UPDATE!

$4.95 OCTOBER 1988

ComputerSystems

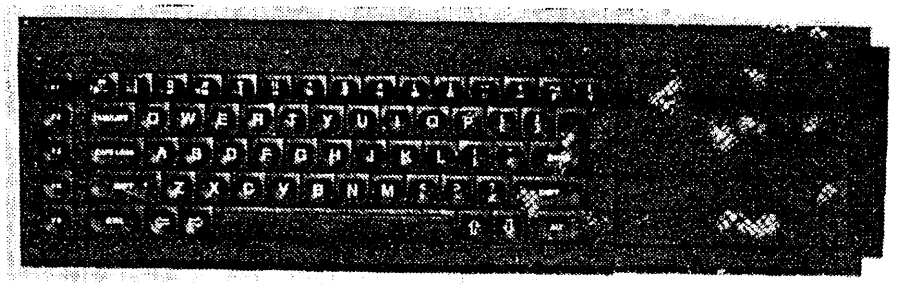
Take it to the Limit

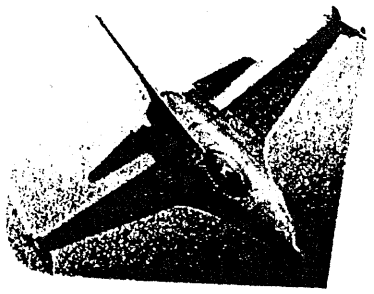
A Quarterly Magazine for the users

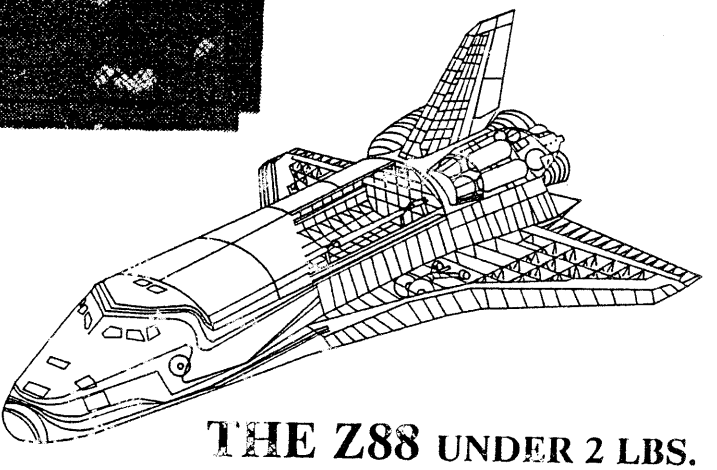
OF

SINCLAIR COMPUTERS

The Sinclair QL
-a 32 bit CPU Powerhouse!

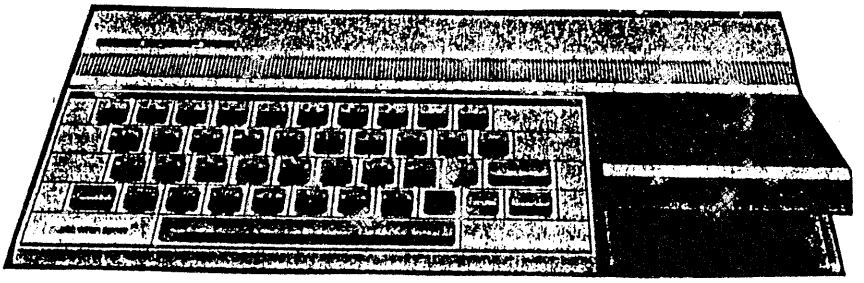




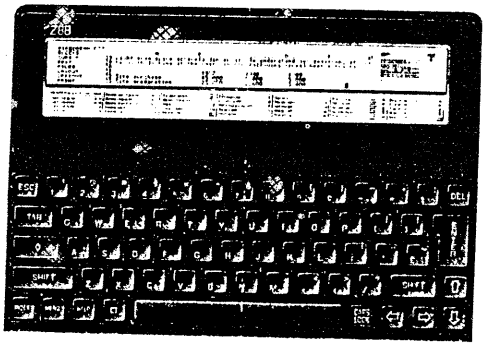


THE Z88 UNDER 2 LBS.

A Computer Without Compromise



The TS-2068 Workhorse



UPDATE Magazine
1317 Stratford Ave.
Panama City, FL 32404

Tel 904 871 3556

NEWS AND NEW PRODUCTS

A Christmas Wish for _____

[] SPECTERM-64(TS-4.1)

300/1200 Terminal Software for T/S2068
On JLO, AERCO, Larken disk or tape.
Only $25.00 + $2.50 S&H

ALSO

[] Z-SI/O Bare Boards

RS-232c Serial Interface for T/S2068
With FREE 2050 card
Only $24.50 + $2.50 S&H

OR BOTH FOR ONLY
$50.00 inc. S&H

(Outside 48 contiguous states, $6.50 S&H)

****

LARKEN Disk Hardware:

[ ] LARKEN Disk System

Includes LARKEN Floppy Disk Interface, LKDOS (v.3) Cartridge, Operating Manual and LKDOS Utilities' Disk.
Only $120.00 + $5.00 S&H

[ ] LKDOS Cartridge for JLO

Specially configured LKDOS (v.3) Cartridge that works with your JLO SAFE Disk hardware. Includes Operating Manual and LKDOS Utilities' Disk.
Only $65.00 + $3.50 S&H

[ ] LARKEN Ramdisk

No RAM chips are included. You can add up to 256K to this Nonvolatile Ramdisk. Easy to use as a floppy disk with LKDOS.
Only $60.00 + $3.50 S&H

[] Tandon TM101-4 Quad Disk Drives

These disk drives are brand new in factory sealed cartons. They are double sided, double density, 96 tpi (quad), full height drives. They are compatible with JLO, LKDOS, AERCO and QL disk systems.
$45.00 + $5.00 S&H

Buy 2, Get FREE Drive Cable!
Buy 3, Get 4th for Half Price!
Buy 4, Get One Disk Drive FREE!

[ ] Cable (2dr)..$12.50+$2.50 S&H

****

Hayes Compatible MODEMS

[ ] Avatex 2400 (3/12/2400)....$179
For CP/M, IBM and other systems.

[ ] Avatex 1200hc (300/1200).....$99
For T/S2068 (w/ Z-SI/O) and QL.

[ ] Avatex 1200<P> (300/1200).....$125
<P>ocket size for the Z88.

[ ] Avatex 1200<E> (300/1200).....$85
Great for SPECTERM-64!

Epson Compatible PRINTERS

[ ] Seikosha 1200AI (120/25)..$199

[ ] Seikosha 180AI (100/20)..$169

[ ] Star NX1000 (144/34)....$229

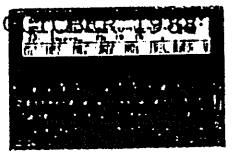
All modems add $3.50 for S&H, all printers add $10.00 for S&H (Canada double S&H on all items).

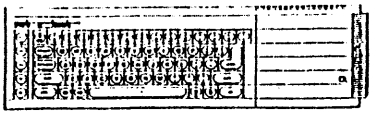
This Christmas Wish List is brought to you by:

Ed Grey Enterprises

P.O. Box #2186, Inglewood, CA 90305
Info >> 213-759-7406 << Order

Call The Grey Matter BBS & RCP/M (213-971-6260)







INDEX
UPDATE OCTOBER 1988 ISSUE

General Information

Disk Libraries	4
Flea Market	70
Invitation to Writers	48
Issue Disks	6, 72
MDV Dubbing	6,72
North American User Group	8
SINCLAIR Z88	12
Supply Support	*15, 70-73*
Writer's Format	43

CLUBS

Boston Computer Society	27
CCATS Portland, OR	25
Milwaukee, WI TSUG	33
SINCUS, New York	8
Toronto, Canada TSUG	33
Vancouver, Canada TSUG	17

ADVERTISING RATE SCHEDULE--- page 77

ZX APPEAL	17
SINCUS NEWS	8
PLOTTER	25
BoSTUG	27
SINC-LINK	33
QZX	40, 17

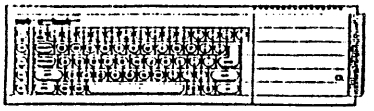
UPDATE SOFTWARE LIBRARY------page 74

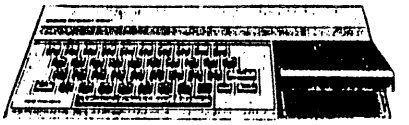
TS-2068 Articles and Subjects

Align Decimals	11
Color Screens Copy	9
Disk Multi Mgr (Oliger)	21
Extra Memory Recap	49
Extra Memory, Charles Bothner	50
Graphics LPRINT	25
Hex Printer Dump	25
INTeger Function	11
Interface for Color Copy	9
LPRINT CHR$	25
MAX DOS 3 Stacked DOS's	2
MSCRIPT for LKDOS	34
ON ERR Tutorial	14
OPEN#, CLOSE#, Streams	36
Okimate 20 Printer	9
PRINT USING	11
Round Numbers UP	11
TOS DISK with OS-64	13
Telecomputing	41
Test Printer	25

QL Articles and Subjects

ARCHIVE Tutorial	56
Address File Program	44
Arithmetic for Kids	64
Beginner Course	30
CABLE Column	56
CABLE ISSUE DISK/MDV CART	69, 72
COPY MDV Cartridge	23, 31
HELPER, Super Basic Tool Kit	63
Initializing SPEED SCREEN	23
Initializing TK2	23
Microdrive Formatting	23, 30
PRINT Primer	32
QL Tips	23
QUILL, Configure printer	31
QUILL, File Saver	29
QUILL, LLISTing with	31
RAM EXTENSION	19
REFIELD, a Major Software	60
SCAN, an ARCHIVE Software	65
Sound Experimentor	18
TAX Software to beat Deadline	15
TELECOMPUTING	17, 41
TELECOMPUTING with PACKET	42
TOOL KIT 2	28
TRUMP Card	20, 72





HOW GOES IT, UPDATE??

UPDATE Magazine launches into the next four years of publishing with near 100% renewals of subscriptions, and a flood of new subscribers. Most gratifying are the many favorable comments and good wishes received with the subscription renewals. Actually, subscribers more than trippled since the last issue. It is difficult to take score of QL users and TS-2068 users. Around 2/3 of the subscribers are QL owners, but half of them also own a TS-2068.

MAX DOS Versas the IBM MS DOS Versas QL DOS

In past writings I've kinda given the IBM PC a bit of criticisms, mainly to re-assure the TS-2068 and QL users that they really aren't missing much by not having a IBM Compatable machine. AND, we need to feel good about what we are doing! A letter came in from a subscriber who was one of the original designers of the IBM operating system. He ask for anonymity, so I will quote only half of his letter:

"Dear Bill, Enclosed please find my 1989 renewal. Unfortunately I do not run any of the disk systems that you support, however, the organization and the quality of the articles in Update are well worth the price. I have invested so much of myself into SPDOS, using many of the system components, that I cannot face even evaluating another system. I am, however, beginning to worry about equipment failures. If you hear of any MILLENNIA K disk systems available I would appreciate a call. I could also use a disk drive case with power supply for two 5.25" drives.

I was pleased to note your reaction to MS DOS. In 1965 we were similarily concerned about OS/360 (similiar philosophy) then being developed for the main frame family. Two others and I built the prototype virtual machine operating system at IBM Cambridge Scientific Center.

INTEL has built a (8086) Virtual Machine Facility into their 80386 chip (Intel 231252-001) which now seems to be supported by VM/386. Perhaps when this is expanded to use paging and allow VM/386 to run under VM/386 I may desert the 2068 for purely personal reasons. (Signature)"

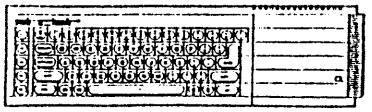
To the writer, and others interested: Thanks for your comments. The CUYAHOGA VALLEY SOFTWARE WORKS, 615 School Ave, Cuyahoga Falls, OH 44221, is a group of users and programmers using SPDOS and the Millennia K Disk system. The address given is that of Thomas Simon, one of the leaders of the group. This group has developed SPDOS on disk to load into and operate with the TS-2068 Oliger Disk System. SPDOS loads to RAM and is co-resident with the Oliger DOS, providing two disk formats and a "combined" DOS Tool Kit language structure, much more powerful than either DOS.

The nice part for Oliger Disk owners is that no additional hardware is needed. All they need to do is order the SPDOS Disk ($29.95 from address given).

The nice part for the Millennia K SPDOS users is that the Oliger disk system provides a new "hardware home" for SPDOS, where all of their existing software disks can be loaded and used in an even more capable environment than that of Millennia K.

So Friend, dont give up the ship, just add the Oliger Disk, which is less expensive than an IBM controller card, and you will continue to be comfortable using SPDOS. Also I appreciate your nice comments and your agreement with the position I take on MS-DOS. There are not too many who are willing to "buck the tide" and tell it like it is, in the face of the "MS DOS Compatability fraud" that the industry has perpetrated upon the public. People actually develop an inferiority complex over not having a "IBM Compatable" computer! By the way— why dont you do some articles for UPDATE? We could benefit form your expertise. The next subject broadens upon this discussion.

MAX DOS: (Reference article in UPDATE, July 88 issue for background). I have continued to use the Oliger DOS and





Larken DOS together, and have experimented with stacking SPDOS "on top of both", THREE Disk Operating systems together. Actually the Larken DOS and SPDOS have many of the same functions. So there is not a great difference in the capabilities of "Stacked Oliger/Larken" and "Stacked Oliger/SPDOS", so we could dub the two "MAX DOS L" and "MAX DOS S"

So, I will continue to discuss MAX DOS as the combo of "Oliger DOS Hardware interface", and the "LARKEN LKDOS Dock Cartridge". I have been operating with this combination of hardware for the past several months, hardly ever bothering to remove the cartridge or the Oliger B board. A couple of months ago I received a "loaner B board" from John Oliger. This one has a "EPROM OFF" switch, which my earlier board did not have. The significance of this added switch in the MAX DOS environment is that the LARKEN NMI functions will work when the OLIGER EPROM is switched OFF. BUT, to use the OLIGER NMI functions, you must remove the LKDOS Cartridge. Now we need to convince Larry Kenny to give us a little switch modification to his LKDOS cart to do the same, in order to allow the OLIGER NMI functions to work as well.

Otherwise, MAX DOS is indeed a very powerful DOS, despite the minor NMI clash . Now I dont want this to sound like criticism, so will lead in with with compliments. The LKDOS EXTRA BASIC functions are SUPER! AND, so are the Oliger Extra Basic functions. When operating in the MAX DOS environment, one can use ALL of BOTH system's extra basic tool kit functions. The nice part is that the LKDOS functions are speeded up by the Oliger system's faster disk operation. This extra speed is most noticable when popping in screens from disk, MERGING in program segments, bringing in data files from disk and saving data to disk.

The OPTIMUM disk drive situation is TWO drives. One drive is used for DISCRETE Oliger DOS use, and the other is for DISCRETE LKDOS use. Each are "assigned their drive number" by either programming in software or by direct commands. EXAMPLE: LET/d=0 would assign drive zero as an OLIGER DISK. AND- PRINT #4:GO TO 1 - would assign disk 1 as a LKDOS disk. From then on, the system "remembers", and a LKDOS disk command automatically accesses DRIVE 1, or a Oliger DISK command will access disk zero.

But that sounds like we have two separate disk systems, which we have-. We can load and operate disks that have been produced by anyone's Larken or Oliger system. And, we can Format and prepare disks in either Larken or Oliger disk formats. BUT, we also have a COMBINED SYSTEM. Programs may contain any combination of DOS commands. Boot a screen in from the Oliger disk, or save a file to the Larken disk. Programs that contain BOTH DOS DISK commands may be SAVED to either disk. AND, the operator does not even need to think about which disk will be operating. A Disk Command in the software will automatically use the proper disk, as designated by the previous LET/d= and the PRINT #4:GO TO.

MAX DOS opens up tremendous power for programmers to use, employing all of the command functions of two disk operating systems in one or more programs to command two disk drives. The software can be controlling up to nearly 2 mega bytes of disk capacity (2 80 track drives). Good planning can result in "programs in one disk and Data in the other disk" and keeping the program disk write protected. SO- whether MAX DOS is Larken DOS, or SPDOS- combined with the Oliger system, it is one tremendous Disk Operating System. All we need is for more TS-2068 users to discover this.

THE QL DOS

The SINCLAIR QL DOS is excellent. Nobody needs to feel inferior about using the QL. I look at it this way. How many IBM's have you seen that have a pair of additional fast tape drives? It is made even more powerful by the BIG Tool kit functions available in ROM cartridges, TRUMP CARD, and SANDY BOARD. Disk Drive works with the QL with nothing more than a Controller card, same as any other system, but it is less expensive than any other. Memory expansion is also less expensive. Do we need IBM Compatability?

TS-2068 and SINCLAIR QL

ISSUE DISKS and MICO DRIVE DUBBING

The Issue Disk program for the TS-2068 will be carried on for support of the QL users, but with a small difference. TS-2068 programs and utilities given in the UPDATE pages are actual listings given for those who want to try keying in the listings. Some are quite long and keying in may not be very practical. Yet they are there to try. QL utilities will be listed for key-in also. But the software programs for the QL are generally just too long for practical key in projects and the listings would take up too much page space. These long QL programs will be written up as "Mini Manuals" in UPDATE. Although, this issue has one complete (but short) QL software "Mail Merge" that will be listed as a key in project. So, whether the QL programs are listed or not will be determined by their program length.

The ISSUE DISK for the TS-2068 programs will include all of the software and utilities of a given issue. The QL Issue Disk will be variable, and will include at least one major software and the utilities given in the pages of the current issue. Sometimes there will be two issue disks for the QL users in one magazine issue. It just depends upon the program lenght and applications of the software.

PRICING: All Issue Disks for both computers will be priced at $20.00. This is an increase of $4.00 over past TS-2068 disk price because UPDATE was not breaking even at $16.00 each. UPDATE supplies the disks, does the recordings, pays the mailing expenses, and shares half of the proceeds of issue disk sales with the authors of the programs. It really is a non-profit situation for the magazine and the authors don't realize more than a token payment for their support of us all.

MICRO DRIVE ISSUE Recordings: Due to the higher price of the cartridges and their scarcity, UPDATE will provide Issue Dubbing service. Those who want the programs and utilities may send in a FORMATTED MDV cartridge and it will be returned with the Issue programs recorded. "Formatted" is emphasised because if the user formats the cartridge first, he is assured that the cart will take the programs and there will be no mail delays. The price for MDV dubbing will be the same, $20.00 per named Issue Disk.

UNCONDITIONAL GUARANTEE: Subscribers may be sure that these programs on disk or mdv are worth the price or more. If not not completely satisfied, the user may keep the disk or cartridge and just ask for full refund. Sometimes a software program may require more documentation than UPDATE can accommodate in the issue pages. PAPER documentation and the mailing of paper is expensive, and the author may need to charge extra for an extensive manual. These instances will be announced. Documentation within a disk or micro drive is a much better method, where text can be illustrated with screen examples, and the text sent to a printer.

COPYRIGHTS

Unless specifically stated, all programs presented in UPDATE magazine are the property of the individual authors and are protected by copyright. The authors give their limited permission for publication in UPDATE Magazine and for the distribution of recorded copies to the purchasers of the Issue Disk series. No other sale or transfer of these programs are authorized by anyone or any other publication. Some Programs and utilities may be donated to public domain by specific statements made by the author within the text of the programs presented in the magazine. In the absence of such statements, the programs are priviledged.

MICRO DRIVE V/S DISK: Both mediums are excellent. A huge software may be recorded on disk, but to use micro drive, such a program would need to be broken down into several individual segments, as is the four PSION sofwares. By the same token, an un-expanded QL can run large programs that are broken down in "boot in segments". Most large programs presented in UPDATE will have two versions, one for disk and another for micro drive. All programs presented will work on standard QL computers, but some may have annexes to expand into extra memory.

QL SUPPORT IN THIS ISSUE

Remember that UPDATE questionare? Well, a tabulation reveals much about the QL user's equipment and desires. About 60% of present user have un-adorned QL computers- no memory expansion, and micro drive only as a storage device. The other 40% have disk and extra memory. Of the group, about 70% say that they need beginner level programming information, tips, and utilities to use. The 30% minority ask for more sophisticated information about disk drive, trump card tool kit programming, more advanced software, conversion of IBM MS DOS software, and other hi brow stuff. UPDATE will try to accommodate all. This issue begins a feature article titled, "THE QL BEGINNER COURSE". This series will go on "forever". I guarantee that it will because I'm the author. If you'll follow this series you will learn how to program the QL, and how to get the most out of the four PSION Softwares. You'll learn it as I learn it, because I am a novice on the QL. But better stay up with each issue or you'll get behind! I'm a mean and lean TS-2068 programmer and learning the QL will come naturally. So, start with me in this issue and progress with me.

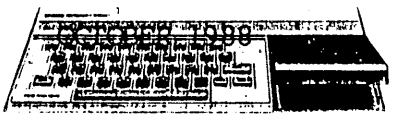
MEET some QL Programmers

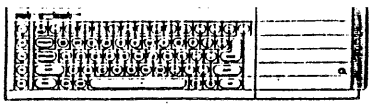
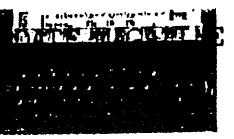
Bill Cable, Wood and Wind Computing, RR-2 Box 92, Cornish, NH 03745 (tel 603 675 2218). I lead right off with a plug! This is because I want you to know Bill. I want you to know all of our supporters, dealers, and publishers. Bill Cable is one of them extraordinary programmers who specializes. Bill is a QL dealer who developed a customer base of businesses, selling QL computers to medium and small business people, and then supporting them with custom software. Needless to say, his feet were cut from beneath when Sinclair stopped production and support of the QL. I am sure that all QL users will encourage Bill Cable to turn his programming efforts to support the users. This issue brings in one of Bill's outstanding softwares, titled "REFIELD". This software is designed to augument the PSION ARCHIVE to make it infinitely more flexible and powerful. REFIELD is presented in versions for both Floppy disk and for Micro Drive, and will occupy one of the two October Issue Disks. REFIELD will also be offered for Micro Drive Dubbing. See the last page of this issue for details.

NEXT, another well known author and programmer has invaded the QL Group. Bob Hartung has been a regular writer and programmer for UPDATE since it's first issue, but doing TS-2068 articles. Bob continues to write and program for the TS-2068 while producing for the QL as well. Thats kinda like alternating between both seats of a bicycle built for two while its rolling down hill. Now get this! Bob's "QL ADDRESS FILE" software, given in this issue, is designed so that it works in three different computers, the QL, the TS-2068, and with Beta Basic in the Spectrum. QL Address File is designed as a simple program that works independently, and is an excellent one to tackle as a key-in project. Whatever, this one will also be offered on disk and for micro drive dubbing. See the last page for ordering information.

BOB DIGGS KV3Q, a HAM Radio Operator, has been tackling the QL and its problems associated with Telecomputing, only on a little different tack, with PACKET. This gave me a chance to open up the subject, "Telecomputing". Hope others join in to cover Telecomputing in detail.

Finally, several re-prints of QL articles from the various Sinclair User Clubs are given in this issue. One of the very best paths to learning computers is to join a User Club, and participate. This way, one meets folks who have like interests in computing. Friendships are formed and telephone numbers exchanged so that help is but a few feet away at your telephone. Some travel over a hundred miles one way to attend User Group Club meetings and benefit greatly from the association. The idea is to form a "special interest group" of QL users within the special interest Sinclair Club. Then you get together and exchange information and tips, swap public domain software, and copy programs from the club library. This issue brings information about several live wire clubs, and samples some of their newletter articles. If there is not a club listed near to your home, call me and I'll put you in touch with one.



SINCUS PROGRAMS

Time Designs did the Timex Sinclair users an excellent service by producing an up to date listing of all Sinclair Timex Clubs, listing all Club publications, and rating them on a scale of from 1 downward. SINCUS NEWS, 1229 Rhodes Road, Johnson City, NY 13790 copped FIRST PLACE. Subscription is $8.00 per year (underpriced). SINCUS provides more service to both members and non members than just a good newsletter. See clips below.

SINCUS OFFERINGS:

RENEW or Subscribe to SINCUS NEWS for $8 per year. Send in a Mailing label of either UPDATE, Time Designs, or Syncware news and receive and extra issue of Sincus News FREE, added to your subscription. (Offer ends 12-31-88)

SINCUS TAPE/DISK LIBRARY: Write for details (An OUTSTANDING BARGAIN- UPDATE RECOMMENDS! ed)

SINCUS "Computus Interruptus"- a 63 page programming manual by Wes Brzozowski. (UPDATE rates this as EXCELLENT) Comes in two forms- a paper manual for $10 member or $12 non member of SINCUS. or, a disk in 40 or 80 track Oliger or Larken Format- $5 member, or $6 non member of Sincus.

SINCUS CLONE:- TS-2068 tape copier- $6 for tape and documentation.

The SEP/OCT issue of SINCUS NEWS is in and contains some very interesting stuffin. The Editor gave UPDATE MAGAZINE some very appreciated praise, for which I thank him for profusely. BUT- he didn't get everything right! He said that UPDATE is "virtually a one man show". Actually, about 75% of the red meat pages in UPDATE is contributed by GUEST WRITERS. (Count um!) The Editor does a few new utilities and rambles on in editorials to give the UPDATE its personality.

Without the generous participation of expert writers, UPDATE would be just a little nothing newsletter. The editor realizes that some folks just don't like OPINIONS expressed in editorials, so pages are piled on to make up for editorial verbosity. So, Paul, Thanks, but stand corrected. UPDATE is a "participating" magazine, and owes its reputation to the writers of fine articles, programs, and utilities.

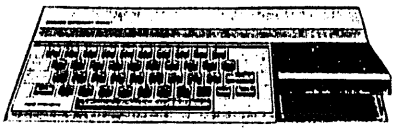
The North American Sinclair Society

Now to slip out of the third person: Paul Hill, editor of Sincus News, has some darn good ideas about consolodation of publishing efforts into ONE MONTHLY Magazine which would encompass all existing publications and all Club Newsletters. It's a good idea, Paul. What is lacking is "Management Control", so that such a publication would not become a "Sinclair publishing monopoly", setting advertising rates up to unfair prices, and reducing page (red meat) content down to minimal in order to make higher profits. But still, it is a good idea to pursue. I would invision a Sinclair Timex publication that would be managed and controlled "by the user group".

WHAT USER GROUP? There is no organized user group. ENTER "The North American Sinclair Society", or some such named organization. The Florida Groups coined "SNUG" which everyone has trouble breaking down into prose. But WHATEVER, we need a PARENT ORGANIZATION of Sinclair Timex user groups of North America. ONLY When such a parent group is formed can there be concerted effort to do things for the good of all. This, I believe is where all of the present User Group officials should be working and coordinating. A North American Group wont be formed without such coordination amongst the existing strong clubs.

I cannot form a North American Group. Neither can Tim Woods, or the other publishers. We do not represent anyone. But, we stand by to lend our support. You present officials of Clubs DO represent people. Why dont you establish direct coordination between Clubs and form the North American Group? Vote in a management board of officers who will operate as a Senate to form up the charter and by-laws. Use the existing publications as tools for publicity membership drive, and COMMIT your own club members en-mass to become members of the North American Group.

THEN, when the North American Group is fully formed and has establised a sense of directions, it will be time to explore the idea of a Single Publication, of ample size, to become the voice of Sinclair Timex Computing in North America. Membership in the Parent Group should carry with it a subscription to the Group publication. UPDATE Magazine will cooperate if the interests of the subscribers are guaranteed to be fulfilled. So there you are. A gauntlet has been thrown to the Officials of our current Sinclair Timex Clubs. Who will accept the challange and assume leadership for coordination? Its your move now.



EARL V DUNNINGTON
4356 KING THEODORE DR.
BOYNTON BCH., FL 33436
PHONE 407732-6219

A SCREEN COPY IN FULL COLOR AND MORE!!

A REVIEW BY EARL DUNNINGTON

John McMichael has done it again! That is improved the capability of our 2068's by developing software that will make a full color or black and white screen COPY, when used with his interface, the OKIMATE 20 color printer, and the Okimate Commadore Plug and Print module.

He has also developed two other programs for this hardware. A Patch for Jack Dohnay's MSCRIPT revisions V5/V5.2 and a machine code OKI-DRIVER program that supports the LLIST and LPRINT commands. Each program is on a separate tape and were $ 8.95 each.

The Screen Copy program requires that the screen be saved on tape with the SAVE "name" SCREEN$ command or the equivalent SAVE "name" CODE 16384,6912 command. The options allow you to specify color or black and white, small or large size, the horizontal position on the paper, and the number of copies to be made. After the specified number of copies has been made, you can again enter your choice of options and either use the same screen or load a new one. The small size produces a screen copy 1.55" by 1.25". The large size is 3.15" by 2.55". Approximately 30 copies can be made with each Color Ribbon ($ 5.65) using the large size.

The black and white screen copy using the black ribbon is much darker than that produced by the 2040 printer. However, there is no shading for the different colors like that produced by Zebra's Z-PRINT 80 or like a color screen viewed on a green screen monitor. Due to this the color copy is a vast improvement, even though a large block of paper color only characters reproduces somewhat unevenly, the colors being slightly darker along the horizontal juncture of each character line with another. The individual pixels are not apparent to the naked eye. For testing, I used the third color screen from Knighted Computers "NIGHT GUNNER" (c) Digital Integration. It has the following colors: white, black, green, yellow, blue, cyan, and red. I consider the rendition of the ink colors, excellent. The instructions are also excellent.

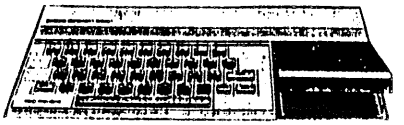
As I do not have Jack Dohany's revised versions of MSCRIPT, I can not comment on the Patch program, other than the instructions seem very complete.

The OKI-DRIVER program is strictly machine code. It loads in starting at address 64874 and is 494 bytes in length. The program is compatable with the Larken DOS V3 and the Larken 256K RAM DISK. It is also compatable with the Zebra FDD 3000 with TOS. As written the code is not relocatable and therefor is not compatable with the Zebra OS-64 cartridge or any other program using two display files. In addition to supporting LLIST and LPRINT there are options for sending printer control codes with support for LPRINT only, switch the character transposition off, and switch backand forth from the big printer to the 2040. If the 2040 is also connected you can always use COPY. The options and switches are made using RAND USR commands. There are eight 2068 character codes that will print as Commadore characters:

92 Always
95 Always
96 Always
123 Always
124 In printer control code mode
125 Always
126 In printer control code mode
127 Always

The instructions for using the program are excellent and quite complete. The utility of the program would be greatly enhanced by fronting the code with a BASIC loader and code mover program to adjust the internal jumps and the calls to any location desired.

The McMichael interface emulates a Commadore serial port. It was originally developed for use with the 2068 and the Commadore four color pen 1520 plotter. The interface can be purchased as bare board ($14), a kit with all parts ($20), or assembled ($30). I ordered the kit. As I do not have the Oliger expansion board, I also



ordered the edge connector ($4.95). The documentation is excellent and I had no difficulty in assembly. Do not worry if you think you may have a solder bridge between pins 8 and 9 of U3. They are supposed to be connected.

The OKIMATE 20 ($136) is an 80 column "State of the Art", square 24 pin, NLQ, dot matrix, thermal color printer. It uses color or black thermal ribbon cassettes. Thermal paper without a ribbon cassette can also be used (4-80 foot rolls $16). The footprint is about 13" x 7" and is about 2" high. A tractor is built in for paper with sprocket holes. Single sheets can also be used. A serrated plastic edge is built in for tearing roll thermal paper. The printhead is removable for cleaning or replacement ($21.95) by the owner. The overall quality of construction appears to be excellent and it is very quiet when printing. There are printer control codes for draft quality, character size (Pica, Elite, Fine, double width), Italics, Super and Subscripts, underlining, reverse printing, form feed, carriage return, line spacing, starting position, skip over perforations, and graphic modes.

The printer documentation, registration form, a color ribbon cassette, a black ribbon cassette, computer paper, a order form for supplies, and the plug in printer module are all contained in the Okimate Commadore Plug and PrinT package ($79.95). Dip switches in the module allow you to use different international character sets.

I consider this, for the price (total about $270) one of the best full size printer systems available for the 2068. It could be improved by a color screen program that would produce a printout twice the present large size. The prices quoted are approximate.

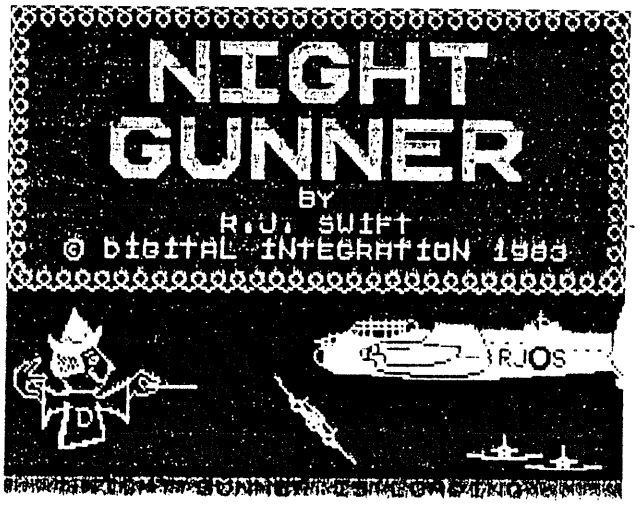
The OKIMATE 20 and the Commadore Plug and Print module can be purchased at TOYS R US, Service Merchandise, and Sears. Some mail order computer supply houses also carry them. Check the ads in Commadore magazines, Compute, and the Computer Shopper.

Send a legal size SASE to:

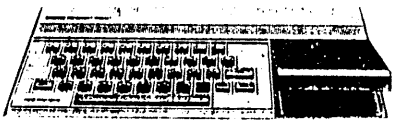
John McMichael, 1710 Palmer Dr., Laramie WY 82070

for an order form and price list. If you enclose one dollar he will also send a sample color screen copy.





Editor's note: Two screen copies were supplied, one a B&W and another in full color. These are reproduced below if full size. Unfortunate a B&W copy cannot do justice to the color screen so I will describe it. The large print "NIGHT GUNNER" is in Cyanide on a bright red background. The smaller print below is white, as is the fancy border. The two bombers and ships below are in bright yellow on a blue background. The knight with pointed rapier is sketched in white. "Night gunner is loading" is black print with green background. The colors and contrast are vivid, about what you would see on screen, but with with no gaps between drawn pixel lines. Focus is sharper than a screen image. "NIGHT GUNNER" stands out on its background in a way that looks three dimensional. This is the same size screen copy as we are used to with other softwares. I would suspect that if the printer could be put into an expanded mode that the print out would cover a larger area, as does other screen dumps. Only one gripe and thats against Digital Integration; "What respectable Air Force paints their aircraft YELLOW?"



THE ROUND UP

AN UPDATE PROGRAMMING TUTORIAL

This is a program designed for UPDATE to demonstrate several abilities of dealing with numbers. FIRST, the "INT" function "ROUNDS DOWN" figures. All businesses do the opposite, ROUNDING UP to the nearest cent. This can make a busy department store quite a few dollars during a day's operation.

Next, PRINT USING is a ROM routine used with some computers to "Format a Column of Figures". Mr. Sinclair left PRINT USING out of his Sinclair Basic ROM. But anything that a ROM routine can do can also be done by programming in RAM. Here we use a little routine that simulates a PRINT USING function to print the dollar sign, then print a column of figures with decimal points aligned. Also, Zeroes are added when necessary.

The use of INT (INTEGER) in math calculations to assign the resultant to a variable causes errors. The error is only in the assignment to the variable. The "Floating Point Calculation" has the right value "in there somewhere", but a small fraction is lost when INT is used in a calculation. EXAMPLE: try this group of direct commands: <LET a= 1.0954><LET b=INT a> <LET c=INT (100000*(a-b)) > <PRINT c>. Floating point calc drops a tiny fraction that can be troublesome. LINE number 9710 uses a little "floating point k factor" of .0001 and then adds .005 for "rounding up" to the nearest cent.

Line 9712 adds a whole number if a round-up changes 99 cents to $1.00. Then the integer number and the decimal fraction are assigned to a$ and b$ for printing. This is because the TS-2068 prints decimal numbers as "0.4", "0.25", which is not convenient to print directly. The "STR$" function transfers a numeric value into a literal value to place in a string. Try this EXAMPLE: <LET a=12> <LET a$=STR$ a><PRINT a$>. Amazing, eh? Now lets go the opposite way: <LET a$="386"><LET a=VAL a$><PRINT a>. Stupendous!! Mess around this way with STR$ and VAL, and you'll get good at using numbers in both simple variables and in string variables.

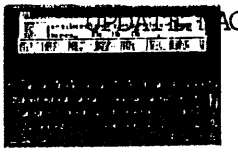
Next, line 9718 does a bit of "String Slicing". Some of the odd presentations of fractions may result in B$ being ".020". To just present "cents", we chop off the last "0". Lets try that separately now: <LET b$=".020><PRINT b$(TO 3) >. Or: <LET b$=b$(TO 3) ><PRINT b$>. Try several other examples like: <LET b$="I LOVE THIS"><PRINT b$><LET b$=b$(TO 6) ><PRINT B$>.

Next, line 9720 "sums" the figures, but only IF x=1. In the beginning we LET x=0. So, each input figure is summed as they are typed in. When we want to present the sum, we type "s" instead of a number. Back at line 9700, the input of "s" instead of a number gives us an error (unless variable s has been assigned a number). The error trips the first ON ERR GO TO 9800. Then at line 9800, the ERR is RESET, a line is drawn, and a dollar mark is printed. Then the GO TO 9710 processes the SUM to be printed.

Finally, line 9770 prints the figures, with all decimals aligned at your TAB setting. A$ carries the WHOLE dollar value and B$ carries the CENT value. Then line 9780 stops if the SUM has printed (x=1), or line 9790 returns to pick up the next input number. ONE MORE line needs to be discussed. LINE 9700 has a second <ON ERR GO TO 9706>, which is the next line in the program. "STOP" creates the error so that the ON ERR programming is cancelled (no open ON ERR to mess you up later).

Now thats a lot of programming functions in a few short BASIC lines, and likely less memory is used than if Sinclair had put the routines in ROM and given us a PRINT USING syntax to use. And, because it is in BASIC programming in RAM, we can alter it as needed to tailor the routine to our needs. There are many ways you can modify this utility to use as a SUB-ROUTINE in your programs. Change PRINT to LPRINT in line 9790 to put it on paper. But if you just want a "Column Adder", use it the way it is.

```
 9680>REM [ THE ROUND-UP          ]
          [ Summing of Columns    ]
          [ PSEUDO PRINT USING    ]
 9690 CLS : LET x=0: LET t=0: LET
  tb=25:
 REM [ variables are "t=Total.
       tb=tab.  x=switch that
       turns on by ERROR for
       totalling. ]
 9691
```



```
9692 REM [ Lines through 9700
           may be eliminated.
           Then the input to
           the routine would
           be numbers assigned
           to varable a.
9694 REM [ The Routine end is
           line 9780. Change
           line 9780 to RETURN
           for Sub-routine use.
9697 REM
9700 ON ERR GO TO 9800: INPUT "K
ey in Number- or ""s"" for SUM
 ";a: ON ERR GO TO 9706: STOP
9706 ON ERR RESET : IF t=0 THEN
PRINT TAB tb-5;"$";
9710 LET b=INT a: LET c=INT (100
*(a-b+.0001+.005)): LET d=c/100
9712 IF c>99 THEN LET b=b+1: LET
 c=0: LET b$=".00"
9714 IF c>=0 AND c<10 THEN LET b
$=".0"+STR$ c
9716 IF c>9 THEN LET b$="."+STR$
 c+"0"
9718 LET a$=STR$ b: IF LEN b$>3
THEN LET b$=b$( TO 3)
9720 IF x=0 THEN LET t=t+b+(c/10
0)
9770 PRINT TAB tb-LEN a$;a$;b$
9780 IF x=1 THEN LET x=0: STOP
9790 GO TO 9700
9800 ON ERR RESET : PRINT TAB tb
-4;"        "'TAB tb-6;"$";: LET
x=1: LET a=t: GO TO 9710
```

This is really a CLUB Issue of UPDATE

If the reader has concluded that UPDATE is plaggerizing and stealing articles for the various Club newsletters, then the objective is missed completely. The reason for the selected re-prints is to give samples of the excellent material that is produced in these excellent club publications. So, you get two benefits from these club articles. First, you get the good articles to digest. Second, you get a chance to savor the personality of the clubs and the quality to expect in club publications. Then, of course, you get the address to use to JOIN the club or Subscribe to their newsletter. Actually UPDATE has a bank of excellent NEW articles about the TS-2068 and the QL, enough to fill at least the next five issues. The Sinclair user groups need an influx of new members to spark new life and direction. Especially, the clubs need QL and TS-2068 members to join and form a special interest group within the existing Sinclair special interest Club. This is UPDATE"S way of helping to round up new members of the Sinclair User Clubs. Unfortunately, space does not permit the introduction of all clubs in one issue, but we'll get rountuit in future issues.

*** THE SINCLAIR Z88 "lap top" COMPUTER ***

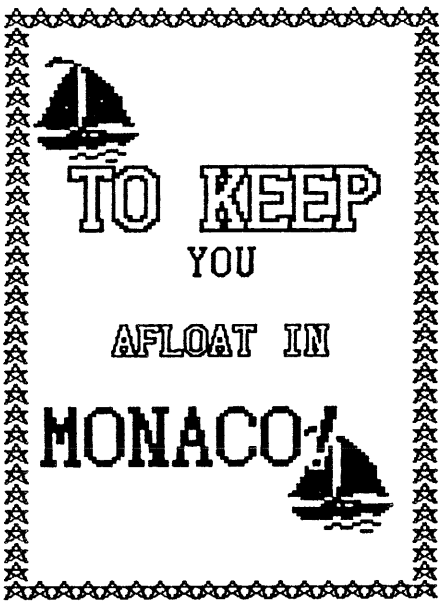
So far as I can tell, there are only praise and enthusiasm coming from the owners of the Sinclair Z88. Frankly, I dont like the term "lap top", as it seems to have a connotation of "tinker toy" which demeans the extraordinary capabilities of the Z88. Lap tops come in many varities of both size and price. Last week I looked at a Mitchubitchi lap top which comes in at a measly $4999.99 unit price. Then there are earlier lap tops which didn't sell like hot cakes and are now priced below $500.00. (Read "not designed to user needs").

We have several subscribers to UPDATE who already have the Z88, and I suspect that there will be many more on December 26. Why not, the Z88 will fit into a Christmas stocking! Mike Fink, a subscriber and a Z88 Dealer, has just about finished a new "Z88 Instructional Manual" which will become a best seller to the Z88 group. Now Mike has built up a real head of steam (enthusiasm) for this little and powerful computer. I'll Quote some of his sentences: "You will never need to load any software, even though you can if you wish to". "All of the programs that you'll ever need for normal business or home is already in the computer and available by the touch of a few keys". "It is not possible to design or build a better word processor than is built into the Z88. "It will print to any printer. Just pull the cable out of your pocket and plug into the nearest printer". "The spread sheet capabilities are fantastic and easy to use." The memory add ons can reach 3 mega bytes, plus any number of insertable memory cards". "It has a built in EPROM Programmer which operates automatically". "It holds data in memory forever, whether the computer is on or off". "It operates on inexpensive pen light batteries which can be changed without the computer memory losing any data". "The Z88 is truly the most versatile combination of computer design, function, and portability, in the World at this time".

As I said, Mike Fink is an enthusiastic Z88 Dealer, so we cannot exactly call him un-biased. UPDATE welcomes and will publish a detailed review of the Sinclair Z88 FROM AN OWNER ONLY. We need to know from the horse's mouth more about this new Sinclair product, whether it is worth its price, and an evaluation of its operation. Only an owner who has operated the Z88 for several months can provide such a comprehensive review. How about you Z88 owners telling us about this new computer?

(Mike Fink, 355 W. 39th St., NY, NY 10018-1401)

LETTERS: Just had to show you this one (reduced 50%).

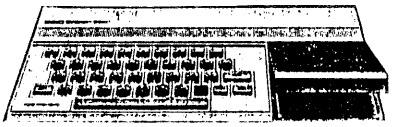


Jon A. Gardner
2937 N. 59th St.
Milwaukee, Wisconsin
53210

Dear Bill,
I enjoy reading UPDATE and have found the info very helpful. I have also purchase some very good software that was advertised in the magazine.

Jon





USING THE OS-64 CARTRIDGE WITH TOS

At no time should the TOS copyright notice remain displayed, otherwise you are not properly booted up.

BOOTING UP

1. With the computer and drives off, insert OS-64 cartridge in the 2068 cartridge port.
2. Turn on computer. The screen will clear and then display the OS-64 copyright notices.
3. Power up the drives and then insert the TOS DEMO disk in DRIVE A. The drive light will flicker as the disk is read. The OS-64 copyright notice will remain displayed.
4. When the light stops flickering, type in the CAT* command and press ENTER. You are now booted up in TOS with a 64 column screen. The DICE game must be revised for a 64 column screen.

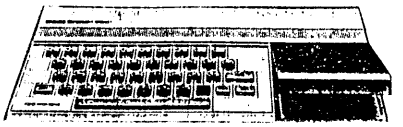
REBOOTING

1. With a TOS disk only in drive A, while pressing and holding the red reset button on the drives, pressand release the grey reset button on the interface.
2. Release the red button after the screen clears and the OS-64 copyright appears on the screen.
3. The drive light will flicker as the disk is read.
4. When the drive light stops flickering, type in CAT* command and press ENTER.

TO TURN OFF THE SYSTEM

1. Remove disks from both drives.
2. Turn off the disk drive power
3. Then and only then turn off the computer, otherwise you may damage either the disks or the drives.

EARL V DUNNINGTON
4356 KING THEODORE DR.
BOYNTON BCH., FL 33436
PHONE ~~305~~ 407-732-6219



<ON ERR> Programming

A person whom I'd never heard of sent me a cassette, saying that it contained an important program that "hangs up every time it is loaded". I loaded the program and sure enough it did hang up. A touch of the NMI button on the Oliger DOS board gave me "O.K" at the screen bottom. Then LIST put the listing on screen where I discovered <ON ERR CONTINUE> in the first program line. I EDITed the line and removed the ON ERR, then found a subscript error in a following program line. The moral of this story is that if you use <ON ERR CONTINUE> you'd better have every conceivable error eliminated. I returned the cassette with the ON ERR removed and the program re-saved, with expanations, never to hear from my pen pal again- no thanks, no nuttings. AND, I paid the postage. The morale of that story is, if you ask someone to do something for you and he does it, you should thank him.

The TS-2068 "Remembers ON ERR" and the line number in an ON ERR statement "well after" the program line that contains the <ON ERR GO TO> statement has been executed. Then the first error condition forces the abandonment of sequential line read, and the line that is designated in the ON ERR statement is executed. Therefore, if you use <ON ERR GO TO line>, it is best to terminate the ON ERR "whether an error exists or not". One DOS (AERCO FD-68) will carry forward a ON ERR condition when a new Basic Program is loaded without a computer reset. This can cause the next program loaded to give you an unpleasant suprise.

ON ERR programming can let the programmer do many things that cannot be achieved by any other means. But it is not good programming technique to leave a ON ERR condition dangling. I'll give a couple of examples that can be modified and used in different ways. First, the cure for the FD-68 programs that may be loaded during a ON ERR condition in a preceeding program.

```
  5 REM ** RESET a ON ERR
 10 ON ERR GO TO 9000: STOP
 20 STOP : REM **Line to begin
your programming.
9000 ON ERR RESET : GO TO 20
```

Let your auto run line be line 10. The ON ERR changes any existing ON ERR programming to line 9000, and then the STOP forces line 9000 to execute, where the ON ERR is cleared with RESET. Then your first real programming can begin at line 20.

The next illustration is more complex. Here we use ON ERR almost as we would use <IF THEN>. Say you need to present on screen the "status of a data base" contained in a character array "O$()". A subscript error would occur if O$ were not dimensioned. Then you may want to know the actual Dimension of O$ array. Then you may also want to know the last "cell of O$" that contains data (not empty spaces). We will use ON ERR within counters to do that in the following subroutine.

```
  5 REM ** Use ON ERR to find
the Dimension of a CHR$ Array.
 10 LET a=500: FOR n=1 TO a: ON
 ERR GO TO 9000: LET b=LEN o$(n)
: NEXT n
 20 CLS : STOP : REM **Line to
begin your Basic programming.
9000 ON ERR RESET : IF n<2 THEN
PRINT "O$ is not Dimensioned."
9005 IF n>2 THEN PRINT "The Dime
nsion of O$ is> "; FLASH 1;"O$("
;n-1;")"''
9010 IF n>1 THEN FOR n=1 TO n-1:
 ON ERR GO TO 9020: IF O$(n)(1)=
" " THEN STOP
9015 NEXT n
9020 ON ERR RESET : IF n=1 THEN
PRINT "O$ has no data": STOP
9025 ON ERR RESET : IF n>1 THEN
PRINT "The last cell of O$ that
has  Data is "; FLASH 1;"O$(";
n-1;")"
9030 BEEP 3,16: LET n=a: GO TO 2
0
```

Each of this routine's actions to check the status of O$ serves the same purpose as an <IF THEN>, except that <IF THEN> cannot be used because it could result in a subscript error. So ON ERR allows us to check the status of a data base "whether the data base exists or not". It is important to note that at the end of the <ON ERR> Subroutine, the <ON ERR> condition is cleared (RESET). Then any subsequent error will be reported as normal with a "report code" at the screen

bottom.

Lets do one more that is simpler. This time we will check the status of a simple string content (A$). A subscript error will occur if A$ has not been initialized in this example. <10 PRINT "The Data Base is "; LEN A$;" bytes">. The ON ERR routine would be:

```
  10 ON ERR GO TO 9000: PRINT "T
he Data Base is ";LEN a$;" bytes
.": ON ERR GO TO 9010: STOP
  20 STOP : REM * The next line.
9000 ON ERR RESET : CLS : LET a$
="": GO TO 10
9010 ON ERR RESET : GO TO 20
```

If A$ has not been initialized, the ON ERR trips to line 9000. But if A$ has data the LENgth of A$ is reported and the second ON ERR designates a new line to GO TO if an error exists. Then STOP triggers the GO TO 9010. In either case <ON ERR RESET> clears the ON ERR programming. It might be worth noting that <ON ERR RESET> clears the error condition.

But, <ON ERR RESET> is ignored if the error condition has already been cleared. There is no adverse effect if <ON ERR RESET> is encountered. The follow on programming in the line is executed whether or not the RESET actually reset an error or not.



CORNUCOPIA

Gr. Myth. a horn of the goat that suckled Zeus: it would become full of whatever its owner wanted

GETCHA MAGNIFYIN GLASSES OUT! THIS ISSUE WILL BRING SOME ANNOUNCEMENTS ABOUT SOFTWARE THAT IS AVAILABLE. NOW DONT FUSS ABOUT THE SMALL PRINT. UPDATE IS DOIN THIS AS A PUBLIC SERVICE AND TRYING TO GET AS MANY IN THAR AS POSSIBLE WHILE MAKING THE PRINT CLEAR NUFF TO SPY GLASS UP. THE MAIN THING IS TO SEND A SASE TO THE FOLKS AND GET BACK MORE INFO THAN WE CAN PUT IN A LITTLE COLUMN. - ED.

McBrine Computer Products

TS2068 SOFTWARE!

LIFE for the TS2068 v. 2.0 - See John Conway's revolutionary pattern-generating game for yourself! 3 times as fast as our original; includes 24 x 32, 48 x 64, and now 96 x 128 in one program, plus saving & loading and an improved keyboard routine.....$9.95

POWER PRINT - The ultimate in support for your Epson-compatible printer with Aerco, Tasman, or A&J parallel interface. Features: 100% correct LLISTings ‡ Accurate aspect-ratio b/w screen dumps in 2 sizes and two resolutions (192 x 256 and 64-col., or 192 x 512) ‡ Text-only screen dumps (both resolutions) for speed ‡ Print to memory ‡ Interrupt-driven print spooler ‡ TOTALLY RELOCATABLE! ‡ Specify interface and printer. (Will customize for your printer; send print code data.).....$9.95

BRAH 512 - Works in both 192 x 256 and 192 x 512, b/w only. Draw (w/ OVER or INVERSE) ‡ Line ‡ Box ‡ Square ‡ Circle ‡ Invert ‡ Block invert ‡ Block erase ‡ Merge screens 3 ways ‡ Rotate (scroll) screen in 8 directions, a pixel at a time ‡ Flip left/right or up/down ‡ Mirror 4 ways ‡ Convert back and forth between 32-col. and 64-col. modes in several ways ‡ Use joystick or keyboard ‡ Screen dumps from POWER PRINT.....$9.95

SUPERTOOLS - Remkill ‡ Header reader ‡ Renumber (with GO TO's and -SUB's) ‡ Variable dump ‡ Program tracer ‡ Finder (find string in BASIC program) ‡ All RELOCATABLE except tracer, partly relocatable.....$9.95

Postage paid on all orders ‡ Check or Money Order

SEND LSASE FOR FREE CATALOG!

McBrine Computer Products
514 S. Jackson St. ‡ Salisbury, NC 28144

SOFTWARE FOR THE QL

It is time to start thinking about an ugly word, "TAXES". "But'' you say, "this is only October". Wal, the next issue of UPDATE may be too late to start worrying. This is an unpleasant way of introducing you to a DEALER of SPECIALIZED SOFTWARE for the QL.

Peter Hale, of EMSOFT has announced the 1988 version of "TAX-I-QL", which has a template for your PSION ABASCUS Software. The TAX-1-QL program produces most of the IRS form schedules that you need, plus common forms and worksheets. Your taxes are computed from input data on the proper forms. A 16 page manual guides you through the task of inputing data to compute your or someone else's taxes.

Other EMSOFT titles are "MAILBAG", an address data base, "QLAND_LORD", which completely manages up to 99 property rental units and produces the required IRS Schedule E. Another is "TRUST_FUND" to manage your investments.

All EMSOFT softwares require 256K of RAM extension and are available on 80 track Floppy for $19.95, or Micro Drive Cart for $21.95. Write for catalog: EMSOFT, PO Box 8763, Boston, MA 02114-8763 (tel 617 889 0830).

ARCHIVE MASTER

ARCHIVE Database is a powerful, flexible and unique program that was supplied as a "bonus" with most new QL computer systems. However, due to it's complexitity and lack of complete documentation, it is perhaps the most misunderstood QL program...that is, up until now. Enter the new book, ARCHIVE MASTER, from Executive Workshop. ARCHIVE MASTER explores all of the inner workings of ARCHIVE, and contains tips & tricks to help you make ARCHIVE an even more powerful tool for personal use and business. Complete procedures are included to set up an Inventory program, a Customer/Supplier file, an Accounts Receivable/Invoicing program, a Mail List Manager, and much more! ARCHIVE MASTER is supplied with a durable three ring binder, and is priced at $44.95 (plus $3 S&H).

Order your copy from:
RMG ENTERPRISES
1419 1/2 7th Street
Oregon City, OR 97045
(503) 655-7484

EMSOFT

software for the QL

SOFTWARE CATALOG

APPLICATIONS SOFTWARE for the QL

MAILBAG	Address database you won't believe.	256K+
TAX-I-QL	Spreadsheet for IRS	384K+
QLAND_LORD	Manages up to 99 units and does Sched E (Now)	256K+
TRUST_FUND	Tracks investments and net worth (Soon)	256K+

Agents for ZX/TS SiriusWare
4K Word processor with
TS 2040 lowercase printer driver

Orders, Catalogues and Dealer info:

Estate Management Services
P.O. Box 8763, Boston, MA 02114-8763 (617) 889-0830

PROMISE LAND ELECTRONICS
Dan Elliott, *Owner*

Servicing COMPUTERIZED
electronic equipment

COMPUTERS / MODULES / PRINTERS / MONITORS			
Sinclair QL	TS-2068	TS-1500	TS-1000
16K Ram Pack	ZX-81	Memotech Modules	
Commodore 64	VIC-20	PLUS-4	C-16
Coleco ADAM	TI-99/4A	Commodore 128	

REPAIRS / UPGRADES / ASSEMBLING / TESTING

Write for prices:

Dan Elliott
Route 1, Box 117
Cabool, MO 65689

Ph. (314) 739-1712 evenings, Sunday - Thursday.
Ph. (417) 469-4571 weekends.

KEEP IN TOUCH



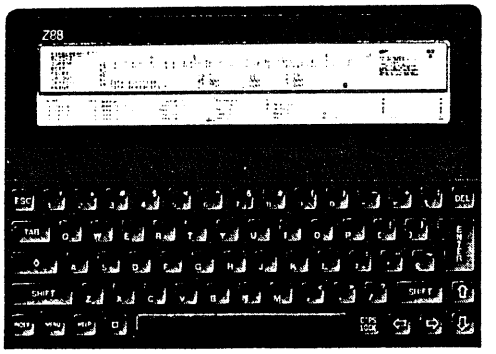
with
Q_LINK!
The Ultimate QL Terminal

US$19.95 + 3.00 shipping
[specify medium]

Meta Media Productions
726-West 17th
Vancouver BC
Canada V5Z 1T9

THE Z88 UNDER 2 LBS.

A Computer Without Compromise



• Where laptops compromise on display and RAM capacity to achieve portability, and desktops seem to equate price with power, the Z88 is a personal computer which makes no compromises • A CMOS-technology computer with the power to address 4 Mbytes of memory • A computer with a work-free display of 8 lines of 80 characters, an LCD screen which outdates all others, and a unique dynamic page map on screen • A computer with solid-state permanent storage • A computer with advanced word-processing, spreadsheet and ingenious time- and data-management software built-in • A computer which is completely self-contained, which gives you up to 20 hours active computing from just 4 AA batteries, yet which talks and listens to your IBM • A computer with a full-size keyboard, in a package less than the size of an 8½x11, with a total weight of less than 2 lbs. • The Z88. A computer without compromise.

Z-88 Computer	$479.00	SERIAL to PARALLEL I/F	$ 45.95
PC to Z-88 Linkup	45.00	1/2 MEG RAM	359.95
32K RAM	39.95	DATABASE S/W	119.95
32K EPROM	39.95	CARRYING CASE	19.95
128K RAM	89.95	Z-88 MAGAZINE	5.00
128K EPROM	89.95	X-MODEM S/W	89.95
RS 232 LEAD	19.95		

$5 SHIPPING ON Z-88. VISA/MC Accepted with 3% surcharge. C.O.D. charge is $2.25. If there are any questions, feel free to call or write.

SHARP'S, INC.
Route 10, Box 459
Mechanicsville, VA 23111
(804) 746-1664

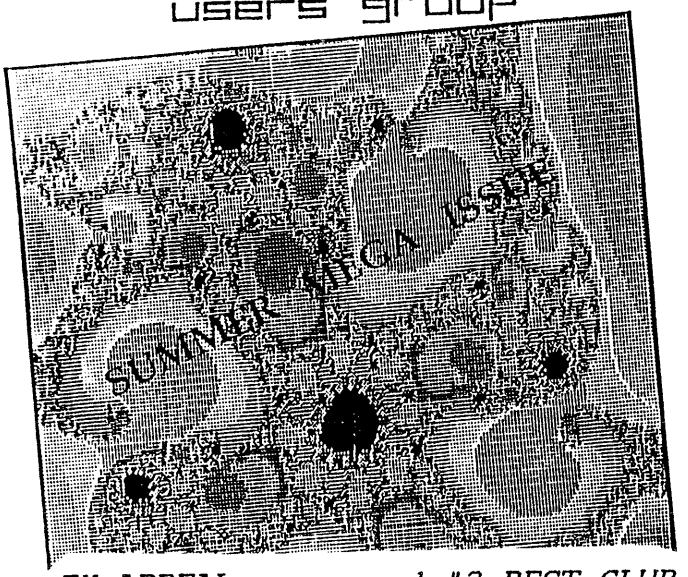
MasterCard VISA

Add $5 for shipping and handling
3% surcharge on credit cards

Summer/88

ZX-Appeal

vancouver sinclair users group



ZX-APPEAL was named #2 BEST CLUB publication by Time Designs panel. See re-print articles in this issue. ed.

VSUG

The Vancouver Sinclair Users Group has been in existence since 1982. We are a support group for the owners and users of all SINCLAIR and TIMEX computers.

Pres:- Gerd Breunung PH#(604) 931-5509
V/Pres:- Glenn Read
Sec:- Harvey Taylor
Treas. & N/L Editor:- Rod Humphreys

Our membership dues are only $15.00/year and may be sent to the Treasurer:

Rod Humphreys
2006 Highview Place
Port Moody, B.C., V3H 1N5

Members of VSUG receive a monthly issue of ZXAppeal - our newsletter.

ZXAppeal accepts advertising. Our **PREPAID** rates are:

$10.00 – full page
$8.00 – 1/2 page
$5.00 – 1/4 page

ZXAppeal is distributed to approx 30 other SINCLAIR User Groups throughout North America as well as overseas via the NETWORK. NETWORK correspondence may be directed to the Editor at the above address.

Copyright of all articles appearing in ZXAppeal is retained by the author with the understanding that other T/S User Groups may reprint any article appearing in ZXAppeal provided credit is given to the author and VSUG.

Reprinted from the May/88 issue of the Nite-Time News - the N/L of the Chicago Area T/S Users Group

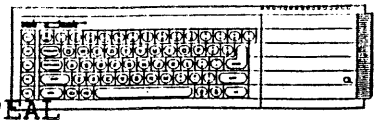
TELECOMMUNICATIONS ON THE QL

By Michael Mitchell
(Reprinted from QZX)

The Ql for all its power and versitility is limited in telecommunicating at 300 baud without a hardware link between the serial port and the modem. The following program which appeared in the Data Expansion or the Dallas/Fort Worth group will get you on line to a TNC or a telephone at 300 (or even 1200) baud.

```
  1 REMark        300/1200 baud terminal
  2 REMark             MIKE_TERM
  3 REMark           Developed by
  4 REMark          Michael Mitchell
  5 REMark        attribution requested
  6 REMark       Sinclair/Timex User Group
  7 REMark        Boston Computer Society
100 WINDOW 465,204,6,0
105 BORDER 1,0,7: LB=127
110 CLS: POKE 163976,255: CLS#0
120  PRINT#0,,"Choose Baud:   H=1200,  L=300
130 IF INKEY$(-1)=="h"THEN BAUD 1200: ELSE BAUD 300
140 CLS#0: OPEN#5,SER2i
150     PRINT#0,,"F5=QUIT","ALT"C="C","^G=BELL"
160 REPEAT TERM
170 TERMINAL: PRINT#0,,"Exit?(y/n)"
180  IF INKEY$(-1)=="Y" THEN EXIT term
190 END REPterm
200 CLOSE#5
205 PRINT#0,,"Re-run/Quit? (r/q)"
210  IF INKEY$(-1)=="R" THEN RUN: ELSE STOP
220 DEFine PROCedure terminal
230  REPeat key_loop
240   a=CODE(INKEY$(#5,0))&&LB
250    IF a=13 THEN PRINT
260    IF a=7 THEN BEEP 2000,15
270    IF>31 THEN PRINT CHR$(a)
280   b=CODE(INKEY$)
290    IF b=248 THEN EXIT key_loop
300    IF b=255 THEN b=0
310    IF b=0 THEN NEXT key_loop
320    IF b=10 THEN b=13
330    IF b=208 THEN b=7
340   PRINT#5, CHR$(b);
350  END REPeat key_loop
360 RETurn
370 END DEFine terminal

Save as MIKE_TERM_bas
```



Reprinted from ZX-APPEAL

Sound Experimentor

Rob Miles

Unlike most home computers the QL has a very limited noise making capability. What there is can only be dragged out by using the enigmatic BEEP command. This has 8 arguments, all of which are obscure. The only way to produce something vaguely melodic is by trial and error. The following quickie allows you to do exactly this.

```
100 REMark **** QL User -Beep experiment
110 REMark ****  Rob Miles 1985
120 duration=5000:pitch=100:pitch_2=0:grad_x=0:gra
d_y=0:wrap=0:fuzzy=0:random=0
130 REPeat sounder
140 CLS
150 set "Duration (-32768..32767) ?",duration,2
160 set "Pitch (0..255) ?",pitch,4
170 set "Pitch_2 (0..255) ?",pitch 2,6
180 set "Grad_x (-32768..15) ?",grad_x,8
190 set "Grad_y (-8..7) ?",grad_y,10
200 set "Wrap (0..32767) ?",wrap,12
210 set "Fuzzy (0..15) ?",fuzzy,14
220 set "Random (0..15) ?",random,16
230 AT 18,0:PRINT "Beep ";duration;",";pitch;",";p
itch_2;",";grad_x;",";grad_y;",";wrap;",";fuzzy;",
";random
240 BEEP duration,pitch,pitch_2,grad_x,grad_y,wrap
,fuzzy,random
250 dum$=INKEY$(-1)
260 END REPeat sounder
270 DEFine PROCedure set (name$,var,position)
280 LOCal buf$
290 AT position,0:PRINT name$;" ";var
300 AT position,0:PRINT name$;" ";
310 INPUT buf$
320 IF buf$<>"" THEN var=buf$
330 AT position,0:PRINT name$;" ";var;"     "
340 END DEFine set
```

Easel Print

D Duncan

The following 7 liner shows exactly how to use the graphics dump provided with EASEL. All you need is an EPSON compatible printer and the EASEL cartridge in mdv2.

```
120 MODE 8:PAPER 2:CLS
130 FOR n=1 TO 150
140 INK 7:FILL 1:CIRCLE n,50,20,.5,-n/2
150 INK 2:FILL 0:CIRCLE n,50,20,.5,-n/2
160 NEXT n
170 REMark Dump to Epson printer
180 a=RESPR(1024):LBYTES mdv2_gprint_prt,a
190 CALL a:OPEN #3,ser1:LIST #3:CLOSE #3
```

QL RAM extension

Re-print from ZX-APPEAL

Sinclair's QL has as standard a 128 K RAM, which sounds like a lot in comparison with most 64 K machines. Unfortunately, the software writers, in the knowledge that there is more than enough memory, have been rather wasteful in their work, so that at the end of the day, there is not all that much more in the QL than in the 64 K machines. So, you need more memory...

The accompanying circuit is an application of the TMS4500A as RAM extension for the 68008. This chip can drive a maximum of 128 K dynamic RAM and provides virtually everything: multiplexing of the address lines, $\overline{RAS}$, $\overline{CAS}$, and REFRESH.

The memory ICs are 64 K × 1 (128 or 256 refresh are both permitted) and have a speed of better than 150 ns. Since the QL uses a clock frequency of 7.5 MHz rather than the normal 8 MHz, such a RAM can run without wait cycles. An 8 MHz CPU that regularly has to carry out a wait cycle is appreciably slower than a 7.5 MHz type!

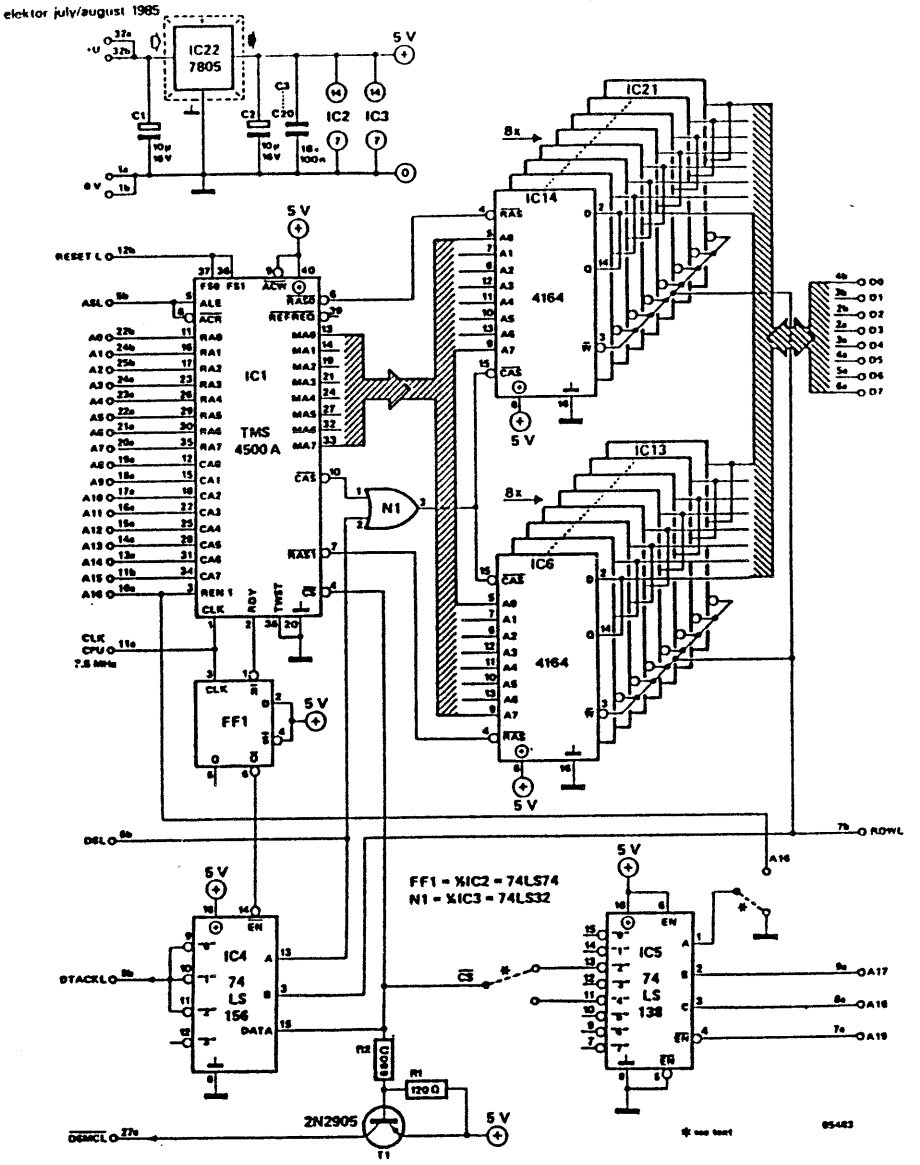
The 68000 family is provided with a data acknowledge input. As with other processors, the CPU places addresses and data onto the bus and indicates the validity with an address strobe and data strobe respectively. It continues to do so until the memory sends a $\overline{DTACK}$ signal. The present extension generates this signal with the aid of the LS156. Normally, this acknowledgment is given almost immediately, but it may happen that the 4500 is in the middle of a refresh. In that case, the CPU has to wait, which is arranged via the ready output (pin 2).

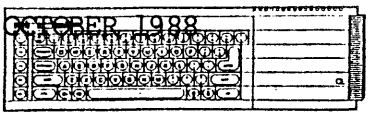
To prevent the QL waiting forever when an address is read that has no memory, the $\overline{DTACK}$ is generated internally: this must, however, be disabled for addresses where the RAM extension is located, and fortunately this can be done easily via $\overline{DSMC}$. By making this logic high as quickly as possible, the internal $\overline{DTACK}$ is cancelled.

If you cannot get the 2N2905 transistor, you may use a BS250, in which case resistor R_1 can be omitted and R_2 should be replaced by a wire link.

The circuit as shown is for the 128 K version. It is also possible to omit the eight RAMs connected to RAS1 and make a 64 K extension. Input A of the LS138 must then be connected to A_{16} and pin 11 instead of pin 13 must be used as $\overline{CS}$.

There is no 5 V supply available on the connector, but there is a 9 V line. This can be reduced to 5 V by a standard 7805. The current drawn depends on the types of RAM and will be 200...300 mA. It is important to decouple the supply lines properly: each RAM IC and the 4500 require a 100 n capacitor!





The Trump Card

Re-print from ZX-APPEAL

by David Hoshor

Reprinted from the Jan/Feb issue of The Ramtop - the N/L of the Greater Cleveland Sinclair User Group

The Trump Card is simply the most useful single addition you can get for the Sinclair QL. It combines a disk interface, 768 kilobyte RAM expansion, Toolkit II for SuperBASIC extensions, a screen dump, static and dynamic ramdisks and a dynamic printer buffer. When you consider that even the lowest cost 512 Kb RAM expansion and a disk interface will cost at least $250, and Toolkit II costs $70, the Trump Card is a bargain at $309.95. The entire unit only sticks out about three inches from the expansion port side of the QL, so it's compact. I got mine from Curry Computer in Glendale, Arizona.

The disk interface is pretty standard fare. It will allow you to store information on any combination of forty and eighty track drives, single sided or double sided, 5 1/4" or 3 1/2". The interface is fairly intelligent in that it will pick up on what format the disk was created in automatically. For example, if you have a 80 track, 5 1/4" double sided drive, the interface will automatically read from a 40 track, single sided disk. Of course, you can't make a single sided drive read from a double sided disk, or put a 3 1/2" floppy in a 5 1/4" drive since the interface can't change the limitations of your disk drives. But if you have an 80 track 5 1/4" double sided drive, you'll be able to read any disk with the exception of 3 1/2" disks. The interface is only able to support two floppy disk drives. That's its only shortcoming as far as I'm concerned.

The 768 Kb RAM expansion maxes out the QL. Added to the 128 Kb that are built into the QL, you have a whopping 896 Kb of RAM. (While the 68008 CPU can address 1 Mb of RAM, 128 Kb of memory in the QL are reserved for device addresses and ROMs.) You'll have room to run any program that has ever been written for the QL - probably several copies of the program. This can be very helpful on a multitasking machine like the QL. You'll also have plenty of room for ramdisks too.

The ramdisking capabilities of the Trump Card are really neat. The ramdisks come in two varieties, static and dynamic. The difference is that the static ramdisk has to be formatted, almost like a microdrive cartridge, and the dynamic ramdisk can be created by just using the device name. For example, to create a static ramdisk with two hundred, 512 byte blocks of space, just enter:

```
FORMAT RAM1_200
```

A 100 Kb section of memory will be set aside for use by the machine to use just like a microdrive or floppy disk, but will be located in fast RAM. To create a file in dynamic ram, it's just as simple as using the device's name. To copy a file to ramdisk ram3_ just enter:

```
COPY dev_filename to RAM3_filename
```

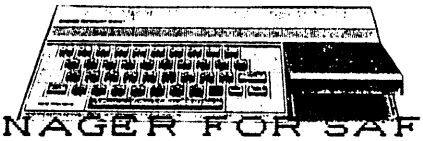
A bit of memory will be set aside in RAM as ram3_filename. It will grow only large enough to hold the file(s) that have been send to ram3_. The difference between the static and the dynamic ramdisks is that the dynamic ramdisks are more likely to fragment the memory of the QL. If the memory becomes sufficiently fragmented, it can degrade the performance of the machine because the operating system will have to scrounge around to find blocks of memory. Up to eight ramdisks can used with the Trump Card. One other neat feature of the ramdisk is that it can make a very fast image of everything on a microdrive. It will copy everything on the microdrive in just about fifteen seconds.

One of my favorite features of the Trump Card is its printer buffer. It's a snap to use and can really save time. In its simplest form, you just use the device name "PRT" instead of "SER". What happens is that whatever you are sending to the printer is sent in its entirety to a buffer in RAM. The QL then sends bytes to the printer as a background task. The advantage to this is that you can return almost immediately to an editing session if you send a file to "PRT" from The Editor or Quill. If you send a file to "SER" from an editor, you must wait until the printer has finished printing before you can continue editing. Several files can be sent to the printer buffer and they will be sent in the proper order. There is a method of changing the device that you want to have the "PRT" device use. Unless you have a parallel interface that doesn't use the serial port, it's hardly likely that you'll change it.

The printer dump is pretty good. It supports about any Epson dot matrix printer plus a few other printers including the Brother HR4, Olivetti JP101, Seikosha GP-100A, GP-250X, and GP-700A, and the Canon PJ1080A. You can specify what section of the screen to copy, and there is a provision for using a "hotkey" to copy any screen from any program that you might be running. There's also documentation on how to open a printer device and get a printer dump from programs written in near any computer language - not just SuperBASIC. I've tried it from "C" and it does work.

Toolkit II is a collection of SuperBASIC extensions. Some of them are much more useful than others, but on the whole, they are very worthwhile. To me, some of Toolkit's most useful commands are "ed", a screen based SuperBASIC editor; "wcopy", a file copier that allows the use of wildcards; "wstat", a wildcard directory command that also provides the file size and last update time of files; "spif", another printer buffer or rather "spooler" that works great when used with "prt"; and an assortment of job control commands that allow the user to monitor, change the status of, and remove jobs that are the in the QL. Toolkit II allows the use of default directories, has clock features including alarm clock(s), has improved SuperBASIC error handling, permits the user to define "hotkeys", altkey/ single key combinations to enter large strings or commonly used commands, and has an easy way to repeat of the last command typed. Some of the more obscure commands deal with networking, direct unbuffered access to the various i/o devices, and memory management commands. All in all, its a pretty impressive array of extensions to SuperBASIC. Once you start using the Toolkit II commands, you'll never want to use a machine without them. An additional benefit of having Toolkit II fitted on the Trump Card is that it leaves the ROM cartridge slot on the back of the QL free for other ROMs.

When you consider the features that the Trump Card offers, I feel that it is an unbeatable combination.



A MULTI_MANAGER FOR SAFE DOS V2.52

Bob Hartung, 2416 N. Co. Line, Huntertown, IN 46748

Since the final additions have been made to SAFE DOS, I have had several requests to write a "do-all" disk management program that includes all these features, plus the file-indexing provided by DOS_DEX and AUTO_DEX. It seemed to me to be more practical to call these file-indexing routines and the disk-verify routine from the initial (LOAD) menu and from the MULTI_MANAGER menu than to integrate them all into one program.

To make it possible to utilize only one set of data for both DOS_DEX and AUTO_DEX, the original DOS_DEX was revised to save DATA c$() arrays instead of the entire routine with variables. This in turn required finding a way to preserve current variables whenever a new file directory was installed by the machine code routine. I utilized the UDG file for this, eg: POKE USR "D and PEEK USR "d" which saves several bytes compared to using the usual PEEK/POKE syntax. The underlined "D must be entered in GRAPHIC mode but not the lower-case "d".

Some have wondered why the fast FOR/NEXT loop was not used in these routines. Actually, because of the various key-stroke tests and item-numbering very little gain in display speed is obtained by this, but for anyone who is interested, the eighth REM byte may be changed to a 14 to allow for a LET m=SGN PI definition before the DIM c$ (VAL "178",VAL "20"). An easy way to do this with the original DOS_DEX listing is to insert a line 6 STOP. Key RUN, then POKE (a+7),14. Remove line 6, make the m definition between CLEAR and DIM c$() and revise all the m loops.

Note that this fast FOR/NEXT definition is NOT provided in the revised DOS_DEX listing given below. To move the REM statement to line 0 if you are starting with the original listing, POKE 26711,0. (Refer to April UPDATE for poking the machine code to a REM line from scratch.) As given, the listing provides for a two-drive system, 0 and 1, at lines 1 and 522, which may be changed if you prefer to designate other drives. Omit these drive designations if you have a single-drive system.

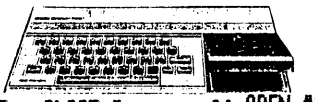
After the ERASE function appeared for SDOS v2.52, I found that the CHR$ 229 " RESTORE " test I had used in the original AUTO_DEX, VERI_DISK, and MARK_MOVE listings would not work after an ERASE had been used on a disk. Upon checking with John Oliger, I learned that a CHR$ 128 in the first byte of the first unused file directory is what must be looked for. So, in all the original listings, wherever " RESTORE " appears the respective lines should be changed to IF CODE c$(n)) VAL "127" ... or IF CODE c$(K)) VAL "127"... or IF CODE c$(M)) VAL "127" or IF CODE c$(m)) VAL "127"....

The MULTI_MANAGER listing does not utilize the fast FOR/NEXT loop either, so when entering the listing be sure that the eighth byte of the line 1 REM is an 8 and not a 14 as described previously if you are starting with a copy of the DOS_DEX machine code in line 1 REM. The program prompts should be self-explanatory--I HOPE! Note the warning given in the SDOS User Manual (p. 37a in the final edition) that the MOVE / command to transfer an entire disk must be used only with identical drives, otherwise the MARK_MOVE method is the only safe way to do it. The MULTI_MANAGER menu assumes that DOS_DEX resides in file zero and that AUTO_DEX and VERI_DISK are on the same disk, preferably with AUTO_DEX in file 1.

These routines are available on an UPDATE issue disk.

TWO-DRIVE DOS_DEX WITH DATA-SAVE

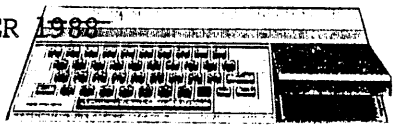
```
  0 REM STEP ?*K\?? FOR ! &? CLOSE #
  1 POKE USR "D",SGN PI: LET /d=NOT PI
  2 CLEAR : DIM c$(VAL "178",VAL "20"): LET t=VAL "10":
 LET s=SGN PI: LET w=s+s: LET o=NOT PI: LET d=PEEK USR "
d"
  5 LET a=VAL "PEEK 23635+PEEK 23636*256+5": LET c=INT
(a/VAL "256"): POKE VAL "23549",VAL "195": POKE VAL "235
50",a-(c*VAL "256"): POKE VAL "23551",c: LET fi=USR VAL
"23549": LET row=w: LET col=VAL "9": LET n$=c$(VAL "178"
, TO VAL "16")
 20 LET a$="": DIM f$(VAL "6",w): LET f$(w)=" DATA n":
LET f$(INT PI)=" DATA $": LET f$(w+w)="CODE ": LET f$(VA
L "5")="ABS ": LET f$(VAL "6")="VAL "
 300 CLS : PRINT AT o,o;" Disk#";d-s;": ";n$;#o;"DOSDEX
d0/d1:""Key: NEXT CAT SAVE LOAD (1)#";d
 400 LET f=s: LET c=INT (fi/VAL "18"): LET dif=INT ((fi/
VAL "18"-c)*VAL "18"+VAL ".4"): LET loop=VAL "17": ON ER
R GO TO VAL "700"
 410 LET q$="p": LET it=s: IF loop>=fi THEN LET loop=fi:
 GO TO VAL "425"
 415 FOR i=s TO c: FOR m=o TO loop: PRINT AT row+m,t;a$;
AT row+m,t-LEN STR$ it;it;" ";c$(it, TO t);" ";f$(CODE c
$(it,VAL "11")+s): LET it=it+s: NEXT m: GO SUB VAL "500"
: NEXT i: FOR i=s+s TO VAL "19": PRINT AT i,t;a$: NEXT i
: IF NOT dif THEN GO TO VAL "410"
 425 IF f THEN FOR m=o TO dif-s: PRINT AT row+m,t;a$;AT
row+m,t-LEN STR$ it;it;" ";c$(it, TO t);" ";f$(CODE c$(i
t,VAL "11")+s): LET it=it+s: NEXT m: IF loop>=fi THEN LE
T f=o
 427 IF NOT f THEN LET it=fi+s
 430 GO SUB VAL "500": GO TO VAL "410"
 500 FOR L=o TO m-s: PRINT AT row+L,col-w;"  "; INVERSE
s;")": IF q$<>"" THEN FOR a=s TO PI↑PI: NEXT a: LET q$="
"
 510 LET a$=INKEY$: IF a$="" THEN LET q$="p": GO TO VAL
"510"
 520 IF CODE a$=VAL "13" THEN GO TO VAL "600"
```

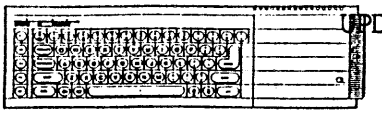



```
 522 IF a$="C" THEN LET /d=s: OPEN #w,"P": CAT : CLOSE #
w: LET /d=o: GO TO w
 524 IF a$="L" OR a$="l" THEN LOAD /STR$ d DATA c$(): LE
T n$=c$(VAL "178", TO VAL "16"): LET d=CODE c$(VAL "177"
): LET fi=CODE c$(VAL "177",w): GO TO VAL "300"
 526 IF a$="S" THEN LET d=d+s: POKE USR "D",d: LET c$(VA
L "177",s)=CHR$ d: LET c$(VAL "177",w)=CHR$ fi: SAVE //S
TR$ (d-s) DATA c$(): GO TO w
 528 IF a$="N" THEN INPUT " INPUT NEXT #";d: POKE USR "D
",d: GO TO w
 530 PRINT AT row+L,col-w;"   ": NEXT L: LET a$="
          ": RETURN
 600 ON ERR RESET : CLS : LET pos=it-m+L: LET d$=c$(pos,
 TO t): LET a=CODE c$(pos,t+s): IF NOT a THEN LOAD /d$
 610 IF a=s THEN LOAD /d$ DATA n()
 620 IF a=w THEN LOAD /d$ DATA n$()
 630 IF a=INT PI THEN LOAD /d$CODE
 640 IF a=w+w THEN LOAD /d$ABS
 650 IF a=VAL "5" THEN LOAD /d$VAL
 700 ON ERR RESET : PRINT #NOT PI; FLASH SGN PI;" FILE N
OT FOUND ": PAUSE VAL "100": GO TO SGN PI


                    MULTI_MANAGER


   1 REM STEP ?*K\?? FOR ! &? CLOSE #
   2 ON ERR RESET : CLEAR : DIM c$(VAL "178",VAL "20"):
LET s=SGN PI: DIM m$(VAL "177",s): LET f=VAL "4": LET t=
VAL "10": LET o=NOT PI: LET a$="": INK VAL "7": PAPER o:
 BORDER o: CLS
   5 PRINT TAB t;"MULTI-MANAGER"''TAB f;"1 - Mark/move f
iles";TAB f;"2 - Move ALL files";TAB f;"3 - Re-name file
s";TAB f;"4 - Re-name disk";TAB f;"5 - ERASE files";TAB
f;"6 - VERI_DISK";TAB f;"7 - DOS_DEX";TAB f;"8 - AUTO_DE
X";TAB f;"9 - SAVE MULTI_MGR";TAB f;"0 - LPRINT CAT ": P
AUSE o: LET opt=CODE INKEY$-48: IF opt)9 THEN GO TO 5
  10 IF opt=6 THEN LOAD /"VERI_DISK"
  15 IF opt=7 THEN LOAD
  20 IF opt=8 THEN LOAD /"AUTO_DEX"
  25 IF opt=9 THEN CLEAR : SAVE /"MULTI_MGR" LINE 2: GO
TO 2
  30 IF opt=0 THEN INPUT "Printer ON, then key ENTER ";a
$: LET /p=o: OPEN #2,"P": CAT : CLOSE #2: LET /p=t: GO T
O 2
  35 DIM f$(VAL "6",s+s): LET f$(s+s)=" DATA n": LET f$(
INT PI)=" DATA $": LET f$(VAL "4")="CODE ": LET f$(VAL "
5")="ABS ": LET f$(VAL "6")="VAL "
  40 INPUT "No. of SOURCE drive (0-3): ";source''"No. of
DESTINATION drive (0-3): ";dest: IF source=dest OR sourc
e)3 OR dest)3 THEN GO TO 40
  45 LET s$="Place source disk in DRIVE "+STR$ source+"
&  target disk in DRIVE "+STR$ dest: LET k$="KEY ENTER T
O "
  50 PRINT #o;s$;"          Key any CHR$ TO CAT source d
isk or key ENTER TO NOT CAT ": LET /d=source: PAUSE o: I
F CODE INKEY$()VAL "13" THEN OPEN #2,"P": CAT : CLOSE #2
  60 IF opt=1 OR opt=2 THEN INPUT "": PRINT AT o,t;"FILE
-MOVE",#0;" FORMAT target disk (drive #";dest;")   y/n?"
: PAUSE o: IF INKEY$="y" THEN INPUT " INPUT FORMAT   ↑ t
itle to arrow";a$: LET /d=dest: FORMAT /a$: LET /d=sourc
e: OPEN #2,"P": CAT : CLOSE #2:
PAUSE 300
  65 IF opt=2 THEN INPUT "Key ENTER for DEFAULT t (track
s) OR enter no. of tracks to move: ";t$: IF t$()"" THEN
LET /t=VAL t$: MOVE /: LET /t=83: GO TO 2: REM  Make t
re-set definition=default
  70 IF opt=2 THEN INPUT (s$'k$);" MOVE ALL FILES ";a$:
LET /d=source: MOVE /: GO TO 2
  75 IF opt=5 THEN CLS : PRINT AT 19,o;"Mark ONLY last f
ile to ERASE andrepeat for any others --        Key 7 to
 proceed": PAUSE o: IF INKEY$()"7" THEN GO TO 2
  80 CLS : IF opt=s THEN PRINT AT 20,s;"Key M to auto-ma
rk ALL files": PAUSE o: LET a$=INKEY$: IF a$="M" THEN FO
R m=s TO 177: IF CODE c$(m))VAL "127" THEN GO TO 90
  85 IF opt=s AND a$="M" THEN LET m$(m)=" ": NEXT m: GO
TO 90
  90 CLS : PRINT #o;"Key:SPACE to mark CHR$ to review0 t
o revise "+k$+(" MOVE " AND opt=1)+("RE-NAME" AND opt=3)
+(" ERASE " AND opt=5)
 100 PRINT AT 20,s,,: LET a$="": DRAW INK s;VAL "255",o:
 DRAW INK s;o,VAL "175": DRAW INK s;VAL "-255",o: DRAW I
NK s;o,VAL "-175"
 200 LET a=VAL "PEEK 23635+PEEK 23636*256+5": LET c=INT
(a/VAL "256"): POKE VAL "23549",VAL "195": POKE VAL "235
50",a-(c*VAL "256"): POKE VAL "23551",c: LET fi=USR VAL
"23549": LET row=s+s: LET col=VAL "9": LET n$=c$(VAL "17
8", TO VAL "16")
 210 IF opt=4 THEN PRINT AT t,f;"RE-NAME DISK in drive #
";source: INPUT "OLD disk name: ";(n$)''"NEW disk name: "
;b$: LET /d=source: RESTORE /b$: GO TO 2
 300 IF n$(LEN n$)=" " THEN LET n$=n$( TO LEN n$-s): GO
TO VAL "300"
 320 PRINT AT o,VAL "16"-(LEN n$/VAL "2"); OVER s; INK V
AL "5";n$;AT o,VAL "8"; OVER s;"________________": PAPER
 o
 405 LET L=o: LET f=s: LET c=INT (fi/VAL "18"): LET dif=
INT ((fi/VAL "18"-c)*VAL "18"+VAL ".4"): LET loop=VAL "1
7"
 410 LET q$="p": LET it=s: IF loop)=fi THEN LET loop=fi:
 GO TO VAL "425"
 415 FOR i=s TO c: FOR m=o TO loop: PRINT AT row+m,t-f;a
$;AT row+m,t-LEN STR$ it;it;m$(it);c$(it, TO t);" ";f$(C
ODE c$(it,VAL "11")+1): LET it=it+s: NEXT m: GO SUB VAL
"500": NEXT i: FOR i=s+s TO VAL "19": PRINT AT i,t-f;a$:
 NEXT i: IF NOT dif THEN GO TO VAL "410"
 425 IF f THEN FOR m=o TO dif-s: PRINT AT row+m,t-f;a$;A
T row+m,t-LEN STR$ it;it;m$(it);c$(it, TO t);" ";f$(CODE
 c$(it,VAL "11")+1): LET it=it+s: NEXT m: IF loop)=fi TH
EN LET f=o
 427 IF NOT f THEN LET it=fi+s
 430 GO SUB VAL "500": GO TO VAL "410"
 500 FOR L=o TO m-s: PRINT AT row+L,col; INVERSE s;")":
IF q$()"" THEN FOR a=s TO PI↑PI: NEXT a: LET q$=""
 510 LET a$=INKEY$: IF a$="" THEN LET q$="p": GO TO VAL
"510"
 512 IF a$="0" THEN LET m$(it-m+L)=" ": PRINT AT row+L,c
ol+SGN PI;" "
 515 IF a$=" " THEN LET m$(it-m+L)=" ": PRINT AT row+L,c
ol+SGN PI;" "
 520 IF a$=CHR$ VAL "13" THEN GO TO VAL "550"
```





```
 530 PRINT AT row+L,col-s-s;"   ": NEXT L: LET a$="
           ": RETURN
 550 GO TO 600*(opt=1)+700*(opt=3)+800*(opt=5)
 600 BORDER VAL "7": PAPER VAL "7": INK 0: CLS : INPUT (
s$)'(k$);"start MOVE ";a$: CLS : PRINT FLASH 1;" MOVING!
  DO NOT STOP till end  "
 601 FOR M=1 TO VAL "177"
 602 IF CODE c$(M)>VAL "127" THEN GO TO VAL "670"
 605 PRINT AT 1,1;"File: ";m
 610 LET d$=c$(M, TO 1): LET a=CODE c$(M,VAL "11"): IF N
OT a AND m$(M)=" " THEN MOVE /d$ TO dest
 615 IF a=1 AND m$(M)=" " THEN MOVE /d$ DATA TO dest
 620 IF a=VAL "2" AND m$(M)=" " THEN MOVE /d$ DATA $ TO
dest
 630 IF a=INT PI AND m$(M)=" " THEN MOVE /d$CODE TO dest
 640 IF a=VAL "4" AND m$(M)=" " THEN MOVE /d$ABS TO dest
 650 IF a=VAL "5" AND m$(M)=" " THEN MOVE /d$VAL TO dest
 660 NEXT M
 670 PRINT AT 0,0,,AT 1,1; FLASH 1;" MOVE OVER ": INPUT
(k$);" NEXT MOVE "'"    7 FOR MENU ";a$: IF a$="" THEN P
APER 0: BORDER 0: INK 7: DIM m$(177,1): GO TO 90
 680 RUN
 700 REM Re-name files
 702 LET o$="Old name: ": LET n$="New name: "
 705 FOR m=1 TO 178
 708 IF CODE c$(M)>VAL "127" THEN GO TO 9999
 710 LET d$=c$(M, TO 1): LET a=CODE c$(M,VAL "11"): IF N
OT a AND m$(M)=" " THEN INPUT (o$;d$)'(n$);b$: RESTORE /
d$ TO b$
 715 IF a=1 AND m$(M)=" " THEN INPUT (o$;d$)'(n$);b$: RE
STORE /d$ DATA TO b$
 720 IF a=1+1 AND m$(M)=" " THEN INPUT (o$;d$)'(n$);b$:
RESTORE /d$ DATA $ TO b$
 730 IF a=INT PI AND m$(M)=" " THEN INPUT (o$;d$)'(n$);b
$: RESTORE /d$CODE TO b$
 740 IF a=VAL "4" AND m$(M)=" " THEN INPUT (o$;d$)'(n$);
b$: RESTORE /d$ABS TO b$
 750 IF a=VAL "5" AND m$(M)=" " THEN INPUT (o$;d$)'(n$);
b$: RESTORE /d$VAL TO b$
 760 NEXT M
 770 GO TO 9999
 800 REM Erase files
 802 FOR M=1 TO 178
 805 IF CODE c$(M)>VAL "127" THEN GO TO 9999
 810 LET d$=c$(M, TO 1): LET a=CODE c$(M,VAL "11"): IF N
OT a AND m$(M)=" " THEN ERASE /d$
 815 IF a=1 AND m$(M)=" " THEN ERASE /d$ DATA
 820 IF a=1+1 AND m$(M)=" " THEN ERASE /d$ DATA $
 830 IF a=INT PI AND m$(M)=" " THEN ERASE /d$CODE
 840 IF a=VAL "4" AND m$(M)=" " THEN ERASE /d$ABS
 850 IF a=VAL "5" AND m$(M)=" " THEN ERASE /d$VAL
 860 NEXT M
9999 CAT : PAUSE 500: GO TO 2
```

A Conglomeration of QL TIPS

MUCHODAT

A crazy title for a crazy situation. This is a little tip on FORMATTING MICRO DRIVE CARTRIDGES. While formatting a bunch of carts just received in the mail there were two that just wouldn't take a format. First, did you know that a title isn't necessary to format a mdv cart? format mdv1_ ENTER will do it. I never could see much value to a title that one never sees anyway. Anyway, after using a short title "dta", as: < format mdv1_dta > I had formatted 8 carts and there were two that were cranky. I tried about 5 formats on each to no avail. Now that exercise took about 2 hours of fooling around. Then I said, "just one more time". This time I whacked in: format mdv1_muchodat ENTER. Lo and behold, the cartridge formatted perfectly! So I tried the other cranky one, with: format mdv1_muchodat. It formatted too! Now ain't that a crazy one! There is more. After that trick, I dug up three more mdv carts that I couldn't format before and had stashed away. Two of the three formatted with muchodat as a title. There is something fishy bout this! Maybe theres something about a long title which makes the thing format? Beats me! But there's the tip for what it's worth.

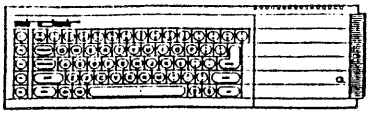
INITIALIZING TRUMP AND SPEED SCREEN
IN BOOT PROGRAMS

Right now I'm gonna discuss HOW TO EASILY MAKE YOUR MICRO DRIVE OR DISK DRIVE BOOT PROGRAMS INITIALIZE THE TRUMP CARD. If you dont have Trump, this won't be a dunnin you no good unless you just want to read about it.

The Trump Card has a lot of additional Super Basic Tools to work for us. One is a routine to copy a WHOLE mdv cart, all programs instead of just one at a time. This can save about 30 minutes of copying single files. First though, lets ramble a bit.

When we first turn on the computer it can see all of that additional RAM in the Trump card, but it dont know nothing about the new Super Basic commands that the Trump has tucked away in its own ROM chips. The whole bunch of Trump Card tools must be initialized once. Then after doing that, from then on all of the extra basic functions are available for use.

These Trump utilities must be initialized by a direct command of: TK2_EXT ENTER. Now if you allow the auto boot in of a program



from either Micro Drive or Disk Drive, you may not be given the opportunity to enter the TK2_EXT command. Some programs just wont let you break in to do that.

So, the thing to do is to modify the boot program so that the first program line contains the command: TK2_EXT.

Now if you also have SPEED SCREEN, you should also add its initialization command. That would be: < TK2_EXT: _SPEED 2 >. Then the BOOT program that has these commands in the first line of programming should be re-saved. After this, every time you load that particular mdv cart (or disk), the Trump utilities will be initialized and the Speed Screen will be turned on.

Now you can go ahead and TRY to do that but RE-SAVING the boot program will give you a report at the bottom "already exists". So lets be a showin how to dun it- next in 1,2,3, order:

1. RESET the QL "without a mdv cart" installed, then hit F1.
2. Install a write enable tab on the QLWP cartridge, and install it in mdv1.
3. Type; load mdv1_boot ENTER. The boot program will load but wont auto run.
4. Type; list ENTER.
5. Type; edit 1 ENTER. Now line 1 of the listing will appear at screen bottom. It will read; < 1 CLEAR >
6. Type onto the line to make it read; < 1 CLEAR:TK2_EXT: _SPEED 2 > Then touch ENTER, and the line should appear, as above, in the listing.
7. Type: tk2_ext ENTER. This initializes the Trump Card so that the next step will work.
8. Type: save mdv1_boot ENTER. This will execute one of the TRUMP routines to give a "SAVE OVER" an existing program in micro drive.
9. A prompt will appear, "OK to Overwrite? y/n?" Touch "y"
10. Now your new boot program replaces the boot program in the mdv cart, and the next time you use this cartridge the TRUMP and SPEED SCREEN are automatically initialized.

You can do the same procedure for each of your mdv cartridges. I advise that you make back up copies of all original mdv carts and do this modification on the Back Up copies.

Later, in another issue, we will explore making a "Library BOOT Menu" utility to do the above Trump and Speed Screen initialization and then give a menu of MDVs or DISKs in your program library to select from.

COPY A COMPLETE MICRO DRIVE CARTRIDGE With TRUMP

QL Basic does not provide simple commands to COPY a complete mdv cartridge, only single files or programs. Later in this issue Bill Cable will present a utility to copy whole mdv cartridges WITHOUT Trump card or Sandy board utilities. This procedure is for using the Trump or Sandy Super Tools which are dormant when you first turn on the QL.

First, reset the QL, and leave MDV 1 vacant. When F1 is pressed, you'll get the flashing cursor, which means that you can type in program lines, or do Direct commands. Now we will do the steps with a few direct commands to enter into the QL.

First though, INSTALL a formatted mdv cart in mdv1 (with write enable tab installed). For safety sake, remove the write enable tab of the new qlwp boot cart that we just made (above), and we will copy it. Install it in mdv2.

Type in the following direct commands, one after another, using ENTER after each:

tk2_ext ENTER
wcopy mdv2_ TO mdv1_ ENTER

What happens next is a prompt with several choices. The prompt will read exactly as follows:

MDV2_CLONE TO MDV1_CLONE Y/N/A/Q

Touch "A" for ALL, and the system takes over and copies all files and programs from mdv1 to mdv2.

Back to Business with the QL

OK, if everything has worked out, you now have two copies of your new QLWP program which has the modified boot program that comes up and initializes TRUMP Super Tool Kit, and SPEED SCREEN. These two cartridges are in the microdrives and youre QL is in the BASIC mode.

Now lets get the QUILL Word Processor up and running. Type: lrun mdv1_boot ENTER. Note that a while back we loaded the boot program to modify it by a command: < load mdv1_boot >, which just loaded in the boot program without letting it auto run. < lrun mdv1_boot > makes it load in and run. This time when the qlwp boot program loads in the TRUMP and SPEED SCREEN will be initialized. BUT, I am told that the QLWP program does not use TRUMP utilities. Never the less, you can use these procedures to modify boot programs that do use the TRUMP Super Tools.

655 N. W. Territorial Rd.
Canby, OR 97013
July, 11 1988

TS-2068 UP-DATE
1317 Stratford Ave.
Panama City, FL 32404

THE PLOTTER is the newsletter of the OCATS User Group, located in OREGON, and is one of the better Club publications, due largely by the expert writing of Dick Wagner and Rod Gowen, both of whom are Sinclair Activists. Thanks Dick!

Dear Bill:

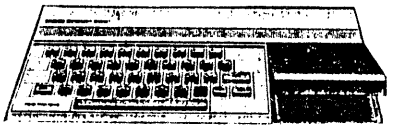
Here are two articles for UP-DATE magazine. The one on using LPRINT CHR$ was based on an inquiry to John Oliger. I was having difficulty in conveniently using OUT 127 with lots of DATA in printer graphics. He suggested trying these two pokes which he hadn't used (he thought they should work). Well, they did the trick and now I can copy programs direct from source as far as LPRINT CHR$ is concerned. He suggested sending you the idea if it did work.

The second article is from a suggestion in an EPSON printer manual for duplicating DATA in printer graphics. Because the loop for repeating numbers is tied to the numbers produced by READ, the same loop can be used for several sets of repeating numbers. As an example, the DATA in line 90 can be extended from the 60 to 120 dots by duplicating what is there. Line 20 would be changed from "60" to "120". This should produce a line twice as long by duplicating the original, and just by making these 2 line changes.

Yours truly,

Dick

Dick F. Wagner
Editor, THE PLOTTER

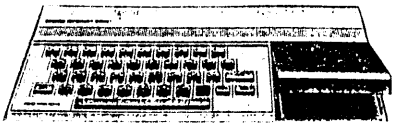


LPRINT CHR$ WITH OLIGER PRINTER DRIVER

Readers who have programmed in BASIC with the OLIGER printer driver use the LET/P=0 command to send data to their large printer. This command works fine for LPRINTing text and messages but not so good for printer codes. When a programmer wishes to use printer codes the accepted method is to go direct through port 127 by use of the OUT command. For reference see the OLIGER print driver manual and parts of Bill Jones's Smart Text program plus UP-DATE (Printer Programming, 1988)

There is a way that follows conventional programming methods such as LPRINT CHR$ n where "n" is a character code number, or using a series of CHR$s to make up a printer code such as LPRINT CHR$ 27; CHR$ 52 (EPSON Italics ON). See your printer manual for codes used by your printer. Probably examples are also displayed in your manual.

According to John Oliger the procedure for using LPRINT



CHR$ n is as follows: start with LET/P=0 and then make these 2 pokes, 23300,60 and 23001,3 in the immediate mode. Then program accordingly, possibly try examples from your printer manual. When RUN the programs should perform properly. This change will not permit LLIST to be used so just use LET/P=0 and make a LLIST. Make the pokes again to set up the program to RUN. What happens is that these two pokes turn off the BASIC TOKEN crunching.

If the reader's printer provides a HEX dump capability then a simple test of a printer driver is easily made. With LET/P=0, enter this one line program:

```
10 FOR N=0 TO 255: LPRINT CHR$ N;:NEXT N
```

Set up the printer for HEX dump (see manual) and RUN. My printer puts 20 HEX numbers to a line, from 00 to FF, many incorrect. If the last line does not print just press the ON LINE button to empty the printer buffer. Now run the same test after the 2 pokes have been made. Now I get a perfect 00 to FF print-out. This shows that the printer driver will correctly send all code numbers to the printer.

GRAPHICS PRINTER TRICK

Readers who have tried any extensive printer graphics programming (not a screen dump) probably got discouraged with the amount of data required to key in. Consider that a printed figure covering 1"x1" requires 60 head pin definitions per line and 9 lines are required, making 540 definitions. Remember that a space also must be defined.

Often there are repetative groups of code numbers of the same values. The following procedure will print those duplicate codes--in the DATA statement use negative numbers that denote the number of repeats and the next number is the code to be repeated. Then program the READ command to send negative numbers to a loop that prints the required number of duplicate codes, then returns to the next DATA number.

An example will illustrate this. Let's make a 1 pin line with up-turned ends 8 pins high, total length of 60 dots.

```
10 LPRINT
20 LET A$=CHR$ 27+"K"+CHR$ 60+CHR$ 0
30 LPRINT A$;
40 FOR X=1 TO 60
50 READ N: IF N<0 THEN GO TO 100
60 LPRINT CHR$ N;
70 NEXT X
90 DATA 255,-58,1,255
100 READ R: FOR J=1 TO -N
110 LPRINT CHR$ R;: NEXT J
120 LET X=X-N-1: GO TO 70
```

The reader must adapt this to the printer driver program being used and the printer code for standard density 60 dots per inch or similar. For Oliger systems do not use A$ as OUT 127,n must be used for LPRINT CHR$ n (delete line 30).



One Center Plaza
Boston, MA 02108

BOSTON COMPUTER SOCIETY
Newsletter of the BoSTUG (Sinclair/Timex User Group)
Vol 7 No 5

The Sinclair/Timex Newsletter is the official publication of the Sinclair/Timex User Group of the Boston Computer Society (BoSTUG).

Annual membership for New England residents is $35.00 (otherwise $28.00) per year. Subscription to BCS/Update and one other newsletter is included without cost.

The newsletter has an exchange policy with Sinclair/Timex User groups willing to reciprocate. Please forward exchange issues to the editor pro tem, Peter Hale, P.O. Box 8763, Boston, MA 02114.

Unless copywritten by the author, articles are in the public domain. Attribution is requested.

BoSTUG is a sub group of the Boston Computer Society. BoSTUG produces this nice 18 page newsletter, which is heavily in support of the Sinclair QL and moderately in support of the TS-2068. "BCS/UPDATE" is a large Magazine published by BCS for its members and is not to be confused with UPDATE Magazine, our name deriving from "UPDATE COMPUTER SYSTEMS", computer consultants.

Two articles from the BoSTUG Newsletter are re-printed to sample the quality and depth of coverage in this nice newsletter. ed.

REPRINT FROM: BoSTUG Sinclair/Timex Newsletter

ABOUT TOOLKIT II

Are there any QL owners who if they don't have TK2, have not been strongly advised to get it?

Anyone who has keystroked COPY mdv2_fred to mdv1_fred a zillion times to backup media finds WCOPY worth the price alone.

Certainly, the Toolkit II manual is a forbidding document. There is no tutorial; just a list of keyword extensions with limited and sometimes incomplete directions on their use.

There is not even a clue as to where to get your feet wet, so a sampling of easy and immediately useful keywords is offered here.

Toolkit II extensions that are easy and that you will enjoy are: ALARM, CLOCK, DLIST, EX, EXTRAS, EW, FREE_MEM, RENAME, SPL, SPLF, STAT, TK2_EXT, VIEW, WCOPY, WDIR, WMON, WREN, WTV, and WSTAT.

All can be explored from direct mode and you can't get in (much) trouble (except for difficulty understanding the manual).

But a lot of people are missing out on three excellent features incorporated in Toolkit II: ED, ALTKEY and last line recall.

Last line recall <ALT><ENTER> is usually useful in direct mode in SuperBASIC. It recalls all characters since the last ENTER. Done several times in sequence, it recalls several lines at a time. Once recalled, lines can be edited or reexecuted. It also works from Psion programs, as a quick and dirty copy, up to Psion's 22 character buffer limit.

ALTKEY is worth exploring. It is buried in the back of the manual in Section 21 and little is said about it. Essentially it permits writing macros. The syntax is ALTKEY 'x','....' where x is any one of the QL's character set. In the example, x is lower case.

Everything after 'x', and within the quotes happens when the combination <ALT><x> is pressed. If <ENTER> is desired for executing, the ALTKEY definition can end with ,'' (comma and two single quotes), which is <ENTER>.

How to use ALTKEY is not conceptually easy. One use is ALTKEY '1', CHR$(240)&'lflp1_setup',''. This line is incorporated in my Quill boot. The same disk also has the file 'setup_doc' on it.

Setup_doc is my style sheet for correspondence. It has my return address, the date and salutation position and no footer. I don't reorganize Quill's default page each time I write a letter; I press <ALT><1>.

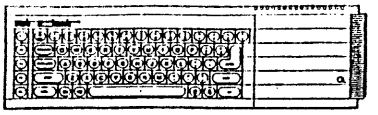
Programmers assign ALTKEYs to oft used keywords (without an ENTER) to emulate single-stroke keywords popular in earlier Sinclair's. A boot with a batch of user defined ALTKEYs and a final line NEW is run at the start of programming. NEW clears SuperBASIC memory (but not the ALTKEY definitions.)

Finally, there is ED that invokes a full screen SuperBASIC editor. The cursor keys move the cursor over the full screen. They can be combined with ALT and SHIFT for special effects. It is easier to edit existing programs with Overwriting (SHIFT F4, as in Quill).

SuperBASIC programs often have a number of duplicate or near duplicate lines. Editing a line number replicates the line at a new position and retains the old line. If slight modification is required, overwriting speeds any needed changes.

For fun, slip in ED as a line anywhere in a SuperBASIC program. Run it. Predictably, when the program reaches ED, it halts and goes into full screen edit mode. Now press <ESC>. What happens?

-Peter Hale



REPRINT FROM: BoSTUG Sinclair/Timex Newsletter

QUILL FILESAVER

More than once, my QL has mangled a long Quill document through poor disk or microdrive I/O. Either I removed the cartridge before closing a file, or the QL tried to write part of the file to a non-existent track on the disk or the QL crashed. The result is the same - any attempt to reload the file into Quill results in a **File I/O Incomplete** error, disasterous if you forgot to periodically backup your work.

One such encounter forced me to take a deeper look at how Quill files are stored. I could do this with "BGET", which reads a single byte from a file. Most disk systems and toolkits have this command.

I used BGET to print out the numeric value of every character in my partially-corrupted Quill file to learn its structure and so recover at least part of it.

The most striking discovery was that Quill files use the Null character [CHR$(0)] in place of Carriage Returns. I also noticed that a good portion of the Quill file is formatting information, stored towards the end of the file. It is mostly strange control characters. All the text was present, but some formatting information was cut off during the incomplete SAVE.

I tried renaming the file with a _LIS extension and importing that, but Quill didn't like the NUL's and stopped importing the file at the first occurance of a NUL.

So I had to figure out a way to extract the 'text' portion of my document and re-write it as straight ASCII. BPUT came to the rescue and the result is the listing below. In a nutshell, the program reads a byte from the damaged file in channel #5. If it is a null, it's changed to a carriage return [CHR$(13)]; if a control character (less than SPACE [CHR$(32)] but not CR), it is changed to a space. Otherwise, it is a regular ASCII character and left as is.

The now ASCII character is written to the new file in channel#6. The character is also printed to the screen so you can watch as it goes through the file. Then the loop repeats for the next character until there are no more characters to read. What is left is a plain text file which can be imported into Quill. I was able to recover a 15 page document this way, though it took about ten minutes to do so.

Hopefully you will never need to use this utility, but if you do, here it is.

- Jim Rodlin

```
 10 REMark       "Filesaver"
 20 REMark Corrupted Quill file
 30 REMark    recovery utility
 40 REMark       by Jim Rodlin
 50 REMark    Requires Toolkit
 60 REMark commands BGET & BPUT
 80 REMark In the public domain
 90 REMark Attribution requested
100 PLP_EXT: REMark or TK2_EXT
    enable extended command set
110 CLS
120 OPEN#5, "mdv2_MYFILE_doc"
    : REMark file to save
130 OPEN_NEW#6, "mdv1_BBS_lis"
    : REMark output ASCII file
140 REPeat loop
150  BGET#5, a%: REMark get byte
     and put in a%
160  IF a% =0 THEN LET a% =13
     REMark convert null to CR
165  REMark replace all control
     chars with SPACE
170  IF a% <32 AND a% <>13 THEN
     LET a% =32
180  REMark print character to
     screen for visual feedback
190  IF a% =13 THEN PRINT
     : REMark Carriage Return
200  IF a% <>13 THEN PRINT
     CHR$(a%);
210  BPUT#6, a%: REMark write
     ASCII byte to _LIS file
220 END REPeat loop
```

THE QL BEGINNER COURSE

By: Bill Jones

Here we go! Being a beginner myself with the QL, I am a bit intimidated by the thought of writing an instructional series. I hope that those of you who are better qualified will Koreckt my mistooks, and broaden upon the subjects covered. Then the subscribers will benefit from your inputs. So dont think that I will be offended by feed back that tells me when I'm wrong and gives more detail to the coverage. We want a interchange of programming information. IF there are 8000 QL users, THEN 4000 KNOW SOMETHING that I dont. THEN IF half contribute, it is sure that at least 4000 will gain more knowledge. Thats what its all about.

This will be a continuing feature article. It will grow in complexity as the scribe's proficiency grows. Right now, the scribe has about 30-40 hours experience operating the QL, so it isn't likely that you'll find anything that is "above your head" in these first few episodes. In fact I am strevin and strovin to make the QLWP program zap this stuffin onto the paper. WHICH makes a good starting point. Lets get started now covering--

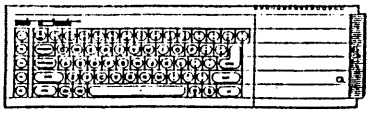
HOW TO DO IT WITH QLWP

IF you have dug around as I have between the four sections in the back of the QL manual, you know that these books were not written in a 1, 2,3, sequence that lets you get right to it. I believe that, for this discussion, we should start with the printer itself. If you have a SERIAL printer, just connect the cable directly to the SER 1 or SER 2 port at the back of the QL. Lets use the SER 1 port. But if you have a PARALLEL printer you will need one of the Parallel interfaces connected between the printer and the QL SER 1 port. I was most successful by setting the printer to its start up initialization set up, ie, no margins set, no LINE FEED, no auto PAGE end, and ELITE 96 character lines, and 66 lines per page.. These are "dip switch" settings on the printer itself. Later we can come back to the printer switch setting if we need to. Generally, most users do not know enough about their printer, and there is a need to study the printer manual to know how to set it up for the best printing.

Now that we "think" that the printer is set up right, lets turn it off. Turn ON the QL and keep Micro drive #1 VACANT until the F1 Key has given you the blinking cursor. Then INSERT a blank MDV cart in either of the slots and FORMAT it. Now that may be a problem for the first time user. Most of the MDV carts in use are the ones that have had the "write enable" tab removed. If you are accustomed to using disk drives, the MDV cart is "write protected" by REMOVING the tab that covers the little triangular notch in the right hand edge of the cart "as you are looking down on the label". In order to FORMAT the MDV you need to cut a thin piece of tape to stick over the notch. I find that the best tape to use is a piece of Disk Write Protect tab, cut across with scissors to be just slightly thinner than the width of the shoulder of the MDV cartridge. This way, the tab is not rubbed off when it is inserted in the micro drive. Once the "write enable tab" is installed, the mdv cart will FORMAT, with the command: FORMAT mdv1_buwp Bu indicates that it will be a Back Up cartridge for QLWP. Format the mdv cart TWICE, using the same procedure. From what I'm told the mdv cart will be more reliable if it is formatted twice.

Next, we want to make a back up copy of the Word Processor cartridge. Do that by installing the original cart in mdv1 and the one just formatted in mdv2. Type dir mdv1_ and touch ENTER. The directory of the master cart in mdv1 will appear on screen. Then, using the directory as a guide, copy each file listed in the directory, using: COPY mdv1_name TO MDV2_ After all files have been copied stash away your master Word Processor cartridge for safe keeping.

Now we are ready to go to work. Now install your back up cart you just made into the MDV1 slot. Next we will setup the software for your printer.



Taking a little break from the details, the first page did print, so I'm becomming an expert fast! Well, not quite, as I really didn't want that page number and I dont know how to kill it. Anyway, for the most part it does prove that I've been the route that I'm about to tell you about, which is--

HOW TO CONFIGURE FOR PRINTER

Now what we're gonna do is to CHANGE part of the QLWP software to make the word processor print with the type of printer that you use. Dont worry about making a mistake, as nothing you do will hurt the software, and besides, you are working with a back up copy. Your equipment status should be: NO micro drive cart installed-- A RESET QL, F1 touched to give the blinking cursor-- PRINTER OFF. When you have that status, then: INSTALL the QLWP back up cart in drive 1. Type; lrun mdv1_install_bas ENTER. Wait until the first menu comes on screen, and THEN: follow the prompts to select the PRINTER TYPE. Usually this is all that is required. But I had to be different! I use a DAISY printer that has the Diablo compatable command set. So, after quiting this install_bas program, the printer would kick up 6 lines before starting to print, print 3/4 of a page and then print the page number. Now tell me WHO wants that kind of a page of print? SO, back to <lrun mdv1_install_bas> again- and again- and again. Next day, I called ole Bill Cable and got straightened out. Now this mought help you too: "READ THE DURN QUILL SECTION"- all of it! The durn install_bas program was set up right after all. The problem was in my not understanding how to trapsy about in the WP menus to DESIGN the page. So, what I'm telling you is that the procedure of setting up for your printer should work out OK. When finished, what you have done is to bring up YOUR PRINTER's command set as the one which will boot in when you use QLWP and select PRINT at the word processor menu. After you select your printer type, the mdv runs and re-records your printer set up so that it boots in for printing. Later, you can go back and lrun the install program again to set up for a different printer. Now that we have that done, lets create a document with the word processor program---

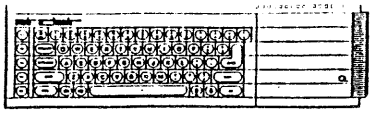
WORKING WITH QLWP

Insert you back up copy of QLWP in mdv1, and RESET. When the QL has reset, F1 starts the program loading and you have time to pour a cupa while waiting. Now there isn't any point in draging this out, as the WP works very well with its menus to guide you. The important part is to use F3 to get the COMMAND menu and set up the page with D-Design and M-Margins. Likely you wont need to use H-Header, as this is a repeating header for every page. To do a single header, I find it best to type the header and back up with the arrow key to its start, then pad in spaces to center the header. Then run the cursor to the end of the header and use ENTER to shift to the next line. When ready to print, F3 again gets the COMMAND MENU, where P-PRINT will start the printing sequence. Now we will move on to another way to use the QLWP program for doing controlled LISTings of programs.--

USE QLWP for LLISTings

Some folks never need a paper listing of software. But others need to do that quite often. One can get a LLIST in the direct mode (cursor flashing) by: <OPEN #4,serlc:BAUD 9600:LIST>. This makes LIST go to the printer. BUT- The listing splatters across the whole page, 96 characters wide if thats what the printer is set at. Most printers wont let you confine print with printer settings to less than 64 characters per line. I particularily need to LLIST at controllable line width for printing listings in UPDATE. So, back to the phone for answers. Ole Mark, at Sharps Inc. does more than just sell QL stuff. He's a programmer too, and he told me how to do it with the QLWP software. Here it tis---

Load the QLWP program with F1. Then use F3 to get the Command menu. Punch O-Other, then I-Import. Next, the prompt appears to key in a MDV file to load. Input <mdv1_name> and the next prompt asks for "by line?". Touch ENTER at this point and the program lines load into the Word Processor as a ASCII file.



After a short while the entire program listing will load into the Word Processor and the program lines will appear almost like you would see them in the Basic mode. But the listing is TEXT, and can be formatted as text. Hit F3 for the Command menu, then D-Design, then M-Margins, and set the margins for whatever line length you want to print the program listing. Then F3 again to P-PRINT. Thats all there is to it! AND alla time I was digging through the manual trying to find a RAM address to POKE to get shorter lines in a LLIST. This is a better way after all..

By the time the next issue rolls in maybe I will have learned a bit more to present in this QL BEGINNER COURSE. If you have things that you are bothered about with the QL, but are ashamed to ask, this is the forum for discussing them. I'll even let you remain anonymous. See you next issue!

A QL PRINTING PRIMER

Here is a little tutorial for setting up the QL to print to paper "without a word processor". It does not do much but print un-formatted lines to the left margins as set at the printer. But, it will demonstrate how to make a printer print with simple programming. You can key the lines in and operate it with either RUN or GO TO 5. Line 10 gets the INPUT typing routine at line 200. Then when finished, ENTER executes the RETurn at line 220. Then line 20-28 presents a menu of options. When <1> PRINT is elected, the typing data is printed, and the program repeats to get the next typing burst.

The tutorial about "output to the printer" begins at line 50, which executes when <1> PRINT is elected. <OPEN #4,serlc> OPENS the CHANNEL to the printer "through Serial Port 1". Then the "baud rate" is set to 9600. Line 60 then uses <PRINT #4,a$> to print the data. <print "#4"> directs the print to the "channel that was opened". Look at the lines 70 and 80 which executes the other two print options, for "line space" and "LIST to Paper".

When <5> QUIT is elected, line 300 executes, and is important to note. When a CHANNEL has been OPENED, Channel #4 in this case, the channel should be CLOSED when finished. There are many CHANNELS to route the computer's tasks though. Each channel should be OPEN only for the task to be accomplished, and if the channel is not CLOSED afterward then the next task to be accomplished could be sent to the wrong OPEN channel.

The Print Tutor Listing

```
 5 CLS
10 GO SUB 200
20 PRINT "Options:"\\"<1> Print the data"\
22 PRINT "<2> Line Space"\
24 PRINT "<3> INPUT New Data"\
26 PRINT "<4> LLIST"\
28 PRINT "<5> QUIT"\
30 INPUT a:CLS
40 IF a=3 THEN GO TO 10
50 OPEN #4,serlc:BAUD 9600
60 IF a=1 THEN PRINT #4,a$
70 IF a=2 THEN PRINT #4,\
80 IF a=4 THEN LIST #4
90 IF a=5 THEN GO TO 300
100 CLS:GO TO 20
200 PRINT "Type the Data"\\
210 INPUT a$
220 CLS:RETURN
300 CLOSE #4:STOP
```

The programming at left could be condensed a lot by chaining the different programming commands using a colon between. For example, all of the lines from line 5 through 30 could be inclosed in one longer line of programming. I used extra lines in order to make the tutorial easier to decipher.

Also, the IF THEN conditionals used are in "simplese" basic syntax which the QL reacts to nicely. You can use almost all of the simple Basic syntax of the older computers with the QL, which makes it a very easy computer to adapt to. We will ease into QL Basic and Super Basic as this series continues.

SINC - LINK

Vol. 6 No. 4 JULY - AUGUST '88

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

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

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

HIGHLY RECOMMENDED!

The Toronto, CANADA Sinclair User Club is one of the most active clubs in North America. Two of UPDATE's best writers are mainstays of the Toronto Group, Bob Mitchell and George Chambers. SINC LINK, the club newsletter, is an excellent source of information. The Toronto User Group has many USA members and welcome more.

ONTARIO QL USERS TAKE NOTE! Join the Club and form a special interest group within this Sinclair Special interest Club! Club membership and participation is the best route to take for continuing to learn more about your computer. -ed.

TORONTO TIMEX - SINCLAIR USERS CLUB

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

Canada

Another Fine Club

Dear Bill:

I am sorry for the delay. The heat is getting to me and I have been lax in my duties. Enclosed is a little discription of our group as you asked for.

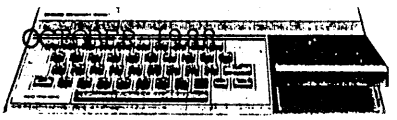
Our group is called Sinclair Milwaukee Users Group also called SMUG. He were established in 1983 and work with all forms of the Sinclair computers and their clones. He have ZX80's, ZX81's, TS1500's, TS2068's, Spectrum's and Spectrum emulater's, and the QL. He are in contact with 22 other Users Groups around the U.S. and Canada and swap newsletters with them.

He meet once a month on the second Hednesday of the month, our next meeting is July 13, 1988. The place we meet is in Elm Grove Wisconson, a suburb of Milwaukee, and called Verbrics' Inn. He chose them because of the hall size and centralized location.

Our membership fee is $15 for the individual or $20 for a family. This entitles you to library privileges, you must check them out in person, our newsletter, you can pick this up at the meeting our supply us with SASE's and I will mail them to you. The newsletter has a subscription rate of $10 per year, this includes mailing.

Hope to see you at a meeting.
Bill Heberlein, VP SMUG,
5052 N. 91st Street,
Milwaukee, WI 53225

Sinclair Milwaukee. User Group - A Live Bunch!



MSCRIPT VERSION 5.3 FOR LARKEN DOS
SOFTWARE REVIEW by Bob Mitchell
880814

My long-awaited copy of Version 5.3 of MSCRIPT by Jack Dohany finally arrived early in June of this year. Because I had told him that I was using a non-standard, earlier version of the Larken DOS, he sent the program on cassette, to be on the safe side. I was delighted to find that a back-up on disk was readily made by selecting DISK on the main menu and then pressing B.

Documentation was on paper and the only other program on the cassette was the LCODES text which contains all the various codes that a wide printer would recognize. I understand that the disk version contains all the documentation as files.

The main problem that I had had with my earlier version MS5T was one of the printer dropping characters every now and then. This problem has been resolved, I am happy to report.

The cost to me for the new version was US$25 which included $5 for mailing and handling. The program is so much better than the earlier version that I was happy to pay the price.

Here are some features of MSCRIPT (not necessarily in order of importance) that make the program quite attractive and make me a convert:

1. Finding and Changing any string of up to 30 characters is simple and very fast.

2. Headers and Footers (titles at the top and bottom of each page of a document) are easy to add.

3. Moving, copying, deleting, saving and loading blocks of text is fast, easy and very professional, with markers used to identify the block to be worked on.

4. The cursor moves very fast across the page. There is also a tab feature that is very useful when developing tables and lists.

5. When making a back-up copy, it is optional to include the text as part of the back-up: ideal for making a version that includes, say, a letter head for page 1. If you want to make a LARKEN AUTOSTART copy, this can be done by using <CLEAR 45000:RANDOMIZE USR 102:RUN>. You have to CLEAR a higher address if you include any text in your save.

6. View Memory allows you to PEEK and POKE the contents of memory; this starts at the LCODE file which is handy if you want to inspect and change any printer control codes.

7. The name of the text you are working on appears on the home menu. I find this to be very useful particularly as I hop from text to text fairly often. It let's me know where I am at the moment. The GIVE NAME command allows you to assign or change the name of your text at any time.

8. There is an alternative fattened character set that makes reading the text on my monitor much easier. The use of different screen colours is restricted by the TS2068 hardware and I have settled on white ink on black paper as the best for me.

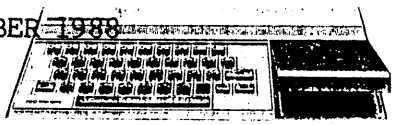
9. The disk catalogue in the selected drive can be added to the text in progress. The drive is selected by entering a number from 0 to 3 (ie, 1 to 4 respectively). The LARKEN Ramdisk is selected with <4> and this can be accommodated easily by changing lines 131 and 132 in the BASIC with no addition to the BASIC length.

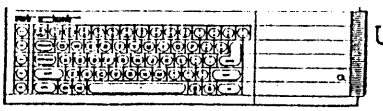
10. The new LARKEN Version 3 EPROM uses the token MOVE to call a file rename routine and this was easily added to the home menu by a few lines at line 220 in the BASIC. Watch out for an out of memory report here. I had to sacrifice lines 420 to 440 (VERIFY, which I seldom use anyway) to make room for this.

11. I like being able to put comment lines in the text to serve as reminders to me and that won't appear in the printout. Tab settings are one example as you have to reset these whenever they default to their normal settings.

One thing that could be better: It's too easy to stutter in more spaces than wanted when entering text. However, on right justification, these extra spaces are not really noticed.

MSCRIPT is a totally different word processor from TASWORD which I have used for some time and which has its devotees. But there are so many advantages to MSCRIPT that I have begun to switch over to it for my word processing tasks. Once you get used to the protocol and press the right keys, MSCRIPT is just as easy to use as TASWORD and a lot more rewarding. I recommend the purchase of MSCRIPT Version 5.3 and look forward to experimenting with Version 6 when it comes along. Hurry up, Jack!





REPRINT FROM SYNC LINK

2068

LARKEN RAMdisk for the TS 2068
by George Chambers

Our last newsletter carried an article by Greg Lloyd, describing a RAMdisk unit that he had recently obtained from Larken Electronics. I have recently purchased one of these units, and should like to tell something of my programming experiences with it.

Firstly, when I got it I could not figure out how to use it effectively. After all, while it can contain a total of 256K memory, mine only came equipped with 64K and there really is a limit as to how many program one can hold with that. Which programs should I place on it, etc?

On top of that I already had a second drive on my system that seemed to be only used when I wanted to copy a program. Of what use was another bit of expensive memory. And it is expensive, if compared to the memory capacity of a disk. A disk will hold 360K and costs about 65 cents. A 32K SRAM chip costs about $15. And the RAMdisk is really a solid-state drive, not a memory expansion to your computer. That is to say, you still have only 38K of memory in your computer to play with. The RAMdisk memory is not bankswitched in to the computer; it it loaded into the computer memory the same as from a conventional drive.

However, despite this unpromising prospect I have found a very fascinating application. What I have done is to place 58 of my most useful utility files on a single disk (I call it my OMNIBUS disk), and installed it in my second drive.

I then placed the several menus (the menu program listing in the Nov/Dec '87 issue of this newsletter) required to access these files into the RAMdisk. In the beginning I had one AUTOSTART (menu) program, and several other menu programs i.e. menu.81, menu.82, etc.,that I called up from that AUTOSTART menu. I found that presently I had used 7 of the 12 available tracks on my RAMdisk; horribly extravagant. (The RAMdisk uses the same tracks format of the other drives)

It was then suggested to me that I combine several menus into one program. This made sense, because as it was I was storing a program of about 1500 bytes onto a 5090-byte track. The other bytes, 3410 of them, were lost. I started combining menus onto a single program, and now I have 5 menu screens combined. More will follow. I use a menu option to move to the various menus.

Most of the programs themselves are on the OMNIBUS disk, installed more or less permanently in the second drive, and are called up from the menus obtained from the RAMdisk. Where a program is called up that resides on other than the OMNIBUS disk, the menu displays a request onscreen for that disk to be placed into the first drive, PAUSE's until you indicate 'ready' by a keypress, then loads from that drive.

Larry Kenny has revised his DOS EPROM, giving a new command which has proved very useful in this application. The format of this command is PRINT USR 100: NEW (or PRINT #4: NEW). This provides a 'warm' AUTOSTART capability to the DOS.

I use it this way. I have modified many of my programs to provide an exit from the program. The exit normally would take you to a STOP command. To use this new DOS command I terminate a program to the following line sequence.

```
LINE 9000   PRINT #4: GOTO 4: PRINT #4: NEW
```

The 'GOTO 4' points the DOS toward the RAMdisk and the 'NEW' initiates an AUTOSTART routine in the RAMdisk to bring up the 'menu' program. The effect is to bring up the OMNIBUS disk menu at any time from within a program with a single keystroke. Note that this new command is for all drives; it's use is not confined to the RAMdisk.

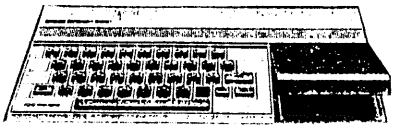
This new DOS command also offers another interesting programming possibility. Invoking the 'PRINT #4: NEW' command will bring up an AUTOSTART program in the drive to which it is pointed. It does this without shutting down the computer. Thus, any M/C program presently in the computer (so long as it is not overwritten by the AUTOSTART program) remains in place awaiting use.

Larry advises that this command is in Version 2 of his DOS EPROM. Before you all rush out and ask him for it, I suggest that you wait until he has had a chance to incorporate some other features into the DOS, that he speaks of doing. Improvements to the printer driver, and changes to the NMI routine, I have heard; among others.

Now, I could hardly recommend that everyone go out buy one of these RAMdisks. But what I have found is that for those persons with two drives it certianly maximises the use of the second drive. And it most assuredly reduces the number of keystrokes required to operate the Larken system, plus reducing the number of drive startups by possibly a half. And of course it is silent and speedy, when compared to a regular drive!

****

Newsletters need news!
Write and tell us what
you know!
Anything (almost!)
will be printed



REPRINT FROM SYNC LINK TUTORIAL (TS2068 & SPECTRUM)

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

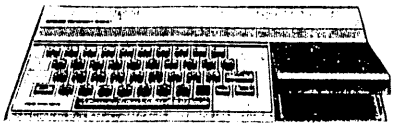
Let's delve into the mysterious realm of channels and streams and those two odd tokens OPEN # and CLOSE #. These two commands are extremely powerful and have many uses and yet there is nothing at all about them in the TS2068 User Manual except to say that "these and other commands like FORMAT, MOVE,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

The TS2068 communicates with peripheral devices with the aid of the channel information area which lies between the system variables and the BASIC program. This area starts at address 26600 and ends at 26709, just short of the program area which starts, normally, at 26710. A short program given in the listing below will print out all the information on channels and streams. Part of its output is shown below as Table 1 (with some explanatory notes); the code has been divided into 5-byte blocks, each block representing one channel.

TABLE 1 - CHANNEL INFORMATION AREA

Address	Chan	Value	Notes
26600	K	0	1st 2 bytes = ROM add.1280d
26601		5	(5*256+0):ROM PRINT routine.
26602		14	2nd 2 bytes = ROM add.3086d
26603		12	(12*256+14): get input key.
26604		75	Code for letter "K".
26605	S	0	1st 2 bytes same as for "K".
26606		5	
26607		191	2nd 2 bytes = add. 4543;
26608		17	input routine called.
26609		83	Code for letter "S".
26610	R	231	This block is reserved for
26611		10	use by the computer and
26612		191	is not available to the
26613		17	user.
26614		82	Code for letter "R".
26615	P	0	1st 2 bytes also point to
26616		5	1280d: ROM Print routine.

(c)1988 Bob Mitchell



TUTORIAL (TS2068 & SPECTRUM)

REPRINT FROM SYNC LINK

26705	191	Input routine called;
26706	17	(see explanation below).
26707	80	Code for letter "P".
26708	128	Stop byte.

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

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

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

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

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

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

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

(c)1988 Bob Mitchell

REPRINT FROM SYNC LINK

TUTORIAL (TS2068 & SPECTRUM)

being altered and the latter identifies the single character of the channel to which the stream is to be directed.

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

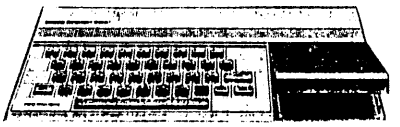
At switch on, only four of the 16 streams are opened.
Stream 0 points to channel "K"
Stream 1 points to channel "K"
Stream 2 points to channel "S"
Stream 3 points to channel "P"

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

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

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

```
  3 REM This program works in
    either TS2068 or Spectrum
  5 REM ====================
  6 REM Use GO TO 10 for info
on channels & streams.
  7 REM Use GO TO 200 to PEEK
```

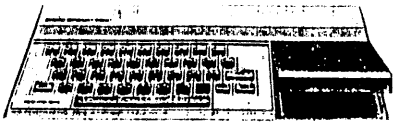



```
  stream info in system
  variables. Use GO TO 220 to
  PEEK channel info area up to
  start of program area.
   9 REM ====================
  10 LET a=PEEK 23631+256*PEEK 2   :start of CHANS
3632: LET b=a                       from sys variables
  20 IF PEEK a=128 THEN GO TO 70   :last byte in table
  30 PRINT "CHANNEL ";CHR$ (PEEK   :channel letter
 (a+4))
  40 PRINT "Output Address=";PEE   :output address
K a+256*PEEK (a+1)
  50 PRINT "Input Address=";PEEK   :input address.
 (a+2)+256*PEEK (a+3)
  60 PRINT : LET a=a+5: GO TO 20
  70 FOR a=23574 TO 23605 STEP 2
  80 PRINT "Stream ";(a-23574)/2   :stream number.
;TAB 10;
  90 IF PEEK a+256*PEEK (a+1)=0    :closed channel
THEN PRINT "CLOSED": NEXT a: STO
P
 100 LET c=(PEEK a+256*PEEK (a+1   :channel letter
))+b+3
 110 PRINT "points to Channel ";
CHR$ PEEK c: NEXT a
 120 STOP
 130 RANDOMIZE USR 100: SAVE "ch   :Larken SAVE routine
ans.B1" LINE 10
 140 STOP
 200 FOR i=23568 TO 23605: PRINT   :addresses of
 i,PEEK i: NEXT i                   channels attached
 210 STOP                           to streams.
 220 LET prog=PEEK 23635+256*PEE   :start of prog area
K 23636
 230 FOR i=b TO prog-1: PRINT i,    :examines chan
PEEK i;TAB 20;CHR$ PEEK i AND PE    info area up to
EK i>=32 AND PEEK i<=90: NEXT i     start of prog area
```

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

For Larken users, note the extensive use of OPEN # in the extended BASIC commands available and particularly note that when any or all windows or large printer channels are opened, 38 bytes (total) of memory are

(c)1988 Bob Mitchell



REPRINT FROM SYNC LINK TUTORIAL (TS2068 & SPECTRUM)

added to the channel information area. This changes the location of the start of the program area and can cause a problem if there is machine code in a REM statement in the first program line. (This problem can be avoided by PEEKING the start of the prog area and then using this value plus 5 as the call address for the machine code routine in the REM statement; eg, <LET x=PEEK 23635+256*PEEK 23636: LET x=x+5: RANDOMIZE USR x>). In these 38 bytes which you can examine by using line 220 of the listing, you will find that new channels have been set up at the beginning of what was the start of the program area.

An interesting and illuminating exercise is suggested for Larken users:

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

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

(c)1988 Bob Mitchell
Page # 5 of 5


```
*************************************************************
*                                                           *
*          QQQQQQQQ     ZZZZZZZZZ   XX      XX            *
*         QQQQQQQQQQ    ZZZZZZZZ     XX    XX             *
*         QQ      QQ         ZZ       XX XX               *
*         QQ      QQ        ZZ         XXX                *
*         QQ      QQ       ZZ          XXX                *
*         QQ    QQ QQ     ZZ           XX XX              *
*         QQQQQQQQQQ     ZZZZZZZZ     XX   XX             *
*          QQQQQQQQ     ZZZZZZZZZ   XX      XX            *
*                QQ                                         *
*                                                           *
*************************************************************
```

The Journal Covering Amateur Radio & Sinclair Computers
ZX-80; Micro-Ace; ZX81; T/S 1000, 1500, & 2068; QL
Alex. F. Burr, K5XY, Publisher
2025 O'Donnell Drive
Las Cruces, New Mexico
88001

SUBSCRIPTION $15 PER YEAR. SAMPLE ISSUE FOR SELF ADDRESSED STAMPED ENVELOPE WITH $0.45 POSTAGE.

TELECOMPUTING, WHERE WE STAND

By Bill Jones WD4FJI

Back in 1948 Television came into the homes of America. Almost all people thought of TV as an adjunct of RADIO. Radio was THE INDUSTRY, and TV was thought of as a part of it. Of course, TV became its own industry. This bit of history is used to make the point that TELECOMPUTING is destined to become a giant industry of its own. Another illustration: If one looks down upon a ant bed, one sees much activity, but all seem to be going in different unorganized directions. Telecomputing is kinda like that today. Telecomuting is now riding piggy back on two existing industries, Computers and Telephone. But a break-out is about to happen. Amateur Radio Operators are leading the way. Most think of "HAMS" as guys pecking away sending and receiving "di dah dits" and making not so funny interfering lines on TV screens.

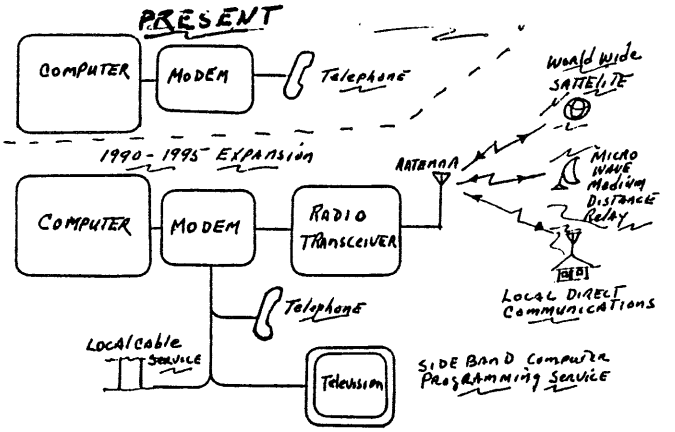
But many HAMS now are operating computers and are involved in a form of Telecomputing called PACKET. The principle involved in Packet is the use of Radio and Computers to Communicate. Most of us are aware of Modem Communications with computers via telephone. I became acutely aware when I received a shocking telephone bill a month after getting my modem. PACKET bypasses the telephone, and provides direct computer link up via radio.

NO TELEPHONE BILLS!! But, as with any fledging industry, there are problems, and there are "turf fights". A&T wants the business and would like to confine Telecomputing to the use of their telephone network. As more computers with modems come on line, local telephone organizations want to develop means of charging for service. The HAMS have a good thing going with Packet, but the Federal Communications Commission regulates Amateur Radio and confines their activities to "Hobby Use", no commerical operation allowed. Hams can operate with Packet, but ordinary folks who are not licensed radio operators cannot operate Packet. It's against the law.

During the latter five years of the 1980's, the Computer population skyrocketed from less than 1% of homes having computers to more than 40% by 1988, and will be up to 80% before 1990. The natural progression is for families to take advantage of the computer's inherent ability to communicate with the surrounding World. To accomplish this step there must be a breaking out and away from the confines of the telephone industry and the Federal regulation of communications, which is now the most limiting factor. Of course, there must be an organized way to accommodate the tens of millions of people who will be using computers to communicate. Otherwise there would be chaos, everyone trying to communicate, but interfering with each other. This is the next step that must take place between Government and industry planners.

I would invision the decade of the 1990s bringing in a coordinated "starter plan" with Federal regulations licensing telecomputing via satellite, similiar to Packet, plus UHF repeater nets, and local cable service. As the industry builds, and commerical applications expand (profit potential), we will see the Television industry wanting to get into the act, and TV will become a valuable partner to Telecomputing. Each TV channel can accommodate quite a number of discrete side band channels for computer communications without interfering with their regular video programming. One could just subscribe to the service and plug their computer modem into the TV set to receive a daily schedule of services.

In the meantime, awaiting the break through, and the concerted organization needed to thrust telecomputing into tens of millions of homes, a few intrepreneur HAMS are taking the forefront and developing Packet. Bob Diggs is one of these leaders. His letter and article that follow reveal some technical problems being experienced and the level of current telecomputing technology with the Sinclair QL, and all computers as well.. Read between the lines to discern Bob's personal dedication to the advancement of telecomputing technology. The following sketch depicts where the industry stands at present and the paths for expansion during the next 5-10 years.



Bob Diggs, KV3Q
North East, Maryland
August 22, 1988

To: Bill Jones, WD4FJI
Alex Burr, K5XY

Subject: QL on Packet

This memo is addressed to both of you so that you both will know what is going on here and can help me get some information to those HAMS who may be interested. I started a draft of an article for QZX which somehow got lost in a micro drive with a message saying, "file incomplete". I had made a print-out but didn't feel up to re-doing it because of hot weather and too many other things to do. Also, among the three of us, I dont think that Packet is all it's supposed to be. There is too much QRM (interference, ed) from re-trys of hams trying to download a bulletin board list of messages or nodes that it becomes too difficult to connect for a QLO (discrete two way interchange, ed) or to get a gang together for an unproto bull session. Never the less, I do use my QL for Packet, using MIKE_TERM almost all of the time. I am, however getting ahead of myself.

Alex, what started this memo was a note from Bill saying that he saw my interest in QL Packet in QZX and wanted to publish the details in UPDATE, a magazine that I am not familiar with because it has been a publication for the TS-2068, which I dont have. My Ham interest says to write for QZX, but my public spirit says to write for anyone who is interested because there are so few out there on QL Packet.

First, do I have anything new or worth publishing? Enclosed is a printout of what I lost in the file. Writing this stuff is new to me so I'd like to know if I'm on the right track. Second, I dont know the format to present this material for either publication. I belong to C.A.T.S. and each newsletter clearly states how articles should be prepared for publication.

Third, I hope that you can both use it, rather than have it go to one and not reach some guy or gal who could use it. I dont know much about computers and programs and such, but if I satrt the ball rolling, someone who knows more can make things better. So Bill, this is my answer to your card. I hope that it helps. Let me know if you are interested in what I started and I'll try to put it together.

Best regards,
Bob Diggs, KV3Q

Comment by the editor; Bob's address is left off to protect his privacy. Letters to him mailed to UPDATE will be forwarded. QZX is a monthly newsletter for Ham Radio operators. QZX, 2025 O'Donnell Drive, Las Cruces, NM 88001. Subscription is $15.00 per year. Those of you who are interested in more about Telecomputing and Packet are requested to send in an interest card. The amount of response will dictate the amount of future coverage. Bob's article follows.

PUTTING THE QL TO PACKET
Bob, KV3Q

As Alex noted in the June issue of QZX, I'm trying and can now say that I'm using my QL on Packet. It wasn't easy but I finally made it. A monitor was one mighty important item, in fact trying to do it without a monitor was a failure. My Packet activity is strictly 2 meters, using a MFJ-1270B TNC connected to SER 2. I have two programs on micro drive cartridge, QL TERM by Richard Moldovan, and MIKE_TERM from the April issue of QZX. Each has a special use. Each has different problems and neither is completely satisfactory. However I am using my QL on Packet and it is there whenever I want it. SER 1 has my Citizen 120D printer, and SER 2 is Packet.

My old set up was either the TS-1000 or 1500, ALPHACOM 32 and a COMLINK II connected to the TNC. That was a very satisfactory set up indeed, which allowed me to dump the screens to the printer. This is something that I cannot do with the QL.

Before leaving for South Africa I ordered a QL to be shipped to our Winter home in Florida and when we got back in January it was waiting. I went to the Winter Fest in Orlando and found that there was no one there who had any idea of what I was talking about when I brought up Packet. But I did learn one important thing, and that is that the QLUG SER2 pin connections are not as printed. That turned out to be academic, because the standard SER cable that I bought there works OK when connected between SER2 and the MFJ TNC. It was suggested that QL-TERM was a good program for a Modem and possibly might work with Packet, so I ordered it.

QL-TERM arrived soon after and what a disiappointment when I tried to use it! I ask the dealer if my problem could be because I was using a TV as a monitor. Of course, he couldn't answer that question. So, Packet went to the back burner while I concentrated upon finding a printer that I liked at a price that I could afford. In the meantime I was wondering if anyone was getting anywhere with using their QL



on Packet. The US Postal service did not forward QZX or CIM, so there I was in a black hole, so to speak, and no way out. Then lo and behold, Michael Mitchell came through with a program in April QZX. When I put the program on Micro Cartridge I learned that it needed a Monitor. By this time I had determined that a monitor was needed for general use anyway.

When I got the monitor on line, the first program to try was QL-TERM. It worked but there seemed to be a problem between the computer and the TNC. They couldn't seem to communicate without messing up some of the characters. The radio communications over Packet seemed to go out and come back OK despite the garbage on screen. So, I tried the MIKE-TERM program only to see single letters scrolling up the left of the screen. I searched for the problem but didn't find it until much later.

When using the TS-1000, the COMLINK-II required a baud rate of 300, so I stuck with that until I discovered that by switching the TNC switch to 1200 baud the garbage on screen disiappeared.

At last I had a program working, but couldn't back space or erase in the normal way. The solution is to use CTRL H instead of the normal DELETE. Yet to be resolved is how to duplicate the screen to printer as I used to do, but that will come later, I hope. Another bother is that the CTRL S takes a while to stop the scroll, longer than the TS-1000 and most likely because of the baud rate.

Then it was back to MIKE_TERM. A intense study of the program listing revealed a missing ";" in line 270 at the end of the line. That did it, but backspace reared its head again. A partial solution is to use CTRL H again, but the lack of cursor means that you are typing without reference. Now if I make an error, I use CTRL X, which shows a "/" and scrolls to the next line. The advantage of MIKE_TERM is that it loads fast. The disadvantges are, no screen dump to the printer and no flashing cursor. I then changed the window to have a program for TV use, and that works fine. Neither programs have macro keys.

So, that is where I stand with using QL on Packet. It is working fairly well but there is still a long way to go. To summarize;

1. Use a Monitor or MIKE_TERM modified for TV Screen.

2. Ser2 port pins are: 1=ground, 2=Tx D, 3=Rx D, 4=DTR, 5=CTS, 6=GRND, 7=GRND, 8=blank, 9=T/2V

3. QL_TERM: Needs Monitor. 1200 baud. Must anticipate CTRL S or lose some screen. No screen dump. Erase and backspace with CTRL L. Can't go to QL programs when you Quit.

4. MIKE_TERM; Easy to load and use, but lacks cursor and screen dump. Use CTRL H to erase, but no cursor makes it hard to use. Use CTRL X to cancel line and start over again with new line. Be sure line 270 has a ";" at line end. Modify window if using TV. Can quit and load QL programs and vice versa. A good program. I use it daily to check my BBS.

I have used two other programs, QCODE and QLINK, without success. The copies that I am using could be at fault. I will try to learn more about them and report when I make significant progress with these or the other programs. Bob, K3VQ———

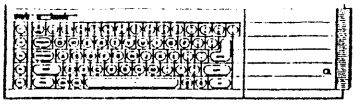
Ed, comments: Screen Dump to printer is supported by the special language contained in the PISON ARCHIVE software. This may be an avenue to pursue to link with the modem softwares. Cross talk between those using the QL within the telephone modem environment and Packet users should be fruitful. UPDATE will open up a channel for such cross talk in the pages, and will publish the names, addresses, and phone listings of those who have keen interest in Telcomputing, either via telephone or Packet, and who would like to establish direct communications with each other. Hopefully, we can have something useful in every issue of UPDATE about Telcomputing with the QL, the TS-2068, Via Telephone and Bulletin Boards, via Packet, and via direct telephone between users. The extent of coverage will depend upon the amount of inputs from the readers, as there is little "in-house" expertise on this subject.

FORMAT OF UPDATE MAGAZINE

Back last year, UPDATE began with a couple of ideas that became too difficult to manage. One was Coloured page sections. The trouble with that is that paper suppliers have no standard colors. On month pink is rose and next month it is some other shade. So, I abandoned that Idea.

Another plan was to have sections by major subject area. This worked out well for only a few major sections, Telecomputing, TS-2068 Extra Memory, Schematics, and The Electronics Data section. All other subjects seem to blend together.

This October 1988 issue is the first issue to be printed by a commerical printer, and that makes it not to be feasable to continue with the colored page section idea. The editor will avoid "skip continuations" of articles from an earlier page to a page that is deeper within the magazine. This will allow those who want to file pages by subject area to do so.



A QL Address File

Bob Hartung, 2416 N. Co. Line Rd., Huntertown, IN 46748

Some like myself who are new kids on the block with the QL may be looking for info on getting started in QL SuperBasic programming. The QL User's Guide (QLUG) is something less than a masterpiece of clarity, and most current publications seem to have moved on to more advanced topics, so, from the perspective of a rank beginner, here are a few "QLues" I have stumbled onto so far. Assuming that not everyone has installed Toolkit II (TKT) or Trump Card, I will use only the syntax accepted by the unexpanded QL unless otherwise noted. Direct command entries will be indicated by <command> with a slash / between those keys to be pressed at the same time.

Since the best way of learning is usually by doing (there are exceptions such as learning what kinds of wild mushrooms are safe to eat!) I will give the listing for a SuperBasic program, then explain some of the whys and wherefores of it. A reader requested a mail list program, which seems a practical way of illustrating some points. Of course the ARCHIVE database SW bundled with the QL may be adapted for this but all its bells and whistles and whirring microdrives sometimes seem overkill for a simple mailing file and printout.

To make this into something for everyone, I will also try to use syntax that can most easily be translated by TS2068 (and Beta Basic) users, rather than exclusively-QL structures and commands that might be more elegant but less familiar. Purists who are bothered by my use of subroutines instead of structured procedures may easily replace the REM labels with DEFine PROCedure name, the RETurns with END DEFine, and the GO SUBs with the respective procedure names. I ran into an anomaly, however, on the SORT routine when defined as a procedure in that it re-orders a file perfectly the first time but then seems to lose track of some file items on subsequent calls. I'll leave that as a challenge for a sharp T/S UPDATE reader to figure out!

The minimum essentials of a mailing list are to provide the means of entering, storing, retrieving, and printing out the files. It is also useful to have the means of re-ordering them alphabetically, usually by last names, selecting particular ones for display or printout, adding to the file, and editing or deleting. This program will do all these, albeit more slowly than a whiz-bang machine code or compiled one. To economize on page space a 56-column listing format was used instead of the 40-column MODE 8 or the 64-column MODE 4. Now with all these disclaimers stated, let's get on with it.

For less pain and strain in entering the listing, <AUTO> supplies line numbers starting at 100 in increments of 10 or <AUTO n> will start at line n in steps of 10. Use <CTRL/SPACE> to break out of AUTO. During entries of a new listing or when editing an old one, <AUTO 10> or in the same increments as the listing will automatically bring each line into edit mode and <ENTER> will scroll to the next. If <EDIT n> is used to bring down a specific line, then the up or down cursor keys may be used to scroll other lines into the edit window as long as ENTER has not yet been keyed. Of course TKT users have the many conveniences of full-screen editing but for those who don't have TKT the QLUG doesn't go into much detail on how to use these normal EDIT functions to make life a bit easier.

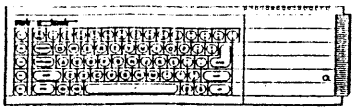
I would suggest starting by entering line 1540 DEFine PROCedure sv: DELETE MDV1_addrfile: SAVE MDV1_addrfile: END DEFine sv. If you do not have the TKT FREE_MEM command you might also want to add line 1550 DEFine PROCedure freemem: PRINT PEEK_L(163856)-PEEK_L(163852) -1024: END DEFine freemem. Make the first save of this with SAVE MDV1_addrfile, or change all the addrfile names to BOOT if you want it to auto-load. Now whenever you want to make a save during entry of the listing all you need to do is key <sv>. Enter <freemem> to show free memory remaining. Procedure and function definitions like these may be located anywhere and are transparent until called, i.e. processing passes right over them as it does for a DATA line.

Use RUN to initiate and key "9" from the menu to save the listing to mdv1. To set up an all-new file, key "0" and "y" at the ERASE prompt. 820 allows entry of 1, 2, 3, 4, or 5 for auto-entry of the respective titles (no space between number and first name). If 9 is the first character entered here then any other title or name may be keyed in letter-by-letter following the 9. $STOP will close the file and GO SUB to the sort and re-order routine 20, otherwise inputs continue for last name, address, P.O., state & ZIP, and ID or phone no.

The state and ZIP entry may be skipped with just ENTER (N/L) if the P.O. entry was made with a number. 860 & 890-910 may be expanded to include more auto-entries of P.O., state, and ZIPs for frequently-occurring ones. To avoid confusion, use line-increments of 1 for this and wait until the entire listing has been entered and checked before using RENUM 10,10.

Printer control commands for font, margin, etc. may be set by your panel switches or a PRINT #7,command-sequence for your particular printer may be inserted at line 1300 after an OPEN #7,ser1. Be sure to end this sequence with a CLOSE #7.

TV MODE 8 (40-column display with two screens) is assumed so you may wish to adust some of the PRINT and



10 sets black PAPER and white INK for the #2 main display screen and white PAPER and black INK for the #0 editing window. This gives a crisper display on B&W TVs and composite monochrome monitors than the power-up colors. Just use 10 GO TO 510 here if you prefer to keep the original screens.

Note that in lines 130-200 you may use apostrophes rather than quotation marks to define character-strings. The QL accepts either. The apostrophe is a little handier as it requires no SHIFT, but for clarity and of course for T/S translation the quotes should be used. The reverse slash \ used by the QL for a line feed corresponds to the apostrophe for the T/S. PAUSE without a given value is the same as T/S PAUSE 0.

A little surprise for T/S users (that users of many other computers have had to reckon with all along) is that it cannot be taken for granted that dimensioning a numeric array as DIM a(n) always sets every element to 0. Neither does DIM a$(n) set every string element to a space or CHR$(32). The DIM sets the string array elements to CHR$(0), and then if LET a$(n)="" is used they become spaces. If no key is depressed, CODE(INKEY$) also returns 0. This is the reason for the value-checks shown in 210, 610, and 810.

The IF-THEN structure of the QL allows nested test-statements, with ELSE alternative conditions that are active when IF conditions are not met, as in 330 and 350. 350 could also have additional statements appended such as ELSE OPEN #7,scr: INK #7;7 (which sets scr INK to white or the color of your choice). The end of a particular set of IF-THEN-ELSE statements is marked by an END IF to indicate that set of conditions no longer apply to what follows.

The SELect ON sequences in 370-430 and 630-740 may be translated as IF-THEN statements for the T/S. Beta Basic (BB) provides the ELSE function and also has an ON command that would require this change:

610 LET m$=INKEY$, m=VAL m$+1: IF m$="" THEN GO TO 610

This allows lines 630-730 to be entered as only one line to use the BB ON function:

630 ON m: GO TO 770: GO TO 790: GO TO 970: 110: GO SUB 20: GO TO 1210: GO TO 1320: GO TO 1420: GO TO 1310: SAVE "addrfile" LINE 10.

Lines 1330-1510 show how the QL sends and receives data to and from devices such as microdrives, modems, printers, networks, etc. For the T/S all this SAVE and LOAD footwork can be replaced by the simple commands SAVE "name" DATA f$() and LOAD "name" DATA f$(). If LET f$(400, TO 3)=STR$ last is inserted before the data save, then LET last=VAL f$(400, TO 3) would re-define the value of last after the loading of new data. (The QL has no VAL function.)

LISTING 2 gives a search routine for T/S users. When using it, the target-line to be searched is first marked by number, then any consecutive set of characters found at the very beginning of that line may be used as a search-word. Either a hard-copy printout or screen display may be selected. Line 350 could be changed to IF y$="y" THEN OPEN #2,"P" and make 470 CLOSE #2 to direct the PRINT output of lines 380-460 to a printer or to the screen if y$="n". IF-THEN statements would then be substituted in 380-420, and the #7 omitted. In BB an ON command-sequence could be used here if 360 is made LET p=CODE f$(n)-48 so the number-sequence 1-9 is produced.

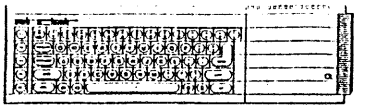
In BB the sort routine at 30-90 may be replaced by the simple command: 30 SORT f$(1 TO last)(21 TO 40) which will re-order 150 array elements by last names in just over 1 second. The QL search routine would be replaced by the following in BB:

```
130 INPUT " LPRINT y/n? ";y$
140 INPUT "CHR$ in search-word (#=wild):  ";c$
150 FOR n=1 TO last
160 LET p$=f$(n)
170 IF INSTRING(1,p$,c$) THEN PRINT n'f$(n, TO 94)'f$(
n,95 TO 114): IF y$="y" THEN GO SUB 350
180 NEXT n
280 PRINT #0;"SEARCH complete": PAUSE 300: GO TO 500
```

INARRAY would search the entire f$ array but line 150 allows INSTRING to search one array element at a time. Most BB commands and functions allow similar savings in memory-usage as well as processing time which offsets to a great extent the limitations imposed by the 22K free memory remaining with BB in-residence. About 150 file-items may be handled by this program using BB in a Spectrum-emulated TS2068.

In edit mode, ENTER scrolls through the file items. Entering "%" will delete that item, or if a number is entered then that line-item may be re-entered as desired. If more than one line is to be changed the "6" repeat option may be used at the end of the edit mode sequence, or else it might be simpler to use "%" to delete that whole item, then go to the "Additional entries" mode from the main menu. While entering new file items, if $TO is entered instead of an ID or phone no. then that entire file-item may be re-entered to correct an error, or else you can wait and use the "List & edit mode" from the main menu.

Again BB has a much easier way of editing an entire array element at a time by replacing lines 1000-1200 with simply 1000 EDIT f$(n) and 1010 NEXT n. The entire file item is brought back into edit mode for revisions



by moving the cursor up and down as well as horizontally through it, just as when it was originally entered. Keying ENTER scrolls through the file.

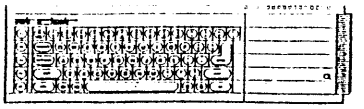
The "SAVE file" and "LOAD file" modes require a cartridge in mdv1 (or mdv2 with required changes in listing) so that the CAT routine may list all files already on the cartridge. This allows checking file names already in use to avoid overwriting any you wish to keep. Since I have TKT installed which doesn't require the DELETE mdv1_name before saves, the very first save you make of the listing (and the save of a file using a given name for the first time) may require adding an optional procedure without the DELETE on unexpanded QLs to provide for such saves. If a save is being made with the same name as an existing file without preceding it with the DELETE MDV1_name command, TKT users receive a warning prompt and the "Y/N" option to abort or proceed.

I think the remaining program REMarks and prompts should be self-explanatory, but then I'm looking at it through different eyes than someone seeing it for the very first time, so feel free to bring any errors or unclear points to my attention. For reply please enclose a SASE. I have written an all Beta Basic version of this program which will appear in a future SYNCWARE NEWS if all goes well.

It would serve a real need, I believe, if QL, T/S, and BB users would send in your own programming hints, especially those procedures and functions you have created, so they can be shared here in UPDATE with other users. And be sure to let Bill and his advertisers know you appreciate their support. As Rod Gowen of RMG ENTERPRISES, one of the very supportive surviving T/S venders, stated in recent correspondance with me, "We Sinclair users have to stick together!"

LISTING 1 - QL ADDRESS FILE

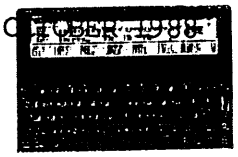
```
10 PAPER 0: PAPER #0;7: INK #0;0: CLS: CLS #0: GO TO 510
20 REMark sort
30 LET m=last: LET s=m
40 LET s=INT(s/2): IF s<1 THEN RETurn
50 IF s/2=INT(s/2) THEN LET s=s+1
60 FOR n=1 TO m-s
70 LET j=n
80 IF f$(j,21 TO 40))f$(j+s,21 TO 40) THEN LET x$=f$(j,1
    TO 115): LET f$(j,1 TO 115)=f$(j+s,1 TO 115): LET f
   $(j+s,1 TO 115)=x$: LET j=j-s: IF j>0 THEN GO TO 80
90 END FOR n
100 GO TO 40
110 REMark search
120 CLS
130 INPUT " LPRINT y/n? ";y$
140 INPUT "CHR$ in search-word (#=wild):   ";c$
150 FOR n=1 TO last
160 LET p$=f$(n)
170 IF INSTRING(1,p$,c$) THEN PRINT n'f$(n, TO 94)'f$(n,
    95 TO 114): IF y$="y" THEN GO SUB 350
180 NEXT n
190 PRINT #0;"Search is complete": PAUSE 300: GO TO 500
340 REMark Printout subroutine selects printer or screen
350 IF y$="y" THEN OPEN #7,ser1: ELSE OPEN #7,scr: INK #
    7,7: REMark Or INK n of your choice for scr
360 LET p=CODE(f$(n,1))
370 SELect ON p: REMark Alternative to IF-THEN sequence
380   =49: PRINT #7,"Mr. & Mrs. ";
390   =50: PRINT #7,"Mr. ";
400   =51: PRINT #7,"Mrs. ";
410   =52: PRINT #7,"Ms. ";
420   =53: PRINT #7,"Miss ";
430 END SELect
440 END IF
450 PRINT #7,f$(n,2 TO CODE(f$(n,115)));" ";f$(n,21 TO 4
    0)\f$(n,41 TO 65)\f$(n,66 TO 80)\f$(n,81 TO 94)
460 IF y$="n" THEN PRINT #7; f$(n,95 TO 114)
465 REMark For labels insert PRINT #7,\\\ here for LFs
470 CLOSE #7
480 RETurn
490 REMark Menu
500 CLS: CLS #0
510 PRINT "0 - Start ALL NEW file"
520 PRINT "1 - Additional entries"
530 PRINT "2 - List & edit mode"
540 PRINT "3 - Search mode"
550 PRINT "4 - Sort & re-order"
560 PRINT "5 - LPRINT"
570 PRINT "6 - SAVE file"
580 PRINT "7 - LOAD file"
590 PRINT "8 - Set printer"
600 PRINT "9 - Listing SAVE"
610 LET n=CODE(INKEY$): IF n=0 THEN GO TO 610
620 CLS
630 SELect ON n
640   =48: GO TO 770
650   =49: GO TO 790
660   =50: GO TO 970
670   =51: GO TO 110
680   =52: GO SUB 20
690   =53: GO TO 1210
700   =54: GO TO 1320
710   =55: GO TO 1420
720   =56: GO TO 1310
730   =57: sv
740 END SELect
750 GO TO 500
760 REMark Initialize ALL NEW file
770 INPUT #0;"CAUTION--ERASE OLD FILE y/n?      (Maximum
    no. files<400)   ";y$: IF y$="y" THEN DIM f$(400,1
    15):CLS #0: ELSE PRINT "FILE NOT ERASED": GO TO 510
780 REM Input to file
790 FOR n=1 TO 400
800 LET p=0: REMark p=zero
810 IF f$(n,1) <> CHR$(0) AND f$(n,1) <> " " AND f$(n,11
```

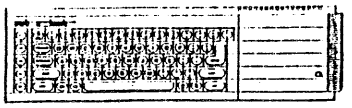


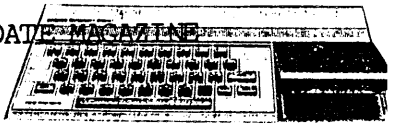

```
    4) <> "%" THEN NEXT n
820 INPUT (n)\"$STOP 1Mr/Mrs 2Mr 3Mrs 4Ms 5Miss or 9 the
    n title or first name "\p$: LET f$(n,1 TO 20)=p$: L
    ET f$(n,115)=CHR$(LEN(p$)): IF p$="$STOP" THEN LET
    f$(n)="": LET last=n-1:  GO SUB 20: GO TO 500
830 IF CODE(f$(n,115))<2 THEN LET f$(n,115)=CHR$(2)
840 INPUT "Last name"\f$(n,21 TO 40)
850 INPUT "Address"\f$(n,41 TO 65)
860 INPUT "P.O. or 1AUB 2BUT 3HNT"\f$(n, 66 TO 80): IF f
    $(n,66)<="9" THEN LET p=1
870 INPUT "State (2 chrs+2 spaces) & ZIP or N/L"\f$(n,81
     TO 94)
880 INPUT "ID or phone no. (or $TO for re-entry)"\f$(n,9
    5 TO 114)
890 IF f$(n,66)="1" THEN LET f$(n,66 TO 80)="Auburn": LE
    T f$(n,81 TO 94)="IN 46706)"
900 IF f$(n,66)="2" THEN LET f$(n,66 TO 80)="Butler": LE
    T f$(n,81 TO 94)="IN 46721"
910 IF f$(n,66)="3" THEN LET f$(n,66 TO 80)="Huntertown"
    : LET f$(n,81 TO 94)="IN 46748-nnnn"
920 IF p THEN PRINT f$(n,66 TO 94)\
930 IF f$(n,95 TO 97)="$TO" THEN PRINT "RE-ENTER": GO TO
     820
940 END FOR n
950 LET last=n-1: GO SUB 20
960 GO TO 500
970 REMark EDIT file
980 CLS #0: PRINT #0;"Key ENTER to keep or % to delete
      ITEM NO. to EDIT": LET z=0
990 FOR n=1 TO last
1000 PRINT "FILE ";n\1;"="; f$(n,1 TO 20)\2;"=";f$(21 TO
    40)\3;"=";f$(n,41 TO 65)\4;"=";f$(n,66 TO 80)\5;"="
    ;f$(n,81 TO 94)\6;"=";f$(n,95 TO 114)
1010 INPUT m$;: IF m$="%" THEN PRINT "DELETE!!": LET f$(
    n,114)="%": LET f$(n,21 TO 40)="zzzzzzzzzz": LET z=z
    +1
1020 IF CODE(m$))37 THEN GO SUB 1130
1030 PRINT
1040 END FOR n
1050 GO SUB 20
1060 LET x=last
1070 FOR n=x-z TO x
1080 IF f$(n,21 TO 30)="zzzzzzzzzz" THEN LET last=last-1
    : LET f$(n)=""
1090 END FOR n
1100 CLS
1110 CLS #0: INPUT #0;"Key ENTER to return to menu or 6
    to repeat "; y$: IF y$ <>"6" THEN GO TO 500
1120 GO TO 970
1130 PRINT "-LINE REVISION:"
1140 IF m$="1" THEN INPUT p$: LET f$(n,1 TO 20)=p$: LET
    f$(n,115)=CHR$(LEN(p$)): IF CODE(f$(n,115))<2 THEN L
    ET f$(n,115)=CHR$(2)
1150 IF m$="2" THEN INPUT f$(n,21 TO 40)
1160 IF m$="3" THEN INPUT f$(n,41 TO 65)
1170 IF m$="4" THEN INPUT f$(n,66 TO 80)
1180 IF m$="5" THEN INPUT f$(n,81 TO 94)
1190 IF m$="6" THEN INPUT f$(n,95 TO 114)
1200 RETurn
1210 REMark LPRINT envelopes or labels
1220 INPUT "Continuous printout y/n? ";y$
1230 FOR n=1 TO last
1240 IF y$="n" THEN CLS #0: INPUT #0;(n)\(f$(n,1 TO 40))
    \"KEY s TO SKIP or ENTER for next ";m$: IF m$="s" TH
    EN NEXT n
1250 LET p=CODE(f$(n,1))
1260 OPEN #7,ser1
1270 GO SUB 370
1280 END FOR n
1290 GO TO 500
1300 REMark Insert YOUR printer commands here
1305 GO TO 500
1310 REMark Save-data routine
1320 CAT
1330 INPUT #0; "SAVE file name? ";y$: LET m$="mdv1_" & y
    $: REMark Or use "mdv2_" here & at line 1430
1340 DELETE m$
1350 OPEN_NEW #7,m$
1360 FOR n=1 TO last
1370 PRINT #7,f$(n)
1380 END FOR n
1390 CLOSE #7
1400 CLS #0: GO TO 500
1410 REMark Load-data routine
1420 CAT
1430 INPUT #0;"LOAD file name? ";y$: LET m$="mdv1_" & y$
1440 DIM f$(400,115)
1450 OPEN_IN #7,m$
1460 FOR n=1 TO 400
1470 IF EOF(#7) THEN LET last=n-1: GO TO 1510
1480 INPUT #7,f$(n)
1490 END FOR n
1500 LET last=n-1
1510 CLOSE #7
1520 CLS #0: GO TO 500
1530 DEFine PROCedure CAT: CLS:DIR MDV1_: END DEFine CAT
1540 DEFine PROCedure sv: DELETE MDV1_addrfile: SAVE MDV
    1_addrfile: END DEFine sv
1550 DEFine PROCedure freemem: PRINT PEEK_L(163856)-PEEK
    _L(163852)-1024: END DEFine freemem: REMark Allows
    free memory to be displayed by keying-in freemem as
    direct command or as a listing command
```

LISTING 2 -- T/S Search Routine

```
130 PRINT "KEY LINE NO. TO SEARCH:"\
140 PRINT "1 - Title no. & first name              (1Mr/M
    rs 2Mr 3Mrs 4Ms 5Miss)"
150 PRINT "2 - Last name"
160 PRINT "3 - Address"
170 PRINT "4 - P. O. or city"
180 PRINT "5 - State abbrev."
190 PRINT "6 - ZIP"
200 PRINT "7 - ID code or phone"
210 LET m$=INKEY$: IF m$="" THEN GO TO 210
```







```
220 INPUT "Leading chr-set in item ";(n$)\c$
230 INPUT "LPRINT y/n? ";y$
240 PRINT #0;"ANY KEY FOR NEXT"
250 LET s1=1*(n$="1")+21*(n$="2")+41*(n$="3")+66*(n$="4"
    )+81*(n$="5")+84*(n$="6")+95*(n$="7")
260 LET s2=s1+LEN(c$)-1
270 FOR n=1 TO last
280 IF c$=f$(n,s1 TO s2) THEN GO SUB 340
290 PAUSE
300 PRINT
310 NEXT n
320 PRINT "SEARCH completed"
330 CLS #0: INPUT #0; "Key ENTER for menu or 6 to repeat
    "; y$: IF y$="6" THEN CLS #0: GO TO 110
335 GO TO 490
```

Invitation to participate

ARTICLES and PROGRAMS SUBMISSION

UPDATE ENCOURAGES writers to submit articles, utilities and programs for publication. The FORMAT is very relaxed. Just pay attention to the width of the pages and allow about 3/4 inch top and bottom and make the left and right margins wide enough to accommodate a 3 hole punch that does not cut out some of the text. We have enlarge and reduction capability to adjust print to fit a page. Most important is legibility. Draft quality dot matrix does not reproduce well. Print size preference is ELITE 12 characters per inch. Pica 10 characters per inch is OK also, but please NO condensed print. If an article has illustrations, tables, schematics, pictures, figures, thats alright. Just let us know how to organize them.

Some of the print in the articles of this issue is smaller than desired because the pages were reduced to provide margins, and some of the articles have quite a bit of wasted page space due to excessive margins. But if written text can be salvaged we will do that instead of re-typing it. Solid type that is clear but faded can be enhanced at UPDATE, but draft quality dot matrix cannot, espically that produced by the TS-2040. That is all the guidance to be given. Dont want to make it too tough to participate and scare you away. The users NEED the SHARING OF KNOWLEDGE.

TS-2068 PROGRAMS and UTILITIES

PROGRAMS to publish for the TS-2068 may be LLISTED in 32 chr lines, but that is not necessary. Each program should be supplied on 40 or 80 track disk or cassette. Oliger, Larken, or Aerco FD-68 formats are fine. SPECTRUM programs should be capable of running in the TS-2068 that is equipped with the Spectrum ROM Switch. We ESPECIALLY need programs and utilities designed for use with DISK DRIVES.

SINCLAIR QL PROGRAMS and UTILITIES

QL Programs and utilities should be supplied on either 80 track floppy or micro drive. All programs must be usable with the UNEXPANDED QL. If programs are supplied on disk, a version should be recorded that will COPY TO MICRODRIVE and operate from micro drive. Your letter should contain enough instructions about the disk or micro drive so that the editor can load and operate.

UPDATE welcomes QL programs which have expansion capabilities to use with the QL that has RAM expansion, but such programs should also have a micro drive version which works with the un-expanded QL.

Exceptions to the above are utilities designed to be specifically for disk drive, micro drive, Trump or Sandy, or supporting utilities for other devices.

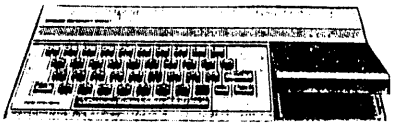
Full Fledged Software for the QL tends to be too long for key in projects and these programs should be accompanied with an article designed as a "mini manual". QL Utilities and short softwares can be LISTED in the magazine as key in projects and the users will get much programming knowledge from an explanation of HOW the program works.

The QL and the TS-2068

GENERAL INTEREST ARTICLES

REVIEWS of new software and equipment are welcomed. UPDATE will provide enough space for comprehensive reviews.

PROGRAMMING: This is what the users ask for most and is the most flexible. The users need programming informtion ranging from "one liner tips" to simple basic programming, to advanced programming. TS-2068 users are also eager for programming information about disk drive use. QL users are asking for Super Basic programming information, ranging from beginner level to such topics as networking between two QL's, telecommunications with modems, piping data between two computers, and conversions of MS-DOS programs for use with the QL. So, let the limits of your writing be the limits of your knowledge.



TS-2068 EXTRA MEMORY RECAP

Back in January 88, UPDATE began a series about EXTRA MEMORY. The first objective was to explore the ways and means of using EXISTING Extra Memory Hardware. Briefly these hardware devices are: The Oliger EPROM cartridges for Dock Bank use: The 32K non volatile memory cartridge for use in the Dock and XROM banks: The Aerco FD-68 Disk Controller which contains un-assigned extra Ram 64K blocks. Recently one more addition is the LARKEN Ram Disk cartridge.

The existing TS-2068 allows us to use programming to enable the Dock and XROM banks of memory when RAM or EPROM is installed in these two banks. There is no automatic way of using extra memory in the TS-2068, unless such auto usage is provided for by PROGRAMMING. So, the January UPDATE delved into the programming needed to use these existing devices, which generally had not been covered in sufficient detail for the devices to be widely accepted and used. This involved "collecting" pieces of data about programming from several sources and putting together a group of utilities to allow the user to move BASIC programs or utilities into the Dock or XROM banks and operate the programs. The UPDATE series on EXTRA Memory continued in tutorial form to include Operating programs in both HOME and DOCK banks and switching from one to the other at will. Then the Disk Drive was brought into the picture and utilities developed for Auto Booting programs into Home RAM, followed by AUTO MOVING the program into Dock bank, and then Auto Booting another basic program into Home Bank— to operate two programs at will. This effectively magnifies the SIZE of programs that the TS-2068 is capable of handling and frees up the FREE MEMORY (ram) for an extraordinary increase in DATA handling capability (within RAM).

The above represents the TOTAL capability of the TS-2068, as it was manufactured, to utilize MEMORY. We cannot go beyond using the Dock and XROM banks without re-designing the TIMEX ROM. AND, it isn't easy for a user to get that far, because the user must learn programming to do it himself. The tutorials and utilities given in UPDATE were designed to allow a person who is interested in programming to learn how to do it, and for programmers to use to develop software to operate from Extra Memory.

UPDATE realizes that many users wont delve into the detailed study needed to become profficient in using Extra Memory, and that this series is of little interest to them. But there are some who are eager to extend their programming abilities. Then there is a small group of real hackers who want to go the limit and build onto the TS-2068. So, just as UPDATE plays to the novice programmers and to those who are interested in only using software, not constructing it, we also want to fill the needs of the intrepreneur and the expert.

ENTER CHARLES BOTHNER, to extend the series on Extra Memory "to the limit". This issue brings another detailed article by Charles. In addition to his extensive project to extend Memory Addressing, Charles has given other building schematics. One is a Auxillary Power Supply to power external add on devices, which most of us need to prevent TS-2068 overload. I have already burned out one TS-2068 due to overloaded it with add ons. Hopefully one of our hardware builders will pick up on this need and offer a power supply that plugs on to the external Buss of the 68 and power everything else that is tagged on. WE NEED THAT!

In this issue, Charles casually verifies the previous UPDATE information about Dock and XROM bank transfers, and with a snort of impatience, he proceeds to the sublime, leaving most of us in a fog of miscomprehension. But not really. If you really dig in and cross reference the Timex Tech manual, things come into focus. A good many are digging this and are asking for more. We have some real technocrats using the TS-2068!

Actually, if you will notice, Charles does not apply TITLES to his works, just branch headers. If I had to coin a title, I would use: "ENGINEERING A Z80 BASED COMPUTER". Then I might add: ("Using the TS-2068 as a Starter"). His works go beyond the UPDATE series title of Extra Memory while he stretches the Z-80 CPU to it's limit. There's a guy in England who did something similiar, and his product is called "The Sinclair Z88". I don't know where Charles Bothner is taking us, but UPDATE will continue with his works until the end. If someone really wants to learn the practical aspects of computer engineering, address coding, assembly language, and the Z-80 CPU, this is a GREAT tutorial on all of that.



By: Charles Bothner

SEPT. 2, 1988

Bill Jones
TS-2068 UP-DATE
1317 Stratford Avenue
Panama City, FL 32404

Dear Bill,

First of all, I would like to comment on your proposed use of both the HOME and DOCK banks for programs simultaneously. The switching between these banks by manipulating bit 7 of the ARSFLG variable @23750 should work as you described. The only known problems in the use of the DOCK bank are those described in paragraphs 6.2 and 6.3.1 of the Timex technical manual and these deal with cartridges which only use machine code. (They are easily corrected by changing the byte @0930H in the EXROM from 40H to 55H and the two bytes @38A0H and 38A4H in the home ROM to 60H and 40H respectively.) I have not analyzed the DOCK bank access routines in detail up to this time because I have not had any problems using the DOCK bank for any of its normal stand-alone functions.

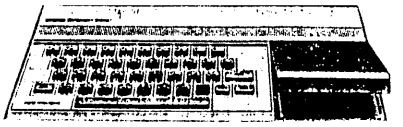
You mentioned power problems with the 2068 and I agree completely that extra loading on the internal power supply should be kept at a minimum. I even went so far as to relocate my LKDOS controller board from the cartridge slot to one of the bus expansion boards at the rear of the computer so that it would be on the external power system. The only extra loading on the internal supply is the 2 chips which were added for proper EXROM address decoding and DOCK/EXROM disabling when an expansion bank is active as described in my last article.

My present bus expansion port plug-ins are lined up as follows:

1. RGB monitor interface and phoneme speech synthesizer. (The external power supply is now connected to this board.)
2. EXROM bank expansion for chunks 1-7.
3. LKDOS controller board.
4. Expansion bank controller board.
5. Port I/O buffer board with real time clock
6. LKDOS disk port board
7. 16 channel A/D board.
8. Text-to-speech board.
9. Digitalker speech synthesizer board.
10. RS-232 interface board.
11. TS 2050 Modem interface.
12. AERCO printer interface.

(You can see that I have been experimenting with several different approaches to speech synthesis for computers.)

Now let me bring you up to date on my progress in developing the expansion bank hardware and software. I have completed



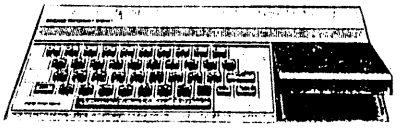
By: Charles Bothner

the debugging of the EXROM initialization routines. The original Timex protocol was followed in all respects except for the deletion of the routines which reassigned expansion bank numbers according to their interrupt priorities. This was done for several reasons. First, the interrupt service routine address field which is stored in the SYSCON table for each expansion bank is never accessed by any of the existing Timex ROM code. Second, there is no way of currently making use of interrupts from the expansion banks without changing the interrupt mode of the Z-80 processor and if a different interrupt mode were used and the HOME bank then made active, the system would crash. Third, it is unlikely that there would ever be so many expansion banks on line that polling would present an excessive time delay within system routines.

Before I go any further on the system debugging I will refer any interested readers to the excellent 5 part series of articles by Wes Brzozowski entitled "The Mystery of the Missing 253" which appeared in Time Designs Magazine starting with the July/August, 1986 issue. These articles provide a good background description of the ROM routines. These articles actually got me started on the debugging of the system ROM's.

I completed the debugging of the initialization routines dealing with the ROM (or PROM) based expansion banks over a month ago but the RAM based banks were more of a problem. The RAM based expansion bank initialization routine consists of testing each chunk in the bank for the presence of RAM by means of first writing a test byte, then reading it back and comparing the read back data with the original byte to see if RAM is actually present in that chunk. This test is repeated using two different bytes to verify that there actually is RAM in the chunk and not an accidental match with random garbage read back from an unused address.

The original test routine in the EXROM blithely continued chunk by chunk until chunk 3 was enabled in the expansion bank without bothering to move the O.S.RAM routines from chunk 3 in the home bank to chunk 7 so the system crashed at this point. I added the necessary code to move the O.S.RAM routines to chunk 7 prior to testing chunk 3 in the RAM banks, but the system still crashed. The problem this time was that the routine in the EXROM which moves the O.S.RAM code makes use of data stored in the system variables area which have not been initialized at this point in the overall system initialization sequence. The addition of some code to temporarily initialize these variables solved that problem but we were not out of the woods yet. The system now initialized properly and the SYSCON table was loaded with the correct type of data but the chunk map stored in SYSCON 02 for RAM banks did not show the proper status of chunks 2 and 3 for these banks. Subsequent testing of the write and read-



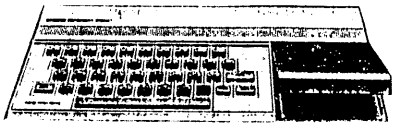
By: Charles Bothner

back of memory in these RAM bank chunks indicated that the data was being written in correctly but was read back correctly approximately 10% of the time. Apparently the SCLD chip in the computer was somehow insisting that the home bank should retain control of chunks 2 and 3 since both of these are used for screen display storage. (Chunk 3 is only used for extended display modes.) Attempts to override this control by means of DI software instructions and also by setting bit 6 in port FF proved fruitless. Since the internal architecture of the SCLD chip is unknown, I decided to try the same type of hardware override approach that Timex used when they added the U21 circuit to tristate two of the address and data buffer chips (U5 and U9) when writing to the DOCK bank. The added circuit is shown on one of the enclosed schematics. After adding this circuit which tristates the buffer chips whenever chunks 2 or 3 of either the EXROM bank or expansion banks are active, the SYSCON table was correctly loaded during the initialization sequence. A copy of the SYSCON table data is enclosed along with a brief description of the function of the entries in the table. The SYSCON 03 through 17 entries for the RAM banks is whatever garbage was in the chunk 0 RAM at power up. It is unused data for these banks.

I am also enclosing a schematic of a LED display add-on for the expansion bank boards which I found to be very useful to show which chunk in each bank is currently active.

As promised in my last letter to you, I am enclosing an annotated copy of the modified O.S.RAM routines. Most of the routines have either been corrected or entirely rewritten. The only routine which is not annotated is the DISPATCH routine which serves no useful purpose at the present time. This routine will be the only one which will be entirely deleted if I can come up with some useful routines which will help make use of the expansion banks transparent to the programmer working only in BASIC. This will be one of my next projects now that the debugging of the original Timex code and associated hardware seems to be complete. I want to apologize for the handwritten annotation for this table but much of this was done during extensive business travelling over the past few months. Also enclosed is an updated FIX table for the O.S.RAM code which is used by the code which moves the RAM routines back and forth between chunks 3 and 7 in the home RAM.

During my analysis of the O.S.RAM routines, I originally thought that there was an error in the CALL BANK routine. This routine is used extensively by the tape I/O routines in the original Timex ROM's, but never with any parameters passed on the stack, either in or out. Detailed analysis of the CALL BANK routine indicated that it is correct as originally written, and the called routine is responsible for cleaning up the stack by deleting the passed parameters



By: Charles Bothner

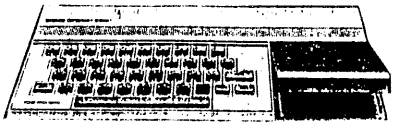
after the data has been used by the called routine.

Prior to calling a routine in an expansion bank with parameters passed to the routine, the following items have been stored on the stack in the order indicated:

1. Bytes containing the status of expansion banks. (One byte per bank containing the active chunk map.)
2. Two bytes containing the status of ports F4 and FF.
3. The address inside the CALL BANK routine which will be returned to at the completion of the execution of the code in the expansion bank.
4. The chunk mask and bank number of the called bank.
5. The number of parameters (bytes) to be output to the called routine.

The stack pointer is left pointing to item 3 when the expansion bank is jumped to. I am using the following routine in the expansion bank to recover the stack space utilized by the passed parameters while preserving all Z-80 registers:

```
F3      DI              Disable interrupts.
DDE1    POP IX          Save return addr. in IX.
E5      PUSH HL         Save HL(overwrite return
                        address)
C5      PUSH BC         Save BC(overwrite chunk
                        mask and bank number)
210000  LD HL,0          Make HL=SP
39      ADD HL,SP
2B      DEC HL          Point to high byte of #
                        of output parameters.
46      LD B,(HL)       Get high byte of # of
                        output parameters.
2B      DEC HL          Point to low byte of #
                        of output parameters.
4E      LD C,(HL)       Get low byte into C
F9      LD SP,HL        Assign new SP value.
D5      PUSH DE         Save original DE value.
110600  LD DE,6         Offset from SP to start
                        of parameter list.
19      ADD HL,DE       Point HL to parameters.
1100C0  LD DE,C000      Destination address in
                        bank RAM for storage of
                        parameters.
EDB0    LDIR            Move the parameters from
                        the stack to the RAM in
                        the expansion bank.
D1      POP DE          Restore DE data.
C1      POP BC          Get # of parameters back.
D5      PUSH DE         Save DE again.
F5      PUSH AF         Save AF.
2B      DEC HL          Point DE to top address
E5      PUSH HL         in stack where parameters
```

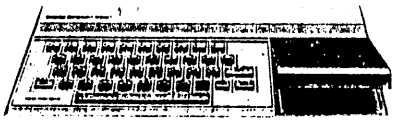


By: Charles Bothner

```
D1      POP DE          were stored.
A7      AND A           Clear carry, if set.
ED42    SBC HL,BC       Offset HL by number of
                        parameter locations.
010800  LD BC,8         8 bytes on stack to be
                        moved.
EDB8    LDDR            Move the stored register
                        data to recover the stack
                        space previously occupied
                        by the parameters.
D5      PUSH DE         Point SP to start of the
E1      POP HL          stored register locations
23      INC HL
F9      LD SP,HL
F1      POP AF          Restore original register
D1      POP DE          values.
C1      POP BC
E1      POP HL
DDE5    PUSH IX         Put the return address on
                        the stack.
CDxxxx  CALL xxxx       Call the routine in the
                        expansion bank which
                        makes use of the passed
                        parameters.
FB      EI              Enable the interrupts.
C9      RET             Return to the CALL BANK
                        routine.
```

I have also developed a routine for loading parameters onto the stack within a called expansion bank routine when the calling routine requires parameters to be input rather than output to the called routine but I will save that for next time.

A brief explanation of the expansion bank hardware operation may be in order. The following refers to the schematics which were published in your last edition. U10, U11, and U18 on the expansion bank controller board are used to buffer the address and data lines which are not already buffered inside the 2068. U19 provides bidirectional buffering of the data to and from the expansion banks. U21, U3, U4, and U7 provide the full port address decoding for either reads of or writes to ports FC and FD. The port FD read function is not utilized at the present time. U17 latches the 3 high order data bits that are transmitted from the computer when a port FD WRITE command is executed. The latched data is then decoded by U13 into a one-of-eight outputs for selection of the appropriate expansion bank port commands.
The subsequent write to or read from port FC will utilize the latched U13 output to determine the function being executed. For example, a read from expansion bank port 40H will be the low nybble of the currently selected banks chunk mask. The bank was selected by sending the bank number to U2 via a write to port 80H. U1 decodes the output of U2 to



By: Charles Bothner

select one of the 15 possible banks. U15 is used to deselect all chunks in all expansion banks either during a power up reset or by writing a 0 to port C0H. U14 is used as a controlled deselect of chunks common to all expansion banks when a specific chunk is to be reassigned to a new bank. (This is done by the BANK ENABLE routine in the home bank RAM.) The rest of the logic on the controller board is straightforward combinatorial logic to select the required control functions.

Proceeding to the schematic for the individual expansion bank boards, we see that U6 is used for selection of the appropriate chunk selection when memory read/write is active. The memory chips on the board will only be enabled if the specific chunk was previously assigned to the bank. This is determined by the state of the U3, U4, U9, and U10 registers. They are clocked by a write to expansion bank port 40H of the required chunk mask which is presented to the D input pins of the registers. The clear pins are directly driven by the chunk deselect outputs of U14 on the controller board. U1 is used to determine the status of the bank. Bit 2 is read back as a low to verify that a bank is really present when testing for banks during the system initialization. Bit 3 is used (via U8) to determine whether any chunks in the system chunk mask are currently assigned to this bank. A low indicates that one or more chunks are active in this bank and a high read back indicates that no chunks are currently assigned to the bank. U12 is used to pull the BUSISO line low whenever a chunk in the bank is actually being read or written to.

This is a quick summary of overall circuit operation. I'm sure there will be questions from your readers so we will try to answer them in future articles. Now that the basic system hardware and software are in place, the big job remaining is to develop useful software for the ROM expansion banks and available space in the home bank to make use of the expanded memory. Some of your readers that are familiar with the home ROM code may be able to help out on this proect.

One last item for this time. I am enclosing a disk containing the clock setting and reading routines for the circuit which was shown in the last article. You can list them out on whatever printer you prefer. Next time we will start out with a listing of the debugged home ROM code.

Note: For those readers who have a need, UPDATE will supply the two missing items:

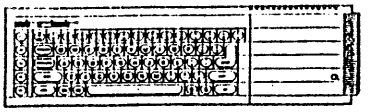
1. Copy of disk mentioned, Oliger or Larken format- $5.00 for postage and handling.

2. 26 page code listing starting at SYSCON 5EEA hex through 683F hex.- $8.00. (ed.)

sincerely,

C G. Bothner

Charles G. (Chuck) Bothner



The CABLE COLUMN

This feature column, by Bill Cable, starts with this issue. There are several excellent programs in this one episode. All of these programs will be consolodated on one ISSUE DISK and ISSUE MDV CART. There is just barely enough space on one mdv cart to contain them all. This probably sets the record as the largest offering of QL software in one disk or cartridge EVER, at anything near the price.. Bill describes each in the following articles. The first article explains a lot that most of us do not know about ARCHIVE, why we should use it, and how to use it. Then he gives us a tutorial on using ARCHIVE. The articles following are SOFTWARE programs that are for ANY QL computer, with or without expansion devices.

The second article brings the software "REFIELD" and is a mini manual on what REFIELD does and how to use it.

The third article brings another software "HELPER" which is a set of TOOLKIT functions for any QL computer.

The fourth article brings another software "SCAN" which also gives the actual listing of the program, extensively documented with REMarks for those who would like to key in the program or learn its programming.

The fifth article is about a Math program for the kids which will be included in the "Cable Disk and MDV Cart".

WOOD AND WIND COMPUTING

Bill Cable : RR 2 Box 92 : Cornish, NH 03745 : 603 675 2218

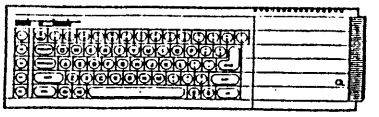
WHY USE ARCHIVE ?

Bill Cable

This is an open letter to all QL users to urge you to use ARCHIVE. There are many new US QL users out there plus a lot of older ones who have virtually ignored the ARCHIVE database program supplied with the QL. Maybe you took a cursory look at it and decided because it doesn't have any immediate flashy screens and it has a large set of commands that have to be learned that it wasn't worth the effort. Although it is not real flashy at first, it can be made flashy, but that is not the point. It is a workhorse. It can manipulate and organize whatever information you put into a database with amazing speed and ease of effort. Unfortunately, even if you know the language you can't do much until you enter enough information to manipulate. So it does take an effort. But you can also purchase ready made archive programs. These programs are run within the ARCHIVE environment and provide specific services to the user. You may still have to enter the data but the programs make it as easy as possible. There are several address databases available and perhaps other software as well.

I would also like to urge more programmers to use this language for program development. It works very well for complex business systems and many other applications where QL users would be willing to pay for software.

Below I will give 12 reasons to use ARCHIVE. I hope this will convince some of you nonusers to explore it a little and see what a wonderful piece of software you got for nothing. Next, I will point out a few shortcomings of ARCHIVE. After all, I don't claim it to be perfect. And last, I will give a mini ARCHIVE lesson for beginners which will include a single line of ARCHIVE commands which will calculate and



print on the screen how many countries of the world have a population density less than the US and what their sum total population is, using the Gazet database provided on your database cartridge. Beat that, HyperCard! Maybe that will whet your appetite for extracting trivial information from Gazet.

12 Good Reasons To Use ARCHIVE

1. The price is right. That lonely cartridge is setting there covered with dust. All you need to do is open your manual, press <F1> and you are on your way. The supplied database, Gazet, gives you something to play with. At the end of this letter you will find a mini ARCHIVE lessons for total beginers.

2. There is a great need for information handling. All of us could use more help in storing and organizing information in our lives. This is what a computer does best. ARCHIVE provides a very good environment to handle information. You can do it yourself or buy ready made programs.

3. Learning to use a database is important for all serious computer users. The concepts used in ARCHIVE are the same used in other databases. Learning ARCHIVE makes learning others easier.

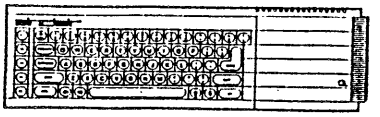
4. It is universal among all QL users and is not protected. There are a few new database systems beside ARCHIVE for the QL and it nice to have a choice but it will be some time before a large number of us owns a copy.

5. There is an IBM compatible version of the PSION progrrams ($100). So any work done on the QL can be transferred to the IBM compatible environment.

6. The ARCHIVE operating system and language is very well designed. I have systems of procedures with over 70k of source code. I find ARCHIVE's behavior completely reliable and predictable. No lockups or program crashes. Whenever I think I have found a system bug and track it down it turns out to be my bug. In my opinion it is a at least a couple of notches better than the other PSION programs.

7. The language is excellent. It is like a structured SUPERBASIC with no line numbers. Some commands differ but it is procedure oriented. The built in editor is good. If it had a search/replace no other editor would ever be needed. Its automatic indenting makes program legibility very good. I hardly ever list out a program in hardcopy. I can do all my design and debugging on screen. Error messages are very good. Leaving out an 'endif' etc. cause a structure fault message that is a heaven send. Errors messages pin point the offending procedures by name. Errors can also be totally supressed to make friendly crash proof programs for others.

8. Archive is the best place to keep the source of any information destined to ABACUS, EASEL, or QUILL. Because it has a language, you can restructure the information in any way desired for the other PSION programs. You can print fancy reports to files for import to QUILL. You can send information and numbers to ABACUS for easier manipulation. you can send summary information to EASEL for graphs. You have total power over your information if you know how to use ARCHIVE or purchased a program from someone who does.



9. Archive has excellent number crunching power. Much better than SUPERBASIC.

10. Compared to some other databases, its handling of fields, sorting, and ordering is nice and straight forward although maybe not as flexible. It can order a 2000 record file in 2 minutes. Not super fast but very good for most applications. It can handle many files at once.

11. It can be made to do a lot even on an unexpanded machine. Programs can be broken into parts and loaded and run automatically as each part is needed. This allows more free data area and running program too big for memory. You can also write programs that write programs and merge and run them without restarting anything. A program I wrote to easily change the fields of a database (REFIELD), takes advantage of both these properties.

12. It has a separate screen editor which makes complex screen design and construction relatively easy and different screens can be loaded at any time.

Some limitations of ARCHIVE

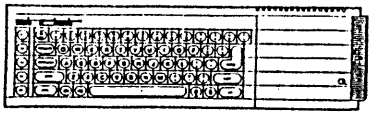
1. The worst problem is the possiblility of corrupting a database file by 'opening' it for insertions, alteration, deletions and not closing it before turning the computer off, losing power, or QL crash. The fact that this happens mandates keeping multiple backup copies. Any time changes are made to the database back it up. Backing up is important for any computer user anyway but beginners are often slow to learn this until they really get burned. Using microdrives calls for even more caution. If your QL unpredictably locks up, get it fixed or be very careful. This may appear to make ARCHIVE too dangerous to use but in practice, if you back up you will lose little if anything ever. When 'looking' there is no danger. There are programs to recover ARCHIVE files but they are not easy to use and nothing can be guaranteed. Preventive action is safest and best. I like to use 2 or 3 backups which I keep alternating between. I put the day of the year in pencil on the backup. That way I can find my latest easily.

2. No detailed documentation of Archive. I think the ARCHIVE section of the QL manual is pretty good as far as it goes. There are a few typo mistakes in the examples but it is worth using as a guide. The sections defining each command and function and error message should be studied carefully. There is ARCHIVE MASTER ($45) from RMG enterprises. I haven't seen it. Also TAKING THE QUANTUM LEAP ($26) from TIME DESIGNS is a more general book about the QL.

3. Slow screen handling. You can design very nice screens and data displays but you can't get around the screen as fast as you would like. This may be part ARCHIVE and part QL.

4. No data arrays allowed. There are field related functions that use field numbers as indexes which are very handy. You can make a dummy database and use it like an array. Position can be one index and the field number the other. If it is in RAM it will be fast.

5. On an unexpanded machine it is easy to run out of memory when doing selections and orderings of large databases.



6. Not much software yet. This will improve. There is a good address database from EMSOFT (MAILBAG), which will handle printing labels, postcards, repeat letters. I have a utility to allow easy addition of fields (REFIELD) to any already created databases. I have a tutoring database (DBTUTOR)to help learn any subject. I am working on a very comprehensive program for the small business to handle all record keeping, invoicing, check writing. It has been under development for over a year. I also offer an address database (DBADDRES). There are probably more US ARCHIVE based programs not known to me.

Conclusion

We are very lucky that PSION did such an excellent job developing this program. It will allow your QL to do first rate work for you for many years to come. You only need to learn a little about using it.

Mini ARCHIVE Lesson For Beginners

1. Turn on or reset your computer. Put cartridge labeled 'Database' in the left microdrive slot (mdv1_). Press <F1> key.

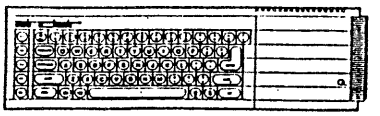
2. ARCHIVE will load and you will see a little '>' at the bottom of the screen waiting for a command. All commands can be seen by pressing <F3> and watching the upper center box. If you type a command and <ENTER> it will be executed. Press <F5> to cause the last command line to be retyped (an easy way repeat a command or correct a mistake). Pressing the <ESC> key aborts a command.

3. You were given a database,Gazet, with information about all the countries of the world on your database cartridge. Let's play with it for a little while.

```
look "mdv1_gazet"<ENTER>          {access the database}
print count()<ENTER>                        {total records}
display<ENTER>        {display a record with field names and values}
select continent$="S.AMERICA"<ENTER>     {select a sub group}
print count()<ENTER>                        {how many included}
display<ENTER>                              {display again}
order pop;a<ENTER>                {order population in ascending order}
last<ENTER>                       {most populated}
back<ENTER>                       {next most populated}
first<ENTER>                      {least populated}
next<ENTER>                       {next least populated}
reset:print count()               {reset back to all countries}
{Try on your own ordering and selection}
reset:select pop/area<.0229:let p=0:all:let p=p+pop:endall:print count();":";p
        {above is the promised command. US density = .0229, notice the result}
        {play some more}
        quit<ENTER>                       {always quit when done}
```

You just learned the basics of using a database.

Sure did! Ed



REFIELD PROGRAM v1.1u

An ARCHIVE Program by Bill Cable

A. What does it do?

Refield is a large program running under ARCHIVE that allows one to add, restructure, delete fields from your ARCHIVE database. If you are familiar with ARCHIVE programming you can do that on your own but it is tedious and easy to make errors. There are also ways to do it by exporting to ABACUS and re-importing. Refield attempts to make it fun and error free, even for a beginner. It works on an unexpanded machine.

What is a field you ask? Well it is a category to hold information for each entry in your database. The Gazet database provided with your computer has these fields for each country : country$, continent$, capital$, currency$, languages$, pop, area, gdp. For each country record we have this information. Now if we also wanted to keep track of miles of roadway in each country we would have a problem. With Refield it would be very easy to add a field called 'roads'. What would happen is that it would make a new database with the extra field and copy the records over. Suppose we named a field badly in our database. With refield we could easily rename the field. Suppose we were running out of room on our cartridge and there were a few fields we could do without. With refield we could make a new database and leave out those fields to gain a little more room. Or we could break the database into two parts and keep them on separate cartridges. It takes the tedium out of such problems and is fun to watch because it tells you what it is doing as it chuges along.

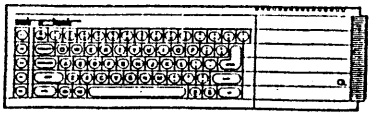
B. Starting up

1. Set up files in RAM if you want fastest running
2. Start archive normally
3. Type: run "{}step1"<ENTER> where {} = device holding refield programs(mdv1_,mdv2_,flp1_,flp2_)
4. Follow directions

C. Brief Explanation On How To Use The Program

To start from archive : run '{}step1'<ENTER> {} = device programs are on.
It will work on an unexpanded QL but only works in 80 column mode.
The Source database is what you have and the Target database is what you want.
It transfers records from the Source to the newly defined Target.
The program is divided into step1_prg to step7_prg which are run in sequence.
It works by writing the procedures needed to make the Target database.
These are merged and run at the appropriate times.
Prompts are usually at the very bottom of the screen.
All responses require the <ENTER> key being pressed as part of the response.
Possible answers to prompts are often within [ ] and separated by /.
Comments within *8 9* clarify prompts.
Comments within { } indicate action program is taking.
Often just <ENTER> can replace the normal expected response.
Error messages appear at top right of screen.
You can usually restart the step if an error occurs.

This series of procedures works with an archive database. It allows for the addition of new fields, removal of fields, reordering of fields, renaming of fields, changing a numeric field to a string and viceversa. It uses your original (Source) database as a template and creates a new (Target) database with the changes you want. The Source records are then transfered to the Target database. You are given the opportunity to select and order the Source records to be transfered so the Target database can have the structure you want. One use of this program would be to make an exact copy of your database because the new version would be cleaner as it would not have the holes and



disorder of the original caused by record deletions and alterations. I have attempted to trap most errors and give ample directions. You should have some knowledge of archive fields and the select and order commands. If you use ramdisk you should set that up before starting archive. The procedures have been broken up into a series of steps so they will work on an unexpanded machine. If you don't have ramdisk or floppies then put your Source database in mdv1_ after archive is loaded and have the refield programs in mdv2_. Near the end you will be notified that the programs are no longer needed so you can remove them from mdv2_ and put the Target cartridge there so the transfer can take place. The Source and Target files can be on the same device but must have different names.

D. Hints about using this program

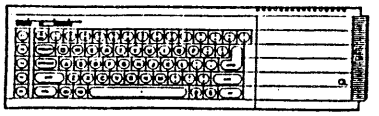
1. Pressing <ESC> a few times will stop the program and get command mode.
2. Typing help<ENTER> when in command mode will fix screen and give help.
3. When an error is detected you will be given a message in top right of the screen with a prompt. Then you usualy are given the option of restarting the step you are on or from step1 or going to command mode.
4. Some errors may not be detected and will put you at command mode with an archive error message and a line of code from the program. You will have to deal with that as best you can. Typing help<ENTER> will give help.
5. Some errors cause archive to lose track of what files are accessed. If you really get stuck, quit archive and start it up again.
6. I have used the archive convention of field numbers (#) starting at 0. so if a database has 8 fields, the field numbers will be 0 through 7.
7. I gave the field name 'tag' a special meaning. If you give the Target database a new field called 'tag' it can be given a special value so that each record in the Target database will be given a unique increasing integer in the 'tag' field.
8. Apparently the basic rules of field names are : up to 13 characters, the first character must be A-Z, a-z then 0-9 and _ can also be used. If a field ends in $ then it is string, otherwise it is numeric. I may not catch all illegal field names. Especially if you use an archive reserve word in a field name. This could cause step5 to crash. Type help<ENTER> to restart.
9. When you are defining the Target fields, giving it the same number of fields as the Source and making then the same will create an exact copy with a 'cleaner' structure.
10. If you have a new name for an old field, you enter the new name where you want it and give the Source field # when prompted.
11. With a completely new field you will be prompted in step 6, if you want it to have a default value different than for string or 0 for numeric.
12. If you give a numeric Source field a string name in the Target database it will automatically make the conversion when transferring.
13. While defining target fields, a little note (S#, where # is the Source field #) is put by the Target field name when Target value = Source value.
14. Using ramdisk for programs, Source, and Target is pretty fast. Some messages appear and disappear instantly. Using all microdrives can take a very long time for large databases.
15. If you want to multitask, you can shut off the screen updating of progress.

E. PROGRAM STRUCTURE

There are 7 files of programs chained together

step1_prg : Asks for device programs are on (mdv1_,mdv2_,flp1_,ram1_ ,etc.). It checks that temporary files can be written to that device and provides these detailed instructions.

step2_prg : Asks for Source database device and file name. It opens the Source database for 'look'.



step3_prg : Creates an select/order procedure file on the target device and writes your select/order commands there.

step4_prg : Merges with the select/order procedures and carries them out. Only activated if selecting or ordering are requested.

step5_prg : Makes creater procedure file for Target database. Prompts for Target fields. Can have partial or complete duplication of the Source. You can type in a Target field name or get it from the Source. Each new name can get its value from the Source. A temporary database is created on the programs device.

step6_prg : Makes transfer procedure file. prompts for special values for new fields (normally string = and numeric = 0).

step7_prg : Merges creater and transfer procedures. Gets Target device and file name. Creates Target database and does the transfer.

F. ERROR/HELP Message

```
Suggestions
 Type 'help<ENTER>' to get this screen again when in command mode"
 Make sure specified devices and files are correct
 Make sure a necessary drive is not full
 Make sure tabs aren't removed from a necessary cartridge

 Jot down error number and look it up on page 53 in archive manual
   Last error #  =
 If you can fix the problem without closing files  type   start<ENTER>  or
 Try starting from step1 again or that failing quit archive and restart

   Specified refield programs device :
   Specified Source database         :
   Specified Target database         :

Restart this step  - [ y/n ] :
Restart with step1  - [ y/n ] :
Close all files before stoping - [ y/n ] :
```

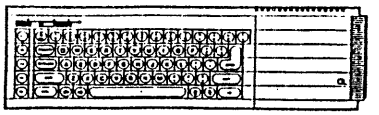
C. Support software development

This software is not intended to be free. If you obtained it from UPDATE Magazine or directly from me, Thank You. If you got it for free and find it helpful, please support my effort in trying to make good applications software for the QL. Send me $5.00.

Bill Cable
Wood and Wind Computing
RR 2 Box 92
Cornish, NH 03745 USA
603-675-2218

Specializing in Business Applications for the QL

You should find this to be a very reliable and flexible program with lots of different applications. Please notify me of any problems or useful features to add.



HELPER

A SUPERBASIC Program for the QL by Bill Cable

The helper program is a set of useful toolkit-like functions presented in a menu format. It would be very useful to anyone without any of the toolkits available for the QL and works on unexpanded machines.

The basic functions provided :

1. Directory of a device with file copy,delete,view options. The directory is constantly displayed while the functions are being carried out.
2. Formatting a device 1 to 10 times.
3. Copying the contents of one device to another (clone).
4. Set time and date.
5. Load and run a program from a specified device.
6. Set windows and exit to QDOS.

There is a fair bit of input checking to prevent errors happening but not every one can be prevented. Trying to copy a nonexistent file is an example of an error that will stop the program. I decided not to use WHEN ERROR to trap errors because of the system bug with that function and the objective to keep things simple. If it does crash it won't hurt anything just note the cause of the error and restart the program by typing : 'run<ENTER>'.

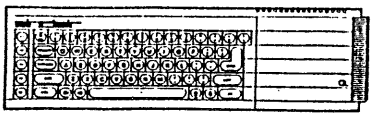
The code is straight SUPERBASIC. I indented the multiline IF statements for easier reading. It would be easy to customize and add your own special functions. The single line 'IF' statements don't require an 'ENDIF' and I wasn't always consistent in including one. I prefer using them for easier program reading.

To Run :

type : 'lrun ()helper' where ()=device holding helper
mdv1_,mdv2_,flp1_,flp2_

You are warned about errors and encouraged to use <CTRL><F5> to freeze screens and <F5> to unfreeze. Note that these always work on the QL.

You are next presented the HELPER Menu which should be self expanatory. Experiment a little. The date and time need be set only once after the QL is turned on even if it is reset. Exit to SUPERBASIC to quit. I hope you find it helpful.



ARITHMETIC

An Arithmetic Drill Program for the QL By Bill Cable

This program is designed to aid children learning arithmetic. It uses only whole numbers 0 through 9 for addition, subtraction, multiplication, division. Numbers can get as large as 81 (9x9). It picks the numbers randomly and presents the problem on the screen waiting for the users answer. The faster the response the better the rate (problems per minute). The challange would be to have a high rate with good accuracy. The results can be printed out so a printed record of progress can be kept. The elasped time is also kept so one can determine when they have been working for a certain period of time.

The program is pure SUPERBASIC. The Menu portion uses mode 4 while the problems are flashed in mode 8 to get big characters. There should be no way to crash the program except <CTRL><space bar>.

To Run :

Type : 'lrun ()arithmetic' where ()=device holding program mdv1_,mdv2_,flp1_,flp2_

You are first asked your name for display purposes and to use on the print out (if any).

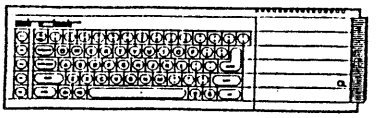
Next you are asked if you want a print out. It would be a good way to record long term progress. The rate of doing problems should slowly increase. You must keep the printer online if you do.

If print out is wanted you are asked for an identifying entry such as the date. This will allow you to keep track oflong term records.

Next you are presented a menu of choices :

a - Addition
s - Subtraction
m - Multiplication
d - Division
p - Progress (anytime)
q - Quit (anytime)

A single key press will activate your choice. If you choose to do problems you will be given the problem and the cursor will flash in the place for the answer. When you give enough digits for the answer it will be checked. If right you will go on to another. If wrong you will be asked again. Pressing the 'p' key at any time will show your progress. It is a good way to take a break as it stops the clock. Whenever you see your progress a print out will be made if you wanted print outs. Pressing 'q' will get you the menu and pressing 'q' again stops the program. Happy Arithmetic



SCAN

An ARCHIVE program by Bill Cable

Scan is an ARCHIVE program that will work with any database. It allows one to see up to three selected fields displayed on a line for all currently active records. It is a fast way to scan through the file looking for particular values or trends. If any of the fields are numeric then they are displayed with a specified number of decimal places and their sum is printed at the end. The sum may or may not have meaning depending on what the field is. The sum will appear anyway. The output can also be sent to the printer at the same time. This could be handy for many purposes. You could list names and phone numbers from an address database to keep by your phone. The listing will be of the currently selected/ordered structure of the file. Pressing the <space bar> will pause the output and allow for continuation or abort.

This program is nice because it will work with anyone's database and will not interfere with any already existing procedures or variables as long as they are not named 'scan'. It makes use of the ARCHIVE fieldt(),fieldn(),fieldv() functions so it can deal with any database without having any previous information about it. The only way to crash it if my code is good, is to call it without having an active 'look' or 'open' database file. I documented the code heavily to show my logical process. In a way it makes the listing hard to read. You may want to omit or delete all the 'rem' statements to shorten it and see what real ARCHIVE code looks like. To get the ARCHIVE editor, type : 'edit<ENTER>' and follow instructions in center box.

To Run :

Start ARCHIVE as usual

Access your database with 'look' or use the Gazet database
look "mdv1_gazet"

Load or merge scan as :
load "()scan" where ()=device holding scan
mdv1_,mdv2,flp1_,flp2_

Select or order if you want

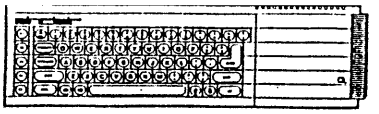
Type : 'scan<ENTER>' to start scan

Direct output to printer if you want

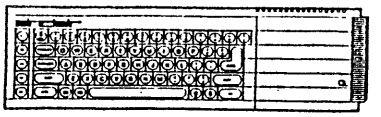
The fields of your database will be displayed. Give the numbers of up to 3 you want on the output line for each record. If the field type is numeric you will be asked for the number of decimal places to show.

The listing will begin. Press <space bar> to pause.

It will print any sums and pause. Stops When you press <ENTER>

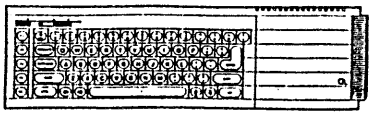



```
proc scan
  rem  SCAN : A procedure for listing all current records on the screen and
  rem   to the printer. It prints one line with up to 3 selected fields for
  rem   each record. It asks for decimal places for the numeric fields to be
  rem   displayed. It prints the sum of any numeric fields displayed at the
  rem   end of the listing. A <space bar> press will pause it and then it
  rem   can be continued or aborted. It will work with any database you have
  rem   accessed by 'look' or 'open'. All variables are local so it should
  rem   not interfer with your procedures or variables. All these 'rem'
  rem   statments can be removed to shorten it.
  rem
  rem  To Use :
  rem          1. Start ARCHIVE as usual
  rem          2. Access your database by 'look'
  rem          3. Load or merge scan : 'load {}scan' or 'merge {}scan'
  rem          4. Select or order you records you want to see if necessary
  rem          5. Type : 'scan<ENTER>' and follow prompts
  rem          6. It quits after reaching the end or you abort it
  rem          7. Call it any time by typing its name after it is loaded
  rem
  rem
  rem  Set up all variables as local so no interference with other procedures
  rem   All other symbols are commands  or functions defined in archive section
  rem
  rem  Uses : f1,f2,f3  are field numbers or -1
  rem         d1,d2,d3  are decimal places or -1
  rem         f1$,f2$,f3$  are used to hold values in form for display
  rem         s1,s2,s3  are used to hold sums of numeric fields
  rem
  local i$,i,j,k,f1,d1,f1$,s1,f2,d2,f2$,s2,f3,d3,f3$,s3,print_out
  rem
  rem  Initialize potential decimal places for three fields to -1
  rem
  let d1=-1: let d2=-1: let d3=-1
  rem
  rem
  rem  Set mode 0 for large screen, print heading, if no file is open an error
  rem   will occur at 'print count()' since there is no count
  rem
  mode 0: print "Scan up to 3 specified fields of ";count();" selected records "
  print at 2,2;"The records will be scrolled by with selected fields on a line"
  rem
  rem  Loop until answer is 'yes' or 'no'
  rem
  let i$="": while i$<>"y" and i$<>"n"
    print at 4,0;rept(" ",80);
    input at 4,2;"Output to printer also [ y/n ] : ";i$
    endwhile
  rem
  rem  Set print_out 'true' if 'yes' or 'false' if 'no'
  rem
  if i$="y": let print_out=1: else : let print_out=0: endif
  rem
  rem  Print out all fields with field number in front of it. If more than 50
  rem   fields they are not printed
  rem
  cls : print "Fields that can be scanned": print : let i=0: let j=2: let k=0
```




```
while i<numfld()
  print at j,k;num(i,2);"  "+fieldn(i): let i=i+1: let j=j+1
  if j=12: let k=k+15: let j=2: endif
  endwhile
rem
rem  Prompt for first field of scan. If it is not a number or out of range
rem   return otherwise set f1 to the field number
rem
input at 15,0;"First field to include in scan [ #/<ENTER> if none ] : ";i$
let f1=int(val(i$)): if f1=0 and i$<>"0": mode 1: return : endif
if f1<0 or f1>numfld()-1: mode 1: return : endif
rem
rem  Get decimal places desired in listing if field is numeric, set sum = 0

rem
if not fieldt(f1)
  print "  Decimal places for "+fieldn(f1)+" : ";: input i$;: let d1=val(i$)
  if d1<0: let d1=0: endif : if d1>10: let d1=10: endif : let s1=0
  endif
rem
rem  Prompt for second field to scan. If not a number or not in range set
rem   f2 = -1 otherwise f2 set to the field number
rem
input at 17,0;"Second field to include in scan [ #/<ENTER> if none ] : ";i$
let f2=int(val(i$)): if f2=0 and i$<>"0": let f2=-1: endif
if f2<0 or f2>numfld()-1: let f2=-1: endif
rem
rem  Get decimal places desired in listing if field is used and numeric
rem   set s2 = 0 , keeps track of sum of field 2
rem
if f2<>-1 and not fieldt(f2)
  print "  Decimal places for "+fieldn(f2)+" : ";: input i$;: let d2=val(i$)
  if d2<0: let d2=0: endif : if d2>10: let d2=10: endif : let s2=0
  endif
rem
rem  Prompt for third field to scan. If not a number of not in range set
rem   f3 = -1 otherwise set f3 to field number
rem
input at 19,0;"Third field to include in scan [ #/<ENTER> if none ]  : ";i$
let f3=int(val(i$)): if f3=0 and i$<>"0": let f3=-1: endif
if f3<0 or f3>numfld()-1: let f3=-1: endif
rem
rem  Get decimal places desired in listing if field is used and numeric
rem   set s3=0, keeps track of sum of field 3
rem
if f3<>-1 and not fieldt(f3)
  print "  Decimal places for "+fieldn(f3)+" : ";: input i$;: let d3=val(i$)
  if d3<0: let d3=0: endif : if d3>13: let d3=13: endif
  endif
rem
rem  Print out heading and fields selected and message about halt
rem
cls : print "Scan selected fields of ";count();" selected records"
print at 2,2;"Press any key for temporary halt"
print at 3,2;"The number in the left column is the record number"
print at 6,0;"Rec #"; tab 11;fieldn(f1);
if f2<>-1: print tab 31;fieldn(f2);: endif
```



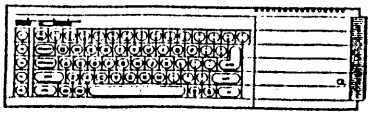

```
if f3<>-1: print tab 54;fieldn(f3);: endif : print : print
rem
rem  If print_out warn to put printer online. Print heading to printer
rem
if print_out
  print at 4,2;"***  Make sure printer is online  ***": print at 8,0;
  lprint "Scan selected fields of ";count();" selected records": lprint
  lprint "Rec #"; tab 11;fieldn(f1);
  if f2<>-1: lprint tab 31;fieldn(f2);: endif
  if f3<>-1: lprint tab 54;fieldn(f3);: endif : lprint : lprint
  endif
rem
rem  Set to beginning and begin printing line by line until end of file

rem
first : while not eof()
  rem
  rem  Check for key press signaling pause request. Allow for restart if only

  rem   if <spacebar> pressed otherwise return
  rem
  rem
  if inkey()<>""
    print : print "Press <space bar> to continue or any other key to stop : ";
    if getkey()<>" ": mode 1: return : endif : print
    endif
  rem
  rem  Store value of 3 fields to f1$,f2$,f3$ making decimal if numeric
  rem
  if fieldt(f1)
    let f1$=fieldv(f1): else : let f1$=dec(fieldv(f1),d1,12): endif
  if f2<>-1
    if d2=-1: let f2$=fieldv(f2): else : let f2$=dec(fieldv(f2),d2,12): endif
    endif
  if f3<>-1
    if d3=-1: let f3$=fieldv(f3): else : let f3$=dec(fieldv(f3),d3,20): endif
    endif
  rem
  rem  Keep sum of any numeric fields
  rem
  if d1<>-1: let s1=s1+fieldv(f1): endif
  if d2<>-1: let s2=s2+fieldv(f2): endif
  if d3<>-1: let s3=s3+fieldv(f3): endif
  rem
  rem  Start printing line. First record number then first field then 2nd and

  rem   3rd if the are defined. Likewise print to printed if print_out true

  rem
  print recnum(); tab 8;f1$;
  if print_out: lprint recnum(); tab 8;f1$;: endif
  if f2<>-1
    print tab 28;f2$;: if print_out: lprint tab 28;f2$;: endif
    endif
  if f3<>-1
    print tab 48;f3$;: if print_out: lprint tab 48;f3$;: endif
```




```
    endif
  rem
  rem  Finish print line
  rem
  print : if print_out: lprint : endif
  rem
  rem  Advance to next record
  rem
  next
  endwhile
rem
rem  Listing done. Print out sums if any
rem
print : print "<";count();">";
if print_out: lprint : lprint "<";count();">";: endif
if d1<>-1
  print tab 8;dec(s1,d1,12);
  if print_out: lprint tab 8;dec(s1,d1,12);: endif
  endif
if d2<>-1
  print tab 28;dec(s2,d2,12);
  if print_out: lprint tab 28;dec(s2,d2,12);: endif
  endif
if d3<>-1
  print tab 48;dec(s3,d3,20);
  if print_out: lprint tab 48;dec(s3,d3,20);: endif
  endif
rem
rem  Wait for acknowledgement, set mode to 1 and return
rem
print : if print_out: lprint : lprint : endif
print : input "Scanning Done. Press <ENTER>";i$;: mode 1
endproc
```

Editor's comments: I learned more about ARCHIVE by reading these articles and a few hours working the programs than I could have in six months of study and experimenting without them. We can keep excellent programmers such a Bill Cable working for us ONLY IF we respect their propriety rights to their works and PAY THEM FOR WHAT WE GET.

Frankly, I see dozens of SHAREWARE (IBM) softwares being given about, in this small city, and VERY FEW bother to send the author payment. The tenancy is for one to think of his friend as the giver, and not the author of the software. I sincerely hope that the Sinclair QL users will not destroy their software support by disregarding INTEGRETY.

All of Bill Cables works in this issue will be in the October 88 "CABLE" ISSUE DISK and MICRO DRIVE ISSUE. You may Order the disk or send a mdv cart in for dubbing. Send orders or mdv cart to UPDATE, $20.00 postpaid, and specify "CABLE DISK".

Also, you may send your thanks and comments directly to Bill Cable. If you expect a reply, please inclose a SASE.

UPDATE SUPPLY SUPPORT

UPDATE ITEMS FOR SALE: These items are purchased in bulk and can be offered at lower than most local prices. All prices include shipping. Order by "UD-#". Add 25% for Canadian orders.

UD-#1 Desk top Disk Case, holds seventy 5.5" floppies, nice smoked plastic with hinged top— $7.50 each

UD-#2 Six outlet 115V Surge Suppressor- ON-OFF SW, 6' cord, 15 amp reset ckt breaker, ON-OFF led indicator— $19.95 each

UD-#3 Six outlet 115V Surge Suppressor as above, includes EMI/RFI/Noise/Surge suppression— $27.95 each

UD-#4 White Labels (same as UPDATE Mailing label), 5000 labels on continuous feed tractor form— $22.00

UD-#5 Diskettes 5.5" hi quality DS/DD 40 track, includes sleeves and tabs— $8.00 box of 10

UD-#6 Diskettes 5.5" Dysan Premium DS/DD 80 track 96 tpi. Use for all Sinclair QL and TS-2068 80 track drives— $20.75 box of 10.

UD-#7 Printer Ribbon Cartridge for Star Micronics 1000 series. Box of six cartridges— $39.75 (Single cart—$8.00).

UD-#8 Un-Priced: UPDATE will order ribbon cartridges in units of six cartridges and ship to you with a invoice. Price depends upon size of cartridge and will be within plus or minus $5.00 of UD-#7 above.

UD-#9 PAPER, 2500 sheets 20 lb white bond, tractor feed, laser perf for easy tear off, standard letter size break down to 8.5X11"— $31.90 (Check locally, you may be able to beat this)- US orders only.

THE FLEA MARKET

Subscribers may participate FREE to move surplus items. Limit is one 3 line insertion per issue. Make the print Elite 12 characters per inch, line lenght of 76 characters, including spaces. No fancy printing. Use the following for examples of format.

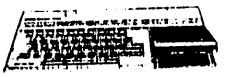
UPDATE SURPLUS items: For TS-2068: One Zebra OS-64 Cartridge— $16.00. One Tasman CPI— $16.00.

One A&J Micro Drive for the TS-2068 plus about 15 cartridges, Has printer interface and is in working condition—$40.00

One AMDISK III twin 3 inch Disk Drives in case with power supply and cable to fit all TS-2068 disk controllers, includes manual and 19 diskettes. Works great, just plug it in— $99.00. Know what your are buying! This DD is unsupported by software, but makes an ideal additional pair of drives. Data cable has 2 extra connectors for hooking up to other drives.

ONE TS-2040 Printer (works A-OK)—$25.00. One Seikosha SP-1000 dot matrix printer, works now but print head beginning to skip some pins—$55.00.





UPDATE FLEA MARKET, Continued

TS-1000 items for projects: 5 TS-1000 ckt boards all chips OK— ? Two Oliger Color VDP boards OK— ? Several EPROM Read Boards— ? A few ULA chips, Z-80 CPU's and Z-80 periphreal chips— ? These parts will be given to anyone who needs them for building interface projects for the TS-2068 or QL.

ONE TS-1500 like new—$35.00. ONE TS-1000 NEW—$30.00 (both have all assessories).

For the hacker: a cornucopia of electronics parts, diodes, transistors, IC's, relays, etc, on surplus circuit boards. Boards weigh about 2 pounds and must have over a hundred salvagable parts—$7.00 per board (cost nearly that to ship).

UPDATE Magazine is a sister to GULF MICROELECTRONICS, which has a most unusual stock of hard to find IC chips, analog devices, regulators, CPU's, interface chips, power transformers, vacuum tubes, etc. If you are looking for something and cannot find it, its worth a CALL.

One 1954 EDSEL—try $49000.00 or trade for new Mercades

One Missouri Mule 49 years old, mean and uncooperative, unusual personality, can hear him bray from four miles away. Come and get him for free. Might pay you something.

One island across from New York City. Indian tribe repossessing and reselling. Inquire for confidential deal.

Prospector's map to lost gold mine in Nevada—$100.00

Tickets to last year's Super Bowl 50 yd line, reduced price.

Persons wanting FLEA MARKET items should CALL first to reserve the item needed.

SOFTWARE AND HARDWARE SUPPORT FOR SINCLAIR COMPUTERS, (Complimentary listings)

WOOD AND WIND COMPUTING

Bill Cable : RR 2 Box 92 : Cornish, NH 03745 : 603 675 2218

SPECIALIZING IN SERIOUS SOFTWARE FOR THE SINCLAIR QL

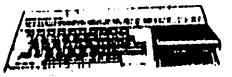
RMG ENTERPRISES
1419 1/2 7TH STREET * OREGON CITY, OR 97045
503/655-7404

For our big 50+ page catalog
send $3.00—Refunded on 1st order

QL and TS-2068 SUPPORT: RMG Enterprises has a full line of support for both computers. TS-2068 support includes all disk drive systems, hardware and software. QL support includes most periphreals and many software items.

TS-2068 Hardware Support: Disk Drive System, memory cartridges, EPROM programmer, Expansion ports, RGB Monitor Interface, Printer Interface: The Oliger Company, 11601 Whidbey Drive, Cumberland, IN 46229.





Millenia K Disk and SPDOS Software Support: Munson Cockayne, 342 Trotter Court, Sanford, FL 32763.

TS-2068 Disk Drive System and Ram Disk Cartridge: Larken Electronics, RR-2, Navan, Ontario, Canada K4B 1H9.

TS-2068 Software to PLOT and Paint Color Screens to paper; John McMichael, 1710 Palmer Drive, Laramie, WY 82070.

TS-2068 Software, also AERCO FD-68 Software: Chi-Chia Chao, 73 Sullivan Drive, Moraga, CA 94556-1209.

TS-2068 Software: Herb Bowers, 2588 Woodshire Circle, Chesapeake, VA 23323.

Millenia K Disk and SPDOS Software Support: Cuyohoga Valley Software, 615 School Ave., Cuyohoga Falls, OH 44221.

TS-2068 and QL Software: Paul Bingham, Box 2054, Mesa, AZ 85204.

TS-2068 Software, plus AERCO FD-68 software: Jack Dohany, 390 Rutherford Ave., Redwood City, CA 94061.

TS-2068 Software: Lemke Software, 2144 White Oak, Wichita, KS 67207.

TS-2068 Support, and Magazine on tape: Byte Power, 1748 Meadowview Ave., Pickering, Ontario, Canada L1V 3G8.

TS-2068 PROGRAMS

Diamond Mike II and *Great Game & Graphics Show!* Two great programs for only $10.

Money Machine II: The Ultimate Word Game based on a popular TV game show. $11.

Word Twister! Customize your own word-search puzzle. Vocabuilder! Easy-to-use educational program. Both for only $10.

Programs above are available on tape or Aerco FD-68 5 1/4" double density disk.

Following programs are for Aerco FD-68 only. Available on 5 1/4" disk or tape.

Disk File Manager! Copy individual files, even on single drive system. Detailed catalog, and many other functions. $13.

OmniDisk! Convert Omnicalc II to disk. Customize many options. $9.

Prices include postage. Please send 22-cent stamp for complete catalog or check/money order to!. Chia-Chi Chao 73 Sullivan Drive, Moraga, CA 94556-1209

Pixel Print PLUS!
THE DESKTOP PUBLISHER
by Lemke Software

What's the PLUS?
PERFORMANCE!

Checkout these SPECS:

1) WYSIWYG (What You See Is What You Get!)
Create your text on the screen... as easy as typing! LOAD graphics... HIGH FUNCTION: LOAD ICON (position cursor, Cmd. LOAD)

2) AUTOMATIC and MANUAL line and character adjustments.

3) RESTORE FONT letter using the BOLD/MODERN/ITALIC modifiers.

4) KEEP/UNDO/SAVE/LOAD/LOAD ICON LOAD SCREEN#/SAVE SCREEN# WIDE/HIGH/CLS/SCROLL SPEED

5) OVER/INVERSE/CAPS LOCK UP TO 16 POINT FONTS (Font package in develop.)

6) COPY/ERASE/INSERT/DELETE/MEM

7) AERCO/TASMAN/A&J CPI

8) IBM/EPSON/PROWRITER Printers

Pixel Print PLUS! has all of the features found in v 2.0, it is 100% compatible with the Pixel Print ICON #1 & #2 packages, FONTS #1, #2, & #3 as well as the TASWORD text, and your Pixel Print files!!

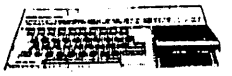
The PLUS is 10 more functions to make this an even more powerful/easy to use program.

Pixel Print PLUS (v3.1)
Order yours today!
(This ad created 100% with the Pixel Print PLUS program with FONTS and ICONS...!

Lemke Software
2144 White Oak
Wichita, Ks.
67207

PIXEL PRINT PLUS!....	$19.95
TASWORD TEXT CONV....	$19.95
ICON PACKAGE #1.....	$19.95
ICON LIBRARY #2.....	$14.95
FONT PACKAGE #1.....	$19.95
FONT LIBRARY #2.....	$14.95
FONT LIBRARY #3.....	$19.95





Computer Repair: TS-2068 and QL; Dan Elliott, RR-1 Box 117, Cabool, MO 65689.

TS-2068 and QL: Curry Computer, PO BOX 5607, Glendale, AZ 85312-5607.

TS-2068 Software: WMJ Data Systems, 4 Butterfly Dr., Hauppauge, NY 11788

QL Computer Repair: Also QUANTA SOFTWARE LIBRARIAN, 9015 Flicker Place, Columbia, MD 21045.

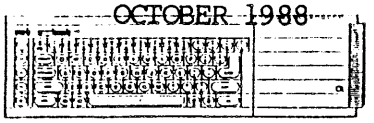
THE SINCLAIR Z88 Computer: Known dealers and suppliers are: Mike Fink, 355 W. 39th St., N.Y. City, 10018-1401. (212 695 1401), RMG Enterprises, 1419 1/2 7th St., Oregon City, OR 97045 (503 655 7484), SHARP'S INC., RR-10 Box 459, Mechanicsville, VA 23111 (804 746 1664).

See other ads and announcements in this issue. The above are just part of the supporters of QL, TS-2068, and Z88 Computer systems. Others will be announced as UPDATE receives more information that specifys support areas.

A caricature of the UPDATE News Room, relaxation between issues.



Actually, the above photo is a courtesy re-print from NATIONAL GEOGRAPHIC MAGAZINE, but it closely resembles the neat efficiency of the UPDATE editorial office and your editor relaxing between issues. Across in the other corners of the office are equally neat TS-2068 Lab, QL Lab, and A IBM AT clone system. The center of the office is taken up by a copy machine and a huge wastebasket, neatly overflowing with discarded ideas.





THE UPDATE SOFTWARE LIBRARY

Separate libraries for the Sinclair QL and TS-2068

Most of the diskettes and micro drive cartridges contain the softwares and utilities described or listed in a quarterly issue of UPDATE Magazine. Most will contain more than one software plus all of the utilities given in the quarterly issue. Some will also contain other programs not given in the pages of UPDATE. Each disk or mdv cartridge is "unconditionally guaranteed to please". The price of each named disk or mdv is $20.00 (postpaid).

The Sinclair QL Software Library

Both Disk and Micro Drive are supported. All softwares presented will run on the UNEXPANDED QL. But some may contain extensions for utilizing extra memory and tool kit functions. Disk format is DS/DD 80 track 96 tpi. Due to the cost of mdv carts and availability, orders for a issue series on mdv cartridge must be accompanied by a mdv cartridge. UPDATE then records the programs and returns the cartridge. UPDATE is Just getting started with the QL Library and only two Issue Series are offered at this time. Individual Issue Series are as follows:

OCTOBER 88 "Cable Issue"

The CABLE issue, on disk or mdv cart, contains the FOUR softwares given by Bill Cable in this current issue. Two of these are softwares to enhance and extend the capabilities of the PSION ARCHIVE software.

"REFIELD", runs under ARCHIVE to allow you to ADD, RESTRUCTURE, and DELETE fields of the ARCHIVE DATA BASE.

"SCAN", an ARCHIVE program, lists three selected fields on a line for each record, and SUMS the numeric content of the fields. The output can be sent to the printer.

"HELPER", is a Superbasic program which provides a new "TOOLKIT" set of functions. These functions are; Directory with Copy, DELETE a file or program, TOTAL COPY of a MDV or Floppy, Multiple FORMATTING, Screen Window Set-up, and Date-time setting.

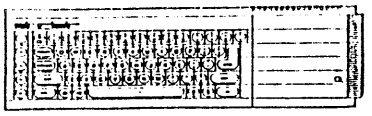
"ARITHMETIC DRILL", in Superbasic, is a simple program to help learn programming and let the kids use to drill in math.

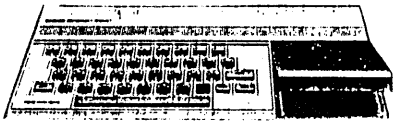
The above FOUR programs, and their annexes, come on TWO mdv Carts or one Diskette. The REFIELD program alone takes up a whole mdv cart, and the other three take up all space available in another mdv cart, So, no room for the other October issue programs and utilities. This is an outstanding value at $20.00.

OCTOBER 88 "Hartung Issue"

The HARTUNG issue, on disk or mdv cart, contains Bob Hartung's "QL Address File" program, plus all of the utilities of the October issue which could be stuffed into a mdv cartridge. "Address File" is a simple, quick and easy program to use. Also, it can serve as a Inventory data base. Its printing ability makes loading up to print labels a easy chore. Screen listings and print out can be by alphabetical or by last name. Individual names may be searched out. In his article, Bob gives full instructions and even allows for modification to use the 40 column TV display. This is a good program to study for simple basic programming.

UTILITIES are added to fill up the mdv cart. Some of the utilities that are given are not included in the UPDATE magazine pages. (Just had to quit at 76 pages!). The Hartung Issue comes on either one MDV cartridge or one Diskette.- same price $20.00.





THE TS-2028 Disk Software Library

TS-2068 Disk Drive Systems supported are, Oliger Safe, Larken LKDOS, and Aerco FD-68. Disks are available for these systems on 5.25 inch DS-DD 40 or 80 Track. Some individual programs in disk will be for a specific DOS and will be identified in the listings below. Most of the programs and utilities are those published in UPDATE Magazine issues, but some will be other programs not given in the issues, included to add value. Price is $20.00 per disk (postpaid). Orders for more than one disk may take 10% off total price. Individual issue series are listed as follows:

The October 88 "SMART TEXT" ISSUE

For Larken LKDOS, Oliger DOS, Aerco FD-68 Your scribe decided to donate the Smart Text software to the UPDATE Library. Since the Smart Text manual is 70 pages in length, and the program listing is a quarter of a mile long, it isn't practical to give the documentation in the magazine. Smart Text is the most extensive software ever offered for the TS-2068. It is a Multiple Data Base, and a Disk Integrator, that has a Word processor. Also included is MAIL MERGE. These four main systems are integrated to provide many automatic functions not present in other softwares. For instance, a thousand letters of the same or different text may be printed NON-STOP. Or, one can just type and print some data. Or, one may elect to print one or a thousand labels from the mailing list; a 1 page or a 100 page document, an outline, an inventory listing. Smart Text uses disk as if a whole package of diskettes are extensions of TS-2068 FREE Memory. The Smart Text issue comes on either two 40 track diskettes or one 80 track disk and includes a 70 page manual. $20.00 (postpaid).

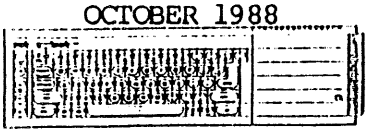
Earlier buyers of Smart Text may want to combine an order for this Newest version, which has column printing option, alternate format for using pre-printed letterhead stationary, elective for Dot Matrix or Daisy Wheel printers, Full Edit during input typing, mail file sorting, and postscript option for letters. Earlier buyers, add only $10.00 to another disk order to receive this version.

The October 88 "HARTUNG" ISSUE"

This is for OLIGER DOS ONLY, and is a collection of 10 "Disk Management" programs presented in four issues of UPDATE Magazine. Bob Hartung started with a early program given by John Oliger and worked for a year expanding to provide a most capable disk management system. Bob named these programs "The DOSDEX" System. This October 88 UPDATE Magazine issue documents the last of this outstanding group of programs. Earlier magazine issues documents the others. This disk issue includes all of the DOSDEX programs on a single disk that will let you index and manage a complete Diskette Library, or to re-organize the individual diskettes, programs, or files, in any manner that you choose. OR- Each of the individual programs may be merged to another program to give that program new capability. This is an outstanding group of programs! $20.00

The Apr/July 88 "MITCHEL DISK" Issue

For Oliger DOS, Larken LKDOS, and Aerco FD-68. Two UPDATE issues were required to present the listings and documentation of Bob Mitchell's excellent HOUSEHOLD BUDGET software, which is included in this issue disk. Also included are all of the utilities given in both the April and July 88 issues. These are; MS-TAS, a utility that prints out either Mscript or Tasman Data files "without need for the host word processor", PEEK VARS utility, some printer code initialization utilities, a Disk Menu that can be converted, and several modified public domain programs. All of these programs and utilities maxes out a 40 track disk. $20.00 (postpaid)





The Oct 87/Jan 88 "UPDATE" Issue

For Oliger DOS, Larken LKDOS, and Aerco FD-68. The programs and utilities of the two first UPDATE issues are consolodated into one issue disk to give an outstanding value. The main Software is:

MAIL MERGE, a data base and printing software to create and print out mailing lists, inventories, and other data files that need arrangement in a format of from one to 7 lines per field. Another data base "Outline" allows "school and Church" type outlines to be produced and printed in indented format for "major topics, and sub topics". Two sorting routines allow sorting by 'first or last' name of the mailing list. Then other nicities, such as "letter heads" may be printed. The print products are: Complete print out of the mailing list, print out of selected fields, print to labels of the entire list or individual fields. All printing may be to your elected TAB position, and formatted to your elected number of lines between fields. Automatic page control is optional so that the printing can be either continuous or under page control.

Two annexes of the Mail Merge program are included:

"Purity", a program that takes a mailing list produced by Mail Merge or Smart Text, and purifies the listing by eliminating duplicate names and skips in the listing. PURITY also allows you to break down the larger listing into smaller groups of mail files that may be processed by the Smart Text software for continuous printing of automatic personalizes form letters.

"LIST LOOKER", is a program that supplements both Mail Merge and Smart Text. LIST LOOKER loads a Mail File, tells you the Dimension of the Character array, lets you screen print the listing, or print the listing to paper. Print out is to your elected format. Also, Label Printing is an option.

ALL THREE of the above programs work automatically in the disk drive environment and supplement each other.

EXTRA MEMORY UTILITIES: This disk contains four utilities given in the January 88 issue to use with DOCK RAM. These utilities reconfigure the Home Bank of memory, MOVES a Basic program from DISK into DOCK BANK RAM, then moves another program from disk into HOME BANK, in order to allow the joint operation of programs in both Home and Dock banks.

STILL MORE! Included are 20 sound and visual utilities to plaggerize and use in your own programs. Dont worry, UPDATE gives these 20 utilities to you for that purpose.

"J-UTILITIES", a group of useful utilities by UPDATE, crunches numbers and disassembles memory. One of these actually uses mixed inputs of Binary, Hex, and Decimal to multiply, divide, add, and subtract, and gives the answers in all three number bases. (dont believe this one has ever been done before).

This great group of software and utilities offered for $20.00? Gotta be kidding, but they are! And, you wont find them anywhere else. They're exclusive.

About The Issue Disk Program

The UPDATE Software Library will continue to build as more issues of the magazine are published and more programmers participate. Half of the proceeds of issue sales go to the authors of these fine programs. The other half is absconded by UPDATE Magazine to pay for the disks, mailing expenses, and the upkeep of the corporation's 300' luxury yacht which plys the seas between Monaco, Aruba, and Nassau on official business. So far we have had nothing but praise and satisfaction expressed by the purchasers of the issue disks, so with some trepidation, and a touch of boldness, we added the comment, "Unconditionally Guaranteed to Please" to the sales pitch for the Issue Disks. Update really wants the issue disk program to be successful because it provides our programmers at least SOME compensation for their work (besides their personal satisfaction).

UPDATE! $4.95 OCTOBER 1988

ComputerSystems

A Quarterly Magazine for the users

OF SINCLAIR COMPUTERS

ADVERTISING RATE SCHEDULE

effective until July 1989

UPDATE MAGAZINE is published quarterly each January, April, July, and October in support of the users and suppliers of the SINCLAIR QL, SINCLAIR Z88, AND SINCLAIR-TIMEX TS-2068.

The following Ad sizes are based upon a page size of 8 1/2 inches by 11 inches. Ads are black and white only. Reduction and enlarging capability exists for ads that are off size. Camera ready copy should arrive at least 15 days before publishing date.

DISPLAY ADVERTISEMENT RATE SCHEDULE

AD SIZE	PRICE	
2 pages	$220.00	
1 page	120.00	
2/3 page	90.00	(vertical or horozontal)
1/2 page	65.00	(vertical or horozontal)
1/4 page	35.00	(vertical or horozontal)
1/8 page	20.00	(vertical or horozontal)

Special placement of full page or double page ads, add 10% and call for reservation (inside front cover, first page, inside back cover, back cover).

Line ads: 78 characters per line across page, 1 line minimum, $6.00 per line. Will be type set to Elite 12 Chrs per line.

Your existing Ad copy for other publications is fine. Just send copy with instructions for sizing (up, down, or same). Enhancement of legibility is usually possible. Draft quality dot matrix print is not so good.

TERMS: Check Payment with order 10% discount, or billing on publication date. (no credit cards).

Send Ad copy to: UPDATE MAGAZINE, 1317 Stratford Ave., Panama City, FL 32404. Tel 904 871 3556 or 904 9871 4513

ALL FD-68 USERS! WE NEED YOUR HELP!

We're trying to find out if you, the AERCO FD-68 users of the world, would be interested in the following item:

ADVANCED USER'S GUIDE FOR THE FD-68 DISK I/F

As we see it, it would be a definitive guide to the AERCO DOS AERCO never gave you. It would include many examples of how to implement the commands, eg.:SAVE, LOAD and MERGE. It would show how to SAVE arrays/data and how to use the extra 64K of RAM that AERCO forgot to explain to you! There would be a complete appendix and all items would be cross- referenced. We also anticipate including a section on the AERCO/ LKDOS cartridge and its operating system and commands.

What we NEED to know is:
1> WOULD THIS BE OF INTEREST TO YOU?
2> WHAT WOULD IT BE WORTH TO YOU?
3> WOULD YOU PAY $14.95 FOR IT?

PLEASE-If you would like to see such a volume put together, let us know! Drop a line or call for more info. If enough interest is shown, IT WILL BE DONE! The writer is a VERY well-known writer of software and TS articles. You won't be disappointed!

RMG ENTERPRISES
1419 1/2 7TH STREET
OREGON CITY, OR 97045 (503)655-7484

12

NOTICE TO AERCO FD-68 USERS!!!

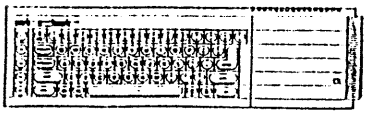
Rod Gowen, RMG Enterprises is breaking a leg trying to support FD-68 users with excellent software and utilities. NOW Rod is trying to decide whether there is sufficient interest in having a detailed manual for the FD-68. He MUST have your interest expressed before spending the money to produce this manual that all FD-68 users need.

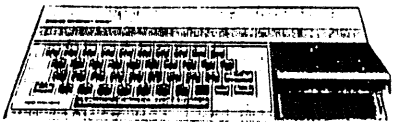
FD-68 WRITER SUPPORT IS NEEDED

UPDATE Magazine ALSO has been beating the bushes TRYING to develop writer support for the FD-68 users, with only small success. We NEED writer support and ask that those users who are doing such things as MODEM DOWN LOAD of CPM programs, and other FD-68 use, CHIP IN your knowledge for the good of all. Aerco FD-68 users need more interchange of information. This page in UPDATE is devoted to stirring up such an interchange of programming tips, DOS extensions, CPM/RPM, use of FD-68 extra memory, supporting utilities, etc.

There are many TS-2068 users who spent a lot of money for the Aerco FD-68 Disk system. We once had a excellent newsletter being published in support FD-68 users. Somehow, the source of information dried up. But, there are several expert programmers still doing work with the FD-68. All need your information published in UPDATE, Time Designs, and Syncware news. Lets all cooperate to re-establish interest in the FD-68 disk system.

The first step is to notify Rod Gowen that you DO want a good manual published. Rod's effort to bring this manual isn't profit motovated, but is for the good interests of the user group. (ed).





ADDITION TO TS-2068 MAIL MERGE (October 1987 UPDATE, OCT87 Issue Disk): The following annex to Mail Merge was added to print out the categories of this INDEX after the Mail Merge program had Sorted the subjects by title. <GO TO 6000> prints the individual columns.

```
6000>INPUT "Nbr of Items ";x1: INPUT "INPUT First TAB ";tb:
INPUT "2nd Tab ";tb2
6002 FOR n=1 TO x1: LET m$=o$(n,1): LET y$=o$(n,2)
6004 IF m$(LEN m$)=" " THEN LET m$=m$( TO LEN m$-1): GO TO 6
004
6006 IF y$(LEN y$)=" " THEN LET y$=y$( TO LEN y$-1): GO TO 6
006
6008 LPRINT TAB tb;m$;TAB tb2;y$
6010 NEXT n: STOP
```

TIP: For all computers that use DIMENSIONED CHARACTER ARRAYS. Is there one that doesn't? The above routine will be explained. The DATA FILE is O$(a,b,c), a three dimensional array. "a" is the number of names in a mailing list. "b" is the number of information lines. "c" is the number of characters reserved for the lines. In the case of the INDEX on the opposite page, we use the first two "b" lines. The first line is the topic and the second line is the page number. We combine these two lines into one for printing. If direct print of the combined lines is done, the "empty spaces in the dimensioned line" would not allow flexible printing to TAB. GO TO 6000 executes the routine.

Line 6000 assigns var x1 to the number of titles to print. The n counter is used to count to x1. The data in line 1 is assigned to M$, and the data in line 2 is assigned to Y$. Line 6004 uses "recursive" programming to trim all empty spaces from line 1. Line 6006 does the same for line 2. In both instances, the line loops itself until the IF THEN condition is false, then drops to the next line. Each loop trims one empty space from the data line. Then line 6008 prints the two data lines on one paper line at tab. Line 6008 is adjustable, and could print the data to whatever two TAB positions desired. Line 6010 <NEXT n> gets the next two lines to process. Approximately one-half second is consumed while processing each two line group.



TRUMP TIP

Kinda hard to read that inset, eh? The manual for the Trump Card is kinda like that. I've got it figgered that Miracle bunch has a printing office in Scotland, and you know how thrifty the Scots are, eh? Wot? Anyhow, the Trump manual is just too teenie weenie to read and kompersend. BUT, ther's lotsa good puddin in the pages that we need to read. In fact iff'n we dont be a readin it we wont be a larn'ng it atall. The answer is to truck it down to a copy shop and blow it up exactly 4 times, so's the pages comes out 8 1/2 X 11 inches. That'll cost ya bout $5.00 that the Scots saved, and then you can be a learning all about the Trump stuff that you wont unless you do that.

PAY ATTENTION! The above was throwed in to fill space which was an afterthought. UPDATE ISN'T CASUAL READING. What we try to do is to give you a THREE MONTH project so that when you are about through with one issue the next one hits your mail box. Sometoimes we mix in bit of Cajun and Nawrth Flurda Cracker talk when we don no no nuther ways ah tellin at it.

Sharp's, Inc.

Rt. 10, Box 459
Mechanicsville, VA 23111
(804) 746-1664 or 730-9697

QL Computer

$149.95 includes PSION s/w

We carry **ALL** hardware and software lines for the Sinclair QL. Compare our prices. VISA and MASTERCARD accepted (3 % surcharge).

WRITE FOR FREE CATALOG