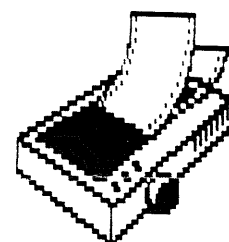
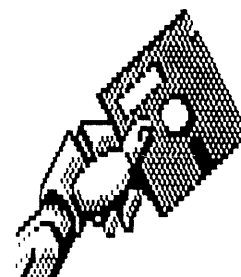


SINC - LINK



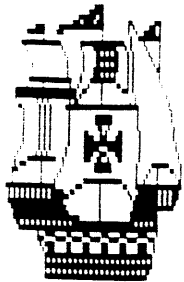
MAY-JUNE '92 VOL 10 #3



SUPER SUMMER 1992 ISSUE

page 2	Club Stuff	page 26	Locked Out? How to Recover (2068)
page 3,5	Bob's Notebook	page 27,30	ZX91 Magazine (ZX81)
page 6,8	Questions & Answers (2068)	page 31	A Tape Index (2068)
page 9,14	Beginners Corner (QL)	page 32,33	Steve Gunhouse Writes (2068)
page 15,16	64K SRAM Memory (ZX81)	page 34	Tool Kit 2 Notes (QL)
page 17	Bill Harmer's News	page 35,36	A Challenge to Programmers
page 18,21	QL Success (QL)	page 37	QLips (QL) & Printer Reset
page 22,23	Britain's Spectrum Scene	page 38	Tight Security (ZX81)
page 24	4x Times Screen\$ (2068)	page 39	Disk Drive Tips
page 25	A O REM Generator (ZX81)	page 40	Graphics Page

TORONTO TIMEX-SINCLAIR USERS CLUB



SINC - LINK



MAY-JUNE '92 VOL 10 #3

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.

SINC-LINK IS PRODUCED ENTIRELY ON SINCLAIR AND TIMEX-SINCLAIR COMPUTERS.

SEND CORRESPONDANCE TO:

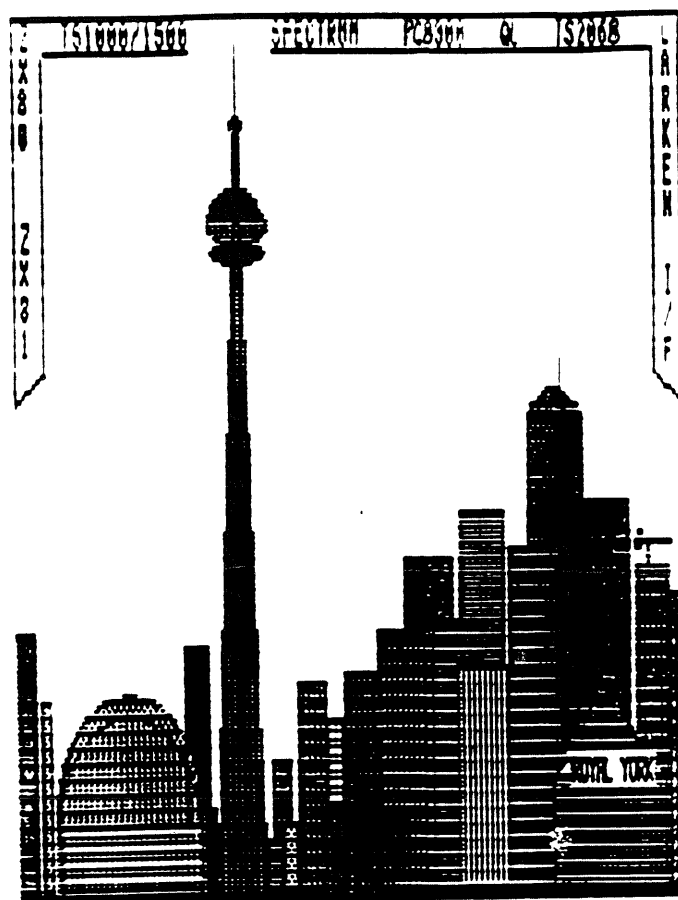
Attention: SINC-LINK
TORONTO TIMEX-SINCLAIR
CLUB, 14 RICHOME
SCARBOROUGH, ONTARIO,
CANADA M1K 2Y1

EDITOR
USERS
COURT,

EXECUTIVE OFFICERS:

PRESIDENT:
TREASURER:
SECRETARY:
ACTIVITIES:
QL CONTACT:
NEWSLETTER:
LIAISON OFFICER:
(Out-of-town members)

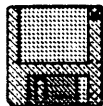
(Area Code 416)
RENE BRUNEAU (531-9749)
BILL LAWSON (444-8772)
GEORGE CHAMBERS (751-7559)
LOUIS LAFERRIERE (820-3725)
HUGH HOWIE (634-4929)
JEFF TAYLOR (244-8583)
GEORGE CHAMBERS, 14 RICHOME COURT,
SCARBOROUGH, ONTARIO, M1K 2Y1
(416-751-7559)



TORONTO TIMEX-SINCLAIR
USERS CLUB

TORONTO TIMEX-SINCLAIR USERS CLUB

SINC-LINK



BOB'S NOTEBOOK
by
Robert H. Mitchell, Willowdale Ontario



SINC-LINK CONTENTS

Each entry is prefixed with the year and first month of issue (eg, 8501 = Jan/Feb 1985, 9203 = Mar/Apr 1992); and with the volume & number (eg, 3/1 = volume 3 number 1) . These are usually followed by the (first) page number of the entry.

1985

8501 3/1 p.4 COLOUR ADJUSTMENTS ON TS2068RGB Monitor details. Comments by George Chambers.
Pot adjustments are given in careful detail.

8501 3/1 p.8 EPROM Burner program by Virgin Roman (ZX81)

8503 3/2 p.11 Use of IN function by George Chambers. Sample ETCH-A-SKETCH to demo the IN function.
Also erasing tapes by Fred Schakel London.p.12 Pin Connections for Spectrum, TS2068 and ZX81

8505 3/3 Sincbits by Ian Robertson. NOTE: This column in TTSUC newsletter contains too many items to catalogue in this file. LOAD file <sincb> for separate listing

8505 3/3 p.7 Keyboard notes C. Hayne; IN; value table

8505 3/3 p.8 Keyboard notes J.Roach; port address table

8505 3/3 p.9 Monitors RGB F. Schakel

8505 3/3 p.10 Cursor Joystick F. Schakel

8505 3/3 p.10 Keyboard problems TS2068p.10 F. Schakel

8507 3/4 p.8 Printer interface 2040C. Hayne

8507 3/4 p.9 Bob's Notebook HEX to DEC conversion; COPY 24lines; increasing array size without losing its contents; ON ERR how to use; screen save in machine code

8507 3/4 p.10 J Burns Telecommunications see also Telecomputing TT 8511 p.6 D.Ridge TT 8509 p.10

8507 3/4 p.12 Keyboard customization L. Crawford; printer codes ESC; circuit diagrams

8509 3/5 p.2 SCROLL outwards P.McMullin

8509 3/5 p.6 Bob's Notebook Interrupts IM2; sample autoline program

8509 3/5 p.11 Larken Disk Drive review G. Lloyd

8511 3/6 p.2 Text Manipulation enhance display of text S.Pocock

8511 3/6 p.8 Tape signal booster L. Crawford; circuit diagram

8511 3/6 p.10 Bob's Notebook Quicksort <<erratum: l. 9625 should read IF q0>p..... Title line in display;
machine code(with examples

8511 3/6 p.11 String input and use of LINED. Mosher

1986

8601 4/1 p.2 Larken Disk Drive boot tape construction G.Chambers

8601 4/1 p.6 Tape copying A.Frailich

8601 4/1 p.7 QL Corner E. Gardonio ICE; Disk Drives; Cartridge Doctor; QL Toolkit

8601 4/1 p.8 Bob's Notebook Binary Search subroutine

8601 4/1 p.10 PLOT CLS SCROLL on TS2068C. Hayne

8601 4/1 p.11 Built-in Clock in TS 2068F. Schakel; also use of IN to sense multiple keys

8601 4/1 p.12 Disk Droppings G.Lloyd; Larken; power supply; controller

8603 4/2 p.6 Bob's Notebook Fontmaker

8603 4/2 p.10 Larken Disk Drives P.Hacksel; power supply

8603 4/2 p.11 QL tip All Caps

8603 4/2 p.12 Spectrum Emulator Zebra Systems Cartridge D. Lindhorst
 8603 4/2 p.12 Spectrum emulator Omni-Emu review D. Lindhorst
 8603 4/2 p.12 Customized ROMs 2068 & Spectrum L.Crawford; EPROM burner; added routines: 2068 has tape copy and Oliger NMI; Spectrum has renumber, header reader, tape copy; details on EPROM burning and mounting them
 8605 4/3 p.2 Sincbits I.Robertson; repairs; Tasword; Timachine
 8605 4/3 p.4 Larken Disk System P.McMullin; also EPROM programmer (burner)
 8605 4/3 p.6 Quilling with QL L.Sadler; tips
 8605 4/3 p.7 Bob's Notebook compactor converts all numbers to VAL expressions; machine code; Larken tips on REMDOS, disk menus; saving arrays
 8605 4/3 p.9 Larken Notes: CRC errors; spectrum stack; CLEAR61000; UDGs; LDOS16.C5; LPRINT USR to print directory; machine code between 23552& VARS is saved with LDOS
 8605 4/3 p.10 Screen addressing TS2068. M Lemyre
 8605 4/3 p.12 Disk Droppings G.Lloyd; Larken
 8605 4/3 p.12 Keyboard codes 2068 Diagram of top row F.Schakel
 8607 4/4 p.2 Purge blank spaces in arrays; utility
 8607 4/4 p.3 Sincbits I.Robertson; cassette booster; spectrum emulator
 8607 4/4 p.4 Larken Disk Copier Utility G. Chambers
 8607 4/4 p.6 TS2068 Reset & Initialization Problems C.Urban; circuit diagram for reset
 8607 4/4 p.8 Bob's Notebook Spectrum; ON ERR and block delete machine code routines
 8607 4/4 p.9 Bob's Notebook: Spectrum items Larken Rename utility program
 8607 4/4 p.10 Disk Droppings Printer DMP 105; Tasman; Larken; modem; Tasword; cheap disks; fast programming
 8607 4/4 p.11 Telecommunications hints and tips; BBS; Tinyboard; Mterm
 8609 4/5 p.2 Hacker scare JT Nguyen
 8609 4/5 p.3 Bob's Notebook Font Loader
 8609 4/5 p.4 Mterm tips G.Lloyd MTERM Ramtop; change Buftop; Autorun Ramex MDOS; Read saved files with loop
 8609 4/5 p.5 TS2068 repairs G.Chambers
 8609 4/5 p.7 Expert Systems; Artificial Intelligence
 8609 4/5 p.8 Larken Disk Drives P.Hacksel
 8609 4/5 p.9 ROM routines G. Lloyd
 8609 4/5 p.10 Disk Droppings G.Lloyd; erratum
 8609 4/5 p.10 Function Dispatcher M. Lemyre
 8609 4/5 p.11 QL review of QSpell L.Sadler; not recommended
 8611 4/6 p.5 Hacker Scare JT Nguyen
 8611 4/6 p.7 Signal booster; amplifier
 8611 4/6 p.9 Larken utilities list Hacksel; Chambers; Michaud Lloyd; McMullin; Mitchell; tips by C. re RAMTOP LDOS16
 8611 4/6 p.10 Mterm for transmitting D Gray
 8611 4/6 p.11 Bob's Notebook ONERR tips; passwords; id
 8611 4/6 p.12 QL Corner R.Cottle Notes on support available
 8611 4/6 p.13 BBS item; communications Mterm specterm-64
 8611 4/6 p.13 Communications TS2050 Modem Mterm II Zterm 64 Specterm 64 Xmodem protocol R Mulder

1987

8701 5/1 p.2 Hacker Scare JT Nguyen
 8701 5/1 p.3 QL in Sincbits I. Robertson magazines: QUANTA; QUANTUM Levels; QL World
 8701 5/1 p.6 QL Corner R.Cottle Summary of QL events in 1986
 8701 5/1 p.7 QL notes QUANTA; Sharp's (QL World)
 8701 5/1 p.10 Disk Droppings G.Lloyd Tasword; Larken; USR calls; big printer control codes
 8701 5/1 p.11 Bob's Notebook Timachine tips
 8701 5/1 p.12 TS2068 colour adjustments F. Shakel; use of VCR
 8701 5/1 p.14 BBS for TS2068
 8701 5/1 p.14 Printer SC Fastext 80 G. Chambers; graphics mode

8701 5/1 p.14 ROMSWITCH Spectrum emulator R. Zannese
 8701 5/1 p.14 Sound enhancement using Radio Shack amplifier; BEEP; SOUND
 8703 5/2 p.4 TS2040 printer C.Urban; description
 8703 5/2 p.5 Calling USR routines D. Mosher; machine code
 8703 5/2 p.6 Communications; R.Mulder; modems; casboard 2068 K. Casby; BBS
 8703 5/2 p.7 Attributes: Change without affecting display TT
 8703 5/2 p.7 BBS for TS Users; modems
 8703 5/2 p.7 Headerless file saving G. Chambers
 8703 5/2 p.8 Larken Power Supply G. Chambers
 8703 5/2 p.8 Screen saving on Larken drive G. Chambers
 8703 5/2 p.10 Printer Smith Corona Fastext 80 review R. Mulder
 8703 5/2 p.11 QL block printing saving and loading with QUILL software @ \$15 CDN
 8703 5/2 p.12 Bob's Notebook organizing program info
 8703 5/2 p.13 Printer ribbons: re-inking
 8705 5/3 p.3 QL in SincBits I.Robertson QLWP QUILL; Schonkeyboard; Disk ControllerWD1770 should be replaced byWD1772-02
 8705 5/3 p.13 QL Wayfarer ex QL Corner R. Cottle . QL Library kicks off
 8707 5/4 p.2 QL in SincBits Ian Robertson software: Spellbound; Desktop Publisher; Front Page Extra; CP/M Emulator; updates to Taskmaster; The Editor; Turbo; Supercharge; QLiberator; Eye -Q; Super Media Manager; QL TrumpCard 768K RAM; Thor cartridge 68020 CPU 16.7 Mhz, 3.5" drives; 20Mb hard drive
 8707 5/4 p.5 QL in Disk Droppings Greg Lloyd QUILL=Word Processor; printer i/f needs CoCo II serialcord; Superbasic; DD via Cumana with 3" Amdek disk drives
 8707 5/4 p.11 Bob's Notebook Chart of Control Code used with Fastext 80.Dot Matrix Printer Copy of SCREEN\$ (22 lines)
 8707 5/4 p.13 QL programs Sound experimenter; Easel print Epson printer only(?)
 8709 5/5 p.3 QL in SincBits Ian Robertson TEXT 87 WP; modems; CP/M; RS232; QLTERM 2.0; QUILL Columnizer
 8709 5/5 p.5 Bob's Notebook Hacksel printer code modified for Spectrum. Timachine shortcuts
 8709 5/5 p.8 QL Wayfarer R Cottle Sharp's; ROM fixes; toolkit; Qflash; EPROMs; G_SAVE G_LOAD; Hex Loader; Clone; QUILL mailmerge; Labeller; Merge File;3.5 inch disk; microdrive; Rod Gowan
 8711 5/6 p.2 Disk droppings by Greg Lloyd. QL info. Ramdisk notes
 8711 5/6 p.3 QL notes from Bill Lawson
 8711 5/6 p.4 Timex repairs by George Chambers Also: on small computers by Edward Abisdreis. And: Help wanted on Hacking Spectrum Games by Douglas Jeffery
 8711 5/6 p.5 Larken/TS2068 Autostart Menu. Disk Menu listing. By George Chambers
 8711 5/6 p.6 Bob's Notebook PEEKing and POKEing About Disk Indexer shows current status etc
 8711 5/6 p.7 Digital Imagizing Part II by Eric Michaud. Circuit diagram and details
 8711 5/6 p.9 TS2068 tutorial on Input Monitoring; tape header formats. Save, Verify, Load and Merge. from Long Island group
 8711 5/6 p.11 Larken DSK400 Disk Operating System. Review by George Chambers
 8711 5/6 p.12 Larken and the Hunter by Mel Richardson; diagrams and details
 8711 5/6 p.13 Coloured SCREEN\$ dumps for the TS2068 by Jeff Taylor.VIC-1520 printer/plotter; J McMichael's PIC-PLOT
 8711 5/6 p.14 A few words from the new president Eddie Maybe.

End of Part 1 of 3 parts.

This file, comprising 17 pages, can be provided on an IBM compatible disk in Word Perfect 5.1 format or straight text format. It could be downloaded to a TS 2068 on request.
 Bob Mitchell 920307.

NOTES FROM MY CORRESPONDENCE
by George Chambers

It was suggested at one of our meetings that we should devote a part of our newsletter to questions and answers from our members. A good idea indeed. I thought maybe these excerpts take from a letter that I wrote to one of our members might prove interesting. I hope you like it!

PEEKING THE LARKEN LKDOS

Now, to answer one of your questions; the one about being unable to POKE values into the Larken LKDOS RAM. From looking at the values you give in your letter, i.e. 16090,8 and 16094,14 I can see that you are PEEKing the 2068 ROM, not the Larken RAM.

If you look at Larry Kenny's instructions it will become clear. I can see that you know how to POKE a value into The LKDOS RAM, but you do not know how to PEEK the LKDOS addresses. The values you are getting are from the 2068 ROM, not the LKDOS RAM. What you have to do to PEEK the LKDOS is first place a pointer into the LKDOS. Do this by the command PRINT USR 100: POKE 8200,xxxxx, where xxxxx = the address you wish to PEEK, such as 16090. Having done that, then enter the command PRINT USR 110. You will get the value contained in the LKDOS RAM, not the 2068 ROM. In his manual (middle of page 6, in my copy), Larry K. has a routine to print the first 10 bytes of the LKDOS EPROM.

THE OMNIBUS PROGRAM

I rather think that the OMNIBUS did not provide for changing the LF on/off function. A bit of an oversight, I think. The LKDOS defaults to a LF with CR, and probably our printers happen to need the LF. I could give you a change for the OMNIBUS if you would like to be able to vary it from the menu, or the program could be modified to automatically give you no LF. A line in OMNIBUS in the vicinity of 1200-1230 which read "PRINT #4: POKE 16092,0" would give you a NO LF condition.

MODEMS

The TS2050 modem plugs directly into the rear of the 2068. Requires no other interface circuits. It is powered by much the same power supply as was used on the TS1000. If you were to use a regular modem, then you would need a RS232-type serial interface. Such as the ZSI/O board that was put out by Ed Grey.

I have a retired Bell System modem and it required a serial i/f board. The advantage of this modem over the 2050 is that it give me a choice of 300 baud or 1200 baud. Nowadays 300 is almost passe, even amongst bulletin boards.

I'm interested in your reference to Bill Jones' SMART TEXT and Daisy 6. Bill talks about how marvellous a program it is; how many copies have sold. But Bob Mitchell and I are not that taken with it. Seems to be just too convoluted, and difficult to get into. Our question is, well maybe that's how many copies have sold, but how many are being actually used. A different story, I rather suspect. Let me know if it becomes your standard management system. Incidentally, I see an advert in the Indiana T/S user newsletter by Bill Jones, selling off what appears to be his complete 2068 equipment, for \$750. He seems to be getting out of the 2068 field. He was into the QL, and I don't see any of that part for sale.

LARKEN RAMDISK

You ask about the RAMdisk. There is really no mystery about it. I'll try to keep it simple. The easiest way to use it is to simply treat it as a solid-state drive, having 48 tracks. Though you can format it without having to load any formatting software; you simply "PRINT USR 100: FORMAT

"8", (for the number of chips you have on the board), and that's it. Then you SAVE, LOAD, ERASE, etc., the same way as with a regular drive.

You ask about the merits of using the Larken disk Editor program to format the RAMdisk. I suspect they are both the same, just that the Editor may have some extra frills. I've never done it any other way.

However you can do bank-switching with it. That is what Larry Crawford has done. He bank-switches one RAMdisk chip at a time (which amounts to 5 tracks, incidentally). What happens is that the upper part of the 2068 memory, from addresses 32768 to 65535 is swapped with any selected RAMdisk chip.

Larry C.'s program is able to do this automatically as required, swapping banks back and forth, as needed by it's SEARCH and SORT routines. The BASIC part of his program must stay below the 32768 cut-off, or that part of Basic above 32768 would be switched as well, with disaster ensuing. If you look at his basic listing you might see some OUT 244,7's and other suchlike. Those are the bankswitching commands.

Larry C.'s program does a SAVE of the RAMdisk banks before making use of them for the database. The program, as a first order of business, saves the RAMdisk to a disk in your choice of drive. It then loads the database data into the RAMdisk. Whether it formats it first, I'm not sure. I suspect that it simply bankswitches a bank, then with this bank effectively in the 2068, it fills up the memory, swaps another RAMdisk bank, and so on until all the data is loaded. In the case of L. C.'s "movie title" demo, I think it is only one bank. But I think Bob Mitchell had a "colour slide" database that used a second bank (chip) of the RAMdisk.

Now, when you are finished with the database you have to execute a graceful exit, so that the program restores the RAMdisk with the programs that were initially saved. It does work, I tried it, but I would caution you to also have a spare RAMdisk copy, since I seemed to frequently exit ungracefully; i.e. turn off the computer, and then be unable to restore the RAMdisk contents I had saved on disk.

Larry Kenny's RAMdisk backup program operates in the same fashion. He switches each of the RAMdisk chips in turn into the 2068 memory, and does a 5-track SAVE. Richard HURD has another RAMdisk backup program which saves the RAMdisk on an individual program basis. I tend to like this one better, since it means that I could pick out a single program if I cared to load it into the computer.

I find that the RAMdisk works perfectly OK when the Larken disk interface is the first add-on to the computer. I think this is the most useful position. I never considered any other position for it, actually. My RGB monitor plug is situated at the back and it crowds the Larken boards quite severely. In fact I had to remove the joystick jack and bring out an extension cable for it, in order to give the RGB cable room. Incidentally I use a VCR cassette case, suitably trimmed, to cover the RAMdisk. I opened the case on one side, and it simply drops over the board. Very neat. I enclosed my LKDOS board also, using a case from a Memotech 16K rampack for the ZX81. But one could also use a VCR cassette case for the LKDOS i/f as well.

EXPANDING THE RAMDISK

You mentioned having a problem with the RAMdisk to backup. I think I may have an idea. Were you using 4 chips on the RAMdisk initially, then upgraded to 8 chips? I had an experience like that, where the RAMdisk said it only had say, 24 tracks, when I knew there was a full complement of 48. The problem is this. When you did a disk-save, i.e. RAMdisk Backup of a

4-chip RAMdisk; then added another 4 chips to the RAMdisk, then reloaded the RAMdisk Backup, the directory track still says there is only 24 tracks. Even though you may have formatted it to a full complement of 48 tracks, when the RAMdisk Backup is re-installed it overwrites the Directory track, to show 24 tracks, or whatever. It doesn't know there are now 48 tracks, the LKDOS looks at the Directory track, and it says there's 24. What you have to do is save the programs individually to disk, reformat the RAMdisk to 48 tracks, then reload the programs individually. Then make a new RAMdisk backup.

This will only happen when you add chips to a partly-populated RAMdisk. Once it is full with 8 chips the problem will not recur.

If one knew the command to bankswitch a RAMdisk bank, it would be a simple matter to save a RAMdisk bank to disk. I have never gone into the OUT commands for that purpose. Never had a reason to do so. If I did I would probably study Larry Kenny's RAMdisk Backup program. Or the definitive article on the subject, by Larry Craweford, in the Mar/Apr '90 issue of Sinc-Link (Vol.8 No.2). It contains the necessary OUT command information.

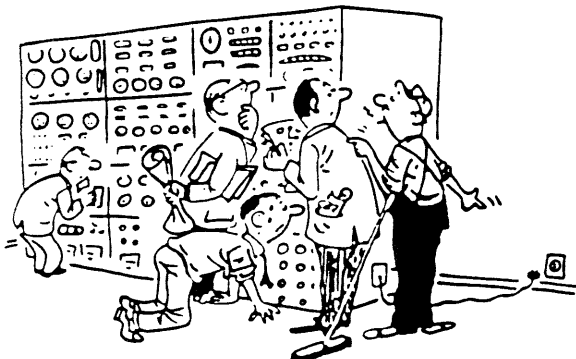
Oh, a couple of things. I removed the black line from the bottom of your SCREEN\$ images. I did it with the doctor.B1 program. I determined which tracks the offending black area was located, loaded these tracks individually, and worked them over. I broke into the "doctor.B1" program and made a change to the "alter" option, so that it changed the code values from 7 to 56. The tracks, when loaded into doctor reside at 50000 upwards. I determined that the offending code was at addresses 51750 to 51845, so I introduced a FOR/NEXT loop to POKE 56 into these addresses. I then re-SAVED the track, and voila! the black bar was gone. Did this to all the SCREEN\$.

Also, in the GYP10X.Bx program, in line 510 (was it?) there is a variable that needs to be changed from 1 to 2, to make the program work properly with the Larken system. Jack Dohany has used a clever method of making his programs universal to all disk systems.

I shall close off now, and get this into the mail.

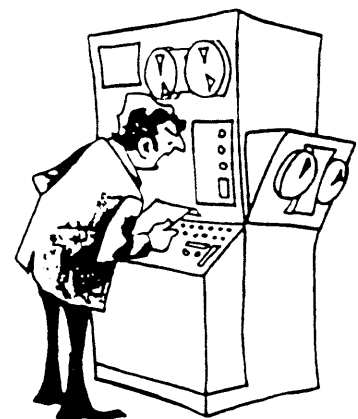
Sincerely,

George Chambers



DICK LUCAS, MASTERS AGENCY

oops!



"Look at me when
I'm programming you!"
HOEST IN BOYS LIFE

Q L I P S

by Hugh H. Howie

Some time ago when I was busy on another matter, QUANTA arrived at my door, and I did not take time to look at it for a month or two, and when I did read it, I found an excellent tutorial on Beginning Programming, by Alan Pywell, and I decided to write asking him for permission to run the series in Sinc-Link. This is the result of that request.

Alan had intended the series to run a bit longer than it did, but owing to lack of encouragement he dropped the subject and destroyed the disk on which had saved the tutorial, so I now have to re-type it so that you, the reader, can have the benefit of his work. At the time he wrote this he was Software Controller for QUANTA, and has now been promoted to Editor.

There is a third chapter partly completed, and Alan says he will try and get back to it. So if you want more of this type of thing, drop Alan a line and let him know what you want.

B e g i n n e r s C o r n e r

Elementary Programming ----

An occasional series for the occasional programmer.
Copyright © Alan Pywell 1991.

Introducing buzz-word number one..... The word is 'parameters' - it's a lovely one, provided that you know what it means!. It usually goes with the word 'passing' and becomes a 'buzz-phrase' - 'passing parameters'. I've passed cars, exams and even a pentop I accidentally swallowed once but you don't wish to know about that.

Most programs ask the user for input of some kind, often to select from a menu of options. Let's have four options. Also a rule:- the options will be numbered so that the user will be asked to type a number instead of the frequently used and untidy-looking 'type the initial letter'. We can then painlessly produce a routine which can be incorporated easily into any program we care to write. The menu will look something like this:-

```
100 print "Choose an option - press a key (1 to 4)"
110 print "1.....Option 1"
120 print "2.....Option 2"
130 print "3.....Option 3"
140 print "4.....Option 4"
```

The above is largely a matter of personal preference as regards wording and screen layout. It looks better than, say:-

```
110 (p)rint a page
120 (s)ave game
130 (n)ot easy to read
```

Of course we can have as many options as we need - we have just four. So far so good, now to get the users choice.....

"Beginners" Page 1 of 6

Remember (you have, haven't you?) that **INKEY\$** will not wait for a key-press (you have to be already pressing a key when the program reaches **INKEY\$**). **INKEY\$(-1)** will wait until you press a key. We also need to check that the key pressed is valid and go back to wait for another key-press if it isn't. Our first effort might look something like this:-

```
200 v$=inkey$(-1)
210 if v$<>'1' AND v$<>'2' AND v$<>'3' AND v$<>'4' THEN GOTO 200
    :REMark if user has pressed anything but keys 1,2,3,4 go back for
    another keypress.....
220 REMark we'll be here ↓ if a key 1,2,3 or 4 was pressed....
230 IF v$ = '1' THEN go to another part of program
240 IF v$ = '2' THEN etc. etc.
```

This would work but it does not take advantage of SuperBasic's power. The first step is to get rid of that cumbersome line 210. We can do this very easily by putting the allowable keypresses into a string and checking if the number typed is in it. SuperBasic provides a powerful keyword for this purpose - **INSTR** (pronounce it 'in string' or better still 'is in string'). First put the permissible numbers into a string - remember our numbers are 1,2,3,and 4:-

```
190 LET permitted$ = "1234"
200 v$ = INKEY$(-1)
210 if NOT v$ INSTR (permitted$) THEN GOTO 200: REMark read it as "if v$
    is not in permitted string."
```

It's looking a mite less inelegant and will run a trifle faster. Now let's get rid of the GOTO. For this we need three more keywords:-

REPEAT **END REPEAT** **EXIT** wow!

```
200 REPEAT keys 210 v$ = INKEY$(-1)
220 if v$ INSTR permitted$ THEN EXIT keys: REMark (read as "if v$ is in
    the permitted string then leave keys" i.e. goto the next line after END
    REPEAT KEYS in this case line 240
230 END REPEAT KEYS : REMark come here when EXITing REPEAT
240 IF v$="1" etc. etc.
```

The **END REPEAT** is the magic bit. By the time the QL reaches it, it "knows" whether the condition(s) we've programmed have been met. So **END REPEAT** either causes a jump back to the line after **REPEAT** (210 in our case) or it is skipped over, control going to the next line after the **END REPEAT** (240). The bit between **REPEAT** and **END REPEAT** is a loop. To put it another way, if the condition(s) are not met then go back to the line after **REPEAT** keys (stay in the loop), but if they are met then don't loop but exit the loop instead (skip over **END REPEAT**).

So what does all this have to do with passing parameters? I thought you'd never ask. Just for the moment don't worry about what a parameter is - like British Rail I'm getting there.

The routine for getting the users input isn't finished yet but I want to digress for a moment.... There may be many places in your program where you wish to get an input. Instead of writing the routine in every place it's needed we can write it just once and call it up from wherever we need it in the program. The routine then becomes a subroutine. The word subroutine means that it's not part of the main flow of the program.

"Beginners" Page 2 of 6

We need another two words- **GOSUB**

RETURN

GOSUB means "go to (not GOTO!) the subroutine but expect to come back eventually" i.e. GOSUB 10000 means go to the subroutine beginning at line 10000. The last word in the subroutine must be RETURN- guess what RETURN means? It means to the line immediately after the line that contained the GOSUB. When our clever QL comes across GOSUB it "remembers" to where it needs to return when it eventually finds RETURN.

```
100 GOSUB 10000
110 rem return to 110 from subroutine
120 main part of program
130 more of same 140 etc
9999 end of main part
10000 rem this is a subroutine
10010 PRINT "Press a key to select:-"
10020 Print "1---option 1"
10030 etc
10100 RETURN: remark QL mutters "what's the line after the line that called
      me? Ah, I remember, it was line 100, so I will return to line 110"
```

Using GOSUB is quite easy but I won't dwell on it because YOU SHOULD NEVER USE GOSUB! "Why?" I hear you cry. "Because SUPERBASIC has something much better" comes the smug reply.

Instead we're going to give our little subroutine a name. This is another powerful feature of SuperBasic - the ability to give a subroutine a name and to call it by name instead of by line number. Let's call it **"CHOICE"**. We name a subroutine by use of another keyword. But first, let's call our subroutine a "procedure" then we're really 'with it!' "Procedure" is a Buzzword meaning 'subroutine'.

The keyword:- **DEFPROC** (read it as "define procedure") Type in DEF PROC and the QL knows you mean "DEFine PROCedure", clever chap.

```
200 DEFine PROCedure CHOICE (minimum, maximum) : REMark Hey! Two
      parameters have crept in! Okay don't panic. At some earlier point in
      the program, just after we'd printed the menu, we needed to ask the
      user for his choice of options and we needed to ensure that our
      program will respond only to keys 1,2,3 or 4. It may come as a
      surprise to realise that 1 is the minimum and 4 is the maximum! So the
      parameters turn out to be simply bits of information that we are
      passing to our procedure,
90 LET permitted$="123456789" : REMark let's go mad - we might decide we
      need ten options somewhere else in the program....
100 PRINT "SELECT AN OPTION"
110 PRINT "1.....OPTION 1" etc.
```

Now for the juicy bit....

```
150 CHOICE 1,4
```

We're telling the QL to go to (not GOTO!) our procedure CHOICE and giving it two numbers (parameters) to play with..... The QL is telling itself "I must remember where I was when I jumped to CHOICE so that I know where to come back to when I've finished with CHOICE" O know I will have finished with line 150 so I'll return to the next line - looks like line 160 to me... and

I'll need the numbers 1 and 4 in CHOICE by the look of it.... Now what line does CHOICE start on, I'll quickly skim through all the lines 'til I find it, Ah well, here goes...."

160 some more program.....

200: **DEFINE** PROCEDURE CHOICE (minimum,maximum) We've passed two numbers (parameters) to the procedure and the procedure has two parameter name in brackets after its name. QL starts chuntering to himself again. "CHOICE on line 150 has sent me two numbers and now I've found two words. What to do, what to do? I'm going to call them variables and give them the numbers, that will sort that out. The first number is 1 and the first variable is minimum so I'll say minimum represents the number 1. Now for the second - I'll let the next variable represent the next number passed to me, so maximum represents the number 4. And the third variable, Oh, there isn't a third one, but if there was....think I'll call the variables "parameters"

210 REPEAT keys

220 v\$=INKEY(-1) seen that before somewhere.

230 if v\$ INSTR (permitted\$(minimum to maximum)) THEN EXIT keys: REMark (read as "if v\$ is in the part of the permitted string between the minimumth and maximumth character in that string i.e. between the first and fourth (inclusive) then fine, leave the REPEAT loop, we've got what we wanted" so QL chunters along, ignoring all lines that don't say 'END REPEAT'-when it finds it, it "exits" the loop (by skipping to the next line after END REPEAT) But if v\$ isn't in permitted\$ then loop (to 200)

240 END REPEAT keys (EXIT finds this line then goes here)

250 END DEFINE (the END DEFINE ends the definition of our procedure CHOICE. It acts in exactly the same way as RETURN i.e. the program returns to the line after the line that invoked (called) the procedure. The QL mutters to itself "Now where was I when CHOICE was called? Ah, yes, I remember, line 150 so I must return to line 160...."

We called CHOICE simply by putting its name on line 150 i.e. 150 CHOICE replaces 500 GOSUB line-number-at-start-of-subroutine. The line DEFPROC CHOICE is the same as the above line-number-at-start-of-subroutine. ENDDEF is analogous to RETURN(type ENDDEF and QL knows you mean END DEFINE)

With named procedures the program is easier to read and understand, GOSUB doesn't really tell you much. If you leave a program for a year and then decide to work on it again you'll be glad you used named procedures (if, that is, you used meaningful names...you will, won't you?)

I wrote a sophisticated adventure writing tool. Being accustomed to a Spectrum at the time I used short variables and procedure names (to save a few bytes of RAM!!!). I returned to it after about two years and could not fathom it. I've started from scratch again. Eighteen months hard work wasted. BE WARNED!. Another big advantage of breaking a program into procedures is that you can write a procedure and test it before writing any more- if there's a fault you have to check only a few lines instead of possibly the whole program.

Let's recap and at the same time leave some lines for some more program and just for a change we'll have 6 options.....

10 some program

20 some more


```

25 LET v$ = "1234567890"
30 CHOICE 1,6: REMark 6 options remember
40 REMark when the procedure CHOICE which we've just called hits its
   END DEFINE it will RETURN to here ↓
50 IF v$= "1" THEN (do option 1)
60 IF v$="2" ( and so on )
100 if v$="6" THEN (do option 6)
110 to 9999 some more program (some more program)

```



```

10000 DEFine PROCedure CHOICE (min,max): REM min,max are still
        meaningful?
10100 REPEAT keys 10200 v$= INKEY$(-1)
10300 if v$ INSTR (permitted$(min to max)) THEN EXIT keys
10400 END REPEAT keys
10500 END DEFine
If you want 9 options then 30 CHOICE 1,9

```

If you want options 3 and 4 only then 30 CHOICE 3,4

Now let's get rid of all those IFs. To do this we need two more words:-

SELECT END SELECT

SELECT means "select one of the following options" while END SELECT means "there are no more options". Imagine that you have written four procedures already and you are satisfied that they all work. Of course these procedures can have any name you care to choose. Let's suppose they are called:- **SAVE_GAME**, **HISCORE_TABLE**, **FINISH** and **LOAD_STATE_OF_PLAY** Note that spaces are not allowed. Use underscore instead. Underscore is the (shifted) key between 0 and =. Now instead of IF v\$="1" then SAVE_GAME IF v\$="2" then HI_SCORE_TABLE IF etc.

```

LET V=v$: REM V takes the "value" of v$ i.e. if v$="7" then v=7
SELECT V
=1:SAVE_GAME replaces IF v$="1" THEN SAVE_GAME=2:HISCORE_TABLE
replaces IF v$="2" THEN HISCORE_TABLE =3:FINISH replaces .....=4:
LOAD_STATE_OF_PLAY END SELECT no more options.

```

I have sneakily replaces THEN with a colon. You may do this anywhere in a program to save typing. It is obligatory inside a SELECT-END SELECT bit. The above is called the short form of SELECT. Each = and its associated bits must be all on one line. If you wished each option to consist of many instructions use the long form of SELECT, i.e.

```

SELECT ON V
  ON V=1: PRINT "Make sure disk is in FLP1"
        PRINT "PRESS SPACE to save game"
        (space pressed and game saved....)
        PRINT "Continue game?"
        and so on..... all the above lines will be acted upon if v=1
  ON V=2: etc.

```

As you've had enough ado for one day, here without more ado, is a complete listing which, if I've done well, you will understand. NOTE how one named procedure can call another. This procedure MENU_1 prints a menu then calls GET_CHOICE to get a keypress.

```

100 permitted$ = "123456789"
110 REMark MENU-1 asks if you want instructions,1=yes,2=no
120 MENU-1:REM return here from MENU-1
130 SELECT V
140 =1:INSTRUCTIONS
150 =2:GAME
160 END SELECT
170 REM end of game returns here- then, ANOTHER-GAME?,STOP etc-up to
    you
200 DEFine PROCedure GAME
210 REM start of game
1990 REM end of game
2000 END DEFine GAME
2010 DEFine PROCedure GET-CHOICE (min,max)
2020 REPEAT KEYPRESS
2030 V$=INKEY$(-1)
2040 IF V$ INSTR (permitted$(min,max)) THEN EXIT KEYPRESS: ELSE PRINT
    "NUMBER MUST BE BETWEEN ";min;" AND ";max
2050 END REPEAT KEYPRESS
2060 V=V$:REM convert V$ now instead of many times in main program
2070 END DEFine GET-CHOICE
2071REM.....divider for clarity
2080 DEF PROCedure MENU-1
2090 CLS:PRINT "Do you want instructions?"
2100 PRINT "1-----YES"
2110 PRINT "2-----NO"
2120 GET-CHOICE 1,2:REM this procedure calls another
2130 END DEFine MENU-!
2131 REM.....divider for clarity
2200 DEF PROCedure INSTRUCTIONS
2210 CLS
2220 PRINT "The object of the game is to enable unsuccessful and impover-
    ished would-be authors to get their names in print by writing a piece on a
    specialised subject, safe in the knowledge that only a few readers will
    realise that it's a load of old cobblers,"
2230 PRINT "Don't call us, we'll call you"
2400 END DEFine INSTRUCTIONS

```

I hope the above will be of some use to those who program only occasion-ally and therefore have not become very familiar with SUPERBASIC. I would appreciate some reaction to the content and/or style as well as suggestions and queries for possible future articles. Drop me a line please. All letters will be examined by the bomb squad. In the unlikely event that I get a lot of feedback I will not answer individual letters but will acknowledge them in these pages. Thankyou 30/5/1991

Alan Pywell, 13 Sandyfields Close, Sea Lane, Saltfleet, Lincs, LN11 7RP. England.

There will be another instalment in the next issue, but do not let that stop you from dropping Alan a note asking him to attend to your own particular interest or need.

Hugh H. Howie.

ZX81 RESOURCES - 64k SRAM MEMORY

Rene Bruneau April 1992

Gladstone 64k memory finally bit the dust. I didn't really want to go out and find another one, so I considered building one with static ram. Tom Stoddard wrote an excellent article in Time Designs for modifying a ZX81 for 64k internal ram. I did not want to do anything with the CPU board, so I developed a printed circuit board using Mr. Stoddard's design. As with most devices for the ZX81, the SRAM board plugs into the back of the computer with a feed-through for other units. In its basic configuration, the SRAM board will replace a 16k rampack with ram in the 8-16k block for utilities or SHREB Hirez graphics.

THE HARDWARE

The pcb mask, board layout, and parts list are shown below. As with past pcb projects that I have done, the pcb mask is a 2X mirrored image that can be xeroxed onto TEC-200 mylar film at 50% reduction. TEC-200 is a clear mylar that doesn't allow the xerox medium to bond to it. A regular iron with heat setting at cotton (hot!) will transfer the design onto a clean copper blank which can then be etched.

Stuffing the board is straightforward. Start by inserting the jumpers (J3), note that 3 jumpers are under the IC sockets. Install the sockets, resistors, diodes, and switches. Install the capacitor, being careful of its orientation and the transistor with the flat side facing the top of the board. Install the edge connector allowing at least 3/8" between the connector and the board for clearance. Check the board for cold solder joints and solder bridges.

Install U1 and U2 (pin 1 to the top right for both) and plug the unit into the computer. Power up and confirm that the memory board is working. If the screen remains blank, power down and check your work.

At this point you should have a working 32k computer. To get 64k you will have to mount U3 on top of U2. Unless you are very good with a soldering iron, I would recommend that U3 be mounted in a socket soldered onto U2 with pin 20 bent out at a right angle. This pin is connected to the pad provided on the pcb adjacent to the socket.

CONFIGURATION

Switch 1 is used to enable the 8-16k block for use with other peripherals mapped to this area such as the Larken disk interface or an eeprom board.

Note that the traces connecting the data and address lines of the sram to the cpu are not in order, therefore it is not possible to mount a eeprom in place of the second sram. Though it hasn't been tested, an eeprom should work because you can program it in situ.

An optional feature is a reset switch consisting of R6 and SW3 located on the right side of the board.

PARTS LIST

U1	74LS145
U2,U3	43256-15L, 256k x 8 SRAM
D1-D3	1N4148, signal diode
Q1	2N3904, NPN transistor
R1-R5	2.2k 1/4 watt resistor
C1	2.2uF 16v Tantalum cap

1 Z81 edge connector
2 28-pin dip socket,
1 16-pin dip socket
2 SPST switches (SW1, SW2)
RESET SWITCH (Optional)

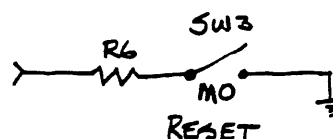
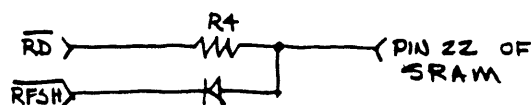
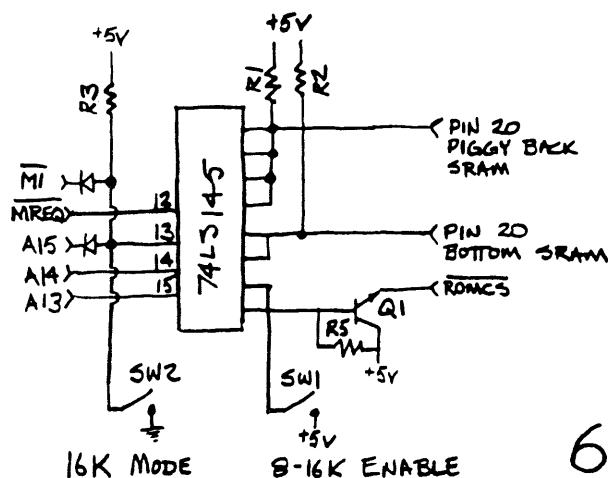
R6 1k 1/4 watt resistor
SW3 SPST Momentary on

Printed circuit boards, kits, and assembled boards can be made available if there is enough interest.

INFORMATION

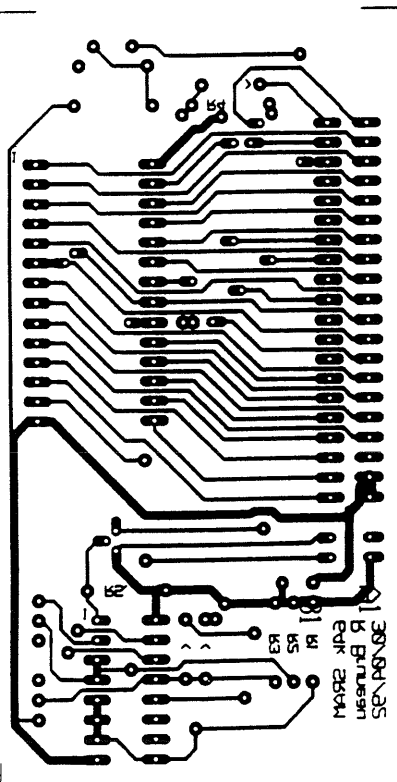
If you are interested in working with TEC-200, it is available from

The MEADOWLAKE Corp.
Dept. 6, P.O. Box 497, Northport NY 11768
5 sheets for \$3.95, 10 sheets for \$5.95
add \$1.50 for postage

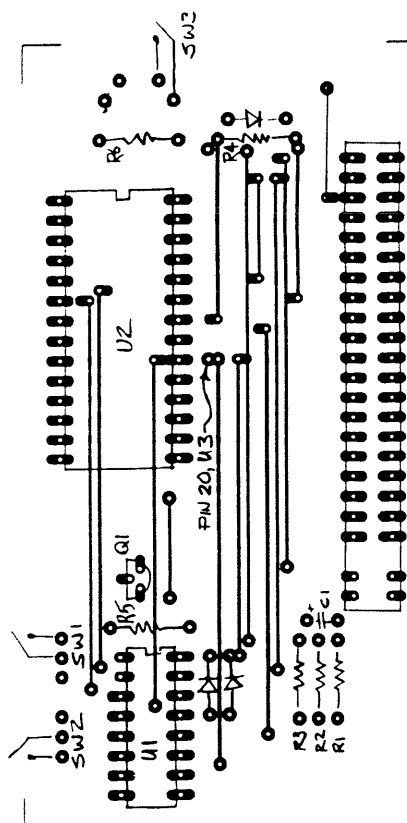


64K SRAM T. STODDARD

Cont.



64K PCB



64K SRAM Component
Layout

VIDEO PRIMER

What do you do when you want to upgrade the video display for your TS1000 or 2068? When you look in the trade papers and advertisements you'll see things like CGA, RGB, EGA, VGA, Composite Monochrome or TTL. The technical jargon can confuse even the experts. What follows is a short thumb nail dictionary to add to your list of computereze:

CGA: Colour Graphics Adapter....this refers to the video driver card installed in MS-DOS (ne IBM and clones). It is capable of displaying four colours in standard resolution mode and only two colours in high resolution. Sometimes video monitors will be advertised as such, though technically they should be listed as RGB (see below).

RGB: Red/Green/Blue....specifically a video monitor capable of displaying up to 16 colours (This is what you want to get for your 2068).

EGA, VGA: Extended Graphics, Video Graphics Array...video drivers or monitors for MS-DOS machines. The Timex Sinclair computers are not capable of working with these.

TTL: A monochrome monitor that has a similar pinout to the RGB monitor (except for the colour, of course). To date, we are not aware of anyone who has connected one to a 2068. With surplus prices starting at \$10.00, this may provide a very clear display. It bears investigating.

Composite; Either monochrome or colour, both the TS1000 or 2068 provide a signal that will drive these monitors directly or with suitable circuitry. Note that the Colour monitor performs poorly; an RGB monitor is far better.

If anyone is interested in more information, contact George Chambers or myself

Happy computing

Rene Bruneau

Bill Harmer
97 Ruskin St.
Ottawa, Ont.
K1Y 4B3

April 6, 1992

Mr. George Chambers
Toronto TS Users Club

Dear George,

Just thought that I would drop a little note to you on some amazing things that are resulting from something that your people started.

The chain of events began when a note from the German QL User Club (about 1000 to 2000 members strong) inviting foreign contacts, was passed on to be published in the Toronto club newsletter.

I began exchanging newsletters and letters with them just for fun and mentioned the contact information to others, including SLIX, run by Bill Miller in Cupertino, California. Well, he was able, using some electronic address numbers passed on this way, to contact them by electronic mail, taking (he thinks) 5 days. Mr. Franz Herrman, a 21 year old German student and executive of the club replied and the news is quite exciting.

In Germany they have been porting a lot of interesting stuff to the QL including the ZOO and LHx data compression utilities (formerly available mainly for IBM PC's), and a new experimental one, HAR, from Japan, which, to quote Mr. Hermann, "has been much more adapted to QDOS" and "it is the best archiver for the QL". There are enough QL users in Germany that work for this sort of software has been easy enough to get, so some rather fancy software is floating around there, a lot of it in the public domain and obtainable from that club by mail.

German commercial products include a hardware attachment for the Atari ST to let it run Sinclair QL software.

The long chain of events that got this interesting information flowing to this this side of the Atlantic is typical of the reason why we need user groups! One person just can't have the time or energy to find out all these little niches in which support of hobby computers is to be found.

SLIX in California also has some interesting news of their own. Their attempts to get UNIX on to the disk interface-equipped QL in cheap or public domain versions has been given a recent boost by the emergence of a sort of UNIX and AT&T clone, LINUX. Previously they experimented with MINIX and the less elaborate XINU, but both were copyrighted although source code in C language is cheaply available for viewing and analysis (via Austin Code Works, Austin, Texas) in Turbo C of MicroSoft C for the IBM PC.

Slix also mentions QL microdrive innards at about £20 and QL keyboard membranes still available from the UK, from Tony Firschman, TF Services, 12 Bouverie Place, London W2 1RB, England.

Seeking out sources of support overseas seems the way to keep hobby computers going today. The world really is shrinking, and to the benefit of our hobby.

Sincerely
Bill Harmer

QL SUCCESS

SOME TIME AGO REAL GAGNON THEN THE EDITOR AND PUBLISHER OF QL__DOC THE SINCLAIR NEWSLETTER FROM MONTREAL (IN FRENCH) HAD WRITTEN AN ESSAY ON THE BENEFITS OF USING SUCCESS , THE CP/M EMULATOR FROM DIGITAL PRECISION. I WAS NEVER SUCCESSFUL IN USING THIS EMULATOR , BUT THE OBVIOUS BENEFITS OF BEING ABLE TO USE THE MULTITUDE OF PROGRAMS WITTEN FOR CP/M ALWAYS TEMPTED ME. I HAVE TRANSLATED THE ESSAY AND IT IS AVAILABLE EITHER FROM THIS NEWSLETTER OR FROM THE EDITOR OF SINC-LINK. IF YOU ARE SUCCESSFUL IN USING SUCCESS PLEASE LET US KNOW.

LOUIS LAFERRIERE

QL + CP/M = SUCCESS

DIGITAL PRECISION , IS RECOGNIZED FOR THE QUALITY OF THEIR SOFTWARE . SOME OF THEM HAVE BEEN LANDMARKS TO IDENTIFY THE VARIOUS STEPS IN THE HISTORY OF THE QL. WE SHOULD MENTION SUPERCHARGE , THE EDITOR , DESKTOP PUBLISHER AND NOW THE CP/M EMULATOR SUCCESS. THIS EMULATOR WILL OPEN THE DOOR TO THOUSANDS OF PROGRAMS WRITTEN FOR THIS SYSTEM. DON'T GET CARRIED AWAY , BECAUSE THE ROAD TO SUCCESS IS PAVED WITH ROADBLOCKS, NOT NECESSARILY INSURMOUNTABLE BUT THEY ARE THERE ANYWAY.

1 = SUCCESS , THE SYSTEM

IN FACT, THE HEART OF SUCCESS CONSISTS OF A SOFTWARE TO TRANSLATE COMMANDS PRIMARILY DIRECTED AT THE Z80, SO THAT THE 68008 CAN UNDERSTAND THEM. THE CP/M WAS DESIGNED FOR THE Z80 OR 8080 CPU. IN ADDITION SUCCESS COMPRISES THE TOOLS TO CHANGE THE QL SO THAT IT CAN USE CP/M. THEN WE HAVE CP/M ITSELF. EACH COMMAND OF THE CP/M , OR ANY PROGRAM RUNNING WITH CP/M IS TRANSLATED INTO A COMMAND FOR THE THE 68008. NATURALLY THIS TRANSLATION AFFECTS THE SPEED OF OPERATION OF THE VARIOUS PROGRAMS. THE QL WITH SUCCESS IS EQUAL TO A Z88 OPERATING AT 1 MHZ ON A MACHINE WITH 128 K OR ABOUT 1.8 MHZ WITH A MEMORY EXPANSION.

YOU MUST HAVE AT LEAST ONE DISK DRIVE, OF 728 K [1440 SECTORS] BECAUSE THE PROGRAMS FOR SUCCESS CP/M ARE ENCLOSED WITHIN A LARGE FILE OF MORE THAN 500 K CALLED CPMFILES. SUCCESS CAN SUPPORT 6 MEDIA TO SAVE THE DATA. FLP1 AND FLP2 ARE CALLED DRIVE A; AND B; OF 512 K EACH, MDV1 AND MDV2 ARE DRIVE C AND D OF 180 K EACH, RAM1 AND RAM2 ARE DRIVE E; AND F; OF 200 K EACH AND DRIVE G CAN BE DESIGNATED BY THE USER. IT ALLOWS THE READING OF DRIVES FROM THE CP/M COMPUTER. PLEASE NOTE THAT THE DRIVES A: TO F: WHEN YOU ARE USING QDOS, FOR EXAMPLE IF YOU CALL DIR MDV1_, YOU WILL NOT GET THE FILES CP/M THAT YOU SAVED THERE BUT , YOU WILL ONLY SEE THE FILE CPMFILES. THE CP/M FILES ARE ONLY AVAILABLE WHEN YOU ARE IN CP/M MODE.

AS FAR AS RAM1 AND RAM2 YOU WILL NEED A PROGRAM OF RAMDISK TYPE QFLASH, BECAUSE THE RAMDISK MUST BE FORMATTED FIRST UNDER QDOS BEFORE IT IS FORMATTED UNDER CP/M.

2- SUCCESS: CP/M

VERSION 2.2 OF CP/M IS OFFERED WITH SUCCESS , IT IS THE MOST WID
DH!AVAILABLE VERSION. CP/M IS AI DED OPERATING SYSTEM, THEREFORA" T IS
NOT THE EASIEST TO OPERAS&

CP/M FILES ARE IDENTIFIED BY A MAXIMUM OF 8 CHARACTERS AND AN
EXTENSION OF 3 CHRS TO MARK THE TYPE OF FILE. E.G. IF WE HAVE A FILE
CALLED TOTO.COM, IT MUST BE A PROGRAM BECAUSE OF THE EXTENSION ".COM ",
THEREFORE WE CAN LOAD IT SIMPLY BY TYPING A>TOTO . THE " A> " IS THE
PROMPT, IT IS THE DEFAULT DISK DRIVE. IF "TOTO" HAPPENS TO BE IN DRIVE
B THEN WE WOULD TYPE A>B:TOTO TO START THE PROGRAM. THERE ARE SEVERAL
EXTESIONS WHICH HAVE BEEN STANDARDIZED AND ARE LISTED IN THE CP/M
MANUAL.

QUITE A FEW OF THE COMMANDS ACCEPT " WILDCARDS",ALLOWING SEVERAL
FILES TO BE ACCESSED BY ONE COMMAND. FOR EXAMPLE DIR WILL LIST THE
DIRECTORY OF A DISK, BUT IF YOU USE DIR *.DOC , THEN THE DISPLAY WILL
ONLY LIST THE FILES WITH THE EXTENSION ".DOC" FROM THE DEFAULT DRIVE.

HOWEVER IF YOU TYPE DIR G:DB*.* , THEN ALL THE FILES ON DRIVE G
STARTING WITH "DB" WITH ANY EXTENSION WILL BE LISTED. THE CHARACTERS
"*" AND "?" ARE USED AS "WILDCARDS", THOSE FAMILIAR WITH MS-DOS WILL BE
ON KNOWN TERRITORY.

HERE IS A QUICK LOOK AT THE COMMANDS CP/M :

DIR WILL LIST THE CONTENTS OF A CP/M DISK.

TYPE WILL DISPLAY THE TEXT OF A FILE ON THE SCREEN

REN WILL RENAME ONE OR MORE FILES

ERA WILL ERASE/DELETE ONE OR MORE FILES.

THESE ARE NOT LOADED INTO MEMORY , THEY ARE STORED SOMEWHERE !
THERE ARE OTHERS WHICH MUST BE ACTIVATED WHENEVER NECESSARY.

ASM IS AN ASSEMBLER 8080 " DO NOT CONFUSE WITH Z80 "

DOTZ IS A MONITOR-MACHINE LANGUAGE , WITH CP/M IT IS USED TO MERGE
THE PROGRAMMES.

ED IS A TEXT EDITOR BUT YOU CAN FORGET IT AS IT IS VERY
COMPLICATED TO USE.

PIP IS USED TO COPY FROM ONE DRIVE TO ANOTHER OR SEND TEXT TO THE
PRINTER.

STAT WILL PROVIDE INFO ON A DRIVE/FILE/SYSTEM.

IN ADDITION SUCCESS SUPPLIES SOME PROGRAMS.

ANALYSE IS USED TO CONFIGURE DRIVE G: TO BE ABLE TO READ/WRITE
SOME DISKS WITH A DIFFERENT FORMAT.

CAT WILL LIST THE CONTENTS OF A QDOS DISK.

DEFINE WILL DEFINE THE FORMAT OF DRIVE G:.

DEFKEY WILL DEFINE THE KEYS FOR FUNCTIONS.

DELDOS WILL ERASE/DELETE A QDOS FILE.

FORMAT READIES A DISK/MICRODRIVE/RAMDISK TO BE USED WITH CP/M.

RDQDOS WILL COPY A FILE QDOS TO CP/M.

WRQDOS WILL COPY A FILE CP/M TO QDOS.

3-SUCCESS: THE CP/M PROGRAMMES.

THERE ARE THOUSANDS OF CP/M PROGRAMS BUT WHERE ARE THEY ?
VERY FEW NEW CP/M PROGRAMS ARE BEING WRITTEN TO-DAY. AS FAR AS I
KNOW ONLY THE COMMODORE 128 AND THE AMSTRAD PCW ARE USING CP/M TODAY.
THE PROGRAMS CAN BE FOUND ONLY IN THE PUBLIC DOMAIN OR BBS'S.

"QL_DOC WILL GO TO ANY LENGTH TO PLEASE IT'S CUSTOMERS WILL SUPPLY
A NUMBER OF CP/M PROGRAMS READY TO USE WITH SUCCESS FOR THE ASKING.
WRITE ME IF YOU ARE INTERESTED."

WITH CP/M THERE ARE A NUMBER OF PROGRAMMING LANGUAGES:
MACRO-ASSEMBLER Z80, " C ", TURBO PASCAL ALL WORK REASONABLY WELL WITH
SUCCESS. AS FAR AS DATA BASE IS CONCERNED THERE IS DBASE II. FOR WORD
PROCESSING THERE IS WORDSTAR, BUT IT IS NOT VERY GOOD AND IS VERY SLOW.
FOR COMMUNICATIONS ,THE VARIOUS PROGRAMS DEAL DIRECTLY WITH THE
HARDWARE. THERE IS A MULTITUDE OF SOFTWARE IN THE PUBLIC DOMAIN,
WHICH WILL DO EVERYTHING AND NOTHING ALL AT THE SAME TIME ?

4-SUCCESS : USING CP/M PROGRAMS.

ONCE YOU HAVE LOCATED YOUR PROGRAMS, THEY MUST BE TRANSFERRED TO
THE SUCCESS STYLE BECAUSE EACH COMPUTER HAS ITS OWN CP/M FORMAT. YOU
MUST USE THE G: DRIVE TO READ THE DISKS OF A DIFFERENT FORMAT. SUCCESS
HAS THREE DIFFERENT DRIVERS FOR THE G: DRIVE. UNFORTUNATELY THE FORMATS
ARE FOR BBC AND AMSTRAD PCW, OF NO USE TO US HERE. THEREFORE WE HAVE TO
WRITE OUR OWN DRIVER. WE HAVE TO USE THE PROGRAM ANALYSE TO ESTABLISH
THE FORMAT FOR THE DRIVE G:.. BUT WE STILL HAVE TO CHANGE SOME
PARAMETERS. I HAVE BEEN SUCCESSFUL IN COMPLETING A DRIVER FOR THE
FORMAT KAYPRO II WHICH IS ONE OF THE MOST WIDELY FORMAT FOR THE CP/M.
HERE IS A LISTING FOR THIS DRIVER:

```
! DRIVER TO READ ONLY DISKS FORMATTED KAYPRO II
! BY REAL GAGNON FOR QL_DOC 7/20/88
```

```
40 ! SECTORS PER TRACK
```

```
1 ! SIZE OF S AFOR ( 1=1024 BYTES)
```

```
95 ! SIZE EB!DIRECTORY MAXIMUN -1
```

```
192 ! N" ! ERY IMPORTANT , FOR READING EB K
```

```
1 ! NUMBER OF TRACKS IN RESERVE
```

```
!
```

```
! PHYSICAL CHARACTERISTICS
```

```
!
```

```
2 ! NUMBER OF DRIVE
```

```
2 ! SIZE OF THE SECTOR
```

```
10 ! NUMBER OF SECTORS PER TRACK
```

```
40 ! 40 TRACKS
```

```
0 ! DISTANCE BETWEEN TRACK
```

```
1 ! DOUBLE DENSITY
```

```
! NUMBERING OF THE SECTORS
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```


CREATE A FILE WITH YOUR EDITOR ASCII USING THE NAME DISKFMT_DRV ON YOUR DISK SUCCESS. THEN WITH CP/M ENTER :
A>RDQDOS KAYPII.DRV FLP1_DISKFMT_FMT (COPY FILE QDOS -> CP/M)
A>DEFINE KAYPII (DEFINE THE DRIVE G:)

THEN INSERT A DISK FORMATTED WITH KAYPRO II INTO THE DRIVE G: AND ENTER DIR G: . IF IT ALL WORKS YOU SHOULD GET THE CONTENT OF THE DISK ON YOUR SCREEN. OTHERWISE THERE MUST BE A MISTAKE SOMEWHERE.

IF IT WORKS O.K. , WITH COMMAND PIP YOU CAN TRANSFER THE PROGRAMS FROM THE DRIVE G: TO DRIVE A:.

WHEN YOU RECEIVE A GROUP OF CP/M PROGRAMS, GENERALLY THEY WILL FORM A COLLECTION. TO EXTRACT THE PROGRAMS YOU WILL NEED ANOTHER PROGRAM . ONE IS CALLED "NULU.COM" WHICH IS VERY EASY TO USE. LET US SAY THAT YOU HAVE THE PROGRAM "SMALLC.LBR". THIS CAN'T BE EXECUTED DIRECTLY , THE VARIOUS PROGRAMS MUST BE SELECTED INDIVIDUALLY BECAUSE "SMALLC.LBR" MAY CONTAIN POSSIBLY 10 DIFFERENT PROGRAMS. THEY WERE MERGED TOGETHER TO SAVE SPACE ON THE DISK.

IN ADDITION SOME PROGRAMS HAVE BEEN "SQUEEZED". THEY ARE EASY TO RECOGNIZED BECAUSE THE EXTENSION CONTAINS A "Q" IN THE MIDDLE. FOR EXAMPLE "SMALLC.DQC" IS THE SQUEEZE VERSION OF "SMALLC.DOC" AND "SMALLC.CQM" IS THE SQUEEZED VERSION OF "SMALLC.COM". TO UNSQUEEZE THE FILE YOU WOULD NEED A PROGRAM SUCH AS "UNSQ.COM" OR THE EXCELLENT "NSWEEP.COM".

ONCE THE LIBRARY IS EXTRACTED AND UNSQUEEZED, ONE MORE FEATURE NEED TO BE DONE, THE VIDEO SCREEN. EACH CP/M COMPUTER HAD THE BAD HABIT OF HAVING OF ITS OWN TYPE OF VIDEO SCREEN OR TERMINAL. THIS MEANS THAT A CP/M PROGRAM FUNCTIONING PROPERLY WITH ONE MACHINE WOULD NOT NECESSARY OPERATE PROPERLY WITH A DIFFERENT COMPUTER, EVEN IF IT IS A CP/M COMPUTER BECAUSE THE TERMINAL MIGHT BE DIFFERENT. SUCCESS USES THE TERMINAL KNOWN AS VT52 , WHICH IS ONE AMONGST MANY OTHER TERMINALS. IF YOUR PROGRAM WAS WRITTEN FOR AN OSBORNE COMPUTER USING CONTROL CHARACTERS WITH THE SCREEN, THIS WILL GIVE YOU SOME PROBLEMS BECAUSE THE CONTROL CHARACTERS ARE DIFFERENT FOR THE OSBORNE AND THE VT52. GENERALLY THE DOCUMENTATION WILL INDICATE WHERE SOME CHANGES MIGHT BE NECESSARY SO THAT THE PROPER CONTROL CHARACTERS SHOULD BE SENT TO THE TERMINAL. THIS CAN BE DONE WITH THE PROGRAM "DOTZ.COM" INCLUDED WITH SUCCESS. WITH COMMERCIAL PROGRAMS, THE CHANGES ARE DONE WHEN YOU INITIALISE THE PROGRAM BY THE VARIOUS CHOICES OF THE TERMINALS. IT IS MUCH EASIER THIS WAY.

IN WRAPPING UP I WOULD LIKE TO MENTION THAT CP/M PROGRAMS ARE MONOCHROME, THAT IS NO COLOUR . FRENCH CHARACTERS ARE NOT INCLUDED UNLESS YOU CAN PATCH THE FRENCH ALPHABET. THE CP/M REQUIRES 64K OF MEMORY BECAUSE THE Z80 CAN ONLY ADDRESS 64K OF MEMORY.

Britain's Spectrum Scene

by Gil Parrish

I had occasion to be in Scotland recently, and since my tape-based 2068 had just become a "Spectrum compatible" (thanks, Bob!) after the addition of a small Zebra board and a Spectrum ROM chip, I was curious about the Spectrum scene in the United Kingdom. So, I went on my very own "fact finding mission".

First, a few words on the various Spectrum models. You may recall Sir Clive first introduced his Spectrum in England in both 16K and 48K models. The Timex/Sinclair 2068 later sold in this country was very similar to the 48K Spectrum but had a few hardware differences (e.g., the 2068 had joystick ports while the Spectrum did not), and would not run Spectrum machine language software (at least, not without installation of a Spectrum ROM). After the 2068 ceased production, Sinclair went on to introduce improved Spectrum models in Europe including a 128K version, and the "Plus" series. The Plus series models all had 128K and improved sound, with the differences being that the +2 and +2A versions came with a built-in tape drive, while the +3 came with a disk drive.

Upon venturing into computer stores, I discovered that in some the sales help may tell you that "nobody" carries Spectrum items anymore. But a trip to the next store may yield several shelves of Spectrum programs. Mostly, the selection is of inexpensive games on tape instead of disk; some "major" games are available on either tape or disk, but a disk version tends to cost a significant amount more (I don't know why, unless game makers just assume disk drive owners are richer) and may be a "special order" item. Further, while a few tapes in one store I visited required 128K, almost all Spectrum software I saw worked with 48K. Advice to a Spectrum 48K owner considering upgrading his hardware, as given in Your Sinclair, Britain's best selling Spectrum magazine (more on this later), was that "if you only want to upgrade to play better games, there's very little software available that takes advantage of the extra memory or features of the 128K machines." In other words, about all that stands in the way of a 2068 owner and "state of the art" software is an add-on Spectrum ROM chip!

The software available looked pretty decent. For instance, there was a Spectrum version of the recent game "Lemmings"; while popular on the Amiga and other advanced systems, "Lemmings" hasn't been made available for the Commodore 64 or other eight-bit systems-- except the Spectrum! "Robocop 3" also just made an appearance, as has "The Addams Family". Collections of fairly recent software looked popular too; I ended up buying a thirty game package which included Rampage, Hacker, Ghostbusters, EnduroRacer, Aliens, Toy Bizarre, Spindizzy, Space Shuttle, and lots of other titles. So software for my 2068 is not an "impossible dream" in Britain! Yet what I did not find was non-game programs; a single copy of "Tasword Two" was the only software I saw that fit in that category.

Another area I sought to explore was Britain's Spectrum magazine situation. I gather there are several, but Your Sinclair ("YS") certainly appears to be the leader. It is put out by a publisher called Future Publishing; Future puts out separate magazines for the Amiga, Atari ST, Sega, Nintendo, Commodore, Amstrad, PC's, and various other specialty topics (Mountain Biking UK and Today's Vegetarian are some of their other titles), so they must be doing something right. Anyway, I bought YS issue #76, meaning the magazine has been around over six years. It's a bright magazine with a breezy style; it tends to refer to a Spectrum as a "Speccy", and to call its readers and fellow Spectrum owners its "Spec-chums". It contains announcements of new software, reviews of software (with a rating scale based on addictiveness, graphics, instant appeal and "life expectancy"), tips and hints on games (including ways to get extra lives, etc.), a free classified advertisement section for readers with both "for sale" and "wanted" ads, commercial advertising featuring a lot of good software, connecting cables, disk drives, joysticks (including those Kempston joystick adapters that tape-based 2068 owners need when running Spectrum software; disk systems tend to have joystick interfaces already), and perhaps best of all, a cassette of Spectrum games and demos taped right to the cover!

I guess I enjoyed most how tightly focused the magazine was on the Spectrum. It compared quite favorably to a Commodore oriented publication I picked up which advertised itself as "100% C64" but seemed to deliver page after page of British football (soccer) references, reviews of certain Sega and Gameboy titles, rock music trivia, a pull-out rock poster, TV and movie gossip, and plenty of juvenile humor. Alas, YS's tight focus may not last long; in response to a reader comment about the magazine going from 90 pages to 68 pages in the last couple of years, the Editor stated that the larger software houses were beginning to move out of Spectrum software, the smaller houses didn't advertise as much, and to attract more advertisers the magazine was "broadening (its) horizons and looking at life beyond the Speccy in the worlds of film, TV, radio, health and all that kind of thing." Sure enough, the magazine contained some pages of a lovelorn-type advice column for teens, movie reviews, skateboarding tips, etc. And, it should be noted that some of the humor is sufficiently cultural-based that it went right past me. Still, as long as it lasts, the thrust of the magazine is pretty good.

I believe YS used to be available at newstands in Canada, but I gather it isn't any more. Anyone who wants to become a YS subscriber can certainly do so, but it won't be cheap. The normal subscription rate is 26.40 British Pounds (something like \$46 U.S.), but that rate goes up to 40.80 Pounds if you live outside of Britain in the European community, and 53.05 Pounds (around \$93 U.S.) if you live in the Western Hemisphere or elsewhere in the world! But if you're that much of a Spectrum fanatic, I won't stand in your way; just let me know and I'll be glad to supply you a copy of the subscription form. In the meantime, have fun with your Speccy, Spec-chums!

4 Times SCREEN\$

Abed Kahale, 335 W. Newport Rd. Hoffman Estates, IL 60195

Font by ZUNK

Since my last article on the Tandy DMP105 printer, modifications were made to screen dump 4 times the normal size to about a half page worth. Also the print head is now controlled so it can plot to any location on the page. In this case, the font style controls the width; ELITE (12 CPI) & Expanded was used below. One half the vertical pixels were used for one print head pass. The program should be applicable to other printers with proper modifications. To obtain a negative (inverse video) or vice versa add NOT. NOT POINT(C,R-xx)

```
1 REM TANDY DMP105 PRINTER DRIVER by Abed Kahale 1/92
20 RANDOMIZE USR 100: OPEN #3,"LP"
30 RANDOMIZE USR 100: POKE 16092,0: REM No line feed (ot
herwise double spacing)
50 RANDOMIZE USR 100: POKE 16093,32: REM LPRINTs & CHR$s
sent to printer.
80 LPRINT CHR$ 27;CHR$ 23;CHR$ 27;CHR$ 14;CHR$ 18: REM E
lite, Expanded (Elongated!) & Graphic mode.
```

```
90>LPRINT CHR$ 27;CHR$ 16;CHR$ 0;CHR$ 35:REM Locates head (m
argin) at 35 pixels from the left.
95 RANDOMIZE USR 100: LOAD "C.C$*SCREEN$
155 FOR Y=0 TO 175 STEP 4: LET R=175-Y: REM vertical pixels;
twice for 8-pin head.
160 FOR C=0 TO 255: REM Horizontal pixels.
170 LET B=0
180 LET B=B+POINT (C,R-.5): REM divided by 2
190 LET B=B+2*POINT (C,R-1): REM
200 LET B=B+4*POINT (C,R-1.5): REM
210 LET B=B+8*POINT (C,R-2): REM
220 LET B=B+16*POINT (C,R-2.5): REM
230 LET B=B+32*POINT (C,R-3): REM
240 LET B=B+64*POINT (C,R-3.5): REM
250 LET B=B+128
255 IF IN 127<>236 THEN GO TO 255: REM Checks if READY.
260 LPRINT CHR$ B;
270 NEXT C
280 LPRINT CHR$ 27;CHR$ 90;CHR$ 0: REM carriage return.
290 NEXT Y
300 LPRINT CHR$ 30: REM Back to character mode.
```



ZX81 RESOURCES - A O REM GENERATOR

Rene Bruneau 6 March 1992

Reading through some old issues of Your Computer, I found this small machine code utility for creating REM statements of any length. The original listing used Basic to poke the number of bytes required into memory after which the routine was invoked by a RAND USR (address). Because the code was relocatable, I wondered if there was an easier way to pass information regarding the length to the machine code without resorting to Basic input. A long time ago, when we were still meeting up at the North York Library, Peter McMullin had noted that the RAND function conveniently divided a number into Lo and Hi bytes by the computer and stored in SEED (memory locations 16434 and 16435). This is how we can pass information to a machine code program directly from the keyboard without running a basic program. The original listing was modified and an installation program added to transfer the code to any location in memory.

Enter Listing 1, RUN 100 to input the mc data in Listing 2. On completion, enter S, and confirm that the checksum is 5307. You may now delete line 100 to the program end. Save the program several times. On running, the program asks you for the address where the code will be located. Entering a number will relocate the mc. Note that if you intend to locate the code above RAMTOP on a standard machine, you must lower RAMTOP before you load the program into memory. To generate your REM statement, make sure that there is at least 1 basic line in memory, 10 REM, for example, key in

RAND (rem size), ENTER

where 'rem size' is the number of bytes required,

LET L=USR (mc location), ENTER

When you LIST, there is your REM statement on line 0.

HEX LOADER FOR THE ZX81

This hex loader is pretty standard. It allows you to enter as many hex numbers at one input as you wish. When you have finished entering all the code (in hex), press the "S" key to stop.

```
1 REM (FILL THIS LINE WITH ENOUGH
CHARACTERS TO HOLD THE MACHINE CODE
10 LET X=16514
20 LET A$=""
30 IF A$="" THEN INPUT A$
40 IF A$="S" THEN STOP
50 POKE X,16*CODE A$(1)+CODE
A$(2)-476
60 PRINT AT 11,7;X;" ",A$(1 TO 2)
70 LET X=X+1
80 LET A$=A$(3 TO)
90 GOTO 30
```

From Your Computer magazine May '83
p67

GFC

LISTING 1

```
1 REM ---54 SPACES---
10 PRINT "ROUTINE ADDRESS"
20 INPUT ADD
30 FAST
40 FOR X=0 TO 53
50 POKE ADD+X, PEEK (16514+X)
60 NEXT X
70 SLOW
80 STOP
100 LET X=16514
110 LET T=0
120 LET A$=""
130 IF A$="" THEN INPUT A$
140 IF A$="S" THEN PRINT "CHECK SUM = ";T,P
150 POKE X,16*CODE A$+CODE A$(2)-476
160 LET T=T+PEEK X
170 IF PEEK 16442<=2 THEN SCROLL
180 PRINT X;" ";A$(1 TO 2)
190 LET X=X+1
200 LET A$=A$(3 TO)
210 GOTO 130
```

LISTING 2

```
16514 ED 4B 32 40 C5 21 06 00
16522 09 E5 C1 21 7D 40 CD 9E
16530 09 3E 76 EB 77 3E 17 C1
16538 C5 04 2B 77 0D 20 FB 05
16546 20 F8 2B 3E EA 77 D1 13
16554 13 7A 2B 77 7B 2B 77 3E
16562 00 2B 77 2B 77 C9
```

PRINT FACTORY® graphics More Pics & Lower Prices!!

Collection#1	Collection#2	Collection#3
Miscellaneous	Animals	Digitized Misc.
Men/Boys	Funny	Female Faces
Women/Girls	Transportation	Male Faces
Digitized Misc.	Sports	Funny
	Computers	Animals
	Holidays	Zodiac Signs
	Food	Holidays
	Xmas	Transportation
		Xmas

*** [3 NEW COLLECTIONS!] ***

Collection#4	Collection#5	Collection#6
Miscellaneous	Men/Boys	Funny
Digitized Misc.	Women/Girls	Animals
	Male Faces	Xmas
	Sports	Transportation
	Female Faces	Computers
	Holidays	

Over 150 pics/collection - 1077 pics in all!

Only \$7.95 ppd. for each collection.

Available on tape or 5 1/4" DSDD LARKEN
or OLIGER disk - please specify.



Send check or money order to:
John McMichael, 1710 Palmer Dr.,
Laramie, WY 82070



SINC-LINK

ZX-81

LOCKED OUT? HOW TO RECOVER!

T/S 2068 AND THE LARKEN AND OLIGER DISK DRIVES.

Submitted by Donald S. Lambert

Letter to George Chambers 03 02
1992.

In all my computing, I have been trapped by a problem three times. Of course it is all my fault but maybe there is a way out. And that is on each of these times I have either tried to SAVE to a disk drive that I didn't have or on two occasions I tried to SAVE when the disk interface to disk drive ribbon cable was disconnected. Try as I might I have found no way to abort the SAVE and have had to turn off the computer and lose the file. It might be possible to plug in the disk drive ribbon cable but there is always the possibility of zapping something. And while I have a Larken disk interface with a RAMdisk that is full and besides once you have committed to a disk drive the computer hangs onto that drive until it is used. What do you do in that situation?

George replied 03 14 1992.

I have been looking at your letter. You asked about how to break out of a computer lock-up when you try to make a SAVE/LOAD to a non-existent drive. You are correct, I would never plug a drive in while the computer is powered up.

I thought I probably knew the answer so I shut off my computer, disconnected my drive 0. Then I re-powered and tried saving an Mscript file to drive 0. Well all the other drives spun of course, but not the disconnected drive 0. The computer was hung up. I then did

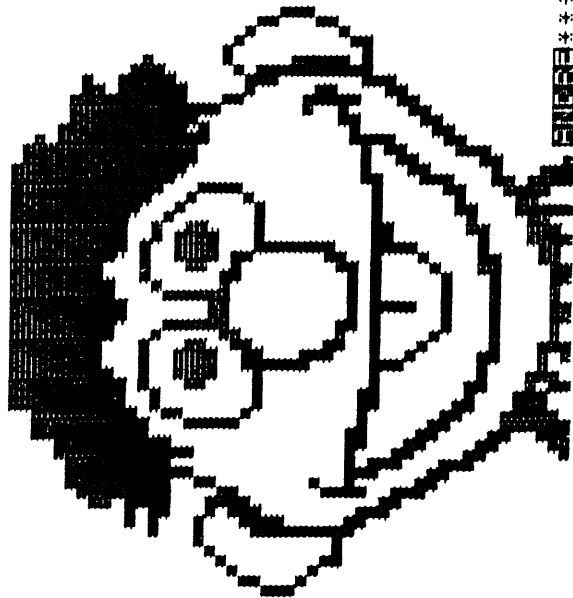
what I would usually do. I pressed the NMI- button and then the A key. As I expected the computer broke out of it's hung-up condition with an error report at the line in the program where the SAVE effort occurred.

That is pretty simple. I thought everyone knew that. To continue, if you inadvertently press the NMI- button you can recover from it without harm by pressing the ENTER key. Also, if you are trying to breakout of a program that is heavily protected, and the NMI- and A key routine does not work, try pressing the NMI- button and then press the 1 key; i. e. do an NMI- type save to a protected disk. You will get a "Protected disk" error report, and you will be back out of the lockup. Sometimes, depending on the nature of the protection the computer simply crashes and you get the Sinclair logo. I use this method particularly with Spectrum games, which for the most part are heavily protected.

Don: I had never considered the use of the NMI- button so I guess I am not everyone.

I have an idea to try out, since both the Larken and the Oliger use the NMI- button will it also work the same on the Oliger? It works with the Oliger except that the A key does nothing but the 1 key will allow you to use the BREAK function and get "D Break - CONT REPEATS 0:1". I don't know if anyone ever gets locked out but this is one way to recover. But here is a possible way out of a dilemma that would otherwise result in a loss of data. 0/0.

ALL MAIL ON ZX81 TIMEX 1000



I will send (FREE) to you the PROGRAM and DATA of the picture(s) seen in this issue if you send me the name of the character depicted. See also bottom of page 2.

SINC-LINK

TO

ZX81

NO. 3

10 YEARS LATER

ALL :
THE NEW NEWSLETTER FOR THE
SINCLAIR ZX81 / TIMEX 1000.

ALL YOU SEE, AND WILL SEE, HAS
BEEN CREATED WITH, AND ONLY, A
ZX81, A 16K RAMPACK AND A TIMEX
2040 PRINTER.

I QUIT!

Yes! I quit! After only three issues I quit answering letters individually. The amount received now forbids such practice. From now on answers will be part of ZX-91. The answers will then benefit all readers and I will not have to repeat myself over and over.

This month's title must have shaken you. Devilish am I not? But trustworthy, ZX-91 will come out every month for at least a year as stated in the first issue.

IF YOU WISH TO RECEIVE
A PRINT-OUT, A CASSETTE,
AN INFORMATION, A REPLY
OR THE NEXT MONTH ISSUE
OF THIS NEWSLETTER SEND
A SELF-ADDRESSED ENVELOPE

TO: ANDRE BAUNE
304 SCOTT,
CHATEAUGUAY, QUEBEC
CANADA J6J 4H5

KNEE-USE-M-SIRS

Can you read the above title? When I started ZX-91, I had piled up topics, articles, print-out and pictures to fill the pages of this newsletter for at least a year. But I soon realized I had forgotten something which would make this newsletter livelier. And if you read carefully the above title you will find out what it is all about.

GOOD SERVICE:

I received what I ordered in reasonable time limits for the price marked from RME Entreprises, from John McMichael and from EMSoft. Future buyers look them up in the SUPPORTERS' List.

Thank you Peter for your note and the subsequent letter. Proper actions have been taken. Your bank should have contacted you by now.

If any of you have any problem with cashing my checks, please let me know your bank manager's name, the bank's address and phone number. I will immediately take the proper actions. I am paying my bank for this service. They deliver.

OLYMPIC'S GOLD MEDAL:

Did you buy COMPUTER MONTHLY of March? Have you read the chapter about ZX-91 in the T/S SURVIVAL column?

To appear in a prestigious international magazine is a GOLD MEDAL!

A gold medal that WE WON, US ALL, vendors, clubs, leaders, programmers, newsletters' editors, users and believers in the ZX81/Timex 1000.

No words could express my emotion and my gratitude to Bill Ferrebee of Mountaineer Software and to the editor(s) of Computer Monthly. I am wordless!

UN PEU DE FRANCAIS:

Merci a Mike Feleerski pour sa lettre. If you tried to startle me, you did! Do not forget to reply to me about your journal.

27

FROM PAGE 3...

3.A.3.E. #1:

I take this space to thank you all who sent me a self-addressed envelope. But please if you are not from Canada do not affix an American stamp on the envelope because I am mailing from Canada.

3.A.3.E. #2:

Thank you all who financially helped me with the cost of sending them back their next issue of ZX-91. It helped me greatly. Needless to say that my budget is busted. But I am hopeful it will be within reasonable limits in the forthcoming months.

SUPER THANKS:

To my great motivator Donald S. Lambert of ZXir Olive Ridge, to Arnold Nieuwenhoff of Sutton MA and to David S. Leech of Byte-Back.

You gentlemen have helped me beyond expectations. I'll have to do something in return.

BAD NEWS

My letters to WORLWIDE SOFTWARE of England and to INSPEC LTD of Ireland were returned with mention SOME AWAY. If you have the new address of these two vendors, please send them to me.

ELECTION TIME:

Malcolm Post has replaced Derruck Turner at the helm of SEPTUS. Best of luck Malcolm!

FACTS:

ZX-91 is a monthly newsletter primarily devoted to the ZX91 and the Times 1000/1500. It is sent free to any one who will send me a request in a self-addressed envelope. Clubs and/or newsletter's editors who exchange newsletters are exempted of the preceding procedures. Vendors (I used the term vendors to conform myself to the remark of Nazir Pashtoon, president of CATUG.) are requested to drop me a word (or a postcard or a flyer) every four months (or less) in order for me to keep an updated SUPPORTERS' list.

--- PAGE 5

FROM PAGE 5...

Visit the fleamarkets and garage sales, you might find some computers there. I know, I did find some.

About your newer II-99 keyboard problem, I do not have a final answer yet. I will try to get such a keyboard and see if it can be wired up. Or maybe some readers already have an answer? I know that the older model works fine because I wired one. The only thing to report is that the ENTER key is up one row. I linked the keyboard to the circuit board with ribbon cable. It makes a neat job. More about keyboards next month.

About software, you say software for the ZX91 times 1000 is hard to get. Well, I sent for catalogs from the dealers listed herein and I received impressive lists of software available and at cheap prices. I have joined clubs and they have impressive lists of public domain software. Let yourself be tempted! Go for it! It's there!

Also for the NEVER than new, in the coming months I will try to put on the market some of the programs I have created. You have seen the quality of this newsletter and the graphics. You can expect the same from my programs.

Yes! I do have a library of programs. The shorter ones are becoming available as print-out in this newsletter. For the longer public domain programs I don't have a satisfactory solution yet.

Pffess! I hope I covered up everything. Surely mishaps will happen but let's look on the bright side! let's find corrective measures! Let's help each other!

ONE LAST

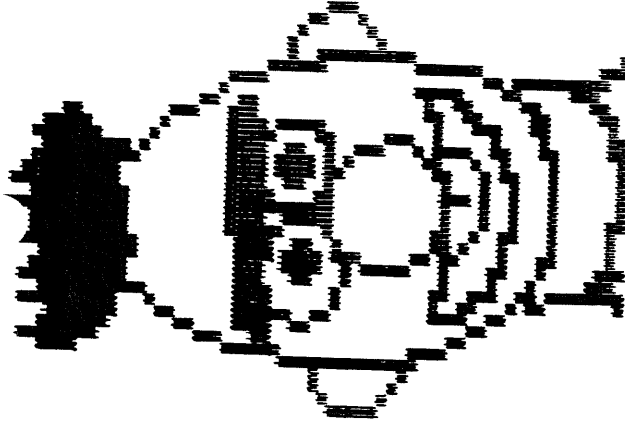
Phil of Sunset Electronics is looking for an EPROM BURNER. You have one? Look at his address in the SUPPORTERS' list.

ALSO:

Welcome to ZX91, the amateur radio club using Sinclair/Times computers who is still going since many firms put out the first copy in the early days of the ZX80.

A great welcome also to the Washon Island Sinclair Times Association.

ANDRE***



You like this picture? See page 1 to see it FREE on your own screen and printer.

FROM PAGE 4...

FRENCH QUEBECER SLIP:

Computer Classics is from CARBOL MC. In issue 3 I wrote the name with a french sound, sorry Can.

MORE AND MORE.

I received a 2 pages letter from Bill Parrish a letter full of subject's questions and grievances. I will try to answer them here and now. And as incredible as it may be I only have good news.

From Hacker Electronics (the 1\$ and 2\$ computers) you have received a refund. That is good. And there is still a possibility that they might have more computers for you later.

Now! What about trying to get some computers from other sources. Contact the dealers listed in the SUPPORTERS' list. Some of them must have some 'returns' or some second hand computers.

Get in touch with groups like ISTUG or SEPTUS. They have sales or auctions where people from out of town can buy or bid.

--- PAGE 5

Yes! The ZX81 (or Timex 1000) is the best computer to learn about computers and about programming in BASIC or in machine code. And we shall learn why in the following chapters.

To qualify as proper material, any book, software or computer must meet certain requirements which are neatness (or clarity), simplicity and feedback.

NEATNESS

It is easy to understand why for a beginner it is important to clearly see and to be able to identify what is shown to him. Oftenly it is regarded as a drawback for the ZX to accept only one instruction per line. But for the learner it is an advantage. The program line remains neat and short, not complicated by added instructions. In other words, easier to read and to understand by the beginner.

Another neatness of the ZX is it's way of listing programs on screen, one page at a time. You don't have to figure out the length of the following lines in order to indicate the line at which to stop. You simply type LIST + line number. The computer then prints out one screen page starting at the line called and stops. It's clear! It's neat! It makes the screen easy to read for the learner.

Now, what about the automatic spaces on each side of the keywords? That is neatness! The keywords seen outstanding are easier to note and to remember. It makes the screen much clearer to read.

All the mathematical functions are spelled on screen 'SQR', 'PI', not represented by a symbol. The beginner has already other symbols to learn. Why add mathematical ones? He is learning computing not math. So being able to read the name on screen is an advantage for the beginner.

---> PAGE 8

CPVETOGEPAN

EV PRIZ COS KEVM IVOBX

CHPEIFY OPHOX IV

FROM PAGE 7...

Here is a review of the four reasons why the ZX is the best in neatness. 1) One instruction per line. 2) Listing only one page at a time. 3) Automatic spacing for the keywords. 4) Spelling of the math functions.

These features keep the screen neater and from there easier to read and to understand by the beginner.

SIMPLICITY

The simple format of the error message: Two numbers separated by a slash. What can be simpler? The first number gives you the error code and the second one is the line number in which the error was detected. No fancy nor confusing sentences. The information is kept simple but complete.

The second (simple) feature is the limited amount of commands and/or functions to learn before you can actually make a running program. All the commands necessary to make a program are there. The extra unnecessary one are not there to confuse you. This is simplicity.

The third feature of simplicity of the ZX is the limited amount of pixels to be managed on screen: 64x44. Compared to the 256x192 or 640x200 or more. A beginner does not need all that complexity to learn how to make a picture. It impedes his learning or worst it will discourage him. This is another reason why the ZX is the best, it has a small amount of pixels to manipulate and you still can make fantastic pictures. Simplicity equals easy learning.

Rapidly we will review the three simplicities of the ZX. 1) The error messages format. 2) List and only the necessary commands. 3) The small amount of pixels for making pictures.

FEEDBACK

The feedback is surely the most important quality in the learning process. It is the recognition and the evaluation of what the student has done. It allows the student to bolster his self-esteem when he responds properly or it should give him hints or help on how to correct his mistakes.

--- PAGE 9

FROM PAGE 6...

When an error is detected the ZX will report an error message indicating the type of error and the line in which it was detected. This informative feedback is very useful for the learner, it is a clear indication of what to look for and where to look. In short it gives him the WHAT and WHERE.

The second feedback is the syntax error cursor. This is a unique feature of the ZX which puts him in a class by itself. The computer will not accept (ENTER) an incorrectly built instruction. More, it will also place an inverse S after the error. The ZX is almost the perfect teacher indicating mistakes before they bug down your program.

And finally the slowness of the ZX. Yes! The most cited fault is the ZX best advantage in the learning process. Are you going to correct a deficient way of programming when the speed of the execution of the program is not affected? Are going to force yourself to learn a better method to make faster animation on screen when there is no noticeable difference on screen? But with the ZX the execution of a program will be slower. The feedback is right there in front of your eyes. So slow that you can see it. Learning and using the best techniques of programming become an easy task because the results are noticeable.

Here is a review of the last chapter about feedback: 1) The errors report provides a simple and efficient feedback. 2) The syntax error cursor feeds us back with the inverse S pointing to a faulty instruction. 3) The noticeable difference in the speed of execution with or without the best techniques of programming.

Now you know why our computer is the number one of all the computers for learning computing. If anybody challenges you about this fact you are now well prepared to defend our computer.

A small car or a jumbo jet will take you from New York to Los Angeles but 99% of the people can learn to drive a small car. I do not know how many will learn how to fly a jumbo jet? Think about it. For learning the ZX81, Timex 1000 is #1.



A Tape Index.... By Les Cottrell

In the last issue of the newsletter it was suggested that we should write with our problems and experiences. Here is one of mine.

I started looking for a program recently that wouldn't run properly from disk. I had started to index my tapes a long time ago by using the CATALOG function in MSCRIPT. That required sorting thru each tape listing. I wished that I had a program like the one I use for disks - Bob Mitchell's DISK INDEX program. I decided that I could accomplish what I wanted by taking the MSCRIPT file and running Bob's "mssort" program. First I added any tapes I hadn't previously run into the MSCRIPT file. They come in looking like:

```
*****
TYP NAME      START LENTH PROGL
BAS Beta Basic      20  1568  1529
BIN BB             46680 18688
BAS BETABdemo        1  1521  1489
BAS BBdemo          9800 22345 22313
```

```
58 Beta Basic
58 BB
58 BETABdemo
58 BBdemo
```

After the names are loaded into MSCRIPT the header and other unwanted info is removed and it looks like the list under the first one. My list was 640 programs long after removing some of machine code files.

```
26 Belltower
36 Belltower
58 Beta Basic
```

Since mssort can only handle 200 lines per pass I divided my file into four sections and sorted on column four. Sample result at the left.

```
Belltower      26,36
Beta Basic      58
```

After the first four passes I used MSCRIPT block file moves to put all A thru M files from each group together, N thru a, b thru p and then q and on. Each of these files was mssort-ed again until my whole file was in alphabetical order. Eight separate runs were used. The mssort part was the quickest part of the operation. (Thanks Bob)

The completed file was scanned and a column was selected so no file name would t extend into the tape numbers. The final file currently looks like the last example. I may add some remarks on the right side of the page later.

The final file is 367 lines long after eliminating duplicates. And I can easily locate any program I have on tape. The bad news is that the program I was searching for has the same problem as the disk version.

323 1/2 N. Church Street
Bowling Green, OH 43402
February 28, 1992

Dear Bob,

Well, as you can probably tell, I got my new computer. My dad splurged a bit: he got me a 386SX portable with Windows. This is being written on Microsoft Write for Windows (not Word; Write is their bargain word processor). As of yet, I haven't really bought any other programs for it, just a printer cable and a mouse. I will probably get some stuff for it later though - you know, math stuff and an assembler or compiler.

I had an idea about how to read DOS disks on our other computers without getting a CRC error every track. There are 2 possibilities. Either you can format a disk on the 2068 and artificially put the DOS first track on it - yes, DOS can read a disk properly formatted this way - or you could write or modify code to read the disk directly.

We discussed earlier that DOS disks only have 9 sectors on them, which is what causes the CRC error - LKDOS expects 10 sectors on every track. DOS will not format a disk with 10 sectors on it, that just isn't an option in the FORMAT program. As it turns out, DOS will read and write properly formatted disks with 10 sectors per track. I've really only tried it with 3.5 inch disks, but it should be true of 5.25 as well.

Of course, that raises the question of what is proper format. If we are talking about a 5.25 disk with 10 sectors per track and 40 tracks, that means we have a total of 800 sectors instead of 720, or 400K instead of 360K. Not a bad deal anyway, but the number 400 becomes significant in another way.

Let me just give it to you immediately. The boot sector, the first 512 bytes of track 0, would have to contain the following information (X means it doesn't matter):

0:	235	60	144	X	X	X	X	X
8:	X	X	X	0	2	2	1	0
16:	2	112	0	144	1	253	2	0
24:	10	0	2	X	X	X	X	X

The rest of that sector is "don't care". I haven't figured out why the first 3 numbers are there, but they are apparently necessary. The next 2 sectors from 512 to 1535 are the FAT. On a blank, formatted disk this would have the numbers 253, 255, 255, and the rest all 0. Since we told it there were 2 copies of the FAT, naturally 1536 to 2559 are exactly the same as 512 to 1535. The directory would start at 2560, run to 1024 on track 1, and should be all 0s on a blank disk. If you do this on a disk with 10 sectors per track, everything should work out.

The other option is more interesting, though a little more complicated. Naturally, the routines in the cartridge can't be changed to read only 9 sectors. I have seen that if you ignore the CRC error it really doesn't matter, but of course you can't tell if there is a real error or just 9 sectors. We could get around this by modifying the routines from Hcode, the ml part of the LKDOS format program. They are not in ROM, and include all of the appropriate routines: read, write, seek, step, and select (plus format, of course). The only problem is making sure you have the right version of the altered code - Hcode.Cl for you, Hcode.Ca for my AERCO interface. So far, I have only found 4 locations that would need to be altered (I think). In the Larken version, these are at 9CE2h, 9CFBh, 9DD7h, and 9EBDh. Of course, I'll check whether the corresponding changes (at different locations) work with the AERCO version. If they do, then we could simply use the altered versions instead of George's code. Though the fact that there are at least 3 versions of the code - one for your Larken i/f, one for my AERCO, and one I think for Zebra - would make things more difficult.

As you can tell, I have both the Larken and AERCO versions of Hcode, but not any others. I know Larry made at least one other cartridge, but he may have made more, and each one for a different disk i/f would have its own version of Hcode. My Hcode is larger than yours, since

AERCO used 16-bit port addresses, but I would imagine the other(s) would be almost identical to one or the other of ours.

Oh, naturally I have hooked up my good printer to this machine and will be using my old Fasttext with the 2068 from now on. I had to tell both the computer and the printer that it was an IBM Proprinter instead of a Panasonic, but that is only a minor inconvenience at best. I had more trouble hooking up the mouse for Windows - I accidentally erased the mouse driver and had to use the 2068 to recover the file. That's why I know reading or writing a 9-sector disk is nothing more than an inconvenience under LKDOS read track and write track. Write track didn't even tell me there was anything wrong!

I'll check out whether there are only 4 changes to Hcode for AERCO before I mail this, so you should find out in the P.S. if that is everything. That will be it for now, though. George has my corrections to the AERCO printer software to make it work with a Fasttext.

Sincerely,

Steven V. Canhouse

P.S. Yes, at least for the AERCO version of LKDOS FORMAT, there were only 4 places that needed changing. I had some old IBM 5.25 disks that I wanted to copy the programs to my laptop, which takes 3.5 inch disks, and so used the copy disk command on the modified FORMAT to copy from drive 1 (the 5.25) to drive 0. It didn't complain (no CRC errors or anything) and neither did the laptop. I was able to copy PC-Write (a shareware word processor) and ISETL (a language) to my hard drive. Nothing to say about actually reading files from one computer on the other, though.

P.S. The DOS information and especially the data earlier is only for 5.25" diskettes. I can of course provide it for 3.5 (and be more certain it is accurate) if you wish.

READERS!

SEND IN YOUR LETTERS, TIPS,
QUESTIONS AND ANSWERS. WE
PRINT ADS FOR FREE TOO!

TK2 - NETWORK - DATA_USE - \SER - ALTKEY

by Hugh H. Howie

Recently I asked you to do some work with TK2 for your self. I also have been trying things with it, and have found a use I never expected to find.

Many may ask what value the DATA_USE function may have. Certainly once the defaults are designated it may be thought that that is the end of that, but that is not necessarily so; there are many ways in which DATA_USE may be utilised. The same as there are many ways any given TK2 function may be used. Have a look at what follows.

I am going to show how the commands in the above header to this article can all be combined, and used with effect.

I have too many disks lying around with too much space wasted in too many backups. I know we should all keep a back-up of everything but sometimes things get out of hand and that is what happened to me, I had too many disks chasing the same information, so I decided to consolidate the contents. And that is when my trouble started. This is my set up:-

QL #1 -Networked To-	QL #2
1 x 3 1/2 720 Drive	1 x 5 1/4 720 Drive
1 x 3 1/2 1440 "	1 x 3 1/2 720 "
Printer	1 x 5 1/4 360 "

I used to have more disks drives, but while my fingers wandered idly over the noisy keys, my feet were getting caught up in pedals, so I removed some drives from each QL. Sure cut down on the draught from all them spinning drives.

My problem started when I wanted a hard copy of the directory on a 5 1/4 disk which meant using #2 QL through #1 QL. I know there are many ways to do this such as changing the drives around etc., but I didn't want to do that.

A lot has been written about the QL Network, but I was never able to make too much sense of a lot of it, which resulted in my having to devise my own methods. This is part of my experimentation. (I like that word - it takes up almost a whole line)

I tried to use the DIR \SER and the WSTAT \SER in various ways (remember told you about them last time out?) Next I thought if I used DATA_USE command I could change the default from #1, to #2 machine, thus:-

DATA_USE N2_FLP1_

It worked! I was now able to get the stats of the disk in #2 flp1_ to the printer on #1 machine.

Problem was I still wanted to be able to use the default in #1 QL, so I decided to put the DATA_USE on an ALTKEY. In fact I could put two commands on Altkeys, one to put me into #2 QL and another to bring me back to #1.

ALTKEY "a","DATA_USE N2_FLP1_" To put me over, and :-

ALTKEY "k","DATA_USE FLP1_" to bring me back to #1.

In between the two Altkey commands, could use "WSTAT \SER" or "DIR \SER" and this would allow me to print from #2 to the printer on #1. All being done with a minimum of key strokes. (I don't like to HIT my keys, they haven't done anything to hurt me!) (yet)

So now when I wanted a printout of flp1_ in #2, all I had to do was press "<ALT> a" and type "WSTAT \SER" to get my copy, then type "<ALT> k" and I was back to flp1_ of my #1 QL.

Now I know that it is not often I want to do this sort of thing, but if I cared to, I could incorporate the two ALTKEY commands in my boot program and they would always be there when I required them.

So there you are, I have done what I set out to do. I have used TK2 to set up the Network and the Data_use and Altkeys, and used them all from #1 QL, to send a DIRectionary from flp1_ of #2 QL, to the printer on #1 QL.

Now what else can I adapt to my needs?

032692

11 01 1991.

A CHALLENGE TO PROGRAMMERS — DON LAMBERT

Want to play GOD and create a solar system? Want to learn about astronomy? Want to set up a system to write a story about? Want to have something more challenging than some of the computer games? This is it. This is a real challenge to a programmer. I hereby toss this out for someone to take it from here. I would tackle it myself but I have too many unfinished projects now and I am not a programmer. But I am sure that if I tackled this I would learn a lot about programming and also astronomy. I received a MSDOS ASCII file disk from Kirk and I converted it with a program from George Chambers (Toronto) to a MSCRIPT file and then with a program from Bob Swager converted the MSCRIPT file to a TASWORD file. Some of the material may be missing but that is what happens when computers convert things. You can either get the material from Kirk or myself. Same price. Except that I can only supply it in the Larken DOS since I do not have TASWORD on Oliger and I would need a way to convert it to Oliger from Larken. Sorry about that, nothing is easy all the time.

11 01 1991.

When I saw the letter in Analog I could not resist writing and then sending a disk in a SASE disk mailer so that I could get a copy of the program in ASCII files. I do not have programming ability but I hereby challenge one of you to take it from here. I do not know if the T/S 2068 has enough memory for the program but I know that the QL does. For those of you that want to work from the original material write to Kirk Thompson and get it from him. His price is send in a disk and SASE mailer for the disk to be returned or else send him Five Dollars.

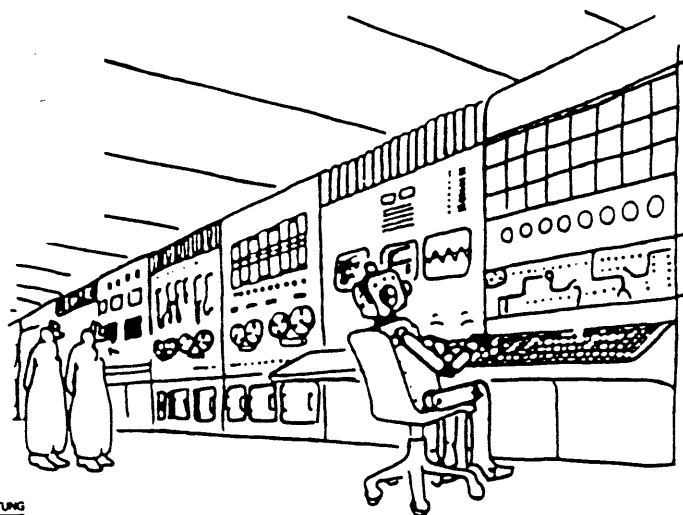
Why did I wait so long to get the disk? I didn't, as soon as I saw the letter in the magazine I got a letter in the mail. You see I am slowly getting caught up in my reading. That is a sign of a successful retirement. Always have something to do and never get it all done.

Cont.

"GEORGE"

"He is the only one
who understands
how it works."

ERIK LIEBERMANN IN STUTTGARTER ZEITUNG



SINC-LINK

QL

ANALOG SCIENCE FICTION/SCIENCE FACT. MID-DECEMBER 1990.

I enjoyed your recent excursions into fictional world creation (Gillette - July '89, and Barnes - March '90 issues). But one thing which neither of them touched on in any depth was software assistance in the design process. I remember a program by Stephan Kimmel, in June, '83, in the now - defunct CREATIVE COMPUTING, called "World Builder." It was written in Microsoft BASIC for the australopithecine Radio Shack TRS-80 Model 1, and dealt with the "physics and astronomy" (to quote the lead-in to Barnes) of the process.

I adapted it for my Neanderthal Heathkit CP/M-based H-909 system. It includes facilities to select 1 of 34 known stars or "design" your own, create a planetary system, add up to 10 satellites per planet, and draw some conclusions on indigenous life and human habitability. Of course, the assumptions it uses are simple, but it could be used as a base for more sophisticated effort. Even so, it is quite interesting to tinker with.

If any of your readers are interested, I would be happy to copy it for them, along with a short ASCII documentation file. I just ask that requests include a FORMATTED disk and postage - prepaid return mailer. I can read/write a host of CP/M (though the Western Digital controllerchip in my machine limits a few, such as Kaypro, to single sided) and PC-XT formats. Sorry, but I can't handle 3 1/2 in. If in doubt write me.

KIRK L. THOMPSON
EDITOR, THE STAUNCH 8/89'ER NEWSLETTER
P. O. BOX 548
#6 WEST BRANCH MOB HOM VIL
WEST BRANCH, IA 52358

QL YOU MIGHT LIKE TO KNOW !!!!
HOWIE , IN THE PAST FEW ISSUES HAS BEEN GIVING SOME
HINTS ON THE BENEFITS OF USING TK2_EXT WITH YOUR TOOLKIT II.
SOME OF THE HINTS ARE DEPENDANT ON THE VERSION OF THE
TOOLKIT.

FOR INSTANCE I FOUND THAT I COULD GET A LISTING OF A
PROGRAM BY TYPING : COPY xxxxx TO SER1

REMINDS ME OF THE LLIST COMMAND OF THE 2068.

LOUIS LAFERRIERE

Q L I P S

by Hugh H. Howie

This little snippet of a program is so beautiful I cannot understand why we have not found it long before this.

Remember all those windows we would like to get rid of when we change programs? The ones that overlap on the new windows? We do not want to do RESET as it might spoil something else; well this is called WINDEX, and just like the name of that wonderful blue stuff you are given in the spring and told GIT OUT THERE!!!! This little beauty will clean up all those fancy windows for you; well not really all, but most of them.

It will not get rid of those EXEC windows but is a dandy for the S/B (watch your langwidge) F1 windows.

It is not in the Library yet as I only received it recently, but it's gonna be one o'dem days.

WINDEX _BAS

```

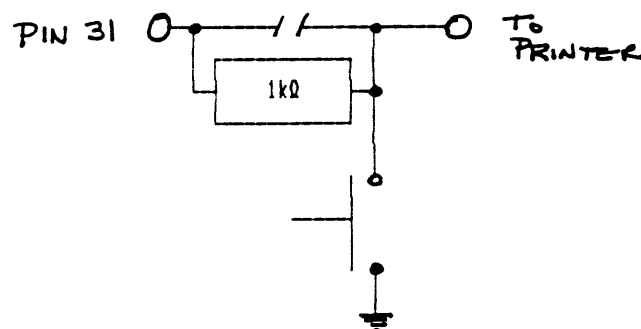
100 REMark =====
105 REMark ***** WINdex_bas *****
110 REMark =====
115 :
120 REMark Restores F1 Screen WINDOWS
125 REMark #0, #1 and #2 to their
130 REMark Normal values if they are
135 REMark "Trashed" by an unruly program
140 REMark Contributed by Ed Kingsley
145 :
150 REMark Extract and add the PROCedure
155 REMark To your own graphic programs
160 REMark & Games to clean your WINDOWS
165 :
170 REMark LRUN WINdex_bas
175 :
200 windex
202 :
300 DEFine PROCedure windex
302 REMark -----
304 MODE 4
306 WINDOW#0,512,50,0,206:PAPER#0,0:INK#0,5:
    CSIZE#0,0,0
308 OPEN#1,con_256x202a256x0_128:PAPER#1,2:
    INK#1,7:BORDER#1,1,255:CSIZE#1,0,0
310 OPEN#2,con_256x202a0x0_128:PAPER#2,7:
    INK#2,2:BORDER#2,1,255:CSIZE#2,0,0
312 CLS#0:CLS#1:CLS#2
314 END DEFine windex (022192)
    
```

PRINTER RESET

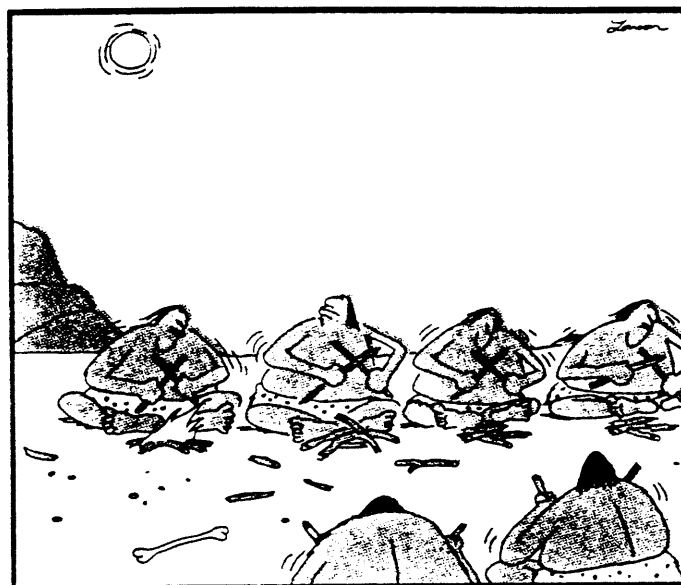
If something goes wrong during the print-out of a document on a big printer, often the only way to stop the print operation is to switch the printer off. This may be a useful, but certainly not an elegant, method. A reset button, on the other hand, is.

Nearly all printers with a Centronics interface have a reset input at pin 31 of the centronics connector (consult the manual). That input is used in many systems to set the printer to a defined starting state and at the same time to empty the buffer.

The input may, of course, also be used to connect a reset switch to. The diagram in fig. 1 shows how such a switch may be made quite easily. The 1 kΩ resistor prevents a short-circuit at the computer output when the printer is being reset.



Retyped by R. Bruneau from ELEKTOR ELECTRONICS July 1989



Early clock-watchers

QL

Tight security

Andrew Ho,
Leicester.

2X-31

IDEAS FOR maintaining program confidentiality have been published before in Basic, requiring the user to input a secret codeword before the program will run. Such security checks are easily bypassed in Basic by the use of Break or Stop keys. This code routine cannot be circumvented. The machine code is stored in a Rem statement containing 50 characters, as the first line of the program. Since many hexadecimal machine-code loading programs have been published, it is not necessary to repeat one here.

The routine starting at 16516 prints a screen prompt, then calls the key-scan subroutine in ROM many thousand times, seeking for a secret combination of keys to be pressed. If this is not found within 27 seconds, it jumps to the New subroutine in ROM and erases the program, thus preventing unauthorised access. To be effective, this idea should be incorporated in a program which auto-runs-on loading from cassette.

The line following the program line containing Save should enter the machine-code routine at 16516 immediately, as illustrated in the sample program. The routine will work in both 1K and 16K machines, although it will probably find more use in 16K programs which might store confidential information. The screen is cleared and the counter reset before entry to the main program, so that further copies of the program will still feature this security check.

The secret codeword may be two, three or any number of any keys to be simultaneously pressed, thus making it impossible for uninformed users to breach the security check.

Your own secret codeword can be used if you understand how the key-scan routine operates in ROM. For those who do not, here are some possibilities:

16535	16536	codeword
9D	D9	AHO
7D	F1	ASM
7B	F3	WM
AE	E6	Shift X 7J

Address	Hexa- decimal code	Explanation
16514	FF 63	Set counter
	3E 28 D7	
	3E 34 D7	
	3E 29 D7	Print screen prompt
	3E 2A D7	
	3E 0F D7	
16531 loop	CD BB 02	Call Keyscan
	11 9D D9	LD DE, secret code
	AF	XOR A
	ED 52	SBC HL, DE
	28 0D	JR Z, +13
16542	2A 82 40	LD HL, (16514)
	2B	DEC HL
	22 82 40	LD (16514), HL
	BC	CP H
	20 EB	JR NZ, loop
	C3 C3 03	JP NEW
	CD 2A 0A	Call CLS
	3E 63	LD A, 63 Reset
	32 83 40	LD (16515), A Counter
16563	C9	RET

Andrew Ho's protection program.

9000 SAVE "program name"
9010 RAND USR 16516
9020 RUN or GOTO start of program
Sample program.

2X-31

Excerpted from a letter from member Harry Miller
A tip about the Larken system and disk drives

"By the way, George, I have found out what my trouble was, trying to run the three drives with my DEC power supply. The DEC disk system has 2 drives and the power supply in the case has three four-prong male connectors for supplying the disk drives. So I made up a cable and hooked up a third drive. It worked for a little while with no trouble. The next time I used the set-up it also worked OK for a few times and then none of the drives would work. I would then take the drives out of the cases and try and see what was wrong with them.

What I was doing wrong was was setting them up one at a time with a separate power supply and they would work. Well, at the time I wanted to use the computer so I would run with the one drive.

Now, this is not the same trouble as I had last year when I could not get the drives to load anything. That was another power supply problem. Then, the voltage would drift down to 2 volts and 8 volts, which was too low.

Well, to make a long story short I presently have them all going, now using two power supplies."

Comment by GFC.... The older drives (the full-height types many of us are familiar with) draw more power than modern drives. Therefore a power supply that can handle the newer drives may not have sufficient power to service the same number of older drives. If you experience drive problems check the power supply when the drives are in operation to make sure the output voltages do not drop much below the 5V and 12V required by the drives.

FOR SALE FOR SALE FOR SALE

One of our members, Richard Hurd, has switched to a PC clone, namely a Packard-Bell Pack-Mate 386 SX. Consequently Richard has some 2068 hardware add-on for sale.

-Larken RAMdisk, with 8 chips...Make an offer.

-Music Machine, a Sampler/MIDI i/f, Runs in Spectrum mode..Requires a "twister board"...Asking \$75 or b/o.

-ZAD 2068 Analog-to-Digital converter i/f with Radio Shack De Luxe Color Mouse....Asking \$30

-Two 3-inch Amdek disk drives/power supply, with 25 disks...Asking \$30

-RS232 Video Terminal "Linger Board" with many terminal emulations. Comes with IBM maxiswitch keyboard and power supply, but no monitor. Monitor capability is very versatile. Composite or TTL comes with both chips/crystals...Asking \$50

Write to Richard Hurd, 1020 S.Columbia #6, Seaside, OR 97138 USA.

May/June 1992

May 15th, 1992

Dear OOT Members,

The SINC-LINK is ready to go and I haven't a thing to say! I'm not sure that much has happened that I can report on.

Well, BYTEPOWER seem to have bitten the dust. I have had no response to the two letters that I wrote to them. I had an early letter from Kristian, a copy of which I have sent to several members whose orders had not been filled. One consolation; only one member that I know of is out of pocket. I'll see what I can do for him.

I have heard from a member who wants to sell some of his gear. Also from a non-member who has some TS stuff to sell as well. Both of them are going into MSDOS et al. A common refrain! I'll itemise them here because they both came in too late to get into the newsletter. Here they are:

James Wilson, 100 Portage Lake Drive, Akron, Ohio 44319-2304
A member.

Two TS2068 Computers	\$ 55 each
Two Silver TOS Disk Systems	\$125 each
One AMDEX III Dual 3" Drive	\$ 30
One 5.25" drive w/power supply	\$ 40
One TS2050 Modem in case, never used	\$ 35
One WICO track ball	\$ 10
One TS2040 Printer	\$ 15

Steve Wyman, 2889 St Clair Apt 3A, Toronto, Ont.
Steve is not a member. I think he received this Timex equipment from a former member. I rather think he may have never used this TS stuff, though he says that it all works.

- 2 - TS-2068 computers in good working order
- 1 - TS-2068 computer for parts.
- 2 - Jutan Intern. Ltd Tape recorders
- 1 - OS64 Cartridge for 2068 dock
- 1 - A & J MicroDrive unit plus 2 I/F boards that fit on rear of 2068. There is also a Larken cartridge which seems to indicate that there was a Larken cartridge for the A & J system. I can't be sure of that.
- 19 - Microdrive cassettes.
- 2 - 5.25" drives, full-height, cased.
One case includes a power supply
- 1 - Tasman Printer interface
- 1 - TS2040 Printer
- 1 - Green Monitor
- 1 - TS 2050 modem
- 6 - YOUR SINCLAIR magazines, about year 1986
- 12 - SINCLAIR USER Magazines, in a binder,
probably about 1986

Some good Spectrum books, such as "Supercharge your Spectrum", "Spectrum Advanced User Guide", "40 Best M/C Routines for the ZX Spectrum", and some lesser ones

An Mscript manual, and a 2068 Pro/File manual.

A mess of other papers pertaining to the TS2068.

Steve Wyman does not have an asking price for the above items. When I asked him he said "best offer". Note to US members, that this is in Canada, so there may be cross-border problems.

I have been using the BYTEPOWER Utility "COPY II" to copy several disks. It is a really neat program. It loads about 4 or 5 tracks at a time, and formats the new disk as it goes. That is, you do not need to previously format the disk. I will probably use it from now on, instead of the Larken "FORMAT.BL".

Our Larken library is up to about disk #50 now. I have not entered them into a catalogue yet. Too many other things on the go. Here are the titles:

- 41 - BobSwoger's LOGICALL Disk Management System
- 42 - A modified Tasword II by Larry Crawford
- 43 - A Sampling of Widjup utilities. Out of date, but novel.
- 44 - 24-pin graphics and screen copy package
- 45 - Speech Synthesis for the TS2068
- 46 - Language Tutor, by Joan Kealy. German and French
- 47 - Miscellaneous 2068 programs, assembled by GFC
- 48 - Selected Spectrum Board Games
- 49 - Games of Skill for the Spectrum
- 50 - Text files from SINC-LINK, in Tasword format

One of our members, John Sampson, died recently, of cancer. His widow says that he had 4 operations in 7 months, and it seemed to be just too much for his system. I'm sorry to lose him; he had phoned me just three weeks previously, and we were commenting on our respective medical systems. That seems to be a hot topic in the US these days.

Oh, I know what I should mention to you. I had an operation to remove a cataract from one of my eyes. Actually they scoured out the cataract material and installed a new plastic lens. It was done under local anaesthetic; I went in at 7am, and was home by noon. Really a remarkable operation. Sight is still a bit fuzzy; I'm told that it will clear up once they remove some stitches. Enough of that.

Our President, Rene Bruneau, has more free time nowadays. You may notice that we are getting more material about ZX81 hardware in our newsletter from Rene. I'm glad. Then the newsletter also is carrying the ZX-91 newsletter, which is very novel and interesting. Really, when I look at our newsletter, I am always impressed and a bit amazed at how we continue to put out such good stuff. Maybe more to the point, how members manage to keep giving us material to put into it. Remarkable!!

I'm going to close now while I'm ahead.

Sincerely,

George Chambers

