UPDATE!



Sir Clive Sinclair

UPDATE! Magazine P. O. Box 1095 Peru IN 46970

$4.50/Issue $18/Year October 1994

* UPDATE COMPUTER SYSTEMS MAGAZINE *

UPDATE COMPUTER SYSTEMS is Edited and Published by Frank and Carol Davis of P.O. Box 1095, Peru, IN 46970 USA. The phone number is 317-473-8031 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 $40 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 DIRECTORY FOR OCTOBER 1994

The computer that an article concerns is marked by using the following mark in the first column of the directory: TS= article for the TS2068 or Spectrum; QL= article for the QL or QXL; ZX= article for the TS1000, ZX81, TS1500; 88= article for the Z88. GI means article of general interest.

Front cover art by Abed Kahale of CATUG, the Chicago Area Timex User Group

Front Covers --Design and Basic Magazine Information
Page 1 --GI--Directory of Issue Contents
Page 2 --GI--Editorial by Frank W. Davis
Page 3 --TS--LogiCall Review by Abed Kahale
Page 4 --ZX--World Map --Part 2 by Paul Holmgren
Page 12 --GI--Mechanical Affinity & Update! Ads
Page 13 --GI--Modems or Magic by Abed Kahale
Page 16 --GI-- RMG Ad for Rod Gowen
Page 16 --TS--The Best of the Plotter Ad
Page 17 --ZX--Serial Printer Output for ZX81/TS1000 by Bob Berch
Page 19 --TS--How To Build A Larken Dock Board by Les Cottrell
Page 23 --GI--Wanted/For Sale/or Help Needed Ads
Page 24 --QL--Review of Chris Boutal's Genealogist 3 by Hugh Howie
Page 27 --QL--QLAMBer Update by Al Feng
Page 28 --QL--T/SNUG QL Public Domain Library
Page 29 --GI--QBOX USA Ad
Page 29 --GI--Computer Classics Ad
Page 30 --QL--Cable Column by Bill Cable --Archive Series Part 18: Form Ed -- A Form Editor & Printer for Archive Databases (2nd and final part)
Page 36 --QL--Pertinent Points on QL Disk Drives by Frank Davis
Page 38 --QL--XTricator, Version 1.75 by Gary Norton, a Review
Page 41 --QL--A Review of QLerk by Eliad P. Wannum (A Financial Program for the QL)
Page 42 --GI--Report on the Dayton Computerfest in August by F. Davis
Page 43 --QL--QLerk Ad by Wood & Wind Computing
Page 44 --88--Z88 Flo-Chart by Mike Fink of Domino Cubes
Page 45 --88--Domino Cubes Z88 Ad
Page 46 --88--Z88 ROM & Internal RAM Upgrade by Dave Bennett
Page 47 --GI--Timex Publication Index, part 1 by Paul Holmgren
Back Covers --TS2068 & QL Update Magazine Issue Disks

THE OCTOBER EDITORIAL
by Frank W. Davis

Once again I set fingers to the keyboard. I often realize just how much I have come to rely on word processors to do my letters and for other text items. It is so much better to see the whole concept and be able to change it in a matter of seconds...and to make as many copies of it as I wish. For someone who likes to write, this is indeed a minor miracle! I do, however, find I do at times go ahead and write letters out by long hand, just to keep my handwriting skills from waning. It can indeed happen, just as any skill left unpracticed can atrophy with little or no use. We have all heard the old saying that "it is just like riding a bicycle, you never forget how". Well, how many of you have tried to ride a bike lately? I fear that the adequate tennis player that I was in my youth is but a memory now. I was today listening to a public radio program from Purdue University that was discussing this subject, as well as memory and motor skills. It appears that we need to exercise as many different skills as we can in order to stay sharp as we age. Doing so can indeed make us better at some things than we were in our youth, provided we never gave it too much effort in our youth. They seemed to feel that the use of things such as bingo and learning computers were both excellent for the aging mind. Perhaps all of us are getting more out of computers than some realize.

I have to still admit a special fondness for my Sinclair computers. I have had them long enough to feel comfortable and skillful with them. It is not that the mere thought of an IBM compatible, or a MacIntosh has never crossed my thoughts. The MAC is beyond what I wish to pay for a basic skills computer, whereas I find Windows to be a clumsy environment to work with and the IBM cannot multi-task in the same manner as a QL. I have also been a user of AMIGA computers for years. I find them to be excellent, but not as quick as the QL at multi-tasking. They are great for graphics and stereo sound (two big weak points on the Sinclairs), but they are not easy to program. The TS2068, Z88 and the QL are easy to program on and almost a breeze to modify programs on. The QL is now coming more and more into its own. Those who think no new programs are available for the TS2068 are probably not active members of a user group, or do not pay close attention to what is written in UPDATE or the quarterly put out by TSNUG. To find out about new programs and uses for the Z88 you would all do wisely to contact Domino Cubes. Mike has a lot of enthusiasm for this machine and can get you pointed towards some power processing with this little two pound wonder.

If you have a TS1000 and are looking for a new peripheral for it contact RMG, Mechanical Affinity, or place a WANTED AD here free (for subscribers) in UPDATE. If you need to expand your TS2068, do the same, or get a disk drive system from John Oliger Company. You can, if skilled enough or know someone who is, follow the instructions in this issue and build a Larken board for that disk drive setup. For the QL there is so much happening that you should call Mechanical Affinity or IQLR (those who subscribe or want to only) and find out just what all is new. There is the Gold Card, Super Gold Card, two types of hard drive interfaces, and a new improved graphics card coming out in about another month. This does not even mention all the new software. We are closer to state of the art.

LogiCall Review

Auxiliary Operating System AOS-LKDOS *by Abed Kahale*

Developer: Bob Swoger

LogiCall contains all the commands required to run LK-DOS (TS-2068 & Spectrum). It prompts for a command and executes these commands with a minimum of typing.

Moves from drive to drive with 2 keystrokes; moves into and out of a word processor, database, spreadsheet, terminal program and other programs using a few key-strokes. Displays word processor files and pictures on the screen without running any program. FORMAT, MOVE, ERASE, CAT, POINT, and VERIFY keys all work without the use of the "RAND USR 100: or PRINT#4: " that have to be typed using LKDOS alone. It really proves itself with multi-drive system and will be of great benefit even with a single drive. And, it is not memory hungry, it only occupies one disk track. It does complete the LarKen Disk Operating System.

LogiCall comes in two versions, 5.0 for all including ROMSWITCH and 5.2 for Spectrum ROM in the doc port. OUT 244, 3 is not required.

A complete file management system, more accurately a HUMAN INTERFACE (you and the machine) shows the years of refinements.

Holding down the ENTER key (or the ENTER and the 'J' keys for the RAMDISK) at power up, displays the CATalog and the `Drive?` prompt. ENTER the disk drive number or 'T' for tape or just ENTER to display the `Program?` prompt.

ENTERing 'H' or '?' at either prompt, displays two screens HELP menu.

ENTERing one of the following at "`Program?`"

'9' displays CATalog.

'V'erbose displays the complete CATalog.

'B'rief displays a CATalog that only shows programs that can be LOADed and RUN.

'N' at the `scroll?` displays the `Program?` on CATalogs that scroll off the screen.

'W' LOADs word processor (TASWORD II)

'T'erminal LOADs terminal (Mterm II) program.

'5' or '8' RENAME "*old*", "*new*". ENTER the old name "*NMI-S2.CM*" to be renamed, then the new "*name.B1*".

'7' or 'E'RASE -- ERASE ".....". ENTER the filename and extension.

'A'dds AUTOSTART to a disk.

'S'ave SAVEs "L.B1".

'Z' COPYs to the TS-2040 printer.

'C'OPYs to a large printer.

'N'EW activates AUTOSTART again.

'0' or 'F'ormat LOADs FORMAT.B_

'6' or 'M'ove LOADs MOVE.BL

'R' VERIFYs for Cyclic Redundancy Check errors.

'K' SAVEs SCREEN$ to disk.

'Q'uits or 'E'xists to BASIC where the program can be customized.

'0, 1, 2, 3 or 4' moves between drives.

At the `Program?` prompt, hitting ENTER without typing a filename activates the SCAN LOAD mode. The space bar or any key in the lower key rows advances a BRIGHT BAR down the screen, the top row of keys will send the BRIGHT BAR back toward the top. ENTER LOADs the BRIGHTened program. The arrow keys do work as normal or without shift.

ENTERing a name with an extension of '.C$' at the `Program?`, LogiCall displays it as a SCREEN$. While ENTERing a name with an extension of '.Cm' or '.CT', it displays MSCRIPT, TASWORD II or SPECTRATERM word processor files directly from disk.

LogiCall disk ensemble includes VUFILE, VUCALC, TASWORD II, disk/tape Records Base and MTERM II among others.

It is like an upgrade from a gear-shift to an automatic transmission with overdrive.

LogiCall is available from RMG Enterprises and from Mechanical Affinity.

```
    ; routine to scroll screen up, down, right, or left according
to DIRN
    ; and fill in the area vacated by scrolling with the map data

    409C 3E67     MSCR LD A,67
    409E 327241          LD (4172),A    ;make instructions at 4171
and
    40A1 32BE41          LD (41BE),A    ; at 41BD be RRD to access
top half of map
    40A4 3E6F          LD A,6F
    40A6 327F41          LD (417F),A    ;make instructions at 417E
and
    40A9 32CB41          LD (41CB),A    ; at 41CA be RLD to access
top half of map
                                        ;
    40AC ED5B0C40      LD DE,(DFIL)   ;DE = address of display file
    40B0 3A8A40        LD A,(DIRN)    ;A = direction to scroll
    40B3 CB4F                BIT 1,A            ;scroll up/down or
right/left?
    40B5 282C             JR Z LTRT       ;if bit(1)=0, go scroll
right/left
    40B7 CB47          BIT 0,A         ;scroll up or down?
    40B9 3A8940         LD A,(YPOS)    ;A = y position of window on
map
    40BC 200F            JR NZ SCRD     ;if bit(1) of DIRN =1, go
scroll down
    40BE FE1A          CP 1A           ;is window at bottom of map?
    40C0 C8            RET Z           ;if so, return to BASIC
    40C1 3C            INC A           ;y = y + 1
    40C2 212100          LD HL,0021     ;HL = 33d = no. chars in
screen line
    40C5 19             ADD HL,DE      ;make HL address of 2nd line
on screen
    40C6 01F702        LD BC,02F7      ;BC = 759d = 33d * 23d
    40C9 EDB0          LDIR            ;scroll screen up 1 line
    40CB 185B          JR NEWY         ;go update YPOS
                                       ;
    40CD FE00     SCRD CP 00           ;check if window is at top of
map
    40CF C8            RET Z           ;if so, return to BASIC
    40D0 3D            DEC A           ;y = y - 1
    40D1 211703        LD HL,0317      ;HL = 791d = 24d * 33d - 1
    40D4 19             ADD HL,DE      ;make HL address of last char
on screen
    40D5 54            LD D,H
    40D6 5D            LD E,L          ;DE = HL
    40D7 012100         LD BC,0021     ;BC = 33d = no. of chars in
screen line
    40DA ED42           SBC HL,BC      ;HL -> last char on next to
last line
    40DC 01F702        LD BC,02F7      ;BC = 759d = 33d * 23d
    40DF EDB8          LDDR            ;scroll screen down 1 line
    40E1 1845          JR NEWY         ;go update YPOS
                                       ;
    40E3 0600     LTRT LD B,00
    40E5 CB47            BIT 0,A       ;check whether to scroll
right or left
    40E7 3A8840         LD A,(XPOS)    ;get x position of window on
map
    40EA 201C            JR NZ SCRR     ;if bit(0) of DIRN = 1, go
scroll right
```

WORLD MAP Part 2

```
    40EC 3C             INC A          ;x = x + 1
    40ED FE55           CP 55            ;check if x>85d (past right
edge of map)
    40EF 2001               JR NZ NWXL      ;if not, skip next
instruction
    40F1 AF               XOR A           ;zero x (make it the left
edge)
    40F2 328840    NWXL LD (XPOS),A    ;store new x
    40F5 13             INC DE          ;make DE address of top left
char on screen
    40F6 62             LD H,D
    40F7 6B             LD L,E         ;HL = DE
    40F8 23             INC HL         ;make HL address of next char
    40F9 3E18           LD A,18        ;make A count 24d times
    40FB 0E1F      NXLL LD C,1F          ;BC = 31d = no. of chars in
line - 1
    40FD EDB0            LDIR           ;scroll this line 1 char to
the left
    40FF 23               INC HL          ;move HL past end of line
char and
    4100 23             INC HL         ;past 1st char in next line
    4101 13               INC DE          ;move DE past last char in
line and
    4102 13             INC DE         ;past end of line char
    4103 3D             DEC A          ;countdown A
    4104 20F5           JR NZ NXLL     ;go do next line if A <> 0
    4106 1823             JR FFIL         ;go find address of screen
area to fill
                                       ;
    4108 3D        SCRR DEC A          ;x = x - 1
    4109 FEFF            CP FF            ;check if x went past left
edge of map
    410B 2002               JR NZ NWXR      ;if not, skip next
instruction
    410D 3E54           LD A,54        ;make x the right edge
    410F 328840    NWXR LD (XPOS),A    ;store new x
    4112 211703         LD HL,0317     ;HL = 791d = 24d * 33d - 1
    4115 19               ADD HL,DE       ;make HL address of bottom
right char
    4116 54             LD D,H
    4117 5D             LD E,L         ;DE = HL
    4118 2B               DEC HL           ;HL = address of previous
char
    4119 3E18           LD A,18        ;count for 24d lines
    411B 0E1F      NXLR LD C,1F          ;BC = 31d = no. of chars in
line - 1
    411D EDB8            LDDR           ;scroll this line 1 char to
right
    411F 2B             DEC HL
    4120 2B             DEC HL
    4121 1B             DEC DE
    4122 1B             DEC DE
    4123 3D             DEC A          ;countdown A
    4124 20F5           JR NZ NXLR     ;if A<>0, go do another line
    4126 1803           JR FFIL        ;skip next instruction
    4128 328940    NEWY LD (YPOS),A    ;store new y
                                       ;
    412B ED4B8840 FFIL LD BC,(XPOS)   ;B = y; C = x
    412F 04               INC B            ;B must be > 0 so DJNZ at
414C works
    4130 3A8A40         LD A,(DIRN)    ;get direction
```

```
    4133 FE22          CP 22          ;check if down
    4135 2004           JR NZ MPST     ;if not, go get start of map
data
    4137 3E17          LD A,17        ;A = 17h = 23d
    4139 80             ADD A,B       ;add 23d to y since we need
to fill
    413A 47            LD B,A         ; the last screen line
    413B 21DB42   MPST LD HL,42DB     ;HL = start of map data - 55h
= 17200d - 85d
    413E 115500         LD DE,0055    ;DE = 85d = no. chars in map
line
    4141 78            LD A,B
    4142 D61A           SUB 1A        ;check if y is in upper half
of map
    4144 3805               JR C NXLM       ;if so, skip next 3
instructions
    4146 3C            INC A          ;adjust y
    4147 47            LD B,A
    4148 CD8B40         CALL ACCB     ;gain access to bottom half
of screen
    414B 19       NXLM ADD HL,DE      ;add 85d
    414C 10FD           DJNZ NXLM     ;loop until HL is address of
line y on map
    414E E5            PUSH HL        ;store it
    414F 09            ADD HL,BC      ;add C (get to xth char)
    4150 3A8A40        LD A,(DIRN)    ;get direction
    4153 CB4F             BIT 1,A          ;check whether scroll is
vertical or horiz
    4155 282F          JR Z VFIL      ;go fill a column if vertical
                                      ;
    4157 E5        HFIL PUSH HL       ;store address of start char
on map
    4158 2A0C40        LD HL,(DFIL)   ;get start of display file
    415B 23            INC HL         ;move past first 118d char
    415C CB47           BIT 0,A       ;check whether scroll was up
or down
    415E 2004               JR NZ STSC      ;if down, skip next 2
instructions
    4160 11F702        LD DE,02F7     ;DE = 23d * 33d
    4163 19            ADD HL,DE      ;get to last line on screen
    4164 EB       STSC EX DE,HL       ;store screen fill address in
DE
    4165 E1             POP HL        ;get address of start char on
map
    4166 0620          LD B,20        ;loop counter for 32d chars
    4168 3E56     NXTX LD A,56        ;the right edge
    416A 0C            INC C          ;C keeps track of x position
    416B B9            CP C           ;check if x went off the edge
    416C 2002               JR NZ CLAX      ;if not, skip next 2
instructions
    416E E1             POP HL        ;get address of start of map
line
    416F E5            PUSH HL        ;restore stack size
    4170 AF       CLAX XOR A          ;A = 0
    4171 ED67           RRD           ;or RLD if accessing bottom
half of map
                                      ; this moves 4 bits from map
data into A
    4173 EB            EX DE,HL       ;get screen address into HL
    4174 77            LD (HL),A      ;copy map char to screen
    4175 CB5F          BIT 3,A        ;should it be inverse?
```

```
    4177 2804              JR Z RSMX       ;if not, skip next 2
instructions
    4179 CBFE           SET 7,(HL)     ;make it an inverse char
    417B CB9E           RES 3,(HL)
    417D EB       RSMX EX DE,HL        ;get map address back into HL
    417E ED6F           RLD               ;or RRD if accessing bottom
half of map
                                       ; this restores map data
    4180 23             INC HL         ;move to next map char
    4181 13             INC DE         ;move to next screen position
    4182 10E4           DJNZ NXTX      ;loop for each char
    4184 C1             POP BC         ;restore stack
    4185 C9             RET            ;back to BASIC
                                       ;
    4186 C1       VFIL POP BC          ;restore stack
    4187 ED4B8840       LD BC,(XPOS)   ;B = y, C = x
    418B CB47             BIT 0,A          ;check whether scroll was
right or left
    418D 200D             JR NZ SVMY       ;if right, go find screen
fill address
    418F 3E1F           LD A,1F        ;A = 31d
    4191 5F             LD E,A         ;DE = 31d
    4192 19               ADD HL,DE        ;add to map address since
destination
                                       ; is rightmost screen column
    4193 81             ADD A,C        ;A = x + 31d
    4194 1E55           LD E,55        ;DE = 85d
    4196 BB               CP E             ;did map address go past
right edge?
    4197 3803              JR C SVMY       ;if not, skip next 2
instructions
    4199 ED52           SBC HL,DE      ;move back 85d chars
    419B 37              SCF             ;carry flag remembers a left
scroll
    419C E5       SVMY PUSH HL         ;store map address
    419D 2A0C40         LD HL,(DFIL)   ;get start of display file
    41A0 23             INC HL         ;move past 1st 118d char
    41A1 3003            JR NC SVSY     ;if scroll was right, skip 2
instructions
    41A3 1E1F           LD E,1F        ;DE = 31d
    41A5 19               ADD HL,DE        ;move to last char on 1st
line on screen
    41A6 EB       SVSY EX DE,HL        ;save screen address in DE
    41A7 E1             POP HL         ;get map address
    41A8 48             LD C,B         ;C = y
    41A9 0618            LD B,18         ;loop counter for 24d chars
down screen
    41AB 180F           JR CLAY        ;skip 1st part of loop
    41AD 3E19     NXTY LD A,19         ;A = 25d
    41AF B9              CP C            ;check if y has crossed to
bottom half of map
    41B0 200A             JR NZ CLAY      ;if not, skip the next 5
instructions
    41B2 CD8B40          CALL ACCB      ;gain access to bottom half
of map
    41B5 D5             PUSH DE        ;save screen address
    41B6 114D08          LD DE,084D     ;DE = 2125d = 25d * 85d =
size of map data
    41B9 ED52           SBC HL,DE      ;adjust map address
    41BB D1             POP DE         ;restore screen address
    41BC AF       CLAY XOR A           ;A = 0
```

```
    41BD ED67           RRD             ;or RLD if accessing bottom
half of map
                                        ; this moves 4 bits from map
data into A
    41BF EB             EX DE,HL        ;get screen address into HL
    41C0 77             LD (HL),A       ;copy map char to screen
    41C1 CB5F               BIT 3,A             ;check if it should be
inverse
    41C3 2804                JR Z RSMY          ;if not, skip next 2
instructions
    41C5 CBFE           SET 7,(HL)      ;make it inverse
    41C7 CB9E           RES 3,(HL)
    41C9 EB       RSMY  EX DE,HL        ;get map address back into HL
    41CA ED6F            RLD              ;or RRD if accessing bottom
half of map
                                        ; this restores map data
    41CC D5             PUSH DE         ;store screen address
    41CD 115500         LD DE,0055      ;DE = 85d
    41D0 19             ADD HL,DE       ;move down 1 map line
    41D1 E3             EX (SP),HL      ;get screen address into HL
    41D2 1E21           LD E,21         ;DE = 33d
    41D4 19             ADD HL,DE       ;move down 1 screen line
    41D5 D1             POP DE          ;restore map address
    41D6 EB             EX DE,HL        ;get map address back into HL
    41D7 0C             INC C           ;keep track of y position
    41D8 10D3           DJNZ NXTY       ;loop for next char
    41DA C9             RET             ;back to BASIC
                                        ;
    41DB 320D        STRM DEFB 32 0D        ;M$, name of M string
variable
    ; routine to fill the BASIC string M$ with 63d chars from the
screen:
    ;  the last 31 chars from the 2nd line and all 32 from the 3rd
line
    41DD 2A1640   FILM LD HL,(CHAD)   ;get current value of CHAD
    41E0 E5             PUSH HL         ;store it
    41E1 21DB41            LD HL,STRM       ;get address of name of
string
    41E4 221640         LD (CHAD),HL    ;put it in CHAD
    41E7 CD1C11            CALL 111C         ;ROM routine to find a
variable
    41EA 23             INC HL
    41EB 23             INC HL
    41EC 23             INC HL          ;get to first char of string
    41ED E3                 EX (SP),HL       ;HL = old CHAD; put M$
address on stack
    41EE 221640         LD (CHAD),HL    ;restore old value of CHAD
    41F1 2A0C40         LD HL,(DFIL)    ;get start of display file
    41F4 112300         LD DE,0023      ;DE = 35d
    41F7 19             ADD HL,DE       ;get to 2nd char of 2nd line
    41F8 D1             POP DE          ;get start of M$
    41F9 011F00         LD BC,001F      ;BC = 31d
    41FC EDB0           LDIR            ;copy 31d chars
    41FE 23             INC HL          ;skip over 118d char
    41FF 0E20           LD C,20         ;BC = 32d
    4201 EDB0           LDIR            ;copy 32d chars
    4203 C9             RET             ;back to BASIC
    4204 2A1040         LD HL,(VARS)    ;find G$ in BASIC variables
    4207 3E80     LKAG  LD A,80         ;end of VARS marker
    4209 BE             CP (HL)
    420A C8             RET Z           ;couldn't find G$
```

```
420B 3E4C          LD A,4C       ;G$ marker
420D BE            CP (HL)
420E 2824          JR Z FDIT     ;JR if G$ found
4210 CB6E          BIT 5,(HL)
4212 2818          JR Z ARST     ;if array or string
4214 CB7E          BIT 7,(HL)
4216 280E          JR Z 1LTV     ;if 1 letter var name
4218 CB76          BIT 6,(HL)
421A 2805          JR Z M1LT     ;if more than 1 letter name
421C 111200          LD DE,0012    ;18 bytes to skip FOR-NEXT
var
421F 1808          JR SKPV
4221 23        M1LT INC HL
4222 CB7E          BIT 7,(HL)
4224 28FB          JR Z M1LT     ;loop to find end of var name
4226 110600    1LTV LD DE,0006   ;6 bytes to skip number
4229 19        SKPV ADD HL,DE    ;add to find next var
422A 18DB          JR LKAG       ;look again
422C 23          ARST INC HL         ;get length of array or
string
422D 5E            LD E,(HL)
422E 23            INC HL
422F 56            LD D,(HL)
4230 23            INC HL
4231 19            ADD HL,DE     ;add and get to next var
4232 18D3          JR LKAG       ;look again
4234 23        FDIT INC HL       ;get length of G$
4235 5E            LD E,(HL)
4236 23            INC HL
4237 56            LD D,(HL)
4238 E5            PUSH HL       ;push start of G$
4239 19            ADD HL,DE     ;find end of G$
423A 227B40        LD (GLST),HL  ;store end
423D E1            POP HL        ;pop start
423E 23            INC HL
423F 228240        LD (GCUR),HL  ;store start of G$
4242 210000        LD HL,0000
4245 228440        LD (TIMR),HL  ;zero timer
4248 3E00          LD A,00
424A 328640        LD (TMR3),A   ;zero 3rd timer byte
424D 218740        LD HL,ELEN
4250 3600          LD (HL),00    ;initialize entry length
4252 212140    SETF LD HL,FLAG
4255 CBC6          SET 0,(HL)    ;assume a key  is depressed
4257 2A8440    KYLP LD HL,(TIMR)
425A 110100        LD DE,0001
425D 19            ADD HL,DE     ;inc. timer
425E 228440        LD (TIMR),HL
4261 3007          JR NC IGN3    ;no carry?, ignore TMR3
4263 3A8640        LD A,(TMR3)   ;if carry then
4266 3C            INC A         ;inc TMR3
4267 328640        LD (TMR3),A
426A 2A0E40    IGN3 LD HL,(DFCC) ;get print position
426D 3A8440        LD A,(TIMR)   ;get low byte of timer
4270 CB77          BIT 6,A       ;flash cursor by bit 6
4272 2004          JR NZ CRON
4274 3600          LD  (HL),00   ;cursor off
4276 1802          JR SCON       ;skip cursor on
4278 3680      CRON LD (HL),80   ;cursor on
427A CDBB02    SCON CALL 02BB    ;ROM key scan routine
427D 44            LD B,H
```

```
427E 4D          LD C,L
427F 5D          LD E,L          ;L=FF if no key pressed
4280 212140      LD HL,FLAG
4283 CB46        BIT 0,(HL)      ;has flag been reset?
4285 2807        JR Z RDKY       ;if so, read key
4287 1C          INC E           ;is a key pressed?
4288 20CD        JR NZ KYLP      ;JR if old key still pressed
428A CB86        RES 0,(HL)      ;ready for new key
428C 18C9        JR KYLP         ;loop for proper key
428E 1C      RDKY INC E          ;is a key pressed?
428F 28C6        JR Z KYLP       ;if not, loop for key
4291 CDBD07      CALL 07BD       ;decode BC to find address of
4294 7E          LD A,(HL)       ;character code of key
4295 FE77        CP 77           ;delete key?
4297 2023        JR NZ ENTR      ;no? check for enter key
4299 218740      LD HL,ELEN      ;get address of entry length
429C 3E00        LD A,00
429E BE          CP (HL)         ;length=0?
429F 28B1        JR Z SETF       ;then don't delete
42A1 35          DEC (HL)        ;dec entry length
42A2 2A0E40      LD HL,(DFCC)    ;get print position
42A5 3600        LD (HL),00      ;erase character
42A7 2B          DEC HL          ;move position back
42A8 3E76        LD A,76
42AA BE          CP (HL)         ;end of line in display file?
42AB 2001        JR NZ NEOL
42AD 2B          DEC HL          ;move past EOL character
42AE 220E40  NEOL LD (DFCC),HL   ;store new print position
42B1 2A8240      LD HL,(GCUR)    ;get current position in G$
42B4 2B          DEC HL
42B5 3600        LD (HL),00      ;erase deleted character
42B7 228240      LD (GCUR),HL    ;store new position in G$
42BA 1896    GOBK JR SETF        ;loop for next key
42BC FE76    ENTR CP 76          ;check for enter key
42BE 2007           JR NZ VLDK      ;if not, JR to valid key
check
42C0 2A0E40      LD HL,(DFCC)    ;get print position
42C3 3600        LD (HL),00      ;erase cursor
42C5 1857        JR EXIT
42C7 5F      VLDK LD E,A         ;store character in E
42C8 2A2940        LD HL,(NXLN)   ;get address of next BASIC
line
42CB 23          INC HL
42CC 23          INC HL
42CD 46          LD B,(HL)       ;B=length of line
42CE 05          DEC B
42CF 23          INC HL
42D0 23          INC HL
42D1 23      NXCH INC HL
42D2 1002             DJNZ CHK=       ;nothing left in REM
statement?
42D4 18E4        JR GOBK         ;then go back
42D6 56      CHK= LD D,(HL)      ;get char in REM
42D7 7B          LD A,E          ;get keyed char
42D8 BA          CP D            ;are they equal?
42D9 281D        JR Z ADDG       ;yes? then add to G$
42DB 23      LK-, INC HL
42DC 1002        DJNZ CK-,       ;nothing left in REM?
42DE 18DA        JR GOBK         ;no? then go back
42E0 3E16    CK-, LD A,16        ;check for - character
42E2 BE          CP (HL)
```

```
42E3 2806          JR Z ENDC      ;go get end character
42E5 3E1A          LD A,1A        ;check for , character
42E7 BE            CP (HL)
42E8 C0            RET NZ         ;not - or , so REM invalid
42E9 18E6          JR NXCH        ;go get next char in REM
42EB 23       ENDC INC HL
42EC 1002          DJNZ LDCH      ;nothing left in REM?
42EE 18CA          JR GOBK        ;no? then go back
42F0 7E       LDCH LD A,(HL)      ;A=end character
42F1 BB            CP E           ;CP with keyed character
42F2 38E7          JR C LK-,      ;is keyed code too large?
42F4 7A            LD A,D         ;A=start character
42F5 BB            CP E           ;CP with keyed character
42F6 30E3          JR NC LK-,     ;is keyed code too small?
42F8 218740   ADDG LD HL,ELEN     ;get address of entry length
42FB 34            INC (HL)       ;add 1
42FC 2A0E40        LD HL,(DFCC)   ;get print position
42FF 73            LD (HL),E      ;print keyed character
4300 23            INC HL
4301 F5            PUSH AF
4302 3E76          LD A,76
4304 BE            CP (HL)        ;end of line in display file?
4305 2001          JR NZ NT76
4307 23            INC HL         ;move past EOL character
4308 220E40   NT76 LD (DFCC),HL
430B 2A8240        LD HL,(GCUR)   ;get current position in G$
430E F1            POP AF
430F 73            LD (HL),E      ;put keyed char in G$
4310 23            INC HL
4311 228240        LD (GCUR),HL   ;store new position
4314 2B            DEC HL
4315 ED4B7B40      LD BC,(GLST)   ;BC points to last byte in G$
4319 AF            XOR A          ;clear carry flag
431A ED42          SBC HL,BC      ;any more room in G$?
431C 209C          JR NZ GOBK     ;yes? then GOBK
431E 0600     EXIT LD B,00
4320 218740        LD HL,ELEN     ;get address of entry length
4323 4E            LD C,(HL)      ;put length in BC
4324 C9            RET
```



MECHANICAL AFFINITY

Serving ALL QL Users

Frank Davis
513 East Main St
Peru, Indiana 46970
317-473-8031 evenings & weekends

Paul Holmgren
5231 Hilton Wood Court
Indianapolis IN, 46254
317-291-6002 evenings & weekends

Trump Cards, disk interface for two 360K or 720K disk drives, 896K of memory and TK2, reconditioned for $100.

Gold Cards,3 drive disk interface, 2 meg memory,TK2 and 16 megsHerz speed. All this for only $230.

Super Gold Cards, built in 4 drive adaptor disk interface, 68020 processor, 4 meg of memory, runs at 24 megaHerz speed, built in true parallel printer port (with printer cable) and the latest TK2. The price is $495. Contact us about our Trump , or Gold Card trade in program.

Sinclair QL Printers, black with QL logo, 9 pin printers that are made to run with the QL (serial). Price is $60.

CUESHELL, the new graphics oriented desktop program for the QL. It gives you program options on the screen, which you can easily control, change or launch. It is pointer environment driven and makes full use of level 2 drivers. Our price is $62.

Page Designer 3, the QLs easiest to use full featured desk top publishing program. Comes with loads of fonts and clip art in a 4 disk set. You can even use HDF fonts from Digital Precisions Professional Publisher, as well as other clip art from other QL programs. This ad was done using PD3. Our price is $63.

Text87Plus4, an excellent word processor that allows you to use drivers for all the latest printers, gives you many fonts as well as columns and precise control of paper size and print size. Our price is $120.

LineDesign Version2, a top of the line superb vector drawing program package that allows you to mix many! fonts and graphics (10 disk set) with your artwork. For $153.

QL--IDE Hard Drive Interface for only $115.

DataDesign 3, a fast pointer driven database that is easy to set up and use for the QL. The price is $95.

Qliberator 3.36, compiles virtually all of SuperBasic, and it is easy to use, with QL and QXL. The price is $75.

CONTACT US FOR ALL YOUR QL NEEDS, WE'LL DO OUR BEST!

UPDATE! Magazine

The only known, privately produced Sinclair computer magazine that covers ALL of the Sinclair computers. We are now starting our 8th year of publication ! ! !

We are a large quarterly magazine that is produced on Sinclair computers. We cover the QL, Z88, TS2068, Spectrum and the ZX81. Minimum issue size is 50 pages, and does include ads from Sinclair dealers. The price is $18 in US$ in North America; $22 elsewhere, and we do accept the British Pound or the German Mark equivalents for a subscription.

Send all funds and requests for a new subscription to:

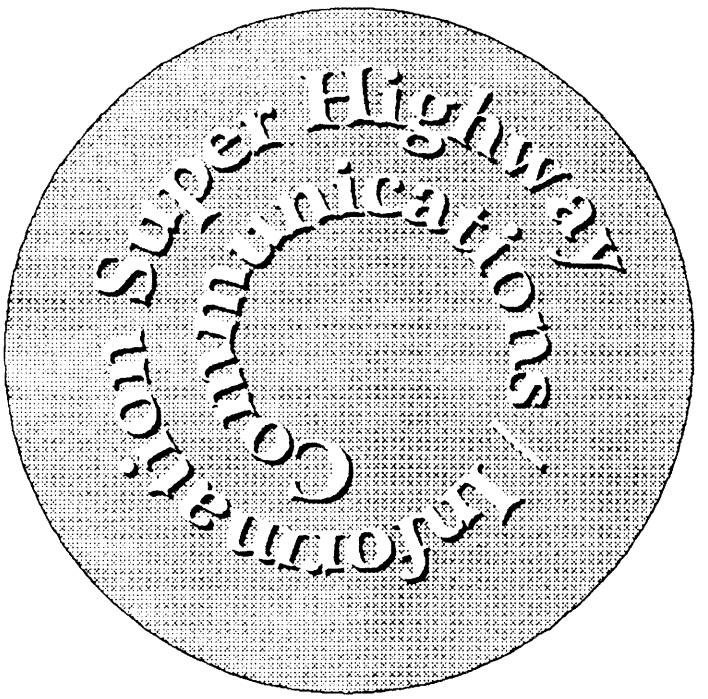
UPDATE! MAGAZINE
P.O. BOX 1095
PERU, IN 46970 USA

Checks, travelers checks, cash are all acceptable.

MODEMS or Magic

by Abed Kahale CATUG-T/SNUG

My previous article (Modems or Black Magic, UPDATE! Jan. 94) discussed the basic elements of communications by modem, but there is a lot more of this *Magic*. The field of the *Information/ Communications Super Highway* is very complex and new developments appear at fast pace, such as an announcement by a company that they are developing a direct wire communications system that obsoletes modems. With the prevailing use of fiber-optics lines, communications will be in the millions of data bits transfer rate such as the cable TV system is now.



Computers normally process 16 or 32 bit words of digital information on their internal data buses. Even the computer's parallel printer port offers a data path that is 8 bits wide. However there is only one transmitting wire (the other wire is for signal return) in a telephone line. Modems must break down (Modulate) each digital word into a sequence of audio signals, and send each signal in turn along the telephone line. An 8-bit data word is sent as eight 1-bit signals.

Most modems for computer communication are asynchronous or *un*synchronized. (synchronous modems are more expensive, they require complex circuitry.) Because two communicating modems are operating asynchronously, the receiving machine must know *when* a stream of data is being sent, *where* each stream of data starts and ends, and whether the stream is correct or not (see the previous article). To accomplish this feat, extra bits are added to the data to organize it into a standard sequence known as a dataframe having a start bit, a data bit, a parity bit and a stop bit.

Port Connections and Signals

The translation from parallel data words to a serial data stream takes place within the computer's serial port circuit in an IC known as a Universal Asynchronous Receiver/Transmitter (UART). Logic 1's and 0's are converted into bipolar signals. Logic 1's are translated to - 5 to - 15 volt levels (mark), and logic 0's are translated to + 5 to + 15 volt levels (space). Those bipolar signals are supplied to the TX (transmit) pin of a standard serial communications port. Those physical connections are also called RS-232C ports. (RS-232C is a standard of the Electronic Industry Association) In Europe, the RS-232C standard is called V.24.

An external modem connected to the serial port MOdulates the bipolar TX signal into an audio signal which is sent over the telephone line. Audio signals received by the modem and DEModulated into bipolar signals are returned to the serial port's RX (receive) pin. An internal modem is a plug-in unit installed inside a computer containing a serial port. For most serial port connections, only TX , RX and ground wires are needed to establish a working port.

Communications Software

All communications software must be able to operate in two distinct modes: the **command mode**, and the **terminal mode.** In the command mode, any commands issued control either the communications software, the host computer, or the modem connected to it. In the terminal mode, any commands issued are sent directly to the modem, or they are sent through the modem and to the remote computer modem.

Typical instructions that might be issued in the command mode include retrieving a telephone number from a data log, setting the proper communications protocol, dialing a number, and saving any data that is received to a disk.

In the terminal mode, all keyboard (or disk) input is sent directly to the modem. If the modem is *off-line* or not connected to a remote computer modem, the input is recognized as modem commands. if the modem is *on-line* and has an active connection to a remote computer modem, all inputs are sent through the modem to the remote computer unless certain special control characters precede the data.

The difference between the command mode (in which commands are acted on locally) and terminal mode (in which commands affect the remote computer) is probably the most confusing aspect of modem communications for beginners.

One common mistake made by newcomers, for example, is to initiate a file UPLOAD in terminal mode without issuing a corresponding command in the command mode. The first command is required to tell the computer to send a file. The second is necessary to instruct the host (remote) computer to receive it and store it on disk.

Flow Control

To control the flow of data between two modems, software codes (such as XON and XOFF) are passed between them. Communications software interprets such codes and controls modem operation accordingly.

However, the flow of data between a computer and modem is not always controlled by software codes. Instead, additional signal lines in an RS-232 port allow for hardware flow control (or handshaking). The Request To Send (RTS) line tells a modem to prepare to receive data from the computer. Once the modem is ready to send, it will return a Clear To Send (CTS) signal to the computer. RTS and CTS signals act together to handle data transfer

When the computer is ready for operation (but not necessarily ready to send data), it asserts Its Data Terminal Ready (DTR) signal. DTR must remain active throughout the entire connection time. The modem sends a Data Set Ready (DSR) signal to the computer after the modem has been activated, and has finished self-tests or preparation for connection. The DTR and DSR signals establish the connection between modem and computer, but they do not control the flow of data between the two.

If a telephone ring signal is detected at the modem, a Ring Indicator (RI) signal is sent to the computer. When the modem subsequently picks up the ringing line and detects the presence of carrier, a Data Carrier Detect (DCD) signal is passed to the computer. Ring and carrier detect signals invoke the communication software to receive communication from a distant modem.

Modulating the Signal

A modem transmits data by generating a carrier that is then modulated. Several different methods of signal modulation have been developed through the years to improve the efficiency of data transfers. Of course, two modems must be capable of the same modulation design for successful communications.

Two modems that are communicating with each other must generate different carrier frequencies. Each communication standard defines the transmitting and receiving carrier frequency. For example, for 300 BAUD communication, one modem must have a carrier center frequency of 1170 Hz, and the second must have a center frequency of 2125 Hz. By convention, the modem initiating the call, or the originate-mode modem, has the lower carrier frequency. The modem receiving the call, or the answer-mode modem, has the higher carrier frequency.

In the early days of modem communication, each transition of the audio carrier signal represented a single bit. Each transition is known as a BAUD, so the BAUD rate equaled the transmission rate in bits-per-second or BPS. Unlike those early modems (like the TS-2050), modern modem designs allow two, three, four, or more bits to be encoded into every audio signal transition (or BAUD). This means that modem throughput now equals two, three, or four times the BAUD rate being carried on the telephone line.

For example, a modem operating at 2400 BAUD (2400 audio signal transitions per second) can carry 4800 BPS if two bits are encoded onto every BAUD. The same 2400 BAUD modem can carry 9600 BPS if 4 bits are encoded onto every BAUD. Today, the modem's BAUD rate rarely matches the modem's bit rate in BPS. If the modem were operating at 4800 BAUD with 3-bit encoding, it would be transmitting 14,400 BPS (14.4 KBPS) V.32bis, and so on.

The concept of encoding is different from data compression. Encoding transfers all original data bits from one system to another. While data compression replaces repeating sequences of bits with much shorter bit sequences known as symbols or tokens. Encoding designs and data compression are described in more detail later in this article.

Modulation Designs

All waveforms have three basic characteristics — amplitude, frequency, and phase. Each of those characteristics can be adjusted to represent a bit.

Frequency-Shift Keying(FSK) is similar To Frequency Modulation (FM) where only the frequency of a carrier is changed, and it is one of the oldest modulation design still in use. FSK sends a logic 1 at one particular frequency (usually 1750 Hz), and a logic 0 is sent at another discrete frequency (often 1080 Hz). Frequencies are typically sent at 300 BAUD, and each BAUD can carry one bit, so FSK can send 300 BPS. This early technique resulted in the classical "BAUD equals BPS" which still exists today. Bell 103 was the widely accepted modem standard — simple FSK modulation at 300 BAUD. This is the only standard in which the data rate matches the BAUD rate.

Phase-Shift Keying (PSK) is a close cousin of FSK, but the phase timing of a carrier wave is altered while the carrier's frequency stays the same. A logic 1 or logic 0 is represented by the alteration of the carrier's

phase. Because phase can be shifted in several precise increments such as 0°, 90°, 180° and 270°. PSK can encode one, two, three, or more bits bit per BAUD. A 1200-BAUD modem using PSK can transmit 2400 BPS over an 1800 Hz carrier. PSK in conjunction with FSK can encode even more bits per BAUD.

In Quadrature-Amplitude Modulation (QAM), both the phase and amplitude of the wave are modulated to encode up to six bits onto every BAUD, although only four bits are usually reserved for data. Most QAM modems have a 1700-Hz or 1800-Hz carrier and a base rate of 2400 BAUD, so they can carry up to 9600 BPS.

Trellis-Coded Quadrature-Amplitude Modulation (TCQAM or TCM) also generates an 1800-Hz carrier at a 2400-BAUD base rate, but it uses the full 6-bit encoding capability of QAM to provide a rate of 14400 BPS. TCM is now the most popular modulation design for high-performance modems because data can be checked on-the-fly with high reliability.

As the speed increases, line noise becomes very critical to communications and *error correction* becomes vital.

MNP Standards

The Microcom Networking Protocol (MNP) is a complete hierarchy of standards developed during the mid 1980s. They were designed to work with other modem technologies to provide *error correction* and *data compression*. Originally developed by Microcom Inc., the protocol is now in the public domain.

MNP provides error control and data compression when one modem is communicating with another modem that supports MNP. MNP class 4 is specified by the ITU (International Telecommunications Union) V.42 as a backup error control design for LAPM (Link Access Procedure for Modems) in the event that V.24 cannot be invoked. Out of ten MNP levels, most modern modems support MNP2 to MNP5. Each MNP class has all the features of the previous class plus its own.

MNP *class 2: (stream mode):* Data is sent in both directions at the same time. That results in a speed about 84% as fast as data transmissions with no error correction.

MNP *class 3:* The sending modem strips the start and stop bits from a data block before sending it. The receiving modem then adds start and stop bits before passing the data to the receiving computer. It is about 8% faster than data transmissions with no error correction.

MNP *class 4*: A protocol (with some data compression) that checks telephone connection quality and uses adaptive packet assembly. On a noise free line, the modem sends larger blocks of data such as 1024 bytes. If the line is noisy, the modem sends smaller blocks of data such as 128 bytes so that less data must be resent in the event of an error.

MNP *class 5:* Provides data compression by detecting redundant data and converting it to fewer bits or tokens, thus increasing effective data throughput. A receiving modem decompresses the data before passing it to the receiving computer. MNP5 can speed data transmissions by as much as a factor of two compared with protocols having no data compression or error correction design.

MNP *class 6: Universal link negotiation* allows modems to obtain maximum performance from a line. The modems start at low speeds, then move to higher speeds until the best speed that both modems can work at is reached.

The Vees — The Most Common ITU Standards

V.17—Describes an application-specific modulation design for Group 3 fax which provides two-wires half-duplex, trellis-coded transmission at 7200, 9600, 12000, and 14400 BPS. Despite its low number, this is a recently introduced standard.

V.21—Provides the specifications for 300 BPS FSK serial modems (based on BELL103).

V.22-Provides the specifications for 1200 BPS (600 baud) PSK modems (based upon BELL212A)

V.32—Defines the first of the truly modern modems as a 9600/4800-bps; QAM, full-duplex modem operating at 2400 baud. This standard also incorporates trellis coding and echo cancellation to produce a stable, reliable, high-speed modem.

V.32bis—A fairly new standard extending the V.32 specification to define a 4800/7200/9600/12000/14400 BPS TCQAM full-duplex modem operating at 2400 baud. Trellis coding, automatic transfer rate negotiation, and echo cancellation make this type of modem one of the most popular and least expensive for general communication.

V.32fast—The temporary name of a standard that the ITU has not yet completed. When finished, it will probably replace V.32bis with speeds up to 28,800 BPS. It is anticipated that this will be the last analog protocol, eventually giving way to all-digital protocols as local telephone systems become entirely digital.

V.34—The ratified version of V.32fast. It provides for data speeds up to 128 KBPS with transmission rates as high as 3429 BAUD.

V.42—Defines a two-stage process of detection and negotiation for LAPM error control.

V.42bis—Extends the V.42 standard to include data compression.

Reference: Stephen J. Bigelow, Electronics Now

RMG ENTERPRISES

Supports Sinclair/Timex Users!



☎ **Call or FAX for information on prices and availability, hardware or software**

You can send a legal ▭ **and 50¢.**
Request a price sheet or send $4 for a GIANT IMPORT GIFT catalog; not computer related

NEW

The Companion Disk for LarKen
$9.95 PP.
We also carry extensive PC shareware.

Send check or money order to:
RMG Enterprises
14784 S QUAIL GROVE CIR
OREGON CITY OR 97045
or call and use your **VISA or MasterCard**
503 655-7484 (8AM-6PM Tue-Sat) FAX 503 655-4116

IT'S READY!

C C A T S & RMG Enterprises
Are Pleased To Announce

The Best Of The Plotter

ZX-81 TS-1000 TS-2068 QL

Contents are exclusively those developed and published by THE PLOTTER authors over the past 12 years.
Price: **$14.95** pp. per copy.

To Order:

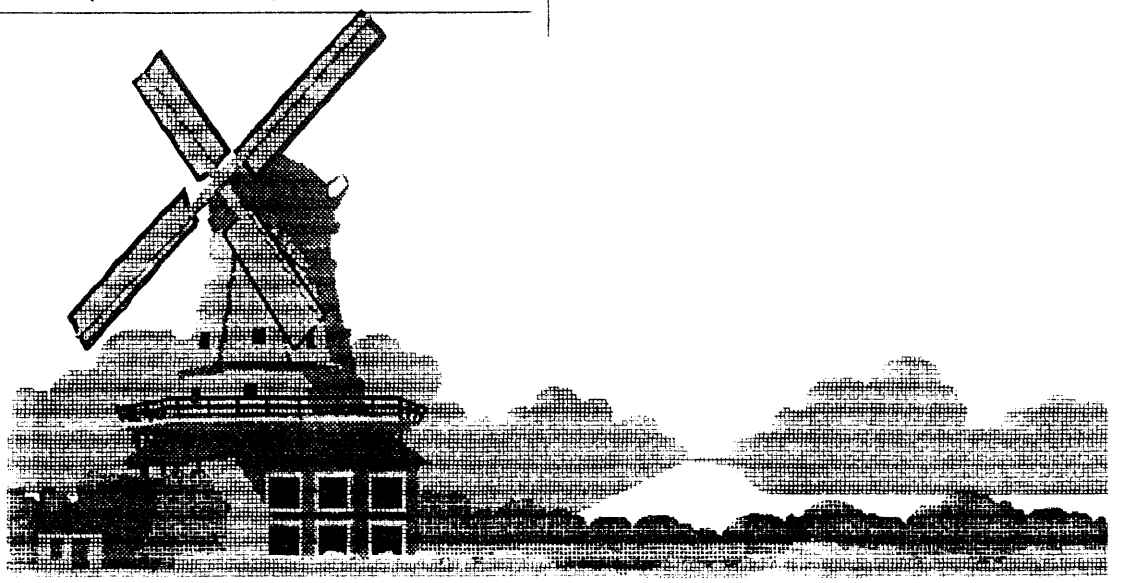
Send your check or money Order to:
RMG Enterprises
14784 S QUAIL GROVE CIR
OREGON CITY OR 97045
Call and use your VISA or MasterCard
503 655-7484 (8AM-6PM Tue-Sat) FAX 503 655-4116

Over 100 pages 8.5X11 Comb Bound

Name .....

Address .....

City/State/ZIP .....



```
  2 REM ORG OGN
 10 REM ;**********************
 12 REM ;* SERIAL PRINTER OUTPT
 14 REM ;*     PROGRAM
 16 REM ;* BOB BERCH     3/83
 18 REM ;**********************
 20 REM JR 50 ; ONE CHAR - ZX
 22 REM JR 360 ; PRINT Z$
 24 REM JR 300 ; COPY
 26 REM JR 200 ; LISTING
 28 REM JR 210 ; CONT LIST
 29 REM ;
 30 REM ;TYPE A CHAR. - ASCII**
 32 REM LD A,(AD1)
 34 REM JR 100
 49 REM ;
 50 REM ;TYPE A CHARACTER -ZX**
 52 REM LD A,(AD1)
 54 REM CALL /500
 60 REM PUSH AF
 62 REM CALL /100
 64 REM POP AF
 66 REM CP 13 ;TEST FOR C.R.
 68 REM RET NZ
 70 REM LD HL,4096; PAUSE FOR
 72 REM DEC HL; C.R.
 74 REM LD A,H
 76 REM OR L
 78 REM JR NZ,72
 80 REM RET
 89 REM ;
 90 REM LD A,13;ONE C.R.
 92 REM JR 60
 99 REM ;
100 REM ;MAIN OUTPUT ROUTINE
110 REM LD C,A
112 REM CALL H02E7 ;SET - FAST
114 REM LD B,11;BIT COUNTER
116 REM JR 130
120 REM RRC C; TEST
122 REM JR C,132
130 REM OUT HFF,A; SPACE BIT
132 REM LD DE,H0190 ;TIME CONST
140 REM DEC DE ; FOR BAUD RATE
142 REM LD A,D
144 REM OR E
146 REM JR NZ,140
148 REM SET 7,C; SET STOP BITS
150 REM IN A,HFE; END BIT
152 REM DJNZ 120
154 REM RET
199 REM ;
200 REM ;TYPE A LISTING********
202 REM CALL /90;ONE C.R.
206 REM LD HL,16509
208 REM LD (AD2),HL
210 REM CALL H0A2A ; CLS
212 REM LD HL,(AD2)
220 REM LD BC,/90;SET RET ADDR
222 REM PUSH BC
224 REM CALL H0745 ;OUT-LINE
225 REM POP BC
226 REM LD(AD2),HL
230 REM LD HL,(DFCC) ;FIND END
232 REM DEC HL
234 REM SUB A
235 REM DEC HL
236 REM CP #; "ENGLISH POUND"
238 REM JR Z,235
240 REM INC HL
242 REM LD DE,(DFIL); COMPUTE
244 REM AND A;  LENGTH
246 REM SBC HL,DE
247 REM PUSH HL
248 REM POP BC
250 REM CALL /400 ;TYPE
252 REM JR 210
299 REM ;
300 REM ; COPY ***************
310 REM LD DE,(DFIL)
312 REM LD BC,726
320 REM JR 400
359 REM ;
360 REM ; PRINT Z$*************
362 REM LD HL,/700;POINT TO
364 REM LD (CHAD),HL;DATA
370 REM CALL H111C;"GET-VARS"
372 REM RET C;EXIT IF NO Z$
380 REM INC HL;PICKUP LENGTH
382 REM LD C,#;IN BC
384 REM INC HL
386 REM LD B,#
389 REM INC HL;POINT TO
390 REM PUSH HL;STRING START
392 REM POP DE; IN DE
399 REM ;
400 REM ;TYPE A STRING*********
402 REM ;DE=START   BC=LENGTH
410 REM LD A,B
412 REM OR C
414 REM RET Z
420 REM PUSH BC
422 REM PUSH DE
424 REM LD A,(DE)
426 REM CALL /54 ;TYPE CHAR
432 REM POP DE
434 REM POP BC
435 REM DEC BC
438 REM INC DE
440 REM CALL H0F46;TEST BREAK
442 REM JR C,400 ;KEY
444 REM RST 8;ERR REPORT
446 REM 12; "D"
499 REM ;
500 REM ;CONV ZX TO ASCII******
502 REM ;IN + OUT THRU A REG.
510 REM CP 128
512 REM JR NC,580
515 REM CP 118
517 REM JR Z,590
520 REM SUB 64
522 REM RET NC;CONTROL CHARS
524 REM ADD 64
530 REM CP 28
532 REM JR C,540
534 REM CP 38
535 REM JR C,538
536 REM ADD A,7 ;LETS
538 REM ADD A,20 ;NUMS+LETS
539 REM RET
540 REM AND A
```

```
  542 REM JR NZ,550
  543 REM LD A,32 ;SPACE
  545 REM RET
  550 REM CP 11
  552 REM JR NC,560
  553 REM PUSH AF ;GRAPHICS
  554 REM LD A,35
  556 REM CALL /110
  557 REM POP AF
  558 REM ADD A,48
  559 REM RET
  560 REM LD HL,/600 ;SPECIAL
  561 REM SUB 11    ;CHARS
  562 REM ADD A,L
  563 REM LD L,A
  564 REM JR NC,567
  565 REM INC H
  567 REM LD L,#
  568 REM LD A,L
  569 REM RET
  580 REM SUB 128;INVERSE
  582 REM CALL /500
  584 REM ADD 32
  586 REM RET
  590 REM LD A,13 ;CARG. RET.
  592 REM RET
  599 REM ;
  600 REM ;LOOKUP TABLE FOR CODES 11 TO 27
  610 REM DATA 34,38,36,58,63,40,41,62,60,61,43,45,42,47,59,44,46
  699 REM ;
  700 REM DATA 63,13; "Z$"
  990 REM END
 1000 LET OGN=32512
 1002 LET DFCC=16398
 1004 LET CHAD=16406
 1006 LET AD1=16417
 1008 LET AD2=16507
 1010 LET DFIL=16396
```


QZX

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

HOW TO BUILD A LARKEN DOCK BOARD

Les Cottrell 108 River Heights Drive Cocoa, FL 32922-6630

To answer a request for schematics of the LarKen system I carefully examined the boards and sketched up what I saw. Peeking under the chips turned out to be more difficult than expected so to validate the dock board schematic two working samples were made. The first was built on a discarded Zebra spectrum emulator board. (The chip had been mounted on the original LarKen dock board.) The Larken dock board can be used with Aerco, Oliger or Ramex to add a second DOS.

To use one of the Zebra cartridge boards you must remove all chips, sockets and traces except the fingers that go into the 2068. Use an X-Acto knife to remove the traces. This is time consuming but the board will be a good fit in the cartridge slot. The locations shown in Fig.2 is the layout of the original LarKen. Use a Dremel or small hand drill to make the holes required for chip sockets and other components.

The second one was built on a plug-in perf board with .1 spacing edge-connector, Radio Shack P/N 276-192. This part is not listed in the current catalog, but I have found 4 boards in 2 different stores this past week. The dimensions of the original Larken dock board are shown in Fig. 2. This is easier to work with since drilling is not required.(The board must be cut out, but the little cutting wheels on a Dremel make this a fun job.) One word of warning here - check to see that the card fits and the cartridge door will close by temporarily placing a chip and socket in the outermost location. (Lesson learned.) The dock area is filled with this circuit so stick close to the locations shown. The transistor should lie as flat as possible.

All chips should be socketed. The sockets can be held in place with a dab of glue. Use 30-gauge wire-wrap wire for the point to point wiring of the circuit. Follow the schematic to make your connections. Start by soldering the wire to the edge connector and then routing the wire to the first point, stripping away enough coating to make one loop around the leg, soldering that point and moving to the next point. Radio Shack's 276-1570 Wire-wrapping tool has a neat little stripper hidden in the handle that you works well. This step involves "worrying" the coating away by back and forth motions and should be done carefully. You can also make the loop without any stripping and melt the coating and make the solder joint at the same time. Perform a continuity check on a few joints to verify your technique if you choose this method.

Note that A8 and A9 don't go to the same pin numbers on the RAM and the EPROM. They have a check mark on the schematic as a flag. On the units I made I socketed the transistor, LED and the big diode using individual sockets call "springs" that a friend gave me. These were especially useful for the large diode whose markings I did not recognize. I found a diode that looked similar (a 1N4003) and tried it. I found one that worked on the first try on unit one, but I had to try several on unit two.

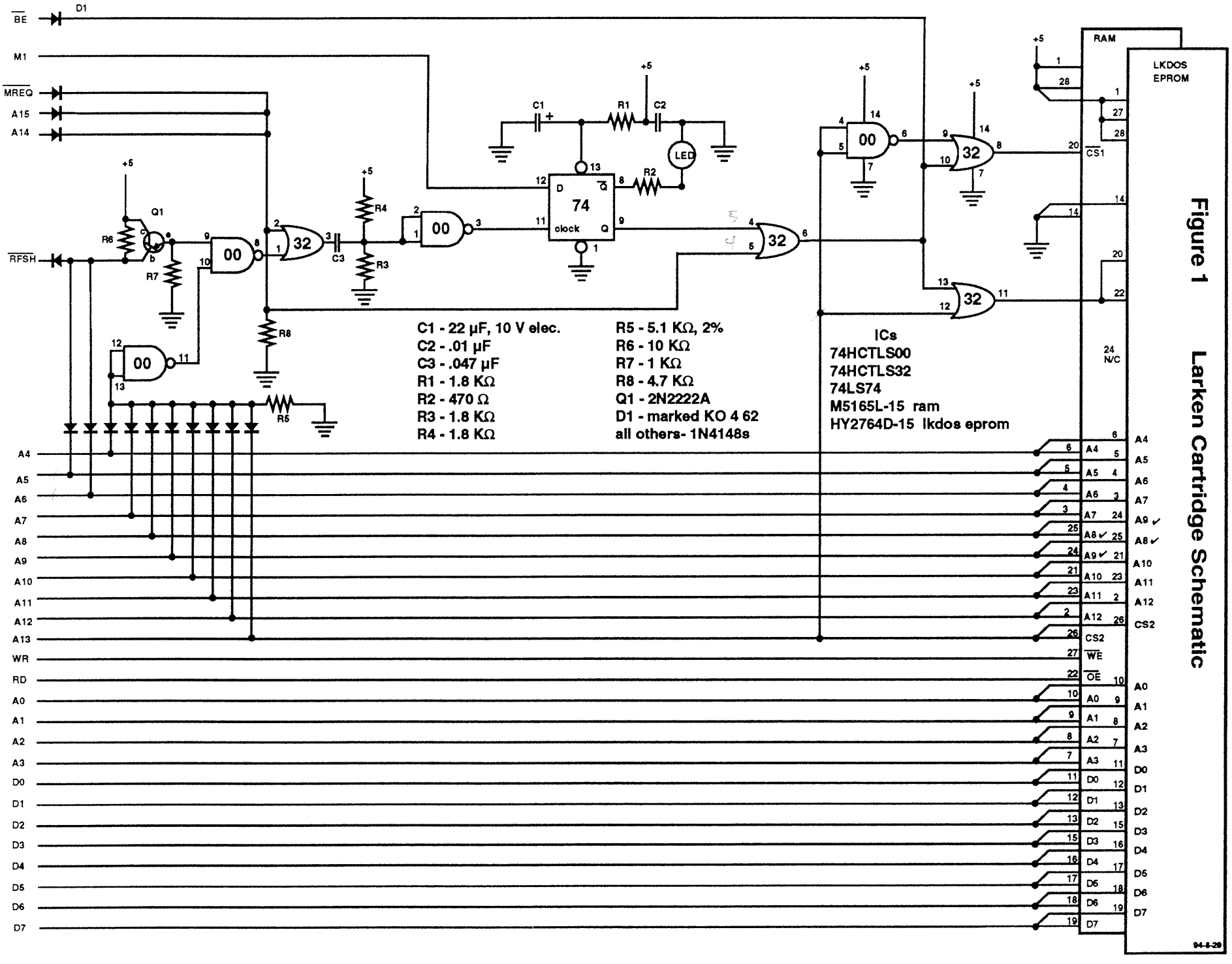


Figure 1 Larken Cartridge Schematic

You may need to "sharpen" the point on your soldering iron. Where the wires cross they have a tendency to fuse together. Use a toothpick to separate them to avoid a "crossed wires" problem. Before you install the chips you should verify that all the connections are proper using an ohmmeter or continuity tester. After the circuit has been tested and works properly carefully coat the wires with clear finger nail polish to coat any bare spots and hold the wires apart.

It is imperative that you use the chips called out on the schematic. Due to differing values required to activate "high" and "low" some substitutions with faster chips will not work! (Lesson learned. Thanks to Nazir Pashtoon for explaining this.) HCT's can be used where HCTLS's are called for and HM6264P-15 works in the ram spot. The small diodes, 1N4148's, should be at least as good as Radio Shack P/N 276-1122. Since the circuit is sensitive "bargain diodes" should be avoided.

Resistors R1, R3 and R4 are 1.8K which is not a standard value found at Radio Shack. I bought a package of 100 elsewhere so I have spares! If anyone is serious enough to start construction on a home-made dock board send me a SASE and I will send (3) 1.8K's and a few of the little "springs".

Figures 3 and 4 are a representation of the traces on the Larken board. They are included for reference only. My drafting skills seem to fade with each birthday. If I can become proficient with the CAD#3 program maybe a useful printed circuit layout could come later.

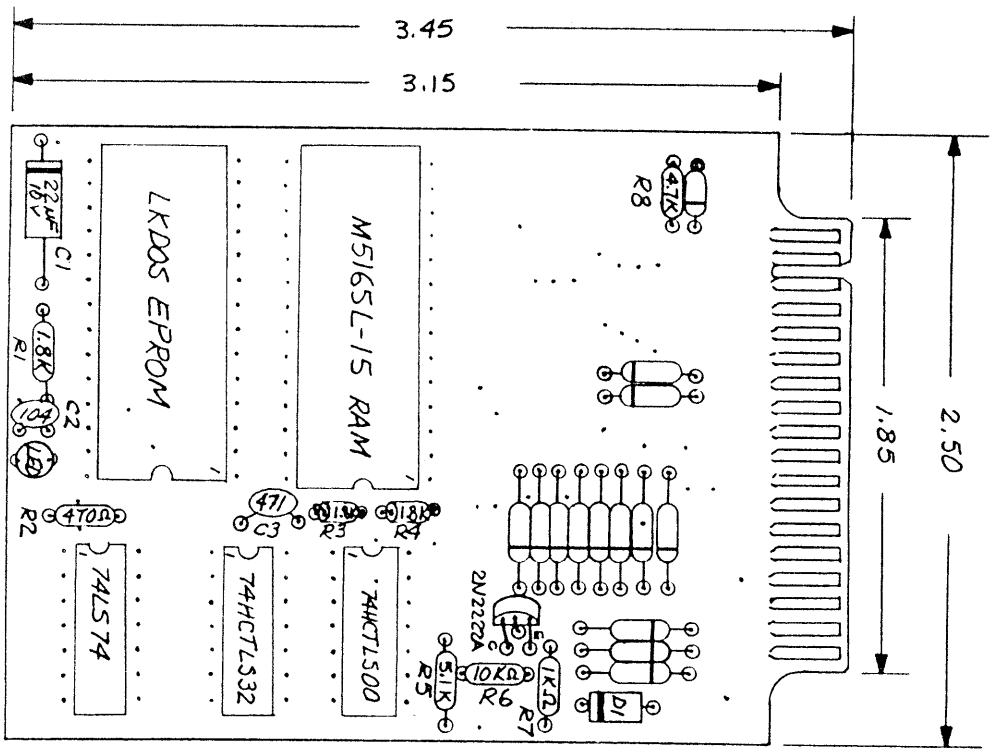


FIGURE 2

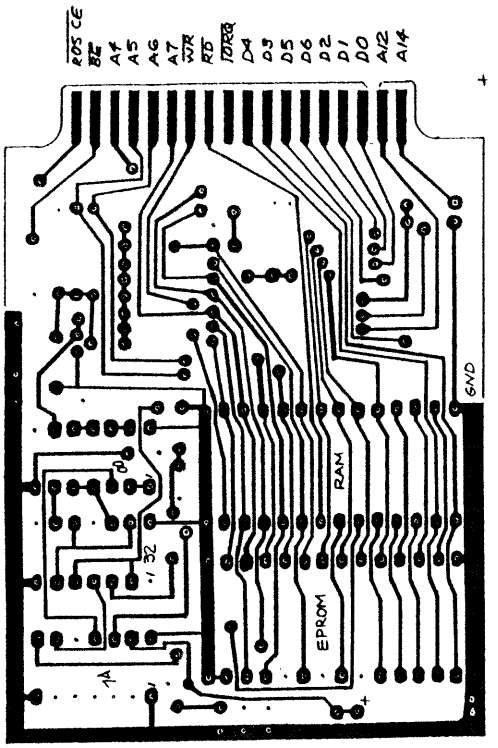


FIGURE 3 SOLDER SIDE TRACES

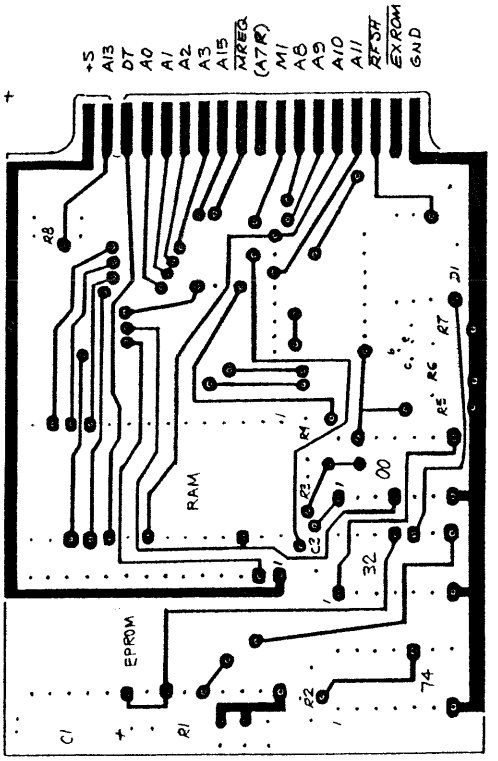


FIGURE 4 COMPONENT SIDE TRACES

WANTED/ FOR SALE/ OR HELP NEEDED

1) "Do you know of anyone that is using the Timex Disk Drive System (TOS)? I have a Seikosha SP-1000AS serial printer. I can get this to print using MSCRIPT and Vernon Tidwell serial patch in text but not graphics. I do not have the owners manual for this $4.00 printer so I do not have the print codes." HELP! A.E. Green, 4600 E. Hillsborough Ave., Tampa, FL 33610.

2) HELP WANTED...."My interest is with the TS1000 computer.....(a) has anyone come up with a battery pack to run the TS1000 away from a source of 110v AC power? My experiment with a single 9 volt transistor radio battery did not work, perhaps several of these batteries connected in parallel to increase the current available is the answer? (b) Along the same lines, can you suggest a small AC/DC portable TV that can be used in the absence of 110v AC power (brand, model #, screen size, price) that makes a suitable monitor? I also have a TS2040 printer but no power supply, can you suggest a source? I'm also looking for sources of TS1000 programs on cassette, books, magazines, and user group newsletters, can you help?" Doug Wagoner, E. 4825 St. Anthony Lane, Post Falls, Idaho 83854.

3) FOR SALE: (a) Fully assembled Hunter Board to provide extra non-volatile memory for the ZX81 or TS1000, for sale for $30. (b) Memopac 32K ram pack for the ZX81 or TS1000 or TS1500, new in box with docs for sale for $20. (c) Two Beeper kits for the TS1000, ZX81 or TS1500 to give sound for keypresses for sale for $10 each. (d) Brand new A & J wafers for the TS1000 model, clear, for sale for $2 each, minimum of 3 per order. Send check or cash to Eliad Wannum, c/o UPDATE! Magazine, P.O. Box 1095, Peru, IN 46970.

4) Does anyone still use the Zebra disk system? I just came by two. Les Cottrell, 108 River Heights Drive, Cocoa, FL 32922-6630.

5) FOR SALE: I just got a new IDE drive set up for my QL, so am now looking to sell my Falkenberg Hard Drive System. It includes the drive interface board, MFM Omti Controller, either a bus driver board for a Gold Card or one for a Trump Card (let me know which you will be using it with) an MFM cable and a hard drive case with power supply. It all works great; it is just that I only have so much room, and need to make room for new stuff, while making sure that someone can get good use out of my used. I am selling ALL OF THIS for the low price of $225. It will easily work on any MFM hard drive from 20 upto 416 MEG of memory! This is a great deal for someone. Send me either a money order, or I can ship it C.O.D. Eliad Wannum, c/o UPDATE! Magazine, P.O. Box 1095, Peru, IN 46970.

AS ALWAYS ADS OF THE SORT LISTED ABOVE ARE FREE FOR SUBSCRIBERS TO UPDATE! MAGAZINE. THESE ARE FOR INDIVIDUAL USERS, NOT FOR DEALERS TO USE TO ADVERTISE THEIR COMPANIES (FOR THOSE PLEASE MAKE ARRANGEMENTS FOR AN AD).

REVIEW OF Chris Boutal's GENEALOGIST 3

by Hugh Howie

In a group of people gathered around the glowing embers of a fire of a winter evening, the talk can touch on many things, and invariably someone will say something which prompts another to say "I remember---" and the talk will often turn to reminiscences of a family member, relation, or acquaintance, who did this, that or something else, then the tales start to flow as to who what where and when. And that is where a programme such as GENEALOGIST 3 comes into its own.

This is a programme by Chris Boutal of the United Kingdom. Great Britain. England, Scotland, Ireland, Wales, call it what you wish. The lands of history and legend. The lands of brave Knights and Fair Damsels. The lands from where so many of us in Canada and the USA came. The land which so many still call "HOME" But do we really know where we came from?

We all know some of our ancestors, but can we put it together in some order and sequence, with the correct dates and times and facts? Who married who, and who begat who?

GENEALOGIST 3 is the programme to do this for you. All you have to do is provide the data and enter it, and the programme takes over and does all the work for you.

Enter names and dates and relationships, and before you realise what has happened you will be able to see a "Tree" grow from almost nothing.

The system is very comprehensive. All your family records, Births, Deaths, Marriages, Burials are stored, and can be displayed on screen, or sent to the printer in many different formats to suit your specific requirements.

GENEALOGIST 3 will keep track of census returns and certificates, and any other detail you have in mind. It has very complete search and indexing facilities.

You want to know your ancestors? your pedigree? Who are the heads of various family groups? Who you are? Where you came from? No, it will not tell you where you are going, that's for you to solve. But I can vouch for this, it will make you want to KNOW the past! You will spend so much time in research into your family background, you will leave a lot of things unattended.

From the various Family Groups which you assemble as you enter data, a tree can be generated, and displayed on screen or paper for your perusal and reference. You can make amendments to the tree as you look at it, they can be made here or in the Family Group format. The Tree can be sent to the printer and this will give you a wonderful display of all the Ministers and Rogues and Vagabonds and Highwaymen, who are part of your heritage.

GENEALOGIST 3 is for the Pointer Environment, but can also be operated very efficiently from the keyboard if you don't have a mouse. Selection of facility can be made by using the arrow keys, or by letter selection from the menus, or mouse. Very versatile.

I have had Genealogist version 1 and version 2. When Version 3 came along I debated whether I needed it, and after a lot of poking around in the recessed area of the wallet - I came up with a few moths and the necessary moolah to purchase version 3, and I have not regretted it for one instant.

It would appear that when so much keyboard work is required, that the constant work of moving a hand from mouse to keyboard and back again, would be rather tiresome, but this I have found not to be the case. When you are entering data, you will find that the arrow keys are fine and dandy for moving around. But when you start to RESEARCH the data, looking for cross-references etc., does that mouse ever scoot around the screen! Definitely - the mouse is a great advantage - but don't forget, if you have no mouse the programme still provides all the facilities required, and at a goodly speed also.

REVIEW OF Chris Boutal's GENEALOGIST 3

by Hugh Howie

Speed, now that is something else to be discussed. The full programme requires at least 512 Kb memory, so that lets out Microdrive and bare bones QL. With Trump Card, you have the extra speed, and also the extra memory required. With the Gold Card which is what I use, I find that the programme is fast enough for me. With the Super Gold Card? I am afraid you might burn up the screen with the extra speed the Super is reputed to have.

But then again, we have an author who thinks of those with less than the epitome of QL's. We have an author who thinks of the person who only has the basic 128k machine (are there any left?) There must be a few somewhere as Chris has provided a Budget version of GENEALOGIST for those people. And this Budget version is on MDV or Disk! You want - you got!

Now say you have been working on an Archive based family, and having problems transferring it to this new system, then for a modest fee (very modest) Chris will convert that to GENEALOGIST for you.

For those who have been using one of the earlier versions of GENEALOGIST, the data made with those versions will not work with Version 3, but once again, all is provided for, there is a conversion programme provided. And it really does work, as I used it myself.

In the manual there is a tutorial family for you to type in, and after following the tutorial, you will have no problem in getting down to your own task. The manual is very well written indeed, with easy and clear instructions as to what to do, and when. There is an excellent "Command Reference" Section, where you can see what each command does, and where it is called from. Printed on pink paper in nice clear distinct black type, it may be thought that the pink paper could be a bit hard to read, but this is not the case, the manual is easy to read, easy to follow, and of great precision.

The programme will take all the data you like to enter, from the heads of families, which is all that is really required, but also all your brothers and sisters and aunts and cousins. Just about anything you want to put in.

If you want a screen or paper copy of a family tree from the start, or from a midway point, to a midway point - that is what is available.

Let me say here and now, when I say you can have a copy, I mean that you can have a copy ON SCREEN or on PRINTER, so those terms are synonymous. When using the printer the option of Normal or Condensed type is available.

GENEALOGIST 3 will provide a geographical map of occurrences. This is based on the grid system in use in British road maps, and when I suggested to Chris that the programme was reaching the far corners of the earth where that grid system might not be in use, and that a Latitude & Longitude option be made available, Chris agreed and said that he would look into this, and perhaps introduce it at a later date. But don't let this stop you from acquiring the programme. This is an enhancement that I do not require, and I wonder how many really do. I only introduce those comments to show how Chris has tried to provide everyone with what anyone would like. This "map" facility, will even tell you the distance between two points when using the grid system.

I almost forgot to tell you that it will even provide pictures to go with a family group. Say you a have a picture of your old Uncle Ned and Aunt Sarah at Xmas, you do have that picture? Then by use of a Video camera and a digitiser, you can save that picture on disk, and later project that picture on screen. There are even a couple pictures provided as an example. I can see this programme taking the place of slide shows. Do you remember them?

For those interested in Kings and Queens of Britain, a Royal Tree is provided, just so you see what a tree looks like.

REVIEW OF Chris Boutal's GENEALOGIST 3

by Hugh Howie

Do you want to know who is related to whom, and what that relationship is? You got it! D'ya really need to know that "Jo Blo is the second cousin twice removed from Jenny Jewell"? Or that "Dan is the great-great-great-grandfather of Dave? The cross-reference is there.

For those occasions when the spelling of a surname has been changed over the years, that also has been taken care of, as multiple spellings are recognised.

The user is even allowed to make up a customised Research Report to suit ones own requirements, as before making the report, you are allowed to select the details (from a list of thirteen) you wish to appear in the report, but not only that, you are permitted to select whether the report should appear in row or column format.

There is even a "Verify" mode provided. Say you enter the birth of a parent as 1900 and the son as being born in 1905, this "verify" will tell you "40 and 35 have an unlikely generation gap". Plus many other things. (The programme gives each family member its own unique number)

Much of the data is available for export to the Psion suite of programmes. This is to enable you to do further processing that GENEALOGIST can't do. For example the Family Network and Research reports can be exported to Archive. Tree data can go to Abacus. Place/Time data can be exported to Easel. Those same files can be imported back into Genealogist from Psion.

The programme comes set up for a red screen with white lettering, but if you don't like that you are provided with a facility to change that to white on black or almost any other combination you can think of. You make your own colour combination choice for any window.

As I am also writing a lot stuff dealing with such things as movements, occupations, anecdotes, family histories and a whole lot of interesting, and some not so interesting facts, that kind of stuff, I have GENEALOGIST and text^{87} plus^{4} running together, this way I can quickly switch from one to the other as I progress. In fact it is possible to have two TREES going at the same time - but watch your memory!

Now to a neat little thing. As the programme starts, there is a little square clock that comes onto the middle of the screen, with real hands! and it keeps popping up as you change from one facility to another - just to remind you of the passage of time! Neat neat neat.

You don't want to keep a record of your family? Then use this programme to keep track of those cats and dogs and horses you breed, it can be used for just about any record of ancestry or breeding you wish.

Now where is this programme available? Why, from my old friend:-

DILWYN JONES COMPUTING
41 BRO EMRYS, TAL-Y-BONT, BANGOR
GYNEDD, UNITED KINGDOM, LL57 3YT

The last price I have is from the December 1993 Price List but I would advise you to check first, I don't think the price has gone up in that time, but then the time between writing and publication is not always predictable.

Genealogist 3 (Pointer)	£60.00 Disk only
Genealogist 2	£30.00 Disk only
Genealogist 1	£19.50 Disk only
Genealogist Budget	£12.00 MDV/Disk

But if you really are serious about this, I would recommend the version 3 (Pointer)

One final thing which I consider to be very important, what happens when you have a problem? Why you write to Chris Boutal and the answer is on your desk before you know it. Service is terrific. But then if you have a programme as terrific as this, then you would also expect the service to be terrific - and it is.

940910

QLAMBer UPDATE

by Al Feng

By the time you read this, the latest "variation" of the QLAMBer program should have been out-and-about for several months.

The good news is that *variation 2x001* can automatically identify the "_doc" extension of a standard Quill/QLWP file, and produce a more easily read HARDCOPY than before. The HARDCOPY output will be something like this, but full width, and in the font to which your printer is set:

vrm1qdf0

&
H
page nnn
followed by your document ... lacking the effects of control co des; but, having paragraph breaks, and a Form_Feed at the end of the meaningful text.

This does not replace printing from within Quill/QLWP. It is me ant to make the program's HARDCOPY function more useful.

Note that there is no word-wrap, *per se.* Words are split if they are incomplete before the end of a line, and are finished after the carriage return.

Program LISTings & _lis/_txt files should be SPooLed within QLAMBer from a file SAVEd/COPied to RAM1_ (for example).

Adjust DATASPACE Please

If you happen to get an "adjust DATASPACE please" message when trying to use QLAMBer it probably means that your QL has a GOLD CARD (or, faster?) with MINERVA 1.97 (or, higher?) and have implemented a "misbehaving" TURBO TOOLKIT (or, the like -- for example v3.20) along with pre-emptive TK2_calls (e.g., SDP_KEY), and/or using a statement within a long SuperBASIC BOOT. WHEW!

The main culprit is apparently the TURBO TOOLKIT in tandem with the later MINERVA code(s). I have to find my MINERVA manual/disk to verify this, but I was told that there is a substitute TURBO TOOLKIT. A temporary fix is to use an older TURBO TOOLKIT (such as v2.05), or to simply "EXEC_W flp1_QLAMBer" from the command prompt.

Sorry for this inconvenience. As many of you may know, the MINERVA ROM code gets more and more finicky (i.e., less sloppy ... forgiving) as the version number gets higher.

HAPPY TRAILS, AND COMPUTING, TO YOU ...

T/SNUG QL Public Domain Library

ABAtra_exp	Psion's Abacus file format explained
ANSISR_zip	Strips ANSI codes, uses TK2 and QLib
BASCON_zip	SB Extension for Linking in QJump Configuration Blocks
C68dl_zip	C68 4.12 disk 1. 'C' Main system disk The "STANDARD for 'C'"
C68d2_zip	C68 4.12 disk 2.'C', Extras etc.
C68d3_zip	C68 4.12 disk 3.'C' Master doc disk-C68
CFG_zip	Place QPTR config. blocks into 'C'
CHECK_zip	SB ext. checks for non-numeric INPUT
CIRCUI_zip	Misc. circuit diagrams and pin-outs
CRITMS_BAS	Critical Mass Game from 88 QL World
DISKS_txt	File on using Teac 2.88Mb disk drives
DJEP_zip	DeskJet printer utility, prints envelopes
DJWDEM_zip	Demo copies; DiscOVER, MULTI-DiscOVER, TexTIDY & FLPClone
EMAIL_txt	INTERNET, addresses of other QL'ers
FILEIN_zip	Pointer environment utility
FNTRC_zip	Source files (SB and Metacomco Assembler) FontEditor v.2.3
FNxxxx_zip	QL InterNational FidoNet Echo Mail messages, 8 merged files so far.
FONTED_zip	FontEditor v.2.3 for the Sinclair QL
HDRRST_BAS	Restores file header of transferred files
HISTDV_zip	Last line recall history device
IBMDSK_zip	Access IBM disks. TK 2 and 3.5 " drives
LHQ_zip	LHQ v 1.0. File archiver and extractor
MAIL_zip	Mail list program for ARCHIVE
minnie_zip	File detailing the Minerva ROM 1.93+
MODPSI_zip	Mods for the PSION suite of software
QLGIF_zip	Code/decode GIF. By Carlos Delhez 1993
QPACER_zip	QPac2 8001 Boot Generator v 1.1
Point_zip	Setting up as QL FidoNet Echo Point
PullDown_zip	Creating PULL-DOWN/POP-UP screens
QED_zip	QED 1.01 A general purpose editor
QEM_zip	First? version of QEM
QEM040doc_zip	Doc files for QEM v4.0
QEM24_zip	QEM v2.4 early version of QEM
QEM35U_rip	QEM v3.5 upgrade QEM 3.0+ to 3.5
QEM36U_rip	QEM v3.6 upgrade from v3.5
QEM40_zip	QEM v4.0 latest version of QEM?
QEMV32_zip	QEM v3.2 . Excellent terminal program
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 & demo
QSI_zip	Determines speed, QXL, Gold Card
QUADRI_zip	Quadratic Equation Plotter
Quanta1/2_zip	Quanta BBS Messages from the UK, 2 merged files so far
Qulltra_exp	Psion's QUILL file format explained
RDS_zip	3 SuperBasic progs rand dot stereograms
RELYQL_txt	Text describing modifications to the QL
RESCUE_zip	Floppy Disk Rescue 8, String Search
Ruleta_zip	Roulette game
SAVER_LZH	Pointer screen saver. Set up as a hotkey
SPECKE_zip	4 graphic files of the Spectrum keyboard
SPECTATR_zip	Spectator v1.00 - A Spectrum emulator
Unzip30	v3.0 The Latest One. **Needed For These Files**
UNZIP9_exe	Unzip program V 0.96 by Jan Bredenbeek, EXE form (early version)
VERS_zip	Provides info about other QPTR progs.
VIEW_LZH	Pointer environment utility
XCHANG_zip	Psion's Xchange v3.90 now PD (Quill, Archive, Abacus, Easel in one package)
XPR210_zip	Latest file transfer protocols for EM v4.0
XPRLIB_zip	XPR library for QEM v3.5, for 3.5 & 3.6
ZIP22_zip	Zip/Unzip version 2.2 (revised)
ZIP30_exe	Zip v3.0 (the \"ES\" version) (re revised)
Zip_txt	Revised Docs on how to use Zip/Unzip
ZIPUTIL_zip	Zip Utilities v3.0. Dealing with zip files
ZXTRICA_zip	Xtricator v1.10 ZX-81 emulator

Spectrum/QL Related Files for the PC/Clone

ENV_bin	Code compatible with QXL and SMSQ
JPP_zip	Excellent Spectrum Emulator for 386SX or greater PC
Make_C_zip	C68 files MAKE & CPC for the QXL
QDIR_zip	Looks at QL disk DIRs on a PC
QLSCR_zip	PC program to view screen files, can convert to GIF
QLTOOL_zip	Lets a PC look/examine QL disk
QXLfmt_zip	For the QXL card. Converts a disk to QDOS format (same as FORMAT?)
Z80V201A_zip	128/48K ZX Spectrum Emulator V2.01 by Gerton Lunter
Z80V201B_zip	LATEX and POSTSCRIPT DOC files for v2.01 Z80 Spectrum

For the MAC

CP2MC1_zip	Converts text/Quill files to the MAC.

Donated to T/SNUG

by Paul Holmgren

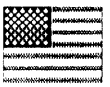
Files were downloaded from various BBS's

Set of 9 5.25" disks. All for $9.00

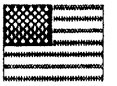
Order from:

John Donaldson
835 Foxwood Cir
Geneva IL 60134-1631
708 232-6147

QBox-USA



810-254-9878



24 hours a day, 300/1200/2400 baud
(a new high speed modem, up to 14400 baud will be online soon)

QBox-USA is a BBS set up by QL enthusiasts for QL enthusiasts. The sytem runs entirely on a Sinclair QL. We maintain a link (Fidonet) with European BBS's that carry QL related message areas. With QBox-USA North American QL users can keep in touch with our fellow QL users in Europe and keep up with the latest developments for the QL and QDOS. We are currently carrying the following message areas: INTERNATIONAL QL, MINERVA, QUANTA and QBOX. There are also file areas that contain many programs available for download. Just a few of the files available are: QIPI, QFAX, Zip/Unzip, QLTerm, QLGIF and many more! There is no charge to use QBox-USA other than normal phone call charges. Please give QBox-USA a try! We look forward to your call!

produced using Page Designer 3

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

CABLE COLUMN

By Bill Cable

ARCHIVE SERIES
PART 18 : FormEd - A FORM EDITOR AND PRINTER FOR ARCHIVE DATABASES (2nd and final part)

FormEd as listed last time worked well for displays but the printing part had not yet been completed. This time we will list the complete program including what was listed last time. This has been a rather involved project but I am pleased with the result. It can nicely display and print any ARCHIVE Database in many different and useful ways. The user interface is easy to use and different print fonts are supported (Bold, Condensed, Italic, Underline, Double Wide). Epson print controls are used. If they don't work for you get out your printer manual and alter procedure 'set_fonts' so they do. The procedures listed here need to be merged with the group3 procedures finished in Part 14 of this series. Notice that procedure 'close_all' in group3 is incorrect. The one listed here is correct. Correct it in group3 procedures or it will replace the correct one when you merge the group3 procedures in. Those wishing to skip all the work send me a disk and return postage and I will send you a copy of Formed.

Forms can be designed on screen for any ARCHIVE database for browsing and printing and saved for later use. Its primary purpose will be for printing out the forms or reports to your printer such as address labels from an address database or financial summaries from a financial database. It is slow at least if you don't have a Gold Card or better but handy. Get into the ARCHIVE editor and start at it. When done merge the group3_prg procedures and save as 'formed'. To run simply load into ARCHIVE and type : formed<ENTER>. Play to learn all its features and uses. It is quite flexible.

```
proc closey;l$
close l$
endproc
proc close_all
while 1: close : endwhile
endproc
proc formed
rem  Written by Bill Cable  6/94  and released to public domain
error form_check: if ermum():formed1: else :formed3: endif
endproc
proc formed1
mode 0:setup: error close_all:set_fonts
let h$="FormEd - A Form Editor, Creater, and Printer for ARCHIVE Databases"
heady;h$: spoolon screen :form_instruct: spooloff
yorn;21,"Print these instructions": if ans$="y"
lprint h$: lprint tab 55,"Bill Cable  9/94": lprint :form_instruct: endif
device;21,"Location of form databases (form_fm1,form_fm2)  eXit to stop"
if ans$="": error close_all: mode 1: stop : endif : let fd$=ans$: let fn$="form_fm"
error looky;fd$+fn$+"1","f1"
if ermum(): if ermum()<>100:erry;23,"accessing "+fd$+fn$+"1": endif
yorn;23,"Create form databases on "+fd$+" now": if ans$="n":formed: return
else : error form_create: if ermum():erry;23,"making "+fd$+fn$+"1"
formed: return : endif : endif
looky;fd$+fn$+"1","f1": endif
error looky;fd$+fn$+"2","f2": if ermum():erry;23,"accessing "+fd$+fn$+"2"
acky;23,"You must fix":formed: return : endif :formed2
endproc
proc formed2
heady,"FormEd - Choosing a Form": use "f1": if count()>0
yorn;4,"Print list of currently defined forms in "+fd$+" Form File now"
if ans$="y":form_print1: endif : endif : error closey;"s"
let ans$="": while len(ans$)<1 or len(ans$)>8
inpy;6,"Name of form [1-8 characters or <ENTER>=quit]"
if ans$="": error close_all: mode 1: stop : endif : endwhile
let form$=ans$: use "f1": locate form$: if form_name$<>form$
yorn;8,"Form '"+form$+"' not defined. Add it now"
if ans$="n":formed2: return
else :form_new: close "f1":looky;fd$+fn$+"1","f1": locate form$: endif : endif
let sd$=dbf_loc$: let sn$=dbf_name$: let ans=1: while ans and sn$<>""
error looky;sd$+sn$,"s": let ans=ermum()
if ermum():erry;10,"Problem accessing "+sd$+sn$
yorn;11,"Change device or name": if ans$="n":formed2: return : endif
device;12,"Database": let sd$=ans$
defy;14,"Database name (with extension)",sn$: let sn$=ans$: endif : endwhile
yorn;20,"Print list of fields of "+sd$+sn$+" used in "+form$+" now"
if ans$="y":form_print2: endif :formed3
endproc
proc formed3
mode 0: let flg=1:form_display
while 1: print at 0,3;upper(form$)," form using ";sd$+sn$;
```

```
print " with ";count("s");" selected"; tab 57;
key_choice;23," FORMED MENU ","","Edit Print Browse Select/order eXit","epbsx"
if ans$="x":yorn;23,"Work with another form": if ans$="y":formed2: return
else : error close_all: mode 1: stop : endif
else : if ans$="e":form_edit;1,high,1,1,wide,1:form_display
else : if ans$="p":form_print
else : if ans$="b":form_browse
else : if ans$="s": use "s": mode 1: cls : print
print "FormEd is halted. Type desired ARCHIVE Select, Order, or Reset Command"
print "Then type : formed<ENTER> to restart": let flg=0: stop
endif : endif : endif : endif : endif : endwhile
endproc
proc form_browse
local a$,f$,i: while 1: use "s"
key_choice;23," BROWSE ","","Next Back Scan Rec# Find More Print eXit","nbsrfmpx"
if ans$="x": use "f1": return : endif : let a$=ans$
if a$="n": next : else : if a$="b": back
else : if a$="s":msg;23,"Scan by Browsing through 10 records"
let i=1: while i<10 and not eof(): next :form_fvshow1;"": use "s": let i=i+1: endwhile : next
else : if a$="r":inpy;23,"Move to record [0-"+str(count("s")-1,2,0)+"]"
position val(ans$): else : if a$="f":inpy;23,"Enter text to find": let f$=ans$
msg;23,"finding '"+f$+"'": find f$
else : if a$="m":msg;23,"more finding '"+f$+"'": continue
if not found():acky;23,"More '"+f$+"' not found": endif
else : if a$="p":form_print
endif : endif : endif : endif : endif : endif : endif :form_fvshow1;"": endwhile
endproc
proc form_check
use "s"
endproc
proc form_color;l,i$
key_choice;l,"","," for "+i$+" color","[Black,Red,Green,White]","brgw"
if ans$="b": let ans$="Black - 1": else : if ans$="r": let ans$="Red   - 3"
else : if ans$="g": let ans$="Green - 5"
else : let ans$="White - 7": endif : endif : endif
endproc
proc form_create
msg;23,"creating form file 1 : "+fd$+fn$+"1"
create fd$+fn$+"1" logical "f1"
form_name$
form_desc$
form_paper$
form_ink$
form_height
form_width
form_line
form_text$
form_font$
dbf_loc$
dbf_name$
dbf_desc$
endcreate
order form_name$;a,form_line;a: close "f1"
msg;23,"creating form file 2 : "+fd$+fn$+"2"
create fd$+fn$+"2" logical "f2"
field_form$
field_line
field_column
field_name$
field_number
field_start
field_width
field_format$
field_join
field_join$
endcreate
order field_form$;a,field_line;a,field_column;a: close "f2"
endproc
proc form_display
```

```
use "f1": locate form$: let wide=form_width: let high=form_height
let los=int((21-high)/2): let cos=int((79-wide)/2)
let fp=val(form_paper$(9)): let fi=val(form_ink$(9))
if fp=1: let dp=3: let di=7: else : let dp=1: let di=7: endif
paper dp: ink di: cls : let k=form_width
let i=1: while i<high+1: locate form$,i
print at i+los,1+cos; paper fp; ink fi;form_text$;
let i=i+1: endwhile :form_fvshow1;""
endproc
proc form_edit;sl,nl,al,sc,nc,ac
local ll,lc,i$,i,j,k,j$
close "f1": open fd$+fn$+"1" logical "f1": locate form$,1
close "f2": open fd$+fn$+"2" logical "f2": locate form$,1,1: use "f1"
if sl>22: let sl=22: endif : if nl>22-sl: let nl=22-sl: endif
if sc>79: let sc=79: endif : if nc>79-sc: let nc=79-sc: endif
let nl=sl+nl-1: if al<sl or al>nl: let al=sl: endif
let nc=sc+nc-1: if ac<sc or ac>nc: let ac=sc: endif
let ll=al+los: let lc=ac+cos
while 1: use "f1": locate form$,al
print at 23,1;"EDIT FORM - Move and type text, <CTRL>1 Field, <CTRL>2 Font, <CTRL>0 Exit";
print tab 80; at 0,68;"[";al;",";ac;"] (";form_font$(ac to ac);")"; tab 79;
print paper fp; ink fi; at al+los,ac+cos;chr(14);
let i$=getkey(): let i=code(i$)
if i=144: close "f1": close "f2":looky;fd$+fn$+"1","f1"
looky;fd$+fn$+"2","f2": use "f1": return : endif
if i<>145 and i<>146
if i=10 or i=11: let j=3: else : if i=12 or i=13: let j=10
else : let j=1: endif : endif
if i=2 or i=10: let al=al-j: if al<sl: let al=nl: endif
else : if i=3 or i=11: let al=al+j: if al>nl: let al=sl: endif
else : if i=4 or i=12: let ac=ac+j: if ac>nc: let ac=sc: endif
else : if i=5 or i=13: let ac=ac-j: if ac<sc: let ac=nc: endif
endif : endif : endif : endif
if i>31 and i<127: print paper fp; ink fi; at al+los,ac+cos;i$;
locate form$,al: let j$="*"+form_text$+"*"
let j$=j$(1 to ac)+i$+j$(ac+2 to len(j$))
let form_text$=j$(2 to len(j$)-1): update
let ac=ac+1: if ac>nc: let ac=sc: endif : endif
if ll<>al or lc<>ac: let ll=al: let lc=ac: endif
else : if i=146
key_choice;23," SET FONT","","Normal Bold Itallic Underline Wide Condensed","nbiuwc"
let j$=upper(ans$):defy;23,"Font Width [1 to "+str(nc-ac+1,2,0)+"]",str(nc-ac+1,2,0)
let ans=val(ans$): if ans<1 or ans>nc-ac+1: let ans=1: endif
let i$=form_font$: if ac>1: let k$=i$(1 to ac-1): else : let k$="": endif
let form_font$=k$+rept(j$,ans)
let i=len(form_font$)
if i<len(i$): let form_font$=form_font$+i$(i+1 to nc)
endif : update : else
key_choice;23," FIELD EDIT","","Add Remove Show List eXit","arslx"
if sn$<>"" and ans$<>"x": if ans$="a":form_fadd;al,ac,nc
else : if ans$="r": use "f2": locate form$,al,ac
if field_form$=form$ and field_line=al and field_column=ac
yorn;23,"Remove "+field_name$+" from specifed position": if ans$="y"
print at al+los,ac+cos; paper fp; ink fi;rept(" ",field_width);: delete : endif
else :acky;23,"No field value starts at current cursor position": endif
else : if ans$="l":yorn;23,"Listout fields on form to printer now"
if ans$="y":form_print3: endif
else :form_fvshow1;chr(5):acky;23,"Showing field widths by underline"
endif : endif : endif : endif : endif : endif : endwhile
endproc
proc form_fadd;al,ac,nc
use "f2": let field_form$=form$:form_fpick
let field_name$=fieldn(field_number,"s")
let field_line=al: let field_column=ac
defy;23,"Start at field charcter","1": let ans=val(ans$)
if ans<1 or ans>255: let ans=1: endif : let field_start=ans
defy;23,"Field width",str(nc-ac+1,2,0): let ans=val(ans$)
if ans<1 or ans>nc-ac+1: let ans=nc-ac: endif : let field_width=ans
if fieldt(field_number,"s")
```

```
key_choice,23,"","Field format","Normal Upper Lower","nul"
else :key_choice,23,"","Field format","General Decimal(2) Integer","gdi"
endif : let field_format$=upper(ans$)
yorn,23,"Will this field be joined with a field following it"
if ans$="y": let field_join=1
inpy,23,"Joining symbol(s) (ie. ' ' or ',' or '/',<ENTER> = none)"
let field_join$=ans$: else : let field_join=0: let field_join$="": endif
yorn,23,"Add '"+field_name$+"' now": if ans$="y": append
if field_join: print at 22,1;"Joins ";field_name$; tab 79;
form_fadd,al,ac,nc: return : endif
locate form$,al,ac:form_fvshow2;"":form_fvshow3;"": endif :liny;22: use "f1"
endproc
proc form_fpick
while 1:inpy,23,"Field of interest [0-"+str(numfld("s")-1,2,0)+"]"
let ans=val(ans$): if ans<0 or ans>numfld("s")-1: let ans=0: endif
let field_number=ans:yorn;23,fieldn(ans,"s")
if ans$="y": return : endif : endwhile
endproc
proc form_fvshow1;u$
print at 0,57,"at Rec# ";recnum("s"), tab 79;
use "f2": locate form$: while not eof() and field_form$=form$
form_fvshow2,"":form_fvshow3,u$: next : endwhile : use "f1"
endproc
proc form_fvshow2;l$
local i,j,k,l,c,ls,fs,fw: let l=field_line: let c=field_column
let ans=field_number: if fieldt(ans,"s")
let ans$=fieldv(ans,"s"): if field_format$="U": let ans$=upper(ans$)
else : if field_format$="L": let ans$=lower(ans$): endif : endif
else : if field_format$="G": let ans$=str(fieldv(ans,"s"),4,0)
else : if field_format$="D": let ans$=str(fieldv(ans,"s"),0,2)
else : let ans$=str(fieldv(ans,"s"),2,0): endif : endif : endif
let ls=len(ans$): let fs=field_start: let fw=field_width
if fs>1: if ls<fs: let ans$="": else : let ans$=ans$(fs to ls): endif : endif
if len(ans$)>fw: let ans$=ans$(1 to fw): endif : let ans$=l$+ans$
if field_join: let ans$=ans$+field_join$: next
if l=field_line and c=field_column and not eof():form_fvshow2;ans$: return
else : back : endif : endif
if len(ans$)>wide: let ans$=ans$(1 to wide): endif
endproc
proc form_fvshow3;u$
local l,c,w,s: let l=field_line: let c=field_column: let w=field_width
if len(ans$)>w: let ans$=ans$(1 to w): endif
print at l+los,c+cos; paper fp; ink fi;u$;ans$, tab cos+w+c;u$
endproc
proc form_instruct
lprint " 1. This program allows you to create forms that incorporate your own"
lprint "    databases. Good for printing them in a structured way or as labels."
lprint " 2. Two databases (form_fm1 & form_fm2) hold form information. If they"
lprint "    don't exist already the will be created. Please back them up."
lprint " 3. The database you use on a form will be accessed for read only. Using"
lprint "    the form display you can browse or print it as selected/ordered."
lprint " 4. Each form has an identifying name (1-8 characters) and is saved for"
lprint "    later access. The form is Edited on screen by typing text at the"
lprint "    cusor position with curosr coordinates shown in [] in upper right."
lprint " 5. <CTRL>1 when editing allows Adding and Removing Fields at cursor."
lprint " 6. <CTRL>2 when editing to define Print Fonts for areas on form. Upper"
lprint "    right of screen show Font at cursor. Normal, Bold, Italic, Underline"
lprint "    Wide,and Condensed possible. For non Epson change procedure set_fonts."
lprint " 7. To do the ordering and selecting of your database the program is"
lprint "    halted so you can type in the Select, Order, or Reset command at"
lprint "    the ARCHIVE prompt '>' and then restarted by : formed<ENTER>."
endproc
proc form_new
close "f1": error openy;fd$+fn$+"1","f1"
let form_line=0: let form_text$="": let form_font$=""
while 1:heady;"Adding New form : "+form$: let form_name$=form$
print at 3,1,"Form information"
inpy,5,"Form description": let form_desc$=ans$
form_color,6,"Form paper": let form_paper$=ans$
```

```
print at 6,0;" Form paper color : ";ans$; tab 80;: let form_ink$=ans$
while form_paper$=form_ink$:form_color;7,"Form ink"
let form_ink$=ans$: endwhile
print at 7,0;" Form ink color : ";ans$; tab 80;
defy;8,"Form width (10-78)","78": let ans=val(ans$)
if ans<10: let ans=10: endif : if ans>78: let ans=78: endif
let form_width=ans: print at 8,0;" Form width : ";ans; tab 80
defy;9,"Form height (1-20)","20": let ans=val(ans$)
if ans<1: let ans=1: endif : if ans>20: let ans=20: endif
let form_height=ans: print at 9,0;" Form Height : ";ans; tab 80;
yorn;12,"Use information from a database on this form"
if ans$="y": print at 12,1;"Database information"; tab 80;
device;14,"Location of database to be used in form"
let dbf_loc$=ans$: print at 14,0;" Database location : ";ans$; tab 80;
inpy;15,"Name of database to be used in form"
if not instr(ans$,"_"): let ans$=ans$+"_dbf": endif
let dbf_name$=ans$: print at 15,1;"Database name : ";ans$; tab 80;
inpy;16,"Description of database": let dbf_desc$=ans$: endif
yorn;23,"Add form now": if ans$="y": append
let i=1: let j=form_height
let form_desc$="": let form_paper$="": let form_ink$=""
let dbf_loc$="": let dbf_name$="": let dbf_desc$=""
while i<=j: let form_line=i: let form_text$=rept(" ",form_width)
let form_font$=rept("N",form_width)
append : let i=i+1: endwhile : return : endif
yorn;23,"Still want to add a "+form$+" form": if ans$="n": return : endif
cls : endwhile
endproc
proc form_print
local i$,j$,h1$,h2$,br,er,lpp,fpp,norm$,bar$,bp,tp,atline,formct,pg$,nl: use "s"
let bp=recnum(): let norm$=rept("N",wide): let bar$=rept("-",wide): let nl=high: let ffp=0
let i$=str(count()-1,2,0): let j$=str(recnum(),2,0):liny;23
defy;22,"Begin print at record [0-"+i$+"]",j$: let br=val(ans$)
if br<0: let br=0: endif : if br>count()-1: let br=count()-1: endif
if br>recnum(): let j$=str(br,2,0): endif
defy;23,"End print at record ["+str(br,2,0)+"-"+i$+"]",j$: let er=val(ans$)
if er<br: let er=br: endif : if er>count()-1: let er=count()-1: endif
let pg$="n": let h1$="": let h2$="": let lpp=0
if br<>er:liny;23:yorn;22,"Print heading for the "+str(er-br+1,2,0)+" records": if ans$="y"
print at 23,1;"=> ";: input h1$:liny;23: print at 23,1;"=> ";: input h2$: endif
if h1$<>"": let tp=2: if h2$<>"": let tp=4: endif : endif
liny;23:yorn;23,"Fit forms on a page": let pg$=ans$: if pg$="y"
defy;23,"Lines per page ["+str(nl,2,0)+"-200","66": let lpp=val(ans$)
if lpp<nl: let lpp=nl: endif : if lpp>200: let lpp=200: endif
let fpp=int((lpp-tp)/nl)
if fpp>1:defy;23,"Form per page [1-"+str(fpp,2,0)+"]",str(fpp,2,0)
let fpp=int(val(ans$)): if fpp<1 or fpp>(lpp-tp)/nl: let fpp=int((lpp-tp)/nl): endif
endif : endif : endif
if er<>br and lpp
yorn;23,"Page length = "+str(lpp,2,0)+" , Heading = "+str(tp,2,0)+", Forms per page = "+str(fpp,2,0)
if ans$="n": return : endif : endif
use "f1": locate form$: next
let atline=tp+1: let formct=1
liny;22:msg;23,"printing": let ln=len(form_text$): let blk$=rept(" ",ln)
if tp: lprint h1$: if tp=4: lprint : lprint h2$: endif : lprint bar$: endif
use "s": position br: while recnum("s")<=er and not eof("s")
use "f1": locate form$: next : while not eof() and form_name$=form$
let l$=form_text$: let f$=form_font$: let l=form_line: use "f2": locate form$,l
let ans$="": while not eof() and field_form$=form$ and field_line=l
let fl=field_line: let fc=field_column
if fc>len(ans$)+1: let ans$=ans$+l$(len(ans$)+1 to fc-1): endif
form_fvshow2;ans$: next : endwhile
if len(ans$)<len(l$): let ans$=ans$+l$(len(ans$)+1 to len(l$)): endif
let lp=2: let sf$="": let ef$="": let lf$=f1.form_font$(1)
if lf$="B": let sf$=bolds$: let ef$=bolde$
else : if lf$="C": let sf$=conds$: let ef$=conde$
else : if lf$="I": let sf$=itals$: let ef$=itale$
else : if lf$="U": let sf$=unders$: let ef$=undere$
else : if lf$="W": let sf$=wides$: let ef$=widee$
```

```
endif : endif : endif : endif : endif
let i=1: while i<=ln
if i=1 and (ans$=blk$ or f1.form_font$=norm$): let i=len(ans$)
else : if lf$<>f1.form_font$(i): let i$=f1.form_font$(i)
let nsf$="": let nef$="": let lf$="N"
if i$="B": let nsf$=bolds$: let nef$=bolde$: let lf$="B"
else : if i$="C": let nsf$=conds$: let nef$=conde$: let lf$="C"
else : if i$="I": let nsf$=itals$: let nef$=itale$: let lf$="I"
else : if i$="U": let nsf$=unders$: let nef$=undere$: let lf$="U"
else : if i$="W": let nsf$=wides$: let nef$=widee$: let lf$="W"
endif : endif : endif : endif : endif
lprint sf$+ans$(lp-1 to i-1),ef$,; let sf$=nsf$: let ef$=nef$: let lp=i+1
endif : endif : let i=i+1: endwhile
if lp<=ln: lprint sf$;ans$(lp-1 to ln);: endif : lprint ef$: let ef$="": let lf$="N"
use "f1": next : endwhile : use "s": next : let atline=atline+high
if pg$="y": if formct>=fpp: let i=atline: while i<=lpp: lprint : let i=i+1: endwhile
if tp and recnum()<=er and not eof()
lprint h1$: if tp=4: lprint : lprint h2$: endif : lprint bar$: endif : let formct=1: let atline=tp+1
else : let formct=formct+1: endif : endif
endwhile : use "s": position bp: use "f1"
endproc
proc form_print1
msg;23,"Printing listing of defined forms"
lprint : lprint "FORMS DEFINED IN FILES form_fm1 AND form_fm2 ON ";fd$
lprint tab 60;date(2)
lprint : lprint : search form_line=0
while found(): lprint "Form name : ";upper(form_name$)
lprint " Description : ";form_desc$
lprint "  Form paper : ";form_paper$; tab 30;"Form ink : ";form_ink$
lprint "  Form length : ";form_height; tab 30;"Form Height : ";form_width
if dbf_name$<>"": lprint "  User database : ";dbf_loc$;dbf_name$
lprint "      Database desc : ";dbf_desc$: endif
lprint : lprint : continue : endwhile :liny;23
endproc
proc form_print2
msg;23,"Printing listing of "+sd$+sn$+" field names"
lprint : lprint "FIELDS OF DATABASE ";sd$;sn$;" USEABLE IN FORM ";upper(form$)
lprint tab 60;date(2)
lprint : let i=0
while i<numfld(): lprint i; tab 5;fieldn(i,"s"): let i=i+1: endwhile :liny;23
endproc
proc form_print3
msg;23,"Printing fields on form "+form$
lprint : lprint "FIELDS OF "+sn$+" DEFINED ON FORM - "+upper(form$)
lprint tab 60;date(2)
lprint : lprint : use "f2": locate form$
lprint "Line/Column  Name       Start   Width   Format Join Symbol"
while field_form$=form$ and not eof(): lprint
lprint "[";field_line;",";field_column;"]";
lprint tab 15;field_name$; tab 30;field_start; tab 40;field_width;
lprint tab 50;field_format$; tab 58;field_join; tab 64;"""";field_join$;""""
lprint : next : endwhile :liny;23
endproc
proc openy;d$,l$
msg;23,"open access of "+d$: open d$ logical l$
endproc
proc set_fonts
let pc$=chr(0)+chr(27)
let bolds$=pc$+chr(69): let bolde$=pc$+chr(70)
let conds$=chr(0)+chr(15): let conde$=chr(0)+chr(18)
let itals$=pc$+chr(52): let itale$=pc$+chr(53)
let unders$=pc$+chr(45)+chr(0)+chr(1): let undere$=pc$+chr(45)+chr(0)+chr(0)
let wides$=pc$+chr(87)+chr(0)+chr(1): let widee$=pc$+chr(87)+chr(0)+chr(0)
endproc
```

Next time some new project I haven't yet thought of. Any suggestions? Until then Happy Archiving!

PERTINENT POINTS ON QL DISK DRIVES
by Frank W. Davis

I have recently ran across some items that I thought it best to pass on to the rest of you. I realize that many of you know some, or perhaps all of this, but it is always good to bring everyone else up to the same speed whenever possible.

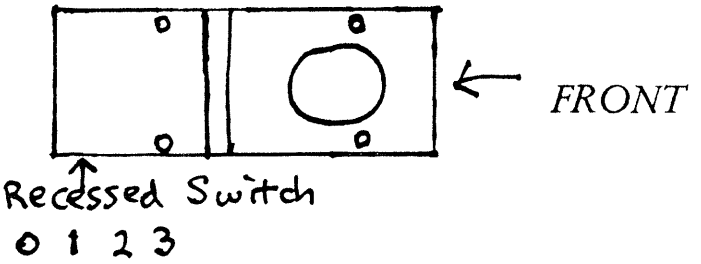
The first came about from the inquiry of two separate people. The question was "Can I use HD (1.44 meg or 2880 sectors) disk drives with my QL that is using a Trump Card?" The answer to this question is a qualified one. Normally the answer would be no. The QL and the Trump Card would not even recognize that it existed. The lone exception to this would be if you had equipped this QL with the Level Two Driver. It would then recognize that the drive was there, and format and use it as a 720K drive (1440 sectors). It would still not see it as a 1.44 meg disk drive; the Trump Card does not have the needed hardware to do this. If you still want to use a 1.44 meg disk drive with either a Trump Card or a QL Sandy Board, then you need to obtain the FLP/LEVEL 2 RAM CHIP from either Jochen Merz or from Mechanical Affinity. This will allow you to use these drives as well as making the QL Read/Write and format IBM and Atari disks, and a few other goodies.

The next question was, "Can I use an IBM compatible 1.2 meg 5 1/4 drive with my QL?" The answer to this would normally have been a definite no. If you check with the magazine IQLR, they ran an article on how to set up one of these so that it could be read by the QL as a 720K (1440 sector) drive, but it could still not be used as a 1.2 meg drive. None of the QL disk drive interfaces supports the 1.2 meg drive in its native mode. I wish that they did, particularly when it comes to IBM software that comes in that format. I could then use it with PC Conqueror or Solution.

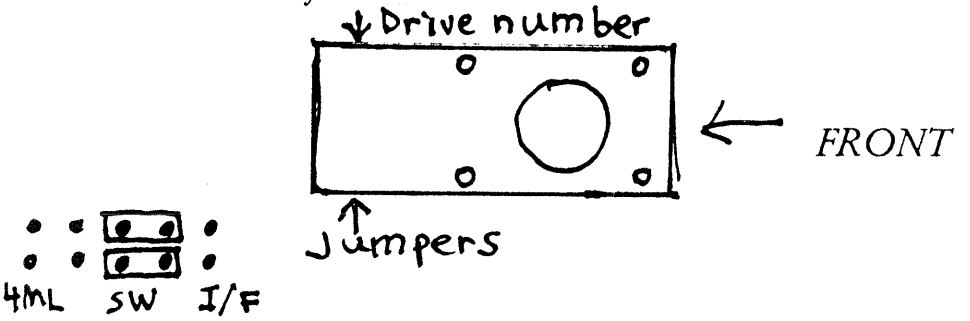
The third question I get from many folks concerns, " Just what ED (extended density 3.2 meg) drives can I use with my Gold Card or Super Gold Card, and how do I set the switches or pins on them?" The next page will detail what is needed to set up TEAC, SONY and MITSUBISHI ED drives. Do not attempt this with CHINON ED drives. So far CHINON does not work with the QL. If you find a drive that works that I left out, let me hear from you. The more we help each other the more we gain.

I wish to thank the following for the information contained here. Stewart Honeyball of Miracle Systems, Paul Holmgren of Mechanical Affinity, John Impellizzeri and Don Walterman of the Detroit area, and Bob Dyl of IQLR. Thanks for the information.

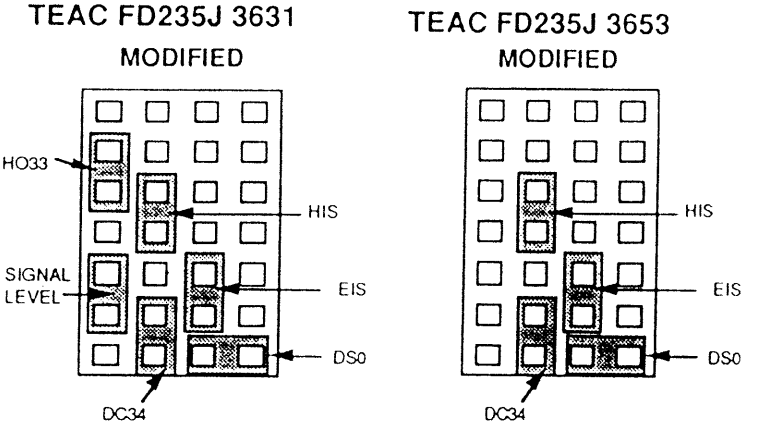
SONY ED DRIVE -- MP-F40W-23 and MFD-40W-21 are the two model numbers that have been shown to work with the Gold Card and the Super Gold Card. The only setting available on these drives is the drive number, a recessed switch at the rear side of the disk drive, as shown in this illustration.



MITSUBISHI ED DISK DRIVE--It has the drive numbers 0 to 3 on one side towards the rear, and has the jumpers that must be set as in the diagram on the other side near the rear of the disk drive.



TEAC ED DISK DRIVES--Set the drive jumpers as indicated in the illustration, and the drive numbers on the side, for the FD235J 3631 and the FD235J 3653.



XTRICATOR, Version 1.75
by Gary Norton

XTRICATOR, by Dr. Carlo Delhez, is a ZX-81 (T/S 1000) emulator for the QL, offering 64 KB of RAM. XTRICATOR runs in the Pointer Environment, is SerMouse compatible, and is multi-tasking.

XTRICATOR is a shareware program, available free for testing. As with all shareware software, if it is to be regularly used, you need to register with the author. The registration fee is 60 Dutch Guilders or 40 US $. With the registration, you receive a twenty-five page manual, three disks of ZX-81 software containing 240 programs (original as well as commercial), a free update to the next version of the emulator, and information about later releases.

To run XTRICATOR, you will need a QL or clone, 300 KB of free RAM, Toolkit II (optional, but recommended), and the Pointer Interface Environment file **PTR_GEN** (not included with the program due to copyright). Though not required, I highly recommend that the Gold Card or Super Gold Card be used.

When run on a standard QL, the program runs painfully slow, but will run. Expect a speed of about a third of a ZX-81 in SLOW mode. Whereas with the Super Gold Card, it runs at over twice the speed of the original ZX-81.

XTRICATOR runs in all resolution modes of QXL, but it does not yet run on machines running SMS2, but that is being worked.

If you're using a QL, then why would you want to emulate the ZX-81? Many of us began our home computing with the ZX-80/1 and discovered the fascinating world of computing. ZX-81 owners discovered much about computers and programming by trying to get every ounce of power from the computer. We all went through the frustrating times of the system crashing when we were about to save a program we just spent two hours typing. But it kept us interested in computing, and most have fond memories of this little machine. When the T/S 2068 or the QL came along, many of us moved on, packing the ZX-81 into a box in the closet. Others remained faithful to the ZX-81 - I have mine set up next to my two QLs.

If you had a ZX-81 that didn't crash, had internal 64 KB RAM (no memory pack wobble), had a (quasi) real keyboard instead of a flat membrane, and had fast disk drives, would you have been as eager to replace your computer when a new model was released? You get all this, and fast speed as well with the Gold Card or Super Gold Card. Later, I'll touch on several additions/enhancements added that will bring the ZX-81 to another level.

The program disk contains a file 'XTricator_Xample_Boot'. Edit this to your boot-up requirements. When XTRICATOR is being executed, you have the option of automatically setting the parameters you wish. The parameters allowed at start-up are: /I gives a black screen and white print; /R selects the initial contents in the 8-16K area, Clear, ZX-Assembler-2, or Coral Basic; /S selects initial size of the RAM memory, and /W selects a half wide screen display.

When loaded, XTRICATOR loads a white screen with the familiar 'K' cursor. No more fuzzy BW television images for the ZX-81! If you would prefer a black screen with white print, simply press CTRL & F2 (or F12 for those with

a Falkenberg QL-Keyboard-90). To revert back to black on white, press CTRL & F2 again (or F12).

With a few exceptions, when running XTRICATOR, use it as you would the ZX-81. You'll recognize the one press Keywords. Press 'R' for RUN, for example. To load a program, press 'J' for LOAD then either SHIFT P, as in the ZX-81 or the Quote key next to the Enter key for the Quotation mark, then the program's name, then another quote: 'LOAD "PROGRAM"'. Do not press 'LOAD ""'. Instead of loading the first program that is contained on disk, as the ZX-81 does with its tape loading, this command gives the directory of the disk in the default drive. The same rules apply for keyboard operation as with the ZX-81. This includes the Function and Graphics modes also. If you have problems remembering the location of the keys or what function is located on which key, press CTRL & F1 or F11 to view a digitized image of the ZX-81 keyboard. This can also be selected by the 'H' option of the main menu. Press ESC or press the left button on the mouse to return to the screen. There are two key emulator modes, using CAPS LOCK or not. These simply give different characters with certain keypresses, basically either the ZX-81 layout or the QL layout.

The feature that really sets XTRICATOR apart from the original ZX-81 is the Main Menu. This is selected by pressing F1. The menu consists of twelve selections. To make a selection, point and click with your mouse or press the letter given and then enter. This article is not intended to make it unnecessary to register and receive the manual, so I will only highlight the menu options. The primary options allow for changing defaults such as LPRINT output (to SER1 or to a file) and default directory. Make screendumps that can be loaded from SuperBasic or from ZX81 mode. Reset the ZX81. Unlike on a ZX81, you don't have to disconnect the power supply to reset. Load ZX-Assembler-2 (ZXA2) or Coral Basic Interpreter (CBI) [more on these later] into the 8-16K area of ROM. Change screen settings such as color (black or white), the refresh rate, set the emulator speed, and select between full size or half size screen width. For QXL users, move the ZX81 window to wherever you'd like on the screen.

XTRICATOR supports several different resolution modes. These are the Low Resolution or normal mode, with a 32x24 screen; Extended Low Resolution, with a 32x192 screen; Semi High Resolution with 256x192 graphics; and True High Resolution with 256x256. The command to move between hi-res and low-res is supposedly CTRL-F3, however, with my copy of XTRICATOR, I was unable to load a hi-res program. I kept getting the message "RAMTOP NOT LOWERED". This in spite of pressing CTRL-F3. Unfortunately, the manual does not mention this or what to do about this situation.

As mentioned earlier, additions and enhancements were provided to bring the ZX81 to another level. Starting with the Special Commands. In Version 1.75 of XTRACATOR, there are fourteen of these commands. They are loaded like a file, but the command name is preceeded with an * and ended with a :. Since the ZX81 operating system does not have a Disk Operating System, the Special Commands, add the necessary commands to operate from disk. They allow a disk directory to be shown as well as movement through the directory. Subdirectories are allowed to be created, removed, and made the default. Files can be deleted, the default drive can be set, and Jobs can be killed. Also, special files can be saved and loaded.

SIMDOS2 is a RAMDISK driver file that adds a few other commands and gives a simplified command structure, such as LB:<filename>: which loads a BASIC

program. No manual is provided, but in the ProgInfo_txt file, there is sufficent information to use this driver.

What truely sets XTRICATOR above the ZX81 (and many other computers as well) is the supplied Coral Basic Interpreter (CBI), written by Carlo Delhez. When running in CBI mode, the first things you notice are the K cursor turns into a solid black (or white) cursor and the command words have to be fully typed in. CBI comes with a 16 page TXT file manual on disk. To quote from the manual, "It combines the efficiency of a new and fast Pseudo Screen Editor with the ease of a powerful extension of the ZX81 BASIC." Regular ZX81 programs will run in the CBI mode. Since CBI warrants an article of its own, I'll just say, give it a try, you may never go back to the ZX81 BASIC.

To those who register, on Library Disk 1 there is another ZX81 BASIC extender, EXTENDED BASIC, by Frits Beniest. This BASIC extender allows you to add READ and DATA statements, adds the RESTORE command, as well as LEFT$, MID$, RIGHT$. Multiple statements may be on the same line. Graphics are brought up to the TS-2068/Spectrum level with DRAW, UNDRAW, CIRCLE, UNCIRCLE, FILL, PAPER, UNPAPER. A few other commands are also provided as well as existing ZX81 commands enhanced. There's also a nifty demo program that lets you see things you never thought you'd see on a ZX81. EXTENDED BASIC gives the ability to easily translate programs written for other BASICS. It nicely adds features not included on the original BASIC, and gives another alternative to CBI, but unfortunately a TXT file manual is not included. With either CBI or EXTENDED BASIC, you'll never want to go back to basic ZX81 BASIC!

Another included file is ZX-Assembler-2, revised by Carlo Delhez. The ProgInfo_txt file gives sufficient information to get the Assembler loaded, but a TXT manual is not provided. One is, however, available upon request. I don't have the manual and am not familiar enough with this program to comment.

One oddity I discovered was when the LPRINT output channel was selected, the mouse stopped functioning. When this was cleared the mouse became operational again. In the next version of XTRICAATOR I would like to see this corrected.

I'm greatly impressed with XTRICATOR. Its a superb piece of programming with a great deal of effort put into it. The price is right, free, but I would strongly recommend paying the small amount asked for to register your copy. With the library disks provided, there is enough programming to keep you occupied for some time. If you have some favorite ZX81 programs you would like to transfer from tape to disk to run on XTRICATOR, Dr. Delhez can provide QZ Fileserver to do the job. There is a disk provided to run on the QL and a tape to run on the ZX81. To do the transfer, you'll need a serial interface. A schematic can also be provided by Dr. Delhez that shouldn't be too difficult for someone to construct if they're knowledgeable in this sort of thing. Unfortunately I'm not, so have been unable to give this a go, but if I can overcome my handicap in this area, I plan on transferring several tapes and will report on the operation of the fileserver.

A REVIEW OF **QLerk** *by Eliad P. Wannum*
A Financial Program for the QL

There have been several finance programs brought out for the QL since the year 1983, but none that I have felt was good enough to use for the middle-class home or for the small business (earning less than $500,000 per year). I felt that way until Frank Davis gave me a review copy of QLerk to look at and try as a means of keeping track of my financial records of my counseling business. I have noticed that even Frank has gone to using it for Mechanical Affinity, and for UPDATE! Magazine. He had been using either an old TS2068 program or one from his Amiga. He had always said that the programs for financial management on the TS2068 and Spectrum were as good as those for the QL, and easier to use. He has now, as well as I, changed his mind. This little story of using this program was done off of Version 3.21, which was released on September 14, 1994.

This program requires you to use either Psion XCHANGE (now in public domain) or ARCHIVE, 2.38 (supplied with the program). Bill Cable, author of the program, says that his preference is using ARCHIVE, as it does not have the same memory limitations as when starting XCHANGE. I tried using XCHANGE and had no problems, but I used only a relatively small database at that time, so perhaps this problem would have shown up later. You should have at least 2- 720K disk drives and a Trump Card to make adequate use of this program. The program really comes into its own when used with a QXL Card, Gold Card, or a Super Gold Card. A hard drive and ED disk drives make the program even that much more useful.For reasons of money, these devices make sense, the less time a business spends at the screen, the more time is spent on the business, and not on the financial house keeping chores of mailing lists and billing.

Just what can you do with this program? Well, you can at the same time keep track of upto 5 check writing accounts from a bank, and Bill says that there is no limit to how many charge accounts you can keep track of. I only have 7 credit cards and found it quite easy to use the program to track these and my two checking and one savings account. I know at all times who is in my database of clients and have at a glance information on billing them for services. When I want to do a mailing to clients, suppliers and vendors I do business with, the print facilities allow me to do this in a generally business accepted manner. It works great with my HP Deskjet printer. I have not tried it with my Canon Bubblejet yet, as it is off for repair.

The statements I have generated with this program and given to my accountant have impressed her, as she swore they had to have come from an expensive MAC or IBM program. I had to have her over to the office to see just what control I actually had over records and the ease with which it was done. I just may get a QL put to use in her office yet! She tells me I should , at tax time be able to give her all the records I need to prepare both my personal and business taxes. She gave me only one barrier to stand in the way of doing this. THE ONLY THING THAT CAN GO WRONG IN THIS AREA IS FOR THE USER TO FAIL TO KEEP UP WITH ENTERING THE DATA. It is just like vitamins or medicine, it does you little good to have them, if you do not use them when you should.

The program needs to be configured for your system when you get it. The well supplied manual tells you what to do in order to set up the program for your particular uses. You can always amend your set up or add items, such as new names or addresses (as well as changes), on the fly. I have had two manuals for this program, the first was 8 1/2 by 11 inches in size, the next was 6 1/4 by 8 1/4 inches. While the print is clear and of a readable font in the new version, some of the print was rather small and may be hard on the eyes of some.

One limitation to keep in mind, is that due to this being an ARCHIVE based program you will have a 2,000 record limitation if it is ordered on two fields. There will come a time that you may need to keep more than one record, or you can use the QLerk process REMASTER to remove a portion of old records. This is also still a DATABASE type program, so for those who have never had the nerve to approach one of these, you will find it needed on at least a limited basis.

If I was a movie reviewer I would give this program a thumbs up. I hope in the next issue of UPDATE! to tell you more about this useful, USA made program. Check the ad in this issue.

REPORT ON THE DAYTON COMPUTERFEST IN AUGUST

by Frank W. Davis, Editor

For the 6th year in a row now, we at Mechanical Affinity, and for the 5th year for UPDATE! Magazine....found our way to the Hara Arena in Dayton, Ohio on the last weekend in August. It always requires far more work to get to the show and set up than it may appear, what with banners, catalogs, pricing items, etc. So, do I think it is worth it? Yes.

It is shows like this and the Miracle in Newport show that give us our best chance to get together with a large crowd of QL, Timex and Cambridge Z88 users each year. It helps put a face to go with the voice and letters that we get from so many subscribers and mail order customers. I have become personal friends with many of you, and am grateful for having done so. All of the people I have met have in some way or another enriched my life. Thanks. More than once I have read where some "so-called expert" has talked and said that computer users were people who got wrapped up in their machines and lost social skills. I have met only a few people who I think fall into that category. For most of us, it has been a learning experience that has caused us to meet and interact with many people we would otherwise never have met or talked to.

Thanks to Gary Ganger and to all of the support that Dayton Microcomputer Association gives for the production of this large scale show that they put on each year and for setting aside part of it for Sinclair dealers. Socially and financially it was rewarding. Thanks to the guys from SMUG, Neal and Bill for being there each year with your tables next to ours. Thanks to Don Lambert and Bob Swoger for having your table on the other side of us for TSNUG. And thanks for the last two years to Tim Swenson for putting on such a great picnic Saturday evening, giving all of us time to meet and socialize away from the show.

Shows like these are fun, and a chance to buy from a large selection at bargain prices. It brings out the "old Hoosier horse trader" in me. I hope to see you there next year.

PRESENTING QLerk

A FINANCIAL PROGRAM FOR THE SINCLAIR QL

By
Wood and Wind Computing : Bill Cable : RR3 Box 92 : Cornish NH 03745
Phone : (603) 675-2218

For the first time you have the capability of keeping complete and accurate financial records for the Home or Small Business with your QL. A friendly Financial Clerk to serve you. The code is written in the ARCHIVE Programming Language and is completely accessible to the user. All functions are selectable from standardized menus. No knowledge of ARCHIVE is required. The program works from a common sense point of view without imposing accounting theory on the user. Although it has many powerful features the user can use only those features desired, ignoring the rest. Recommended minimum system is a Trump Card with 2 DD Drives. It works much faster on Gold Cards and Super Gold Cards. Latest Version is 3.21

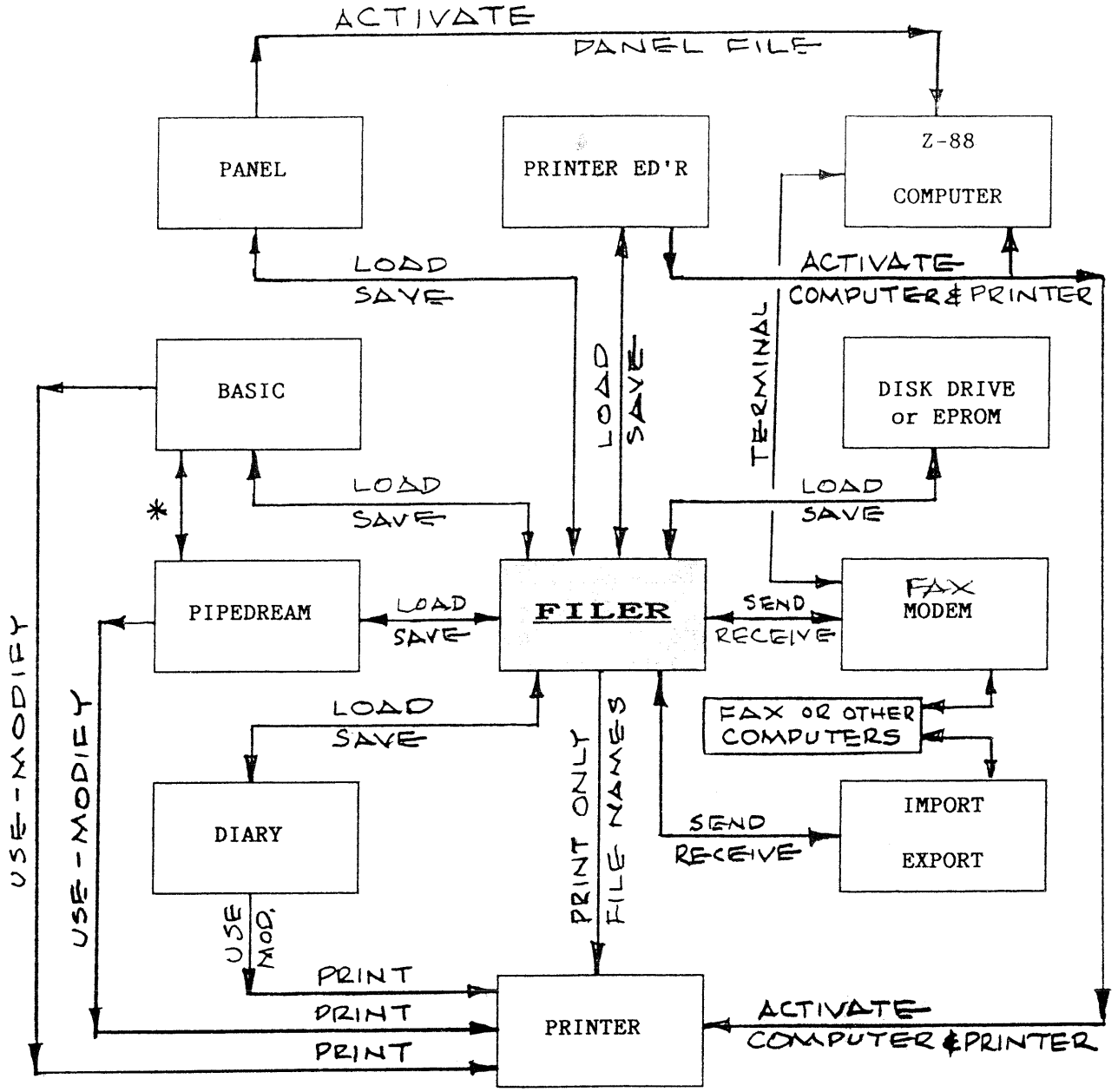
Pays Bills	*Receives Income*	*Reconciles Bank Accounts*
Makes Invoices	*Makes Purchase Orders*	*Prints Checks*
Prints Address Labels	*Handles Sales Taxes*	*Handles Income Taxes*
Does Payroll	*Keeps Inventory*	*Handles Periodic Payment*
Periodic (Cyclic) Payments	*Periodic (Cyclic) Income*	*Maintains Savings Accounts*
Maintains Cash Accounts	*Maintains Charge Accounts*	*On-line Help*
Elaborate Data Protection	*Easy Built-in Data Backups*	*Easy Data Correction*
Category Report (why,when)	*Activity Report (who,when)*	*and much more*

PRICING

Public Domain Demonstration Version of QLerk (refundable if QLerk latter ordered)	$5.00 US/Canada $7.00 Elsewhere
QLerk Program on Disk with Tutorial Doc File	$29.00 US/Canada $31.00 Elsewhere
QLerk Manual (150 pages of details)	$29.00 US/Canada $34.00 Elsewhere
QLerk Program with Tutorial and QLerk Manual	$50.00 US/Canada $57.00 Elsewhere

The Demonstration version will allow you to play with most of QLerk's features so you can decide if you really want the program. Some features are absent and the code is not inspectable. The program with Tutorial is the complete program with inspectable code and enough instructions to try out the basic features of QLerk. It is sufficient for users with simple needs. The QLerk Manual is indexed and covers all features in detail for those with more complex needs or with an interest to know all the details. **Order it today increase the usefulness of your QL dramatically.**

Z-88 FLO-CHART (C) 1989 MIKE FINK



ANY DATA, TYPED INTO PIPEDREAM, DIARY OR PROGRAMS IN BASIC, WILL AUTOMATICALLY BE TEMPORARILY SAVED IN SUSPENDED ACTIVITIES, WHEN GOING FROM ONE APPLICATION TO ANOTHER. TO SAVE A FILE PERMANENTLY, IT MUST HAVE A NAME AND BE SAVED TO THE FILER. DATA TYPED INTO THE PANEL CAN BE ACTIVATED IMMEDIATELY, OR SAVED AS A FILE, OR BOTH.

PRINTER EDITOR MODIFICATION MUST BE ACTIVATED TO BE USED, OR CAN EXIST UNACTIVATED AS FILES TO BE LOADED OR SAVED (OR ACTIVATED).

FILES FROM PIPEDREAM, DIARY AND BASIC CAN BE SENT TO A PRINTER FOR PRINTING, EVEN IF THEY HAVE NOT BEEN NAMED AND SAVED. FILES FROM THE FILER TO THE THE DISK DRIVE, CAN ONLY BE SAVED OR LOADED, TO USE OR EXECUTE THEM, THEY MUST FIRST BE LOADED INTO WHERE THEY CAME FROM.

FOR THOSE PEOPLE THAT DESIRE A PORTABLE COMPUTER, OR NOTEBOOK COMPUTER, THAT IS LIGHT, INEXPENSIVE .. AND DOES NOT REQUIRE THE USER TO BUY ANY SOFTWARE, THE Z88 EXISTS FOR THEM. LUCKILY, IN THE UNITED STATES, THERE IS A COMPANY CALLED DOMINO CUBES, THAT HAS CREATED ADDITIONAL FEATURES FOR THE Z88, THAT MAKE THE POWER OF THIS COMPUTER MORE FORMIDABLE THAN IT ALREADY IS, AND ARE NOT AVAILABLE AT ANY OTHER PLACE IN THE WORLD !!

A PARTIAL LIST FOLLOWS;

1. A 9600 BAUD FAX/MODEM, THE ONLY FAX/MODEM IN THE WORLD THAT NEEDS NO SOFTWARE! IT WEIGHS 7 OUNCES AND IS THE SIZE OF A PACK OF CIGARETTES.
2. A BATTERY OPERATED 2 LB. PORTABLE DISK DRIVE.
3. A BAR-CODE READER.
4. A 1 LB. "BATTERY PAK" THAT WILL RUN THE Z FOR 3 MONTHS.
5. REPAIR SERVICE.
6. UNDERSTANDABLE MANUALS WRITTEN BY MIKE FINK
7. THE ONLY MAIL MERGE PROGRAM THAT WORKS!
8. A SHIRT POCKET EPROM ERASER.
9. TWO TYPES OF EDITING PROGRAMS WHILE IN BASIC.
   a. A THIRD TYPE OF EDITING FOR BASIC USING PIPEDREAM.
10. A PRIME NO. PROGRAM UNIQUE TO THE WORLD.
11. A MONEY TABLES PROGRAM OFFERING ANSWERS TO ALL QUESTIONS INVOLVING TIME, MONEY AND INTEREST.
12. MANUALS EXPLAINING FORMATTING AND PRINTER EDITOR WAYS NOT FOUND IN ANY OTHER MANUAL OR PUBLICATION.
13. A SPELLING CHECKER, (FROM ENGLAND).
14. THE ABILITY TO MAKE MULTIPLE COPIES AND HOW TO USE MACROS.
    a. AN INTERNAL MAGICAL REMOTE CONTROL ROBOT!*!*!*!
15. A PROCEDURE TO ALLOW LOADING ALL EPROM FILES WITHOUT DANGEROUS CLI FILES BEING CREATED IN :RAM.- (WHICH CAUSE FAILURE)

TO RECEIVE ALL OUR LITERATURE AND A PRICE LIST, SEND $3 IN STAMPS OR CASH (no checks, please) THE $3 WILL BE RETURNED AT YOUR FIRST ORDER.

MAIL ADDRESS IS: 130 W. 42nd ST. 28th FLOOR, N.Y. CITY 10036
FAXES CAN BE SENT TO 212-869-1526

MIKE FINK PRESIDENT OF DOMINO CUBES

Z88 Rom and Internal Ram Upgrade

by Dave Bennett

The Z88 Users' Club in England is no longer viable. Membership had fallen to 1000 members from 4000 members. Roy Woodward decided to discontinue the Group. He is still providing support for the Z88 in a business called Woodward Technology.

Roy is selling back issues of the Club Magazine 'Z88 Eprom'. The Club also sold Z88 parts. Roy is selling some of these including Version 4.0 Roms and larger size internal Rams.

I had ordered a Version 4.0 Rom and a 512K Internal Ram last year. I waited months and months for them to arrive. I talked to a couple Z88 users in the U.S. who told me it was best to call Roy. I finally called him in April.

Apparently he could not get 512K Rams. This was holding up the order. I said that I was willing to accept a 128K Ram. The 128K Ram is also cheaper and easier to install. The Ram and Rom arrived a couple weeks later. It was another couple weeks before I installed them.

Installing the Ram and some Roms on a small number of Z88s requires soldering and desoldering. So I would not attempt this yourself unless you are comfortable soldering circuit boards. And you need the proper equipment. A low wattage soldering iron and a solder sucker are essential. If you are not comfortable doing this I would have a computer repair facility do it for you.

If your Z88 is working fine and you have no need for additional internal Ram, you may not want to install the Ram and Rom at all. The Rom provides support for additional internal Ram. It also corrects a few bugs such as the famous :Ram.- bug. The software still works the same. Pipedream still is exactly the same.

The instructions for installing the Ram and Rom are very clear and well written. It is very easy to remove the circuit board. It only takes a few minutes.

The circuit board on the Z88 is sensitive to heat. When I desoldered the Ram one of the lands came off the board. I fixed it with a jumper cable made out of a very thin wire strand. When I was finished I put the Z88 partially together. I tried it and there was no display. In addition it was making a slight humming sound. It should not make a sound at all!

I thought that my Z88 was gone. I let it sit around for a few weeks without touching it. Finally I decided to look at it again.

I was going to remove the socket for the Ram and install another one. But decided to resolder the old socket. I put it together and it worked! Everything works on it. However the instructions say that you should have 118016 free Ram. I get 115446. I talked to another Z88 user who performed the installation. He got the same result. Puzzling! Is it an error in the instructions or a typo. Or did we both make the same mistake. I am not going to worry about it. The Z88 is working fine otherwise. 115K is a lot better than the 12K I had before.

I did not experience any crashes with the Version 2.2 Rom. But this upgrade gave me a chance to increase the amount of Ram without using up any more slots. All three of my slots are filled.

The Rom was 24.95 and the Internal DIY 128K Ram was 19.95. Both of these upgrades make a fine addition to your Z88. They are both available from Woodward Technology, P.O. Box 15, Belper, Derbyshire, U.K. DE56 OXE. If you pay by credit card the exchange rate is automatically taken care of.

Z88 This and That

As far as I know the Z88 is still being manufactured in Scotland. I wonder just how many Z88s have been sold? What is your Serial Number? I have an early Z88. Mine is 2996.

My Z88 tends to give off a lot of Radio Frequency Interference or RFI. I have noticed that other Z88s give off much less. Is it because I have an early Z88? My Z88 has been opened a few times. Could some seals have been disturbed causing RF leakage? Does your Z88 cause interference to nearby TVs and Radios?

Timex Publication Index part 1

Many moons ago I had to prioritize the many ideas I had for my computing projects that were filling my spare time.

I also had a LOT of Timex/Sinclair publications with tons of neat articles that I was interested in doing with my Timex/Sinclair machines.

Since I had a cartridge version of Pro/File for the Timex 2068, (room for up to 37,000 bytes of database records) I indexed all the software listings and hardware articles contained in this 4 foot pile of paper. This allowed me to quickly find things to do and to answer questions (if answers were printed) for my Timex projects. The results ended up occupying 2 files totaling more than 39,000 bytes and filling about 317 record entries.

File Record Explanation:

I only listed articles that had program listings or real hardware work that I could do. Reviews and simular articles were left out because there was nothing there that would directly benefit my computing at the time of need.

I chose to index 7 publications because they contained the most information, and I had an average of 90% of the total issues they printed.

So here you are, if you can use this information then this has done its job.

Key to the listings

The listings were done with the Title or at least the reason for inclusion in the index first, followed by a 1 or 2 word clue that told me if the article was a program that was type-in-able (software) or about a (hardware) modification/fix or improvement. I then used a 3 letter key identifing the publication containing the article, followed by the issue or volume and lastly some clue as to why I listed it if the title was not informative enough.

The Key used for the Publications:

SUM =	**Vols**	**2-6 to 4-7**
SWN = SYNCWARE NEWS		**1-1 to 5-4**
SYN = SYNC	**Vols**	**1-1 to 4-2**
TD = TIME DESIGNS		**1-3 to 4-4**
TMZ = TIMELINEZ	**pages**	**1-8, 17-358, 364-415, 443-458**
TSH = T-S HORIZONS		**#1 to #21**
TSU = TIMEX SINCLAIR USER USA version	**Vols**	**1-1 to 1-7**

1000 TIPS hard

SYN	2-1	ZX80 RESET, INVERSE VIDEO
"	3-5	ROM, RAM
TSH	6	NVM DISABLE
TSU	1-4	REPLACE CHARACTER SET

1000 & ZX80 TIPS soft

SYN	1-2	logic
"	1-3	fix for above, CHR$ on more logic, simulate PAUSE, convert variables
"	1-4	fix for above (logic),
"	"	VARS conversions, DIMed arrays
"	1-5	REM usage, fix for DIMed arrays, PEEK & POKE
"	1-6	MC in REM, more logic using Galaxy game
"	2-1	Flags, PEEK/POKE fix
"	2-2	SLOW, PLOT, INKEY$, ZX80 plotting,
"	2-3	store 3 words in array
"	2-4	flag tips, DEF function
"	2-5	DEF func fix
"	2-6	" speed up
"	3-1	" " " , line inputting
"	3-5	logic operators
"	3-6	" " fix
"	4-2	array storage

1000 TIPS soft

SWN	1-1	EDIT usage
"	1-2	memory saving
"	1-3	" "
"	1-5	set ramtop, fix for load circuits
"	2-1	trap inputting nos.
"	2-3	ramtop & Compusa
"	3-3	string input rout. (Get)
"	3-5	fix for above
TD	1-5	set ramtop
"	2-3	Strings & things
"	3-6	simulate PAUSE
TMZ	60	BASIC tips
"	67	" "
"	72	" "
"	73	cracking AUTO-RUN
"	80	BASIC tips
"	96	" "
"	106	" "
"	120	Vu-Calc mods & tips
"	125	BASIC tips
"	155	POKEs, PEEKs
TSH	#2	INKEY$,
"	#10	password, hide prog lines, SAVE/LOAD time, print last 2 lines, COPY full screen, find print coordinates
"	#12	unLIST lines, unlock tapes, MC storage, clock unused codes
"	#16	program stopper, Vu-Calc mods
"	#20	Files & ?
TSU	1-5	mem save, lock up, vary COPY size
"	1-6	PEEK & POKE

6 SHOOTER soft game

SYN	2-2	8k

16 PIN BOWLING soft game

TSU	1-2	1000
"	1-3	" fix

64 COLUMN MODE soft

SWN	3-1	2068
TD	2-3	"

2040 PRINTER hard soft

SWN	2-6	make work with MemoTech 16K
"	4-1	Remove caps
SYN	4-2	& ROM bugs
TD	3-1	printer instead of display
"	4-2	long cable
TMZ	66	off/on
"	73	fix for above
"	234	2068 off/on
"	256	on/off switch

2050 hard

TD	4-1	Spectrumize

2068 GAZER'S GUIDE soft

TD	2-1	astronomy

2068 TIPS soft

Sum	3-11	SAVE, SCROLL, LOAD INKEY$, INPUT
"	4-4	OmniCalc 2
SWN	2-6	INPUT prompts
"	3-1	" " extra
"	3-4	print using MC
"	3-6	fix for above
SYN	4-2	
TD	2-5	Poly-Scroll
"	3-6	simulate PAUSE
TMZ	61	inputting progs tips cracking progs
"	63	POKEs, BEEP, SCROLL, CAPS, DELAYS, SCREEN
"	66	cracking, BIN
"	73	store MC, Passwords
"	107	" "
"	135	all kinds

"	144	INKEY$, VARS
"	172	timer, VARS
"	234	screen save, POKEs, invert display, print full screen
"	234	POKE BORDER color, ramtop without CLEAR, time, POKEs, odd/even, darken display, dbl. space LISTings
"	235	store nos. as characters, time
"	311	INK tips
TSH	#7	Displays, 22 & 23 line Scroll, POKEs, Flags, BEEP
"	#11	Screen LPRINT
"	#12	OPEN#, CLOSE#
"	#17	many POKEs
"	#18	fix for POKEs #17, pixel screen scroll
TSU	1-6	display primer

ABC-123 soft game

SWN	4-3	1000 game
"	4-4	" " pt 2
"	4-5	bug fixes

ACEY DUCEY soft game

SYN	1-1	ZX80

ADDITION soft

SWN	5-5	2068 math tutor

ADVENTURES IN THE RAM JUNGLE soft

TD	1-6	explore RAM structure
"	2-1	corrections, pt 2
"	2-2	1000 based, pt 3
"	2-3	fix's for 2-2

ADVENTURE soft game

SYN	1-3	ZX80

ALIEN LURE soft game

TSU	1-2	1000

ALIEN TREASURE soft game

SYN	2-4	
"	2-5	fix

ANIMALSLLL soft game

SWN	4-6	animal quiz

ARTILLERY soft game

SYN	1-2	ZX80
"	1-3	fix for above
"	1-4	" " " , (2)
"	1-6	another

AUDIO FREQ COUNTER soft

SWN	3-1	1000
"	3-3	" bug fix

AUDISY soft

SYN	3-6	store sound in 1000

AUTO ANALYSIS soft

SWN	3-4	1000

A & J ENHANCEMENT soft

Sum	3-4	Tasword
"	3-1	Tasword, VU-Calc
SWN	3-5	notes
TD	3-6	file manager
TMZ	30	1000
"	114	Tasword &
"	129	2068 cat prog
"	144	Tasword
"	170	CPI tips
TSH	#14	SAVEing LOADing
"	#16	" "
"	#18	tips

BANK SWITCHING 1000 hard soft

TSH	#4	pt 1
"	#5	" 2
"	#6	" 3 pt 1 bug
"	#7	" 4
"	#9	" 5
"	#10	" 6
"	#11	" 7
"	#12	" 8 last
"	#16	bug fix pt 7

BANK SWITCHING 2068 hard soft

SWN	2-3	add 128K & decode EXROM
"	3-4	OUTs explained
"	3-6	using extra banks
TD	2-5	pt 1
"	2-6	" 2
"	3-1	" 3
"	3-2	" 4
"	3-3	" 5, end
"	4-5	
TSH	#19	pt 1

" #20 pt 2
" #21 pt 3 & 4

BANNER soft
Sum 4-5 both
SWN 2-4 1000
SYN 2-6 large letters on screen
TMZ 244 2068 mini/maxi banners
TSH #10 1000
" #11 2068 bug fix
TSU 1-3 2068

BASCII soft
TMZ 251 convert BASIC to ASCII file
" 257 fix for above
TD 3-1 " " "

BASIC soft
Sum 3-1 Beginning BASIC pt 1
" 3-2 " " " 2
" 3-3 " " " 3
" 3-4 " " " 4
TSU 1-2 How to Program pt 2
" 1-3 " " " " 3
" 1-4 " " " " 4
" 1-5 " " " " 5 last

BATTLESHIP soft game
SYN 2-1 ZX80

BEETHOVEN soft music
TSU 1-1 music from the 1000

BETA BASIC soft
SWN 5-6 tips & utils

BINOMIAL BINGO soft game
TMZ 225 1000/2068

BINOMIAL DISTRIBUTION soft game
TSU 1-4 1000

BIT GRAPHICS soft
SWN 4-6 for big printers

BIT TESTING soft
SWN 2-3 depict 255 bit patterns 2068
TMZ 104 1000/2068

BLABBERMAN soft
TSU 1-6 1000

BLACK HOLE soft game
SYN 1-3 ZX80
" 1-4 fix for above, (3)
" 1-5 " " "

BLINKER soft
SWN 5-3 blinker code 2068

BLOCK DELETE soft
SYN 3-2 1000
TD 4-1 Spectrum

BOULE soft game
SYN 4-2

BRICK BUSTER soft game
SYN 3-5
" 3-6 fix for above
" 4-1 mods
" 4-2 fix

BUILDING HEAT LOAD soft
SYN 2-6

BURGLAR ALARM hard soft
TD 2-3 2068
" 2-4 " fix's

BUS EXPANSION soft
Sum 2-1 Spectrum to 2068
SWN 3-5 2068 prog to draw bus for cart board

CALCULATING THE DAYS soft
SYN 3-1

CALENDAR soft
SYN 3-1 figure the days
TMZ 86 Perpetual 1
" 95 " 2
" 98 " "

CANNONADE soft game
SYN 1-5 ZX80

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 UPDATE ISSUE DISKS

These disks contain at least one MAJOR piece of software written specifically for disk drives and are guaranteed to be worth the money. The rest of the disk space is filled with various utility programs, or support files either for the major piece of software or from various issues of UPDATE. Some files are taken from public domain or shareware if deemed of sufficient use. Half of the money goes to the contributor of the issue disk on a bi-annual basis. The rest goes to UPDATE to support the issue disk program. All are $20 US$, except where noted. Add $5 US$ extra for shipping outside of North America. All known QL disk formats are supported; please tell us which you need.

1) HARTUNG UTILITY ISSUE DISK- Here are some excellent programs, such as a stand alone database, Address and QSO files. All are written in SuperBasic. This gives lots of programming hints and tricks for QL programmers. This has been recently updated and improved by Bob Hartung. The Address File can be used as an Inventory program, or use it to print out labels. Both paper or screen printout can be Alpha sorted or by last name. $15.

2) CABLE ARCHIVE ISSUE DISK- Written by Bill Cable. Contains many useful ARCHIVE programs that work on any Archive database. 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 dupes within a database), WINDEX (word index any text file), GROUP 1 to 3 (useful procedures from UPDATE articles), QUERY (interrogate any database). Also includes extensive DOC files about the programs and ARCHIVE in general. The price is $20.

3) QLUSTER 5s109 ISSUE DISK- A great program from Al Feng to provide you with many utilities to handle & unclutter your disks & MDVs (and it now supports sub-directories). Some of the features concern COPY, DELETE, FORMAT, VIEW, as well as extended use of some of the TK2 commands (TK2 needed for this program).The program is TURBO compiled for a speedy program. It is MINERVA compatible, multi-tasks and allows you to use minimal keypresses to do the job. The price is $15.

4) QLuMSI DOS 4.30 ISSUE DISK- The latest version of Al Fengs extensively updated MSDOS simulator and front end program for the QL. Other programs on the disk enhance file management and cloning of other programs. Educational and useful. The price is $20.

5) QLAMBer- Al Fengs latest issue disk. He calls it A-Moving-Box/enhancedrelease. This greatly extends the selective file management capabilities of the QLUTter program by additionally accessing six TK2 keywords, while reducing CODE size, easily supports sub-directory access, and easily multi-tasks within QRAM or Taskmaster. TK2 must be on ROM or loaded prior to start up of program. The price is $15.