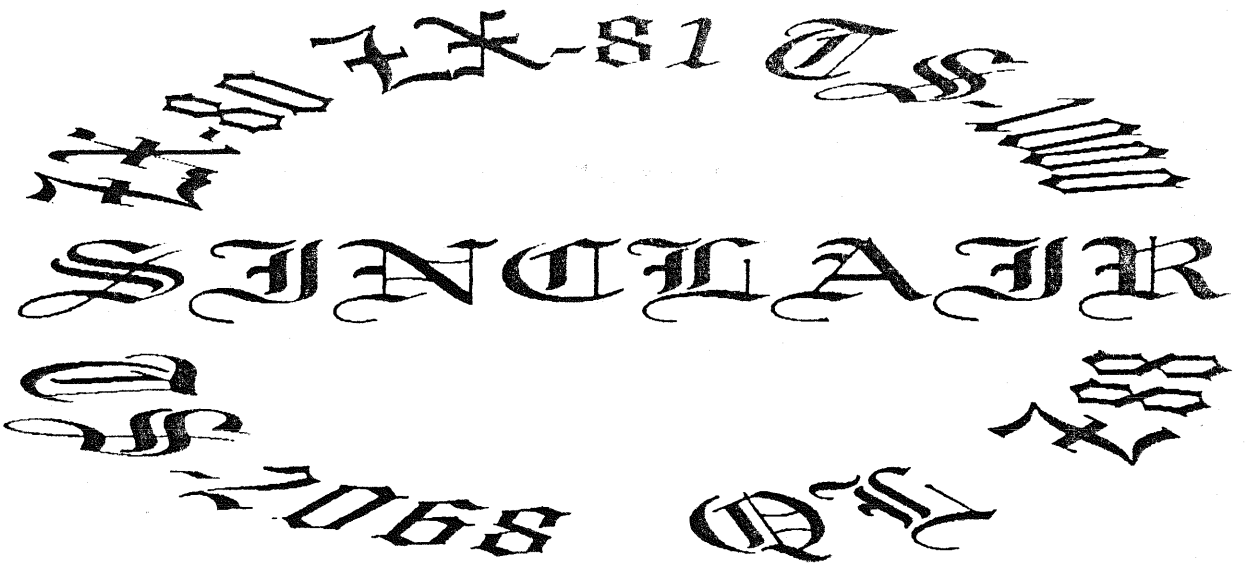
HAVAGE

Magazine



P. O. Box 1095
Peru IN 46970

$4.95/Issue $18/Year

January 1994

* UPDATE COMPUTER SYSTEMS MAGAZINE *

UPDATE COMPUTER SYSTEMS is Edited and Published by Frank and Carol Davis of P.O. Box 1895, Peru, IN 46970 USA. The phone number is 317-473-8831 for both voice and fax, with normal phone hours being between 5 P.M. and 9:30 P.M. Eastern Time during the week and noon to 6 P.M. on weekends. Please use the answering machine if we are not home. Most answers to questions left on the machine will be by mail, long distance charges are too costly for a small magazine.

Mailing date of the magazine: all issues will be mailed out near the 20th of the months of October, January, April and July. All mailings within the USA are by bulk mail and may take a few weeks to reach you at the most. Those wishing to have faster service may pay $4 extra for First Class Mail. The present rate for North America is $18 in US$, and $22 for the rest of the world. Back issues of the magazine are available for $16 per year in North America and $18 elsewhere (it is cheaper to mail a bunch as opposed to one issue at a time). There are four issues to a year, with each year of a subscription starting in October and ending with the July issue, at which time your subscription renewal is due. Timely renewals are what keep us in business!

Assistance in publishing this magazine is provided by you the readers, many of whom have contributed often in the way of reviews and articles. We offer you our heartfelt thanks. Our main assistant locally is longtime friend and colleague, Eliad P. Wannum, Poet and Psychologist, as well as Sinclair computer user. Many thanks to our regulars such as Mike Felerski, Bill Cable, Peter Hale, Paul Holmgren, Al Feng, Don Lambert, Bob Hartung and many others. You are all welcome to submit material for inclusion in the magazine. Please make all hard copy submissions letter or NLQ; no draft print copies, as we do not have much time for re-typing. Send at least two copies hard copy and the article or artwork on disk where possible. No audio tape submissions, as we do not use tape as a media, please. Try to avoid flowery or hard to read fonts...unless you are showing us a sample of the output of a program. If artwork is to be included in the article, please let us know in what order you think it should be displayed.

Those wishing to place ads in UPDATE MAGAZINE: We have two ways of handling ads. ONE, we will do reciprocal ads for other publications (generally on a year for year basis, with you sending us a copy of the issues the ad is placed in). The other way is to purchase ad space from us, with the following rates in effect for now: $15 per quarter page; $25 per half page; and $48 per full page ad. This is per issue. For inclusion in all four issues, you pay for three issues, in advance, and get the fourth ad free. Should you have questions on this please contact Frank Davis, by mail or phone as listed above.

We hope to be of service to you. Thank you!

** UPDATE COMPUTER PAGE DIRECTORY **

The computer that an article concerns is marked by using the following mark at the start of the page: TS = article for TS2068 or Spectrum; QL = article for QL; ZX = article for TS1000, ZX81, TS1500; 88 = article for the Z88.

Front cover art by Abed Kahale of CATUG, the Chicago area user group.

GI-- Page 1: Directory of articles in January Issue
QL-- Page 2: A-DEE-DOO-DAH 1.0 A Front-End For the Unzip Utility by Al Feng
GI-- Page 3: Wanted, For Sale, and Short Newsclips
GI-- Page 4: RMG ad, for Rod Gowen
ZX-- Page 5: Fix the ZX81/TS1000 Loose Jack Problem by Bill Harmer
TS-- Page 6: 2068 Block-Move Error by Bob Hartung
TS-- Page 6: The Ultimate AUTOSTART by Bob Swoger
GI-- Page 8: Mechanical Affinity ad
TS-- Page 9: Code for screen dumps to 24 pin printers on 2068 by Charles Bothner
TS-- Page 14: TS2068 Tape To Disk Projects and Availability by Frank Davis
TS-- Page 15: A Guide to Using "ctas.B1", The Customizer, and the Printer-A Revised Version of Tasword 2 by Larry Crawford
QL-- Page 20: Miracle in Newport, 2nd Time Around
TS-- Page 21: Some Comments on "bitim.B1", A Utility to Create and Print 1 or 22 one-line screen files to a 24-pin or bubble jet printer by Larry Crawford
GI-- Page 24: QZX-ad
GI-- Page 24: Back Issues of Update Special Offer
GI-- Page 25: Modems or Black Magic by Abed Kahale
QL-- Page 26: QBOX (BBS)
GI-- Page 26: Computer Classics-ad
QL-- Page 27: Using the QL by Tony Blizzard
QL-- Page 29: Example Using XChange + MailMerge by Tony Blizzard
QL-- Page 30: Mechanical Affinity-ad
QL-- Page 31: QL Public Domain + Shareware by Paul Holmgren
QL-- Page 33: QLerk - A Review by Hugh Howie
QL-- Page 35: Archive Series Part 15: Entering Query- a Program to Interrogate Any Archive Database by Bill Cable
QL-- Page 38: A-DEE-DOO-DAH 1.0 Listing
88-- Page 41: Correction for Z88 COM_CLI from John Pegram
88-- Page 42: Various Z88 Ads
ZX-- Page 44: ZX-Text Article, plus AFR Software Ad
GI-- Page 47: New Sinclair Newsgroup on USENET by Tim Swenson
88-- Page 50: Z88 Source Book + QL Survivors Source Book

Back Covers -- TS2068 & QL Issue Disks

A-DEE-DOO-DAH 1.0
A FRONT-END FOR THE UNZIP UTILITY

By Al Feng

A month-or-so-ago, I received several ZIP'ed files from Nazir Pashtoon (CATUG) and the UNZIP utility (version 2.00) which had been "ported to Qdos by Erik Slagter" [January 1993]. ZIP'ed files are files which have been compressed by algorithm.

The UNZIP utility -- like its DOS cousin -- is implemented from the "command line" prompt. If you make a MIStake during implementation of UNZIP, you get to see a menu of "options" that you can (should) select from.

If you are like me, you will find the abbreviated "sample" syntax that seem prevalent to be cryptic. This is partially alleviated by Paul Holmgren's addendum in the Unzip_txt that accompanies the program; but, it took me a few tries to realize that Paul had created a SUB-DIRectory on his harddrive ("... where I keep the unzip file").

Being the dense one, it seemed to me that the only way I would be able to "remember" how to unzip a program was if I nested the program's function within another. Hence, A-DEE-DOO-DAH (DooDah_Bas is how I actually SAVE it) is a SuperBASIC front-end to implement the UNZIP program from a single-key select menu.

Because this implementation was "carved" from a version of QLUTter's SuperBASIC source, it has A LOT OF EXTRA BAGGAGE in its program LISTing. The EXTRA BAGGAGE means that there are some end-user conveniences; but, sub-directories are not one of them.

The DooDah_Bas program can be used as the basis for almost any other simple "utility" that you might wish to implement -- the core of the program is in the DEFined PROCedure named "MAIN" (Lines 610 to 710).

Line 700 is the active line. This is a "straight" UNZIP'ing which does NOT implement any of the options -- this took some trial and error on my part, and there is probably a more elegant way to implement it. Note especially that there is a set of double quotes after the semi-colon that separates the command from the filename where you may wish to insert an option.

DATA_USE (Line 140) is a TK2_EXTension the last time I looked. While your disk interface probably has TK2_EXTensions, this is an optional line. Paul Holmgren suggests using RAM1_ as a destination, and I agree as it will expedite the UNZIPping process.

Since I don't have the ZIP portion of the utility (there must be one, right?), this is theoretically only "half" the utility. Still, I think that you will find it worthwhile to INPUT and SAVE if you use UNZIP with any frequency (more so if used only ocassionally).

UNZIP should be in the public domain. Check with someone in your user group.

HAPPY TRAILS, AND COMPUTING, TO YOU ....

WANTED, FOR SALE, AND SHORT NEWSCLIPS

A) Sorry, but no Editorial for this issue, nothing to harp on, or ax to grind.

B) Extra, extra, for all of those who have wished to use a fax modem on your QL, the software is on its way. It is being Beta-tested now by the two authors. One version will be for Class 1 fax modems, and the other for Class 2 fax modems. This will be as shareware or public domain. Stay tuned, or check with us in March or April for more news about this...such as how and where to get it. This will add a whole new dimension to your use of the QL. I highly recommend that if you wish to do this, then you should obtain the Hermes co-processor for your computer from either T F Services or Mechanical Affinity.

C) Pardon the lateness of this issue; it has been the cold and flu season here in Indiana, as well as doing without the assistance of Eliad on this issue. Short handed and feeling out of sorts. Will be more on track next issue.

D) News from Taylor Penrose of Jupiter, Florida is that he is trying to start a QL BBS near the first of this year. Hours of operation will be 0100 to 1400, or as posted. VT100 Terminal Emulation, 300 to 2400 BAUD, No Parity. The number will be 1-800-942-6721. At the moment he would like to limit it to text (if it works out perhaps the users can talk him into adding uploads and downloads). It will be running on his IBM clone, and his 1-800 covers North America and Puerto Rico.

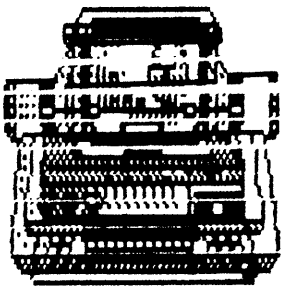
E) AVAILABLE! Have you bought a copy of an Issue Disk from the following authors, perhaps awhile back? Well...we here at UPDATE! have had upgrades on these programs over the last year or so. Some of them are quite significant upgrades! If they are written by Al Feng, Bill Cable, or Bob Hartung, or the QL Compendium, then either return the original disk with $3, or proof of purchase and $4, to either the author or UPDATE! Magazine. It can often pay to have the latest version due to added features as well as any bug correction and interface changes.

F) How about those of you who like to mess around with code and languages on your QL get some contributions off to the QL HACKER'S JOURNAL, in care of its fine editor, Tim Swenson, 5615 Botkins Rd, Huber Heights, Ohio 45424.

G) WANTED: One TS2050 modem for A.F.R. Software, at 1605 Pennsylvania Ave. #204, Miami Beach, Florida 33139, phone 305-531-6464. Help Al out folks.

H) FOR SALE: Complete TS2068 computer system that includes computer with Romswitch, Aerco printer Interface, color Magnavox monitor, Larken disk drive system, disk drives, A & J Microdrive, 2050 modem, software and books. Contact Keith Worrell, with SASE for complete list, at Box 173 RR1, Lakeview Dr., North Hero, VT, 05474. If not sold may be trashed. Sounds like a good system, so how about some of you readers helping Keith out.

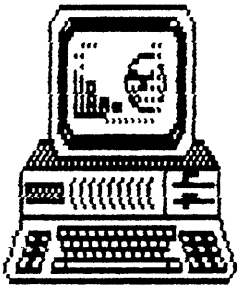
RMG ENTERPRISES



Supports Sinclair/Timex Users!

Send legal S.A.S.E. with request for price sheet. Specify model interest. Send $4 for GIANT GIFT catalog. (Includes ALL price lists) Phone or FAX for information on prices and availability.

Mail to:



14784 South Quail Grove Circle
Oregon City, OR 97045
503/655-7484 * FAX 503/655-4116

Fix the ZX-81/TS1000 Loose Jack Problem

by Bill Harmer

The three jacks on the side of this computer that accept MIC., ear and power plugs are of a somewhat flimsy construction that may become apparent after a few years of use (or earlier on rough usage). The cure is fortunately rather simple for the mechanically/electronically handy user or technician.

Step 1 - Take the bottom half of the case off (without removing the printed circuit board inside or the keyboard membrane plastic sheet at this time) There are 5 screws that must be removed from the bottom of the case. This is not as easy as it might seem as only 2 are visible. The other three are found under the rubber feet (rectangular strips glued on the rear corners). These can be simply pulled off in order to remove the screws under them. Once the screws are completely removed, the bottom half of the plastic case should separate with only light prying. Any force indicates that the screws are not completely removed.

Step 2 - The printed circuit board is now secured to the top half of the plastic case by two screws. See where they are (upper to mid-left near the RAM pack opening and near the center, more towards the middle of the case, typically). For reassembly, note where they came from, with a soft pen, say). The printed circuit board must be moved up and away from the top half of the case about one or two inches only (so as not to disconnect or damage the fragile keyboard matrix/ membrane plastic from the connector on the board).

Step 3 - Holding the printed circuit board so that one can see the three little plastic boxes that house the jacks, any prongs that are bent too far up and out, may be carefully pushed back into position with a small screwdriver or tooth-pick. Several pushes back may be necessary until the springiness is overcome in its attempt to reposition the prong outwards. Now test the jacks with the cable for the cassette recorder (no power applied) for mechanical fit.

Step 4 - Now you are ready to reassemble the computer making sure that the screws for the printed circuit board are not inserted in the holes meant for the screws that hold the outside bottom shell of the case. If you put one or both of the set screws in the wrong place, the screws will not go in when you try to screw the back on again and all the screws on the outside will have to be removed and the bottom taken off again to find the source of the problem. Do not disconnect the little metal bar or whatever; that ground contact with the metallic paint inside the case that provides some measure of protection from RFI for the TV. etc. on VHF models of the computer.

Step 5 - Once the printed circuit board is secured inside the case with its two screws and the bottom of the case secured with its five screws and the little rubber feet or runners stuck back on, you are ready to test the computer to see if the original problem is fixed (and no new ones added). Note that loose power jack into the computer can also cause the save/load crashes and some users have either tightened up that jack too, or replaced it with hardwiring the power supply wire to the P.C.B. and adding some safety device like a switch on the power line or pilot light (standard LED and 1/2 watt, 1K ohm for example) to make sure it is not left on.

2068 BLOCK-MOVE ERROR

-- Bob Hartung --

My face was red when I opened the fall issue of UPDATE and saw I had transposed two digits in the number of bytes in a full-screen display of DFILE1. Anyone who has read Appendix C of the User Manual knows the display file plus the attribute file includes a total of 6912 bytes, as shown in the correction of line 40 below. The original listing will work in black and white but will not transfer colors because it does not address the attribute file. My disk drive system died on me as I was doing the article for UPDATE and so I lost the listing I sent in until it appeared in print. Sorry!

```
40 IF k$="1" THEN LET STP=1638
4: LET INS=INT (STP/256): LET NO
B=6912: LET INB=INT (NOB/256): G
O TO 80
```

The Ultimate AUTOSTART

by Bob Swoger

In the January 1993 Update! Magazine page 20, Peter Hyman gave a good clue to writing a better AUTOSTART program for the LarKen disk system. Like Peter, I have seen AUTOSTARTs that use two tracks. They are usually made from another program on the disk, sometimes called Menu.B1. Menu.B1 is provided to allow changes to the AUTOSTART program and is, therefore, identical to it. Three tracks on the disk now contain the same program!

AUTOSTART programs instruct the computer what to do when the machine is turned ON with the ENTER key held down or when the RAND USR 100: NEW sequence is typed in on either the TS2068 or SPECTRUM systems. Here are a few simple rules:

RULE 1 The ULTIMATE AUTOSTART should do little more than reset RAMTOP, call another program from the disk (LOAD the program and RUN it) and save itself to that disk.

RULE 2 The ULTIMATE AUTOSTART should be 5089 bytes long. Using the CHAMBERS UTILITIES disk, you will find that an AUTOSTART made in both the TS2068 mode and the SPECTRUM mode will always start at address 22490 and contain 5089 bytes.

RULE 3 The ULTIMATE CLEAR number for both TS2068 and SPECTRUM is 27577. One need not execute the PRINT 65536-FREE misprinted in the UPDATE! article to find the lowest point to move RAMTOP, just use 27577 instead. The area saved will always be one track long, 5090-1 bytes.

Since many menu programs are too long to fit on a single track as an AUTOSTART SAVE in the TS2068 mode, (SPECTRUM gives us 2955 bytes more) I submit the following subprogram that could be included in any menu program to create AUTOSTART. Since this is the AUTOSTART program included within the lines of the LogiCall program, it is numbered accordingly:

```
 49 LET H=CODE "d": LET Z=PEEK(
PI+PI)=CODE "1"
 110 GO TO VAL "460-(30*Z)"
 400 CLEAR VAL "65367"
 410 REM RANDOMIZE USR CODE "d"
: OPEN # VAL "4","dd"
 420 RANDOMIZE USR CODE "d": LO
AD "L.B1"
 430 DELETE RND,VAL "399"
 440 DELETE VAL "461",VAL"9999
"
 450 DELETE VAL "421",VAL"459"
 460 CLEAR VAL "27577":
RANDOMIZE USR CODE "f": GO TO V
AL "400"
```

In line 49, H is assigned the value 100 to save RAM space

when subsequent needs for the value 100 are needed. The heavy use of VAL and CODE is to save RAM space and will be explained later in this article.

LET Z = PEEK (PI+PI) = CODE "1" reduces down to LET Z = (PEEK 6 = 49) assigning Z the value 1 if the software finds itself running on the TS2068 and 0 if running on the SPECTRUM. EXPLANATION: If the TS2068 ROM is operating the system, byte 6 hold the value 49. The SPECTRUM ROM holds 203. Therefore, if the TS2068 ROM is operating, the boolean operator (PEEK 6=49) will be true and Z will equal 1. If the SPECTRUM ROM is operating, the boolean operator (PEEK 6=49) will be false and Z will equal 0.

Line 110 bypasses lines 430 to 450 in the Spec'y mode thus avoiding a crash since the SPECTRUM does not have a delete command.

Line 400 sets RAM TOP to a place compatible to fonts and other favorite machine code. You may choose any other number you like up to 65535.

Line 410 is included in case one MUST open stream/channel 4. Since opening stream/channel 4 conflicts with ZEBRA Talkers and eats up precious RAM space, it is REMarked out. Opening stream/channel 4 really has no added programming value in most cases. Just edit REM out if you need to open stream/channel 4.

Line 420 calls the start-up program.
Lines 430 to 450 delete all un-needed lines on the TS2068 from the program which contained this AUTOSTART sub-program thus keeping the required space to one track. This is not required on the SPECTRUM because: 1) not having a DELETE command in its vocabulary it can't delete lines! and 2) SPECTRUM gives us 2955 more bytes of program space on an AUTOSTART save.

Finally, line 460 sets RAM TOP to the ULTIMATE value which will save exactly 5089 bytes, exactly the size of one LarKen track, the AUTOSTART save routine is called, and 400 is given to the AUTOSTART header as the line to auto RUN once this program is loaded.

To use this sub-program, simply merge it into your favorite menu program using the appropriate re-numbering of the statements and cause some keyboard action to send it to the beginning statement of this sub-program. If you use it as a stand alone program, the DELETE statements can probably be removed. When the sub-program is executed, the NMI tone will sound and you will either have to press the 'D' key to continue the save to disk or the 'A' key to interrupt the program to change 'L.B1' to 'Menu.B1' in line 420 if you want to call Menu.B1 instead of LogiCall when you boot the disk. Next type GOTO 460 <ENTER> and 'D' after the tone to save the modified AUTOSTART.

My concern, however, is that you have not yet discovered LogiCall! Many folks who now use LogiCall have discovered that without it on every disk, LKDOS is a 'NAKED' system which requires you to do a lot of typing (i.e. RAND USR 100: GOTO 1: RAND USR 100: LOAD "filenm.B1" as an example of LOADing "filenm.B1" on drive 1). LogiCall creates the ULTIMATE AUTOSTART for you simply by pressing the 'A' key <ENTER> and 'D' at the 'Program?' prompt. Get and use LogiCall!

Because the AUTOSTART programs others have written often use two tracks, I have usually been able to find room on even a full disk to add LogiCall (L.B1) by using LogiCall's embedded AUTOSTART program to call the other person's Menu.B1 program on boot-up and the disk still works the way it did before I added LogiCall!

MECHANICAL AFFINITY

513 EAST MAIN ST.
PERU IN 46970
317 473 8031
FRANK DAVIS

OR

5231 WILTON WOOD CT
INDIANAPOLIS IN 46254
317 291 6002
PAUL HOLMGREN

Bringing the very best to you in SINCLAIR software and hardware from around the world. We want to provide service to you the customer. Need something, give us a call. Make all checks or money orders out to either "Frank Davis" or "Paul Holmgren". Thanks! Payment in US$.

SPECIALS GOOD TILL CURRENT SUPPLY EXHAUSTED

MAGNAVOX RGB MONITORS ---These 80 column monitors contain RGB, Composite video, and can also accept a VCR hookup to play movies on. They have built-in sound, and with the flick of a switch can give you a green mono screen. They are ideal for the QL, Spectrum or TS2068 (even for the ZX81 if you have fixed it for monitor input). They are offered for $125 US$, and this includes shipping.

TS2068 TIMEX-SINCLAIR COMPUTERS---Color computer that works with monitor or with TV, with power supply and monitor, as well as composite cable and tape recorder cables...plus we throw in two TS2068 tape programs as a bonus. All of this for only $50 and this includes shipping. Supply is limited.

TOM BENTS INCREDIBLE REPLACEMENT ROM FOR THE TS1000---The ideal upgrade for your ZX81 or TS1000 to improve readability of some characters, correct bugs, and improve math functions. This requires you to open your case, so only attempt this if you are a tinkerer, or have a friend who is familiar with electronic hardware. We are including one free TS1000 program tape with each ROM kit. This is available for $12.

PRINTER SUPPLIES---For a limited time we have (a) boxes of Olivetti Inkjet Ampules selling at $5 per box, each box contains 4 inkjet cartridges; (b) replacement ribbons for the Okimate 20. The black ribbons are $4, and the color ribbons are $5. At these prices it is a good idea to stock up while you can.

PAPER FOR TS2040 THERMAL PRINTERS---Stock up on paper for this great little dot matrix printer while we still have a good supply. We are offering it for $5 a roll or three rolls for $10. This includes shipping. We also have TS2040 printers for $25 which also includes shipping. These are still great printers for your TS1000, TS2068 or TS1500 and are the best printers to use to do listings.

Frank,

This is an assembly language program to copy T/S 2068 screens to an EPSON compatible 24 pin printer. The program resides in the printer buffer area of system RAM and it can copy screens in either the normal 32 column mode or the expanded 64 column mode. It is presently tailored for the EPSON LQ-510 printer which I am presently using, but the ESC codes should be compatible with virtually all 24 pin printers with EPSON graphics compatibility. The program contains its own printer handler to avoid the necessity of having to patch any other resident handlers to accept non-ASCII codes. My present printer port is an AERCO interface residing at port address 127 (7FH).

An assembly code listing is included. The program operates in the 32 column screen copy mode as listed. To convert to the 64 column mode, it is necessary to POKE 4 locations within the code. These locations are:

ADDRESS	FOR 64 COL. MODE	FOR 32 COL. MODE
23366 (5B46H)	2	3
23474 (5BB2H)	0	24
23475 (5BB3H)	0	15
23545 (5BF9H)	4	3

For the 32 column mode, each of the screen pixels is output in a 3X3 dot expansion to the printer. Since the printer is conditioned for a graphics mode operation of 180 dots/inch vertical and 180 dots/inch horizontal, this results in a display of 60 pixels/inch in both directions. Since there are twice as many bytes output per line in the 64 column mode, the resolution is changed to 3 dots/pixel in the vertical direction and 2 dots/pixel in the horizontal direction. There are a total of 3 ESC code sequences in the program. The first is at address 23534 (5BEEH). This sequences the graphics mode vertical spacing to 180 dots/inch. The second ESC sequence is at 23540 (5BF4H). This sequence sets up the output format for one line of graphics code to 768 vertical columns/ line in the 32 character mode and 1024 vertical columns/ line in the 64 character mode, since there are 512 pixels/ line in this mode. The third ESC sequence starts at 23548 (5BFCH) and this restores the normal printer vertical dimensions after the screen dump is complete. Several spare bytes have been incorporated into each table to allow for patching of these tables if non-EPSON compatible printers are used. The first entry in each table must be the number of characters to be sent to the printer in the ESC sequence.

If a printer interface not using port address 127 (7FH) is used, it will be necessary to patch a total of 3 locations with the correct port address. These addresses

are 23521 (5BE1H), 23528 (5BE8H), and 23530 (5BEAH).

I am supplying you with a disk containing both the 32 column and 64 column versions of the program. You will also find samples of both types of screen copys. The program is entered with a RAND USR 23296 command.

Best regards,

Charles G. (Chuck) Bothner

```
This is a demonstration of the 64 character per line screen dump
 program. The following are filler characters to put more data
on the screen.                    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

"CPYPRT.C1" code for screen dumps to 24 pin printers.

```
5B00 ENTR: E5      PUSH HL       ;save registers
5B01       D5      PUSH DE       ;
5B02       C5      PUSH BC
5B03       CD225B  CALL OUTP     ;set up vert.spacing on printer
5B06       210040  LD HL,4000   ;start addr. of disp.block 0
5B09       CD365B  CALL OUTB     ;output 1 block of display data
5B0C       210048  LD HL,4800   ;start addr. of block 1
5B0F       CD365B  CALL OUTB     ;output block 1 data
5B12       210050  LD HL,5000   ;start addr. of block 2
5B15       CD365B  CALL OUTB     ;output last screen block
5B18       21FC5B  LD HL,REST    ;set up for normal printer space
5B1B       CD255B  CALL AA       ;output the ESC sequence
5B1E       C1      POP BC        ;restore regs.
5B1F       D1      POP DE
5B20       E1      POP HL
5B21       C9      RET           ;done
-----------------------------------------------------------------
5B22 OUTP: 21EE5B  LD HL,SETV    ;addr.of printer vert.spc.table
5B25   AA: C5      PUSH BC       ;save regs.
5B26       3E0A    LD A,0A       ;send LF for new line
5B28       CDD65B  CALL OTPR     ;
5B2B       7E      LD A,(HL)     ;get # of chars.to be output
5B2C       47      LD B,A        ;store in B reg.
5B2D   BB: 23      INC HL        ;point to next character
5B2E       7E      LD A,(HL)     ;get next char.
5B2F       CDD65B  CALL OTPR     ;send it to printer
5B32       10F9    DJNZ, BB      ;continue until done
5B34       C1      POP BC        ;restore regs.
5B35       C9      RET           ;done
-----------------------------------------------------------------
5B36 OUTB: 0608    LD B,08       ;8 lines/block
5B38   CC: 0E20    LD C,20       ;32 chars./line
5B3A       E5      PUSH HL       ;save the address
5B3B       21F45B  LD HL,OTFM    ;graphics setup for 1 line
5B3E       CD255B  CALL AA       ;output the ESC sequence
5B41       E1      POP HL        ;restore address
5B42   DD: C5      PUSH BC       ;save char.& line counters
5B43   EE: 1680    LD D,80       ;mask bit for horiz.scan byte
5B45   FF: 3E03    LD A,03       ;3 horiz.bits per pixel
5B47   GG: F5      PUSH AF       ;save the count
5B48       1E00    LD E,00       ;clear the storage reg.
5B4A       E5      PUSH HL       ;save the address
5B4B       7E      LD A,(HL)     ;get the byte & test pixel
5B4C       A2      AND A,D       ;       for ink or paper
5B4D       2802    JR Z,HH       ;jump if its paper
5B4F       1EE0    LD E,E0       ;if ink,set 3 upper bits
5B51   HH: CDCF5B  CALL GTNX     ;get next line of char.
5B54       2804    JR Z,II       ;jump if pixel=0
5B56       7B      LD A,E        ;get the partial data byte
5B57       C61C    ADD A,1C      ;add next 3 bits
5B59       5F      LD E,A        ;store the partial result
5B5A   II: CDCF5B  CALL GTNX     ;get next line of char.
5B5D       2804    JR Z,JJ       ;jump if pixel=0
5B5F       7B      LD A,E        ;get partial data byte
```

"CPYPRT.C1" code for screen dumps to 24 pin printers (Cont.)

```
5B60        C603    ADD A,03      ;set the last 2 bits
5B62        5F      LD E,A        ;store completed data byte
5B63   JJ:  7B      LD A,E        ;get the first of 3 bytes/col.
5B64        CDD65B  CALL OTPR     ;sent it to printer
5B67        CB43    BIT 0,E       ;see if LSB was set on last byte
5B69        1E00    LD E,00       ;clear the storage register
5B6B        2802    JR Z,KK       ;jump if the LSB=0
5B6D        1E80    LD E,80       ;set MSB in next byte
5B6F   KK:  CDCF5B  CALL GTNX     ;get next line of character
5B72        2804    JR Z,LL       ;jump if the pixel=0
5B74        7B      LD A,E        ;get the partial data byte
5B75        C670    ADD A,70      ;set the next 3 bits
5B77        5F      LD E,A        ;store partial data byte
5B78   LL:  CDCF5B  CALL GTNX     ;get next line of character
5B7B        2804    JR Z,MM       ;jump if pixel=0
5B7D        7B      LD A,E        ;get partial data byte
5B7E        C60E    ADD A,0E      ;set the next 3 bits
5B80        5F      LD E,A        ;store partial data byte
5B81   MM:  CDCF5B  CALL GTNX     ;get next line of character
5B84        2801    JR Z,NN       ;jump if pixel=0
5B86        1C      INC E         ;set LSB otherwise
5B87   NN:  7B      LD A,E        ;get completed data byte
5B88        CDD65B  CALL OTPR     ;send it to printer
5B8B        CB43    BIT 0,E       ;test LSB of last byte
5B8D        1E00    LD E,00       ;clear the storage register
5B8F        2802    JR Z,OO       ;jump if the LSB=0
5B91        1EC0    LD E,C0       ;set 2 highest bits otherwise
5B93   OO:  CDCF5B  CALL GTNX     ;get next line of character
5B96        2804    JR Z,PP       ;jump if pixel=0
5B98        7B      LD A,E        ;store partial data byte
5B99        C638    ADD A,38      ;set next 3 bits
5B9B        5F      LD E,A        ;store the byte again
5B9C   PP:  CDCF5B  CALL GTNX     ;get last line of character
5B9F        2804    JR Z,QQ       ;jump if pixel=0
5BA1        7B      LD A,E        ;get partial data byte
5BA2        C607    ADD A,07      ;set the last 3 bits
5BA4        5F      LD E,A        ;store completed data byte
5BA5   QQ:  7B      LD A,E        ;get completed byte
5BA6        CDD65B  CALL OTPR     ;sent it to printer
5BA9        E1      POP HL        ;restore original disp.file addr
5BAA        F1      POP AF        ;restore bit counter
5BAB        3D      DEC A         ;decrement counter
5BAC        2099    JRNZ,GG       ;send 3 sets of same bytes
5BAE        CB3A    SRL D         ;shift bit mask
5BB0        3093    JR NC,FF      ;cont.until char. is complete
5BB2        180F    JR,SS         ;jump if 32 character mode
5BB4        7C      LD A,H        ;get upper byte of address
5BB5        FE5A    CP A,5A       ;test location of address
5BB7        3006    JR NC,RR      ;jump if not chunk 2
5BB9        3E20    LD A,20       ;offset to 2nd display file
5BBB        84      ADD A,H       ;point to addr. in chunk 3
5BBC        67      LD H,A        ;
5BBD        1884    JR,EE         ;dump same char. pos.from file 2
-------------------------------------------------------------------
5BBF   RR:  7C      LD A,H        ;get high byte of address
```

"CPYPRT.C1" code for screen dumps to 24 pin printers (Cont.)

```
5BC0        D620    SUB A,20      ;convert back to chunk 2 addr.
5BC2        67      LD H,A        ;
5BC3    SS: 23      INC HL        ;bump character address
5BC4        C1      POP BC        ;get line & char. counters
5BC5        0D      DEC C         ;decrement character count
5BC6        C2425B  JP NZ,DD      ;cont. until done with 32 or 64
5BC9        1001    DJNZ,TT       ;cont. for all lines in block
5BCB        C9      RET           ;done with display block
-------------------------------------------------------------------
5BCC    TT:C3385B   JP CC         ;do next line in block
-------------------------------------------------------------------
5BCF GTNX:  010001  LD BC,0100    ;256 byte offset to next line
5BD2        09      ADD HL,BC     ;update address pointer
5BD3        7E      LD A,(HL)     ;get the pixel byte
5BD4        A2      AND A,D       ;test designated pixel
5BD5        C9      RET           ;return with Z bit status
-------------------------------------------------------------------
5BD6 OTPR:  F5      PUSH AF       ;save registers
5BD7        C5      PUSH BC       ;
5BD8        4F      LD C,A        ;save the char. in C reg.
5BD9    UU: CD0920  CALL 2009     ;test for BREAK
5BDC        3802    JR C,VV       ;jump if no BREAK
5BDE        CF      RST 08        ;return with BREAK message
5BDF        14      DEFB          ;
-------------------------------------------------------------------
5BE0    VV: DB7F    IN A,(7F)     ;read printer port
5BE2        CB67    BIT 4,A       ;test BUSY bit
5BE4        20F3    JR NZ,UU      ;loop back if BUSY
5BE6        79      LD A,C        ;get character back
5BE7        D37F    OUT (7F),A    ;send it to printer
5BE9        DB7F    IN A,(7F)     ;reset AERCO port
5BEB        C1      POP BC        ;restore registers
5BEC        F1      POP AF        ;
5BED        C9      RET           ;done
-------------------------------------------------------------------
5BEE SETV:  03      DEFB          ;3 chars. to be output
5BEF        1B      DEFB          ;ESC char.
5BF0        2B      DEFB          ;"+"set graphics mode
5BF1        30      DEFB          ;  to 180 dots/inch vertical
5BF2        FF      DEFB          ;spare bytes
5BF3        FF      DEFB          ;
-------------------------------------------------------------------
5BF4 OTFM:  05      DEFB          ;5 chars. to be output
5BF5        1B      DEFB          ;ESC char.
5BF6        2A      DEFB          ;"*"set for 180 dots/inch horiz.
5BF7        27      DEFB          ; 768 vert.columns per line
5BF8        00      DEFB          ;lo byte
5BF9        03      DEFB          ;hi byte
5BFA        FF      DEFB          ;spare bytes
5BFB        FF      DEFB
-------------------------------------------------------------------
5BFC REST:  02      DEFB          ;2 chars. to be output
5BFD        1B      DEFB          ;ESC char.
5BFE        32      DEFB          ;restore normal vert. mode
5BFF        FF      DEFB          ;spare byte
```

TS2068 TAPE TO DISK PROJECTS AND AVAILABLILITY

Over the years those of us who have steadfastly used our TS2068s have contended with the fact that many of our old favorite programs that came on tape only were never upgraded to disk for us to use on our favorite disk interface. We have had some great disk drive interfaces over the years, ranging from Aerco, Oliger, Larken, TOS and the Millenia K to name a few. Quite a few of us got to where we could convert most programs to disk. Some have preferred to use canned disk programs and never learned to do this conversion, and never had the time or inclination to type in the changes to make these changes when they were given a listing. I hope to be able to remedy this for you.

Here at UPDATE! Magazine we have over the years received the benefit of many of our best TS2068 programmers, and with their kind indulgence, I would like to pass on some of these software conversions to the rest of you. I have practically given up the use of audio tape myself years ago (it was great when that was all I had, but that was then!) and acquired two Oliger, two Larken and a couple of Aerco interfaces to use with my TS2068s. I want to do this in a legal manner. The reason I say this is that some folks have been handing out pirated copies of some software. This may not seem like a big deal on an orphan computer, but some of these companies still object. An example is TASWORD 2; it is still for sale by TASMAN (though not easy to get a legal copy of here in the Americas) and they still object to it being passed around for free. In this issue we have shown you the excellent updates and upgrades to this program by Larry Crawford. You may find these changes available in your local user group or club library, but for those who do not have or use this source I am making the following offers to you on this and a few other items of software.

For those who want to upgrade their tape copies of TASWORD 2, or PRINT FACTORY to either Larken or Oliger disk, or want a disk copy of 'bitim.B1' for use on your 24-pin printer or Bubblejet, send me $5 per program to cover disks, postage and packing materials and copy charge, and I will send them to you either First Class or Priority Mail. I have too many calls upon my time and finances to offer to do this for free, but I feel this is a fair offer. If you ordered, paid for, but never received Print Factory, phone or write me and I will try to help you out in that area also.

I hope this will be of help to you. To the best of my knowledge, the upgrades for MSCRIPT are still being offered by Jack Dohaney. Has anyone heard otherwise? If anyone else has tape to disk conversions done for other TS2068 programs that they would like to see made available to the public for legal ownership, please contact me. As for those ordering from me, since most of us do not have our receipts from years ago to prove we purchased these programs, I will rely upon your honesty. If you want to make a legal purchase of some of these older programs, then contact either RMG or Mechanical Affinity.

A GUIDE TO USING "ctas.B1", THE CUSTOMIZER, AND THE PRINTER

Larry Crawford / 357 Reynolds Rd / London Ont / Canada / N6K 2P8
(519) 657-9119 Public Domain 10 Oct 93

SOME FEATURES OF THIS EXTENSIVELY REVISED VERSION OF TASWORD 2:

1. Can be used with an Oliger system.
2. Can be used with a LARKEN system.
3. Can be used by a combined Oliger/LARKEN system.
4. Color coded screen prompts indicate which system is active.
5. Files can be loaded from one system and saved to the other.
6. When a text file is to be Loaded, Added,or Saved, a catalog of all text files on the active drive is displayed.
7. One or 2 keystrokes can then perform the desired operation.
8. The Print option has its own menu which has 12 options. Mew ones are:
    a) change size of top margin
    b) change size of bottom margin
    c) pause at end of each page for manual positioning of the paper or automatic positioning
    d) number of copies to be printed
    e) print number of page at top or bottom or not
    f) print a header or footer message or not.
    g) select Oliger or LARKEN DOS
9. A companion Customizing program allows you to define the graphics, printer, and interface codes as in the original Tasword. Additional Customizing features have been added:
    a) edit the help pages (fast)
    b) set color of text file paper, ink, & left margin
    c) set the default values for the printout:
        i) size of the top margin
        ii) size of the bottom margin
        iii) size of the left margin
        iv) the length of the form in lines
        v) clear text file or not when ctas.B1 is loaded
        vi) pause or not at end of page
        vii) select fanfold or cut sheet paper operation
10. Compress or uncompress a text file.
11. Save the whole Tasword package to another drive except for the booting program which must be handled separately.
12. Change from one DOS to the other.
13. Change from disk to tape operation.
14. The 1st 6 bytes of the text file can be used to define a default file name. This can make the saving of a text file a two-keystroke operation.
15. The whole 1st line can be used for header or footer data.
16. The main menu displays the following extra information:
    a) name of the file in the text buffer
    b) length of file in bytes
    c) length of file in lines
    d) length of file in words
    e) number of tracks needed to save the file
    f) active drive number (flashing)
17. The active drive can be changed by simply pressing a # key. Drive #4 is supported if in LARKEN mode.
18. The drive used to load the program is defined as the source

drive. The same number is used to define the default # of the data (destination) drive. When some of the options are chosen, the source drive is activated so that the appropriate program or codes can be loaded.

19. The data drive, as selected by the main menu, is used to load or save text files.
20. Whenever the program is loading or saving anything, a message is displayed to inform you of what's going on.
21. A single beep sounds when a keyboard input is expected.
22. A double beep sounds if an invalid value has been typed in. Control then goes back to the appropriate menu.

NOTE: If you have broken into the program, <GOTO V> will get you back to the main menu. If you have broken into the printing process, type <RANDOMIZE USR 59806> before <GOTO v>.

****

****
** MAIN MENU OPTIONS **
****

J: LOAD text file: The text file is cleared out and a catalog of the Tasword text files on the disk in the active drive is displayed. Each file name is identified with a letter. The file is selected by this identifier.

You can escape to the main menu by pressing the <STOP> key.

If more than 26 files are on the disk, you will be prompted to press the space bar to scroll to the next page or the UP cursor (key 7) to return to page 1 of the catalog.

A: ADD text file: The same comments apply as for the LOAD option except that the file buffer is not cleared and the desired file is added to the end of the existing text file rather than being loaded into an empty file buffer.

S: SAVE text file: When the SAVE option is selected, the same catalog of all text files is displayed as in the load option but without the identifying letter. This will help to avoid wiping out an existing file by saving a new one with the same name. You will be prompted to press <ENTER> to continue with the save. After this is done, the default name is displayed and is the name under which the file will be saved unless changed from the keyboard.

P: PRINT text file: The ten options and their default values are then displayed on the screen as is the type of paper feed. You are prompted to press a letter key to change any of the default values. Otherwise, hit the <P> key to LPRINT with the displayed parameters. If the printer is not on line at this time, a prompt will remind you to turn it on. The <M> key will escape to the main menu.

If page 1 is to be printed, you will be prompted to set the paper to the top of the page. This will be the position of the first line of print and will be the line immediately following the top margin spacing that you want. This is critical since all vertical spacing is determined from this position. Most printers automatically feed to line 5 from the top, giving a 5/8" top margin which is usually acceptable.

If the Page Pause option is active, set the first page as above

but set all subsequent pages to print on the 3rd line from the perforation.

If the page pause feature is not active and the top margin on subsequent pages is incorrect, change the value of the upper limit of the loop counter, 'i', in line 8050 from d [4] to an appropriate value. (I have a bubble jet printer now and have no way of checking the operation with fanfold paper)

If you have just printed a file and attempt another print, you may get an "OUT OF MEMORY" error message. If so, simply <CLEAR> and <RUN>.

==========

****
THE PRINTER OPTIONS
****

A: Start line: Always initialized to 1. Can be changed if the <A> key is pressed. Automatically changed to line 2 if a header or footer is to be printed and printing is to begin with the first line of text.

B: End line: Always initialized to the last one in the file and identified as "last". Can be changed if the <B> key is pressed. If it has been changed and you want to revert to "last", press <B> again and then <ENTER>.

C: Line spacing: Always intialized to one. Can be changed if the <C> key is pressed.

D: Left margin: The permanent default value can be set with the <B> option of the Customizer program. A new temporary value can be entered if the <D> key is pressed.

E: Top margin. The permanent default value can be set with the <A> option of the Customizer program. A new temporary value can be entered if the <E> key is pressed.

F: Btm margin: The default value is displayed. This can be set with the <B> option of the Customizer. A new temporary value can be entered if the <F> key is pressed.

The maximum number of lines/page is displayed on the screen.

G: Page Pause?: The default status is displayed. This can be set with the <M> option of the Customizer. The "YES" status will require the operator to position the paper manually before the next sheet is printed. Toggled with key <G>.

H: # of copies: Always initialized to one. Can be changed if the <H> key is pressed.

I: Number pages?: Default value is "NO". When key <I> is pressed repeatedly, the mode cycles through "YES" (top), "YES" (btm), and "NO".

J: Header/Footer?: Default value is "NO". If the <J> key is pressed repeatedly, cycles through "hdr", "ftr", and "NO".

If active, the first line of the text file is used to hold the header or footer message and the start line is set to 2 if

printing is to start at the first line.

-Main Menu options continued-

Y: RETURN to text file: Just what it says.

T: SAVE BASIC only: Allows you to save the version that you have created for your own needs without disturbing the codes.

W: Save Whole Tasword package: THIS OPTION CAN BE USED ONLY IF THE TWO DISKS ARE FORMATTED FOR THE SAME DOS.
Clears out the text file and saves "ctas.B1" to the current disk. Loads the Customizer program which loads the "taspak.C1" CODE from the source drive and saves it, "ctas.C1" CODE, and "custtw.B1" to the destination drive.
Reloads "ctas.B1" from the source drive.

At this point, if you want to put the LARKEN AUTOSTART on the new disk, reboot the source drive and break in when the prompt to switch off the Oliger board appears. Then GOTO 9999. You will be prompted to enter the destination drive number then to press the <D> key to save the AUTOSTART.
With Oliger disks, boot the source drive with <LOAD> and break in before "ctas.B1" starts to load. The timing here is kind of tricky. When the loader program on track 0 is successfully captured, select the destination drive and then <SAVE /0>

C: Customize Gr/Prntr/Help pages: Loads the Customizer, "cust.B1", from the source drive and its menu displayed:

****
CUSTOMIZER OPTIONS
****

G: Define Graphic codes: As in Tasword 2.

E: EDIT help pages: You will be prompted to select the Help Page to be amended. Almost instantly, the selected Help page will be displayed, ready to be amended.
Upon returning to the Customizer menu with the <STOP> key, a prompt will inform you that the amended page code has been put into high memory and that you should choose the <S> option after all other desired customizing changes have been made.

P: Define Printer/interface codes : As in Tasword 2.

S: SAVE new code and reload Taswrd: Saves "ctas.C1" CODE to the source drive and reloads "ctas.B1" from it.

T: Save this customizing program: Saves "custtw.B1" to the active drive.

Y: Return to Tasword: Reloads "ctas.B1" without saving the codes.

C: Change Colour of paper and ink: Permits the colours used in the text file display to be changed. The formula for calculating the colour codes is displayed on the screen.

O: OLIGER/LARKEN: Toggles between the two DOSes. This will permit the user to save an amended version of the Customizer to either type of disk.

Z: Fanfold paper?: Toggles between the two types of paper.

PRINTOUT DEFAULT VALUES

A: Top margin. When chosen, the current default value is displayed flashing. It can be changed by entering a new value or retained by pressing <ENTER>.

B: Bottom margin: When chosen, the current default value is displayed flashing. It can be changed by entering a new value or retained by pressing <ENTER>.

L: Left margin: When chosen, the current default value is displayed flashing. It can be changed by entering a new value or retained by pressing <ENTER>.

F: Form length in ": When chosen, the current default value is displayed flashing. It can be changed by entering a new value or retained by pressing <ENTER>. The standard value for an 11" page is 66 lines.

X: Clear text file on Load?: The current status is displayed. It can be toggled between "YES" and "NO" with the <X> key.

M: Page Pause?: The current status is displayed. It can be toggled between "YES" and "NO" with the <M> key. The "NO" status requires you position the paper manually at the top of the next page. It is needed if you are using multi-copy forms.

++

NOTE: If you have broken into the program, <RUN> will get you back to the customizer menu.

++

-Main Menu options continued-

K: Compress/uncomp text file: You are given the option of saving the current text file with the <S> key (in case you have just created it and haven't saved it). Any other key will activate the "CAT text files" function.

After the name of the file has been found by pressing the file identifier letter key, the Customizer is loaded as is the "taspak.C1" code. A menu is displayed offering the two options.

After the choice has been made and the compression or uncompression is complete, the name of the old file is displayed and you are prompted to enter a name for the new file. When this is done, the file is saved to the active drive and both "ctas.C1" and "ctas.B1" are reloaded from the source drive with

the new file in place.

O: OLIGER/LARKEN: Toggles between the two systems. This allows you to load a text file fron one disk type and save it to the other. In fact, all options except <W> can be used.

If OLIGER is active, all screen prompts will be in green. If LARKEN is active, they will be red.

A prompt will remind you to change drives after changing DOS.

D: Disk or tape LOAD/SAVE of Tasword file: This option is strictly for the handling of Tasword text files. It toggles between the two storage devices. The active device is highlighted on the main menu screen.

==========

MIRACLE IN NEWPORT, 2ND TIME AROUND

On Saturday, May 14th of 1994, IQLR (International QL Report) will be sponsoring the second annual North American QL/QDOS get together. It will once again be held at the Salvation Army Building on Memorial Boulevard in Newport, Rhode Island to the best of our present knowledge. The vast majority of those who attended last year really enjoyed this show.

***UPDATE! Magazine** and **Mechanical Affinity** were both there last year and will be there for this event! We here at the magazine hope to see all of our QL users and readers at this years event. There is more than just hype in the last sentence, because we happen to really like the people we have as subscribers!, and enjoy meeting them at computer shows. The only complaint we had about the show last year was that there were too few hours and we here in the Midwest prefer to relax and take our time. From what Bob Dyl has told me and published in IQLR, we will have a longer show this year. Come and see old friends and meet new QL users. This may be your best way to see what is new for the QL.*

*Vendors who have said they will be there, so far, are: **IQLR, Mechanical Affinity, Miracle Systems Ltd, UPDATE! MAGAZINE. QBOX-USA will be there demoing their QL supported and QL operated BBS. Bob Gilder of LIST will be there demoing the CL Systems "Real Time Digitizer". I have spoken with Dilwyn Jones of Dilwyn Jones Computing and he has put in to take off time from his regular job at the BBC. Freddy Vachha of Digital Precision is trying to make the time to attend this years show. We have also been informed that W.N. Richardson & Co. will be there.***

For more information about vendor or user group tables, or to get your tickets early to attend (save a few bucks), call Bob Dyl at (USA) 401-849 3805 for either phone or FAX. I want to see all of you there. Before going last year, Carol and I had never thought of Rhode Island as a vacation spot, but we sure changed our minds. If you love sea food and jazz, this is a great spot to be, and not far from Cape Cod and the islands.

SOME COMMENTS ON "bitim.B1", A UTILITY TO CREATE AND PRINT 1 OR 22 ONE-LINE SCREEN FILES TO A 24-PIN OR BUBBLE JET PRINTER

Larry Crawford / 357 Reynolds Rd / London Ont / Canada / N6K 2P8
(519) 657-9119 Public Domain 10 Oct 93

==========

For background information on 24-pin operation see the Tasword file "1bibkg.CT" or choose option [C] of the disk menu.

==========

When loaded, a prompt will ask you to enter the name of the screen file. This demo disk contains "mag.C1" and "alb.C1".

It will probably be prudent to have only one file on a disk with its name on the label so that you will know what to type in at this point.

Since the screen name must be in variable 's$', never RUN the program after you have broken into it. Use <GO TO 1> instead. Should you inadvertently use <RUN> then GO TO <8888> after the error message and reenter the screen file name.

After you have entered the screen name, a menu will appear, offering a choice of 13 options:

BIT-IMAGE MENU

A: CREATE 1 FILE FROM SCREEN
B: CREATE 22 FILES FROM SCREEN

C: PRINT 1 FILE IN SINGLE DENSITY
D: PRINT 1 IN DOUBLE DENSITY
E: PRINT 1 IN TRIPLE DENSITY
F: PRINT 1 IN HEX DENSITY

G: PRINT 22 FILES IN SINGLE DENSITY
H: PRINT 22 IN DOUBLE DENSITY
I: PRINT 22 IN TRIPLE DENSITY
J: PRINT 22 IN HEX DENSITY

K: PRINT 1 FILE TO 2040 PRINTER
L: PIRNT 22 FILES TO 2040

M: DISK MENU

Six of these deal with one line file: you may create a file whose name will be the first 3 characters of the screen file name plus "1.C1".

Yuu may print it to the large printer in one of four densities or the 2040 printer. The 2040 does not produce good quality printouts, of course, nor are circles round. It does provide a quick and easy way of checking the contents of the line file, especilly to see if it is inverse.

The others deal with a full screen: you may create 22 line files

whose names will be first 3 characters of the sceen file name plus "n.C1" where n is the number of the line file. A message at the bottom of the screen inform you which file is being created or saved. The process of dealing with all 22 files is about 6 minutes.

You may print them to a large printer or the 2068. All 22 files will load and print in about 45 seconds.

When any of these options is selected, you will be prompted to choose between a normal or inverse file. Remember that a normal screen image is actually a negative so that if you want to print out a positive bit image copy, you will probably want to choose the inverse option.

The final option is to return to the disk menu.

==========

SOME COMMENTS ON "bitim.B1"

LINE 10: Defines the BEEP subroutine 'bp'. Assigns to 'h$' the first three characters of 's$', the name of the screen file plus ".". Sets the BORDER, PAPER, and INK colors. Sets the keyboard input to lower case.

LINE 20: The menu.

LINE 30: Inputs and error traps the menu choice.

LINE 35: Reboots the AUTOSTART if the <M> option was selected.

LINE 40: Prompts for an INVERSE? creation or printing of the file(s).

LINE 60: Sets the number of line files to be created. Transfers control to line 140 to create the file(s).

LINE 70: If the large printer is to be used, loads the code routine. Determines the number of lines to be handled.

LINE 80: Waits for the printer to be put on line before attempting to print. NOTE: the test value used here is 236. Change it if necessary to conform to your printer interface.

LINE 90: Defines the address that the code routine uses to determine the number of times a bit image DATA byte is to be sent to the printer (LET den=24394). Sets the left margin and Line Feed spacing.

LINES 100 to 140: Set the parameters for the density of the large printer bit image printout.

LINE 150: Determines the number of lines to be sent to the 2040 printer and jumps to 5000 to do so.

LINE 160: Displays the screen file if all 22 line files are to be printed. This is not needed but lets the user know what

output to expect.

LINE 170: Sets up a loop to send 'nf' lines to the printer. Sets the address for the start of each line's data to 26296 (5B00).

LINE 180: Defines the name of the line file and loads it into the 2068 printer buffer.

LINE 190: Sends codes to the printer to tell it how many bytes of bit image data to expect.

LINE 200: Calls the code routine at 24331 to send the data to the printer (see Tasword file "bidis.CT" for a disassembly of this routine). Terminates the loop. Returns to menu after all files have been sent.

LINES 500 to 530: Subroutine to send codes to the printer. NOTE: the test value of 236 in line 520 may need to be changed.

LINE 4000: The start of the "create-a-file" routine. This first line loads the screen file, initializes the line file number to zero, and defines the address of the first data byte as the beginning of the printer buffer.

LINE 4005: Sets up an 's' loop to look after the 3 sections of the upper screen. Sets up an "l" loop to take care of the 8 print lines in each of the top two sections and the 6 lines in the bottom one. Increments the line file counter and tests it to see if all the desired lines have been created. If so, control passes back to the menu.

LINE 4010: Assigns to 'f$' the line file name. This will be the first three characters of the screen file name plus the line number plus ".C1".
Initializes the pixel column counter, 'k', to -1. This will be incremented 256 times for each of the print lines.
Sets up the scan line loop, 'i', which points to the first to last pixel scan lines in steps of 256.
Sets up the print column loop, 'j', to point to each of the 32 columns in steps of one.
Assigns to the data byte, 'b', the contents of the screen file pointed to by 'i' and 'j'.

LINE 4015: If an inverse file was requested, address 26671 will contain a value of 1 and the data byte will be subtracted from 255 to change all its zeros to ones and vice versa.

LINE 4020: Pokes the byte into the printer buffer and repeats the process for the next column and scan line.
Saves the resulting 256 bytes.

LINE 4030: Repeats for each of the 8 print lines in the 3 thirds of the screen except for the last one: line 4005 returns to the menu after the sixth print line of the bottom third.

LINES 5000 to 5030: The "send-to-2040" subroutine. Lprints the required number of bit image files after converting them to

inverse if requested.

LINE 7000: The BEEP subroutine.

LINE 8888: The entry point when the program is first loaded. Prompts for the name of the screen file to be used, then passes control to line 1 to initialize variables and call up the menu.

LINE 9999: Saves the program to run at line 8888.

==========



The Journal Covering Amateur Radio and Sinclair Computers
ZX80, Micro-Ace; ZX81; TS1000, 1500, and 2068; QL; Z88
Alex. F. Burr, K5XY, Publisher
2025 O'Donnell Drive
Las Cruces, NM 88001
1993 July 7

QZX is a monthly newsletter publishing articles of a technical nature of interest to Radio Amateurs and other technically oriented people from all over the United States and some overseas countries. The articles relate to any of the Sinclair computers. The US subscription rate is $15.00 per year. For a sample issue please send $1.00 (postpaid).

BACK ISSUES OF UPDATE SPECIAL OFFER

In order to reduce our present inventory of back issues of UPDATE! Magazine we are offering the following through the end of March, 1994. The normal price for a four year subscription is $18.00. Despite the rising cost of mailing, paper, photocopy toner, etc., we have kept the price the same for going on 4 years. Prior to our taking over UPDATE!, the magazine was quite ably published by Bill Jones for three years. This gives us quite a treasure trove of back issues to offer to those who have missed out on them. Here is the deal: We are able to offer back issues on all but two of the early ones; this means (not counting this issue year which is still covered under current subscriptions and prices) we can offer complete sets of 22 issues and at the great price of only $70. This would (normally under our usual back issue price of $16 for four issues) have been $88. You get to save $18, and we get some more room around the house! For those not quite so adventuresome who only want four back issues at a time, we are offering to do so for only $14, but only through the end of March 1994. When an issue is gone, we will not be making any more copies of it, so I urge you to take advantage of this offer soon.

Modems or *Black Magic*

by Abed Kahale CATUG



Computers use binary codes of "ON" bits (which have a value of 1) and "OFF" bits (which have a value of 0) for sending information to components across short distances to serial peripherals such as serial printers.

Computers voltages from 5 to 15 volts for an "on" bit and from -5 to -15 volts for an "off" bit. But to send information over the telephone lines, computers need to modulate this digital information into analog format the telephones use, in other words convert it into variable pitch sound instead of beeps, then demodulate it at the other end so that the receiving computer can understand it. Hence, the device that does this takes its name from MOdulate DEModulate ⇒ MODEM.

Modems communicate one bit at a time using serial transfer. The standard serial transfer uses 9 wires of which only two wires actually transfer data, the rest manage the transfer process and provide signal ground, to prevent statics and interference.

Your communication software's dial command transmits a signal ATDT (ATtention: Dial Tone) followed by a phone number to your modem, which goes **off-hook** — *picks-up the phone so to speak* — and dials out using DTMF (tone) as opposed to pulse PT dialing in this case. When the modem on the other end receives a ring, it sends a signal (R1) to the software telling it to issue a command to pick up the phone.

When the remote modem picks up the phone, your modem sends out a hailing tone, and the remote modem responds with a higher pitch tone. The modems then exchange information (cackle) about how to transfer data in a *handshake*, rate (BAUD, bits per second), type of transfer (ASCII, Xmodem etc.) and settings (8N1) to determine the size of data packets the systems will exchange. Hence the word PROTOCOL.

Once the communications are established the modems send a Data Carrier Detect signal (squeal), which continues until the modems either hang up or gets cut-off.

Stop 0 0 1 0 1 0 1 1 Start

A data packet is a group of bits that constitutes a single character. The 8 in the setting 8N1 refers to the number of data bits in the packet. Each packet is enclosed by a START bit and a STOP bit that tell the receiving modem where the packet begins and ends. The 1 refers to the number of stop bits in the packet.

Stop Parity 0011101 Start

Data packets sometimes contain a special PARITY bit, which the modem use for basic error checking. Parity can be set to ODD, EVEN or NONE. When parity is even, the transmitting system sums the 0s and 1s and gives the parity bit a value of 1 or 0, whichever makes the total an even number (check-sum?). If the receiving modem gets a different result, it assumes that the transmitted data is incorrect and requests a re-transmit from the other modem.

When a communications session is over, each of the computers software sends an ATH (ATtention: Hang) command to its modem, instructing it to **on-hook** — hang-up. Or when the other modem loses the data signal carrier (squeal) it also hangs up.

DTR ⇔ Data Terminal Ready
DCD ⇔ Data Carrier Detect
DSR ⇔ Data Set Ready
ATA ⇔ ATtention: Answer
ATO ⇔ ATtention: (go) Online

QBOX-USA - Utica, MI, USA

John Impellizzeri and Don Walterman have a QBox BBS set up in the US. This BBS is carrying four QL related echomail message areas from the European BBS's that exchange mail via the Fidonet. In these echomail areas, a North American QL user can exchange messages with users in Europe. TF Services is the host BBS for the echomail. The message areas are: International QL, Minerva, Quanta and QBox-Sysop.

There are also local message areas along with file areas that will have public domain programs available for download. Uploads are gladly accepted too! The BBS is running the QBox software from Jan Bredenbeek on a QL computer. There is more info online about the hard and software used on the system. Please give it a try!

313 254 9878
24 hours, up to 2400 baud

COMPUTER CLASSICS
RT 1, BOX 117
CABOOL, MO 65689

Repair Service for America's
Favorite Home Computers
and their accessories

ADAM ATARI COMMODORE
IBM OSBORNE SINCLAIR
TIMEX TI TRS-80

BUY - SELL - TRADE - UPGRADE

Write for prices
SASE appreciated

Ph. (417) 469 - 4571

COMPUTER CLASSICS
*** FOR SALE ***

IBM XT COMPUTER $200 +shipping
Includes 10 MEG hard drive, 640K RAM, 360K floppy, serial port and parallel port, monochrome monitor, and keyboard.

IBM 286 COMPUTER $350 +shipping
Includes 20 MEG hard drive, 1MB RAM, 1.2MB & 1.44MB floppies, serial & parallel ports, keyboard, and monochrome monitor.

QUAD DENSITY 5 1/4 in. (720K)
half height floppy drives
$25 each or 10 for $150

IBM HARD DRIVES, CONTROLLERS,
and RGB COLOR MONITORS
Write for prices

Tony Blizzard
512 E Court, Weiser, Ida. 83672

My son, Ron, tells me that the QL supporting journals need articles on QL use and that, since I'm a heavy user, I should write some. Okay. But most of you are computer buffs; I'm not. The QL is the only computer I've ever used, excepting a couple hours playing with a Timex-Sinclair. When my sons advised me to get one of those "super-buy" QL computers I asked, "What do I need with a computer?" With a little argument they convinced me that if I only used the word-processor it beat my WD-40-soaked IBM Selectric every way imaginable. Turns out they were right. That is, after I nearly beat the thing to death learning Quill (old dog, new tricks situation). And I went through the same trauma with Text87. (This is a Text87 doc.)

Materially, my mind works on mechanical and electrical logic; not computer logic. Therefore, after I read a new program's manual I call Ron to find out what it means. He's developed into quite a translator.

My present hardware amounts to a Magnavox RGB monitor, 896K Trump Card, dual 5 1/4 Mitsubishi discs from Sharp, a Seikosha SP 1200-AS 9 pin and an Epson LQ-500 24 pin printer and a back-up QL with a monocrome monitor. The Seikosha just prints mailing labels these days; the 24-pin doing the heavy work. I picked up a cheap Tandy A/B switch to put between them and made up a cable from switch to serial 1. Both lights of the *å#!çø disk drivers come on every time a floppy is called, always have; Stuber wasn't very anxious to help fix it and it still drives me nuts. Otherwise, I'm happy. (Ron has about talked me into networking the backup QL; might be fun.)

Software? More than I use. Taught myself Quill and Archive out of the originals. Ron bought me Taskmaster; so far it's mastered me. (But I hope to write its "screen saver" into every boot I have!) Bought Stuber's early version of Xchange and have had to learn to use it for its 'mail-merge feature, which use I will describe herein. Also have Bill Cable's Archive enhancements and like what I've used very much but for the awfully slow screen changes.

Following is one layman's use of the QL:

As mentioned, I was interested in a good typewriter - Quill. It soon made the old way comparable to chiselling stone. And, until someone writes a mail-merge for Text87, I'll continue to use it for special purposes. Being politically oriented, if I'd had Quill 25 years ago, a lot more politicians would have had to make a lot more excuses concerning their actions.

Being, at the time of my QL purchase, maintenance manager of a good-sized vending machine business, I soon saw the advantage of a machine data-base for the whole operation and began spending my evenings typing in data for every machine in the business. My crew groaned when I first exposed them to the print-outs, but soon they were looking for updates of equipment on locations to help their preparations for repairs, etc.

Next, two of my sons turned their correspondence into a magazine and I got elected to keep the address data-base - now up to about 1400 names.

Have also developed my own six-page or so monthly(?) mail-out sheet (Text87 produced) and regularly update its address data-base.

Lastly, I run an irrigation-ditch lateral in our part of town 7 months of the year and my wife has now become secretary-treasurer. A tax bill must be sent to each address on the ditch-line each year. In the past the secretary had triplicate legal forms printed, crash-numbered and bound at considerable cost. Then she had to laboriously hand-enter all the information for each address.

Last year I used Xchange's mail-merge feature and set up the whole bill on an 8 1/2 X 11 sheet in Quill; then

created an Archive data-base to hold each lot's detailed info. This entailed looking back through years of dusty records and making trips to the county assessor's office to find the answers to riddles caused mainly by individualistic penmanship.

To create a form including all the necessary legalisms on one sheet really challenged Quill's versatility. (Samples are provided if the publisher has display space.) I could never have managed it without Quill's ability to use custom translates in the printer_dat. Happily, I had made drivers beforehand using translate codes (as shown by Mike de Sosa in an article from an old Time Designs) to take advantage of the LQ-500's founts.

When it came time to merge the data into the form most everything had to be jockeyed around some more. Through frustrating trial and error I discovered that the Xchange mail-merge only worked so many lines down the page. Because the tax-payer tear-off payment stubs had to be at the bottom of the sheet and had to carry the same number as the main bill, this meant that I couldn't merge bill numbers I had set up in the data-base. Finally I used a manual numbering machine. However, Ron recently brought his new public domain, updated, Xchange and we discovered (just before my QL's original membrane, as well as that of the back-up, decided to quit) that that problem has been corrected. This year's bills went out with the numbering merged perfectly.

To have everything align properly on the merged sheet I discovered that I had to allot the proper number of spaces on the doc (even though it pushed things out of place in the Quill template) and then be sure to fill every space of every field in the database. That meant a lot of spaces run into the 31 fields on the Archive screen I had created.

During the course of all this research and set-up I discovered that through the passage of time, selling off of parcels, etc., the ditch tax had lost much of its equity. So before the 1993 billing this cantankerous old dog attacked Abacus hoping to equalize the tax-base per square foot. But after learning its basics, I found it easier to use a calculator. Seems that, given length & width, Abacus could figure square footage of each lot, but I know zilch about math formulas and programing. Maybe someone can explain to me how to set up Abacus to do that job in the future.

Last year I printed the bills on continuous feed paper. This time I used a cut-sheet feeder. One seems to work about as well as the other although the cut-sheet eliminates the extra time needed to tear all the perforations.

By the way, a local programer wrote a special PC program for another ditch-lateral company in town which went on computer a year before us. They still have a glitchy situation with botched bills while, except for the manual numbering - now corrected - our bills printed out without a hitch first time out.

Don't know if this is what QL publications need, but this is how I use my QL. As for the technicalities of how some of the objectives are accomplished - I tend to forget the details until I have to do it again. Then re-reading the manual usually refreshes my memory.

Again, I'm not a computer buff, never ran a PC, but I know a useful tool when I see it operate. Now when the grandkids play the simple Quanta library games on my QL, I get nervous. And when the membrane gave up I wasn't sure how life would progress. Luckily, Frank Davis of Mechanical Affinity had me fixed up with new-production membranes in just two days. Real service!

Hey, when the QL "died" I was too busy using mine to attend the funeral.

- end -

MIDDLE WEISER IRRIGATION LATERAL DISTRICT

NOTICE OF TAXES

TAX LEVY FOR 1993

No.102

Weiser, Idaho, November ,1993

Myers Enterprises
537 E Court
Weiser, ID 83672
c/o J. Myers

Please take notice that your taxes for the year 1993 on the below described property amount to the sum of

Fifty-seven and 28/100- DOLLARS $ 57.28

DESCRIPTION OF LANDS	Lot	Blk	Apportionment	AMOUNT AND DISTRIBUTION
Hanthorn Add'n.	1-2-3	2		Total Tax $ 57.28
158' X 190' X 100' 22910 sq'				1/2 Pd. 1993 _____
				Balance Due
				Penalty
				Advertising _____
				Total Paid $

THE FOLLOWING TAXES ARE DELIQUENT ON ABOVE PROPERTY

Year	Tax	Interest	Amount

IMPORTANT - Notice is hereby given that the assessments above indicated are due and payable at the office of the District, same being the home of Barbara Blizzard in Weiser, Idaho. **Said assessment will become deliquent at 6:00 o'clock p.m. on the third Monday of December, 1993,** provided that if one-half of the same shall be paid on or before said date, the remainder may run without interest but will become delinquent if not paid on or before 6:00 o'clock p.m. on the third Monday of June, 1994. **A penalty of two percent will be added to all delinquent assessments and an interest charge of ten (10) percent per annum from January 1st will accrue until settlement. THESE AMOUNTS ARE SET BY LAW.**

Make remittance payable to: BARBARA BLIZZARD, Secretary
512 East Court, Weiser, Idaho 83672

If the owner's name is not correct please indicate below:

Present Owner_____

Address_____

_____

MEMO:

_____Secretary

Your cancelled check is your receipt

FOR PROPER CREDIT CUT AND RETURN THIS PORTION WITH PAYMENT

No.102 1993

FULL OR FINAL PAYMENT

_____, 19_____

Barbara Blizzard, Secretary
512 East Court, Weiser, Idaho

Enclosed find $_____ in Final and Full payment of taxes levied for the year 1993 on the property described in the Tax Notice bearing the same number and year as this stub and from which this stub has been detached.
Respectfully,

FOR PROPER CREDIT CUT AND RETURN THIS PORTION WITH PAYMENT

No.102 1993

FIRST INSTALLMENT

_____, 19_____

Barbara Blizzard, Secretary
512 East Court, Weiser, Idaho

Enclosed find $_____ in payment of the First Installment of taxes levied for the year 1993 on the property described in the Tax Notice bearing the same number and year as this stub and from which this stub has been detached. Respectfully,

MECHANICAL AFFINITY

513 EAST MAIN ST.
PERU IN 46970
317 473 8031
FRANK DAVIS

OR

5231 WILTON WOOD CT
INDIANAPOLIS IN 46254
317 291 6002
PAUL HOLMGREN

Bringing the very best to you in SINCLAIR software and hardware from around the world. We want to provide service to you the customer. Need something, give us a call. Make all checks or money orders out to either "Frank Davis" or "Paul Holmgren". Thanks! Payment in US$.

SPECIALS GOOD TILL CURRENT SUPPLY EXHAUSTED

TRUMP CARDS: 896K RAM, Tool Kit 2, disk interface for QL. These are used, but in good condition and are being offered for only $100 till current supply is sold.

512K MIRACLE EXPANDER RAM CARDS:for those with a disk interface and no extra memory, or an unexpanded QL and would like to increase the available memory in order to use larger programs, then these are excellent, and are available for only $45.

SINCLAIR QL PRINTERS: in QL black, 9 pin dot matrix printers made to work specially with the QL. Tractor feed, so they are great for labels, and since they are an impact pin printer (as opposed to a laser, deskjet or bubblejet) they can be used with multi-part forms. These are available for only $60.

SINCLAIR QL MONITORS: we have these in limited supply, with more on the way, so please call or write first to check on availability. These are black TTL RGB monitors bearing the QL logo and designed to work great with the QL. On sale for $125

SERIAL PRINTER CABLES for the QL: for the low price of $14 each.

3D PRECISION: a suite of programs which permits the creation and rapid manipulation of three-dimensional objects in any color, or combination of colors on the QL. It even comes with its own 3D Editor, SuperBasic Toolkit and Assembler Toolkit. Objects are of the wire frame type. We normally offer this program for $40, but on sale for $25.

EYE-Q: an advanced graphics package for the QL to create sophisticated designs of any size on the QL for saving, reloading in other programs or dumping to a printer (hundreds of printers supported). Normally $54, on sale for $40.

HERMES REPLACEMENT FOR THE 8049: for dependable upto 19,200 baud operation, improved sound and independent serial bauds all this and more for only $43. To install requires opening your computer.

Psst!! Looking for more Programs?? Cheap??

Here is a very small list of what can be found out there in the world for the QL if you have a modem and can call a BBS. For those of you without a modem, think about it. There are several BBS's in the USA that cater to our line of computers.
There is also INTERNET. A world wide method of communicating via computer. They offer programs and an opportunity to actively exchange electronic correspondence with other QL users.

These programs have all been uploaded somewhere out there for QL'ers to use.

ABAtra_exp	Psions' Abacus file format explained
ANSISR_zip	strip ANSI codes from log files, uses TK2 and Qlib
BASCON_zip	SB Extension for Linking in QJump Configuration Blocks
C401A_zip	C68 4.01 disk 1. **C** for the QL various utilities
C400B_zip	C68 4.00 disk 2. **C** for the QL Main system disk
C68400_zip	C68 runtime disk 3. Master doc disk for C68 The current "STANDARD for '**C**'" on the QL.
CFG_zip	place QPTR configuration blocks into '**C**'
CHECK_zip	SB extension to check for (non) numeric INPUT
CIRCUI_zip	Miscellaneous circuit diagrams and connector pin-outs
CRITMS_BAS	Critical Mass Game (1 or 2 players) from Dec. 88 QL World
DISKS_TXT	Text file describing using Teac 2.88Mb disk drives
DJEP_zip	DeskJet printer utility, prints envelopes
DJWDEM_zip	Demo copies; DiscOVER, MULTI-DiscOVER, TexTIDY & FLPClone
EMAIL_txt	Known INTERNET addresses of other QLers
FILEIN_zip	Pointer environment utility
FNTSRC_zip	source files (SB and Metacomco Assembler) FontEditor v.2.3
FONTED_zip	QL FontEditor v.2.3 for the Sinclair QL
HDRRST_bas	needed to restore file header of some transferred files
HISTDV_zip	Last line recall/history device for QL jobs
IBMDSK_zip	Lets QL with TK 2 and 3.5 inch drives access IBM disks
LHQ_zip	LHQ v1.0 for the QL. Another file archiver and extractor
MAIL_zip	mail list program for ARCHIVE
Minnie_zip	Text file detailing the Minerva ROM 1.93+ features
MODPSI_zip	Some mods for the PSION suite of software
POINT_zip	Description of setting up as QL FidoNet Echo Point
PullDown_zip	This is a method of creating PULL-DOWN/POP-UP screens
QED_zip	QED 1.01 A general purpose editor for the QL
QEM_zip	first? version of QeM
QEM24_zip	QEM 2.4 early version of QEM
QEMV32_zip	QEM v3.2 An excellent terminal program for the QL
QEM35U_zip	QEM v3.5 upgrade files. upgrade QEM 3.0+ to 3.5
QEM36U_zip	QEM v3.6 upgrade from v3.5
XPRLIB_zip	Latest XPR libraries for QEM v3.5. needed for 3.5 & 3.6
QLGIF_zip	Utilities to code and decode GIF files. by Carlo Delhez (1993 version)
QLHD_zip	Information on hooking up an IBM type MFM/RLL hard drive to a QL
QLIBRT_zip	Qliberator runtimes version 3.36
QLSPIT_zip	ZM-1 Spectrum Emulator for the QL. working demo
QPACER_zip	QPACER QPac2 Boot Generator Issue 1.1
QSI_zip	Determine speed of your QL, compared to QXL, Gold Card

QUADRI_zip	Quadric Equation Plotter
QUILtra_exp	Psions' QUILL file format explained
RELYQL_TXT	A reliable QL! Text describing modifications to the QL
RESCUE_zip	Floppy Disk Rescue & String Search Utility
Ruleta_zip	Roulette game
SAVER_LZH	Pointer environment screen saver. Set up as a hotkey
SPECTATR_zip	Spectator v1.00 - A Spectrum emulator for the QL
SPECKE_zip	4 graphic files of the Spectrum Computer keyboard layout
VERS_zip	determines program info about other QPTR progs.
VIEW_LZH	Pointer environment utility
XCHANG_zip	Psion's QL Xchange v3.90 is now public domain (Quill, Archive, Abacus, Easal in one package)
ZXTRICA_zip	Xtricator v1.10 - ZX81 emulator for the QL

Unzip22	Unzip version 2.2 for the QL. (revised)	***NEEDED FOR THESE FILES***
UNZIP9_EXE	QL UnZip program Ver 0.96. by Jan Bredenbeek, exe form (early version)	
ZIPUTI_zip	Zip Utilities v1.0. utilities for dealing with zip files	
ZIP22_exe	Zip version 2.2 for the QL. (The "ES" version) (revised)	
Zip22_txt	Revised Docs on using Zip2_exe	

FNxxxx_zip	QL InterNational FidoNet Echo Mail messages, 14 files so far
QLxxxx_zip	QL Quanta BBS Messages from the UK, 5 files so far

Spectrum/QL related files for the PC/Clone

ENV_bin	ENV_bin code compatable with QXL and SMSQ
JPP_zip	Excellent Spectrum Emulator for 386SX or greater PC
Make_C_zip	C68 files MAKE & CCP for the QXL
QDIR_zip	Looks at QL disk DIRs on a PC
QLSCR_zip	PC program to view QL screen files, can convert to GIF
QLTOOL_zip	Lets a PC look/examine QL disk
QXLfmt_zip	for the QXL card. Convert a disk to QDOS format (same as FORMAT?)
Z80V201A_zip	128K/48K ZX Spectrum Emulator V2.01 by Gerton Lunter
Z80V201B_zip	LATEX and POSTSCRIPT DOC files for V2.01 Z80 Spectrum

QL related file for the MAC

CP2MC1_zip	convert text/Quill files to be used on the MAC

These programs and files fill 8 720 disks and are available for $8.00 (postage paid) from;
Paul Holmgren
5231 Wilton Wood Ct.
Indianapolis, IN 46254

The heading may make you think you are listening to a flock of ducks on the local millpond, or a skein of geese on their annual migratory pilgrimage flying high overhead. But really "QLerk" is the name of a program by the renowned New Hampshire programmer, Bill Cable.

Bill has spent a lot of time, a lot of work, on this program, and he has tested it in various businesses in his area. The result is a program that will keep your business records in order, or even just your household finances in order. The only limitation on the uses of this program are what the user places on it himself.

I was lucky in being able to look at version 2.1, but now I have version 3.0, and the improvement is significant. I will not mention the previous version again, as that would serve no useful purpose, and I only mention 2.1 to show that Bill is in a constant phase of updating and improving the system, and that is what it is - it is a system to keep your records up to date.

This would be a good time to mention that the program is vast. It is very complex, very comprehensive. It is going to take the user some time to get to know the ins and outs of it. It is not for the faint hearted, but once you have seen what it can do - you will be impressed.

This is an Archive program, Archive being provided on the program disk, but can also be run using your own Archive. It requires that you have extra memory, a minimum of 512K extra memory would be required. QLerk will run on a 7.5MHz QL, but slowly. The Trump Card much better, the Gold Card being best of all because of its extra speed of operation.

Disk drives are essential, MDV could not possibly hold everything. QLerk will run from two 720 disks. Or, if you have HD or ED drives or a hard drive, it could all be run from the one disk. The options are there for your own configuration.

To start with, there is an excellent manual which contains a wonderful demo taking you through the example setup on the disk, one keypress at a time.

The demo starts off by telling you how to make your first backup of the program disk, and takes you through each and every step to get the demo up and running. You are shown how to use the Menus. How to check what payables are in the system, and how to make payments. How to handle receivables, how to write a cheque, make purchase orders, invoices, get various reports from the system.

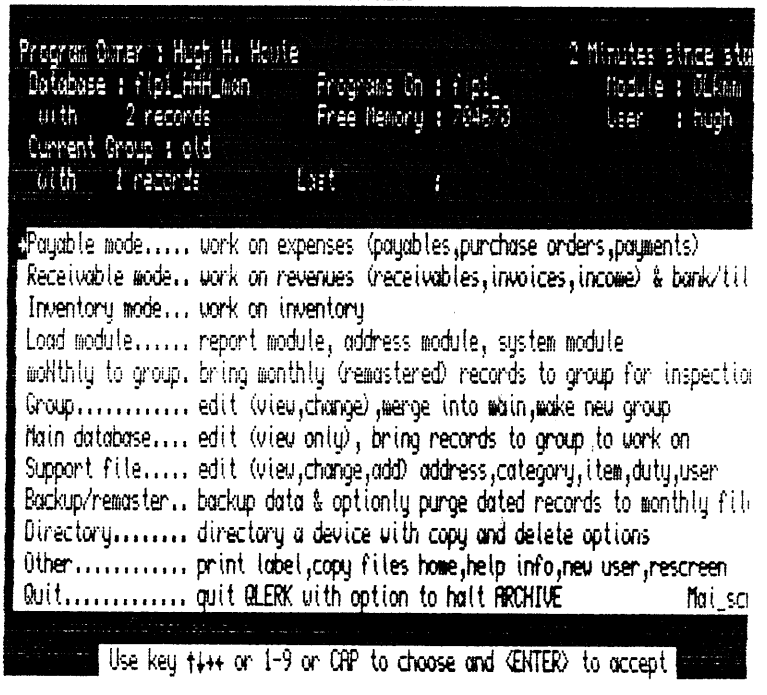
Once you have run the demo a couple of times, you will have an excellent idea of how the whole system operates.

The Manual is easy to read and follow. It describes the various Program Modules and what each contains and does.

The whole is menu driven and there are about a dozen and a half of them. Each menu contains about 10 - 12 options, it varies, and each option has its own set of options.

Here is what the Main Menu looks like:-

QLerk MAIN MENU



Now to keep track of all those Menus, could be a formidable task, but there is a complete page allocated to each menu. That page starts off with a screen dump of what the menu looks like, just as I did above, and there is an explanation under the dump, telling you what occurs there. This Screen dump method of displaying the menus, means that you can refer to the manual for a particular menu, and see what it does for you. Thus helping you decide whether to go back or forward to complete a task.

The facility on each menu can be selected in a number of ways. You could use the up/down arrow and ENTER, or you could select by the key letter of the facility, or, if you can remember the number of the facility required, you can make the selection by number!

Should you at any time get out of QLerk by pressing ESC by mistake, just type 'm' and you will be back into the Main Menu, so all is not lost.

The system is completely inter-active, in that an entry made in one mode, is transferred to all other modes where it is required, so that if a payment is made by cheque, the money is removed from the bank account, the account is reduced accordingly. Where the money went and where it came from, is all recorded automatically. One aspect that Bill mentions in his manual, will show just how interactive the system is. If you were to pay an account by VISA, the account would be paid, and the system would make out a new account showing that you owed VISA that amount.

The system can handle up to five cheque writing accounts at the same time. Plus two or three cash registers.

The cheque is filled out by the printer, and an address label printed ready to be pasted to the envelope. Everything is optional to your own requirements.

Payables, Receivables, Purchase orders, Cheques, Invoices, are all available. Inventory control, Bank Accounts, Cash Registers, Taxes, Duties, Deductions, Cyclical payments / receivables are all taken care of.

You want a hard copy of some aspect of the business? By date? By occurrence? Frequency? How often has someone done business with you this last few weeks? months? You name it, and you will probably find the information is in there somewhere, all you have to do is dig it out, and with those easy-to-use menus, it is not so difficult.

There is a complete Inventory system so that you keep track of everything in stock. Does its own value evaluation for you. Enter Item, Price, and Number, and everything is taken care of.

No matter what you do, it is almost impossible to make an error. For example, when you make some entries, you are asked if you wish the entries to be merged, and you must answer with Y or N, thus you always have the opportunity to make corrections BEFORE the documents are altered or changed.

If you have any questions to ask about this program, why not write to Bill, after all he can answer much better than I can, and he is always willing to help.

His address is:-

Bill Cable
Wood and Wind Computing
RR 3. Box 92
Cornish, NH 03745. USA
Phone (603) 675-2218q

CABLE COLUMN

By Bill Cable

ARCHIVE SERIES
PART 15 : ENTERING QUERY - A PROGRAM TO INTERROGATE ANY ARCHIVE DATABASE

The program (collection of procedures) listed below when merged with the procedures we have entered up until now gives us a useful program that can define and execute complex searches in any ARCHIVE database using simple menus. It allows us to not only find records by the occurrence of text within a record but also by the fields of a record meeting specified criteria. For instance we could query the GAZET_DBF supplied with ARCHIVE for all countries with high population densities which we might define from menus as (area/pop)<5. We can view and print out the results of each query. Besides being a fun program to play with it provides us with ARCHIVE statement building procedures which will be very handy for later programs.

In this issue of UPDATE you will find a review of my financial program QLerk by Hugh Howie. He kindly agreed to be one of my Beta testers and put together his experiences in the review. As he mentions it is a comprehensive program and not for the casual user. Financial programs are like that as you can't half do them. They depend on a consistent and steady supply of information and are of little use if not so used. QLerk has changed very little in the past 8 months since it has been released (on a very low key basis) as it has proved to be very dependable. Unfortunately I haven't completed a comprehensive manual yet. It is currently almost 50 pages but should be close to 100 when it covers all features in detail. I am now working hard on the manual after hardly touching it since the Miracle in Newport Fest because of other commitments. If you are interested in the program please contact me. I am selling it for $75. For the time being until more detailed reviews and advertisements are available I am willing to let potential users inspect the program before purchasing it. I have no desire to sell it to people who won't use it but I have to protect my very large investment of time in its development and future support for the very small potential QL market.

PREPARATION

Start ARCHIVE in your customary way so you see the ARCHIVE command prompt (>). Load the group of procedures entered last time and saved in a file called "group3". Start the editor and first modify "by_arrow" and "field_display" procedures that have small changes and then enter the new procedures. I have removed indentions in order to make the listing fit compactly in this listing. Save this old and new work in a file called "query". To run first load "query" and type "query" to start.

LISTING OF PROCEDURES WITH CHANGES

```
proc by_arrow;sl,nl,sc,nc,bw,s$,ap
rem
rem  Choose by moving arrow over options drawn on screen and
pressing <ENTER>
rem
rem Passed arguments defining screen area in blocks of rows
and columns
rem  sl - start line, nl - number of lines
rem  sc - start column, nc - number of columns
rem  bw - block width for choices on screen
rem  s$ - string of key characters, ap - at position (between 1 and
nl*nc)
rem
rem Local internal variables
rem  nl - becomes end line number, al - at line (current line)
rem  ac - at column (current column), ll - last line position
rem  lc - last column position, es - length of key string
rem  i$ - key press character,  i - key press character ASCII code
rem  k - acceptable key press flag
rem
rem Global returned variable
rem  ans - returns choice value as position in s$ or 0
rem
rem Uses s$ length (es) to keep within defined boundaries
rem
local al,ac,ll,lc,es,i$,i,k
if ap<1 or ap>nl*nc: let ap=1: endif : let nl=sl+nl-1
print rv$;
let al=sl+int((ap-1)/nc): let ac=sc+((ap-(al-sl)*nc)-1)*bw
let ll=al: let lc=ac: print at al,ac;chr(14);"";: let es=len(s$)
let i=0: while i<>30 and i<>48: let i=0: let i$=upper(getkey())
if ap<es: let k=instr(s$(ap+1 to es),i$): else : let k=0: endif
if not k: let k=instr(s$,i$): else : let k=k+ap: endif
if not k: let i=code(i$): if i>48 and i<58: let k=i-48
if k>es: let k=0: endif : endif : endif
if k: let al=int((k-1)/nc+1): let ac=(k-(al-1)*nc)*bw+sc-bw: let
al=al+sl-1
else : if i=30 or i=48: let k=i
else : if i=2: let k=i: let al=al-1
if al<sl: let al=nl: endif : else : if i=3: let k=i: let al=al+1
if al>nl: let al=sl: endif
else : if i=4: let k=i: let ac=ac+bw
if ac>sc+bw*(nc-1): let ac=sc: endif
else : if i=5: let k=i: let ac=ac-bw
if ac<sc: let ac=sc+bw*(nc-1): endif : endif : endif : endif : endif
if nc*(al-sl)+(ac-sc)/bw+1>es: let ac=sc: let al=sl: endif : endif :
endif
if ll<>al or lc<>ac: print at ll,lc," ";: let ll=al: let lc=ac: endif
let ap=(al-sl)*nc+(ac-sc)/bw+1
if k: print at al,ac;"";: else : print at al,ac;"?";: endif : endwhile
if i=48: print at al,ac,"0";rv$;chr(15);
else : print at al,ac;"+";rv$;chr(15);: endif
let al=al-sl+1: let ac=(ac-sc)/bw+1:rem let ans=10*al+ac
if i=48: let ans=0: else : let ans=ap: endif
endproc
proc field_display;l$,sl,nl,sc,nc,bw,fw
let n=0: let al=sl: let ac=sc: let cc=1: let fld_list$="": while
n<numfld()
let i$=fieldn(n,l$): if len(i$)>fw-2: let i$=i$(1 to fw-2): endif
if len(i$)=1: let i$=upper(i$)
else : let i$=upper(i$(1))+lower(i$(2 to len(i$))): endif
let fld_list$=fld_list$+i$(1)
```

```
print at al,ac;i$,: let n=n+1: let cc=cc+1
if cc>nc: let ac=sc: let al=al+1: let cc=1: else : let ac=ac+bw: endif
endwhile
endproc
```

LISTING OF NEW PROCEDURES

```
proc arrow_clr
print chr(8);rv$," ";rv$,
endproc
proc arrow_mess;m$
clx;21,rv$: let i=(80-(21+len(m$)))/2
print at 22,1;rv$; tab i;"Move arrow to choose "+m$; tab 79;
print at 23,1; tab 25;"<ENTER> to select or 0 to exit"; tab 79;rv$;
endproc
proc clx;n,r$
print at n,0;r$;cs$; at n,1;r$;
endproc
proc do_tmp;c$
let i$=sd$+"tmp_tmp": kill i$: spoolon i$ export
lprint "proc tmp_tmp"
lprint c$: lprint "endproc": spooloff
merge i$: error tmp_tmp: if ermum():erry;23,"searching"
let l$="": endif
endproc
proc field_pick;m$
arrow_mess;m$,by_arrow;1,fld_rows,0,fld_cols,fld_wid,fld_list$,1
arrow_clr
endproc
proc field_size_1
use "s": let n=numfld()
if n<29: let j=2: let k=20: else : if n<43: let j=3: let k=20
else : if n<57: let j=4: let k=15: else : if n<71: let j=5: let k=15
else : if n<85: let j=6: let k=13: else : if n<99: let j=7: let k=11
else : if n<113: let j=8: let k=9: else : if n<127: let j=9: let k=8
else :acky;23,"Too many fields to display":get_files: stop
endif : endif : endif : endif : endif : endif : endif : endif
let i=int((n-1)/j+1)
let fld_rows=i: let fld_cols=j: let fld_wid=k
endproc
proc prt_end
if prt: lprint : lprint : lprint "Query completed with ";cnt;" matches"
endif
endproc
proc prt_head;m$
if prt and l$<>"":msg;23,"printing"
lprint "QUERY OF ARCHIVE DATABASE "+sd$+sn$;
lprint tab 65;date(0)
lprint tab 65;count();" records"
lprint "Query : ": lprint "  ";m$: lprint : endif :liny;23
endproc
proc prt_record
if prt: lprint tab 70;cnt;" - ";recnum()
if prt=1: if d1>-1: lprint fieldn(d1);" : ";fieldv(d1): endif
if d2>-1: lprint " ";fieldn(d2);" : ";fieldv(d2): endif
if d3>-1: lprint "  ";fieldn(d3);" : ";fieldv(d3): endif
else : let i=0: while i<numfld(): lprint fieldn(i);" : ";fieldv(i): let i=i+1
endwhile : endif : endif
endproc
proc query
mode 0:setup: error close_all
heady;"Query an ARCHIVE Database"
print " 1. The source database will be looked at (not altered)."
print " 2. For Arrow Menus use arrow keys, 1-9, or a-z of capped
letter in"
print "    choice for picking selection then <ENTER> to activate."
print " 3. You can Query the database by matcing text within a
record (Find)."
print " 4. You can Query the database by building from menus a
true/false"
print "    statement using fields of the record and ARCHIVE
functions and then"
print "    searching for records for which the statement is true."
print " 5. Statements have values which are either 'TEXT' or
'NUMBER'. Only"
print "    Numeric statements can be searched and nonzero is
true, zero false."
print " 6. Up to 3 statements can be stored in slots to be used in a
search"
print "    or used for building more complex statements."
print " 7. Only 3 fields of a record are displayed but are user
selectable."
print " 8. Matches can be manually or automatically stepped
through."
print " 9. Output goes to the screen and can go to the printer
also."
print "10. A temporary file 'tmp_tmp' may be written to source
device."
device;21,"Device where source database is located. eXit to
stop"
if ans$="": mode 1: stop : endif : let sd$=ans$: dir sd$
inpy;23,"  File name of source database (with extension)"
let sn$=ans$
if sn$="":query: return : endif
if instr(sn$,"_")>1: let sn1$=sn$(1 to instr(sn$,"_")-1)
else : let sn1$=sn$: let sn$=sn$+"_dbf": endif
error looky;sd$+sn$,"s": if ermum():erry;23,"accessing "+sd$+sn$
query: return : endif
let d1=0: let d2=-1: let d3=-1: if numfld()>1: let d2=1: if numfld()>2
let d3=2: endif : endif
let s1$="": let t1$="": let s2$="": let t2$="": let s3$="": let t3$=""
field_size_1: mode 0
print rv$; tab 5;"Fields of ";sd$+sn$; tab 67;count();" records"
field_display;"s",1,fld_rows,1,fld_cols,fld_wid,fld_wid: print rv$;
let prt=0: while 1: first :clx;16,rv$:show_record:clx;21,rv$: let l$=""
if prt: let i$="on": else : let i$="off": endif
print rv$; tab 24;" Query Options"; tab 72;"prt ";i$
print " Find  Search  Build  Display  Print  Quit ";rv$;
by_arrow;22,1,1,6,12,"FSBDPQ",1
if ans=0 or ans=6: error close_all: mode 1: stop : endif
if ans=1 or ans=2: if ans=1:inpy;23,"Find what": let l$=ans$
prt_head;"find '"+l$+"'": if l$<>""
clx;16,rv$: print rv$;"Finding '";l$;"'";rv$: find l$: endif
else :stm_display:stm_pick
if l$="":acky;23,"Build a search statement first"
else : if t$="TEXT":acky;23,"Can't search with TEXT statement"
let l$=""
else :prt_head;"search "+l$:clx;16,rv$: print rv$;"Search ";l$;rv$
error do_tmp;"search "+l$: if ermum():erry;23,"search"
let l$="": endif
endif : endif : endif : if l$<>"": if not found():acky;23,"Not found"
else : let ans=1: let qy=1: let cnt=0
while ans=1 and found() and not eof()
show_record: let cnt=cnt+1
print at 23,53;"Match # ";cnt;"  Rec # ";recnum();
prt_record: if qy
print at 23,1;rv$;"  Continue  Automatic  eXit ";rv$;
by_arrow;23,1,2,3,12,"CAX",1: endif : if ans=1 or ans=2
continue : endif : if ans=2: let ans=1: let qy=0: endif : endwhile
```

```
if ans<>0 and ans<>3 and l$<>""
acky,23,"Done after "+str(cnt,2,0)+" matches"
prt_end: endif : endif : endif
else : if ans=3:stm_display:stm_build
else : if ans=4: let d2=-1: let d3=-1
let j$=" field to display"
field_pick,"First"+j$: let d1=ans-1:show_record
field_pick,"Second"+j$ let d2=ans-1:show_record
show_record:field_pick;"Third"+j$: let d3=ans-1
else : if ans=5: if prt: let prt=0
else :yorn,23,"Print 3 fields like display rather than all fields"
if ans$="y": let prt=1: else : let prt=2: endif
endif : endif : endif : endif : endif : endwhile
endproc
proc show_record
clx,18,"": if d1>=0: print fieldn(d1);" : ";fieldv(d1): endif
if d2>=0: print "  ";fieldn(d2);" : ";fieldv(d2): endif
if d3>=0: print "    ";fieldn(d3);" : ";fieldv(d3): endif
endproc
proc stm_build
while 1:clx,21,rv$: print rv$; tab 16,"Statement Building"
print "  Build a new statement      eXit statement building",rv$,
by_arrow,22,1,1,2,28,"BX",1: if ans=0 or ans=2: return : endif
if ans=1:stm_construct,"First Part"
if l$="": return : endif : let la$=l$: let l$="": let ta$=t$: let t$=""
stm_construct,"Second Part": let lb$=l$: let tb$=t$
if (ta$<>tb$) and lb$<>""
acky,23,"1st and 2nd Statement parts have mismatched types"
let la$="": let lb$="": endif
if lb$<>"": let i$="Relate ("+la$+") to ("+lb$+")"
if len(i$)>70: let i$=i$(1 to 70)+"..": endif :clx;21,rv$
print rv$,i$," by  ", at 22,0;"  =    <>   >    >=   <    <=   +"
let i$="=<>=<=+": if ta$="TEXT": let i=1: else
print " -    *    /    ^    And   Or   eXit"
let i=2: let i$=i$+"-*/^AOX": endif
print rv$,:by_arrow;22,i,1,7,7,i$,1
if ans and ans<>14: let i$=i$(ans): if ans=2: let i$=i$+">"
else : if ans=4 or ans=6: let i$=i$+"="
else : if ans=12: let i$=" and "
else  if ans=13: let i$=" or ": endif : endif : endif : endif
let l$="("+la$+i$+lb$+")"
if ans=7 and ta$="TEXT": let t$=ta$
else : let t$="NUMBER": endif : endif
else : let l$=la$: let t$=ta$: endif
if l$<>"":clx,21,rv$: print rv$;"New statement : ";l$
yorn,23,"Put new statement in statement slot": print rv$,
if ans$="y":stm_put
endif : endif : endif : endwhile
endproc
proc stm_construct;m$
clx,21,rv$
print rv$, tab 7,"Building new statement : "+m$+" starts with"
print " Field      Text      Number      Statement   Bypass";rv$,
by_arrow,22,5,1,5,14,"FTNSB",1
let l$="": let t$="": if ans=5 or ans=0: return : endif
if ans=1:field_pick,"Field for statment"
if ans<>0: let ans=ans-1: let l$=fieldn(ans)
if fieldt(ans): let t$="TEXT": else : let t$="NUMBER": endif : endif
else : if ans=4:stm_pick
else : if ans=2: let t$="TEXT": else : let t$="NUMBER": endif
print at 23,1;rv$," Enter "+lower(t$)+" : ";rv$," ";: input i$: let l$=i$
if t$="TEXT": let l$=""+l$+"": endif : endif : endif
let i$=l$: if len(i$)>45: let i$=i$(1 to 45)+"..": endif
if t$="TEXT":stm_text;i$: else :stm_number;i$: endif
endproc
proc stm_display
clx,16,rv$: print rv$," Slot  Type       Statement";rv$;
stm_line;1,s1$,t1$
stm_line;2,s2$,t2$
stm_line;3,s3$,t3$
endproc
proc stm_line;l,l$,t$
if len(l$)>60: let l$=l$(1 to 60)+"..": endif
if len(t$): let i$=t$(1): else : let i$="": endif
print rv$; at 16+l,3,l, tab 9;i$; tab 15,l$; tab 79;rv$;
endproc
proc stm_number;m$
clx;21,rv$: print rv$;"Modify ("+l$+") with which function : ";
print at 22,1," None   Abs()  Chr()  Int()  Sgn()  Str()";
print rv$,:by_arrow;22,1,1,6,9,"NACISS",1
if ans<2: return : else : if ans=2: let l$="abs("+l$+")"
else : if ans=3: let l$="chr("+l$+")": let t$="TEXT"
else : if ans=4: let l$="int("+l$+")"
else : if ans=5: let l$="sgn("+l$+")"
else : if ans=6: let l$="str("+l$+",3,0))": let t$="TEXT"
endif : endif : endif : endif : endif : endif
endproc
proc stm_pick
arrow_mess;"Statement
slot":by_arrow;17,3,2,1,20,"123",1:arrow_clr
if ans=0: return : endif
let i$=str(ans,2,0)+"$": let l$=value("s"+i$): let t$=value("t"+i$)
endproc
proc stm_put
arrow_mess;"Choose slot for new statement"
by_arrow;17,3,2,1,20,"123",1:arrow_clr
if ans=0: return : endif : let t4=1: if ans=1: let s1$=l$: let t1$=t$
else : if ans=2: let s2$=l$: let t2$=t$: else : let s3$=l$: let t3$=t$
endif : endif . stm_line;ans,l$,t$
endproc
proc stm_text;m$
clx;21,rv$: print rv$;"Modify ("+m$+") with which function : ";
print at 22,1," None Code() Instr() Len() Lower() Upper() Substr";
print rv$,:by_arrow;22,1,1,7,9,"NCILLUS",1
if ans<2: return : else : if ans=2: let l$="code("+l$+")": let
t$="NUMBER"
else : if ans=3:clx;21,"": input at 22,1;"Text for instr() test : ";i$
let i$=""+i$+"": let t$="NUMBER":yorn;23,"Is "+i$+" the substring"
if ans$="y": let l$="instr("+l$+","+i$+")"
else : let l$="instr("+i$+","+l$+")": endif
else : if ans=4: let l$="len("+l$+")": let t$="NUMBER"
else : if ans=5: let l$="lower("+l$+")": else : if ans=6
let l$="upper("+l$+")"
else :clx,21,"":inpy;22,"Begining position for substring "
let i=val(ans$)
inpy;23,"Ending position for substring  ": let j=val(ans$)
if i>0 and j>=i and j<255: let l$=l$+"("+str(i,2,0)+" to "+str(j,2,0)+")"
endif : endif : endif : endif : endif : endif : endif
endproc
```

Sorry that there is no detailed documentation within most of these procedures but time and space prohibit it. The Cable ARCHIVE Issue Disk has been updated to include the group1, group2, group3, and query programs covered in UPDATE plus matcher, replace, and windex procedures covered in IQLR. Next time another useful ARCHIVE program. Until then, Happy Archiving!

```
    100                                                          REMark
*****************************************************
    110 REMark  *                                    A-DEE-DOO-DAH  1.0
                *
    120 REMark  *                                              by  Al  Feng
                *
    130 REMark  *                                    AN  UNZIP  FRONT  END
                *
    140 DATA_USE RAM1_: REMark ********     SAVE as "DooDah_BAS"
*********
    150 F$="_FLIST_imp" : t$="xxx" : S$="xxx" : W$="1"
    160 pn=0 : POKE 163890,0 : MODE 0
    170 WINDOW#2,512,256,0,0 : PAPER#2,7 : BORDER#2,1,7 :INK#2,0 :
CLS#2  :  WINDOW#0,413,10,50,241  :  PAPER#0,7  :  INK#0,0  :
WINDOW  462,250,25,3  :  OPEN#3,scr_458x200a27x48  :  PAPER#3,7  :
OPEN#4,scr : u$=" A-DEE-DOO-DAH " : AT#2,21,4 :
PRINT#2,u$;" by Al Feng "\TO 4;" © 1994  PLATYPUS Software " :
PAUSE 30
    180 PAPER 7 : BORDER 1,7 : CLS : CL : lne : AT 0,7 : PAPER 5 :
INK 0 : PRINT" SELECT_DEVICE " : cj
    190 DEFine PROCedure CLSc : BLOCK 458,225,0,10,0 : END DEFine
    200  DEFine  PROCedure  CLSd  :  BLOCK#2,330,10,83,240,7  :  END
DEFine
    210 DEFine PROCedure CLSe : BLOCK 458,190,0,45,pn : END DEFine
    220 DEFine PROCedure sx : DIM Z$(40,48) : DELETE t$&W$&F$
    230 OPEN_NEW#6,t$&W$&F$
    240 DIR#6,t$&W$&"_" : CLOSE#6
    250 OPEN_IN#7,t$&W$&F$ : FOR c=0 TO 40
    260 IF EOF(#7) THEN EXIT c
    270 INPUT#7,Z$(c) : END FOR c : CLOSE#7 : c=c-1 : END DEFine
    280  DEFine  PROCedure  sw  :  IF  W$<=8  THEN  g$=W$-1  :
IF W$>=1 THEN h$=g$+2
    290 IF g$=0 AND t$="flp" THEN LET t1$="ram" :          ELSE
t1$="flp" : END IF  : g1$=1
    300 IF g$>0 THEN t1$=t$ : g1$=W$-1 : END IF
    310 END DEFine
    320  DEFine  PROCedure  Rx  :  INK#3,7  :  PAPER#3,0  :  AT  2,9  :
        PRINT t$;W$;"_" : FOR a=0 TO 18
    330 IF a<1 THEN a=a+a
    340  FOR  n=1+a  :  AT#3,a,0  :  PRINT#3,"{";CHR$(n+a+48);"}
";Z$(n+a+1)
    350  FOR  n=2+a  :  AT#3,a,38  :  PRINT#3,"{";CHR$(n+a+48);"}
";Z$(n+a+1)
    360 NEXT a : END FOR a : rx2 : END DEFine
    370 DEFine PROCedure rx2 : BLOCK 18,1,0,236,7 :        BLOCK
18,1,0,247,7 : BLOCK 4,12,20,236,7 :                BLOCK
413,1,20,236,7 : BLOCK 413,1,20,236,7 :             BLOCK
413,1,20,247,7 : BLOCK 2,12,456,236,7 : STRIP#2,7 : AT#2,24,4 :
INK#2,0  :  PRINT#2,"  ←  "  :  BLOCK  2,13,18,235,5  :  AT#2,24,8  :
INK#2,2  :  PRINT#2,t1$&g1$;"_"TO  71;t$&h$;"_"  :  AT#2,24,78  :
INK#2,0 : PRINT#2,"→ " :          BLOCK  4,12,432,236,7  :  BLOCK
6,12,438,236,7 :          BLOCK    18,1,438,236,7    :    BLOCK
18,1,438,247,7 :          BLOCK    2,12,436,236,5    :    BLOCK
458,1,0,235,5 : END DEFine
```

```
    380 DEFine PROCedure Uu : sx : pn=0 : CLSe : Uv : END DEFine
    390 DEFine PROCedure Uv : sw : Rx : k3 : pk : END DEFine
    400 DEFine PROCedure Uw : wx : k3 : pk : END DEFine
    410 DEFine PROCedure pk : k=k-48
    420 IF k=-21 THEN CLS#2 : nd
    430 IF k<=-1 AND k<>-14 OR k=77 OR k=78 THEN Uw
    440 IF k=-14 THEN Uw
    450 IF k>=160 THEN Uw
    460 IF k=144 THEN IF W$>1 THEN W$=W$-1 : Uu : ELSE  : iX : Uw
    470 IF k=146 AND t$="ram" THEN t$="flp" : a$=1 : Uu
    480 IF k=146 AND t$="mdv" THEN t$="flp" : a$=1 : Uu
    490 IF k=146 AND t$="flp" THEN t$="ram" : a$=1 : Uu
    500 IF k=152 THEN IF W$<2 THEN W$=W$+1 : Uu : ELSE  : iX : Uw
    510 IF k=156 THEN W$=W$+1 : Uu
    520 IF k=79 THEN CLS#2 : lne : tre=3 : WCh
    530 IF k=0 THEN GO TO 180
    540 END DEFine
    550 DEFine PROCedure wx : STRIP#2,2 : INK#2,7 : AT#2,24,29 :
PRINT#2,"  ERROR"TO  49;"ERROR  "  :  AT#2,24,36  :  STRIP#2,0  :
PRINT#2,"  WRONG  KEY";  :  INK#2,5  :  PRINT#2,"!  "  :  STRIP#2,7  :
INK#2,0 : B5 : CLSd : B5 : END DEFine
    560 DEFine PROCedure iX : AT#2,24,35 : PRINT#2,"invalid drive"
:                  B5 : PAUSE 10 : END DEFine
    570 DEFine PROCedure k3 : REPeat key
    580 k=CODE(INKEY$) : IF k>8 THEN EXIT key
    590 END REPeat key : END DEFine
    600  :
    610 DEFine PROCedure MAIN
    620  BLOCK  458,12,0,0,2  :  STRIP  2  :  AT  0,36  :  INK  7  :
PRINT"unzip"
    630 BLOCK 458,1,0,11,5 : BORDER 1,5 : BLOCK 458,1,0,235,5
    640 CLSc : STRIP 0 : INK 2 : AT 2,1 : PRINT" UN-ZIP " : sw :
cr=0 : Rx
    650 FREE=(PEEK_L(163856)-PEEK_L(163852)-1024)
    660  INK  5  :  AT  2,57  :  PRINT  FREE/1024;"  K";  :  INK  2  :
PRINT"ilobytes"; : INK 7
    670 k3 : pk
    680 LET ZIP$=Z$(k+1)
    690 CLS#2
    700   IF   k<c   THEN   EXEC_W   t$&W$&"_unzip";""&ZIP$(1   TO
LEN(ZIP$)-4)
    710 CLS#2 : BORDER#2,1,7 : wx : MAIN : END DEFine
    720  :
    730 DEFine PROCedure B1 : BEEP 100,10 : END DEFine
    740 DEFine PROCedure B2 : BEEP 200,20 : END DEFine
    750 DEFine PROCedure B3 : BEEP 900,20 : END DEFine
    760 DEFine PROCedure B4 : BEEP 900,40 : END DEFine
    770 DEFine PROCedure B5 : B3 : PAUSE 5 : B4 : END DEFine
    780  DEFine  PROCedure  lne  :  BLOCK#2,458,1,24,2,0  :  BLOCK
458,1,0,10,0 : END DEFine
    790 DEFine PROCedure nd : PAPER 7 : lne : B2 : tre=1 : WCh :
          END DEFine
    800 DEFine PROCedure CL : BLOCK 450,220,0,11,7 :  PAPER 7 :
    INK 5 :         AT 0,64 : PRINT'← shift TAB ' : AT 0,0 : PRINT"
TAB →"; : INK 0 : PRINT"  SELECT_DEVICE  UN-ZIP "TO 57;" EXIT " :
PAPER 5

    :               END DEFine
```

```
    810 DEFine PROCedure K4 : PAPER 7 : REPeat key
    820 ike=CODE(INKEY$)
    830 IF ike=9 OR ike=32 OR ike=252 OR ike=253 OR ike>47 AND
ike<58  THEN B3 : EXIT key
    840 IF ike>=58 AND ike<252 THEN B4 : K4
    850 END REPeat key : END DEFine
    860 DEFine PROCedure Pick
    870 IF ike=253 THEN B4 : PrvW
    880 IF ike=9 THEN B3 : NxtW
    890 IF ike=252 THEN B4 : B5 : GO TO 1180
    900 END DEFine
    910 DEFine PROCedure NxtW
    920 tre=tre+1 : IF tre=3 THEN tre=0
    930 WCh : END DEFine
    940 DEFine PROCedure PrvW
    950 tre=tre-1 : IF tre<0 THEN tre=2
    960 WCh : END DEFine

    970 DEFine PROCedure WCh : WINDOW#4,186,10,165,14 : lne : CL
    980 IF tre=0 THEN AT 0,7 : PRINT" SELECT_DEVICE " : K4 : Pick
: cj
    990 IF tre=1 THEN AT 0,22 : PRINT" UN-ZIP " : K4 : Pick : CLS
: MAIN
    1000 IF tre=2 THEN AT 0,57 : PRINT" EXIT " : K4 : Pick : GO TO
1180
    1010 END DEFine

    1020 DEFine PROCedure cj : pn=7 : CLSe : BLOCK 90,100,42,11,0
:                   BLOCK 88,99,43,11,7
    1030 PAPER 7 : PAPER 5 : INK 7 : AT 10,7 : S$=t$ : PRINT" [ ";
:                   INK 0 : PRINT S$&W$; : INK 7 : PRINT"_mode ] "
: AT 11,7 :                   PAPER 0 : PRINT u$ : PAPER 7
    1040 INK 0 : BEEP 100,29 : AT 2,9 : PRINT'"R" = ram1_' : AT
4,9 : PRINT'"F" = flp1_' : AT 6,9 : PRINT'"R" = mdv1_' : AT 8,9 :
PRINT'"O" = other'
    1050 s=CODE(INKEY$(-1))
    1060 IF s=9 THEN B3 : CLS#0 : BLOCK 90,112,42,11,7 : tre=1 :
WCh
    1070 IF s=253 THEN B4 : CLS#0 : BLOCK 90,112,42,11,7 : tre=3 :
WCh
    1080 IF s=77 OR s=109 THEN t$="mdv" : W$="1"
    1090 IF s=78 OR s=110 THEN t$="mdv" : W$="2"
    1100 IF s=70 OR s=102 THEN t$="flp" : W$="1"
    1110 IF s=71 OR s=103 THEN t$="flp" : W$="2"
    1120 IF s=72 OR s=104 THEN t$="flp" : W$="3"
    1130 IF s=82 OR s=114 THEN t$="ram" : W$="1"
    1140 IF s=84 OR s=116 THEN t$="ram" : W$="2"
    1150 IF s=87 OR s=119 THEN t$="win" : W$="1"
    1160 IF s=79 OR s=111 THEN WINDOW#0,90,30,69,84 : INK#0,5 :
           AT#0,0,7 : PRINT#0,"      ←_"; : INK#0,2 : AT#0,0,8 :
           INPUT#0,t$; : INK#0,5 : AT#0,0,11 : PRINT#0," _" :
INK#0,2 : AT#0,0,11 : INPUT#0,W$ : WINDOW#0,413,10,50,241
    1170 sx : B5 : nd : END DEFine

    1180 CLS#2 : INK#2,5 : AT#2,24,32 : PRINT#2,"@ PLATYPUS
Software"
```

JOHN B. PEGRAM
1126 STRADELLA ROAD
LOS ANGELES, CA. 90077

Info Copy for Frank Davis.

November 2, 1993

Dave Bennett
329 Walton Street Rear
Lemoyne, PA. 17043

Dear Dave:

I loaded my copy of the program "Z88COM_CLI" from your disk and could never get it to work on the Z88. I did not have much time to spend on it so waited until I obtained a copy of the same program in Tim Swenson's "Z88 Source Book". When I printed out the two programs I found four lines in my copy of your version that were different in the machine code assembly portion of the program. these are as follows for your information if these lines are erronious on your copy of this program.

```
Line   ASCII text
3280  LD HL, RESND:CALL PRTMSG:LD A,(SEC):CALL PRTBLKTBLKTRLE
3280  LD HL, RESND:CALL PRTMSG:LD A,(SEC):CALL CRLF

3380  LD HL,SNDCMP:CALL PRTMSGSGSRLF
3380  LD HH,SNDCMP:CALL PRTMSG:CALL CRLF

4310  LD IX,(HDLCM)CM)CLZ
4310  LD IX,(HDLCM):CALL CLZ

4390  LD IX,(HDLRX)X)XLZ
4390  LD IX,(HDLRX):CALL CLZ
```

Somewhere in the copying of this program the above errors crept in. When I transferred the uncorrected file to the Z88 and converted it into Z88com.bas it loaded with the remark "unknown error" and bombed out when I tried to run it. The corrected copy loaded without an error and works OK.

Sincerely yours,

STRUCTURAL DESIGN CALCULATION SYSTEM

The system consists of menu driven CLIs, an empty calculation file, procedure modules and analysis programs.

The CLIs call calculation modules from storage and integrate with the active calculation, append text files and program in/out data, initiate analysis programs and install printer drivers.

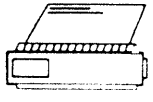
Leading particulars are entered in the empty file along with engineering data which is accessed by the modules, which form the substance of the active calculation.

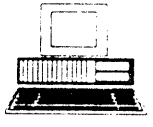
Procedure modules may be expanded by replication and titles presently available include pin shear/bending, lug, sections under comb. loading, bolts comb. loading, vector triangle, eye plate weld, principal stresses, beam procs., stability procs. and wind force.

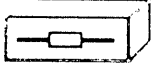
Programs may be run interactively or with plain text input data files. Titles include hydraulic ram, bolt group, polynomial, section props., vector polygon and redundant frame analysis.

All components of the system are available seperately. Full system operates on the Z88. Versions of the analysis programs are available for the BBC B & Master, with data editing under View.

For details, send SAE to:-
Frameworks (SDCS), 58 Charlotte Road, London EC2A 3QT.


Printer


Computer


Disk Drive


Modem

SwitZch Box for Z88 **£29.95** inc VAT & p+p

This switZch box and moulded connecting lead saves swapping leads in and out of your Z88. Plug in up to four of your existing Z88 peripheral cables into the sockets at the rear of the box and switch between them easily.

Specification:
High quality 4-position rotary switch no power required.
Female connectors with gold-plated contacts.
Fully shielded metal case with anti-skid feet.

Saves:-
Wear and tear on the Z88's serial port connector.
Trailing wires on the desk.
Identifying which cable to use with the Z88.

Rakewell Limited
24 Putnams Drive, Aston Clinton, AYLESBURY, Buckinghamshire. HP22 5HH. England
Tel. 0296 630617. Fax. 0296 630292

NEW!

for the Cambridge Z88

VERSION 4.0 ROM

JUST £24.95

Including VAT and UK Delivery. Overseas price is the same - the VAT offsets the extra postage costs.

SEND CHEQUE, POSTAL ORDER OR CREDIT CARD NUMBER & EXPIRY DATE TO:
WOODWARD TECHNOLOGY, PO BOX 15, BELPER, DERBYSHIRE. DE56 0XE.

Z88 Spares Service

Please note that Spares are supplied to Members Only.

Z88 Display - Quick replacement if your display is cracked. Please do not order if your display is not obviously damaged. **£99.00**

Z88 Keyboard - Complete assembled unit which just plugs in. **£34.95**

Battery Cover £2.25

Does your **Z88 Leg** flop about because the little lug has broken off? The leg can be easily replaced for just **£2.95**

We can supply new **Lower Cases** for **£4.95**, new **Case Tops** (the bit that says "Z88") for **£3.95**. The metal **Function Key Strip** is **£3.95**.

Z88 Manual - Issue 3 - **£9.95**

Card Slot Cover - the smoked plastic part with "1 2 3" on the front of the Z88 - **£2.95**

Z88 **Rubber Feet** - 2 front and 2 rear for **£1.00**
Screws for the Z88 - 5 for **£1.00**, 11 for **£1.75**

Expansion Bus Cover £2.50.

Back in Stock - Z88 Developers Notes
Just £39.50

Please note that all parts are supplied with full fitting instructions where applicable. Displays and keyboards are tested before dispatch. No parts are returnable if ordered in error. Send your orders to:

Z88 Users' Club, P.O. Box 15, Belper,
Derbyshire. DE56 0XE

Prices include U.K. Carriage & VAT. Overseas please add £7.50

IMPORTANT NOTICE: By opening your Z88 you will invalidate any remaining warranty on your machine. Neither the Z88 Users' Club nor Cambridge Computer Ltd will accept any responsibility for any damage, howsoever caused, to either your Z88 or its peripherals, caused in the course of fitting any spare parts.

RANGER

Z88 Price List

COMPUTERS	
Z88 (UK)	£189.95
Z88 (Ranger version)	£201.50
Z88 (Int. versions)	£215.95
Rugged Ranger	£299.00
Amstrad NC100	£169.36
Psion Series 3 - 256k	£212.72
RAM MEMORY	
128k RAM Pack	£32.95
512k RAM Pack	£87.95
1M RAM Pack	£136.13
EPROM MEMORY	
32k EPROM Pack	£16.95
128k EPROM Pack	£23.95
256k EPROM Pack	£51.02
EPROM Eraser	£38.95
Z88 COMMUNICATIONS	
RangerLink (25 pin)	£54.00
RangerLink (9 pin)	£54.00
RangerLink Upgrade	£30.00
PC Link II	£27.95
BBC Link	£8.95
Z88-to-Mac	£32.95
Parallel Cable	£18.95
Serial Cable	£7.95
BABT approved Modem	£207.00
CARRYING CASES	
Plastic Case	£6.95
Leather Case	£30.00
Topper	£8.95
Field Case	£70.00
POWER SUPPLY	
Mains Adapter (UK)	£8.75
Nicad Battery Pack	£23.00
Nicad Pack + Charger	£37.50

APPLICATIONS SOFTWARE	
Spell-Master	£52.00
SqueeZ~88	£44.00
RangerUniverse 2	£99/125
PC PipeDream (3.5")	£99.00
PC PipeDream (5.25")	£99.00
Vision-100	£175.00
Z-Term	£43.50
BJ10e Printer Driver	£20.00
PERIPHERALS	
RangerDisk 1- last few	£150.00
RangerDisk Mains Ad.	£17.50
RangerDisk PC Driver	£30.00
RangerDisk3 for NC100	£229.00
ADAM A/D Converter	£250.00
Barcode Wand (HP)	£250.00
CCD Scanner	£277.50
PRINTERS	
Canon BJ10ex	£250.00
BJ10 Sheet Feeder	£55.00
BJ10 Battery Pack	£39.00
BJ10 Ink Cartridge	£19.95
Diconix 150+ (P)	£345.00
Diconix Ink Cartridge	£13.00
BOOKS	
Z88 Developers Notes	£40.00
Z88 BBC BASIC	£14.50
Z88 Magic	£14.95
Z88 PipeDream (Dabs)	£14.95
Z88 User Manual (CCL)	£17.00
DELIVERY	
Next Day	£12.00
Post UK (Small)	£3.50
Post UK (Medium)	£5.00
Post UK (Large)	£6.50

All prices subject to VAT. Deliveries to export destinations can be made VAT free. Delivery charges include insurance. Next day delivery available to England and Wales. Please enquire for prices to offshore islands and Scotland. Postal service is by First Class Recorded Delivery to UK and BFPO. Please enquire for postal rates to other destinations.

ALL PRICES SUBJECT TO CHANGE WITHOUT NOTICE. E&OE.

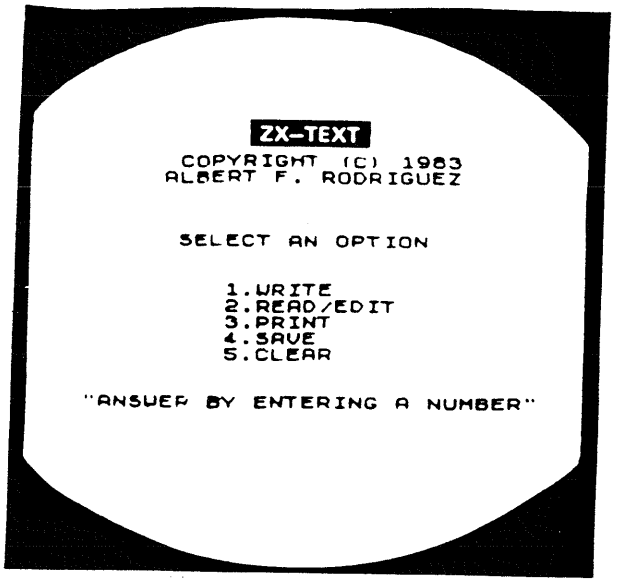
AFR SOFTWARE®

ALBERT F. RODRIGUEZ
PROPRIETOR

1605 PENNSYLVANIA AVE.
NO. 204
MIAMI BEACH, FL 33139

(305) 531-6464

ZX-TEXT



THIS PROGRAM IS ABOUT WORD PROCESSING. USING IT TO WRITE THE TEXT OF THESE INSTRUCTIONS SEEMED A GOOD WAY OF SHOWING WHAT IT CAN DO AND WHAT IT CAN BE USED FOR.

THE PROGRAM IS SELF-RUNNING. IF YOU STOP IT FROM EXECUTING IT CAN BE RESTARTED EITHER BY ENTERING "RUN" (IF YOU DO NOT CARE LOSING WHAT YOU HAVE BEEN COMPOSING) OR "GOTO 20" (IF YOU WANT TO AVOID LOSING WHATEVER YOU HAVE BEEN WRITING).

THE PROGRAM WAS MADE WITH CERTAIN LIMITS IN MIND. IT OFFERS 200 LINES AT 30 COLUMNS AVAILABLE PER LINE. AT AN AVERAGE OF 5 WORDS PER LINE IT SHOULD PROVIDE ROOM ENOUGH FOR ABOUT A 1300 WORD ESSAY, DEPENDING, OF COURSE, ON THE STYLE OF WRITING OF THE AUTHOR. IF EVER THERE SHOULD APPEAR A FLASHING, INVERSE VIDEO MESSAGE SAYING THAT THERE ARE NO MORE LINES AVAILABLE, THEN, YOU SHOULD SEND WHATEVER TEXT YOU HAVE WRITTEN TO THE PRINTER OR STORE THE SAME ON TAPE BEFORE CLEARING THE STORAGE AREA SO THAT YOU MAY BEGIN ANEW.

THE PROGRAM IS MENU-DRIVEN. IMMEDIATELY AFTER IT IS LOADED YOU WILL BE CONFRONTED WITH A GRAPHIC FRAME LISTING THE NAME OF THE PROGRAM WITH A PROMPT MESSAGE URGING YOU TO CHOOSE AN OPTION. THE CHOICE MUST BE MADE USING ONLY A NUMBER BETWEEN 1 AND 5, INCLUSIVE.

THE OPTIONS FEATURED IN THE MENU ARE WHAT THIS PROGRAM IS ESSENTIALLY ALL ABOUT. AN INDIVIDUAL EXPLANATION OF THESE OPTIONS AND THEIR FUNCTIONS WILL ALLOW YOU TO EASILY AND EFFECTIVELY USE THIS PROGRAM. AFTER YOU HAVE A CLEAR UNDERSTANDING OF HOW TO USE THE PROGRAM, IT WILL BE PRIMARILY UP TO YOUR IMAGINATION TO DETERMINE HOW YOU MAY APPLY IT. THOUGH, I WILL MAKE SOME SUGGESTIONS TOWARD THE END OF THIS PRESENTATION ABOUT HOW THIS PROGRAM DOES HAVE SOME PRACTICAL APPLICATIONS.

THE FIRST OPTION PERMITS YOU TO WRITE, STORE AND DISPLAY TEXT ON THE SCREEN IN SCROLL FASHION. THE LINE BEING WRITTEN IS COMPOSED IN THE ENTER LINE AT THE BOTTOM OF THE SCREEN. YOU HAVE HORIZONTAL CURSOR-CONTROL WITH DELETE CAPABILITIES WHEN COMPOSING A NEW LINE OF TEXT. YOU MAY ALSO ENTER READ/EDIT MODE FROM THIS OPTION BY KEYING IN "*E". HOWEVER, WHEN YOU DO THIS, DO NOT USE THE EDIT COMMAND "*G" IMMEDIATELY AFTERWARDS OR BEFORE YOU HAVE USED SOME OTHER EDIT COMMAND.

WHEN THE LINE BEING TYPED REACHES THE 30TH COLUMN THE SCREEN BLINKS. THIS FEATURE IS SIMILAR TO THE BELL ON A TYPEWRITER WHICH SIGNALS THAT IT IS TIME TO ENTER A LINE OF TEXT AND THEN BEGIN A NEW LINE. AFTER A LINE IS ENTERED IT IS STORED IN AN ARRAY FOR FUTURE REFERENCE. BLANK LINES ARE ENTERED BY PRESSING THE ENTER KEY AND ARE STORED IN THE SAME WAY. IF A LINE LONGER THAN 30 COLUMNS IS ENTERED ONLY THE FIRST 30 CHARACTER LOCATIONS OF THE LINE WILL BE STORED AND PRINTED ON THE SCREEN. BEFORE ANY LINE IS WRITTEN AND

ENTERED YOU CAN CAUSE THE PROGRAM TO REVERT TO THE MENU BY KEYING "*M" AND THEN ENTER. THE PROGRAM WILL ALSO RETURN TO MENU IF THE LINE COUNTER "K" IS > THAN 200. UNDER THE LATTER CONDITION THE INVERSE VIDEO PROMPT WILL FLASH INFORMING YOU THAT THERE ARE NO MORE LINES AVAILABLE. THIS CONDITION CAN BE REMEDIED BY EXERCISING THE OPTION "CLEAR".

AFTER YOU ENTER A BODY OF TEXT YOU MAY PROOFREAD IT. THIS IS WHERE THE "READ/EDIT" OPTION COMES IN. ON A PER LINE BASIS THE TEXT THAT WAS WRITTEN WILL BE DISPLAYED ON THE SCREEN IN SCROLL FASHION. AFTER A LINE IS PRINTED YOU HAVE 3 SECONDS TO READ AND DECIDE IF THE LINE NEEDS EDITING. IF IT DOES NOT THEN THE NEXT LINE WILL BE PRINTED, AND SO ON. IF IT DOES NEED EDITING YOU THEN PRESS THE ENTER KEY TO OBTAIN THE EDIT MODE.

THE TEXT EDITOR IS THE HEART OF THE PROGRAM. IT HAS 10 DIFFERENT FUNCTIONS. FIRST, THERE IS THE FUNCTION "CONTINUE", WHICH IS GENERATED BY ENTERING "*G". THIS FUNCTION ALLOWS THE TEXT TO COMMENCE BEING READ AFTER IT WAS STOPPED FROM SCROLLING UP THE SCREEN WHEN EDIT MODE WAS ACCESSED. SECOND, YOU MAY "REPLACE" AN ENTIRE LINE OF TEXT BY ENTERING "*R". AFTER THIS SYMBOL APPEARS ON THE SCREEN YOU TYPE AND ENTER THE NEW LINE. THIRD, THERE IS A "CHANGE" FUNCTION THAT PERMITS ONE OR MORE WORDS, SYMBOLS OR SPACES TO BE MODIFIED. AFTER THE COMMAND "*C" IS ENTERED YOU ENTER THE BLOCK OF CHARACTERS YOU WANT TO REMOVE FROM THE TEXT. NEXT YOU PLACE THE CURSOR UNDER THE SECOND CHARACTER LOCATION, FIRST TO THE RIGHT, FROM THE PLACE ON THE LINE THAT YOU WANT TO REPLACE THE OLD WORD OR EXPRESSION WITH A NEW ONE. AFTER THE CURSOR IS PROPERLY LOCATED THE SYMBOL "/" IS TYPED WITH THE NEW WORD OR EXPRESSION IMMEDIATELY FOLLOWING. NOW KEY ENTER AND YOU WILL HAVE YOUR CORRECTION PROPERLY RECORDED. FOURTH, YOU MAY "DELETE" A LINE THAT IS FIRST ABOVE THE ENTER LINE BY KEYING "*D" WHILE IN THE EDIT MODE WITHOUT LEAVING ANY GAPS BETWEEN LINES. FIFTH, A NEW LINE CAN BE "INSERTED" AFTER ONE LINE OR BETWEEN TWO LINES WITHOUT ERASING ANY PREEXISTING LINES OF THE TEXT. YOU TYPE IN THE SYMBOL "*I" THEN AFTER IT APPEARS TYPE AND ENTER THE CONTENTS OF THE NEW LINE. SIXTH, BY ENTERING "*T" THE TEXT WILL BEGIN TO BE DISPLAYED FROM THE START. SEVENTH, ANY LINE BEING WORKED ON BY THE EDITOR MUST ALWAYS BE THE FIRST LINE ABOVE THE ENTER LINE. TO ASSIST IN PUTTING A LINE INTO THIS PLACE FOR EDITING A "LIST" FUNCTION IS AVAILABLE. AFTER "*L" IS KEYED AND ENTERED THE LINE NUMBER OF THE FIRST LINE ABOVE THE ENTER LINE IS DISPLAYED. IT IS THEN A MATTER OF COUNTING FROM THAT LINE NUMBER UPWARD OR DOWNWARD TO DETERMINE THE NUMBER OF THE LINE YOU ENTER IN ORDER TO BRING IT INTO POSITION FOR EDITING. THE LINE NUMBER YOU TRY TO LIST CANNOT BE < THAN 1, > 200 OR > THAN THE LAST LINE THAT WAS WRITTEN WHEN IN WRITE MODE OR ELSE THE PROGRAM WILL CONTINUE TO ASK FOR A NEW LINE NUMBER ENTRY. IF YOU KEY IN 0, THEN, THE PROGRAM RETURNS TO WRITE MODE. EIGHTH, BY KEYING "*W" AND THEN ENTER THE COMPUTER RETURNS TO WRITE MODE WITHOUT ERASING THE TEXT DISPLAYED ON THE SCREEN. IT WILL BEGIN TO WRITE A NEW LINE AFTER THE LINE THAT WAS LAST WRITTEN. NINTH, THE PROGRAM INCLUDES A SPLIT-SCREEN FEATURE WHICH IS AVAILABLE WHEN THE COMMAND "*SS" IS KEYED IN DURING EDIT MODE. THIS FEATURE PERMITS TWO HALF-SCREENS IN ONE FULL PICTURE SCREEN. TEXT CAN THEN BE COMPARED ON THE BASIS OF EVERY 10 LINES WITH THE SAME EDIT FEATURES JUST DESCRIBED BEING AVAILABLE. WHEN IN SPLIT-SCREEN MODE YOU OPERATE EACH HALF-SCREEN EXACTLY AS YOU WOULD ONE FULL-SCREEN. THE ADVANTAGE OF SPLIT-SCREEN LIES IN THAT HIGHER AND LOWER PORTIONS OF A GIVEN TEXT MAY BE COMPARED TO FIND GRAMMATICAL OR SYNTACTICAL ERRORS. IT MAY ALSO BE USED TO READ/EDIT TWO DOCUMENTS SIMULTANEOUSLY. IF YOU WANT TO MOVE THE TOP-HALF YOU ENTER "T" AND TO MOVE THE BOTTOM-HALF YOU ENTER "B" AFTER 10 LINES OF

TEXT ARE PRINTED ON THE RESPECTIVE SCREEN. AFTER EVERY 10 LINES THAT ARE PRINTED THE CURSOR APPEARS AT THE BOTTOM OF THE SCREEN. AT THIS TIME (AND [redacted] WHEN THE ENTER KEY IS DEPRESSED TO ACCESS EDIT MODE) YOU HAVE THE CHOICE OF MOVING THE TOP OR BOTTOM-HALF OF THE SCREEN OR STARTING FROM THE BEGINNING TO VIEW THE TEXT BY KEYING "N" OR REVERTING TO THE MENU BY ENTERING "*M". FINALLY, WHEN YOU WANT TO RETURN TO THE MENU YOU CAN DO SO BY ENTERING "*M" WHEN IN EDIT MODE.

THE REST OF THE OPTIONS ARE THE "PRINT", "SAVE" AND "CLEAR" FUNCTIONS OF THE PROGRAM. THEY ARE QUITE BASIC IN WHAT THEY DO AND HOW EACH DOES WORK. HENCE, NO EXPLANATION IS NEEDED AS TO HOW TO USE THEM. JUST ENTER THE DESIGNATED NUMBER AND WAIT UNTIL THE MENU APPEARS ON THE SCREEN AGAIN. BE SURE TO TURN ON YOUR TAPE RECORDER BEFORE YOU EXECUTE THE SAVE OPTION.

THE PROGRAM IS, OF COURSE, COMPATIBLE WITH EITHER THE ZX OR TIMEX PRINTER. WITH THE ADDITION TO THE COMPUTER OF EITHER A SERIAL OR PARALLEL INTERFACE ALONG WITH A MORE SOPHISTICATED DOT-MATRIX PRINTER THE APPLICATION POSSIBILITIES FOR THIS PROGRAM ARE EVEN GREATER. THE QUALITY OF OUPUT GENERATED BY THE PROGRAM IS SOMETHING THAT DEPENDS ON THE INDIVIDUAL TASTE OF THE USER. HOWEVER, THE CONTENT OF SAID OUPUT (I.E., WHAT THE PROGRAM CAN BE USED FOR) IS GENERALLY OBVIOUS TO ANY USER.

LETTERS, REPORTS, ARTICLES, MEMOS, STANDARD FORMS, MEDIUM DISPLAYS, INSTRUCTIONS, NEWS BULLETINS, CLASSIFIED ADVERTISEMENTS, GRAPHS AND CHARTS, MANUSCRIPTS, LISTS OF CUSTOMERS OR FRIENDS OR CLUB MEMBERS, TELEPHONE AND ADDRESS DIRECTORY . . . ETC., ARE SOME OF THE MORE PERTINENT, REAL-LIFE APPLICATIONS THAT COME IMMEDIATELY TO MIND.

BESIDES ITS PRACTICAL WORTH USING THIS PROGRAM PROVIDES CERTAIN ADVANTAGES OVER THE MORE CONVENTIONAL WAYS OF DOING THE ABOVE SAID TASKS. SAVINGS IN TIME, PAPER, INK, PRINTING, PHOTOCOPIES, CORRECTING MISTAKES AND ADDING AFTERTHOUGHTS ARE JUST SOME OF THE PROCESSES HANDLED MORE EFFICIENTLY AND WITH LESS TROUBLE THAN DOING THEM EITHER THROUGH HANDWRITING OR USING A TYPEWRITER.

Word-Processing

T/S-TEXT 2000™
ZX-TEXT™

COPYRIGHT (C) 1983
ALBERT F. RODRIGUEZ

PRICE: $19.95 IN CASSETTE
COMPUTERS: ZX81, TS1000, TS1500, TS2068
LANGUAGE: SINCLAIR BASIC
ROM/RAM: 8K/16K
LOAD NAME: "TEXT"
LOAD TIME: 6.03 MNTS.
PROGRAM LISTABLE: NO
PROGRAM LISTINGS: YES, $6.50
EXTRAS: RECORDER AND PRINTER
PROGRAM BYTES: 14,823
SYNTACTIC SUM: 424,665

A.F.R. SOFTWARE®
1605 Pennsylvania Ave., # 204
Miami Beach, FL 33139

(305) 531-6464

FLORIDIANS ADD SALES TAX
DEALER INQUIRIES WELCOME

New Sinclair Newsgroup on USENET
By Tim Swenson

A new newsgroup is now available to those that read USENET News on the Internt. This new newsgroup comp.sys.sinclair is for discussions on all Sinclair computers. It was started by those interested in the Spectrum and it's emulators, but can be used by all to discuss any Sinclair computer, including the Z88 and Sir Clive himself.

Below is a few selections from recent postings to the newsgroup:

Has anybody here ever used their Sinclair for something other than what it was originally intended?

Both a friend and I had Sinclairs, and after we'd gotten bored with them, his father picked up a Timex for $15 in NYC while on business. Since the Timex had a whopping 2K of memory, instead of only 1K, we turned his Sinclair into a Dalek (enemy robot from Dr. Who).

The design was extremely simple, but impressive considering our lack of $$ and that we were only 16 at the time. The robot had two servos which we glue-gunned big rubber wheels to, and a plunger glued to a micro switch. Using 74LS00s for address decoding and power transistors to drive the servos, if you POKEd a 1 to any address between 32k and 48k, it would drive the left wheel forward about an inch, POKEd a 2 would drive the other wheel (and yes, poking 3 would drive them both). Peeking and checking bit 3 would tell you if the micro switch was open or closed.

The program would move the left and right wheels together, and then check to see if the switched was closed (dalek had bumbed into something). If the switch was closed, it would randomly chose to move only the left or right wheel, enough to turn the dalek 90 degrees. Then it would continue to lurch forward.

We later modified this thing to operate off of a 12volt gelcell, and made a little box inside the four foot tall dalek which enabled us to start the sinclair and load the program, then disconnect the monitor and tape recorder and let the thing wobble around the room until the batteries ran out (about 1 hour). If we'd had an LCD TV, I'm sure we would have glue-gunned that in as well.

Dominic Richens (dominic@oeg.carleton.ca)

My dad had the field engineers of General Electric's Installation and Service Engineering Division using TS 2068's with the 1967 ASME steam tables & custom software analysing medium steam turbine performance testing. The binary precision of floating pt numbers was 32bits as opposed to the IBM floating pt precision of 21 to 23 (remember the old 1.000001^10 test?), so double precision wasn't needed.

I helped with some convergence problems going from H,S to temperature or something like that. I got the number of

iterations it took to converge down from somewhere between 60 and infinity to 6 or less.

It all started when they took his link to the GE timeshare machine (a teletype and acoustic coupler) away (it was two steps from his desk) and replaced it with a terminal room two floors up. Any of you that have had to use terminal rooms know that you often forget that one sheet of paper and have to go back to get it - two floors down. This was 1983 so pc's were out, but expensive, and the ts2068 had failed (no s/w, the bane of all these "great" pieces of h/w), and was being sold for $135 including cassette tape recorder and 40 column thermal printer. It was great, where a turbine test was all day taking data, that night in a motel making quick numbers from a Mollier Chart to present to the plant manager the next day, now within an hour after the test was over he could place the preliminary report on the plant manager's desk and head home.

When he told the field service managers that he had a computer and software to make their FSE's lives easier, they said "how much?" (the field service units were profit and loss centers and my dad worked in headquarters i.e. overhead) They were expecting him to quote a price in a pc or clone, but when he said $135, they said "Oh, I'll take 2".

If any of you ever get to Cape Hatteras in North Carolina, USA, stop at the Canadien Hole about a mile north of Buxton and look for the "Microvan" (Plymouth Horizon) with the anemometer on a windsurfing mast - the data acquisition system is being run by one of those ts2068's. There is also a setup at Fox Watersports, a surf shop in Buxton. That one survived Hurricane Emily last august, and when the National Weather Service anemometer turned out to have died just before the worst of it, the NWS and National Hurricane Center asked my dad for his data. Apparently they are running fourier transforms on it - this is the first time they have had hurricane wind speed data in digital format! It's available for anon-ftp from moe.tn.conell.edu if you're interested.

In compiled basic, with a decent a/d chip (the first one used the A0 line for a clock for successive approximation), and compiled basic, the setup reads data at around 100 Hz.

My favorite memory is writing a "capture the mouse" game where you draw lines to force a bouncing pixel into a corner - of course the interpreted basic was too slow and this was before we had the compiler, so I tried my hand at machine code - the first few times you jump into the machine code and the machine hangs, or reboots, after a few debugging cycles the program seemed to be working, but I couldn't see the mouse. So I'm digging through the code trying to find where I'm NOT writing the mouse to the screen, and suddenly I say, "maybe it is there". I turned off the lights in the room, turned up the contrast on the crummy b/w telly, and there, about every dozen pixels or so, and seemingly simultaneously in several places at once because of phosphor persistence, is the mouse flying around the screen at warp speed!

Brian Carcich (carcich@cuspif.tn.cornell.edu)

I guess everybody has their own Golden Age, when programming was purely for fun. My experience has been that each home computer that I buy is more powerful than the last, but I do less actual programming on it than the one before.

So of course I recall fondly the ZX-81, the machine that I fitted with a surplus movable key keyboard (with attached Atari 2600 type joystick) and with which I wrote several cool arcade style games which were never run on anyone else's machine. They were all in Z80 assembler of course, and WOW was I impressed when the ole ZX managed to handle 25 concurrently moving robots on the screen at once (albeit a bit jerkily).

The assembly code was stored as BASIC REM statements, and was still so big (I only had 16k) that I had to assemble the program in two parts, write down the addresses in each part that were visible to the other half, and manually link them together. Sometimes a 'build' of the program took over 45 minutes of tape swapping and cursing.

On one memorable occasion I managed to blow a hole in the 'roof' of the screen and some of my robots escaped up (actually I think it was downwards in the address sense of the word) and off the screen. So I trundled my player dude (who was an 'A' because it had two legs) over to the hole and started shooting bullets up and out of the screen too. "BLAM BLAM BLAM, Get back here! Suck hot asterisks, you little cowards!" Of course, my executing code (and variables) were all somewhere up there too. The screen shook violently several times, and everything became skewed, as if each screen row had grown by a few characters. But the ZX was tough to kill, as the game kept running and my bullets could run wild in all directions now. So I shot up the joint, and after a few more screenquakes (which were quite amazing to watch, because they played havoc with the TV's sync pulses) the whole thing locked up. I almost added the whole scenario into the game as a 'doomsday' option.

Ah, the luxury of it all: a trusty machine, a wild imagination, youthful stamina and way too much time on your hands.

"The Sinclair ZX-81: Tough as nails and twice as fast!"

Mike Chapman (mike@snowhite.cis.uoguelph.ca)

QL Hacker's Journal
Supporting All QL Programmers

Timothy C. Swenson, Editor
5615 Botkins Rd.
Huber Heights, OH 45424
(513) 233-2178

swensotc@p2.ams.wpafb.af.mil

QL SURVIVORS SOURCE BOOK

2ND EDITION -- This book contains worldwide lists of all QL BOOKS; PERIODICALS; BULLETIN BOARDS; USER GROUPS; HARDWARE and SOFTWARE DEALERS; PUBLIC DOMAIN and SHAREWARE SUPPLIERS. Nowhere else can you find all of this data in one easy to use source. This is a work of art by the same great folks that brought you IQLR, and offered by UPDATE Magazine. When we sell out of current stock it will be gone. It is now available for only $10.00 US$, and this now includes all postage and handling in the North and South American areas. Order now and it will be shipped to you by either First Class Mail or Priority Mail; no more waiting around for slow UPS delivery! Whatever it is you are looking to buy for your QL, this will tell you where to look for it, along with phone and fax numbers, as well as addresses.

Z88 SOURCE BOOK

This book was compiled by Tim Swenson and published by UPDATE Magazine, and was designed to be a good and concise reference book on the Z88; how to interface it with the rest of the worlds computers; what products are available for it and where to find them. It will not replace your Z88 user guide, but will instead supplement it, and fill in areas that were missing by putting all this information in one easy to use book. When you order it we also include a disk of utilities and programs that are discussed in the book. This is available in the following formats: QL in 3 1/2 1440 or 2880 sector disks; QL in 5 1/4 720 or 1440 sector disks; IBM 360K 5 1/4 disks; IBM 720K 5 1/4 disks; IBM 720K 3 1/2 disks; IBM 1.4 meg 3 1/2 disks. Please let us know with your order which type and size of disk you need. The price for all of this is $7.00 US$, which includes P. & H. in North America, and elsewhere $7.00 US$ plus $2.00 US$ for the additional postage we have to pay. We will accept for foreign orders the cash equivalent, plus 10% for conversion, if you are unable to obtain a money order or travelers check in US$. We want to make this information widely available, so we are trying to be as flexible as possible.

TS2068 UPDATE ISSUE DISKS

These disks contain at least one major piece of software written specifically for disk drive and are guaranteed to be worth the money. The rest of the disks are usually filled with various utility programs taken from the issues of UPDATE, shareware or public domain. On most of these, half the money goes to the author and is meant to encourage new programming for the TS2068 that makes use of the various disk drive systems. To have your particular disk system supported here requires that someone write or alter the software from one system to the other. We are always open to your help and suggestions, but have limited time and programming resources available here. The prices are as listed beside each piece of software. We can provide all but 3" disk formats as far as size and disk density. We accept cash, checks, money orders and C.O.D.

1) THE WIDJUP COLLECTION, contains most of the popular programs formerly offered by WIDJUP and written by the late Bill Pedersen. This is a two disk set, and does not include his CAD Program. It contains editors, printer drivers, games,TS2068 tutorials, etc. This is a new release and we will have more about it next issue. In Oliger or Larken disk formats. The price is $20.

2) WIDJUP'S CAD PROGRAM, a long time favorite that will give you professional results from your TS2068 in the area of computer aided design and the development of printed circuits. With the right graphics it has been also used to print a page for desk top publishing, or computer art. It requires no expanded memory and is available for the following setups: (a) Oliger, for either the Olivetti Ink Jet printer, or for IBM compatible printers. (b) Larken for IBM compatible printers or for the Olivetti Ink Jet printer. Please specify disk size, format and printer type. The price is $20.

3) OLIGER DISK DRIVE BBS PROGRAM, this creates a single user BBS program, with several message bases, E-mail, and SYSOP Chat area. We have also added many other Oliger disk programs to this collection, as well as some playtime. This was written by Paul Holmgren. The price is $20.

4) 24-PIN BIT IMAGE GRAPHICS FOR 24-PIN OR BUBBLE JET PRINTERS, for Epson emulation modes, by Larry Crawford. This program takes the mystery out of graphics and some of the newer printers out there on the market. We also include some extra software with this one, and for just $15. It is available in Larken and in Oliger disk formats.

Needless to say we are always interested in a new issue disk we can present here for our readers, so those who are out there writing programs, send them to us to look at. This helps to keep the TS2068 alive. It is also a way to pick up some pocket money. We usually make royalty payments twice a year based on previous sales.

QL ISSUE DISKS

These disks contain at least one major piece of software written specifically for disk drive (exceptions are noted) and are guaranteed to be worth the money. The rest of the disk space is often filled with various utility programs taken from the issues of UPDATE thrown in. Half of the money from the sale of these programs goes to the software author to encourage the creation of more QL software for our readers. Some of the programs take advantage of Tool Kit 2 by Tony Tebby, and need it to work. If that is the case, it will be listed in the program description. The price is listed with each program. They are available in both 3 1/2 and 5 1/4 disk formats. This is for 360K, 720K and 1.4 meg. If needed on an ED disk, then you must provide. We accept personal checks, company checks, money orders, cash, and C.O.D. orders. No charge cards, please.

1) CABLE ARCHIVE ISSUE DISK - Written by Bill Cable. Contains many useful ARCHIVE programs that work on any ARCHIVE database. The titles include: DIR (directory within ARCHIVE), SCAN (quick database display and print), FREQ (Frequency distribution of a field), SPLIT (split 1 database display and print), JOIN (join 2 databases into 1), REFIELD (redefine field names), REPLACE (replace text within a database), MATCHER (find duplications in a database), WINDEX (word index any text file), GROUP1, GROUP2, GROUP3 (useful procedures from UPDATE articles), QUERY (interrogate any database). Also included are extensive DOC files about the programs and about ARCHIVE in general. The price is $20.

2) QLUSTER 5s109 ISSUE DISK - This great program by Al Feng provides you with many utilities to handle and unclutter your disk and microdrives, and it now supports sub-directories. Some of the features concern COPY, DELETE, FORMAT, VIEW, as well as extended use of some TK2 commands (needed for this program). These are TURBO compiled to add speed to the program. It will multi-task and allows you to use minimal keypresses to get the job done. This program has had some major upgrading and bug fixing over the years and is Minerva compatible. The price is $15.

3) QLuMSi DOS 4.30 ISSUE DISK - Al Feng also extensively updated this recently. This is both an MSDOS simulator and a front end program for the QL. It also has other programs on the disk to enhance file management and the cloning of other programs. Educational as well as useful. The price is $20.

4) QL COMPENDIUM ISSUE DISK - Now a four disk set of useful programs and utilities taken from UPDATE, plus the latest file compression and decompression utilities, Tcopye, Quad Format, C utilites, E-forth, and alien disk file transfers, and more. The best of UPDATE, shareware and public domain. Still only $20. $5 for an update for past purchasers.

We here at UPDATE encourage all of you programmers out there to submit to us possible future Issue Disks, with DOC files. You won't get rich, but you will possibly make some money and get to help others. Royalty payments are usually twice a year, based upon previous sales.