwhile collection of softwares designed to operate within Archive. Included with the disk were two public domain programs, the first one ARITHMETIC is a combination of the four basic operations ADDITION, MULTIPLICATION, SUBSTRACTION and DIVISION . The results are displayed and it is possible to obtain a print-out to track your progress or lack of !!!.

The next program is called TASKET and is a multitasking software similar to TASKMASTER. Both of these programs as well as the associated DOC files have been placed in our QL Library and are available from H.Howie under the usual conditions.

Louis Laferriere

From the QL Library.....................Random Blocks

Quantum Levels had this spectacular demo by Frank Toemay of Quantum Computing.

This is interesting in B & W, but in colour, it is outstanding. All the colours the QL can produce are depicted in a rapidly changing pattern. There are three patterns to select from.

There is also a YELL which shrieks along with the demo. This shriek can be stilled by eliminating a few lines in the listing.

H.H.H.

See next page

```
10 REMark by Frank Toemay
20 REMark of Quantum Computing
30 REMark From Quantum Levels
40 REMark This is a small demo
50 :
60 WINDOW #0,512,45,0,211
70 WINDOW #1,512,209,0,0
80 WINDOW #2, 512,187,0,22
90 PAPER #0,0:PAPER #2,0
100 INK #0,7 : INK #2,7
110 BORDER#0,2,180: BORDER#2,2,22
120 :
130 DEFine PROCedure Rbks
140         BORDER 4,6:PAPER 7,1,2: CLS
150        FOR 1=0 TO 255,254 TO 0 STEP -2
160           BEEP -1*10,0,1,1/2,1+(1/44),1/18
170           BLOCK RND(10 TO 375),RND(10 TO 145),RND(5 T
O 107),RND(5 TO 47),1
180      END FOR 1: PAUSE 250
190 END DEFine Rbks
200 :
210 :
220 :
230 DEFine PROCedure layers
240       SCALE 175,0,0:BORDER 5,2
250       OVER -1:RANDOMISE RND*63333
260   FOR pass =1 TO 3
270       e=RND(15 TO 155): d=RND(25 TO 244):c=RND(4 TO 8
): PAPER RND(0 TO 7):CLS
280       FOR b=0 TO 170 STEP c*5
290         INK RND(0 TO 255)
300 FOR a=0 TO 275 STEP c:LINE d,e TO a,b:BEEP -70,0,10+
(a*b)/2,1+b,1+(a/40),a/60
310        LINE 0,b TO a,b:FOR blp=1 TO 11:BLOCK 490,195,
0,0,RND(0 TO 255)
320        BLOCK 490,195,0,0,RND(0 TO 7)
330      END FOR b:PAUSE 44+(pass*10)
340 END FOR pass:OVER 0: PAUSE 200
350 END DEFine layers
360 :
370 :
380 REMark *** The Menu Area ***
390 REPeat Demonstrate
400 MODE 0:BORDER 5,56:INK 0:PAPER 4:CLS
410 CSIZE 1,0:PRINT\\\,'Baby Demo Graphics Menu'\\
420 PRINT\\,'1) Random Blocks'\\,'2) Kalidoscopic Blocks
'\\,'3) Run Layers'\\,'4) Quit':BEEP
430 q$=INKEY$(-1):IF q$<'1' OR q$>'4':GO TO 430
440 SelVal=q$:SELect ON SelVal
450           =1: MODE 8:Rbks
460           =2:MODE 8:OVER-1:Rbks
470           =3:MODE 8:layers
480           =4:CLS:PRINT#0\\\\,'All Finished!':EXIT Dem
onstrate
490      END SELect
500 END REPeat Demonstrate
510 :
520 STOP
530 :
```

CQ QL SOS

Sharps were offering QL's for less than $100.00 apiece, so I bought two, one for each child. My suspicion was that microdrives would last not much longer than cartridges to put into my TS2068. According to the British magazine QL World for February, 1990, you will not be able to buy any more QL microcassettes.

So when Miracle Systems offered their Trump Card with dual disk drive at a sale price, I dug into the piggy bank I keep in the UK and purchased one.

Everything seemed to work fine; lights came on, drives whirred and all that stuff, but when I tried to format a disk ... FORMAT FAILED. About then I read in the QL World that older ROM's might not be able to cope with high speed drives. Back to friendly Sharps to buy a new ROM complete with Eidersoft's I.C.E.

Everything seemed to work fine; except that there was a two inch gap along the top of my TV screen and the bottom part of I.C.E.'s display was cut off. After a lot of trial and error I managed nevertheless to reach the FORMAT icon, invisible though it was. Guess what ... FORMAT FAILED. Any suggestions?

The two inch gap syndrome also happened with a game. In another game the screen display would not hold vertically. I presume other people have had similar problems. Any help would sure be appreciated.

Robin BEAUMONT
226 Borden Drive
YELLOWKNIFE, NT
X1A 3R2

323 1/2 N. Church Street
Bowling Green, OH 43402
April 16, 1990

Dear George,

Here is the ml version of Tas>MS, as I talked about in my second letter past. I have gotten Bob Mitchell to test it out for me, and he says it "works very well and very fast."

You will also find a pseudo-assembly listing of the two code routines. Since I don't actually use an Assembler, I find it easier to do my "assembly" listings in something that comes a bit closer to BASIC than any typical assembler. It is easier for me to read, and hopefully easier for anyone else to understand as well.

As far as what this code does. It implements a few other suggestions I made to Bob concerning his original program.

The code has been designed to perform best with unjustified text. As such, it therefore only removes spaces which occur at the end of lines. This means that if the text includes any tables, it will not remove the formatting from between table entries. As with Bob's original code, however, it does add a space at the end of every line if the text calls for it.

In addition, it automatically takes care of most paragraphs. If the paragraph is seperated by a blank line or is indented, it sees a line that begins with a space, which signals the code to put a CHR$ 13 in the file, which MScript uses to seperate paragraphs. The only place this will usually get you into trouble is where the entire paragraph is indented - a format used for quotations in formal papers - or in tables or listings. In the latter, some or all of the lines might not begin with a space, so you will need to add the ENTER manually.

I am not sure how fast the original Timachine compiled version of Tas>MS was, but I am sure this will be much faster. On the files I tested it on, it was too fast to time. Bob tells me it did a 18000 byte file - about the maximum size it can handle - in just over a second. Either way, if you blink, you will miss it. That, and you don't have to have a copy of Timachine to compile it, are the main reasons I thought this was a good thing to write in ml.

The first routine in the listing has the sole purpose of clearing the area the Tasword file will occupy. Among other things, this allows the second routine to automatically find the end of the file, which it does in the first 7 lines of code. So to run it, all you need is the name of the Tasword file. Also, the code returns the length of the result directly, so there is no need to have a PEEK for this.

I have included some error checking in the BASIC to create the Tas>MS, so you don't have to find out the hard way that one of the numbers is wrong.

I hope this program will be useful for the other members. I don't have MScript, but I might write another version for my word processor. Either way, it is short and quick enough that anyone should be able to find space for it. Oh, don't try to use it on the same file twice, the results might be a little strange. It SAVEs its files with the same ".CT" that Tasword uses, so make sure you know whether or not the file is actually a Tasword file if you might have changed it already.

Well, I guess that is all I have to say about it. I hope you had a pleasant Easter. Looks like I have certainly done my part to keep you busy this issue. Peace, and God bless.

Sincerely,
Steven V. Cunhouse

P.S., I will print the rest of this in 12 pitch, since I think that will look better for the newsletter.

Machine Code: (for unjustified text)

```
First Routine: clear buffer
FF00 21 00 6C     :HL = 27648  (Start of Tasword memory area)
FF03 54           :D  = H
FF04 5D           :E  = L
FF05 13           :DE = DE+1
FF06 36 20        :POKE HL,32 (CODE " ")
FF08 01 00 4B     :BC = 19200
FF0B ED B0        :Block Move
FF0D C9           :RETURN (to BASIC)

Second Routine: Reformat file
FF14 21 FF B6     :HL = 46847
FF17 3E 20        :A  = 32 (CODE " ")
FF19 2B           :HL = HL-1
FF1A BE           :IF A = PEEK HL (A is CODE " " throughout)
FF1B 28 FC        :THEN GOTO FF19
FF1D 23           :HL = HL+1
FF1E 22 56 FF     :POKE 65366,HL
FF21 21 00 6C     :HL = 27648  (Start of Tasword text file)
FF24 11 4F B7     :DE = 46927  (Start of MScript memory area)
FF27 06 00        :B  = 0
FF29 0E 40        :C  = 64
FF2B BE           :IF A <> PEEK HL
FF2C 20 06        :THEN GOTO FF34
FF2E EB           :Exchange DE and HL
FF2F 2B           :HL = HL-1
FF30 36 0D        :POKE HL,13
FF32 23           :HL = HL+1
FF33 EB           :Exchange DE and HL
FF34 ED B0        :Block Move
FF36 EB           :Exchange DE and HL
FF37 2B           :HL = HL-1
FF38 BE           :IF A =PEEK HL
FF39 28 FC        :THEN GOTO FF37
FF3B 23           :HL = HL+1
FF3C 77           :POKE HL,A
FF3D 23           :HL = HL+1
```

```
FF3E E5           :PUSH HL
FF3F 2A 56 FF     :HL = PEEK 65366
FF42 A7           :A  = A AND A (Clear carry)
FF43 ED 52        :HL = HL-DE
FF45 E1           :POP  HL
FF46 EB           :Exchange DE and HL
FF47 30 E0        :IF DE <= End THEN GOTO FF29
FF49 EB           :Exchange DE and HL
FF4A 11 4F B7     :DE = 46927
FF4D 36 00        :POKE HL,0
FF4F 23           :HL = HL+1
FF50 A7           :A  = A AND A (Clear carry)
FF51 ED 52        :HL = HL-DE
FF53 44           :B  = H
FF54 4D           :C  = L
FF55 C9           :RETURN (to BASIC)
```

ML loader program:

```
  10 LET t=0
 100 FOR i=1 TO 86
 110 READ a
 120 POKE 65279+i,a
 130 LET t=t+i*a
 140 NEXT i
 150 IF t<>428022 THEN BEEP 1,1:
 PRINT "Error in DATA!": STOP
 200 RANDOMIZE USR 100: SAVE "Ta
s>MS.Cm" CODE65280,86
 500 DATA 33,0,108,84,93,19,54,3
2,1,0
 510 DATA 75,237,176,201,0,0,0,0
,0,0
 520 DATA 33,255,182,62,32,43,19
0,40,252,35
 530 DATA 34,86,255,33,0,108,17,
79,183,6
 540 DATA 0,14,64,190,32,6,235,4
3,54,13
 550 DATA 35,235,237,176,235,43,
190,40,252,35
 560 DATA 119,35,229,42,86,255,1
67,237,82,225
 570 DATA 235,48,224,235,17,79,1
83,54,0,35
 580 DATA 167,237,82,68,77,201
```

BASIC program to control ml

```
 100 RANDOMIZE USR 65280
 110 RANDOMIZE USR 100: CAT ".CT
",
 200 INPUT "File Name? ";LINE n$
 210 IF n$="" OR LEN n$>6 THEN G
OTO 200
 220 RANDOMIZE USR 100: LOAD n$+
".CT"CODE 27648
 250 LET len=USR 65300
 260 INPUT "SAVE file as: ";LINE
s$
 270 IF s$="" THEN LET s$=n$
 280 IF LEN s$>6 THEN GOTO 260
 290 RANDOMIZE USR 100: SAVE s$+
".CT"CODE 46927,len
 300 PRINT #1;"LOAD another file
? (Y/N)"
 310 PAUSE 0: LET s$=INKEY$: IF
s$="Y" OR s$="y" THEN GOTO 100
 320 IF s$="N" OR s$="n" THEN ST
OP
 330 GOTO 310
9800 RANDOMIZE USR 100: LOAD "Ta
s>MS.Cm"CODE
9810 GOTO 100
9900 RANDOMIZE USR 100: SAVE "Ta
s>MS.Bm"LINE 9800
9910 RANDOMIZE USR 100: SAVE "Ta
s>MS.Cm"CODE 65280,86
```

ANLQ WITH YE OLDE FASTEXT 80

by

Philip Hudsmith

As we all know the Fastext 80 is great for draft quality. If only it would give us NLQ we'd all be delighted. Not only that, we'd shelve our plans to buy an NLQ printer and buy a Larken Interface instead. Well - maybe.

Button up your seat belts. It is possible to achieve ANLQ or almost near letter quality provided you have time and patience and a good eye. All you have to do is pass the same sheet through the Fastext about 3 times, making certain that the registration is relatively accurate.

I printed this letter up just that way so you can see what it looks like. Remember I said ' Time & Patience.' No problem at all. All Timex Sinclair Users have that.

TS2068 ROM DISASSEMBLY

prepared by
BOB MITCHELL
20 Wild Briarway
WILLOWDALE ONT
M2J 2L2
900611

Recently, I decided to develop a disassembly of the code contained in the TS2068 ROM. The procedure required the use of the Larken Sequential/Random filing system which is a software product available from Larry Kenny. The LKSR code allows for the saving and loading of as much data as can be held by one disk; whereas the Version 3 EPROM system is limited to 21 tracks per file name. Armed with both systems, I proceeded as follows:

STEP 1
LOAD DISASSEMBLER. OPEN SEQUENTIAL FILE VIA LKDOS.

My procedure for capturing the ROM consisted of first using the Larken Version 3 EPROM RND/SEQ commands to read the output from a disassembler onto a disk.

For a disassembler, I chose Monitor (aka Spectramon) with its deficiencies but with the advantage of sending a line feed every 32 characters. This character (13) provided a separator as required when using the Larken LBASE system. Monitor is a version of Spectramon which appeared in Personal Software back in 1983; I had converted it to TS2068 and mended a few errors in the mnemonics code, then compiled it using Timachine to make it run acceptably fast. (HOT Z is a very good disassembler but it sends no separator at the end of each disassembled line; it did not therefore fill the bill for this particular application.)

Having loaded Monitor, I then broke to BASIC and did <RANDOMIZE USR 100: OPEN #5,"ROM1 OUT"> to a selected drive with a fresh disk inserted; the drive turns as the file is opened. I returned to Monitor and started the readout at 0000H, setting the limit at 3CFF. The SAVE went very well with the drive whirring every little while as a track was filled; that is up to 21 tracks. At that point, the computer gave me a "DISK FULL" error report. I tried various tricks to get more than 21 tracks to load into one file name on the disk, but to no avail. (I later confirmed by telephone with Larry Kenny that 21 tracks is indeed the limit using the LKDOS EPROM.)

I continued on saving sections of code until I had finished the job: in four sections of about 20 tracks each. Each section required a different file name + OUT; ie, "ROM2 OUT","ROM3 OUT" and "ROM4 OUT". This ended the use of the LKDOS EPROM commands.

STEP 2
READ SEQUENTIAL FILE INTO LBASE VIA LKSR COMMANDS.

Now, I started to read these four files into LBASE (my 512 byte version); this employs the special Larken Seq/Rnd (LKSR) code imbedded in a REM line at the start of LBASE. This code must be purchased from Larry Kenny; he also provides a rudimentary form of LBASE which I developed into my version.

To do this, I MERGEd a special line 8030 contained on the disk as lbase.BM. Entering <GO TO 8000> started reading and saving 16 lines (512 bytes) into each record; as each file was read, the additional records were added onto the end of the previous data. There was now no longer a problem exceeding 21 tracks.

When all was done, I loaded the complete single file into LBASE and scanned the results particularly around the "join" areas between the files. I found that I had missed two bytes in one instance and quite a few in one other case. I don't know why. Further examination revealed that the SAVE via the EPROM had indeed caught all the bytes; those that went missing seemed to have done so during the loading into LBASE. So my routine in line 8030 is suspect; however, I did not try to do anything with it as I could not see what I could do.

I got around it by writing some additional BASIC in LBASE to read and save records from a start number to an end number, using the TAB command and a simple FOR...NEXT loop. I read and saved all the records up to an including the one just before the missing two bytes. I then loaded this portion into LBASE proper <GO TO 1> and did an ADD to put the missing bytes into their own 512 byte record; next I started again with the next portion from just after the missing two bytes up to where I had missed quite a few. I repeated the same procedure, adding manually the code for missing bytes.

This area turned out to be the "constants" table and the "calculator" table which I typed in as DATA. I knew what the missing bytes were by having a copy of the disassembly of that area done with Monitor. Compressing the DATA into fewer lines left several records with redundant DATA which I erased line by line and simply marked them by entering "Blank Record".

There is a routine in my LBASE to copy a file and skip some records; this was a rather tedious job with close to 600 records and only a handful of blank records to erase, but it had to be done so I went ahead.

STEP 3
EDIT THE TEXT.

Now I had the rough ROM disassembly in LBASE format as opposed to LKDOS EPROM format. It now needed some considerable editing because there were still numerous tables and floating point calculator instructions that had to be deciphered manually. Fortunately, I had a copy of a Spectrum ROM Guide to help me locate and decode the tables and fp instructions. Most of these followed the Z80 instruction RST 28H (from two to many bytes) which turns on the fp calculator; as well as some single byte error codes following RST 08H. Some disassemblers identify these bytes as DEFB but Monitor does not. In any case, locating RST 08H and RST 28H always identified the following bytes as DEFBs.

During this process, I had to use some short forms to conserve space on individual lines and these were for the fp instructions. Here are a few examples with some brief notes:

DEFB CODE	MEANING	SOME SHORT FORMS
38H	fp exit	fp ex fpex fpx
(this is the byte that turns off the fp calculator)		
04H	multiply	mult mul
C3H	store 3	store3 stor3 st3
(store 3 puts a result into Memory area 3 of 6)		
31H	copy	cpy cy
(this really makes a duplicate of a result)		
01H	exchange	exch exc
(a becomes b and b becomes a in Memory)		
E2H	get 2	get2 gt2
(fetches the byte stored in Memory 2)		
00H	j true	j tru jtru jtr
(jump relative if the result is true (1); followed by a displacement byte, say +4)		

Toni Baker wrote a series of articles for ZX Computing in 1986 which gave very good coverage of the Floating Point Calculator and if you are interested in this subject, you should have that series.

SUMMING UP

Finally, I had all the tables and the fp instructions done; what you would see on the disk as the file "ROM" is the result. It is quite conceivable that there are some errors in my work. Using LBASE, it is easy to correct them and I'll do that as they are discovered.

My intention in doing this exercise was to build the ROM Guide to be able to use it mainly via the screen, yet to make copies as necessary to do a closer analysis of a particular area of code.

Making a complete hard copy was not my intention but it could be done. There would be some gaps wherever there were blank lines on the screen. Use <PRINT USR 100: LPRINT "name" to make the hard copy.

The code now uses up 55 tracks for over 273 kilobytes, ie, almost 280,000 bytes. It is difficult to imagine how all this data could be captured, edited, searched and saved to disk with the limited memory of the TS2068. I don't have many requirements for storing such a lot of data but this example shows that it can be done using the Larken LBASE code and the enhanced BASIC that I have developed and used.

I intend to provide a copy of the disassembly on disk to the club library; it may be read or printed using the Larken LKDOS PRINT or LPRINT commands, but this is rather tedious and it would be better to access it using the LBASE program. To repeat, you must purchase the copyrighted code from Larry Kenny and then use my LBASE which I will provide to bona fide owners of the code.

END

PIE CHARTS and MULTIPLE SCREEN DUMPS

In this monthly contribution, I want to demonstrate in particular the capability of the TS2068 for constructing PIE CHARTS from a given set of values. The program is arranged as a tutorial so that one may substitute other items and values and use it as a utility for a statistical program.

In this illustration we assumed seven items: Salaries, Insurance, etc. The number of items can be any thing but such a chart usually accommodates up to 10 items without crowding. If you have a different number of items, change the values in program lines 20, 30, 100, 300, etc. to suit.

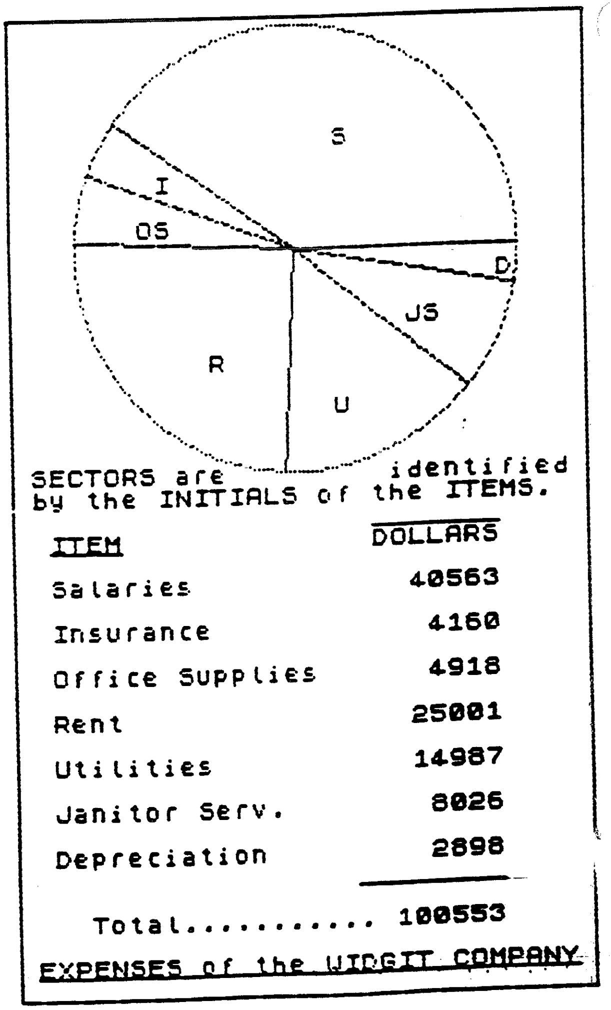
Program line 50 contains a correction factor "n" which allows the resulting pie to look like a circle instead of an ellipse. As you know, circles do not reproduce undistorted on the prinout. As we cannot conveniently correct this distortion in the CIRCLE algorithm, it was not used.

This exercise also demonstrates the capability of the TS2068 to produce printouts of more than one screen dump. In this instance two screens were dumped onto the printer in succession, giving a unified look to the chart and accompanying text. This scheme can be used for more than two, if desired, by a continuation of the process. Refer to program lines 180 and 400. When the run STOPs at these points, get the printer ready, and by direct entry do a CONTinue....ENTER. Better yet; just delete the STOP commands in these lines and the figure builds up automatically.

Now, how many other computers can do this with such a simple and short BASIC program?

Warren Fricke

```
   2 REM ** MODEL of a PIE CHART
   3 REM ** "C-63", 10-7-87, WF
   5 REM ** UPPER BORDER
   7 DRAW 0,175: DRAW 255,0: DRA
W 0,-175
  10 REM ** IDENTIFY and
            EVALUATE ITEMS
  20 DIM q$(7,15): DIM r$(7,5):
LET T=0
  30 FOR a=1 TO 7: READ q$(a): R
EAD r$(a): LET T=T+VAL r$(a): NE
XT a
  40 REM ** UNDISTORTED CIRCLE
  50 LET n=1.23: REM Cor. Factor
  60 FOR a=0 TO 2*PI STEP .02
  70 PLOT 127+n*80*COS a,92+80*S
IN a: NEXT a
  80 REM ** DRAW SECTOR RADII
  90 LET a=0: REM Starting Angle
 100 FOR p=1 TO 7
 110 LET a=2*PI*VAL r$(p)/T+a
 120 FOR R=0 TO 80
 130 PLOT 127+n*R*COS a,92+R*SIN
a
 140 NEXT R: NEXT p
 150 REM ** IDENTIFY SECTORS
 160 PRINT AT 5,18;"S";AT 7,8;"I
";AT 9,7;"OS";AT 15,11;"R";AT 17
,18;"U";AT 13,22;"JS";AT 11,27;"
D"
 170 PRINT AT 20,1;"SECTORS are"
;AT 20,21;"identified"
 175 PRINT AT 21,1;"by the INITI
ALS of the ITEMS."
 180 STOP : COPY : CLS
 255 REM ** LIST ITEMS & AMOUNTS
 260 PRINT AT 1,2;"ITEM";TAB 20;
"DOLLARS"
 270 PRINT OVER 1;AT 1,2;"_____";
AT 0,20;"________": PRINT
 300 FOR a=1 TO 7
 310 PRINT 'TAB 2;q$(a);TAB 22;r
$(a): NEXT a
 323 PRINT AT 16,19;"___________"
 325 PRINT AT 18,4;"Total.......
....   ";T
 330 PRINT AT 20,1;"EXPENSES of
the WIDGIT COMPANY"
 340 PRINT OVER 1;AT 20,1;"______
______________________________"
 350 REM ** LOWER HALF BORDER
 360 PLOT 0,175: DRAW 0,-175: DR
AW 255,0: DRAW 0,175
 400 STOP : COPY
 500 DATA "Salaries","40563","In
surance"," 4160","Office Supplie
s"," 4918","Rent","25001","Utili
ties","14987","Janitor Serv.","
8026","Depreciation"," 2898"
```



```
SECTORS are         identified
by the INITIALS of the ITEMS.

 ITEM                DOLLARS

 Salaries              40563

 Insurance              4160

 Office Supplies        4918

 Rent                  25001

 Utilities             14987

 Janitor Serv.          8026

 Depreciation           2898
                    ___________

   Total..........    100553

EXPENSES of the WIDGIT COMPANY
```

Bob Mitchell
20 Wild Briarway
Willowdale Ontario
M2J 2L2
900525

Hi Les,

Thanks for sending along the disassembly of PPProf and your other comments. I spent an hour or so last night fooling around with it all and finally gave up. It is darned difficult to follow machine code of that complexity. But I'll fill you in on what I did and what I have done today.

It seemed to me that somehow there has to be a change of OUT 7 as well as OUT 244 to get everything working. I got the banks to switch but lost my data (ie, sample columns) in the process. To do this I added some code at address 28930: (the orig code starts at 28950):

```
  LD A,240
  OUT (244),A
  LD A,71
  OUT (7),A
  RET
  NOP
  LD A,0
  OUT (244),A
  LD A,8
  OUT (7),A
  RET
```

Then, I put in some CALLs to the two addresses 28930 and 28940 at 39229, 39236, 39244 and 39259 (CD02710000 to the first and third and CD0C710000 to the second and fourth) to bypass the existing code but apart from messing up the contents of my first chip, nothing good seemed to happen.

If I could get this approach to work, it seems to me it would have to be duplicated in the other companion programs which all seems to be the hard way; there may be an easier one. So today, I decided to write to Larry Kenny and ask for his help. I may have to wait a while for this is planting time on his parent's farm and he spends a lot of time on the tractor.

I'll let you know of any developments. I am surprised that Stan Lemke did not follow up on this himself and complete the job for Larken Ramdisk owners.

The keyboard overlay is a great idea and it would be nice to have similar overlays for other complex programs. I resort to a 3x5 index box with all pertinent info handy within arm's reach for many programs such as Mscript, Zeus, Maxcom, Profile, Tasword, etc, etc. I did a rough job for HOT Z but it needs work. What we really need is a blank template cut out or with press outs that could be used for a variety of programs but I suppose that would be a special order for some print shop. Will have to stay with the exacto blades!

I had already done some of the fixes you mentioned but I adopted yours for line 5000 although I still use a GO SUB 950 which does

I decided to write to Larry Kenny and ask for his help. I may
hav

the same thing in other locations, eg, in line 9999 I have it start GO SUB VAL "950". Line 950 reads <INPUT "Drive #? ";drv: PRINT #4: GO TO drv: RETURN>.

Well, if you have any fresh thoughts about getting RAMDISK to work with PPProf, let me know; otherwise it's on the back of the stove for now.

You remember I was having CRC errors with my last four chips mounted on sockets on top of the first four? It seems to have been poor connections at the sockets (dirt??) and I finally gave them a really big bath of Zero Residue Cleaner (from Radio Shack) and the problem cleared up. For now.

That's about all for now and thanks again for the info.

Best regards,

Bob Bob

Amateur Programmer's Line

By Bill Harmer

Amateur programming with home computers has swept through its first phase, that of mass popularity, and is entering a more mature and perhaps level stage of its unfolding. Will it end up as a niche market like the hobby of radio building, which was done by the mass market when a radio was too expensive for anyone less than the quite rich to afford and dwindled off to a niche hobby of radio amateurs and experimenters, by the 1950's? Something the kids and students of a wide variety would still tackle then (like the old crystal sets) but which held the interests of only a few adults, after passing its peak of popularity in the 1920's. Has amateur programming become a niche in the market, a passing stage of school children but a hobby of only a few adult enthiusiasts?

The home computer has become a reality for perhaps 12% of the homes in America, and perhaps (a guess) two thirds of the homes have an adult using the computer. But probably it is safe to say that other than word-processing for personal use and for running work brought home from the office, only a small fraction of 1% of the home users that are adults do any programming with them and probably only5% of the students that have a home computer use it for hobby programming as opposed to school assignment programming. So we amateur programmers are rare birds indeed.

Now that we know who we are, let's look around and see who are our allies in the battle of the consumer to see the programming hobby supported. The schools and the school children/students are natural allies and indeed we profit considerably from the programming efforts of students, particularly at the university level, whose programs go on to become public domain. A crucial early BASIC compiler was done at a US Naval Academy, placed in the public domain and probably became the starting point for the development of MS BASIC compiler for MS DOS. (It's separate runtime module looks like the inspiration to say the least for the BASRUN module with early versions of MS BASIC for MS DOS). Many utilities, emulators/simulators were developed in the schools as assignments or for teaching. The Utah and Nevada compilers, Pascal, FORTRAN, COBOL, came from universities and offer a way to get into those languages at prices between $50 and $100 (compare MS FORTRAN and COBOL at $400-500 plus). The first microcomputer Pascal, UCSD Pascal using a unique speed code (p-code) which is interpreted during execution, to shrink the memory requirements of a complex language like Pascal to the capacities of the early micro-computers, is another case of university support. Now even the source code for such a Pascal compiler is available for anyone who wishes to port it to a home computer (assuming they have an IBM compatible and C compiler to develop it on). It appears that the Japanese data/disk file compression utility LHARC was done by a student, with the help of his professors. The examples could be paraded on and on.

The professional programmers or technical people who wish to work on a program on their own on their home computer and perhaps even sell it someday out of a post office box address are other allies. They may have seen some new program or idea in the course of their technical work and want to adapt it, redo it or simplify it for the home computer market. Their ads can be found in the classified ad sections of computer (and electronics) magazines. By the way, dBase III was born that way by a programmer from NASA, Wayne Routliff moonlighting to make a database for his early kit computer, and the various versions and improvements led via dBase II to Ashton-Tate's current cash cow. Now Wayne has left Ashton-Tate and has introduced his own database, by Emerald Bay, when he found work stultifying.

Last but not least in support of the amateur programmer are the user groups and the bulletin board systems (BBS) that allow the amateur to network and swap tips and techniques, sources and source code, and a lot of technical data in the form of disk files, that would be uneconomical to publish and distribute in most other forms. In fact, amateur programmers have submitted a whole realm of hard won technical knowledge, in the field called loosely 'reverse engineering' done in finding out how a commercial program or operating system works.

So if we can draw any conclusions they would probably include the point that even if amateur programming is a niche for enthusiasts, it has an outward looking appeal in that the amateur programmer's work is made to share. Whether the work was started to meet a course requirement, to help others or make some money on the side, good amateur programs are typically labours of love.

TS Bulletin

✲WARNING: Inquire First BEFORE You order, Buy or Send Money to any supplier, individual, etc.

BILL HARMER
97 Ruskin Avenue
Ottawa, Ontario
CANADA K1Y4B3

Mar+,1990 (2pp) TS Bulletin News Supplement #3-1990: Pub.by B. Harmer As A Hobby

This last month or so has seen the writer experiment with word processing using an MS DOS wordprocessor from Radio Shack at about $20 in price. A Brother M-1109 dot matrix printer, sold locally at C$169, provides hard copy. The only problem has been that you need a little BASIC program to get control codes to the printer to change the mode of printing,to small type for example, and eventually a program was written to write a file to disk with just the control codes required and then the DOS commands alone can be used to append text file to the control code file. This solves the problem of having the printer connected to a non-standard port no. and may help when used with other operating systems and disk systems than MS DOS, which is why it is mentioned here. Other work includes the preliminary evaluation of a fast and short m/c algorithm for compressing text/files. It seems to compress to about 82% of original size, and better figures are being sought by modifying it. The problems of text compression in a 16K-32K orphan computer require such novel approaches(a short m/c subroutine) since the standard algorithms like Huffman & Zempel-Lev require at least 3 or 4 K of text to make any saving and if the whole thing is locked in with a 2+3K BASIC routine to compress or expand the code, it is hard to make any saving in the little orphan, BASIC operating system computers like the ZX-81/TS1000, TS2068 or Commodore 64. CP/M computers don't suffer from this limitation however, and it is not to these more sophisticated computers (like the QL, and 16-bit computers) that this approach is addressed. It is hoped that the sub-routine code will fit within about 400 bytes! Anyone wishing to exchange tips and tricks and ideas on data compression for orphans (and bigger computers in some cases) may wish to write the publisher at the address above. Utilities like ARC, PAK and SQ use Zempel-Lev and Huffman codes. Somecomparative figures:- (based on straight, dense text files in ASCII;8½K) PKZIP,LHARC 44%, PKPAK,ARC 50%, SQ 57%. In some cases, files perhaps with a lot of spaces in them, ZIP will bring the file to as little as 19% or original size and ARC around 30%, SQ around 35%. LHARC by the way is the first program that the writer has seen from Japan.

The last Ottawa-Hull TSUG meeting included a long exchange of experiences in Pascal programming with Dave Solly doing some programming in HighSoft Pascal on the TS2068. A Pascal compiler for the TS2068 has been placed in the public domain for non-commercial exchange and is up on the club BBS at (613)745-8838, 300 baud, and maybe faster soon, (8/n/1), after normal business hours. Another way to get at Pascal on the TS2068 is to use a CP/M version if you have CP/M (with the Aerco DOS), and of course the QL can also use this with a CP/M software emulator.

....

The BOSTUG,TSUG has mutated, closing down as a general TS SIG within the Boston Computer Society and being reborn with many of the old members as the New England QL User Group, with a planned quarterly n/l & membership at US$10/yr., contact: Richard Taylor,309 Holly Cir.,Tiverton,RI,USA 02878 .to join or Joyce Blaho,971 Fellsway,Medford,MA,USA 02155 about n/l exchanges. The final n/l of BosTUG, mentions that Update Magazine (by Bill Jones, on TS Disk Systems, esp. Oliger), has been turned over to Frank Davis of ISTUG as Bill Jones is going to spend time sailing the Carribean with his wife and take a break from the computer publishing hobby. At US$18 for four issues around 50pp each, UPDATE has been a real buy for subscribers.

....

Donald Lambert of CRAGIST (Iowa n/l) reports that he has his ZX-81 LDOS disk system up and running with a Tandon disk drive and he has the fix for mating the two if you wish to contact him. He also is working with a sort of short board ZX-81 to use m/c programs developed on the ZX-81/TS1000 in a shortened controller board with Z-80 and SRAM/CMOS RAM; Mike LeDuc has developed the board design. Mr. Lambertmay be contacted at 3310 Clover Dr.S.W.,Cedar Rapids,IA,USA52404.

....

Mark Yost, P.O.B. 187, Canton, CT,USA 06019 sent us a notification that he may get info onthe TS1000/TS2068† program tape stock of an out-of-business store. The tapes, if available will go for about US$5.50 and include M-SCRIPT, Vu-Cal,Vu-File Timachine (BASIC compiler), Vu-3D,Aerco Print Master, Chess, 32/64 col,M-Term, Xadom, Pinball, Stock Market, Game Designer, EasySpeak, Frogger,80 col Vu-Cal, TasWordII,Lotto,Candy Factory,Blackjack,Star Trex, Art Works (some†Spectrum) ✲Ask before you order- this is an 'iffy' deal until specifics are tied down it seems!!

May/June 1990
May 1, 1990

Dear Out-of-Town Members,

Summertime is here. But our club members still seem to be busy with their computers. For the last three Mondays the mailman has brought 10 and 11 letters from members. A few of them were exchamge newsletters, but mostly they were members writing.

There's an interesting thing about our club. It is thriving. Our newsletter is thriving. This is in marked contrast to the experience of a number of other Timex clubs. For example today's mail brings word that the SINCUS newsletter will cease with the July issue. They are going to send all Timex articles to the SNUG group to be put in the SNUG newsletter. And further, they are opening up their club to MSDOS machines since, to quote..." Over a period of time more of our declining enrolment has been switching over to MSDOS machines."

The Boston T/S User Special Interest Group is closing up and re-forming as the New England Sinclair QL User Group.

The Harrisburg T/S group ceased publication of their newsletter several months ago.

The Seattle Area T/S Users Group newsletter contains only material provided by the Editor & Sub-Editors.

Even the CATS group, sponsors of last year's ComputerFest writes, to excerpt; "Back to the problem at hand, which is the lack of material being generated by the members. I hope this month is just an aberration rather than a trend. If it is the latter case then I'm going to recommend that we publish the newsletters every two months......I just know we can't keep going on this way."

I said earlier our newsletter is thriving. Take a look at this issue; if that's not thriving I don't know what it is. And our memberships stays up at about 70 members, (55 of them out of town). We could easily increase it but 70 is just about a many as I can handle. Some of you are having to wait for answers from me, as it is.

I have been getting letters from some members containing material which is too good not to share. Some of it is rather lengthy, but with the increased size of the newsletter, it will be easier to put it in without dominating it. Shall do!

Bill Jones (UPDATE magazine) says that Frank Davis is going to take over publishing UPDATE. Frank has been the mainspring of the Sinclair National User Group (SNUG) for some time; it is not clear, but it seems to me that Frank is running UPDATE as a separate operation, not associated with SNUG's newsletter called "SNUG ROUND-UP". With Frank Davis wrapped up in UPDATE, SNUG is going to lose an active member. Does not bode well for SNUG, I should say.

We had an executive meeting a couple of weeks ago. We faced an interesting dilemma; our finances are too healthy!! Since we have been meeting in a school classroom for the past several years, and have been getting a good deal on the newsletter printing, our bank account has been growing. A decision was made to purchase a bare QL for the club. Then, after some consideration the executive decided to give a 6-month membership extension to all who were members as of the meeting date of May 2, 1990. This would benefit all members equally, at the same time without tampering with the existing dues structure. Proposal approved by a quorum of members present at the May 2nd meeting. Note that this does not apply to newsletter subscriptions.

Note to any members who have borrowed the Larken Library disk #28, Pixel Print Professional. Bob Mitchell has been taking a look at it for me and notes an error in one of the programs. I shall identify it here. In the PP+ program, in LINE 2000 there are a couple of PRINT #4 statements together. One of them needs to be removed. Also in the same line there is a statement "LOAD "N$ CODE". It should be changed to read "LOAD N$ CODE 35926". Seems that the icons in the other library disk #10...PP+ will load in the wrong place otherwise.

Note with this program that it all goes on one Single-sided disk. There are no spare tracks, it is a full disk. The LOAD/SAVE routines in the program make the assumption that the program disk will be installed in drive 0, and the data generated by the programs will be directed to a second drive, drive #1. If you do not have two drives, 0 and 1, you should modify the LOAD/SAVE parts of the program to suit your particular set-up.

Last month I mentioned that I had several drives surplus to my needs, and offered them for $20. I have sent out 4 of them. Now I shall be able to buy some more!! No, No, slap my wrist!!

I am relieved a bit by word from Andrew Zettel, an out-of-town member, that he has his up and running. I say relieved, because it would be hard to troubleshoot a drive problem at a distance, as you can imagine. Two of the drives were sold to an in-town member, so that was no problem. Well, there was no problem anyway, but you know how these things can go.

I still have three drives, if anyone is still interested. (I went and bought still another one!)

I should also like to mention that a couple of members have come to my rescue. Bob Mitchell has picked up a printwheel for my Smith Corona L1000; and another member, John Austin, has sent me another three. So I'll be able to print many more articles for the newsletter; and also this letter!! Don't buy any more for me; I have enough to last to the year 2000, I think!

I have another disk ready for the Larken library. It is a disk holding the interbank data base programs developed by Larry Crawford. It also goes into details about bank-switching using the Larken RAMdisk. It is on a DS 80 tracks per side disk. If you do not have a Quad drive, mention that to me and I shall put it onto two DSDD disks. I caution you that it will interest the more serious TS2068 users, but it is a bit heavy for rest of us. Well, if you have a need for a large database file, this is just the thing. There is quite a bit of documentation on the disk, so you are coached on it's use pretty well. It's just that I have not had the time to get into it very deeply, and it's still somewhat of a mystery to me. It will be known as Disk #30 Interbank Database.

Did I mention in the last n/l that I have the RMG Enterprises catalogue on library disk #29. It is available as either DSDD or QD format. Ask for it.

I was into a appliance repair shop, and inquired about used UV bulbs. They are the ones used in clothes driers many years ago to sanitize clothing as it dried. They make good EPROM erasers. But he wanted $4 for the bulb, and a further $4 if you wanted the socket. The socket holds both the bulb and a necessary 40 watt bulb that is used in conjunction with it. Since I already had one, I was not interested in paying his price. But if any one is interested in getting one, I can pick it up for you.

I thought I had written an article about building an EPROM eraser for the newsletter. but when I searched for it, I found that none had been published. Maybe that would make a good article. I'll think about it.

I have just written a good Larken disk header reader. It uses a 178-byte m/c routine to do the task, and it is very fast. To put a screen display of each file name on the disk, its starting address, and length you simply have to do a RAND USR call. I have written a Basic routine to enhance it. This allows you to print the output to screen, 2040, or large printer. Also to select the drive, and to print in regular or compressed mode. Other header readers do the same thing, but I like this one best. Ask for it if you are interested. I am going to put it onto the TTSUC Larken library disk #1.

Some time ago I came across a tape that a member inadvertently sent me in place of one of the club tapes. I did not realize it when it came in; it was only later when I came across it. I then connected it with a member's earlier letter apologising for the late return of a tape. Seems he thought he had returned it, but no. The tape I have in my hand is labelled " Inaugural Florida Unity Breakfast, Holy Family Church, North Miami January 1990". Would the member please drop me a line; I don't recall who it was!!

Larry Crawford has sent me a correction for his article on the Larken RAMdisk, that was published in the last newsletter. It came in too late to put in this issue, but we shall publish it in the next newsletter. But I shall put a copy of it in this letter to put anyone who has tried the article, out of their misery.

May 20th....Jeff Taylor has just phoned and told me that his issue is going to have 34 pages to it. I said to him, I hope that you've saved some material for the next n/l. But he said, we'd better hope that members continue to send in material. So here's hoping....do come through, July and August are slow months!

May 24th....Jeff brought the newsletter over to me today. Well, as you can see by looking at it, Jeff has rather overdone it this day. But don't you like it!! Trouble is, this issue now can't be folded so I have to go out and get some large envelopes and more postage to mail this issue. I'll have to speak to him to keep the n/l under 100 grams, say, or we have to pay another 40 cents to mail it! But anyway, it does look good, doesn't it? I think we outdid ourselves this time!

I had to make up a new set of large envelopes for this issue, including a new set of address labels. If you check the expiry date on your label this time you will see that it reflects the 6-month extension to your subscription. I'd like to clarify something. When I read Jeff's column it gave the impression that the subscription rate was for 18-month periods, rather than the current 12 month. Not so. We are simply giving a one-shot 6-month extension to members.

I shall have to put the following item in the next newsletter, but I shall make brief mention of it right now. I received a letter from a former member, David Solly. He mentions that he and Larry Kenny have a BBS which has been upgraded to run at 1200 baud. It is called "Sir Clive's Castle", and can be reached by dialling (613) 745-8838. It is, of course, dedicated to the Sinclair product. The BBS software is Larry Kenny's MAXCOMM. David says, in his letter, that if you are looking for some specific programs you should download two files on the BBS which lists their holding. 1) a VuFile data file under "Libr.Cv" and 2) an Mscript compatible file under "UD.CT". Feel free to download them, David says.

I've run out of things to say, so I shall put in a short ZX81 routine. We seem to be short of ZX81 stuff this month; maybe this will make up for it a bit.

```
***************************************************
******************* CORRECTION ********************
********** OTHER USES FOR LARKEN RAMDISK **********
**************** ARTICLE MAR-APR '90 **************
***************************************************
```

Most humble apologies to anyone who tried to create a BASIC AROS by following my instructions. The following corrections will remove some of the frustration: 1. The LOAD statements in two of the sections of the article will need to have RANDOMIZE USR 100 added. 2. In the "CREATING AN AROS" section: line 30 should have FOR X= 32768 not 62768. The instruction in para 6 should read: OUT 244, 240: RANDOMIZE USR 100: LOAD "test.Cl" CODE 32776: OUT 244,0. Once again, sorry!! Larry C

```
***************************************************
```

George Chambers

Faster slow mode

G W Hewitt,
Edinburgh.

ZX-81

IF, LIKE ME, you like to type in programs in slow mode, you will have found that it can be annoying when adding lines to a long program to have to wait while the computer writes the entire screen every time. It is especially bad if the line is a long Print or Rem line.

This small program can be temporarily stored near the start of the program using a few spare lines. When one has a screen full of lines, type Goto (first program line). Answer the prompt with the last line you have entered and it will return you to that line with an empty screen below. It also Pokes the listing-system variable to ensure that every relist after that will return to your chosen line.

```
2 CLS
3 PRINT "TYPE LINE NO"
4 INPUT XXX
5 POKE 16419, XXX - INT(XXX/256)*256
6 POKE 16420, INT (XXX/256)
7 LIST XXX
8 STOP
```