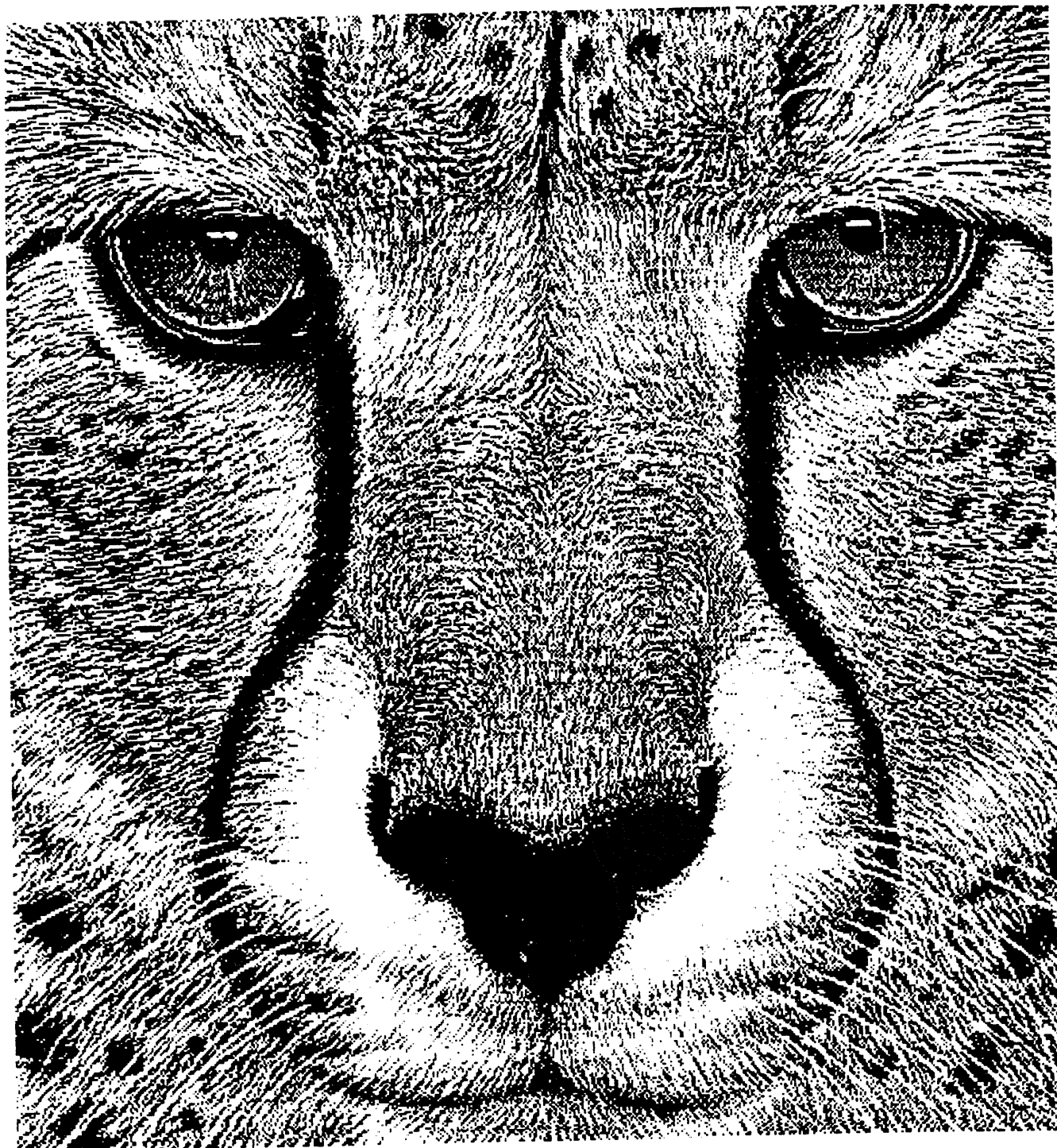


SINC - LINK

JAN-FEB '91 VOL.9 NO.1



SINC-LINK VOL.9 NO.1

JAN-FEB '91

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

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

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

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

SEND CORRESPONDANCE TO:

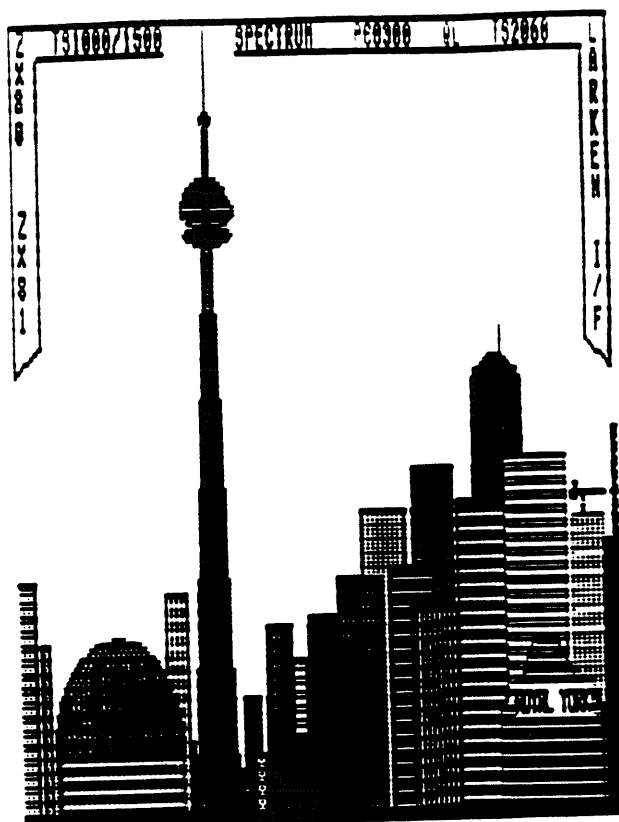
Attention: SINC-LINK
TORONTO TIMEX-SINCLAIR
CLUB, 14 RICHOME
SCARBOROUGH, ONTARIO,
CANADA M1K 2Y1

EDITOR
USERS
COURT,

EXECUTIVE OFFICERS:

PRESIDENT:
TREASURER:
SECRETARY:
ACTIVITIES:
TAPE LIBRARIAN ZX81:
TAPE LIBRARIAN 2068:
TAPE/DISK LIBRARIAN QL:
PAPER LIBRARIAN:
NEWSLETTER:
LIAISON OFFICER:
(Out-of-town members)

(Area Code 416)
RENE BRUNEAU (531-9749)
BILL LAWSON (444-8772)
GEORGE CHAMBERS (751-7559)
RENE BRUNEAU (531-9749)
LYMAN PAQUETTE (482-4479)
RENATO ZANNESE (635-6536)
HUGH HOWIE (634-4929)
JEFF TAYLOR (244-8583)
JEFF TAYLOR (244-8583)
GEORGE CHAMBERS, 14 RICHOME COURT.
SCARBOROUGH, ONTARIO. M1K 2Y1
(416-751-7559)



TORONTO TIMEX-SINCLAIR
USERS CLUB

TORONTO TIMEX-SINCLAIR USERS CLUB

Editorial

Well I thought it was time to change the cover again. What do you think? This cover was done on one sheet of paper using Byte Power's "The Print Factory" for the 2068. The super-sized screen is from their "Companion" series of screens and fonts. Actually the Cheetah is six screens butted together. You load the first two screens into the "The Press", print 24 lines, load the next two screens, print those 24 lines, then load the last two screens and print the last 21 lines. Once you learn how to do it, it's really quite simple. I just wish the documentation gave a little more info - you use up a lot of printer paper experimenting. Are you listening, Kris?

Desktop publishing seems to be the latest rage around the club with members honing their skills on "Pixel Print Professional" or the "Wordmaster" suite or "Print Factory", all for the 2068/Spectrum or "text87" and "Page Designer II" for the QL. Have you had any experience with these or any other DTP programs? Let us know. It can only help to produce a more professional looking newsletter.

Hello Editors

As the T/S world shrinks, newsletter editors may be finding it harder and harder to put out quality products. If your club or newsletter is in danger of folding, don't give up the ship! Let Sinc-Link be your platform. We will publish pertinent stuff on your group for free and you can even keep using your newsletter name within our paper. How's that for an offer!

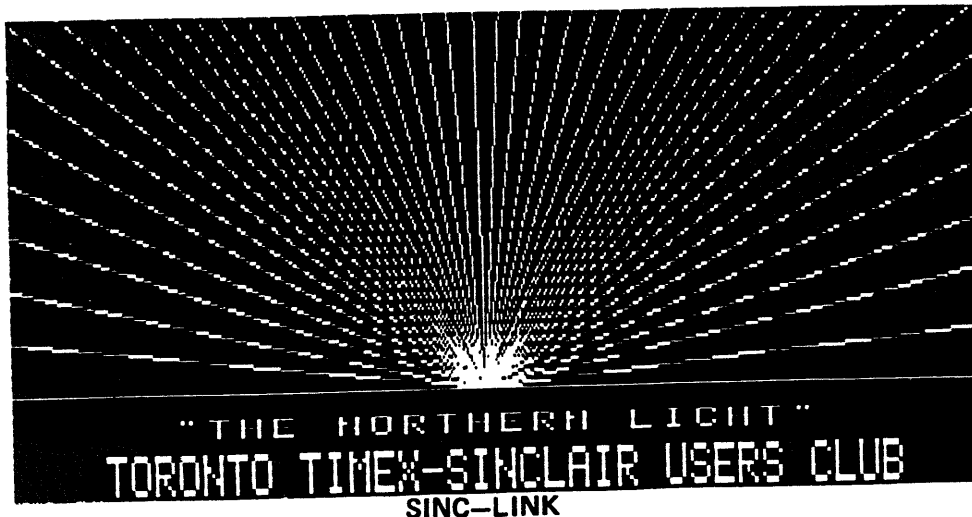
Get well, Ian!

As this issue is going to print, Ian Robertson, a long, long time member is in hospital undergoing tests and treatment. For those of you who are familiar with Ian, you'll know him as a staunch supporter of Sinclair. From the early to late eighties, his "Sincbits" articles were a constant source of new information on both publications and products for our computers. He must have every known product ever made for T/S and Sinclair computers.

Ian, our thoughts are with you.

That's all for now...

J.T.



BOB'S NOTEBOOK ODDS & ENDS

DISK NAME GARBAGE

Have you ever used disk doctor to look at the Disk Name area on track (block) zero of any of your disks? If you have or if you do, you might notice that there is a lot of nonsense after the proper disk name that ought not to be there.

Try this: Load doctor.B1 from the club omnibus disk or other copy. Then put a disk in an appropriate drive and examine track 0. Press <d> to examine the directory name area. The current name will end with a zero. After that there should be zeroes. More than likely there will be a lot of left over "crud" that the Larken system never removes, not even on using the format routine! If you're a nut for tidying up things, try this: BREAK into the BASIC and put in a line (I numbered it 1805) as follows ("NOT " is the token NOT on the <s> key): <IF i\$="NOT "THEN INPUT;:PRINT #0:AT 0.0;"Standby...":FOR i=54500 TO 55119:POKE i,0:NEXT i:GO TO 1650>. Restart "doctor" and get to the point where you're looking at the Disk Name area which starts at byte 4500 and goes up to byte 5119 or so; these bytes being added to the start of buffer 50000 make the addresses 54500 to 55119. Now press the token NOT on the <s> key and you'll see the lower screen (menu) disappear and be replaced by <Standby...>. The <INPUT;> wipes out the lower screen, which is a good trick often useful in many other applications. In about 20 seconds, the display will re-appear; now save the track in the usual way, making sure that no write protect tab is on. If you now examine the disk name area, it will be full of zeroes. You will have to re-enter the current disk name.

Now the source of all this trouble appears to lie with the original Hcode.C1 provided with format.B1 by Larry Kenny. I'm not one hundred percent sure of this, but my original copy on the Larken-provided disk was full of garbage in the disk name area. If you want to clear Hcode.C1, you will have to examine it from about 41000 to 41179 or so; do a similar POKE operation to input zeroes and re-save the code (2000 bytes long).

It would be possible to replace the FOR...NEXT loop with a code LDIR routine, but once the Hcode is corrected, the need to do any more clean up jobs should die out.

CHECK PRINTER READY

In some of my program or disk documentation and in previous columns, I have mentioned that the IN 127 value of 237 represents the reading I get with my Fastext 80 printer when it is ready to print. There's more to this: the problem really lies with the particular TS2068 machine for all are not alike. While most apparently give the 237 result, other do not.

PRINTING A POUND SIGN

I recently ran into a situation where I needed to print a POUND

sign (that's on the <x> key) on the wide printer because that key had to be pressed to make a certain program run properly. I found that wherever the POUND sign was to appear in the text, I had to type a <#> sign.

Then, by proper commands to the printer, it could be made to switch character sets to the British one. Each time it encountered a <#> symbol, it printed the POUND sign. The way to do it probably varies, but with my printer, I did the following:

```
PRINT #4: POKE 16093,32: REM Larken double poke command
PRINT #4: POKE 16098,3
LPRINT CHR$27;"R";CHR$2: REM check your printer manual for
correct commands here
PRINT #4: POKE 16093,0
PRINT #4: POKE 16098,0
PRINT #4: POKE 16094,m: REM needed because POKE 16093 also
changed 16094 (where m is the required margin setting).
The printer will now be conditioned for the British character
set.
```

BUG ALERT! LK2CPY.B1 (Richard Hurd's Program).

This is a very useful utility for making a duplicate copy of a disk between two drives having the same number of sides and tracks. But, if the original disk is FULL, the utility will give a false CRC error as it does not know enough to STOP when all tracks have been copied. The following lines will correct the problem:

```
ADD <2065 LET blx=sid*numtrks>
ADD <635 IF a=blx THEN GO TO 740>
```

I have only confirmed this OK using two DSDDs. But I have just installed a 3.5 inch 720k drive and will eventually check this out again.

BUG ALERT! INDEX.B1

ADD <102 RANDOMIZE USR 100: CLOSE #3> to ensure that the TS2040 will function when called upon by the disk indexer program.

Bob Mitchell 901209

TORONTO TIMEX-SINCLAIR USERS CLUB

During November a meeting of the QL wing of TTSUC was held at H. Howie residence in Burlington.
A number of most interesting demonstrations were held.

CHAOS BUSTER

NETWORKING

MODIFYING AND CLEANING UP ARCHIVE FILES

text87

IMAGIX

CHAOS BUSTER

Louis Laferriere gave a demonstration of CHAOS BUSTER which can be obtained from Wood and Wind Computing (Bill Cable). The total package is a user friendly database running within ARCHIVE . The user doesn't need to know ARCHIVE language to get the most benefit from this software . HIGHLY RECOMMENDED

NETWORKING

HUGH had three machines on a network with the following equipment
1 = The main machine with a colour monitor , expanded memory , TK2, Cumana disk interface and large printer.
2 = The second machine with a monochrome monitor , Trumpcard (896 K) and two 3.5 " disk drives.
3 = The third machine was using a TV set and microdrives.
The demonstration was very impressive with diskdrives under the command of any of the machines (QL) , the monitor screen listing the directory of a disk being copied etc.
The full advantages of NETWORKING and the flexibility of the peripherals was most impressive.

MODIFYING AND CLEANING UP ARCHIVE FILES

The next demonstration was the use of IMPORT and EXPORT commands to modify and clean up ARCHIVE files. It is quite a different experience reading about this process in the manuals or from different publications.

text87

We had all heard about the great advantages of text87. But until you see it in actual operation it is difficult to imagine the complexity and capabilities of this modern word processor. Somebody referred to it as the equivalent to LOTUS 123 .

IMAGIX

The evening was then completed with a complete demonstration of IMAGIX from Emmanuel Verbeeck from BELGIUM. The programmes are most impressives and educational. --

CHANGING TASWORD II HELP PAGE

Dick F. Wagner

Periodically questions come up on problems with using the Tasword II word processor. These have been addressed in the past when the program was "new". A recent problem from one of our members was that the Expanded mode didn't work on the large printer. As it turned out, the codes for expanded mode were 14 for ON and 15 for OFF. BUT, these codes are for 1 line printing which is OK for headings. This means that the code must be on the line to be expanded (double width) and not on the line above. A more universal code is 27,87,1 for on and 27,87,0 for off but the OFF code must be added at the end of the expanded line.

This Tasword II program had been reworked in the past (a used edition) and the main "HELP" page had been revised to some extent. Now the owner would like to put the page back to its original format, but how to do it? Here are the steps for the original Tasword II program--

1. LOAD TASWORD II check CAT for form).
2. After the working page is displayed, use STOP to select BASIC.
3. Edit line 15 by adding after the LOAD"-----"CODE the address and number of bytes, 54784,10751. Check this in line 710 for correctness.
4. Rewind the tape if cassette. RUN the program with this revision. Use STOP for the menu and select "load text file" (j). On "name", input "tasword".
5. Edit the "HELP" page as normal text, moving the cursor and making the changes desired. One suggestion is to add "CS" (CAP SHIFT) above the left set of graphics. Also change the printer to the model in use, besides any mode names. Remember, the modes must correspond to the codes and graphic symbols assigned on the printer codes screen. If you get lost, just use the EDIT key to bring up the main "HELP" page.

6. Use a fresh tape/disk and use "STOP" to return to the menu. Now choose "save text file" (s). Again name it "tasword".

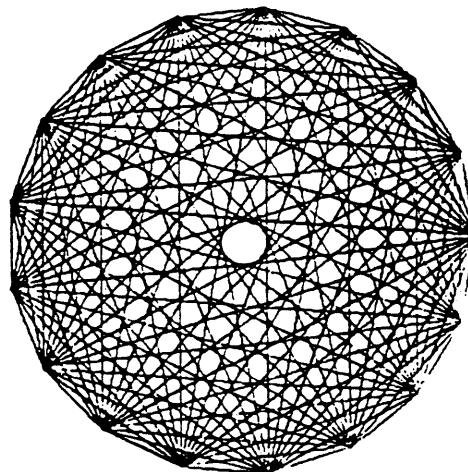
7. This time go into BASIC with "b" and GO TO 15 and reload the new text file

8. The revised "HELP" page should come up on EDIT. Check it out to see if all of the revisions are as you want them. If OK then LIST 15 and delete the added address and bytes. Now make a final save for the finished copy.

This all seems rather complicated but it falls into place if followed exactly.

Readers of Time Designs will find another method in VOL. 3, NO. 2, pages 26-29. The actual program is lines 4000-5080. Bill Ferrebee has made a major change in Tasword II that includes this subroutine. I checked his program against the original and marked each line that was different. Then I made the necessary revisions/additions. His method is of changing the "HELP" page is "self contained" but is slow. This is the reason for the flashing "WORKING" sign, just wait until it stops. After all, it takes time to make 1537 loops. Actually it is faster than the method given at first.

Note: Thanks to Norm Lehfeldt in TIMELINEZ of unknown date for the procedure for revising the "HELP" page.



from The Plotter n/1 of Clackamas County

GEOMETRIC DESIGN

DATA-READ-RESTORE ON A TS1000
By Sebastien Boisvert

This is a DATA/READ/RESTORE program that appeared in the February '87 issue of ZX Computing that was written by David Nowotnik in 1983. I have modified it slightly to make it shorter and a little easier to use. It will let you make DATA statements and READ them like other computers do. Therefore, you will be able to adapt those programs for "bigger" computers on the TS1000.

The first thing you got to do is make a 1 REM statement with at least 107 characters in it. Then enter POKE 16510,0 to change line 1 to 0. Then enter the program from listing 1 and enter the code from listing 2. When you are done, enter the program from listing 3 and SAVE it to tape; that is your master copy.

Using the program is very simple. When a RESTORE instruction is required, replace it by a RAND USR REST. DATA statements are stored in REM lines immediately followed by the gray square on the A key. For strings you don't need to put quotes before and after the text. You must make sure though that the computer READs it as a string. This is done by using the form of:LET (string variable)=Z\$ (TO USR READ). This would be the equivalent of:READ (string variable). The way to READ numbers is:LET (variable)= VAL Z\$ (TO USR READ). If you run out of DATA in your program, error 0/ will occur.

Beware of strings that contain commas in your DATA statements. Suppose that you want to store the sentence "THE TS1000 IS A COMPUTER, THEREFORE IT COMPUTES". Since there is a comma, the computer will think there are two strings and will READ them as:
THE TS1000 IS A COMPUTER
THEREFORE IT COMPUTES

Also beware that there are as many characters in the DIM Z\$ statement as there are in the longest DATA string. You can also use variables other than Z\$ as long as it is the first defined variable in the program. If you have non-DATA REM lines that start with a gray square, just put a space between the REM and the square and the computer won't read it as a DATA line.

This program will let you COPY the number of lines you wish to be copied:

```
1 REM 12345
10 POKE 16510,0
20 POKE 16514,22
30 POKE 16515,NUMBER OF LINES
40 POKE 16516,195
50 POKE 16517,107
60 POKE 16518,8
```

RUN and delete lines 10-60.
Use RAND USR 16514 to use... This routine is fully relocatable.

This program will let you COPY a specific line from the screen:

```
1 REM ...19 spaces...
10 INPUT L Line to be copied (1-24): 1=top line
20 LET L=(L-1)*33
30 POKE 16519,L-256*INT (L/256)
40 POKE 16520,INT (L/256)
50 RAND USR 16514
100 FOR F=16514 TO 16532
110 INPUT A
120 POKE F,A
130 NEXT F
```

Enter RUN 100 and enter the following values:

42:12:64:35:1:0:0:9:1:32:0:17:60:64:237:176:195:83:8

Delete lines 100-130... This routine is fully relocatable.





by Chuck Kere Luck

PIX-FX

One sentence can basically sum up this program:

I haven't had this much fun with a piece of software since Pixel Print!

PIX-FX, written by Michael DiRienzo for the TS2068 computer, thoroughly answers the question of, "Now that I have some graphics on file, what the heck do I do with them?"

Plenty! Do you have something that would look better smaller? Larger?

You can do it! How about just a piece of a graphic? Want to enlarge Catherine Deneuve's eyes to full-screen? Or just one of them?

You can do that, too! How about making your own customized set of graphic eyeballs to replace Cathy's?

Do it! Here's a neat idea: You know that digitized group picture we ran of myself, Robert, Malcolm and Dave?

Well, with PIX-FX, you can "cut out" each face and wrap it around a sphere!

That's right: Make planets out of 'em! Then choose the tilt of each planet and which side you want to look at! After creating this new "solar system", you can pretend they've suddenly encountered a Black Hole and stretch 'em out in 3-D, with one end much smaller than the other!

In fact, just about the only thing that will limit what you can do with PIX-FX is your imagination.

Sounds like a tall order for TS software? Must be a catch, huh? Like being very difficult to operate, weird USR calls or something unfriendly as that. Nope. Nothing like that. Not only is it easy, it's fun!

However, there are a couple of things the user needs to know beforehand to insure a smooth transition into easyness.

One, is that PIX-FX works by employing two different screens.

These screens are displayed to you simultaneously, by the way, or you

can choose to view them separately.

They are the Working Screen, which you initially LOAD your graphic into and the Final Screen, which you usually use to store your revised graphic. Keeping to this convention will save you lots of confusion.

Also, there is a Load/Move Pic screen that is always displayed right after choosing a function from the Main Menu. This screen shows both the Working and Final screens and what's in which. If you have a graphic LOADED into the "WS", it will show it in a mini-version of your regular CRT screen. If something is in the "FS", it will show that, too, in the same fashion. Image reproduction in these 'mini-screens', by the way, is excellent.

Anyway, here is where you make sure that your original picture is in the Working Screen. If a complex picture is in the works, requiring several mods, a lot of screen-swapping from WS to FS will be done. And since every main-menu-selectable function will step-stone you back into this same Load/Move screen, you must never forget what it was you were doing in the first place!

One item I thought odd was the apparent omission of a way to clear any screen. This seemed to be a real problem, since the program comes with a graphic already in the Final Screen.

Well, you could just copy the already-blank Working Screen into the Final, but even that wasn't necessary.

It turns out that for LOADING purposes, it can be done anywhere and if there is something taking up the space, it won't be for long!

Another item to keep in mind is that when executing a function from the Main Menu, PIX-FX will always display the Working Screen after you exit from being popped into the Load/Move Pic menu screen (The WS, is, after all, what you're working from) but will always display the Final Screen immediately after executing the function or after exiting the chosen Load/Move option to view both screens simultaneously. You can, of course, pick the VIEW SCREENS option in the SCREENS STORAGE menu that you picked from the Main Menu.

So, by keeping all the aforementioned tips in mind, you

will greatly ease your transition into the area where imagination finally kicks in begins some serious graphic work. You'll find some of that work elsewhere in this issue.

One thing noticable, by now, is a variety of menus! There are four of them: one Main and three subs. The only sub-menu not mentioned, yet, is the HELP Menu.

The HELP Menu directs you to everything you always wanted to know about PIX-FX. Just choose the subject. After picking one, the user may expect to get popped into another text screen or something. No way!

That would be too boring and conventional. Instead, while you're still looking at your Final Screen, the chosen Help text begins marching a single line across the bottom of the screen, from right to left! This is real fun to watch. The characters are large enough to read from across the room, so you don't have to drag out your reading glasses. Pressing ENTER will terminate the Help text so you can continue creating.

I don't know about you, but I happen to like menu-driven software. It's friendlier and keeps you from digging into that documentation, again. Plus, the menus are like pants: They pull down! The selections are made by pressing a 'down' or 'up' key (6 or 7) and watch a hi-lighted bar roll through the choices. When it stops at something you want, just press ENTER. What could be simpler??

While a pull-down menu can engage the Help files, it's often handy to have the hardcopy available, as well.

In this regard, Mr. DiRienzo has done a very good job of documenting everything with clear explanations.

There are things covered here that you won't find in the Help Menu, such as customization. Since PIX-FX comes on high-quality cassette tape, general disk converting instructions are given, plus address areas to insert full-sized printer driver codes. Very helpful!

For those of you, like a lot of people, who hate instructions, no matter how easy they are to read, he has supplied a PIX-FX Tour. You must, however, read 1 page of Tour instructions. For the rest of us, we can content ourselves with the

detail supplied in just ten pages.

One of the most fun features of PIX-FX is called Planet Maker. This is the previously-mentioned utility that allows you to wrap a SCREENS around a 3-D sphere. Since the graphic must be stretched a bit to accomodate the globe's surface, varying results occur with different graphics. Digitized photos of people's faces are most interesting.

Planet quality depends mostly upon, uh, the face in the photo! See the examples, then guess who!

A real nice technical feature involves the SAVEing of a graphic file once it's created. An option is given to SAVE in compressed format! This has the advantage of saving up to 50% of space compared to the normal 6912-byte SCREENS. This can be a real boon to disk users. You can store more pictures on a single floppy. Tape users will find it helpful, also, since they don't have to wait so long. There's a USR-accessible machine code routine at the head of each compressed SCREENS to pop it back to normal size at the normal address area.

Unfortunately, a real lemon feature is that you can't LOAD them back into PIX-FX to their original form! Maybe an updated version from VectorWare will fix this? In the meantime, one may wish to add a "RANDOMIZE USR 40246" statement in the BASIC to expand the compressed file after LOADING.

Actually, this was the only 'minus' I could find in the whole program.

Once you start using PIX-FX, this becomes negligible, because all the 'plusses' more than eliminates the single 'minus'.

If you are at all interested in graphics and own a TS2068, I strongly suggest you try out this excellent program. For your own copy, send a mere \$20.00 to VectorWare, 4128-1/2 California Ave. SW, Seattle, WA 98116 or call (206)935-9272.

Tell 'em you read about it in SWYM!

Putting the SUPER into BASIC.

I am one of those people who cannot see a bit of code, in SuperBASIC at any rate, without wanting to improve it. This happened most recently when I looked at a few lines by Butch Weinberg called Pri_dir in the Sept/Oct issue of SincLink. It struck me that it might make the starting point for a series of articles on how I go about writing a SuperBASIC program. The program as it stands works, despite an error, and does a useful job, but has lots of scope for improvement; it makes an excellent nucleus for a really useful program. The original listing was:

```
10 OPEN#3,ser1
20 DIR#3,flp1_
30 PRINT#3
40 PRINT#0,'More?'
50 more$=INKEY$(-1)
60 IF more$='y' OR more$='Y'
  THEN GO TO 20: ELSE
70 PRINT#3,CHR$(12);: CLOSE#3: STOP
```

There were three things that I wanted to alter right away, since while they worked they were hang overs from lesser BASICs, and didn't take full advantage of the Super variety. To avoid confusion the BASIC statements I am discussing are given in bold type, to distinguish them from the discussion itself.

1. The condition **more\$=="Y"** tests both cases at once, and is more efficient than **more\$="Y" OR more\$="y"**.

2. In this instance it isn't really necessary to assign the value of **INKEY\$** to an intermediate variable, you can use it just as it comes, so the two lines 50 and 60 can be replaced by:

```
IF INKEY$(-1)=="Y": ....
```

This waits until you press a key before evaluating the IF.

3. The use of **GOTO**. **GOTO** and **GOSUB** are quite unnecessary in SuperBASIC, and are generally considered "bad form" - even though they work! In this instance the **GOTO** is creating a loop, and this is better done using the **REPEAT ... END REPEAT** structure. The way to break out of a loop is via an **EXIT** statement, and in order to use this we actually have to reverse the IF statement I derived above i.e. we want to stay in the loop if the answer is "Y". There are several ways to deal with this: we could change the question to "Finished?", we could change the line to **IF INKEY\$(-1)=="N"** or do what I have done stick a **NOT** in front of the condition. This may seem the most complicated solution - but it is one that is generally available and it may come in useful when there is no alternative. Note that it gets around the absence of a not-equal test that covers both cases at once; **k\$<>"Y"** DOES NOT WORK! The error is the floating **ELSE** at the end of line 60; it serves no purpose at all.

Incorporating these ideas the program becomes:

```
10 OPEN#3,ser1
15 REPEAT loop
20 DIR#3,flp1_
30 PRINT#3
40 PRINT#0,'More?'
50 IF NOT INKEY$(-1)=="y":EXIT loop
60 END REPEAT loop
70 PRINT#3,CHR$(12);: CLOSE#3: STOP
```

Structured Programming - Clase's Rules

But I am still unsatisfied; it is not properly structured. Most programs can be divided into three parts: **set up** - in which the initial conditions are determined (windows, devices etc), **main** - where the main action takes place, and **finish** - which tidies up at the end and restores your QL to its normal state. (As you may have noticed many commercial programs leave out the finish section!) Accordingly my own programs generally consist of a single line:

```
100 Set_up: Main: Finish: STOP
(the STOP is optional.) The rest is all PROCedures and FuNctions. In these I try to keep to the following self imposed rules:

```

1. Each PROCedure and FuNction should be no longer than one screenful (i.e. 20 lines). Anything longer than this almost certainly involves more than one procedure.
2. PROC/FN names have an initial capital and variable names are all lower case. This distinguishes them from each other and SuperBASIC commands and machine code extensions.
3. Global variables - those few required to be used generally throughout the program - should have meaningful names, but wherever possible variables should be LOCAL to their PROC/FN. LOCAL variables can be abbreviated to two or three letters, since rule 1. means that their role should be easy to discern. As well as variables defined as LOCAL, this includes formal variable names from the DEFINITION line. Each PROC/FN should be as general and self-contained as possible making it easy to reuse in other programs.
4. I also like to have a clear demarcation between PROC/FNs, and put their names in a prominent place to make it easy to find them in the listing.

Although I have largely evolved these rules myself they are similar to the way 'C' programmers work and Mike Lloyd in QL World recommends something similar.

Rewriting our program according to Clase's rules we have:

```
100 nm$ = "Prt_dir1"
110 REMark hjc 1990.11.03 Ver 0.1
120 Set_up: Main: Finish: STOP
```



```

130 REMark ~~~~~
140 DEFINE PROCEDURE Set_up
150   OPEN#3,ser1
160   dev$="flp1_"
170 END DEFINE
180 REMark ~~~~~
190 DEFINE PROCEDURE Main
200   LOCAL lp
210   REPEAT lp
220     DIR#3,dev$
230     PRINT#0,"More?"
240     IF NOT INKEY$(-1)=="y":EXIT lp
250     PRINT#3
260   END REPEAT lp
270 END DEFINE
280 REMark ~~~~~
290 DEFINE PROCEDURE Finish
300   PRINT#3,CHR$(12): CLOSE#3
310 END DEFINE

```

Great! We now have 22 lines where we had 7 doing exactly the same job! But to me it is a working skeleton that we can flesh out into a much more versatile program. What we have to do is to look at each statement in turn, and where it can be improved replace it with a call to a PROC/FN that incorporates the improvement. The basic structure we have above will remain, although the statements will be altered and many more PROC/FNs added.

Actually I have already altered it slightly by introducing the global variable dev\$. This is assigned the value 'flp1_' in Set_up, and it is the first statement I want to expand. There is also a programming variable nm\$ which contains the name and current version of the program to make it easy to save the various versions as I go along. I use an ALTKEY definition available from toolkit II, but if you don't have it you can add a PROCEDURE such as the following (adapted to your own favorite device):

```

DEF PROC Sv
Delete 'mdv1_&nm$: SAVE 'mdv1_&nm$
END DEF

```

Enter sv at the keyboard to save the current version. This can be removed from the final version of the program.

Returning to dev\$, to change the device using the original you would have to EDIT the code. Wouldn't it be nice if the program gave you a choice of device, or even better offered you a default, but gave you the option to change it? We should also include code so that only legitimate device names are accepted. This calls for a FuNction rather than a PROCEDURE, since we want it to come up with a device name or "return a string value" in computerese.

From now on I shall not reprint the entire program each time, but just the lines that have to be altered and the additional PROC/FNs.

For obvious reasons I've decided to call the FuNction Get_Dev\$ (it has to end in \$ since it returns a string.) My original idea was to insert a call to Get_dev\$ at line 160, but this fixes the device for the duration of the program, putting it into the Main loop means that you can get directories from different devices in the same run. I've therefore put it in line 210, as part of the DIR statement. This means that each time this line is reached you are asked the name of the device. This works because Get_Dev\$ is a string FuNction and can be used anywhere that a statement expects a string. Get_Dev\$ expects one parameter, the name of the default device, which is defined in line 160. So change line 210 to:

```

210   DIR#3,Get_Dev$(dev$)

```

and add these two FuNctions to the end of your listing:

```

320 REMark ~~~~~
330 DEFINE FuNction Get_Dev$(df$)
340   LOCAL dv$,k$,j$,lp
350   CLS: PRINT"Directory of ";df$;" ?"
360   PRINT "Enter, or type new name."
370   REPEAT lp
380     k$=INKEY$(-1)
390     IF k$=CHR$(10)
400       dv$=df$
410     ELSE
420       PRINT k$;: INPUT j$: dv$=k$&j$
430     END IF
440     IF True_dev(dv$): EXIT lp
450     BEEP 3000,25: REMark bad name!
460   END REPEAT lp
470   RETURN dv$: END DEFINE
500 REMark ~~~~~
510 DEFINE FuNction True_dev(d$)
520   LOCAL l$,n$,p$,ok$: ok%=0
530   p$="mdv/flp/fdk/ram/hdk"
535   n$="12345678"
540   l$=LEN(d$)
550   IF l%=4: d$=d$&"_": l%=5
560   IF l%=5
570     IF d$(1 TO 3)INSTR(p$)
580       IF d$(4)INSTR n$ AND d$(5)="_"
590         ok%=1
600   END IF: END IF: END IF
610   RETURN ok$: END DEFINE

```

One parameter, dev\$, is passed to the FuNction Get_dev\$, QDOS requires brackets for a FuNction, but not for a PROCEDURE! This is received by the FuNction as df\$ (the default); df\$ is used as an alias for dev\$ within the function. The advantage of this is that you could use the same FuNction for programs using more than one device e.g. a file copying program. NOTE any changes made to df\$ within the FuNction will actually happen to dev\$ - although nothing can happen to it here.

Get_Dev\$ first prints the default (350) and then examines the next character typed (k\$). If this is

2

<ENTER> then dv\$ is given the name of the default (390,400) otherwise k\$ becomes the first letter of the name, and the rest is added to it and assigned to dv\$ (420). Either way you can only get out of the loop (lp) if dv\$ is a proper device name (440). If it fails the test then you get a raspberry and another go (450,460). Finally dv\$ is the value RETURNed by the FuNction (470).

True_dev is another FuNction which checks the device name and returns the value 1 (=true) or 0 (=false). The condition which follows an IF is always evaluated by QDOS as true or false in this way. (Try PRINT 2+2=4 and PRINT 2+2=5 directly from the keyboard.) This function first adds the trailing underscore if only four characters are present (550). Then, only if the name has five characters, it checks that the first three form an acceptable device name, that the device number is in the range 1-8, and that the underscore is present by a sequence of long IFs (570-610). You cannot do it all in one go, linking the conditions with ANDs because of a bug in SuperBASIC: it should stop checking as soon as it finds a false condition, but it actually goes on to check them all. So, for example, if d\$ had only two characters an error would be found when it tried to check the characters beyond 2. Putting them in a cascade avoids this problem, each condition is only evaluated if the previous ones are true.

That is more than I intended for one episode. Next time we will look at how to save paper. The program as written lists the names in a single column down the left side of the paper, it would be much more efficient if we could use three or four columns. It would also be nice to have the option of putting them into alphabetical order. You might like to think about how to do this - hint: it involves OPEN_NEWing a temporary file.

This month's assignment is to see if you can modify line 210 so that the default is whatever device the program used last, rather than always reverting to the original one from line 160. It only requires a small change!

In the meantime I'd be happy to hear from anyone with comments, suggestions or requests for more information.

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

©1990 H.J.Clase

email: hclase@kean.ucs.mun.ca



POWER GLITCHES

by Hugh H Howie

At the November meeting I tried to give a demo and it was a total flop.

A number of times we have had problems with QL crashes and corrupted MDV's at those meetings, so this time I took along some disks, including one drive which I purchased on my way to the meeting, hoping that everything would be OK. Boy, was I ever wrong.

Set up the units, and could not get a DIR, or a WSTAT. One disk gave a DIR but would not load, just 'Not Found'. Even the new disk drive was no good. Eventually gave up in disgust.

Got home and spent a while connecting everything up on my own system, and everything was in good order except for the disks which were corrupted at the meeting. Even the new drive was good.

By the way, these drives are new 3 1/2 NEC (uncased) and sell for \$39.95, with a 30 warranty. Lovely little jobs, and quiet as can be. Sold by one of the surplus stores. So take a look at those stores in your own area. I now have two, and they are good. Other members also have purchased them. But I am off the topic.

I wrote to Dan Elliott to see what he had to say, and his reply was that there was possibly a power dip in the supply. Now this is quite possible as our meetings are held in a Collegiate, where the custodians are using large floor polishers in the evenings, and I wonder if this is the source of our problems.

The answer was to install a Coleco Adam Power Supply. This would cost \$50 plus P&P from Dan.

So there you are. If you are having strange crashes and corruption, then it may be the power supply and not the QL. I don't know if I will have the job done as I have no problems at home. Just at the club. 888

FOR SALE! FOR SALE! FOR SALE!
1 A & J Micro Drive \$75
w/15 cartridges in original box.
1 TS2050 Modem like new \$45
in original box
1 TS2040 Printer, in original box \$25

Assorted software for 2068 and 1000 @ \$2 ea.
New, in boxes (send for list of titles)

Willing to trade, also. Looking for Larken
RAMdisk or a 512K RAMpack for the QL.

Write to:
Rick Burt, RR #5 Gr. Box 157
Belleville, Ont., CANADA K8N 4Z5

WANTED! WANTED! WANTED!

Command cartridges for the TS2068
If you have some, drop a line to:
Frank Capell, 5428 West Schubert
Chicago, IL 60639
Or phone 312 745 1817
(no collect calls please!)

HELP! HELP!

One of our members, Phillip Joe, is looking for a couple of books. If anyone can give leads on where to locate them, he would appreciate it. Maybe someone knows a book source in England where he could write, for example.

The books are:
SPECTRUM WARGAMING by Owen Bishop and Audrey Bishop, Pub. Collins
POLITICAL AND MILITARY SIMULATION GAMES by Mike Rose, Pub. Interface

Phillip Joe, 800 Bowie Lane, Miss. 38930

Or you can drop me a line, and I can forward it to Phillip.GFC.

UTILITY TO PRINT OUT THE CATALOG OF A LARKEN DISK

5 REM This program is based on work by a another Programmer. This version is by Thomas Skapinski 7 Atkinson LN, Coram, NY 11727-3004 U.S.A. 10/07/90

```
10 CLS : RANDOMIZE USR 100: OPEN #4,"DD": REM
opens Larken Disk Drive
15 OUT 127,15: REM SPACE
20 RANDOMIZE USR 100: OPEN #2,"lp": REM send i
nformation to printer instead of screen
25 OUT 127,27: OUT 127,51: OUT 127,15: OUT 127
,15: OUT 127,27: OUT 127,83: OUT 127,48:
REM SMALLPRINT
30 INPUT "DRIVE? (0-3) ";DRV: RANDOMIZE USR 10
0: GO TO DRV
35 INPUT AT 1,0;("DRIVE # ";DRV;" SELECTED""D
ISK Nm/# "); LINE A$: INPUT "DATE (mm/dd/yy) ?"
; LINE b$
40 RANDOMIZE USR 100: CAT "",
50 PRINT "DISK ";a$,"DATE ";b$""
55 RANDOMIZE USR 100: CLOSE #2
60 INPUT "1= MORE 0=MENU ";m
65 IF m THEN CLS : GO TO 15
70 RANDOMIZE USR 100: NEW : REM
starts a program called Autostart in my case
a LLoader program
75 STOP
100 RANDOMIZE USR 100: SAVE "CATCP2.BL" LINE 1:
REM disk save self start at LINE 1
```

SAMPLE OF OUTPUT

```
Disk Name :
catcpv.B1 001 quivlb.B1 001 reader.B1 002
header.B1 001 diskac.Cx 001 AUTOSTART 002
FORMAT.BS 010 Hcode.C1 001 format.B1 001
FORMAT.BL 001 Header.BA 001 Header.CO 001
FORNT+.BL 001 CATCP2.BL 001 LIST.BL 001
```

```
LARKEN LKDOS 1985
Track/Side 040/002
Total Files 015
Free Blocks 053
```

```
DISK UTILITY DATE 10/07/90
```

NOTE: As you can see this program will produce a miniature list of the catalog... This mini list can be taped on to the Disk or to the jacket of included with the disk in it's sleeve.

I hope you like the program. You may wish to use this printer set up for other purposes of your own creation. If you come up with anything you would like to share with others please do so. And send me a copy too.

Keep computing!

```

5 PRINT AT 10,5;">>>>TINY PRINT<<<<"
6 PRINT AT 19,0;" ---PRESS any KEY to CONTINUE"
7 PAUSE 0
8 CLS : PRINT AT 5,0;" RETURN to MENU ? (Y/N)": INPUT a$: I
F a$="Y" THEN GO TO 95
10 REM by Thomas Skapinski 7 Atkinson LA
Coram NY 11727-3004 U.S.A. 11/11/90
20 REM For LARKEN Disk System
30 REM Turn printer OFF then ON to resume normal print size
40 RANDOMIZE USR 100: OPEN #3,"lp": REM opens channel for larg
e printer
50 REM
60 OUT 127,27: OUT 127,51: OUT 127,15: OUT 127,15: OUT 127,27:
OUT 127,83: OUT 127,48: REM superscript and modified line spac
ng****AERCO COMPATABLE PRINTER INTERFACE REQUIRED****
70 CLS : PRINT AT 18,1;"Now evrything will print out tiny.
If you use an EPSON compatible printer"
71 PRINT FLASH 1;"***PRINTER MUST BE TURNED ON***": FLASH 0
72 PAUSE 350: CLS
90 STOP
95 RANDOMIZE USR 100: NEW
110 RANDOMIZE USR 100: SAVE "TINYpr.B2" LINE 1

5 PRINT AT 10,5;">>>>TINY PRINT<<<<"
6 PRINT AT 19,0;" ---PRESS any KEY to CONTINUE"
7 PAUSE 0
8 CLS : PRINT AT 5,0;" RETURN to MENU ? (Y/N)": INPUT a$: I
F a$="Y" THEN GO TO 95
10 REM by Thomas Skapinski 7 Atkinson LA
Coram NY 11727-3004 U.S.A. 11/11/90
20 REM For LARKEN Disk System
30 REM Turn printer OFF then ON to resume normal print size
40 RANDOMIZE USR 100: OPEN #3,"lp": REM opens channel for larg
e printer
50 REM
60 OUT 127,27: OUT 127,51: OUT 127,15: OUT 127,15: OUT 127,27:
OUT 127,83: OUT 127,48: REM superscript and modified line spac
ng****AERCO COMPATABLE PRINTER INTERFACE REQUIRED****
70 CLS : PRINT AT 18,1;"Now evrything will print out tiny.
If you use an EPSON compatible printer"
71 PRINT FLASH 1;"***PRINTER MUST BE TURNED ON***": FLASH 0
72 PAUSE 350: CLS
90 STOP
95 RANDOMIZE USR 100: NEW
110 RANDOMIZE USR 100: SAVE "TINYpr.B2" LINE 1

```

Note by Geo Chambers

Tom does not mention which printer he is using. Printers vary in the codes they require to perform print functions. You may need to check your printer manual, if Tom's listing does not work for you.

Also, sometimes the computer puts out the printer codes faster than the printer can act upon them. The GOSUB and LOOP at line 1000 cause the program to wait until the printer indicates it is ready. You may need to check the "printer ready" response of your printer. Use a loop such as " 10 PRINT OUT 127: PAUSE 0:GOTO 10". With you printer turned on and ready to go, the number appearing onscreen is the number to put into LINE 1000. In my case it was 108.

```

50 REM
60 OUT 127,15: GO SUB 1000: 0
UT 127,27: GO SUB 1000: OUT 127,
65: GO SUB 1000: OUT 127,8: GO S
UB 1000: REM
This worked on an SCM FASTEXT 80
Printer ****AERCO COMPATIBLE
PRINTER INTERFACE REQUIRED****

1000 IF IN 127<>108 THEN GO TO
1000
1010 RETURN

```

text⁸⁷
/ (LATEST)
MY / IMPRESSION

by Hugh Howie

--- WHY I Z Z I T ---

Whyizzit that the mere fact of writing about something clears a whole lot of cobwebs away? Whyizzit when you talk to a friend about your latest problem it never seems to be quite the same, never as bad as you at first thought? Even if your friend had nothing to say to help you, he at least listened and that was all that was needed to help come to terms with your problem.

In the Nov/Dec issue of Sink_Link I gave my review of text⁸⁷, and after I sent it in, I realised that a lot of my misgivings about this programme had vanished. I progressed so much faster after writing about it.

I was speaking to a number of QL'ers recently, and the subject of text⁸⁷ came up, and also of what I said about it, and I had to admit in retrospect that my review was somewhat lacking in enthusiasm. At the time when these conversations were being conversed, (hur-umph) I had already forgotten most of my problems. I had progressed to greater things within the programme, and had improved my knowledge considerably.

Take for example where you have a long document to print out, and you would like to do it on both sides of the sheet of paper. text⁸⁷ has no problem with this, a couple of key presses and off you go. I had to print out a document about 50 pages long, so all I did was set text⁸⁷ to do the odd number pages, flip the paper over when they were finished, and do the even numbers. Easy as pie. Had a coffee while I watched the magic at work.

You wish to print out a given page? No problem. You don't like the type used on the printed page, OK so change it, just like that.

I have been reading a lot of reviews of this program as they appeared in various

publications, and in each case the learning period was in the region of a couple of months depending on the intellectual powers of the user, I guess that is why my own learning period was a bit longer than norm. But then text⁸⁷ is so much more powerful than many other word processors.

Will I go back to Quill? I don't think so, I see no reason why I should. The versatility is so great I could not hope to cover all the advantages in this short note. At one time I would, and did, say it was only for someone who did a lot of writing, but after taking some documentation from the larger commercial programmes, importing it to text⁸⁷, and finding it came across with the merest adjustment to the ruler I was using, and getting an excellent hard copy, I have changed my opinion. Quill or ASCII files are treated the same. You can import just about anything into it, and it will be accepted; and you do NOT have to make an export file to do it.

Oh! Did I mention that CTRL-C puts you back into S/Basic, and the same command returns you to text⁸⁷?

I Love It!

888

FURTHER TO THE ABOVE.

After I wrote the above, I realised I had a few lines left on the page, so I wrote this under another title, and merged it with the original. So handy.

One thing I failed to mention, I think, was that in text⁸⁷, you are not restricted to eight characters in a file. And what is more, I think you can use just about anything on the keyboard in the title. Now does that not make it easier to keep track of your documents?

I have even saved a file by the title of **/?_&% and loaded it back in. Now try that in Quill!

You are not happy how it looks after you change the format or type? OK to just REFORMAT the document. Hey man I'm in a rut, I just like to use it. Gotta go. NOW.

HH

888

A LETTER FROM ONE OF OUR MEMBERS
Re: Multiplication programs from last month

One of our members, Steven Gunhouse, has written me a letter which, in part, is a follow-up to two mathematics routines I put in the last Sept/Oct newsletter issue. Excuse my red face!! (I have retyped it because of faint printing. GFC.)

".....I note several typos and some other errors in your "Multiplication Accuracy" program. Line 70 should be N instead of M, line 90 should be "LET B = VAL X\$(M)", line 110 should read "LET I = INT (A(C)/10)". Those are probably all typos, but there should be an extra line for the printing that reads:

```
185 IF A(1)<>0 THEN PRINT A(1);
```

Otherwise, the first digit of something like $99*99$ won't be printed. One other typo is the lack of a semicolon in line 200. With all of those corrections, the program should now be accurate, if somewhat slow.

Of course there is no reason why it has to be quite so slow. Since you have an available accuracy of 8 digits, that means you could multiply two 4-digit numbers with complete accuracy, and doing four digits at a time could take 1/16th of the time. Then again, the output becomes more difficult.

As a mathematician, I did write such a routine, including the correct output, about 2 years ago. I will include it with this letter. Oh, and formally, it is not really accuracy but precision. The computer has about 9 digits of precision; saying it has 9 digits of accuracy requires knowing that your original numbers were accurate. I know, a minor detail, but my grade 9 science teacher required us to distinguish between the two. (Generally, math teachers presume all numbers to be accurate, but science doesn't have that luxury.)

About the Russian Peasant Multiplication; in reality, that is almost the same way that a computer would multiply. In other words, it is equivalent to the algorithm for binary multiplication. The only reason it is slow is because it's in BASIC. If you compare speeds for the Russian Peasant program as opposed to the Multiplication Accuracy program, there probably isn't that much difference. I am not sure how much of a speed the string VALs and the output routine make.

I am glad to see all of the charting stuff as well. Of course, if you are making a chart on the screen, color might also be nice as well as pattern. With the examples used, this would not be difficult to implement.

I looked over Bob's article as well, but since I only have a V2 (LKDOS EPROM), it doesn't matter much. I'm surprised about his remark concerning LPRINT CHR\$ 12, as I don't need an extra LPRINT. Is this something specific to V3, or his interface, or his printer? I don't recall anything like that back when I used a Fastext 80, so I tend to believe it is one of the first two. I also note that my current printer has two extra pitches - semicondensed and semicondensed elongated. I'm surprised he didn't test out other features, such as italics, emphasis, or underline. I suppose his main point is that the margin and line length settings would change for each size change. Actually he didn't bother to change the line lengths.

Well, that is enough rambling for now. Though perhaps I will throw in another fractal for good measure. This one is known as the dragon spiral. Most fractals of dimension between 1 and 2 are made by drawing lines....."

```

5 REM Multiplication Precision in 4 digit blocks
7 PRINT AT 1,3;"Precision Multiplication"
8 PRINT TAB 6;"by Steven Gunhouse"
10 INPUT "First Number?"a$
15 LET a$=" "+a$
16 REM the extra spaces are necessary in case there would be < 4 digits in the first block
20 INPUT "Second Number?"b$
25 LET b$=" "+b$
30 LET a=INT (LEN a$/4)
35 LET b=INT (LEN b$/4)
40 DIM a(a)
45 DIM b(b)
50 FOR i=LEN a$ TO 4 STEP -4
55 LET a(INT (i/4))=VAL a$(i-3 TO i)
60 NEXT i
70 FOR i=LEN b$ TO 4 STEP -4
75 LET b(INT (i/4))=VAL b$(i-3 TO i)
80 NEXT i
90 DIM c(a+b)
100 FOR i=a TO 1 STEP -1
105 LET t1=0
110 FOR j=b TO 1 STEP -1
115 LET c(i+j)=t1+c(i+j)+a(i)*b(j)
120 LET t1=INT (c(i+j)/1e4)
125 LET c(i+j)=c(i+j)-t1*1e4
130 NEXT j
140 LET c(i)=t1
150 NEXT i
200 PRINT a$(4 TO );"x";b$(4 TO );"="
205 LET i=1
210 IF c(1)=0 THEN LET i=2
215 PRINT c(i);
220 FOR j=i+1 TO a+b
225 LET t$=STR$ c(j)
230 PRINT ("000"+t$)(LEN t$ TO 4);
235 REM Must add extra zeros if t$ is too small (<1000)
240 NEXT j
250 PRINT "Done."
990 INPUT "Another Number? ";y$
: IF y$="Y" OR y$="y" THEN GO TO 1
999 STOP
9999 RANDOMIZE USR 100: SAVE "mu
it.B4" LINE 1

```



THE SINCUS LIGHT PEN - TS2068
An Experimental Project
by Wes Brzozowski

Here's a nice little project you can whip up and get running in a couple of hours, or you can choose to spend as long as you want, refining and upgrading it to suit your particular demands. This light pen plugs into the joystick port, and is so simple that you'll spend more time putting together the joystick connector than you will wiring things up!

Before describing the light pen, we need some background. Your TV set works by sending a bright spot flying across the screen, drawing individual lines, eventually making a complete picture. Because our eyes tend to "see" the brightness of the spot after it has been moved elsewhere, we perceive a picture. But if our eyes had a faster response, all we'd see would be a moving dot, and TV displays as we know them would be worthless.

Fortunately, our eyes are a bit slower than that, but the light pen has the ability to detect the spot as it streaks by, and by synchronizing our program to the video display, we can measure the time it takes for the spot to take to go from the top of the picture to the light pen. This tells us the vertical position of the pen, and a sample program, included here, demonstrates this (those who want to learn more about synchronizing a program to the video should read *Computer Interruptus*, part 7, *Sincus* news Jul/Aug 1986).

Finding the horizontal position is a bit more difficult. The D'Ktronics light pen for the Spectrum does this by putting a large white rectangle on the the proper vertical position, and collapsing it until it knows exactly where the light pen is. The demonstrator program does not do this, but I'm sure some of you readers will be glad to expand it. Since this will involve putting white shapes directly into the display file, you may first want to read "Navigating through the display file jungle", (*Sincus* News, May/June 1986).

If you are starting to suspect that there's some reasoning behind the order in which I publish my articles here, you're absolutely right! An experimental project like this ties arcane concepts together into something practical. It also encourages us to work together on projects; an idea that becomes increasingly important as vendor support fades.

In any case, either of the two light pen circuits in Figure 1 send a low signal to the joystick "FIRE button" pin when the spot is in front of the light pen (and the address lines have been diddled properly) and a high signal, otherwise. The fact there are two light pen schematics will accomodate the many different types of phototransistors available.

What's a phototransistor? Basically, it's a transistor that switches on when light is shone on it. For the technical crowd, all transistors essentially start out as phototransistors. Photons of the

proper wavelength can kick electrons in the base region from the valence band into the conduction band, where they'll support conduction, turning the transistor on. The fact that most transistors are sealed in a metal can or black epoxy hides this fact. Phototransistors may have design tweaks to optimise their sensitivity, but we can think of them as ordinary transistors in a clear epoxy package.

Because we often only turn the transistor on with light, some transistors only have two wires, the collector and the emitter. Others have the third "base" lead connected so that the transistor can also be biased into it's linear region, if desired. But you'll find no biases here, Bucko! If you've that third lead, you can tape over it, or cut it off, or whatever, 'cause you don't need it.

For ordinary silicon, the energy difference between the valence and conduction bands is equal to the energy of a photon in the infra-red region of the spectrum, so phototransistors are usually most sensitive to this, as well. It's too bad, because the glass in your picture tube filters most of the stuff out, letting only the visible light pass. This is why there are two light pen schematics.

The Type 1 schematic is what you'll probably want if you're lucky enough to scrounge up a phototransistor that's optimized for visible light. The Type 2 schematic is for infra-red transistors. It adds a transistor to increase the pen's overall sensitivity, and it does it almost too well. With this type circuit, ordinary room lighting affects my prototype in an unusual way. The miniscule amount of light that leaks through the plastic pen case causes false readings. I had to cover the whole pen with black electrical tape to keep the thing from going bonkers.

The pen should be wide enough that the transistor fits all the way down inside. If it's too far away from the surface of the picture tube, it may not reliably see the "spot". Again, the infra-red coming from the picture tube is fairly dim. Even if you have to tape over the pen to keep the light out, the picture tube, which seems brighter, really isn't. Since we can't see in the infra-red band, this isn't at all obvious to us.

It'll take a bit of work to make the connector to plug into the joystick port. It turns out the connector is recessed too far into the 2068 for a standard 9-pin D connector to fit. The plastic insert inside the D connector will fit fine, but it'll go in so far that there'll be nothing left to grip to pull it out. Figure 2 shows a reasonable way to solve this problem. Make sure your wiring is correct before you apply the epoxy; there'll be no second chance. Incidentally, you can easily make a "mold" for the epoxy extension out of cellophane tape. Just wrap it around the insert, and make sure it's on tight with no holes before you pour the epoxy in.

When the pen is finished, plug it into the left port and type in Listing 1. When

running, the computer should beep like crazy, whenever the light pen is pointed at a lamp or the sun. You may get it to beep a bit by holding the light pen against the TV screen, but BASIC is too slow to do this properly; and it seems to have some tendency to synchronize improperly to the video, making only some small portion of the screen usable. To use the light pen on the screen, use Listing 2.

The program will move an asterisk in the left hand column to whatever vertical position the pen is at, on the screen. It will also plot pixels at the proper vertical position, and print the pen's distance from the top of the screen. The small machine code portion uses the HALT instruction to synchronize us to the video, and then updates a counter until the light pen sees the spot. This is a measure of the pen's distance from the top of the screen, and is returned to the BASIC program.

Line 60 of the BASIC program converts this to a pixel distance; by subtracting 180, we remove the approximate number of counts that occurred in the top border area and above, before the spot got into the active part of the screen. By dividing by 5, we can account for the fact that the counter is updated about 5 times per video line. Neither of these numbers is immutable. I worked them out by trial and error, so feel free to change them a bit. By the way, when we add .5 and take the INT, we really just round down to the nearest integer. This part should not be changed.

You may have to adjust the brightness and contrast on your set to get decent results. If your set is real old and the picture tube is shot, you may find portions of the tube where the light pen will not work reliably. There's not much we can do about this.

By the way, if you let the light pen go off the screen border area, or point it away from the screen altogether, you'll get a false reading for a moment. Commercial light pens usually include a switch to tell the computer the pen is at a valid location. You may wish to include something like this, or change the code so that you press a key when you want a reading taken.

Lastly, note that you'll stay looping in the machine code if the pen never sees any light, so if you want to BREAK out of the program, make sure the pen is pointed at the screen or you'll be ignored.

I've found this project to be quite fun. I'd be interested in hearing of your improvements or problems, and help out in whatever way I can. Radio Shack does carry an infra-red phototransistor; if anyone tries it, let us know how it does.

The simplicity of the thing should make it particularly nice for club projects. Once we show the software guys which end of the soldering iron to hold, they'll be glad to go out and whip up some dandy code to put this gadget through it's paces. Until then, it's a nifty project for all you tinkerers, out there. Enjoy!

From the Sept '86 issue of Sincus News

Listing 1. To debug Light Pen

```
10 REM    Light Pen Tester
20 BRIGHT 1: CLS
30 IF STICK (2,1)=1 THEN BEE
P .04,1
40 GO TO 30
```

Listing 2. Light Pen Demo

```
2 BORDER 0
5 BRIGHT 1: CLS
10 DATA 62,7,211,245,62,191,21
1,246,62,14,211,245
20 DATA 1,0,0,118
30 DATA 62,255,219,246,3,254,1
27,32,247,201
40 CLEAR 32767
45 LET oldrow=0
50 FOR j=32768 TO 32793: READ
k: POKE j,k: NEXT j
60 LET line=INT (.5+(USR 32768
-180)/5)
65 IF line>175 THEN LET line=
1
66 IF line<0 THEN LET line=0
70 LET row=INT (line/8)
80 PRINT AT oldrow,0;" ": PRIN
T AT row,0;"*"
82 PRINT AT 10,10;line;"
"
83 OVER 1: PLOT 50,175-line: O
VER 0
85 LET oldrow=row
90 GO TO 60
```

Light Pen Schematics

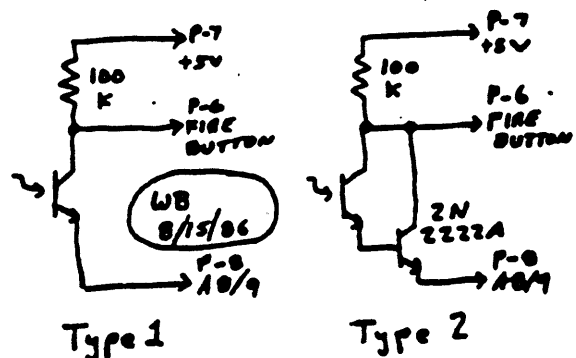
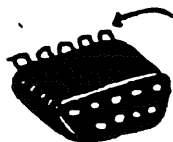


FIGURE 1.

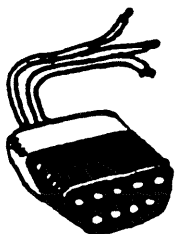
Both types of light pen!
Which will work best depends
on the type of phototransistor
you have to work with



Start with a 9 pin female D-Connector. To purists, this is a DE-9S. Cut off the outer metal shell.



The plastic insert will fit nicely into the joystick port. Solder wires onto the back, and coat and extend the rear panel with epoxy.



The finished connectors are long enough that you can grip it to remove it and has the wires firmly attached, so that they won't break off.

-W.B. 8/15/86-

FIGURE 2.

Making A Joystick Connector

NOW! AFFORDABLE 2068/1520 HI-RES PLOTTING

WHAT IS A 1520? - The 1520 is a HI-RES four color printer/plotter made for Commodore computers capable of printing rotatable characters in 4 different sizes and able to plot lines (solid or dashed) within a 480 by 1,996 step (0.2 mm/step) plotting area. Replacement pens & paper are low priced & can be found at most RADIO SHACK stores.

HOW AFFORDABLE? - The COMMODORE serial port emulator hardware I/F for the TS2068 sells for \$14.95 (barebd), \$20.95 (compl. kit), & \$30.95 (assem./tested). All orders are shipped postpaid.

SOFTWARE? - Currently, there are utilities for interactive and auto transfer of SCREENs to the 1520, for making banners & calendars, plotting demos incl. fractals, & a patch kit allowing CHSript V5/5.2 files to be printed and/or plotted on the 1520. The software is priced @ \$8.95 (post paid) each and comes with complete user notes.

USER FRIENDLY? YES!! - All print/plot directives are sent via BASIC "LPRINT" statements. PLOT & DRAW commands in existing BASIC programs can easily be replaced with corresponding "LPRINT" plotter equivalents.

====> See TIME DESIGNS 4/2 & S.W.N. 5/4 for reviews. <====
Send LSASE for more info./order form and a sample plot to:

♦♦♦ John McMichael ♦♦♦
♦♦♦ 1710 Palmer Drive ♦♦♦
♦♦♦ Laramie, WY 82070 ♦♦♦

The Digitizer and A Problem Found Notes From Marty

By now there are several of you playing around with the digitizer board. Some of you have had great pictures created and some are getting a black scan after 3 to 7 passes. myself was getting a black scan just by bringing my hand NEAR the board. All is not lost. I have found the problem on the board. Somehow when the boards were made a foil run was omitted & that's causing this problem. Please refer to the schematic diagram and note that pins 1 & 20 are BOTH tied to the 5 volt line. On the board a jumper on the component side was omitted. This can easily be corrected. Hold the board in your hand with the component side away from you and refer to the layout diagram. IC6 is the IC in the lower right hand corner & pins 1 & 20 are the pins at the bottom. Add a bridge of solder from the other bottom pin to the foil run connecting the other bottom pin and you are all set. If you have any doubts about doing this feel free to return the board to me @ SMUG, P.O. Box 101, Butler 53007, and I will correct it for you.

I still do not know why these boards worked at all. With that pin floating in space the 8 flip-flops in the 74HCT273 could clear whenever they felt like it. Well I guess I'll see some of you at the Expo. Right now I am going to get started on trying out John McMichael's "VIDEOTEX" program using the OKIMATE 20 color printer to colorize some of these pictures I've digitized. Marty Miller.

UPDATE COMPUTER SYSTEMS
P. O. Box 1095
Peru, Indiana 46970

This month's program is a simple version of the UNIX utility `wc` (word count). This program reads in a text file and counts all of the lines, words, and characters.

The structure of this program is similar to last month's program. The section that gets the file is the same. The while statement is the same. It's the block in the while statement that has changed. One line is different. C has a way of assigning the same value to a number of variables. The line `nl=nw=nc=0;` sets all of the variables to zero. This way you do not have to write three individual lines to set each variable to zero.

Items in single quotes are not taken as literals, but translated to ASCII. `'\n'` stands for newline and `'\t'` stands for tab. The C manual shows about 6 more of these.

Note that the define statements are commented out. In the `stdio.h` file, YES and NO are already defined. In other implementations they may not be.

Define statements are used to create one version of a constant. With a constant defined with a define statement, anytime the compiler comes across the constant it is replaced with the actual number.

In the `printf` statement the `%d` stands for a decimal. C knows to put a decimal value at this point. The first decimal variable it finds in the statement goes in the first `%d`. The C manual defines others besides `%d`.

I'll let the reader muddle through how the program works.

```

/* Program 6 */
#include <stdio.h>

/* #define YES 1
   #define NO 0
   The two constants are defined in stdio.h for this
   implementation, but may not be for others
*/

main () {
    int c, fd, nl, nw, nc, word;
    char file[20];

    printf("Enter File: ");
    gets(file);
    fd = fopen(file,"r");
    if (fd == NULL) {
        printf("File Open Error");
        abort(1);
    }
    word = NO;
    nl = nw = nc = 0;

    while ((c = getc(fd)) != EOF) {
        ++nc;
        if (c == '\n')
            ++nl;
        if (c == ' ' || c == '\n' || c == '\t')
            word = NO;
        else if (word == NO) {
            word = YES;
            ++nw;
        }
    }
    fclose(fd);
    printf("Lines: %d Words: %d Chars: %d",nl,nw,nc);
}

```

Happy New Year, everyone! Was Santa good to you this year?

I thought that I would start off this year with a new format. What I propose to do on this sheet is to print information concerning all of the computers in our club that has come to my attention, through other club members or the media. Hardware projects will continue to be presented, specifically for the ZX81, but compatible for the 2068 where possible. When available, projects for the QL will also be printed.

I have collected many articles for the ZX81 over the past 5 or 6 years and I am now collating them into binders. I hope to put an index together and will give it to anyone who is interested. Many of the articles are hardware orientated.

Remember Thomas Woods? He used to be the publisher of SyncWare News, an excellent T/S journal that ceased publication a couple of years ago. He has written an article in The January 1990 edition of PC HANDS ON called "Parallel Printer Interfaces". It is an in-depth review of how parallel interfaces work. Programming examples are given for the IBM PC, but could easily be adapted for a ZX81 or 2068 with a parallel port. At the end of the article, Woods suggests that you can control a variety of gadgets in addition to a printer, and indeed that is so, as illustrated in several recent hardware projects to be found in POPULAR ELECTRONICS.

Incidentally, check your local library for back issues of magazines such as RADIO ELECTRONICS, POPULAR ELECTRONICS (formerly known as HANDS ON ELECTRONICS), and MODERN ELECTRONICS. These are all published in the States and available at most magazine shops. International magazines are ELECTRONICS

TODAY, EVERYDAY ELECTRONICS, PRACTICAL ELECTRONICS, and ELEKTOR and these can be found or ordered through specialty magazine shops. Prices range from \$3.00 for local magazines to \$6.00-\$10.00 for the international ones. Many of the projects can be adapted or used as resource material for ZX81 or 2068 use.

AS the prices of PC 'AT's drop, more and more of the older, slower IBM type 'XT's are appearing on the market at greatly reduced prices. This means that many T/S users have been and will be shifting over to that machine. For the diehard T/S owner this is considered blasphemous and has generated quite an uproar in the past. Witness the letters and calls to the previously mentioned SinWare News when the editors suggested including PC information. Many of the newsletters received by our club are indicating that they will be folding or changing over to PC format.

Now that it is winter and humidity has dropped, a word of warning about STATIC ELECTRICITY. If you notice that you are getting a lot of shocks from static electricity, you should ground yourself before turning on your computer. By simply touching an object that you know is grounded to the house electrical system you discharge the static electricity built up in your body and avoid zapping your equipment. It is possible to develop several thousand volts in your body which is quite capable of destroying the chips in your computer. In a past newsletter, Hugh Howie, our QL librarian, described how he placed a metallic strip along the edge of his desk and connected it to the center screw of a wall outlet. A portable mat could be constructed using cardboard and aluminum foil bonded to it. To provide a safe connection to Ground, use a 1 meg (brown, black, green) 1/4 or 1/2 watt resistor between the pad and Ground.

STEPPING RATE FIX

by

Bob Swoger and Larry Kenny

The May 1989 issue of Capital District T/S Computer Club's newsletter told of two members using TS2068's with LarKen DOS having trouble reading each other's disks. It seemed that Fred Lewis could read John Warren's disks but John could not read Fred's disks. This sounded familiar to me as I once had trouble reading disks that came from Rod Gowen and Larry Kenny.

The problem I had was head stepping rate. The drives I use have a 30ms stepping rate. This means that to move from track 0 (the directory track) to track 40 requires 1.2 seconds. A 6 ms stepping rate requires only 240ms. What the controller chip does is wait a period of time to begin reading or writing the disk to give the head time to get to the desired track! The information for the stepping rate is placed in track 0 when the disk is FORMATED.

There is no problem reading the directory track because the head starts there, but the head on John's drive is not yet over the correct track when the controller tries for the read. For this reason John Warren could not read Fred Lewis's disks. Fred either has a newer controller chip or has drives with a 6 ms stepping rate and FORMATED his disks accordingly.

I could not read the SYSTEM disk that Larry Kenny sent me. Larry had not run into the slower drives up to that point. Mine are DEC dual drive units using TEAC drives purchased in the early 80's as a group buy and are so reliable I just won't give them up.

Larry was sure he had the answer when I phoned him and to get me started back in February of 1988, he sent a program to the club BBS at 2 AM while I slept to fix his disks so that I might read them. I tried it and it worked! I re-wrote it and called it STEPRT.B1. The stepping rate information is in track 0 and his program fixes the disk to the 30 ms stepping rate.

Now about controller chips, Larry stated that he started using the new Western Digital chips that could only select 6 ms or faster head stepping mechanisms. I told him, each time I ordered, to send me only the old WDC WD1770 controller chip in units he built for me and my friends as over 200 of these TEAC units are in the area and we all want to be compatible.

When passing around disks for the LarKen disk operating system be sure they are formatted single side 40 track with a 30 ms head stepping rate so we all can read them.

To read unreadable disks, just load in the program below, it will autorun. After removing the write protect label from the faulty disk, place it into DRIVE 0 and press <ENTER>. You will be able to read it from then on. I have tested it on LARKEN and RMG disks.

10 REM STEP RATE CHANGER v1g.0
BY LARRY KENNY AND BOB SWOGER

20 REM Further modified by
G. Chambers.

25 CLS : PRINT "" Disk Head
Stepping Modifier""

30 PRINT " A common head stepping rate is 6 ms, however some older drives need a slower head stepping rate. This program allows you to change it to another value."" Remove the write-protect label, select the drive & the desired head speed, and place the disk to be modified into the drive. Then press ENTER"

35 PRINT "TAB 8;"and....."
Presto... The job is done."
38 INPUT "Select drive(0/3) ";
dr: IF dr 0 OR dr 3 THEN GO TO 38

40 RANDOMIZE USR 100: GO TO dr

50 INPUT "Select head speed(6/
12/20/30)";hs
60 IF hs=6 OR hs=12 OR hs=20 OR
hs=30 THEN GO TO 80
70 GO TO 50
80 POKE 40015,(0 AND hs=6)+(1
AND hs=12)+(2 AND hs=20)+(3 AND
hs=30)

90 RESTORE 150
110 FOR n=40000 TO 40030
120 READ b: POKE n,b: NEXT n
150 DATA 243,205,98,0,175,50,29
32,205,126,0,205,123,0,62,3,50,
134,32,205,120,0,58,100,0,251,20
1,0,0,0,0
250 RANDOMIZE USR 40000
300 PRINT "" Head speed changed to ";hs;" ms."
400 STOP
9000 PRINT USR 100: SAVE "rate.B
1" LINE 1

MIKE'S NOTEBOOK

By: Michael J. Di Rienzo

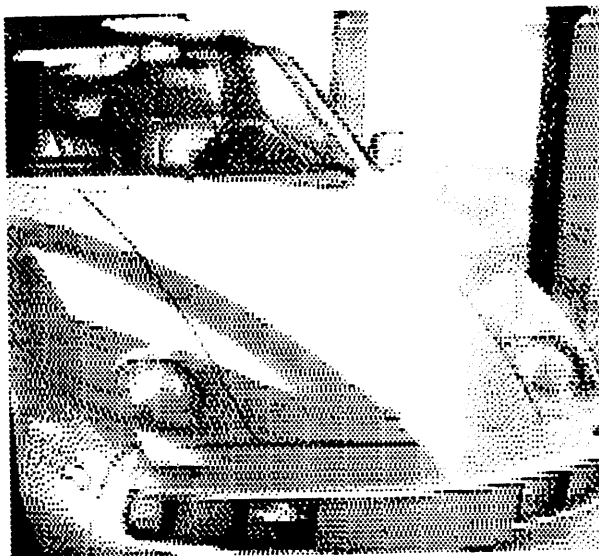
(NOTE: REPRINTING OR REPRODUCING THIS COLUMN WITHOUT THE EXPRESS WRITTEN PERMISSION OF THE AUTHOR IS HEREBY PROHIBITED. FOR PERMISSION, WRITE THE AUTHOR IN CARE OF RMG ENTERPRISES.)

My friends often encourage me to put my powerful TS2068 ROM to work for a good cause. What better way than to win the state lottery. Now, I know that lotto number generators are ubiquitous but this one really works (fast). It can be modified for your geographical requirements. All you need to know is the range of numbers possible and the maximum number of picks per game. In my state, the player specifies six (6) numbers ranging from 1 to 44. To change game parameters, type in the following program and make the following changes: The range of numbers (1 to 44) are specified in lines 40,60, and 120: The number of picks (6) are found in lines 50 and 130. Line 20 asks how many lines you would like to choose. Just follow the prompt and you will be presented with one to ten extremely lucky numbers. GOOD LUCK!!! You may forward me 10% of all your earnings and be assured that you are contributing to a good cause.

Happy TIMEXing...

"LOTTO" By Michael Di Rienzo

```
1 BORDER 4: PAPER 0: CLS : PA
PER 1: INK 6
2 FOR x=0 TO 31 STEP 2
3 PRINT FLASH 1;AT 0,x;CHR$ 1
43;CHR$ 128;AT 21,30-x;CHR$ 128;
CHR$ 143
4 IF x<22 THEN PRINT FLASH 1;
AT x,0;CHR$ 143;AT x+1,0;CHR$ 12
8;AT 20-x,31;CHR$ 128;AT 20-x+1,
31;CHR$ 143
5 NEXT x
10 RANDOMIZE : PAPER 0: INK 7
20 INPUT "How Many Lines (1-10
)? ";q: IF q>10 THEN GO TO 20
30 FOR g=1 TO q
40 DIM s(44)
50 FOR n=1 TO 6
60 LET a=1+INT (RND*44)
70 IF NOT s(a) THEN LET s(a)=a
: GO TO 90
80 GO TO 60
90 NEXT n
100 PRINT AT g*2,5;"(";g;" ) ";
110 LET c=0
120 FOR n=1 TO 44
130 IF s(n) THEN LET c=c+1: PRI
NT s(n);", " AND c<6;
140 NEXT n
150 BEEP .01,20: PRINT "'
160 NEXT g: PRINT #1;AT 1,10;"G
ood Luck!!"
170 PAUSE 0: STOP
```



DID
YOU
KNOW
?

That you can cut disk copy time in half with two of the programs on club disk #29 - Bytepowers Disk Utility Software? I thought I had all the utility programs I needed until I tried this one. Here is a few highlights:

♦ COPY 3 is an impressive program that must somehow compress the data during the copy cycle. I copied a disk full of small programs (fonts) and was amazed to see it copy 36 blocks on the first pass. The whole disk of 77 blocks was copied in 3 or 4 passes. (Club disk #37 was copied in 1 min. 43 seconds - Larken Move takes 4 min. 15 seconds.)

♦ COPY 1 copies 6 blocks per pass and will give you an error message if bad blocks are encountered. Disk #37 took 1 minute and 52 seconds.

♦ COPY 2 copied disk #37 in 4 minutes 36 seconds to an unformatted disk. It formats and copies at the same time.

♦ COLORED DISK NAME allows paper and ink to be used in the disk name. I have coded all my Spectrum disk in red and all Pixel Print disks in yellow. The #29 disk gives a rainbow look when it is called by CATALOG. (Be sure to move the cursor to the bottom line before saving.)

♦ NAME THIEF will pluck the disk name from one disk and install it on another disk.

♦ HELLO! is a menu system that allows you to scroll through all the programs on a disk and load your choice with a single keystroke. It allows you to switch drives and repeat the process. This has become my favorite menu program. You may elect to see only the basic programs or all files.

♦ TINY BOOT is a shorter less sophisticated program similar to HELLO! that displays more programs on screen at one time. It does not show code files, but is Spectrum compatible. I now use this one for most of my Speccy disks.

♦ FORMAT DISK checks for bad blocks during formatting, maps out any bad blocks and reports the results.

♦ ALPHABETIZE CATALOG rewrites the directory such that a CATALOG comes up with an alphabetical listing of your disk contents. This is handy for use with the two menu programs mentioned above.

♦ CLEAN UP DISK clears unused blocks on disk and deletes invalid file names. It also maps out any bad blocks it encounters.

♦ BLOCK ANALYZER moves from block to block showing which file uses it, start address, total length, percentage of block used and if it is in use.

♦ FAST CLEAN UP is twice as fast as CLEAN UP but does not delete invalid files nor map out bad blocks.

♦ FILE ANALYZER displays start address, length, auto-start line variable offset, autorun address, number and list of blocks in use.

The remaining files are either self explanatory or ones I haven't checked out yet:

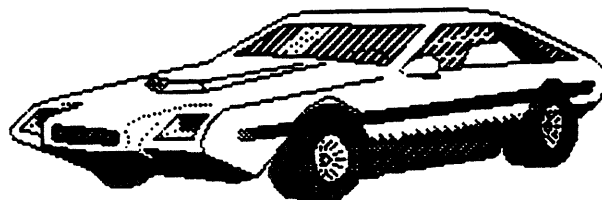
♦ CHANGE DISK NAME
♦ CHANGE HEAD SPEED
♦ BLOCK EDITOR
♦ RECOVER BAD DISK
♦ RECOVER ERASED FILE
♦ TRACK MAP EDITOR
♦ CHANGE TRACKS AND SIDES
♦ INITIALIZE CATALOG

♦ BATCH ERASE
♦ MAP OUT BAD BLOCKS
♦ BATCH MOVE
♦ RECOVER CATALOG
♦ RENAME DISK
♦ DISK ANALYZER
♦ FAST FORMAT

Les Cottrell 108 River Heights Drive Cocoa, FL USA 32922

This column was prepared in Tasword, then converted to Pixel Print using the conversion program in the Professional. The font here is the IBM.Cf that comes with the PPProfessional. The font above is the heavy.Cf from Bob Mitchell's club disk #37.

The picture at the right is my other hobby-a 1968 Triumph Spitfire with a home-made fiberglass body. It was made with programs from disk #22.



MORE MIDI For The 2068

by Richard Hurd

I recently sent an audio tape to George Chambers of simple Christmas song arrangements I put together as a demo of what I've been up to with Lou Champagne's MIDI I/F (SINC-LINK Vol.7 No.4 July-Aug '89 issue). The tape gives a demo also of the sound of the Roland MT-32, a synthesiser with no keyboard and the Casio HT-700 a synthesiser with a 4 octave, mid sized keyboard. George wrote back suggesting that I write an article to go along with the cassette demo.

Most of the songs on the demo tape are one verse, one chorus with a few exceptions. I think most will be familiar. Most of the songs are on each side, one side with the MT-32 voices and the other with the Casio's voices. The recording quality is nothing great. It'll get better in time.

I did not play an instrument to record songs on the demo tape, I went to the library, checked out a Christmas songbook and entered the music into the 2068 via a simple program I wrote. It played the synths like a player piano.

MIDI Interface

I wire wrapped my MIDI i/f with help from Jim Stephens book "POWERFUL PROJECTS WITH YOUR TIMEX/SINCLAIR". It has good instructions on wire wrapping, soldering, and the tools required. Wire wrapping is time consuming but not difficult for the hardware hacking novice. I was hoping that an etched pc board for the MIDI i/f would have been available by now but I haven't

heard of one. This would make it more accessible to those who are not 2068 hardware hackers. I wasn't 'till just the last year or so. Because of the simplicity of the circuit, the cost for an assembled and tested MIDI i/f should be reasonable.

MIDI Recap

MIDI (Musical Instrument Digital Interface) is simply a serial communications interface, similar but not compatible with RS-232. Values from 0 to 255 are sent and received over it. Music can be represented in numbers such as a note value, its velocity (how hard or soft a note is sounded), pitch bend, and much more. MIDI also allows up to 16 separate channels where each can be the sound of a different instrument. Or even multiple sounds on one channel like a drum kit where a different sound is assigned to a particular note.

Alternatives

MIDI is probably most popular with the musician who plays a MIDI instrument for its ability to record what the musician is playing in real time, with a tempo. The same MIDI data can be recorded, tho' much slower in step time where the timing advances after a note/chord is entered.

Two alternatives are music composition and entertainment. Neither of which actually require playing a MIDI instrument. This is what interests me most about MIDI and is also what I'll try to focus on.

sequencer. A device that records the parameters of a performance (generally tempo, when and which notes were played, Program Change commands, dynamics, etc.) but not the actual sounds. On playback, the sequencer feeds this information into an instrument capable of translating the stored data into a replica of the original performance. A sequencer, then, is very much like a player piano except that data is stored in computer memory rather than paper roll memory. from, Craig Anderton's book, The Electronic Musician's Dictionary.

The ideal or much more useful sequencer would allow for multiple patterns of variable length which are linked together to form the song. Going this way has many advantages. This is what I'm working towards.

Sequencing

In order to make use of the MIDI i/f we need a program to record and play back the stored MIDI data or sequence. In other words, a sequencer (see side bar). A sequencer can be either hardware, a single purpose computer in a box, or a software program for a computer. Since we already have a capable computer, all we need is a sequencing program.

About a year ago I wrote a sequencer program so I could experiment with and learn more about MIDI. It is a very simple sequencer. It is step time, or the note data is entered from the 2068's keyboard. I call it mse. The sequencer has one track the length of the song. It allows for only one set tempo and signature. It will hold between 2000 and 2500 notes which can be assigned to any 1 or all 16 MIDI channels. It can play or send out all MIDI status bytes but the only user i/f I've written for it is for note on/off data entry. This

leaves much to be desired. I could go out and buy a hardware sequencer but that would take all the fun out of doing it with the 2068. If anyone is interested in mse, let me know and I'll send a copy to George for the group's disk exchange. It requires the LarKen DOS cartridge.

The ideal or much more useful sequencer would allow for multiple patterns of variable length which are linked together to form the song. Going this way has many advantages. This is what I'm working towards.

More MIDI

In order to make the most of MIDI it is necessary to get a good grasp of the definitions of the MIDI status bytes. Here are a couple of good books that will do the job. You do not need both, if you're only interested in using MIDI, then the first will do fine. If you do program then the second is highly recommended. They are:

For the MIDI user: MIDI for Musicians by Craig Anderton. \$14.95

For the programmer/user: MIDI PROGRAMMER'S HANDBOOK by Steve De Furia & Joe Scacciaferro. Price \$24.95 U.S.

Synthesizers

A little about the synths. The Casio HT-700 has 8 note polyphony (sounded at one

time). The 8 notes can be assigned to one sound or split up into 4 notes for the melody, 3 notes for the chord and 1 for bass. Plus it has short built in rhythm/percussion sequence. It is stereo. It comes with 40 melody voices/patches. The voices and the rhythm sequences are programmable. It is not velocity sensitive. It responds to MIDI sync, note on/off, program change. It can be reassigned from channel 1 to 13. It has a pitch bender. There is no system exclusive communication, any voice or sequence modifications can be saved/loaded to an optional RAM card. It has built in speakers and line outs for external amplification. Discount price new: \$169.

The Roland MT-32 has 32 note polyphony, in a way. Up to 9 different sounds can be heard at one time, 8 of which can be assigned to a different voice. The 9th is a rhythm channel with some pretty good percussion sounds. The synthesis type is Linear Arithmetic or L/A Synthesis. The voices consist of 1 to 4 partials.

So using 1 partial voices would allow sounding 32 notes spread out over the 9 different channels. Using 4 partial sounds allows for only 8 notes to be sounded at one time, with your choice of channels. It comes with 128 sounds quite a few are orchestral instrument sounds, tho' there are some good synth sounds also. The voices are programmable but not from the MT-32 it requires a

computer and editor program. I have written one that allows access to all voice parameters and save/load to LarKen disk. I also got ahold of 256 third party sounds for it in four volumes from the Digital Warrior. They came on an Atari ST disk and I extracted the data in order to use it. Some real nice sounds. The MT-32 responds to the full MIDI note range (1 to 127) velocity and pitch bend. Control change (14 bit) modulation, volume, panpot (stereo left/right) and expression. Program change 0-127, all notes off, active sensing and system exclusive. It also has built in reverb. It requires an external amplifier. Discount price new: \$400, used \$270.

In order to get a variety of instrument sounds (at the same time) requires a multitimbre type synthesizer. There are many to choose from these days. Maybe you have one you could tell us about. I would like to hear about it.

It is helpful for the beginner to go to the local music store for purchases and help, but going mail order for the purchase can definitely save you big bucks. Look for a copy of Keyboard magazine in your neighborhood for a good selection of mail order sources. Also watch the classifieds in the paper.

I hope this sparks some interest in some of you. I know that there are many 2068 users who enjoy making music with their 2068.

ZX 81 LOTTERY FUN

Mel Richardson

Windsor

Like lots of people, I participate in a regular group lottery purchase at my workplace. I recently figured that it would make sense to let the machine check the ticket so here is the result of that brilliant observation.

The program stores lines of numbers regularly played. One only need enter the drawn numbers each week and the screen will show matching numbers as inverse. Don't "RUN" the program after entering your lines and saving the data filled routine.

Because the basic algorithm is fairly short, there was room to make it quite friendly. The "QUIT" section is written for ZX 81 and Larken DOS and you should substitute for your own system. Good luck!

```

1 REM ***ZX81 LOTTERY CHECK***
2 REM
4 REM *****DON'T RUN*****
5 REM
6 SLOW
8 CLS
10 PRINT AT 2,3,1:"L O T T E R
Y 6 4 9 *"
15 PRINT AT 8,5:"ENTER WINNING
NUMBER":AT 10,5:"CHECK TICKET N
UMBERS":AT 12,5:"PRINT OUT TICKE
T":AT 14,5:"NEW TICKET NUMBERS"
17 PRINT AT 16,5:"QUIT OR SAVE
"
20 GOTO 20+(INKEY$<>"")
25 GOTO 20+(140 AND INKEY$="E"
)+(360 AND INKEY$="C")+(280 AND
INKEY$="P")+(180 AND INKEY$="N")+
(7980 AND INKEY$="Q")+(7980 AND
INKEY$="S")
100 CLS
101 PRINT "X TO CONTINUE WILL M
IPE TICKET "
102 PRINT AT 2,4:"PRESS ""X"" T
O CONTINUE."
103 GOTO 103+(INKEY$<>"")
104 IF INKEY$<>"C" THEN GOTO 1
105 REM ---DIM FOR TICKET SIZE---
106 CLS
108 PRINT "ENTER NUMBER OF LINE
S:"
110 INPUT G
114 DIM A$(G,17)
116 PRINT AT 0,0:"ENTER "G:" L
INES(WITH SPACES):"
118 PRINT AT 20,1:"** ** ** **
** **"
120 FOR I=1 TO G
130 INPUT A$(I)
140 PRINT AT I+1,0:A$(I)
150 NEXT I
152 PRINT AT 20,1:"OK? Y/N:

```

```

153 LET X=PI**PI
154 GOTO 154+(INKEY$<>"")
156 IF INKEY$="N" THEN GOTO 106
160 CLS
170 PRINT "ENTER WINNING 649 NU
MBERS...","INCLUDING SPACES AN
D BONUS:"
180 PRINT AT 20,1:"** ** ** **
** ** BB"
182 INPUT B$
184 PRINT AT 5,0:B$
186 LET BB=VAL B$(19 TO 20)
210 PRINT AT 8,0:"ENTER DATE OF
DRAW:"
220 PRINT AT 20,1:"DDMMYY
"
230 INPUT D$
240 PRINT AT 10,0:D$(1 TO 2)"/
"/D$(3 TO 4)"/"/D$(5 TO 6)
250 PRINT AT 20,1:"OK? Y/N:"
255 LET X=PI**PI
260 GOTO 260+(INKEY$<>"")
270 IF INKEY$="N" THEN GOTO 160
290 GOTO 380
295 REM ---PRINT OUT NUMBERS---
300 CLS
310 PRINT "CURRENT NUMBERS:
320 PRINT
330 FOR I=1 TO G
340 PRINT A$(I)
350 NEXT I
370 GOTO 1000
375 REM ---COMPARE AND PRINT---
380 CLS
390 PRINT "649 DRAW FOR: "/D$(1
TO 2)"/"/D$(3 TO 4)"/"/D$(5 T
O 6)
392 PRINT
393 LET B$(19 TO 20)=STR$(BB)
395 PRINT B$
400 PRINT
405 LET B$(19 TO 20)="00"
410 FOR I=1 TO G
420 FOR L=1 TO 16 STEP 3
425 FOR T=1 TO 16 STEP 3
430 IF B$(T TO T+1)=A$(I,L TO L
+1) THEN GOTO 460
435 NEXT T
440 IF NOT B$(T TO T+1)=A$(I,L
TO L+1) THEN PRINT A$(I,L TO L+
1)":"
450 GOTO 520
455 REM ---REVERSE MATCHED NUMBE
RS---
460 LET M$=A$(I,L)
470 LET N$=A$(I,L+1)
480 LET M=CODE M$
490 LET N=CODE N$
500 PRINT CHR$(M+128)
510 PRINT CHR$(N+128)":"
520 IF L=16 THEN PRINT
530 NEXT L
550 NEXT I
560 GOTO 1000
999 REM ---COMBAR---
1000 PRINT AT 21,0:"COPY MENU
QUIT"
1010 GOTO 1010+(INKEY$<>"")
1020 IF INKEY$="C" THEN COPY
1030 IF INKEY$="M" THEN GOTO 1
1040 IF INKEY$="Q" THEN GOTO 800
0
1100 GOTO 1010
7999 REM ---QUIT---
8000 CLS
8020 PRINT "SWITCH FOR #LDOS*"
8030 PAUSE 4E4
8040 CLS
8050 PRINT USR 14336
8060 REM
8070 STOP

```

The Québec Link

by:
Réal Gagnon
8286 St-Hubert
Montréal (Qué)
CANADA H2P 1Z3

Sinclair TRIVIA

You think that you are a Sinclair expert? Ok, try to answer the following questions.

- Who was the manufacturer of the american QL?
- Tony Tebby, the father of QDOS, was hired at Sinclair for what job?
- What was the canadian suggested retail price of the QL when it was launched?
- What is the minimum length of the microdrive magnetic tape?
- The TS2068 has an unique feature for a Sinclair computer, what is it?
- Name 2 other computers that can emulate (by software or hardware) the QL.
- What is the unformatted capacity of microdrive?
- How many jobs can be runned simultaneously on a QL with 128K?
- How many channels can be opened simultaneously on a QL with 128K?

Tips and Tricks

Here is a collection of QL tips to make our life easier. If you are already a "QL power user", those tips are old stuffs but maybe you will find 1 or 2 unknown to you. Please note that Tony Tebby's TOOLKIT II is needed for some of them.

A classic tip.....

Take a simple thing like printing a directory to your printer. The classic solution is to type :

```
OPEN #5,ser1
DIR #5,flp1_
CLOSE #5
```

But with Toolkit II, we do the same thing with only one command :

```
DIR \ser1,flp1_
```

Extras lines.....

We want to put the maximum information on the screen so we need more lines, try this : CHAR_INC #0,6,8
it will do the trick in window #0.

Mysterious CLS.....

These are not really tips but only weird effects of the CLS command when used with silly parameters.

```
CLS 108 will create a green border in window #1
CLS -20 will create a different border
CLS 8 same effect as PAPER 0
```

```
CLS 9 same effect as INK 0
CLS 11 same effect as UNDER 1
CLS 115 equivalent to CURSOR LEFT
ex. PRINT 12345
CLS 115:CLS 115:CLS 115
PRINT 67890
CLS 116 equivalent to CURSOR RIGHT
```

ABACUS and my printer.....

It is easy to control the printing of our worksheet from ABACUS with the command CHR to send control codes to the printer.

If in ABACUS, we type CHR(64), the character "A" will appear. The trick is to put first CHR(0) and then all the control codes for the printing. Those control codes will not appear on the screen but will be sent to the printer.

For example if we want to underline something, we type, say in cell A1:
CHR(0)+CHR(27)+"-1"+"The Quebec Link"
and in cell A2: CHR(0)+CHR(27)+"-0"
to cancel the underline mode.

That feature can be very useful to print wide worksheet. On most printers, it is possible to obtain 127 characters per line when the printer is in compressed mode.

In the first cell, we put CHR(0)+CHR(15) to switch compressed mode on and in the last cell we put CHR(0)+CHR(18) to turn off the compressed mode. However, if we want to print a worksheet larger than 80 characters we must notify ABACUS first by selecting the option DESIGN and set the PRINTER PAPER WIDTH to the correct value (like 127).

How old are you, little udrive?

Even if your cartridge refuse to give you an answer, it is possible to crack the secret by examining carefully the cartridge. On each udrive, there is a 4-number code. The first 3 represents the manufacturing date and the last number is the last digit of the year. For example, the code "1446" means that the cartridge was made the 144th day of 1986.

Solution of Sinclair TRIVIA:

- SAMSUNG
- satellite dishes
- 800\$ and d) minimum 5.1m (about 15ft.)
- ATARI ST (hardware) & AMIGA (software)
- about 180Kb
- 56 jobs with 128Kb
- 168 channels with 128Kb

BIG SPLASH

TWO IDEAS GO UNDER QL'ers Don't WANT to do it! by Hugh Howie

QL Hardware

What do QL'ers do for kicks?

They sit around and ask why someone does not start a Hardware Library. They say they would like to do things with their machine but don't know how to go about it, or what can be done. All they ask for is for someone to do something.

So someone comes along and writes a letter asking for submissions to start a hardware library, and what happens? Zilch. Not a word of encouragement or a single submission. Someone said to me "Good idea", but he was the only one who thought so. No one else could be bothered.

What a sorry state of affairs when we cannot even be bothered to write a note of encouragement, or send something in. There just has to be someone who has done something to improve their machine. Ever since I got my QL I have heard about the limitations of the QL. Yet, when asked for ideas and methods to improve it, No one seems to know what to do.

The conclusion is that the QL is a computer without peer. It is so perfect there is nothing to be done to improve it. I am very happy indeed that in my choice of computer I selected such an excellent machine. I am sorry for those folks who purchased such inferior units they have to be constantly trying to improve them. Not so with the QL. It was perfect to start with.

When I requested submissions I thought I would get some, not many, but some. I got not one. That is correct, not one.

So next time you wonder what to do to correct something, you will just have to do what you do right now.
Wonder 888

SuperBasic Instruction

Last month I asked if anyone would be interested in taking part in a type of Correspondence Course in SuperBasic, either as a user or supplier of such a course. At that time I stated I would publish the results in this Newsletter. Here is the result of that request:-

Applications to LEARN	NIL
Applications to INSTRUCT	NIL

Now how do you like that? Just goes to show the interest taken in the learning process.

I would have thought there would be some replies, not many, but some. but not even one.

The same goes for so many things in this world, we are too inclined to say "Why don't THEY do something for us?" The answer is right there, it is always someone else who has to do something.

Not in my backyard syndrome at work.

The same goes for so many excellent magazines discontinuing publication, not enough interest.

If you were to look through any magazine you would perceive that the same names keep recurring in the articles.

"Why don't sumbidy do sumpin?"
The answer is "Why bother, nobody reads it, and if they do read it, they don't do nuttin"

Come on folks, do something, write something. If you are mad at me, at least kick me in the teeth so that I know you are there. 888

This page was designed and printed from text²⁷. including header and footer, and the two-column setup for text in the middle. ONE PASS. Have you tried this in Quill?

SOFTWARE REVIEW

by Chuck Kerejuck

PIX FIX

JOHN T. NGUYEN

PRESS : MENU

- M - FOR THIS MENU
- H - FOR HELP SCREENS
- C - TO CREATE PICTURE
- L - TO LOAD PICTURE
- S - TO SAVE PICTURE
- X - TO CLEAR PICTURE
- P - TO COPY TO PRINTER
- Q - TO QUIT

No, you're not having a *deja' vu*. We didn't do this review last month. This is a whole 'nother different review.

In fact, the only thing similar between last month's review object and this one is their similar-sounding names and the fact that both are graphics programs. After that, they're worlds apart. Okay?

We're talking, of course, about PIX FIX, written by John T. Nguyen for the 2068 computer.

While PIX-FX is used for manipulating graphics, PIX FIX is used to create them. It does an interesting job of doing that, too.

How does it work? To create art with PIX FIX, said artist will work at the pixel level. Basically, it's a fancy Etch-A-Sketch. What's interesting is the fancy stuff. For example, while working at the "pixel level", PIX FIX will display a full-screen 32-column X 24-line grid.

The grid comes up in "neutral" mode, so you can freely place the "pixel cursor" anywhere you choose. There is the expected "draw" and "erase" modes, to plot and unplot to your heart's content. The screen border will change color, by the way, when flipping between these modes, to give a visual indicator.

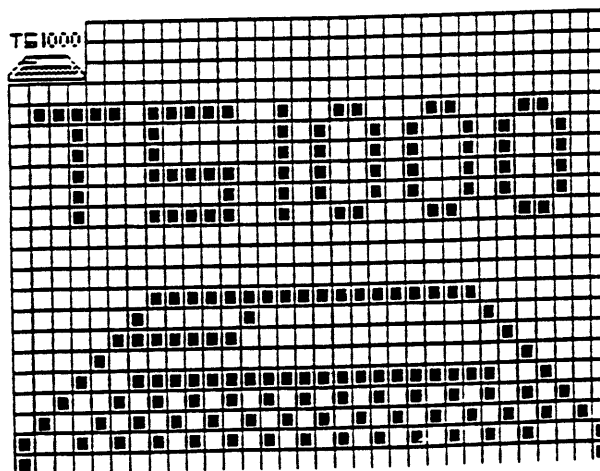
The kicker to this grid approach is that in the upper-left portion of the screen, there is this mini-screen, 4

"pixels" wide by 3 "pixels" high, to show you what you're doing in real-size. It even "follows" your art as it's created! Once you get used to this convenience, you may wonder what other devious purpose it has. A mere press of the 'U' key will pop you into Big Pix Mode, where you discover that little design square you were working on is but a mere building block to a much larger picture! Each block can be moved around in its entirety, to be placed wherever it looks the best.

The only catch is that you first position an empty block and then draw into it. At this point, PIX FIX resembles more of a puzzle that has to be assembled from your pre-designed pieces. Once the puzzle-piece is positioned just right, press 'U' again to get back into Screen Mode, where you resume drawing and/or erasing.

The feature of moving an empty block around offers the added feature of it actually becoming a window to your final picture. This empty-block/window can be positioned partly over a previous "puzzle piece" for the purpose of adding more detail to a section. Do this and press 'U' again to display a partial graphic on the main screen.

This goes on and on until a final masterpiece emerges. An extra pair of features add just a smidgeon to the overall flexibility and easiness. While PIX FIX doesn't have many of the fancy features found in more exotic 2068 graphics programs, it does let you display your entire picture in inverse and turn on row/column count indicators to accurately keep track of where you are, if you feel so inclined. When satisfied, the user will return to the main menu to SAVE (my copy was configured for Zebra disk systems) or COPY to a printer.



While being interesting to assemble a drawing in puzzle-piece format, the program does exhibit some odd peculiarities. One is the method in which to move your plotting cursor and your mini-picture puzzle-piece. One would expect the program to make use of the arrow keys (5,6,7 & 8), since this would be logical. Instead PIX FIX employs the 'I' key for UP, the 'K' key for DOWN, 'J' for LEFT and 'L' for RIGHT. How come? Beats me!

Another oddity is when saving a Screen Mode picture. A prompt will come up asking whether you want to SAVE as a SCREEN\$ or from memory.

SAVEing from memory will SAVE as a SCREEN\$ anyway, but the result will be from the Big Pix Mode; in other words, as a small block. Using the SCREEN\$ option SAVES a blank screen! To get the TS1000 illustration shown, I had to BREAK the program in Screen Mode and SAVE it separately. Another option, of course, is to SAVE it from memory (small) and use PIX-FX to expand it to normal size!

The print option shown in the main menu appears to be designed with the Aerco printer interface in mind (characterized by the "LPRINT CHR\$ 1" command in line 836 of the program).

Presumably, one can freely change this for a 2040 by substituting COPY.

The Help Screens and Menu can be accessed from within any part of the program, which helps when forgetting the I-K-J-L keys. A mere press of ENTER from Help returns your drawing.

While being quite BASIC in nature (There is no separate CODE when LOADING the program.), the puzzle-piece format gives PIX FIX its strongest suit. It's worth checking out.

For a copy of PIX FIX, see our club librarian, Dale Fritz, or you may wish to use your own sources.

It's definitely a lot different than PIX-FX!

Taken from the Sept 1990 issue of SWYM, the n/1 of the Seattle Area T/S Users Group.



TS2068 Digitized Pictures

DOCUMENTATION FOR "CHARACTER SET EDITOR"

by

Bob Mitchell 20 Wild Briarway Willowdale Ont M2J 2L2.

Introduction

This article derives from my using Pixel Print Professional to make keyboard overlays for the TS2068, a technique covered in Les Cottrell's article in the Nov-Dec 90 issue page 25. On page 26 there was a sample overlay: Pixel Print Plus!

This is truly an intriguing application of PPP, bank switching and the old workhorse TS2068 and in the process of making several overlays to suit my particular whims, I found a need for a variety of character sets (fonts) that were turned sideways (90 degrees to the right or a quarter turn clockwise). There was some urgency to create sets with narrow characters that would allow words to be crammed into the small spaces on the overlays.

Changing any one character set manually involved an inordinate amount of work and it seemed the computer ought to be able to do that for me. I set about the task of writing a suitable utility that would enable me to edit a set if need be and then to rotate it in its entirety by one press of a key.

No need to do this from scratch though! I put an old program to use: this one was called "Character Redefinition" written by D. Robinson and appeared in Personal Software Autumn 1985 p.18.

Since the original allowed for several sets to be stored and worked on, I stripped out much of the material in the BASIC to allow only one set to be processed or stored at any one time.

Rotating the character sets has been accomplished using a short 42-byte machine-code routine found in the Supercode 3.5 collection, by F. Vachha BSc & V.B. Rumsey. Although written for Spectrum, it is compatible with the TS2068 and allows a complete set to be rotated a quarter turn clockwise in about 3 seconds. A half turn will produce an upside-down set while a three-quarter turn will result in a set that is rotated 270 degrees to the right (ie, 90 degrees to the left).

The machine-code routine is contained in DATA statements at line 8500 and is loaded into memory beginning at the start address at 63163.

Instructions

FACILITY MENU

There are three options in this menu:

- 1 = DEFINE set.
- 2 = SAVE/LOAD set.
- 3 = ROTATE whole set.

First, it is necessary to load a character set for editing or rotation. This is done by selecting option 2. Any character set or graphics set may be loaded. The Facility Menu will then reappear.

Next, it is wise to use option 1 to view the loaded set which will appear at the bottom of the screen in three lines each of which lies directly beneath the standard TS2068 character set shown in INVERSE VIDEO. At this point it is possible to select a character for editing or to opt for rotating the entire set.

There are three options to rotate a whole 96 character set clockwise: 1) 1/4 turn, 2) 1/2 turn or 3) 3/4 turn. Note that each turn is a 1/4 revolution or 90 degrees clockwise. Thus, if

the initial font has already been turned once, using option 2 will do two more turns and leave the set $3/4$ of a turn from the straight up or vertical position. Using option 3 will take about 10 seconds to complete.

DEFINER MENU

As soon as a character has been selected, the menu appears. This menu is located at the right of the screen alongside two 8 x 8 grids, the left one being used for comparison purposes, the right for editing. The grid cursor flashes and makes a blipping sound.

The keys to be pressed are as follows:

Cursor keys are used to move the grid cursor about.

p = plot or fill in a square; ie, one pixel in the character being worked on.

e = erase a square.

c = store a character <chr>. (You can use a character set to hold any graphics of your choice, not just letters and numbers, eg, borders can be created.)

n = store a character and then go back to the facility menu.

o = display a character in the left grid.

q = overlay the current make up of a character in the editing grid.

m = facility menu.

When no character is being edited, a prompt line appears at the bottom of the screen for entering any character to be redefined. Before editing a character or when just viewing the whole set, you can return to the facility menu by selecting <space> and then using option <m> to get back.

All editing on a set should be completed before the set is rotated. As each character is edited its new appearance shows in the lower part of the screen under the pertinent character in inverse video.

Some uniformity in file names to identify the rotation was needed; I chose to identify fonts rotated to the right by suffixing their file names with a > character; those to the left with a < character; the normal straight-up sets with a _ (underline bar) character. The first character must be a ! to be Pixel Print compatible. For the Pixel Print Professional application designing keyboard overlays, only $1/4$ turn right-rotated fonts will normally be used.

The disk containing this program and a host of fonts is available from the library. Len Cottrell has done several overlays; I have completed two (HOT Z and PRO/FILE) as well as a modified MSCRIPT overlay.

The Listing for User Entry

First, some notes:

The REM statements at the end of some lines are for the user's information when entering the listing. These can be left out.

Line 4080 is used to POKE the address of a character into the rotate code. This could be done with a Larken double POKE but I left this in as it can be compiled while the Larken POKE

cannot.

There is not much percentage in compiling this program though. In fact I tried it and it has the disadvantage of speeding up the grid cursor during edit which I found difficult to slow down (to my satisfaction). The rotate code would operate faster but is ten or twelve seconds for three rotations too much time?

```
100 REM CHARACTER SET EDITOR
110 REM By Bob Mitchell 1990
120 RESTORE 8500
130 FOR i=0 TO 41: READ n0: POKE 63163+i,n0: NEXT i
140 POKE 23658,0: POKE 23607,60: BORDER 0: PAPER 0: INK 7: OVER
0: CLS
150 LET j$=""
!""#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{ }": REM there is only one space
between the first quotation mark and the exclamation point.
160 LET b=1
170 LET po=60
180 DIM c(12): DIM h(8)
190 LET h(1)=128: FOR f=2 TO 8: LET h(f)=h(f-1)/2: NEXT f
200 BORDER 0: PAPER 0: INK 7: CLS : PRINT "Character Set
Editor"""" MENU ": PRINT ""1 > DEFINE set""""2 >
SAVE/LOAD set""""3 > ROTATE whole set":REM MENU in INV VIDEO
210 LET i$=INKEY$: IF i$<"1" OR i$>"3" THEN GO TO 210
220 IF i$="1" THEN GO TO 1000
230 IF i$="2" THEN GO TO 2000
240 IF i$="3" THEN GO TO 4000
1000 PAPER 1: OVER 0: BORDER 1: INK 9: CLS : PRINT AT 0,0:"1 >
Editor"
1010 FOR f=1 TO 6: POKE USR "a"+f,129: NEXT f: POKE USR "a",255:
POKE USR "a"+7,255: REM graphic A is an open square
1020 POKE 23607,60: PRINT AT 15,0: INVERSE 1:j$(1 TO 32)''j$(33
TO 64)''j$(65 TO 96)
1030 LET ch=251
1040 POKE 23607,ch: PRINT AT 16,0:j$(1 TO 32)''j$(33 TO
64)''j$(65 TO 96)
1050 OVER 0: INK 7
1060 POKE 23607,po: INPUT AT 0,0;"Character to be edited ? ";
LINE c$: IF CODE c$>127 OR CODE c$<32 OR LEN c$>1 THEN BEEP
.5,-20: GO TO 1060
1070 PRINT AT 13,0: INVERSE 1:" EDITING: ";c$:REM EDITING: in
INV VIDEO
1080 FOR f=1 TO 8: PRINT AT f,0:"AAAAAAA AAAAAAA": NEXT f: REM
A's are GRAPHIC A's
1090 PRINT AT 0,0: INK 2: PAPER 7:"COMPARE 87654321"
1100 LET x=1: LET y=9: DIM a(8)
1110 DIM b$(8,8): LET k=(CODE c$-32)*8
1120 PRINT AT 0,18;"Cursor keys to";AT 1,19;"move.";AT
2,17;"p=plot.";AT 3,17;"e=erase.";AT 4,17;"c=store chr."
1130 PRINT AT 5,17;"n=store chr.";AT 6,18;"and return";AT
7,18;"to menu.";AT 8,17;"o=compare";AT 9,18;"chr.";AT
10,17;"q=overlay a";AT 11,18;"chr.";AT 12,17;"m=facility menu"
1140 PLOT 135,104: DRAW 0,-38: DRAW 120,0
1150 OVER 1: INK 5: PRINT AT x,y;"@_": BEEP .003,x+y: PRINT AT
x,y;"@_": REM graphic is black square on key 8
1160 LET i$=INKEY$
```

```

1170 IF i$="8" AND y<16 THEN LET y=y+1
1180 IF i$="5" AND y>9 THEN LET y=y-1
1190 IF i$="7" AND x>1 THEN LET x=x-1
1200 IF i$="6" AND x<8 THEN LET x=x+1
1210 IF i$="p" AND b$(x,y-8)=" " THEN PRINT AT x,y; OVER 0;"@_
": LET b$(x,y-8)="@_": LET a(x)=a(x)+2^(16-y): REM graphics are
black squares.
1220 IF i$="e" AND b$(x,y-8)="@_" THEN PRINT AT x,y; OVER
0:"A": LET b$(x,y-8)=" ": LET a(x)=a(x)-2^(16-y): REM graphics
are black square and graphic A
1230 IF i$="o" THEN GO SUB 1300
1240 IF i$="c" THEN GO SUB 1290: GO SUB 1440: GO TO 1050
1250 IF i$="n" THEN GO SUB 1290: GO SUB 1440: GO TO 1030
1260 IF i$="q" THEN GO SUB 1370
1270 IF i$="m" THEN CLS : BORDER 0: PAPER 0: OVER 0: INK 7: GO
TO 200
1280 GO TO 1150
1290 OVER 0: FOR f=1 TO 8: POKE ch*256+256+k+f-1,a(f): NEXT f:
RETURN
1300 INPUT "Chr to be compared ? "; LINE g$: IF CODE g$<32 OR
CODE g$>127 OR LEN g$>1 THEN BEEP .5,-20: GO TO 1300
1310 FOR f=1 TO 8: PRINT AT f,0; OVER 0;"AAAAAAA": NEXT f: REM
graphics A
1320 FOR f=0 TO 7: LET gr=PEEK (ch*256+256+((CODE g$-32)*8)+f)
1330 FOR g=1 TO 8
1340 IF gr>=h(g) THEN PRINT AT 1+f,g-1; OVER 0;"@_": LET
gr=gr-h(g): REM graphic is black square
1350 NEXT g: NEXT f
1360 RETURN
1370 INPUT "Chr to be overlaid ? "; LINE g$: IF CODE g$<32 OR
CODE g$>127 OR LEN g$>1 THEN BEEP .5,-20: GO TO 1370
1380 FOR f=1 TO 8: PRINT AT f,9; OVER 0;"AAAAAAA": NEXT f: REM
graphic A
1390 DIM a(8)
1400 FOR f=0 TO 7: LET gr=PEEK (ch*256+256+((CODE g$-32)*8)+f)
1410 FOR g=1 TO 8
1420 IF gr>=h(g) THEN PRINT AT 1+f,g+8; OVER 0;"@_": LET
gr=gr-h(g): LET a(f+1)=a(f+1)+h(g): LET b$(f+1,g)="@_": REM
graphic is black square
1430 NEXT g: NEXT f: RETURN
1440 LET k=CODE c$-32: IF k<32 THEN LET xx=16: GO TO 1470
1450 IF k<64 THEN LET xx=18: LET k=k-32: GO TO 1470
1460 IF k<96 THEN LET xx=20: LET k=k-64
1470 POKE 23607,ch: PRINT AT xx,k;c$: POKE 23607,po: RETURN
2000 INPUT "1=SAVE 0=LOAD ";sl
2010 IF sl THEN INPUT (" SAVE file name <=6 "); LINE n$: REM
SAVE in INV VIDEO
2020 IF NOT sl THEN INPUT (" LOAD file name <=6 "); LINE n$:
REM LOAD in INV VIDEO
2030 INPUT "drive? ";drv: RANDOMIZE USR 100: GO TO drv
2040 IF sl THEN RANDOMIZE USR 100: SAVE n$+".Cf"CODE 64512,768
2050 IF NOT sl THEN RANDOMIZE USR 100: LOAD n$+".Cf"CODE 64512
2060 BEEP 1,10: GO TO 200
4000 REM rotate font
4010 INPUT (" ROTATE CLOCKWISE ""1=1/4 Turn""2=1/2
Turn""3=3/4 Turn "); LINE r$: REM ROTATE CLOCKWISE in INV
VIDEO
4020 IF r$<"1" OR r$>"3" THEN BEEP 1,1: GO TO 4010
4030 LET rot=CODE r$-48
4040 INPUT ;: PRINT #0;"Stand by..."

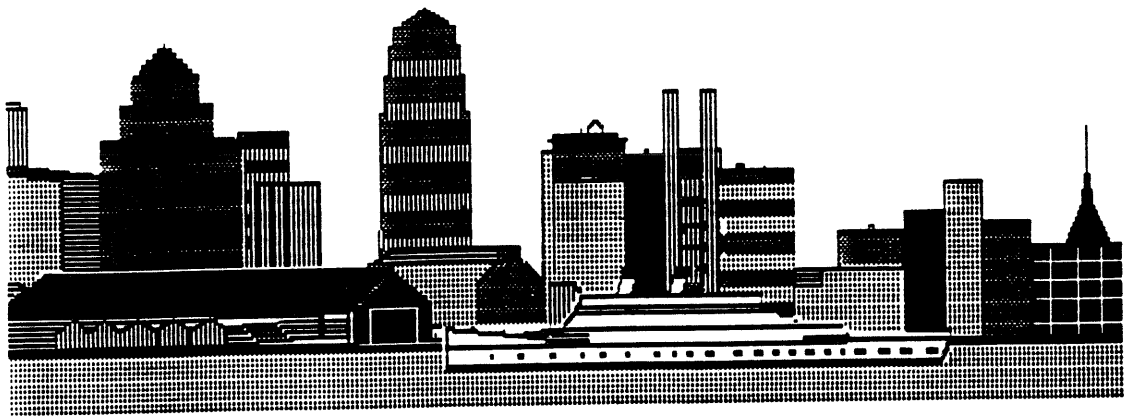
```

```

4050 FOR r=1 TO rot
4060 FOR q=0 TO 95
4070 LET l=64512+q*8
4080 RANDOMIZE l: POKE 63166,PEEK 23670: POKE 63167,PEEK 23671
4090 RANDOMIZE USR 63163
4100 NEXT q
4110 NEXT r
4120 BEEP 1.10: INPUT :: PRINT #0:"Rotation finished.""Press a
key.": PAUSE 0: GO TO 200
4130 STOP
8500 DATA 30,128.33,0,226,6,1,14
8510 DATA 0,229,123,166,254,0,40,3
8520 DATA 121,128,79,203,32,35,48,242
8530 DATA 225,197,203,59,48,231,17,7
8540 DATA 0,6,8,25,209,115,43,16
8550 DATA 251,201
9000 INPUT "SAVE PROGRAM drive? ";drv: RANDOMIZE USR 100: GO
TO drv: RANDOMIZE USR 100: SAVE "redef.B1" LINE 100: REM SAVE
PROGRAM in INV VIDEO

```

1950 - FORTY YEARS AGO - 1990



--- IN 1950 ---

This was the view to greet you after a cruise on Lake Ontario.
The large building on the left is the ROYAL YORK HOTEL.
Just to left of centre is the Canadian Bank of Commerce Tower.

In the fore-ground is the cruise_ship 'CAYUGA'.

This is an impression of a photo in the Archives of Ontario.
Artwork by Hugh Howie on QL, using EYE-Q, Image Processor, and
Page Designer 2.

HOT Z									
C/E=Cursor to END; S/S=Single Step. sw=switch; disa=disassembly; wp=wide printer									
EDIT MODE (WRITE)		READ MODE		CONT.		SINGLE-STEP			
ESC from Ass'y Edit HEX-EDIT Col. Find next matching bytes to part scrn (enter addr) DISPLACE NAME file Load DATA C/E ESC HEX-EDIT/GOTO READ Delete NAME Fill memory with keycode NAME entry (disa/data) RUN C/1st RET Checksum to BODE in S/S Cursor to END (wide/ptmtr) Xfer Code+labels to DEST Code C/E (Set Terms) Hex Arithmetic (e-k) RESTORE Save C/E as DATA Xfer C/E to DEST Find 1st matching bytes S/S at Cursor Ass'y-Edit Col.		Display Jump Table Disa/Data Display Sw Set END address Load Data C/E VERIFY Code format Table SINGLE-STEP Mode ASSEMBLY Mode Disa/Data Display Sw Set END address GO TO BASIC Scroll cont. (BREAK=stop) HEX-EDIT Mode ON Display stack pointer SW Set Border colour (0-7) Flip Page Set INK colour (0-7) Restart HOT Z Copy Screen to 2040		OR OVER NAME files SW PAPER Set PAPER colour (0-7) PEEK FP interpreter SW RND Display NAME list		GO TO READ Mode Display Breakpoint #1 Set Breakpoint #2 Set Breakpoint #3 Back one instr'n/byte ENTER RUN one instruction INT RUN CALL or RST10 LN COPY to 2040 SPACE skip next instruction THEN RUN to breakpoint VAL Set Register (BDEFHSXY) MEMORY MAP BC1A NAME File BF00 HOT Z Vars C000 HOT Z Prog #1 C2A6 DATA Files C930 JUMP Tables CC04 HOT Z Prog #2 F700 End			
BACKSTEP 1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
FILL MEM 1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 7		8 8		9 9		0 0	
1 1		2 2		3 3		4 4		5 5	
6 6		7 							

PROFILE PLUS 5A

REVERSE (BACK) SLASH TIPS		ABBREVIATIONS USED BELOW		MISCELLANEOUS TIPS		NOTES	
<p>BACK SLASH COMMANDS ARE SHOWN IN INVERTED</p> <p>TALLY treats line as 2 halves SUM treats line as 1 whole UPDATE: Use Space bar to pause; BREAK to stop; other key to continue. BLOCK DELETE: Do NOT use "OR" Do NOT use last character in a record as Search chr.</p> <p>End all records with " _ " ie, underline bar to avoid bad searches, block deletes, etc.</p>		<p>CS+* = Use Caps Shift and number in top row SS+ = Use Symbol Shift and option key LP = Print on Wide Printer REC = Record. TALLY = Tally (Key T). K = Continue (Key T). ED. = EDIT (Key I). SRCH = Search DEL BLK = Delete Block (Key D) 2in1 = Use <> vice </> in DEFP to print 2 lines in 1. RVS = Reverse Name (ENTERKey) ## = Preset DEFP formats # = Rec. not to be printed M = with Copy & ENTER</p>		<p>EDIT PHRASE <= 32 Characters UPDATE <= 32 chars; must be at start of line. SEARCH COMMANDS <= 50 chars DATE <= 8 chars. PRINT LINE 16: Use 16 in DEFP. WARNING BEEP with SS+Y means DUPE of rec. made. ----- MENU ----- DRIVE SELECT: Use 0-4 + ENTER. CAT & FULLCAT may be called from Menu. COL = Toggle left/right column for printout.</p>			
CS+* TOGGLE INSERT/OVER	1 1	DELETED LINE 3 3	ERASE CURSOR 4 4	CURSOR 5 5	CURSOR 6 6	CHANGE Edit/ADD Info 9 9	DELETE CHR 0 0
35+ BACK SLASH+ U= UPDATE T= TALLY D= DEL. BLK S= SUM	Q W	E R	CHANGE CLOSE & UPDATE Y	CURSOR 7 7	CURSOR 8 8	INSERT I ED. PHRASE	P
35+ SUM A RECORD	S X	D C	SRCH & K G H	DEFP:2in1 M N	DEFP:2in1 M N	HOLD FOR NAME Enter	RVS: COPY/AUTO Shift
Shift	Z	X	U	B	Symbol Shift	Break (Space)	35+ Shift
<p>PROFILE</p> <p>KEY^B OVERLAY^A RD</p>		<p>3 UPDATE PAUSE 3 CANCEL AUTO-SRCH</p>		<p>Prepared by R. Mitchell for TORONTO TIMEX-SINCLAIR USERS CLUB</p>			

Pixel Print Plus! The TS2000 Desktop Publisher.

<<< MENU SELECTIONS >>>

<p>(K) KEEP-save/keep a doodle pad.</p> <p>(S) SAVE-PIXEL PRINT column.</p> <p>(L) LOAD-PIXEL PRINT column.</p> <p>(I) ICON-LOAD into a doodle pad.</p> <p>(G) GRAPHICS-LOAD/SAVE SCREEN\$.</p> <p>(W) UNDO-replace present doodle pad with last 'KEEP'.</p> <p>(U) WIDE-select character width.</p> <p>(H) HIGH-select character height.</p> <p>(C) CLS-clear display/doodle pad.</p> <p>(M) MOVE-Scroll UP 'T' or DOWN 'B' in three speeds. Speeds 2 & 3 use CS or SS with 'T' or 'B' key.</p> <p>(P) PRINT-to 2040 or full size printer.</p>	<p><<< MENU SELECTIONS >>></p> <p>(F) FONT-activates FONT MENU.</p> <p>Select T-TIME, B-BOLD, M-MODERN, I-ITALICS, S-SPACE, K-KERN, R-RESTORE, or L-LOAD FONTS.</p> <p>(B)-BLOCK FUNCTION MENU.</p> <p>C-COPY or E-ERASE a segment of a column. I-INSERT or D-DELETE pixel row at the top of the cursor.</p> <p>(X)-Shifts to the RIGHT or LEFT column.</p> <p>(Q)-QUIT or terminates the program.</p>	<p>(CURSOR CONTROL)"CAPS SHIFT"+5, 6, 7, or 8 keys to move CURSOR character position LEFT, DOWN, UP, & RIGHT respectively.</p> <p>(CURSOR SHIFT)"SYMBOL SHIFT"+(0)=, (F)(0), (G)(THEN), E(=) to SHIFT the CURSOR 1 pixel LEFT, DOWN, UP, and RIGHT respectively.</p> <p>(DELETE)"CAPS SHIFT"+0 step the the CURSOR backward, erasing the LAST character.</p> <p>(UNDERLINE)"SYMBOL SHIFT"+0 toggles the UNDERSCORE ON/OFF.</p> <p>(INVERSE)"CAPS SHIFT"+4 toggles inverse video ON/OFF.</p>
---	--	---

<<< EDITOR FUNCTIONS >>>

<p>(OVER)"CAPS SHIFT"+3 toggles the OVER type function ON/OFF.</p> <p>(SEGMENT SHIFT)"SYMBOL SHIFT"+(YARD), (D)(STEP), (S)(NOT), or (U)(OR) shifts sections of the doodle pad (screen) LEFT, DOWN, UP, or RIGHT respectively.</p> <p>(CAPS ON/OFF)"CAPS SHIFT"+2 toggles upper & lower case characters.</p> <p>(EDIT EXIT)"CAPS SHIFT"+1 exits the EDIT mode & returns to the MAIN MENU.</p>	<p>(CURSOR CONTROL)"CAPS SHIFT"+5, 6, 7, or 8 keys to move CURSOR character position LEFT, DOWN, UP, & RIGHT respectively.</p> <p>(CURSOR SHIFT)"SYMBOL SHIFT"+(0)=, (F)(0), (G)(THEN), E(=) to SHIFT the CURSOR 1 pixel LEFT, DOWN, UP, and RIGHT respectively.</p> <p>(DELETE)"CAPS SHIFT"+0 step the the CURSOR backward, erasing the LAST character.</p> <p>(UNDERLINE)"SYMBOL SHIFT"+0 toggles the UNDERSCORE ON/OFF.</p> <p>(INVERSE)"CAPS SHIFT"+4 toggles inverse video ON/OFF.</p>	<p>(CURSOR CONTROL)"CAPS SHIFT"+5, 6, 7, or 8 keys to move CURSOR character position LEFT, DOWN, UP, & RIGHT respectively.</p> <p>(CURSOR SHIFT)"SYMBOL SHIFT"+(0)=, (F)(0), (G)(THEN), E(=) to SHIFT the CURSOR 1 pixel LEFT, DOWN, UP, and RIGHT respectively.</p> <p>(DELETE)"CAPS SHIFT"+0 step the the CURSOR backward, erasing the LAST character.</p> <p>(UNDERLINE)"SYMBOL SHIFT"+0 toggles the UNDERSCORE ON/OFF.</p> <p>(INVERSE)"CAPS SHIFT"+4 toggles inverse video ON/OFF.</p>
--	---	---

EDIT MODE EXIT	CAPS LOCK ON / OFF	OVER ON / OFF	INVERSE VIDEO ON / OFF	CURSOR CONTROL ←	CURSOR CONTROL →	CURSOR CONTROL ↕	DELETE & UNDERLINE
CS+1	CS+2	CS+3	CS+4	CS+5	CS+6	CS+7	CS+8
CUR SHIFT	CUR SHIFT	CUR SHIFT	CUR SHIFT	CUR SHIFT	CUR SHIFT	CUR SHIFT	CUR SHIFT
BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+
SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT
BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+
SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT
BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+
SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT
BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+	BY PIXEL+
SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT	SEG SHIFT

by Jeff Taylor

Have you ever saved a really colourful screen file, then tried to print it and been dismayed to see that the attributes had been stripped off?

Have you ever wondered how you could get all of the file printed?

Well, wonder no more. John McMichael, known for his innovative interfacing of 2068s to a variety of printers and plotters, has created a program which converts the file to greyscales for printing.

COLOR2GREY is a very user-friendly program which offers you the choice of editing the greyscale palette to suit your taste and to best set off the various attributes. Upon loading, the user is asked to load a screen file. Once this is done, the conversion process, based on the current greyscale settings, commences. You'll be amazed as the program runs through the sequence of colours, stripping off the attributes and storing them in a greyscale file.

After the conversion is complete, you are presented with the main menu which allows you to view the new greyscale screen, save it, copy it to the 2040 printer, go into the greyscale editor, etc.

If you choose to enter the editor, you are shown a palette of textures which correspond to the colours. If you wish to change any of these textures you pick the appropriate number and are then allowed to alter the pixel density by adding or deleting dots from a grid. You continue until you are satisfied with your new palette then return to the main menu and try the conversion process again. If you are not pleased with the results edit the palette again.

It's that easy.

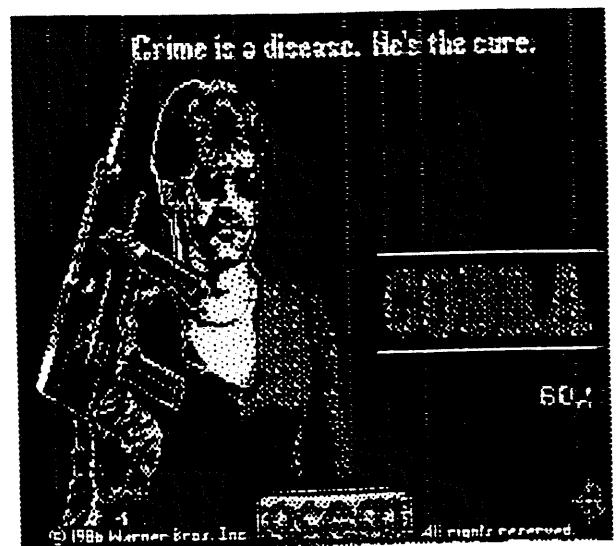
COLOR2GREY can produce some very interesting results and is a great tool for learning how attributes are used. At any time during the conversion process you can break in to view a particular attribute then continue to the end of the process. An excellent tool for anyone interested in colour graphics.

COLOR2GREY comes on cassette and is easily converted to disk by anyone competent in BASIC.

COLOR2GREY is available exclusively from Ed Grey Enterprises, P.O. Box 2186, Inglewood, CA 90305. \$12.95 US + shipping.

Typical Screen Dump

Crime is a disease. He's the cure.



COLOR2GREY With Resident Greyscale



COLOR2GREY (U 1.3)

```

<A> ____ View greyscale screen$.
           (Hold <A> key to view.)
<B> ____ View color screen$.
           (Hold <B> key to view.)

<C> ____ Save greyscale screen$.
<D> ____ Save color screen$.
<E> ____ Save greyscale data.
<F> ____ Load greyscale data.

<G> ____ Copy greyscale to 2040.

<H> ____ Greyscale editor.

<I> ____ Do another conversion.

<J> ____ Quit to BASIC.

```

Main Menu

-----GREYSCALE EDITOR-----
copyright '89-John McMichael

```

<E>dd point
<E>elete point
<E>lear grid
<E>eep g_scale
<E>return to main menu

```

CURRENT GREYSCALE TEXTURES

BLK	BLU	RED	MAG	GRN	CYN	YEL	WHT
2	1	2	3	4	5	6	7

Greyscale Menu

A New Computer Use

To: All Employees

POLICY STATEMENT: In the past, you were permitted to make trips to the restroom under informal guidelines. Effective June 1, 1990, a Restroom Trip Policy (RTP) will be established to provide a method of accounting for each employee's restroom time and ensuring equal treatment to employees.

PROCEDURE: Under this policy, a "Restroom Trip Bank" (RTB) will be established for each employee. The 1st day of each month, employees will be given a Restroom Trip Credit (RTC) of 20. Restroom trip credits can be accumulated from month to month.

Currently, the entrances to all restrooms are being equipped with personnel Id. stations and computer linked voice print recognition. During the next weeks, each employee must provide two copies of voice prints (one normal, one under stress) to Personnel. The voice print recognition stations will be operational, but not restrictive, for the month of June; employees should acquaint themselves with the stations during that period.

If an employee's Restroom Trip Bank balance reaches zero, the doors to all restrooms will not unlock for that employee's voice until the first of the next month.

In addition, all restroom stalls are being equipped with timed paper roll retractors. If the stall is occupied for more than three minutes, an alarm will sound. Thirty seconds after the alarm sounds, the roll of paper in the stall will retract, the toilet will flush and the stall door will open.

from SMUG

Sinclair Milwaukee User Group

*** ARCHIVE FIELDS ***

How to RENAME or CHANGE or ADD a FIELD

by Hugh H. Howie.

One of the problems when creating a file in Archive, is getting your fields right the first time. But there is a little trick which can go a long way to solving this problem.

Say you have created a file and entered many records, then decide you must have another field, or you do not like the title of one of the fields you do have, you have few alternatives, as Archive does not make provision for changing fields after the 'endcreate' command has been used, and you have started the entering of records. This means that you have to scrap everything you have done, and start all over again, and I can hear the muttering and mumbling from here. I know, it has happened to me, and I had 84 records entered, all with 7 to 10 fields. I started to create another file and got tired of it, left it, and it was a long time before I got back to it. All I wanted to do was insert a couple fields in the middle of the file, and change the name of some.

When I finally did return to the chore, which it had now become, I had a better idea of what to do, as I had been playing with Abacus quite a bit and exporting some data to Easel, using Taskmaster. One time I got a bit careless and without thinking I imported a file to Archive instead of Easel. Lo and behold, the file came across, and then the brain started to work, (Hercule Poirot would have been proud of me)

If a file could be imported to Archive from Abacus, what happened if I exported a file FROM Archive to Abacus? I tried it and it came across to Abacus, I played around some more, made some alterations in Abacus, sent it back to Archive, and my file was just what I wanted. And I did not have to re-do the whole thing. Here is how it is done.

Load your file in Archive, by the use of 'LOOK', and then make an export file of it,
export "title" [enter]

Load in Abacus, type:-

F3 'Files 'I'port
'C'olumns (The default is Rows) **title [enter]**

The reason for using columns instead of rows is because the fields will be in columns in Abacus. Your file should now be in Abacus with the labels across the top in row #1, and all the info down the columns.

Now to put in a couple of columns, which is what is required, there are two ways to do it. You can place the

cursor on the cell after where the new column has to go, or, you can change the default as it is presented on screen. We will use the first method, and place the cursor on the column AFTER where the new one has to go. Just watch the prompts and defaults, altering as necessary, and you will have no trouble, the program is very user friendly. Don't be afraid of it!

F3 'G'rid 'I'nsert 'C'olumns (default is Rows)
of columns (default is 1 alter to suit)

To give the new column a name (field) place the cursor on the cell where name has to go, type: **"Name [enter]**

The next step is to RENAME some of the fields, and this is done by placing the cursor on the name to be changed and type **"Name [enter]**. Make any changes to the labels, (fields), you have a mind to. But be cautious as you may create some trouble for yourself later when you are back in Archive, as if the position of a label is changed, the label may not have the correct data. (This will mean extra work in Archive, but not insurmountable.)

When you have made your corrections, save this as an _aba file in case you want to come back to it. You will also have to make an _exp file of the new setup. Once again use COLUMNS not rows. **CAUTION:-** A new name must be found for the file exported from Abacus, such as add a '1' to it, otherwise you would have two files with the title_exp. One Archive, the other Abacus. (Not possible)

At this point, (presuming you used a disk in #2 with only the File you wanted to alter) it would be a good idea to look at the directory of the disk, it should look something like this:-

Title_dbf	data file in ARCHIVE
Title_exp	data file exported
Title_aba	amended file in ABACUS
Title_exp	amended file to go to ARCHIVE

Go back to Archive,

'I'port "title" "title" [enter]

That should be it. You will soon know if you have made an error. Probably the most likely error is if you changed the *position* of a Field in Abacus; this being so, just use **Alter** and change the data around in Archive.

Save the new _dbf file under the new name. Of course if TK2 or Trumpcard is available, then the RENAME command will set everytng straight. Else you have a new name.

Now if this should appear to be a lot of work, then think about the labour of creating a completely new file, and typing all that stuff back in. I think this is easier.

555

More on RESPRing from Nfld.

There is a problem with Réal Gagnon's two line method of CALLing up a machine code extension (Sink-Link Nov - Dec '90). RESPR is a bit more complicated than he implies, and in many cases his method will result in a CRASH since it may CALL the wrong address. Try the following on your QL:

```
100 CLS
110 a=RESPR(0)
120 b=RESPR(193) : PRINT a-b
130 c=RESPR(0) : PRINT a-c
```

You will get the two numbers 208 and 512! (Not 193 and 193 as you might expect.) The first, (a-b), is the amount of space reserved for your code, 208 bytes not the 193 you asked for. This is all right since QDOS always RESPRs an exact multiple of 16 bytes giving the next one greater than your request: there is always enough room, and you know exactly where it starts. However on the larger scale QDOS manages memory in blocks of 512 bytes, so if your request is not a multiple of 512, the rest of that block is wasted, and the next RESPR starts at the next whole empty block; hence the value of 512 for (a-c). Thus RESPR(0) will only be the right address to CALL if the amount of memory reserved is an exact multiple of 512 bytes.

A further implication of this is that if you want to install several extensions there could be quite a bit of wasted space if you RESPR (or LRESPR with TKII) each one separately. It is more efficient to write a few lines in your boot to RESPR the total in one go and then LBYTE and CALL each routine at the appropriate place within this area of memory. The following PROCEDURE would do the trick:

```
500 REMark -----
502 DEFine PROCEDURE Load_Extns(n)
504 LOCAL a,c$,d$,i,n$(n,10),b(n),s: s=0
506 RESTORE 528
508 d$="mdv1_": REMark <<< Alter to suit
510 c$="_code": REMark <<< your system.
512 FOR i=1 TO n: READ n$(i),b(i)
514 FOR i=1 TO n: s = s+b(i)
516 a=RESPR(s)
518 FOR i = 1 TO n
520 LBYTES d$&n$(i)&c$,a: CALL a
522 PRINT n$(i): a = a + b(i)
524 END FOR i: END DEFine
526 REMark -----DATA
528 DATA "replace",168,"ramdisk",2738
530 DATA "edlines",350
```

Insert the statement Load_Extns(3) in the main part of your boot and all will be looked after. You have to tell it how many extensions there are, so make sure there is enough DATA unless you are interested in learning a few more error messages. Too much means that only the first n will be loaded, but won't generate any errors. The names and lengths of the files containing the machine code are given as DATA statements after the PROCEDURE. As written the procedure assumes that there are the programs: replace_code, ramdisk_code and edlines_code on mdv1_. You will have to adapt these to suit your own needs.

LOCAL arrays n\$() and b() are dimensioned (504) according to the value of the parameter n, and the DATA read into them (512). The total number of bytes required is determined (514) and RESPRed, and the files LBYTEd and CALLED in the final loop (520-524). The name of each is printed on the screen as soon as it is active. If you need more than 10 characters per name then increase the second parameter for n\$ in line 504.

Howard Clase,
Box 9947, Station B,
St. John's, Nfld,
Canada A1A 4L4.

(709) 753-6415

e-mail hclase@kean.ucc.mun.ca



Dear George,

".... I do have something new for you. You or one of the other writers in the last issue mention the lack of any sort of Tasword compressor, like all the PC's have and such. Well, that is easy to fix! Here, for everyone's use is what I could only call TasPak, a Tasword file compressor and de-compressor.

Let me back up a little bit first. One obvious way to compress Tasword files is to use TasMS, which for most files will make them smaller. It has two problems, though. First, it doesn't always make it smaller - a file with no spaces, if you can imagine such a thing, becomes 1.5% larger! Second, it isn't completely reversible. I have written a program to convert Mscript files to Tasword files, but in certain cases the spacing is different in the result.

Still, the basic idea is good. If there are spaces at the end of a line, take them out and replace them with some special character. While we're at it, why not convert indents and tables to some sort of TAB character? Actually, there are 2 different ways I could have done this. I chose a tab like on a typewriter as opposed to a TAB(x) like in Basic.

Both of these are good, but they only apply to space. What about something that is mostly characters? Well, I follow what happens in PC compression programs - I do read BYTE, after all. So I can tell you what they do. They take groups of characters, and replace them with a single symbol. But they do so in a fashion that depends on the file. That way, a file of all numbers is compressed efficiently, while a file with no numbers is also compressed.

Now, I don't want to mess around with variable length strings and all that stuff, but I do have one advantage. I know that Tasword uses characters from space to GRAPHICS SHIFT 8, whereas a PC word processor might use others. (In a PC, the other characters are things like ð and ó or Greek letters). In other words, Tasword only has 112 possible characters, and I can use the others for pairs of characters. Since I use one for a tab, and one for an end-of-line, that means I have 142 characters to use for pairs of letters.

Now, I was able to use all this in a way that guarantees the result can never be longer, and that is completely reversible. In other words, the result of compressing and then uncompressing is indistinguishable from the original, regardless of what the original might have been. As a further interesting sidenote, if you try to uncompress a file which was not compressed, nothing will change. Note that trying to compress a compressed file will produce one which couldn't be interpreted by the un-compressor.

I should mention a possibility that I didn't use. Since Tasword only uses 112 characters, it would be possible to use a bit compression routine - make every 7 bytes represent 8 characters. This would always compress things by 12.5%. That isn't bad, but it could be better. I can reasonably say that an average file should be compressed about 40% by my program. Files with a lot of spaces will do even better. The bad news is that there could be a file that achieves no compression at all. I can't imagine that it would be anything meaningful, but who knows?

Of course this routine has been specifically set up for Tasword. Not knowing about things like printer control codes for Mscript, all I can say is, don't use it for that.

There is one problem with TasPak. The buffers overlap just a little in memory. That means that if you have a file that starts with 200 blank lines, it won't uncompress properly. Under presumption that it will only be used on reasonable files, it should never run into problems, but just so you are aware of that limitation. I already said that TasPak was strictly limited to Tasword files, and not to try to compress something twice. Of course, you can always use my Mscript to Tasword converter and then compress - as long as your Mscript file has no printer control characters.

Sincerely,
Steven Gunhouse

BASIC FOR "TasPak.Bz"

```

100 CLS : PRINT AT 3,8;"Tasword
Utility:";AT 4,8;"File Compress
ion";AT 6,5;"by Steven V. Gunhou
se"
110 PLOT 60,132: DRAW 0,23: DRA
W 159,0: DRAW 0,-39: DRAW -183,0
: DRAW 0,39: DRAW 159,0: DRAW 0,
-23: DRAW -135,0
120 RANDOMIZE USR 32768
130 PRINT AT 18,0;"File to Load
?"'"(ENTER to see CAT)"
140 INPUT "File? "; LINE l$
150 IF l$="" THEN CLS : PRINT
#4: CAT ".CT": GO TO 140
160 FOR i=1 TO LEN l$: IF l$(i)
<>"." THEN NEXT i: LET l$=l$+"
CT"
170 IF LEN l$<>1+2 THEN GO TO
140
175 IF l$(i+1)<>"C" THEN GO TO
140
180 IF LEN l$>9 THEN GO TO 140
200 PRINT "Loading ";l$
210 POKE 23728,100
211 PRINT #4: LOAD l$CODE 45056
212 IF PEEK 23728=101 THEN PAU
SE 30: GO TO 100
220 CLS : PRINT "File ";l$;" lo
aded."
230 PRINT AT 8,0;"Options:"
240 PRINT "1. Compress file."
"2. Uncompress file."
250 PRINT "Your choice? "
260 PAUSE 0: LET a$=INKEY$: IF
a$<"1" OR a$>"2" THEN GO TO 260
270 PRINT AT 13,13;a$
300 IF a$="1" THEN LET len=USR
32808
310 IF a$="2" THEN LET len=USR
32928
350 PRINT "'Save new file as?"
'"(ENTER for same name)"
360 INPUT "Name? "; LINE s$
370 IF s$="" THEN LET s$=l$
380 FOR i=1 TO LEN s$: IF s$(i)
<>"." THEN NEXT i: LET s$=s$+"
CT"
390 IF 1+2<>LEN s$ OR LEN s$>9
THEN GO TO 360
400 IF s$(i+1)<>"C" THEN GO TO
360
410 PRINT "Saving ";s$;" , leng
th=";len
420 PRINT #4: SAVE s$CODE 33280
,len
430 INPUT "Another file? (Y/N)
";a$
440 IF a$="" THEN GO TO 100
450 IF a$(1)="Y" OR a$(1)="Y" T
HEN GO TO 100

8990 STOP
9800 RANDOMIZE USR 100: OPEN #4,
"dd"
9810 CLEAR 32767
9820 PRINT #4: LOAD "TasPak.Cz"C
ODE
9830 RUN
9900 PRINT #4: SAVE "TasPak.Bz"
LINE 9800
9910 PRINT #4: SAVE "TasPak.Cz"C
ODE 32768,410

```

M/C LOADER FOR "TasPak.Cz"

```

2 CLEAR 32767: LET x1=32767
3 CLS : PRINT AT 10,5;"Loadin
g M/L ..."
4 LET x=1: LET cs=0: LET a=10
: LET b=11: LET c=12: LET d=13:
LET e=14: LET f=15
5 READ a$: IF a$="END" THEN
GO TO 10
6 FOR n=1 TO LEN a$ STEP 2: L
ET w=16*VAL a$(n)+VAL a$(n+1): P
OKE x+x1,w: LET x=x+1: LET cs=cs
+w*x: NEXT n
7 GO TO 5
10 IF cs<>8896655 THEN PRINT
"Error in DATA!": STOP
15 RANDOMIZE USR 100: SAVE "Ta
sPak.Cz"CODE x1,x+1
9949 DATA "2100B006000E4D3E"
9950 DATA "20772310FC0D20F9"
9951 DATA "0E02067036002310"
9952 DATA "FB0D20F83E90068E"
9953 DATA "77233C10FB3680C9"
9954 DATA "CD7E81230600041A"
9955 DATA "13FE202023E52A00"
9956 DATA "FFA7ED52E138587B"
9957 DATA "E63F200B3E1E2BBE"
9958 DATA "28FC23361F18DCCD"
9959 DATA "6C8138DA361E18D3"
9960 DATA "F505280F78362023"
9961 DATA "10FBFE0338057B3D"
9962 DATA "CD5081F1471AFE20"
9963 DATA "20037018B6E521FF"
9964 DATA "FC0E8F0C237E23A7"
9965 DATA "280FB820F61ABE20"
9966 DATA "F2E17113CD378118"
9967 DATA "9AE1CD1A8118F511"
9968 DATA "0082A7ED52444DC9"
9969 DATA "CD7E81230600041A"
9970 DATA "13FE1E200B362023"
9971 DATA "7DCD6C8138F718EC"
9972 DATA "FE1F200A3620237D"
9973 DATA "E63F20F818DEFE20"
9974 DATA "200CE52A00FFA7ED"
9975 DATA "52E138C318D0F505"
9976 DATA "280E7836202310FB"
9977 DATA "FE0338047DCD5081"
9978 DATA "F1FE1E3804FE9038"
9979 DATA "1B4FD690D5E52100"
9980 DATA "FD16005F1919EBE1"
9981 DATA "1A7713231A77D1CD"
9982 DATA "37811897471AFE21"
9983 DATA "300370188ECD1A81"
9984 DATA "18ED70231A7713D5"
9985 DATA "E52100FD3A70FE4F"
9986 DATA "D6905F16001919D1"
9987 DATA "70231A77EBD1C9D5"
9988 DATA "E52170FE545D7E23"
9989 DATA "B928FB1213FE8020"
9990 DATA "F5121B7912E1D1C9"
9991 DATA "E63F4FD5E50C2120"

9992 DATA "FE545D7E23B928FB"
9993 DATA "1213A720F60D1B79"
9994 DATA "12E1D1C9E63F4FD5"
9995 DATA "1120FE1A13B92804"
9996 DATA "A720F837D1C92100"
9997 DATA "FD3E202BBE28FC22"
9998 DATA "00FF1100B021FF81"
9999 DATA "C9","END"

```

TORONTO TIMEX-SINCLAIR USERS CLUB
November 27, 1990

14 Richome Court,
Scarborough, Ont. M1K 2Y1

Les Cottrell
108 River Heights Drive
Cocoa, FL 32922

Dear Les,

Bob Mitchell has been working on the rotating of character sets. He has produced a disk for the library. I am going to put it into a collection of disks under the library disk number 37. I enclose a copy in this package since I think you would be interested.

Re the computers. I have one left. It is the one that was offered for \$50. I am offering it for \$35, if you are interested. I think I mentioned that while the keyboard works properly, and looks in perfect condition, there is cutout on the rear of the upper case in the vicinity of the metal RF box. The cassette door seems to be slightly dodgy, and maybe more importantly, there does not seem to be any Monitor output. The TV output is OK, though I don't like TV colours, never did! But the colours were proper. I'm puzzled by the lack of a monitor output. Both computers seemed to suffer from that, which puzzles me. I don't think it was my monitor out of adjustment.

I mention all this so that there won't be any sense of disappointment. It isn't my product, I'm only someone doing a good turn, and it wouldn't be a good turn to you if I did not mention these items. I loaded a huge program into the computer via tape, and it worked properly. I could find nothing wrong with the memory. I hooked up a Larken Disk drive to it, and it worked properly. It also saves to tape as well as loads from tape. The cartridge door works properly, but the spring tension seems weak; I mentioned "dodgy".

The computer has a "reset" switch installed.

The 2040 printers have both gone, as has the 2050 modem.

Bob mentioned that the keyboard matrix that we put in the last newsletter was "shrunk". I was thinking that we should run a series of them in our newsletter. Seemed like an interesting idea. But it looks as if we should get an "enlarged" copy for Jeff to use, once we have determined how much his copier shrinks them. Our local library has a Xerox copier that can enlarge/reduce, and you choose the degree of change in percentage points.

The character rotator that Bob used came from the program SuperCode 3.5. Are you familiar with it? It contains 152 m/c routines to do all kinds of things. Well, many of them are fancy scroll routines, but there are some pretty clever ones besides that. I have used one of them, and "ON ERROR GOTO" routine for the Spectrum, that I use in the club Spectrum disks to return to the menu while viewing a Help file.

I have received two exchange club newsletters recently, which show signs of deterioration. The Indiana TSUG newsletter has 12 pages. An illustrated parts list of the QL kit that, was it Sharps, sent out with their \$99 offer of QL kits some months ago, took eight pages. Then two pages were given to a Z88 advert, one page to a Bytepower advert. There wasn't much left.

The Sinclair Milwaukee group's newsletter did not have any material on the Timex computers. There was a lengthy article by Dr Dreger about

Artificial Intelligence. And there were adverts for Timex stuff by EMSOFT, and one from RMG Enterprises, and a half-page humorous story about a man lowering a bucket from a roof. And a notice that they were going bi-monthly because of a dearth of news and member contributions.

Makes one wonder what we are doing right, doesn't it?

I'll allow that The Indiana one was put out by Frank Davis, and I'm sure that he has had his hands full getting out the first issue of UPDATE. I received a complimentary copy of UPDATE, with a note asking if maybe I had overlooked renewing my subscription. I hadn't, because I did not plan on renewing. I had become disenchanted with the contents of UPDATE. UPDATE, a magazine that was started on the premise of serving disk system owners. It never lived up to that promise, as far as I am concerned. In the beginning only Oliger's system seemed to rate. Then, as soon as Bill Jones discovered the QL, that's all the magazine seemed to have after that. And the current issue does not seem to have changed much. Well, there was a lot less of Bill's extravagant programming exercises in it, what a relief!

I shouldn't complain, really. Frank Davis gave us a nice report in this issue of UPDATE! Really, quite flattering. I shall expect a few new members to join after that. I have mixed feelings about that. Especially after just getting off newsletters to 75 OOT members! Enough, really.

Our Larken disk library continues to grow. We are at about disk #38, with one or two in the wings, that I have not had a chance to polish up yet.

RMG Enterprises sent our club a copy of their catalogue the other day. It has 50 pages on it; full of interesting stuff.

One of my Quad drives quit on me. I think I mentioned it in the last newsletter. When I look around in Computer Shopper, for example, I see where the surplus Tandon Quad 5 1/4 inch drives have all been taken up; there's no more being offered in the surplus stores. Interesting. Now they seem to have a surplus of 3 1/2 inch Quad drives!!

Well, I think I shall close this off, maybe I'll think of something else by the time this has printed out.

Sincerely,

George Chambers.

Nov/Dec 1990
November 12th, 1990

Dear Out Of town Members,

I ran off the labels for this issue of the newsletter, and I was quite surprised to see that there are 75 out-of-town members. Our normal complement has been 55 or so. So that's why my work load seems to have increased in past months. I thought I was losing my grip or something. Well, I think that I am starting to lose my grip, but at least I know why!!

I'm not sure just what the reason for the increased membership is. Maybe it had something to do with inactive memberships being carried along with the 6-month extension we made. On the other hand maybe we are becoming popular.

In the last issue there was an article about multiplication accuracy. Steven Gunhouse, a member, picked up a couple of errors in the article and sent in corrections. I prepared an article on it for this issue, but the Editor did not get it in. So I shall quote a bit from Steven's letter, to put you out of pain, all of you who could not get it to work. Edited partly, it reads:

"...I note several typos and other errors in your "Multiplication Accuracy" program. LINE 70 should be N instead of M; LINE 90 should be "LET B = VAL X\$(M)"; LINE 110 should read "LET I = INT (A(C)/10)". Those are probabaly all typos, but there should be an extra line that reads "185 IF A(1) O THEN PRINT A(1);" Otherwise the first digit of something like 99*99 won't be printed."

The complete article will appear in the next issue.

Another correction which holders of the PPPProf disk might like to be aware of. On the "BOOT.B1" program LINE 9410, you should make these changes:

```
"PRINT #F: GOTO SGN PI"....make it  
"PRINT #F: GOTO ddrv"  
"PRINT #F: GOTO NOT PI"....make it  
"PRINT #F: GOTO mdrv"
```

This ties in with LINE 100 in the program where the user can change the disk to suit the drive assignments of your particular system.

Member Ronald Ginardi points out the need of this change, saying that "Since there is an ON ERROR involved, you cannot break out of a LOAD from a disk drive that you do not have."

One of the reasons I have been busy is because I have become the local Neighborhood Watch chairman. The

organisation was in need of some rejuvenation, so I have been busy contacting Block Captains, calling meetings, etc. It all takes some time. My office(?) is a mess, what with computer junk all around, and then NW stuff on top of that.

I had some film left over, and I photographed my office(?). My thought was that I might put a copy on the top of this letter. But when I got the prints back I found so much junk in the background (and the foreground) that I decided to cancel that idea. So many cables and bits of paper, disks, tapes, and I don't know what all, jumped out of the picture! I understand why people, some persons, don't like thir picture taken without primping up!! Enough of that.

A couple of our members report that they have not seen the October issue of UPDATE. Though I read about a month ago in another club's newsletter, that the issue was out. Anybody else awaiting thier copy?

On the first Wednesday of October, while we were having our monthly club meeting, our local public broadcasting station, "TV Ontario", put out a program called "Anatomy of an Inventor". It featured Clive Sinclair. It is one program in a series called ODYSSEY. My wife saw it. I wrote TVO to see if it would be shown again. They said yes, in about 6 months. So I shall try to get a copy of it on video tape, so that any members can borrow it, if they care. I see from another club's newsletter that it did not deal in much depth with his computer experiences. It is a BBC series.

Several of you have taken me up on Spectrum games. I see in the newsletter that Jeff, our editor has reviewed "King's Keep", and seems to have enjoyed it. If any of you wish to know where to find a copy, contact me. When Jeff was hung up on it and asked me if I had any clues, I said yes, but you're only going to get one or two of them at a time!

I wrote a program called "bopeep" which enables me to PEEK and POKE changes to individual addresses in programs which are on disk. Very useful to POKE infinite lives, etc., into games. But I found that if it was made too easy, there was no challenge, and therefore no fun left in the game!!

More recently, and with the aid of S. Gunhouse, I have written a program which enables me to save an NMI-type program to tape. I call it "nmisav" and there's a companion program called "nmilod". I have made up a disk with these programs on it, along with instructions on their use. Anyone interested?

In my last missive I promised an article on what is new in the Larken library. But, as I mentioned earlier, I have been too busy to get this done. But I see where we are up to disk #38. This disk has an assortment of programs assembled by Bob Mitchell, and called MATH, Science & Education. There are still some spare tracks, and I think we shall find some other interesting programs to add to it.

Then there's disk #37. This disk, or suite of disks will hold data files for the PPP+ and PPPProf. programs That are on disks #10 and #28. Disk #37 will have the files for creating TS2068 keyboard matrixes using Pixel Print. Was there an article on this in one of our newsletters, or is it still to come? I can't recall.

Disk 36 is a ShareWare disk. It contains a variety of programmers utilities by McBrine.

Disk 35 is a set of utilities of interest to club members who have acquired a 24-pin printer. The disk has been prepared by Larry Crawford.

I have an interesting story to tell, about our Post Office. You may recall that we discontinued our P.O Box number, maybe 18-20 months ago. Well, about a month ago we started receiving mail from three clubs, and from two individuals, who had only recently had their mail returned. Guess what. It was things they had mailed over a year ago. One was a series of about 6 newsletters from TIMELINEZ. Another was a person who had sent a cheque for \$20 to join the club. It would seem to me that the Post Office kept the mail from discontinued Box numbers in a back room somewhere, and only recently discovered it. Or else maybe they clean out their bags once a year!! I suppose we should be grateful they did not burn it.

In the last n/l I mentioned that I had some used DSDD drives for \$20, plus postage. Several members took up the offer, and have now received them. Everything goes well with them except a member, Mac Pace, who has a QL. Well, I had to send him another drive because the first one would not read Hugh Howie's QL library disks, and vice versa. It would write, and read it's own disks, but no others. Anyway we are on

the way to solving that problem. I learned something though. The Larken has a better set of features re checking out disks. The Larken has a VERIFY command which tests each track in turn for CRC errors. Also, the directory is on a single track, and one can access it and test it more readily, than seems possible with the QL DOS'. Thanks, Larry.

Since then I have bought one or two more drives (though I said I wouldn't!), so if any one wishes to add to thier system, drop me a line. Keep in mind that you need to add a connector to your ribbon cable to add another drive, as well as make sure your power supply is up to it. The older drives take more power than the newer design of drives.

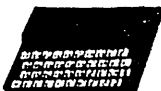
One of my Quad drives stopped working. I have not been able to figure it out. The problem lies in the circuit board. I have two drives and was able to swap boards. The problem went with the board.

So I wrote to JB Technologies to see if they had any Tandon TM101-4 drives for sale. They are not advertising them any more, but I thought that they might still have a few kicking around. I have not heard from them. I also wrote to Doug Hemming, who had them at one time. One of mine came from him. He has not answered either. I may have to buy a used drive from a local surplus store. Though they want \$45 for one, which seems a bit much.

I find myself sort of crippled. Maybe I'm spoiled. But I've had to fall back on Richard Hurd's ODDBALL program to copy my quad disks. ODDBALL, if you are not familiar with it, uses a second drive (of any type) as a buffer, when you want to make a duplicate disk. This program is on our library disk #3, along with a number of neat Larken utiltiy programs.

I've run out of space.
Sincerely,

P.S. No, I still have a bit of space. Just a note to members who may be waiting fo something from me. If you do not get it within a week after this newsletter gets to you, better drop me a line. I may have overlooked a request. I should be up to date by then. There may be some request that I have overlooked though, so don't be afraid to remind me.



-----Sept., 1990-----

2 pages-----

© 1990, In the public domain (this page/side only)

News of a Local Nature - The Ottawa-Hull Timex-Sinclair User Group continues to get together, this last September, 5 members meeting at a members home down by the River (in what was at one time, cottage country). As one member remarked, whether we continue to use Sinclair computers or not, we still get together to enjoy talking to each other. Some members have acquired IBM compatibles, but there is still quite a bit of use of the TS computers among group members, (including those with other makes like the IBM compatible). The club librarian, Dave Solly, 1545 Alta Vista Drive, Apt. 1402, Ottawa, Ont., Canada K1G 3P4 reported that many, even from out of town are using the club BBS (613) 745-8838 (8/n/1, 300 and 1200 baud). Activity in programming in Pascal both on the TS2068 and the IBM PC (using Turbo Pascal) is also a club activity that Dave participates in and the BBS has had on it a Pascal compiler for the TS2068 that has been released into the public domain for non-commercial purposes. Coffee and goodies rounded out the meeting which included discussion of proposals to raise rates on phones using modems, (in the USA), copyright notices/mentions on run-time modules and Mickey Spillane!

Larken Electronics, original designers and makers of the ZX-81/TS1000 LDOS disk controller/DOS and the LKDOS TS2068/Spectrum ROM compatible disk controller, and DOS seem to be turning their attention to the sales of their modestly priced computer controlled shaper/router/plotter that allows the working of plastic and light metal by robot control using an ordinary IBM PC compatible running a program originally written in Pascal on the TS2068. Continued support of the disk systems, at least in terms of being able to buy new boards for new users is now being considered, dependent on demand, as probably anyone who knows about the Larken system and has a TS2068 has got one if they are going to now. No matter what the outcome, conversion of the Aerco, Zebra and Oliger controllers to run the LKDOS operating system and thereby read and write to LKDOS format diskettes (which is now in progress all over the US judging from newsletters that are coming from those user groups there), will guarantee a longer market and opportunity for those who wish to convert to the LKDOS system over time. The software support of the LKDOS system will be handled ably by the Toronto user group, as it is now, and possible further contributions from Larken Electronics. The long-in-preparation book on the LDOS system for the ZX-81/TS1000 is being releases in pre-publication form in chapters or sections by the author instead of going the formal, publishing route (at TS Bulletin amateur publications address above). This unofficial manual for the ZX-81/TS1000 single-density controller board that Larken produced by modifying the software for the original TS2068 LDOS controller has been in preparation for more than a year and has about 110 pages or more. Plans to publish it were abandoned due to the cost of more than \$40, but chapters or full copies have been circulating anyway even though it is not finished.

Congratulations to the Toronto Timex-Sinclair people for producing another magnificent issue of their newsletter, which has been, the last couple of months, the size and heft of a magazine. The Club there is doing a fine job of supporting the TS2068 and LKDOS disk system especially.... The Hacker from Las Vegas never fails to show up with a fine edition, and inspiration to other newsletters in both appearance and amount of good material.... In Canada a postal strike is rumoured, but then that is always the way as Christmas approaches so it is not to be taken seriously yet. More than a rumour is the new G.S.T., a federal retail sales tax in Canada which will be applied even to the postage of newsletters from here. The publisher of this newsletter is seriously considering switching the publications to text files on disk (MS DOS ver. 2.x, 360K, 5 1/4 in) and just posting them to a BBS in order to avoid attracting too much in the way of costs in the future when the G.S.T. comes in. (My TS1000 disk format is not compatible with any other and ill suited for text anyway: LDOS single-density. I have found a way to make a disk LDOS/ZX-81 one-side, MS DOS the other.)

(Further to p.2) Useful Sinclair Programmer Notes (On Telephone Nos.)

In the column "Amateur Programmers' Line" for this issue, mention was made of programming with primitives and simple building block algorithms made as the result of such work. Below is a short listing of one version of a ZX-81/TS1000 program to test a way of compressing the RAM storage space that a 7-digit telephone number requires in a database.

1 REM

Comments

(Here is where we will POKE the telephone no. into, starting at 16514 in the ZX-81/TS1000, fixed position.)

5 REM Telephone numbers storage prog.*@

10 POKE 16525,184

put last 3 digits in RAM

20 POKE 16524,227

put middle 3 digits in RAM

30 POKE 16523,8

put first digit of no. in RAM

40 REM This up to here will POKE the tel. no. into RAM

50 PRINT "TEL.NO. ";

60 FOR X=3 TO 0 STEP -1

This steps 3,2,1,0 -backwards

80 PRINT PEEK(16525-X);

Print on screen the TEL. No.

90 NEXT X

(from RAM)

99 STOP

STOP at end of demo.

*NOTE: THIS IS ONLY A FIRST TEST! (IS NOT USABLE IN A PROGRAM YET! SEE TEXT!)

This will store any seven digit telephone number in 3 bytes of RAM, except (can you see the exceptions...?...think!) The exceptions would be for example a byte with leading zeros, including 822-1000, 800-1001. The program also does not put in the "-" at the appropriate place. One way to solve the leading zero problem would be to assign certain reserved codes for double zeros, but the problems are not too easy to surmount without variable length fields (allowed RAM bytes) for the telephone number. Anyway, theoretically 3 bytes is better than 7 for using the standard alphanumeric code (similar to ASCII in the ZX-81) — one byte for each digit—even if you have to add another byte or two to get the case of leading zeros into your algorithm. Another number that would not fit in 3 bytes but would require only 4 bytes to finalize would be 826-6257 since it would normally be broken down into 8, 266 (too high a number for the 255 limit of one-byte), and 257 (also too high for the 255 limit of one byte). It would have to be broken up into 8, 26, 62, 57 and so require 4 bytes of RAM.

That leads to thought of other ways of handling it. The simplest way numerically and for your conversion routines would be to allow for variable length fields. This would normally create big problems with a database. It also might waste an extra byte or two. An end-of-field byte marker would of course have to be a different number (of the 0 to 255 that it is possible to store in one byte) than would ever occur as a code. If only one telephone number per record (for example, a person, member of a club in a club membership database), is used and there are other variable length fields, marked in ingenious ways for endings, or length, placing the telephone number at the end of the record would allow its end to be marked by the record end marker. So we may have to reserve two codes not one to never use for the tel.no. bytes, end-of-no. (253) and end-of-rec. (254).

This is starting more and more to look like a typical coding problem, that of inventing a new code that will streamline computer operations, reduce even the space required to store data in RAM, and of course cover all possibilities in a logically consistent manner. The zero case is one problem. Perhaps it could be handled by not being greedy to get that extra digit (under 255)*@ in and settling for 2 digits per byte, with 0 coded as one zero and 200 coded as two zeros, the 2 being wiped out by the translation or display algorithm. 100 could remain as three digits, 1-00.

What if there is no phone number given or available? How about a code other than 000-0000 for that? In other words, an error code, non-exist code or a mathematically conceived empty set (null set) member code? These possibilities for coding "nothing" are often overlooked in coding design. We could assign 252 as this code for this system, where legitimate codes beyond 200 (or 252 to push it to the maximum) would not print or be interpreted as phone numbers. Then there are long distance numbers....

@* OR 253 if 252 = end-of-no. 254 end of record, 255 EOF

© 1990, In The Public Domain

Newsletters & Clubs: The Vancouver TSUG (or rather VSUG) is having a little difficulty in getting material for the newsletter (hint for those who would like to send them something, including a program on cassette for listing). It will be prepared by a new editor as of the next issue. It has been a good source of hardware articles and machine code articles for the ZX-81 in particular for the last couple of years. Wilf Rigter, the originator of a system of high resolution graphics for the ZX-81 has written articles for it on how to fiddle with the operating system parameters to get specialized graphics and multi-tasking (the NOVA 1000 program) with the ZX-81. General news and material on the TS2068 have also graced its pages, as well as the schematics for many projects, especially those for the ZX-81 which most folk would have less trepidation in taking apart and hacking..... Larry Kenny of Larken Electronics has put out a call for all who wish to order LKDOS interfaces to speak now or hold their peace (forever?), so that he can judge by orders to RMG Enterprises or himself whether there is enough demand to get another, minimum order of at least 35 p.c. boards manufactured (by an outside company specializing in pc board etching) in order to make up more LKDOS controller boards for the TS2068/Spectrum ROM TS2068. Larken Electronics is at RR#2, Navan, Ont. Canada K4B 1H9 and R.M.G. is at 1419 7th St., Oregon City, OR, USA 97045 (tel. 503-655-7484).....A few of our readers may have heard of the excellent newsletter for the QL put out in French by Real Gagnon. He has written an article for the Toronto TSUG (or rather Club) that helps explain his suspension of publication (temporarily?) of that newsletter, QL_DOC, due to job problems in the winter when his employer went out of business. We can hope that time and finances will permit its re-establishment soon.....The Johnson City, NY (a high tech centre) TSUG, SINCUS has moved to embrace the IBM PC compatible world and also to discontinue publication of its newsletter as of this summer. If you are not getting any more newsletters from them, that is the reason.....Update Magazine, a T/S based quarterly supporting aftermarket disk systems, especially those for the TS2068 has changed hands. The previous publisher, Mr. Jones is taking a holiday and going cruising around in the Florida area ocean and Frank Davis of Indiana TSUG is taking it over. Expectations of further excellent issues coming forth have been expressed by people who know Frank. For subscriptions contact at tel. (317)473-8031 or via ISTUG, 513 E.Main., Peru, IN 46970.EMsoft, the QL software firm at P.O. Box 8763, Boston, MA, USA 02114-8763, tel 617-889-0830 is offering a free catalogue and also has some things other than QL software including a ZX-81/TS1000/TS1500/TS2068 expansion bus connector and housing (also works with PC8300 but with TS2068 connects only to a subset of the bus connections, like the TS printers) for US\$16 for 3 (or two for the same price) and a collection of documentation for the ZX/TS computers, Syntax, etc. for \$30. (from an advert in SMUG, Sept.1990).....If you are wondering about how old your addresses are for a number of groups etc., and whether they are any good, look at the following new addresses: Toronto TSUG (changed last year), c/o Geo. Chambers, 14 Richome Ct., Scarborough, Ont., Canada M1K 2Y1 (subscription to n/1 \$12-m.o).... SINCUS, Johnson City-discontinued N/L, Harrisburg, PA gp. folded, Dallas & Ft. Worth Gps., discontinued n/1 but last heard they were attempting to start one up again, QL_DOC, Montreal, suspended publ., contact Real Gagnon via Toronto TSUG,.... TS SIG of Boston Computer Soc. folded but reincarnated as New England QL Users' Gp. P.O. Box 8763, Boston, MA, USA 02114-8763 Sum.1990...CRAGIST n/1 renamed and now published from new address of Donald Lambert, 1301 Kiblinger Pl., Auburn, IN, USA 46706 (Sept.1990 chg. of address), CATS (National Capital TSUG), new address, (fall 1990), CATS, POB 11017, Takoma Park, MD, USA 20913.....Quanta Library, c/o Paul Holmgren, 5231 Wilton Wood Ct., Indianapolis, IN, USA 46254 -program library/ source code.... TS Bulletin Amateur Publications still going strong (in 4th year of publication), with many back issues still available for postage 97 Ruskin St., Ottawa, Can. K1Y4B3

Suppliers: American Micro Systems, 2175 Aborn Rd., #262, San Jose, CA, USA 95121 -QL,... John McMichael, 1710 Palmer Dr., Laramie, WY, USA 82070-TS2068 printer drivers-Ok!.... Byte Power-TS2068-software & mag., 1748 Meadowview, Pickering, Ont., Canada, L1V 3G8.....Indiv. Users with a Spec. Interest-TS2068/MS DOS Pascal, D.Solly, 1545 Alta Vista #1402, Ottawa, K1G 3M4-TS2068 music, Joan Kealy, POB 1118, Brackettville, TX USA 78832, hardware/ic tester w. TS2068, N. Pashtoon c/o CATS, & Bob Swoger, DOS/OS extensions for TS2068/LKDOS-CATS, 613 Parkside Circle, Streamwood, IL, USA 60107....