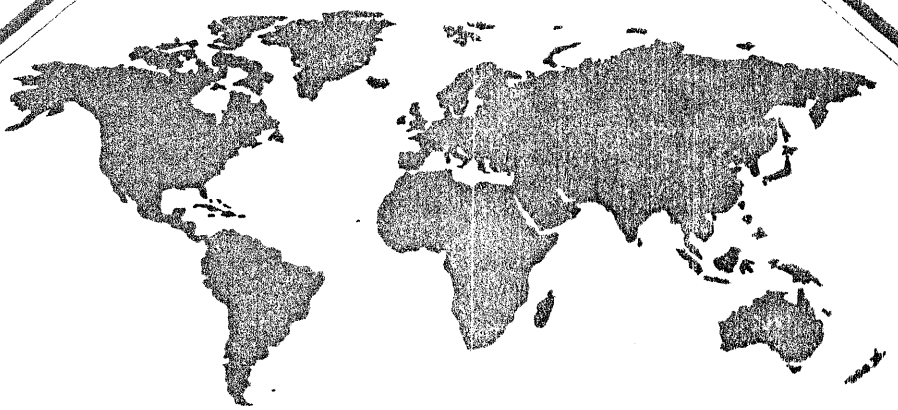
UPDATE!



SINCLAIR

JULY *1995*

P. O. Box 17, Mexico, Indiana 46958 USA
$5.00/Issue $20.00/Year

UPDATE COMPUTER SYSTEMS is edited by Frank Davis and published by Carol Davis of P.O. Box 17, Mexico, IN 46958 USA. The phone number is 317-473-8031 for both voice and fax. Normal hours for voice are 6 to 9 P.M. Tuesday thru Saturday. Please use the answering machine if we are not available, and leave a short, concise message with both phone number and address. The hours for faxing are from 11 P.M. to 6 A.M. Monday thru Sunday, and at other times if you let us know that we have a fax coming. Most answers to questions left on the machine will be by mail, as long distance charges are too costly for a small magazine to be expected to bear.

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 $20 in US$; and 18 Pounds, or 40 DM in European currency, for a subscription. All issue years run from October to July of the next year, and those who subscribe during a year will get all of the issues for that year. In this way all subscriptions begin and end at the same time. Timely renewals are what keep us in business!

Assistance in publishing this magazine is provided by you the readers, many of whom have contributed news, articles and reviews. We offer you our heartfelt thanks. We are mainly assisted by our longtime friend, Eliad Wannum, Poet, Psychologist, and Sinclair computer user. In addition, regular contributors have been: Bill Cable, Paul Holmgren, Al Feng, Don Lambert, Bob Swoger, Abed Kahale, Peter Hale, A. E. Green, to name just a few. We invite you to submit material for publication to UPDATE MAGAZINE. Please make all hard copy (printed out) submissions Letter or NLQ; no draft print, or 2040 thermal print. The quality is too poor to print in a readable manner, and we do not have the spare time to re-type an article. If you do not have a printer capable of this, then send the article on disk to us as: Z88, IBM, QL, TS2068 in Oliger or Larken, or in an Amiga file. We can handle these disk formats. We can use 5.25 or 3.5 disks, in DSDD, HD or ED densities. Send two copies of hard copy. Do not submit stuff on audio tape, as we no longer have tape decks for the TS1000 or the TS2068. Try to avoid flowery or hard to read fonts, unless you are showing us sample output from a printer reviewed or a program. If artwork for an article is to be included, please let us know in what order you think it should be displayed. We can accept articles over the fax.

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 both of us exchanging copies of the issues the ads are placed in. TWO, 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 issue free. For two issues you pay full rate for one ad, and get $5 off of the cost of the second ad rate. Should you have any questions on this please contact Frank Davis either by phone or mail, as listed above. All checks should be made out to UPDATE MAGAZINE.

We hope to be of service to you. Thank you for your support.

TABLE OF CONTENTS for JULY 1995

The front cover for this edition was done courtesy of Abed Kahale of CATUG. Thanks, Abed! This is a quarterly, user supported magazine. The inside page covers subscriptions, ad and article submission to UPDATE. We are about to begin the 9th year of publishing, and look forward to your renewal of your subscription. Perhaps you will even sign up a friend to make us even stronger. If you sign up two paid, new subscribers to UPDATE you will get your subscription at half price!

Front Covers - design and basic magazine information.

Page 1 -- Directory
Page 2 -- Editorial by Frank Davis
Page 3 -- Reference Card for the ZX80, ZX81 & TS1000 (Part 3)
Page 4 -- Adding Two More Drives (A Quick Saturday Morning Task) by Bob Swoger
Page 6 -- RMG ad - Rod Gowen
Page 7 -- EFF's Guide to the Internet V.3.1, Part 1
Page 19 -- Answer to Puzzler (TS2068), from January 1995 issue by Tim Swenson
Page 20 -- Program for serial Logimouse on Z80 by Richard Rodman, modified by William A. DesLauriers
Page 23 -- ad by Mechanical Affinity
Page 24 -- Archive Series Part 21: Bycol - A Program for Displaying and Printing Databases by Columns - Cable Column by Bill Cable of Wood & Wind Computing
Page 30 -- QBOX USA - ad
Page 30 -- Computer Classics - ad (Sinclair computer repairs)
Page 31 -- QLuMSi 4.80 by Al Feng, a tutorial
Page 36 -- Report on Oak Ridge QL Show
Page 38 -- Banter - A Review of QL Banter Program by E. P. Wannum
Page 39 -- Z88BAS.TXT (Internet download of Z88 Basic Commands) by Chris Fenn
Page 42 -- Z88 File Exchange with Other Computers by E. P. Wannum
Page 43 -- Z88 External Battery Box by Tim Swenson
Page 45 -- Timex Publications Index, part 4 by Paul Holmgren
Page 50 -- Z88 ad - Mechanical Affinity

Back Covers -- Information on Issue Disks for the QL & TS2068, UPDATE user generated software.

JULY 1995 EDITORIAL by Frank W. Davis, Editor

Here we are again and it is once more time to....ASK ALL OF YOU TO RENEW YOUR SUBSCRIPTIONS (all except for one new subscriber, an Italian Dr.). Other than that one person, everyone is now due to renew their subscriptions. It is at times like this that I feel just like the people doing the telethons for public TV, trying to get people to once again pledge their support for their favorite public television shows, and telling people that without this support they will lose these shows. That is kind of the case with all small magazines. As we do not have a lot of wealthy advertisers, we depend almost wholly on your support by subscribing to stay in business. Unlike BYTE magazine with over a million subscribers, we can not charge $24,000 for a one page ad. Please give us your honest support by taking the time to renew now, perhaps even before you go on to finish this editorial.

This coming year will be the last year we will be publishing UPDATE Magazine. This will be its 9th year of publication, and 6 years for the team of Carol and Frank Davis, with the help of Eliad Wannum. We would like to do this one more year as our best. With your financial and article support perhaps we will reach this goal. Before you go on, please take the time out to do this now, before it gets put aside on a back burner to be forgotten till it is too late. Inside of this envelope you should find a renewal form, please note the prices for renewal. To those in the know I don't have to tell you that the cost of paper has risen over 200% in just the last year, as well as the prices of postage. I am glad that I only have to make this appeal once a year. I never liked to listen to this sort of thing myself when I was just a reader, but the viewpoint has changed. Thanks.

The next thing to remind all of you of is : THE DAYTON COMPUTERFEST, perhaps the largest computer show in the USA , put on by a local club, is at the end of August. The set up for vendors is Friday, August 25, 1995. The show will run on August 26th and the 27th. We will once again be in the small Sinclair section of the show, selling products from UPDATE Magazine and from Mechanical Affinity. All of us look forward to once again seeing all of those who regularly come to the show. It goes without saying that all the rest of you, and anyone else interested is encouraged to attend the show. We will be there to serve you.

From what I have heard from Tim Swenson, there will once again be a pitch in picnic the evening of Saturday the 26th, after the show closes down. It will once again be held at Tims house. Maps should be available at the show. I know of no one who has not enjoyed these chances to chow down and share conversation about their computers with each other for a few hours. Last year we even had some musical entertainment as a surprise by Doug LaVerne. He also provided this at the Oak Ridge show.

This will be the very last issue that we will be offering ISSUE DISKS. If I can find the time to make some up, they will be available at the show in Dayton, but no later than that. For the small amount of money they bring in I can put the space to better use with information about Sinclair and Cambridge computers.

A couple of last notes before I allow you to get on with the rest of the magazine. One, the Masterpiece graphics card for the QL is not at this time ready for release. The problems with the Super Hermes appear to have been worked out and you can go ahead and place your orders for them with Mechanical Affinity. The Qubide Hard Drive Interface has made it through the testing period and drive incompatibilities have been worked out (get the interface before purchasing a drive as there is a short list of drives that will not work with it). They are in stock at Mechanical Affinity. For those with an interest in an updated QL operating system we have SMSQ/E available for sale. Last,we'll shortly have the Amiga QL emulator for sale.

REFERENCE CARD FOR THE

ZX80®, ZX81®

AND

TIMEX SINCLAIR 1000™

HOW TO USE *FOR . . . NEXT*

The *FOR* statement sets up conditions for executing a series of BASIC lines over and over again until the ending conditions are met.

A SAMPLE OF USAGE:

```
10 FOR A = 0 TO 30
20 Z = Z + Y
30 R = R + Z
40 NEXT A
50 PRINT M
```

When you enter: FOR A = 0 TO 30
You are really saying:
- Assign the value 0 to A.
- Execute code beginning at line 20.
- When you get to the "NEXT A" statement, add 1 to A.
- Then, if A is greater than 30, go to the next line (50).
- Otherwise, go back to line 20.

YOU CAN ENTER THE STATEMENT MANY WAYS:

```
FOR C = A TO B
FOR X = 1 TO W
FOR P = L TO 10
```

HOW TO USE *FOR . . . STEP . . . NEXT*

USAGE IS THE SAME AS **FOR . . . NEXT** ABOVE
EXCEPT **FOR A = 0 TO 30 STEP 2**
- WHEN YOU GET TO THE "NEXT A" STATEMENT
- 2 IS ADDED TO A.
- OTHER FORMS CAN STEP IN A NEGATIVE INCREMENT.
- FOR EXAMPLE:

```
FOR A = 30 TO 0 STEP -2
```

HOW TO USE MORE THAN 1 *FOR* AT A TIME

WHEN YOU USE MORE THAN ONE **FOR** AT A TIME, IT IS CALLED "NESTING."
WHEN NESTING, EACH **FOR** HAS ITS OWN **NEXT**.
THE LAST **FOR** ENTERED MUST FIND ITS OWN **NEXT** FIRST, OR YOU HAVE BAD PROCESSING.

```
FOR A=1 TO 4
FOR B =1 TO 7
FOR C=5 TO 10
NEXT C
NEXT B
NEXT A
```

What happens here:
- The **A** loop is entered. It loops 4 times.
- The **B** loop is called 4 times by the **A** loop.
- Each time the **A** loop calls it, the **B** loop goes 7 times.
- The **C** loop is called 7 times by the **B** loop.
- Each time the **B** loop calls it, the **C** loop goes 6 times.
- So **A** loops 4 times, **B** loops 28 times, **C** loops 168 times.

HOW TO USE *IF . . . THEN*

The **IF** statement allows you to compare items against each other and **THEN** take an action based upon the results of the compare.
You can test for equal, less than, greater than, or any combination of the three.
You can also combine tests for a complex compare.
You can also combine IFs by making the next one the action.
This is called "NESTING."

Example of a simple **IF** statement:

```
IF A = B THEN LET C = D
IF A NOT = B THEN GOTO 1000
IF A > B THEN PRINT L$;
IF A < B THEN STOP
IF A<>B THEN GOSUB 2000
```

Example of a complex **IF** statement:

```
IF A = B AND C = D THEN GOSUB 3000        (Both required)
IF A = B OR C = D THEN GOTO 4000          (Only one required)
IF A = B AND C = D OR E = F THEN PRINT M$;
        (First two, or last one required)
```

When using **AND**, then every test must be true to take the action.
When using **OR**, then only one of the tests must be true.
Of course, if there are **ANDs**, then they must all be true before the **OR** makes its decision.
When none of the tests fits the condition for the **THEN**, control of your program falls through to the next line.

Example of combined **IFs**:

```
IF A = B THEN IF C = D THEN IF E = F THEN GOTO 5000
```

This is easier to see if written as follows:

```
IF A = B THEN
  IF C = D THEN
    IF E = F THEN
      GOTO 5000
```

At each level, when the condition is not true, then the next **IF** is not tested. Instead, all of this code is skipped, and the computer goes to the next line.
When "NESTING" **IFs**, it is the same as using **AND** without the nesting.
Another example:

```
IF A = B THEN IF C NOT = D OR E > F THEN PRINT R$;
```

If **A** equals **B** then we ask the next **IF** question, otherwise we go to the next line.
In the next **IF** question, if **C** is not equal to **D** then we will print.
If **C** is equal to **D** then we ask if **E** is greater than **F**.
If **E** is greater than **F**, then we will print.
Otherwise, we will go to the next line.

ZX80 MEMORY MAP

ADDRESS		DESCRIPTION
DECIMAL	HEX	
16384	4000H	**ERROR-CODE MINUS ONE.**
16385	4001H	*BASIC SYSTEM CONTROL FLAG BITS
16386	4002H	CURRENT BASIC STATEMENT NUMBER
16388	4004H	**ADDRESS OF K OR L CURSOR.**
16390	4006H	**BASIC STATEMENT NUMBER AT) CURSOR.**
16392	4008H	*ADDRESS OF PROGRAM VARIABLES
16394	400AH	*ADDRESS OF WORKING STORAGE (KEY INPUT)
16396	400CH	***ADDRESS OF UPPER SCREEN.**
16398	400EH	***ADDRESS OF LOWER SCREEN.**
16400	4010H	***ADDRESS OF END-OF-SCREEN.**
16402	4012H	***NUMBER OF LOWER-SCREEN LINES.**
16403	4013H	NUMBER OF FIRST BASIC STMT ON SCREEN
16405	4015H	**ADDRESS OF § MARK MINUS ONE.**
16407	4017H	**NUMBER OF STMT TO "CONTINUE" AT.**
16409	4019H	SYNTAX FLAG BITS
16410	401AH	SYNTAX TABLE POINTER
16412	401CH	**RANDOM NUMBER SEED.**
16414	401EH	SCREEN FRAME DISPLAY COUNT
16416	4020H	ADDRESS OF FIRST CHAR OF FIRST VAR NAME IN LAST DIM, FOR, INPUT, LET, NEXT
16418	4022H	VALUE OF LAST VAR OR EXPRESSION
16420	4024H	***LINE POS OF NEXT SCREEN CHAR:** FROM 33 (LEFT) TO 2 (RIGHT). 1 = FIRST COL, NEXT LINE (LINE FULL). 0 = FIRST COL, (E-O-LINE.)
16421	4025H	***CURRENT SCREEN LINE (0=BOT, 23=TOP)**
16422	4026H	*ADDRESS OF CHAR AFTER PEEK OR POKE STMT.
16424	4028H	**USER PROGRAM AREA.**

* = DO NOT POKE. UNPREDICTABLE RESULTS.

ERROR CODES

ERROR CODES APPEAR AS: xx/yy
WHERE: xx is the error code,
AND yy is the number of the last statement executed.

CODE	MEANING
0	Successful execution, or, GOTO-line too big
1	NEXT has invalid variable, but, variable is assigned
2	Variable not assigned, or, DIMensioned
3	Bad subscript
4	Memory exhausted
5	Screen full.
6	Arithmetic number too large.
7	RETURN before GOSUB
8	INPUT attempted in command mode. Illegal.
9	STOP was executed.
ZX81 ONLY	
A	Invalid parameter.
B	Invalid integer.
C	Invalid data in VAL string.
D	BREAK was pressed.
E	Unused.
F	SAVE name is a null string. Illegal.

A QUICK Saturday Morning Task

ADDING TWO MORE DRIVES

by
Bob Swoger

Saturday at our house is a day we set aside to accomplish a multitude of chores in a days worth of time. I try to warm up by knocking out one task before noon, then accomplish the others after lunch. Did you ever start one of those little Saturday morning chores? Here's a **'for instance'** so you may understand what I'm talking about.

Phil Kwitkowski called one Friday night and asked to do a little file transferring from my BBS. "Sure, no problem". After he downloaded some files, it was my turn to get some files from him. With his MaxCom disk in drive ZERO and a now full disk in drive ONE, he said "Darn! I wish I had more drives!" "Well that should be no problem", I said. "I'll be over in the morning"

Although Phil had acquired an Apollo disk drive case for his first two drives, we thought he'd never find another case just like it for drives 2 and 3. But miraculously, a friend of mine had one sitting around and offered it to Phil. These cases include a power supply and three disk drive power connectors but room for only two half height drives. They really look neat and professional in the TS2068 hutch!

Phil had just been to two computer fests and picked up three half height DSDD drives for $5 each. He already had ribbon cable and a quick trip to the local Radio Shack store should give us enough connectors to complete the job. Hey! A lot of great new and close out items around the old Radio Shack store! Well, that trip cost us an hour!!

Abed Kahale had made the first disk drive ribbon cable for Phil almost a year ago. The cable connected the two drives together and then ran out the back of the Apollo case to the LarKen interface board.

Now, if this was to be a quick Saturday morning project, we would leave the first cable pretty much alone and just add one 34 pin board edge connector to the middle of it. We would build a cable for the second Apollo box identical to the first, but without the middle connector, so that the second box could be plugged in and out of the setup with ease. And now the elegant part. We would fashion a printed circuit board out of an old scrap PC board that contained about fifty .1 inch spaced traces, 25 on each side, running the length of the board.

Phil had a band saw and cut a small piece of the board so that 17 traces were left on each side with a width just wide enough to slip into the two 34 pin board edge connectors. For the rest of this writing we shall call this board a cable interconnect board.

We added a Radio Shack 34 pin board edge connector to the middle of the cable of the first box and slipped one end of the little cable interconnect board into it. We then plugged the cable from the second disk drive box into the other end of the cable interconnect board. Get it? Now the two cables are in parallel!

But when we powered up, the lights on all four drive's came on! Hey - no problem! The cable in the first box must have been built with the red tracer of the ribbon cable on pin 34 instead of pin one, right? We will just power down and reverse one of the connectors on the cable interconnect board.

We powered up again. No problem! All the lights are off. At LogiCall's drive prompt we entered '0'. Drive 0 turned and we got a CATalog but at the top it said CRC ERROR 000! We entered '1' at the Program? prompt. Drive 1 turned and another CATalog with CRC ERROR 000! Drive 2 and 3 didn't even turn! We felt the panic begin but we remained analytical. We disconnected the second drive.

We asked for CATalogs again. Both drive still gave us the same CRC ERRORs! Also, drive 1 was now making a grinding sound. We changed the disks in drives 0 and 1 for fresh ones. All was fine now! We had corrupted two disks!

Next we unplugged the first set of drives from the LarKen interface board and plugged the second set of drives. They worked fine! Hey! I thought there was a cable in this system that was made with the red tracer on pin 34 - not so if both drives work independently! The little board we made must be the culprit. Maybe the traces on the gold plated end are too close together and short out the connector. I had seen this in the past. Improperly aligned connectors in the LarKen system caused pins in those connectors not to connect to the board traces. Let's make another cable interconnect board from a section of the board where the traces remain .1 inch the whole length of the board. Also we will be sure that it is wide enough to remain tight at the ends of the connector to insure proper alignment. It took two tries!

Phil fashioned the board and began stripping the solder resist from the foil so that the connectors would make contact with the traces. I looked into the two connectors, side-by-side, that his masterpiece was about to connect together - this time with the perfection of accuracy. ARGH! Wait a minute! This isn't going to work! I screwed up! Pin 1 on one cable is going to connect to pin 2 of the second cable - everything is reversed! That's why all the drives came on the first time!

Brain still working, I grabbed a short six inch long cable from the junk box with 34 pin board edge connectors on both ends and plugged it onto the cable interconnect board Phil had just completed. I put the first cable interconnect board we thought was bad into the connector on the other end. I then plugged the two cable interconnect boards into the connectors of both drives. We powered up on the RAMDISK with LogiCall - at the Drive prompt 0 - 1 - 2 - 3 - 4 - all five drives answered up with CATalogs! Voile - it all worked! See FIGURE 1 below.

Phil's mom invited us upstairs for a dinner of broccoli soup. It was delicious! Dinner!!!? Yes - Phil and I just had just come to realize that we had blown away another perfectly good Saturday morning! [Yea, and a Sunday morning writing this article!] But because we spent the extra time to get it right, you won't have to!!! Keep Timexin'.....

!!!! THE FINISHED PRODUCT !!!!

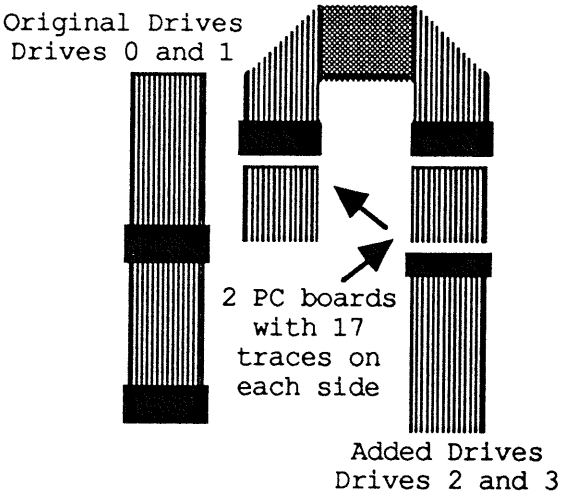
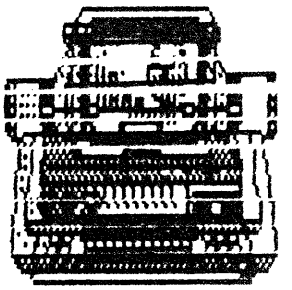


FIGURE 1

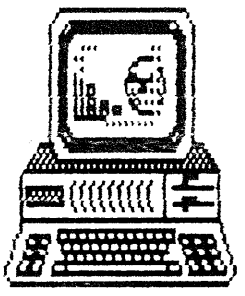
RMG ENTERPRISES



Supports Sinclair/Timex Users!

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

Mail to:



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

(Editors Note: This should be of help to those looking to get on the Internet for the first time. This is the first in a three part series, as a tutorial.)

EFF's Guide to the Internet, v. 3.1

(formerly The Big Dummy's Guide to the Internet)

Adam Gaffi ***February 16, 1995***

Copyright 1993, 1995 Electronic Frontier Foundation, all rights reserved.

DISTRIBUTION:

This guide is available free of charge from the EFF online archives at ftp.eff.org, gopher.eff.org, http://www.eff.org/, EFF BBS (+1 202 861 1223), AOL keyword EFF, CIS EFFSIG forum, and elsewhere.

This guide may be freely reproduced & distributed electronically or in hardcopy, provided the following conditions are met:

1) Please do not qualitatively modify the guide, and leave all copyright, distribution, attribution, and EFF information intact. Permission expressly granted for translation to other languages and conversion to other formats.

2) EFF has signed a contract with MIT Press to publish a hardcopy book version of the guide, entitled _Everybody's_Guide_to_the_Internet_. All other for-profit distribution of printed versions of the guide is forbidden, unless permitted in writing. Non-English hardcopy editions may appear in the future. However, you may print out copies and recoup the cost of printing and distribution by charging a nominal fee.

3) Any for-profit non-paper distribution (such as shareware vendor diskettes, CD-ROM collections, etc.) must be approved by the Electronic Frontier Foundation (the time- and usage-based access fees of online services, bulletin boards, and network access providers are specifically exempted). Donations appreciated.

4) You may not charge a for-profit fee specifically for this guide, only for a collection containing the guide, with the execeptions of nominal copying charges and online usage fees as exempted above).

The Electronic Frontier Foundation
1667 K St. NW, Suite 801
Washington DC 20006-1605 USA
+1 202 861 7700 (voice)
+1 202 861 1258 (fax)
+1 202 861 1223 (BBS - 16.8k ZyXEL)
+1 202 861 1224 (BBS - 14.4k V.32bis)

Internet: ask@eff.org
Internet fax gate: remote-printer.EFF@8.5.2.1.1.6.8.2.0.2.1.tpc.int
FidoNet: 1:109/1108

TABLE OF CONTENTS

Foreword by Mitchell Kapor, co-founder, Electronic Frontier Foundation.

Preface by Adam Gaffin, senior writer, Network World.

Chapter 1: Setting up and jacking in
- 1.1 Ready, set...
- 1.2 Go!
- 1.3 Public-access Internet providers
- 1.4 North American providers
- 1.5 European and other providers
- 1.6 If your town has no direct access
- 1.7 Net origins
- 1.8 How it works
- 1.9 When things go wrong
- 1.10 FYI

Chapter 2: E-mail
- 2.1. The basics
- 2.2 Elm -- a better way
- 2.3 Pine -- even better than Elm
- 2.4 Smileys
- 2.5 Sending e-mail to other networks
- 2.6 Seven Unix commands you can't live without
- 2.7 When things go wrong

Chapter 3: Usenet I
- 3.1 The global watering hole
- 3.2 Navigating Usenet with nn
- 3.3 nn commands
- 3.4 Using rn
- 3.5 rn commands
- 3.6 Essential newsgroups
- 3.7 Speaking up
- 3.8 Cross-posting

Chapter 4: Usenet II
- 4.1 Flame, blather and spew
- 4.2 Killfiles: The cure for what ails you
- 4.3 Downloading messages
- 4.4 ASCII and ye shall receive
- 4.5 Some Usenet hints

4.6 The brain-tumor boy, the modem tax and the chain letter
4.7 Big Sig
4.8 The First Amendment as local ordinance
4.9 Usenet history
4.10 When things go wrong
4.11 FYI

4.1 Flame, blather and spew
4.2 Killfiles, the cure for what ails you
4.3 Some Usenet hints
4.4 The Brain-Tumor Boy, the modem tax and the chain letter
4.5 Big Sig
4.6 The First Amendment as local ordinance
4.7 Usenet history
4.8 When things go wrong
4.9 FYI

Chapter 5: Mailing lists and Bitnet
5.1 Internet mailing lists
5.2 Bitnet

Chapter 6: Telnet
6.1 Mining the Net
6.2 Library catalogs
6.3 Some interesting telnet sites
6.4 Telnet bulletin-board systems
6.5 Putting the finger on someone
6.6 Finding someone on the Net
6.7 When things go wrong
6.8 FYI

Chapter 7: FTP
7.1 Tons of files
7.2 Your friend archie
7.3 Getting the files
7.4 Odd letters -- decoding file endings
7.5 The keyboard cabal
7.6 Some interesting ftp sites
7.7 ncftp -- now you tell me!
7.8 Project Gutenberg -- electronic books
7.9 When things go wrong
7.10 FYI

Chapter 8: Gophers, WAISs and the World-Wide Web
8.1 Gophers
8.2 Burrowing deeper
8.3 Gopher commands

8.4 Some interesting gophers
8.5 Wide-Area Information Servers
8.6 When things go wrong
8.7 FYI

Chapter 9: The World-Wide Web
9.1 Getting snared in the Web
9.2 Alright, already, let's go!
9.3 Addressing a problem
9.4 Imagine that -- downloading pictures
9.5 Lynx, meet Gopher
9.6 Finding things on the Web
9.7 SLIP: But I want to use my mouse!
9.8 More on SLIP
9.9 HTML: Building your own
9.10 Some interesting WWW servers
9.11 Lynx commands
9.12 When things go wrong
9.13 FYI

Chapter 10: Advanced E-mail
10.1 The file's in the mail
10.2 Receiving files
10.3 Sending files to non-Internet sites
10.4 Getting ftp files via e-mail
10.5 Downloading e-mail messages
10.6 Mining for info on Usenet via e-mail
10.7 Just the fax, ma'am
10.8 The all knowing Oracle
10.9 When things go wrong
10.10 FYI

Chapter 11: News of the world
11.1 Clarinet: Associated Press, Miss Manners and Dilbert
11.2 Still more news on the Net
11.3 The world today, FROM Belarus to Brazil
11.4 FYI

Chapter 12: IRC, MUDs and other things that are more fun than they sound
12.1 Talk
12.2 Internet Relay Chat
12.3 IRC commands
12.4 IRC in times of crisis
12.5 MUDs
12.6 Go, go, go (and chess, too)!
12.7 The other side of the coin
12.8 FYI

Chapter 13: Education and the Net
13.1 The Net in the Classroom
13.2 Some specific resources for students and teachers
13.3 Usenet and Bitnet in the classroom

Chapter 14: Business on the Net
14.1 Setting up shop
14.2 Online storefronts
14.3 The check is in the (e)-mail
14.4 The good, the bad and the ugly
14.5 FYI

Chapter 15: The end?

Appendix A: Lingo

Appendix B: Electronic Frontier Foundation Information

Foreword
By Mitchell Kapor,
Co-founder, Electronic Frontier Foundation.

Welcome to the World of the Internet.

The Electronic Frontier Foundation (EFF) is proud to have sponsored the production of EFF's Guide to the Internet. EFF is a nonprofit organization based in Washington, D.C., dedicated to ensuring that everyone has access to the newly emerging communications technologies vital to active participation in the events of our world. As more and more information is available online, new doors open up for those who have access to that information. Unfortunately, unless access is broadly encouraged, individuals can be disenfranchised and doors can close, as well. EFF's Guide to the Internet was written to help open some doors to the vast amounts of information available on the world's largest network, the Internet.

The spark for EFF's Guide to the Internet, and its companion MIT Press book version, Everybody's Guide to the Internet, was ignited in a few informal conversations that included myself and Steve Cisler of Apple Computer, Inc., in June of 1991. With the support of Apple Computer, EFF engaged Adam Gaffin to write the book and actually took on the project in September of 1991. The guide was originally published electronically in July of 1993 as the Big Dummy's Guide to the Internet.
The idea was to write a guide to the Internet for people who had little or no experience with network communications. We intended to post this guide to the net in ASCII and other formats and to give it away on disk,

as well as have a print edition available. We have more than realized our goal. Individuals from as geographically far away as Germany, Italy, Canada, South Africa, Japan, Scotland, Norway and Antarctica have all sent electronic mail to say that they downloaded the guide. EFF's Guide to the Internet is now available in a wide array of formats, including ASCII text, Windows Help, World-Wide Web, PostScript, and AmigaGuide. And the guide is finally available in a printed format, as well.

You can find the electronic version of Everybody's Guide to the Internet by using anonymous ftp to ftp.eff.org and accessing /pub/Net_info/Guidebooks/EFF_Net_Guide/netguide.eff. Or send an e-mail message to info@eff.org. The electronic version is updated on a regular basis with a newletter (/pub/Net_info/Guidebooks/EFF_Net_Guide/Updates/), and occasional new versions of the guide itself.

We invite you to join with EFF in our fight to ensure that equal access to the networks and free speech are protected in newly emerging technologies. We are a membership organization, and through donations like yours, we can continue to sponsor important projects to make communications easier. Information about the Electronic Frontier Foundation and some of the work that we do can be found at the end of this book.

We hope that EFF's Guide to the Internet helps you learn about whole new worlds, where new friends and experiences are sure to be yours. Enjoy!

Mitchell Kapor
Chairman of the Board
Electronic Frontier Foundation
mkapor@eff.org

QUERIES: Please send all queries regarding EFF, and availability and distribution of the guide to ask@eff.org, or one of the other EFF addresses above. Please send all updates, corrections, and queries regarding the content of the guide to the author, Adam Gaffin, at adamg@world.std.com.

Preface

By Adam Gaffin,
Senior Writer, Network World, Framingham, Mass.

Welcome to the Internet! You're about to start a journey through a unique land without frontiers, a place that is everywhere at once -- even though it exists physically only as a series of electrical impulses. You'll be joining a growing community of millions of people around the world who use this global resource on a daily basis.

With this book, you will be able to use the Internet to:

= Stay in touch with friends, relatives and colleagues around the world, at a fraction of the cost of phone calls or even air

mail.

= Discuss everything from archaeology to zoology with people in several different languages.

= Tap into thousands of information databases and libraries worldwide.

= Retrieve any of thousands of documents, journals, books and computer programs.

= Stay up to date with wire-service news and sports and with official weather reports.

= Play live, "real time" games with dozens of other people at once.

Connecting to "the Net" today, takes something of a sense of adventure, a willingness to learn and an ability to take a deep breath every once in awhile. Visiting the Net today is a lot like journeying to a foreign country. There are so many things to see and do, but everything at first will seem so, well, foreign.

When you first arrive, you won't be able to read the street signs. You'll get lost. If you're unlucky, you may even run into some locals who'd just as soon you went back to where you came from. If this weren't enough, the entire country is constantly under construction; every day, it seems like there's something new for you to figure out.

Fortunately, most of the locals are actually friendly. In fact, the Net actually has a rich tradition of helping out visitors and newcomers. Until very recently, there were few written guides for ordinary people, and the Net grew largely through an "oral" tradition in which the old-timers helped the newcomers.

So when you connect, don't be afraid to ask for help. You'll be surprised at how many people will lend a hand!

If you have any suggestions or comments on how to make this guide better, I'd love to hear them. You can reach me via e-mail at adamg@world.std.com. Boston, Mass., February, 1995.

Chapter 1: SETTING UP AND JACKING IN

1.1 READY, SET ...

The world is just a phone call away. With a computer and modem, you'll be able to connect to the Internet, the world's largest computer network (and if you're lucky, you won't even need the modem; many colleges and companies now give their students or employees direct access to the

Internet).

The phone line can be your existing voice line -- just remember that if you have any extensions, you (and everybody else in the house or office) won't be able to use them for voice calls while you are connected to the Net.

A modem is a sort of translator between computers and the phone system. It's needed because computers and the phone system process and transmit data, or information, in two different, and incompatible ways. Computers "talk" digitally; that is, they store and process information as a series of discrete numbers. The phone network relies on analog signals, which on an oscilloscope would look like a series of waves. When your computer is ready to transmit data to another computer over a phone line, your modem converts the computer numbers into these waves (which sound like a lot of screeching) -- it "modulates" them. In turn, when information waves come into your modem, it converts them into numbers your computer can process, by "demodulating" them.

Increasingly, computers come with modems already installed. If yours didn't, you'll have to decide what speed modem to get. Modem speeds are judged in bits per second or "bps." One bps means the modem can transfer roughly one bit per second; the greater the bps rate, the more quickly a modem can send and receive information. A letter or character is made up of eight bits. You can now buy a 14,400-bps modem for under $100 -- and most now come with the ability to handle fax messages as well. For under $300, you can buy a modem that can transfer data at 28,800 bps (and often even faster, using special compression techniques). If you think you might be using the Net to transfer large numbers of files, a faster modem is always worth the price. It will dramatically reduce the amount of time your modem or computer is tied up transferring files and, if you are paying for Net access by the hour, will save you quite a bit in online charges.

Like the computer to which it attaches, a modem is useless without software to tell it how to work. Most modems today come with easy-to-install software that will let you connect not only to the Internet but other services, such as bulletin-board systems and many commercial online networks, such as CompuServe. Try the program out. If you find it difficult to use or understand, consider a trip to the local software store to find a better program. You can spend several hundred dollars on a communications program, but unless you have very specialized needs, this will be a waste of money, as there are a host of excellent programs available for around $100 or less. Among the basic features you want to look for are a choice of different "protocols" (more on them in a bit) for transferring files to and from the Net and the ability to write "script" or "command" files that let you automate such steps as logging

into a host system.

When you buy a modem and the software, ask the dealer how to install and use them. Try out the software if you can. If the dealer can't help you, find another dealer. You'll not only save yourself a lot of frustration, you'll also have practiced the prime Internet directive: "Ask. People Know."

In addition to the software that comes with your modem, you can now also buy special software kits for getting onto the internet -- at least, if you use Windows or a Macintosh. Some of these work only with a particular Internet provider; others let you connect to the provider of your choice. There are both advantages and disadvantages to these packages; we'll look at these programs in a little more detail in Chapter 9. This guide is geared more toward folks using older style software, such as Procomm, Crosstalk and Zterm, but even if you use one of the newer all-in-one internet access kits, many of the things described in coming chapters will generally be good to know (you never know when you'll find yourself stuck in a room with only an MS-DOS computer).

To take full Take advantage of the Net, you must spend a few minutes going over the manuals or documentation that comes with your software. There are a few things you should pay special attention to: uploading and downloading; screen capturing (sometimes called "screen dumping"); logging; how to change protocols; and terminal emulation. It is also essential to know how to convert a file created with your word processing program into "ASCII" or "text" format, which will let you share your thoughts with others across the Net.

Uploading is the process of sending a file from your computer to a system on the Net. Downloading is retrieving a file from somewhere on the Net to your computer. In general, things in cyberspace go "up" to the Net and come "down" to you.

Chances are your software will come with a choice of several METHODS to use for these transfers. These different methods are systems designed to ensure that line noise or static does not cause errors that could ruin whatever information you are trying to transfer. Essentially, when using a protocol, you are transferring a file in a series of pieces. After each piece is sent or received, your computer and the Net system compare it. If the two pieces don't match exactly, they transfer it again, until they agree that the information they both have is identical. If, after several tries, the information just doesn't make it across, you'll either get an error message or your screen will freeze (at which point you get out the computer manual to see how to re-boot). In that case, try it again. If, after five tries, you are still stymied, something is wrong with a) the file; b) the telephone line; c) the system you're connected

to; or d) your own computer.

From time to time, you will likely see messages on the Net that you want to save for later viewing -- a recipe, a particularly witty remark, something you want to write your congressman about, whatever. This is where screen capturing and logging come in.

When you tell your communications software to capture a screen, it opens a file in your computer (usually in the same directory or folder used by the software) and "dumps" an image of whatever happens to be on your screen at the time.

Logging works a bit differently. When you issue a logging command, you tell the software to open a file (again, usually in the same directory or folder as used by the software) and then give it a name. Then, until you turn off the logging command, everything that scrolls on your screen is copied into that file, sort of like recording on videotape. This is useful for capturing long documents that scroll for several pages -- using screen capture, you would have to repeat the same command for each new screen.

Terminal emulation is a way for your computer to mimic, or emulate, the way other computers put information on the screen and accept commands from a keyboard. In general, most systems on the Net use a system called VT100. Fortunately, almost all communications programs now on the market support this system as well -- make sure yours does.

You'll also have to know about protocols. There are several different ways for computers to transmit characters. Fortunately, there are only two protocols that you're likely to run across: 8-1-N (which stands for "8 bits, 1 stop bit, no parity" -- yikes!) and 7-1-E (7 bits, 1 stop bit, even parity). The latter is fairly rare these days, except on some older systems, such as CompuServe (which, as of this writing, does not offer full Internet access anyway). What if you don't know what kind of system you're connecting to? Try one of the settings. If you get what looks like gobbledygook when you connect, you may need the other setting. If so, you can either change the setting while connected, and then hit enter, or hang up and try again with the other setting. It's also possible your modem and the modem at the other end can't agree on the right bps rate. If changing the protocols doesn't work, try using another bps rate (but no faster than the one listed for your modem). Don't worry, remember, you can't break anything! If something looks wrong, it probably is wrong. Change your settings and try again. Nothing is learned without trial, error and effort.

There are the basics. Now on to the Net!

1.2 GO!

Once, only people who studied or worked at an institution directly tied to the Net could connect to the world. Today, though, an ever-growing number of "public-access" systems provide access for everybody. These systems can now be found in most metropolitan areas (at least in North America) several states, and there are several companies a couple of sites that can provide access across the country.

There are two basic kinds of these host systems. One provides what is known as a dial-up account. You'll need a basic communications program (the kind that typically comes with your modem) to connect. Once on, your computer's brain essentially goes to sleep -- virtually all of your interaction with the Internet will be via programs on your host system.

The other is known as a SLIP or PPP provider (SLIP and PPP are two types of communications standards for connecting to the Internet). This sort of host acts mainly as a gateway to your own computer. it passes information (e-mail, say, or graphics) to your computer, which then has to figure out what to do with it. This is where the all-in-one programs mentioned earlier come in (and we'll talk a bit more about them in chapter 9).

Some sites are run by for-profit companies; others by non-profit organizations. Some of these public-access, or host, systems, are free of charge. Others charge a monthly or yearly fee for unlimited access. And a few charge by the hour. Systems that charge for access will usually let you sign up online with a credit card. Some also let you set up a billing system if you'd rather pay by check.

But cost should be only one consideration in choosing a host system, especially if you live in an area with more than one provider. Most systems let you look around before you sign up. What is the range of each of their services? How easy is each to use? What kind of support or help can you get from the system administrators?

The last two questions are particularly important because many systems provide no user interface at all; when you connect, you are dumped right into the Unix operating system. If you're already familiar with Unix, or you want to learn how to use it, these systems offer phenomenal power -- in addition to Net access, most also let you tap into the power of Unix to do everything from compiling your own programs to playing online games.

But if you don't want to have to learn Unix, there are other public-access systems that work through menus (just like the ones in restaurants; you are shown a list of choices and then you make your selection of what you want), or which provide a "user interface" that is

easier to figure out than the ever cryptic Unix (and fortunately, even on unix systems, there are ways to minimize your exposure to its commands, as we'll see in later chapters). Some systems also have their own unique local services, ranging from extensive conferences to large file libraries.

1.3 PUBLIC-ACCESS INTERNET PROVIDERS

When you have your communications program dial one of these host systems, one of two things will happen when you connect. You'll either see a lot of gibberish on your screen, or you'll be asked to log in. If you see gibberish, chances are you have to change your software's parameters (to 8-1-N or 7-1-E as the case may be). Hang up (see your modem's manual for information on this), make the change and then dial in again.

When you've connected, chances are you'll see something like this:

```
Welcome to THE WORLD
Public Access UNIX for the '90s
Login as 'new' if you do not have an account
login:
```

That last line is a prompt asking you to do something. Since this is your first call, type

```
new
```

and hit enter (at least on this particular system; each system has a different log-in procedure for newcomers) Often, when you're asked to type something by a host system, you'll be told what to type in quotation marks (for example, 'new'). Don't include the quotation marks. Repeat: Don't include the quotation marks.

what you see next depends on the system, but will generally consist of information about its costs and services (you might want to turn on your communication software's logging function, to save this information). You'll likely be asked if you want to establish an account now or just look around the system.

You'll also likely be asked for your "user name." This is not your full name, but a one-word name you want to use while online. It can be any combination of letters or numbers, all in lower case. Many people use their first initial and last name (for example, "jdoe"); their first name and the first letter of their last name (for example, "johnd"); or their initials ("jxd"). Others use a nickname. You might want to think about this for a second, because this user name will become part of your electronic-mail address (see chapter 2 for more on that). The main exception are the various Free-Net systems, all of which assign you a user name consisting of an arbitrary sequence of letters and numbers.

18 March 1995

Frank,

In regards the the programming puzzle put forth by Basil (no last name) in the last issue of UPDATE, I decided to take on the problem and see what I could make of it.

For reference here is the short program he had:

```
 100 LET page=1
 110 FOR f=353 TO 375
 150 LET page=page+.1
 160 LET panel=page-INT(page)
 165 LET panel=.1*INT(10*panel)
 250 PRINT page, panel
1000 NEXT f
```

The problem he was getting was skips in the value of Panel. Panel should be equal to what's to the right of the decimal point of Page. If Page equals 2.3 then Panel should equal .3. With the above program (running on the 2068), the problem was this:

```
2.2      0.2
2.3      0.2
2.4      0.3
```

Not having my 2068 functioning, I decided to try the program on the QL and see if the problem was specfic to the 2068. The results I got were different. Panel always equaled .1 less than it should (page = 2.2, panel = .1). But, I did not get the skip that the original program did. I did notice that when Page equaled 3.1, Panel equaled 9.9999E-2.

Without doing any more research, I think I can guess the general nature of the problem. If I remember correctly from my computer courses, computers have a problem with .1. It's the way it's converted to binary and back to decimal. If the program was to use integers and then divide by 10 to get the final result, there should no be a problem. Let page = 22, print out (page/10) and let panel = (page MOD 10)/10. Or the print routine can be played with to put in the decimal place manually.

Well, I hope this helps to solve the problem.

Happy Hacking,

Tim Swenson

```
/* --- MOUSE.C --- Read serial Logimouse on arbitrary hardware
   Implementation given is for Software Toolworks C.
   Hardware port logic is for Z80 SIO and CTC.

   By Richard Rodman.  Any use whatsoever of this code is heartily
   encouraged.
   Modifed by William A. Des Lauriers.  Z80 Mnemonic Codes now used instead
   of 8080 Mnemonic Codes.  Typo Errors Checked.  Dated 4/11/95.
Usage:
   If a command line parameter is used, it will simply display
   the bytes received in hex.  Otherwise, the cursor will track
   movement of the mouse, and the button status of each button will
   be displayed.

   Press esc to stop.

Mouse data packet structure:
   First byte: 10000LMR L = 0 if left button down, else 1
                        M = 0 if middle button, else 1
                        R = 0 if right button, else 1
   Second byte:   delta x, negative = left, positive right
   Third byte: delta y, negative = down, positive = up
   Fourth byte:   Another delta x value
   Fifth byte: Another delta y value

   The entire packet is sent if anything changes.

History:
   870706 rr orig version  */

#include "tprintf.c"   /* for debugging only  */

int cursx, cursy;    /* cursor location */

main( argc, argv )
int argc;
char *argv[];
{
   int i, byte[ 5 ] ;
   char butstr[ 4 ] ;

   minit();         /* init serial port for mouse  */

   clrscn();        /* clear the terminal screen   */

   butstr[ 3 ] =  '\0'; /* terminate string for display */

   cursx = 40;
   cursy = 12;    /* center the cursor  */
   goxy( cursx, cursy ); /* and display it  */

   while( 1 ) {      /* do forever */

/* Check local console for press of ESC key.  */

            if( bdos( 6, 0x00FF ) == ' \033' ) break;

     /* Check the mouse for a character */

            if( mstat() ) {
```

```
/* If command line parameter was present, just display it. */

            If( argc > 1 ) printf( "%02x", minput() );
                 else    {

/* Read the 5-byte packet from the mouse  */

                      for( i = 0; i < 5; ++i )  {
                          while( ! mstat() ) /* wait */ ;
                          byte[ i ] = minput();
                     }

/* Process buttons in byte 0   */

                      butstr[ 0 ] =  butstr[ 1 ] = butstr[ 2 ] =  ' ';
                      if( ! ( byte[ 0 ] f 0x04 ))  butstr[ 0 ] =  'L';
                      if( ! ( byte[ 0 ] f 0x02 ))  butstr[ 1 ] =  'M';
                      if( ! ( byte[ 0 ] f 0x01 ))  butstr[ 2 ] =  'R';
                      goxy( 75, 23 );
                      printf( butstr );

/* The cursor movements are signed characters.  Process these.  Use a
   slew of 256 as full-screen.  The Y movement needs to be negated. */

                      cursx += 80 *  ( extend( bste[ 1 ] )
                          + extend( byte[ 3 ] )) / 256;
                      cursy -= 24 *  ( extend( bste[ 2 ] )
                          + extend( byte[ 4 ] )) / 256;

/* Make sure the cursor stays  on the screen  */

                      if( cursx < 0 ) cursx = 0;
                      if( cursx > 79) cursx = 79;
                      if( cursy < 0 ) cursy = 0;
                      if( cursy > 23) cursy = 23;

                      goxy( cursx, cursy );
                 }
            }
       }
   }

/* extend sign on integer */

int extend( c )
int c;
{
   if ( c > 128 ) c -= 256;
   return c;
}

/* clear the terminal screen */

clrscn()
{
   printf( "\033E"  );
}

/* go to x, y  */
```

```
goxy( x, y )
{

printf( " \033Y%c%c", y + 32, x + 32 );
}

#asm
;Z-80 SIO and CTC routines - IMS 740 slave board

CTC EQU 28H
SIO EQU 2CH
UART    EQU 1          ;uart number on SIO
CHAN    EQU 1          ;channel number on CTC

DATA    EQU SIO+UART
STAT    EQU SIO+UART+2

;   Baud Rate Divisor

B1200   EQU 16         ;baud rate divisor
#endasm

minit()
{
#asm
;   Initialize SIO

    LD A,18H            ;Channel Reset
    OUT STAT
    LD A,4
    OUT STAT
    LD A,11000100B      ;8 bits xmit, *64 Async
    OUT STAT
    LD A,5
    OUT STAT
    LD A,11101010B      ;set DTR on, RTS on
    OUT STAT
    LD A,3
    OUT STAT
    LD A,11000001B      ;8 bits recv
    OUT STAT
    LD A,1
    OUT STAT
    LD A,0          ;no interrupts
    OUT STAT

;   Set Baud

    LD A,01000111B      ;Counter mode, free-run, value follows
    OUT CTC+CHAN
    LD A,B1200
    OUT CTC+CHAN

#endasm
}

/* check serial port status */

mstat()
```

```
{
#asm
    IN  STAT
    AND A,1
    LD L,A
    LD H,0
#endasm
}

/* read serial port data */

minput()

{
#asm
    IN DATA
    LD L,A
    LD H,0
#endasm
}

#include "stdlib.c" /* include code for bzos() and other functions  */

/* end of mouse.c */
)ry   6eMUX5-4 BAK -y ZeMUX5-1 DWG hqy . pLeMUX5-2 BAK [y 8 6eMUX5-3 BAK n[y
```


MECHANICAL AFFINITY

513 EAST MAIN
PERU, IN 46970 USA
317-473-8031

5231 WILTON WOOD CT
INDIANAPOLIS, IN 46254 USA
317-291-6002

Hours 5 to 9 PM only Tues-Sat --FAX 317-473-8031 11 PM TO 7 AM.

We accept check, cash, money orders, or will ship C.O.D. All returned goods require prior approval and a Return Authorization Number. We will repair or replace at our option on hardware. Software that does not work will be replaced with the same title. We accept new products from other dealers.



We are a full line dealer offering almost all known QL, QXL and Z88 products.
SUPER GOLD CARDs, QUBIDEs, HERMES, MINERVAs, QXLs for sale!

CABLE COLUMN

By Bill Cable

ARCHIVE SERIES
PART 21 : BYCOL - A PROGRAM FOR DISPLAYING AND PRINTING DATABASES BY COLUMNS

One QL user I know uses a PC program to print out address labels because there is no QL software that will print out addresses on tractor feed labels with multiple columns. For someone who programs in ARCHIVE it is not difficult to print address labels in one, two, or more columns. I thought it might be interesting to generalize a little bit and create a program that makes it easy to print any database information in columns. Rather than create a long program with a friendly user interface using the procedures I covered in earlier parts to this series (group1,group2,group3), I made a short program that requires the user to do a little programming for each application. The complete program and instructions are included here. Also included are details explaining how the program works and a few examples using the gazet database.

Most likely you would display or print information in columns so that a large amount of information is arranged compactly but can be scanned easily. Usually the information would have an order to it. If the information were an address database then it would be arranged alphabetically by name. If we were to list addresses in columns then we could do it in 2 different ways. The addresses could be arranged so that we would scan from left to right across the columns and then down to the next row and left to right again. The second way would be to scan down the first column for a 'page' then up to the top of the second column and down to the bottom and then to the top of the third column and so forth. The first method (across a row) is the easiest to program since it needs no concept of page and we can just sequence through the data printing a line at a time. The second method (down a column) requires us to define a page and then construct the first column and then the second and then the third and so forth. Since the printer can't print a column at a time we have to store an entire page then print the page a line (row) at a time.

Bycol (the program to print columns described below) is designed to organize the data in a down the column arrangement. To print to the screen you would use a page size of perhaps 24 lines. To print to the printer you would use something less than 66 like maybe 50 lines. Then each screen or page would be filled and you would read down each column to sequentially scan the data. If instead you want the data sequenced across a row then simply set the page size to 1 and the data will still be in columns but will sequence across each row (line).

To construct a page of information the program has to use an array where each element is a line of the page. It goes through the information constructing first the column and either truncating or padding out with spaces so the first column is filled to the set width of column and gap between columns. Next it goes back to the first element of the array and starts doing the same with the second column and so forth. But ARCHIVE doesn't support explicit arrays so how is this done? I simply created a temporary database in RAM with one text field and add enough records to equal the number of lines on a page. Then the first record will hold line 1 of the page and so forth. After all the columns are put in the lines of the page the complete page is printed out (the records of this temporary database) and emptied so the next page can be constructed.

To prepare for entering the program first copy GAZET_DBF to RAM1_ and then start up ARCHIVE and get into the editor. Below you will find a listing of each procedure followed by a little explanation of what it does. After the complete program has been entered you can save it and then there are some other examples to show other things that can be done with it.

Proc bycol

```
spooloff : let ff_$="": let hd_$="": let pg_=1
mode 0: print " BYCOL - Print Database in Columns to Screen, Printer, or File"
print : print " Define columns and database in procedure - def_columns"
print " Define display lines in procedure - def_display"
def_columns: let gap_$=rept(" ",gap_)
print : print "  Database : ";sf_$
print "  Columns : ";col_;"      Column Width : ";wid_;"      Gap : ";gap_
print "  Total width : ";col_*wid_+(col_-1)*gap_
print "  Lines per display : ";lin_;"      Displays per column : ";percol_
print "  Lines per page    : ";lin_*percol_;
```

```
print "    Displays per page   : ";percol_*col_
error closer;"s"
look sf_$ logical "s"
let tpg_=int(count()/(percol_*col_)+0.9999)
print "  Total Pages to display complete database : ";tpg_
print : input " Print to screen (y/n) : ";y$: let y$=lower(y$)
if y$="y": spoolon screen : let out_$="Screen"
  else : input " Print to printer (y/n) : ";y$: let y$=lower(y$)
  if y$="y": let out_$="Printer"
  input "  Heading (if any) : ";hd_$
  input "  Use Form Feed after each page (y/n) : ";d$: let d$=lower(d$)
  if d$="y": let ff_$=chr(0)+chr(12): endif
  else : input " Print to file (y/n) : ";y$: let y$=lower(y$)
  if y$="y"
    input "  Enter output device (ram1_,flp1_,etc) : ";d$
    input "  Enter output file name (will be overwritten) : ";out_$
    if not instr(out_$,"_"): let out_$=out_$+"_lis": endif
    let out_$=d$+out_$
    spoolon out_$ export
    else : mode 1: stop
    endif : endif : endif
print : print "{printing to ";out_$;"}"
tmp_setup
let atlin_=1: let atcol_=1
use "s": first : while not eof("s")
  def_display
  use "s"
  next
  endwhile
use "t"
search val_$<>""
if found():col_out: endif
closer;"s":closer;"t"
if out_$<>"Screen"
  print : input "All done. Press <ENTER> : ";d$: mode 1
endif
endproc
```

Bycol is the main procedure of the program. Note that many of the variable like ff_$ have an underscore (_) in their name. I did this to prevent conflicts between the program variable names and the field names of a database you might want to use the program on. I figured it was unlikely that the field names of your database will use underscore. If a program variable and your database variable do happen to have the same name then your database variable will take precedent and may interfere with the operation of the program. The spooloff/spoolon command is used to control where the lprint will send its characters. Because the program may be restarted after a crash I use spooloff above to make sure printing will go to printer unless the program redirects it. A heading along with information about the defined column display is printed. The procedure def_columns is activated. It defines the columns and records per column and the name of the database to be used. The closer procedure is used to make sure the database is closed before it is looked at. The error statement suppresses an error if it happens to not be accessed. The error statement only works on a user defined procedure so I can't say ' error close "s"' but instead must have the close within the procedure. The user is given options for where the output is to go. The database to be used (source) is looked at and identified with the logical name "s". If the output goes to

the printer then form feeds can be used to make pages with a heading. If output goes to a file then the file is created with a spoolon statement. Any existing file of that name is automatically deleted. An extension of _lis is put on the file name if there is none. A temporary file that will hold a page of information is created in procedure tmp_setup and is given logical name "t". So 2 databases and possibly one text file (if output goes to a file) are accessed. Next a while loop is used to sequentially (first to last) access the records of the source database. The def_display procedure has the information about the printout and is accessed for each record. A column at a time is constructed until the whole page of columns are filled and then they are printed out. This can take some time. After the last record is reached a check is made to see if any data is left unprinted. If so it is printed then the program ends.

```
proc closer;log_$
close log_$
endproc
```

Closer is a simple procedure to close a file. The passed variable log_$ is the logical name of the file to be close. When this procedure is called in bycol it has an error statement in front of it because is will be used even when the file to be closed may not be presently accessed. This would normally cause an error but the error statement suppresses the error.

```
proc col_out
if hd_$<>""
 lprint hd_$: lprint tab 60;date(0);"   ";pg_;" of ";tpg_: lprint
 let pg_=pg_+1
endif
first : while not eof()
 lprint val_$
 let val_$="": update
 next : endwhile
if ff_$<>"": lprint ff_$;: endif
let atcol_=1
endproc
```

Col_out is the procedure which actually prints out a page of columns. After each line of columns is printed it is cleared (set to null). The lines are stored in the temporary database. The lprint statement sends the line to the printer or screen or file depending on what the spoolon was set to (no spoolon defaults to printer).

```
proc col_put;tex_$
use "t"
if atlin_>percol_*lin_
 let atcol_=atcol_+1: let atlin_=1: endif
if atcol_>col_
 col_out
endif
position atlin_-1
if len(tex_$)>wid_: let tex_$=tex_$(1 to wid_): endif
if len(tex_$)<wid_
 let tex_$=tex_$+rept(" ",wid_-len(tex_$))
```

```
endif
if atcol_<col_: let tex_$=tex_$+gap_$: endif
let val_$=val_$+tex_$: update : let atlin_=atlin_+1
use "s"
endproc
```

Col_put is the procedure which fills each column until all the columns of the page are filled. It then calls col_out which prints the entire page of columns. This procedure is repeatedly called by bycol. Each column is padded out or truncated to match the specified width. Each line of columns is stored in the temporary file.

proc def_columns

```
let col_=5: let wid_=14: let gap_=1
let lin_=2: let percol_=12
let sf_$="ram1_gazet_dbf"
endproc
```

def_columns is the procedure which defines the column and page and display layout of the information and also the database file to be used. The variable meanings are :

- col_ - The number of columns on the page
- wid_ - The width in characters of the column
- gap_ - The gap between columns in spaces
- lin_ - The lines used in each record display
- percol_ - The number of displays in a column to make up a page. Each display may have more than one line
- sf_$ - The device and complete name of the source database to be displayed

This procedure needs to be modified each time you need to change the column or page or display length or file name. In this instance we have set up for a 5 column display of 2 lines long with 12 displays per column for the gazet database. This nicely fits the screen.

proc def_display

```
col_put;" "+country$
col_put;"   "+capital$
endproc
```

Def_display is the procedure which controls what the display will look like for each record. The number of lines in the display need to match the lin_ variable value set in def_columns. Each text string to go on a line is constructed and passed to col_put. If the text is too long to fit the column it will be truncated there. You can use the various functions and commands of ARCHIVE to make intricate displays. This procedure needs to be modified each time you need to change a display or change to a different database.
In this instance we display the country name on line 1 and the capital (indented) on line 2.

proc tmp_setup

```
error closer;"t": kill "ram1_col_tmp"
create "ram1_col_tmp" logical "t"
  val_$
endcreate
let i=1: while i<=percol_*lin_
```

```
  let val_$="": append
  let i=i+1: endwhile
endproc
```

Tmp_setup deletes the old temporary file if it exists and creates a new one. Each record has a single text field called "val_$" which will hold values from all columns for that particular line. Enough records are created to hold the correct number of displays to go in a column where each display consists of lin_ lines.

Example 1

This is the complete program. Now save it as "bycol". Try running it by typing :

bycol<ENTER>

Correct any mistakes. If it works right you will be asked if print should go to screen. Answer "y". Notice it takes a while for each screen to be constructed. There are 60 displays (records) on the screen. Remember to have gazet_dbf in RAM.

Now try changing the def_col procedure so per_col_=25. This will make the page have 50 lines (lin_*per_col_). Now trying having the output go to the printer. Use "Country and Capitals" for the heading and use form feeds. Notice it takes even longer as there are now 125 displays per page.

Example 2

Now try changing def_columns and def_display as follows. This will print more information about each country. Notice that the number fields (pop, gdp, area) are changed to text before being passed to put_col.

proc def_columns

```
let col_=3: let wid_=20: let gap_=3
let lin_=6: let percol_=9
let sf_$="ram1_gazet_dbf"
endproc
```

proc def_display

```
col_put;" "+country$
col_put;"  "+capital$
col_put;"  "+languages$
col_put;"  "+currency$
col_put;gen(pop,6)+gen(gdp,6)+gen(area,6)
col_put;""
endproc
```

Example 3

Now try changing def_columns and def_display as show below. It arranges country and capital on a single line in columns.

proc def_columns

```
let col_=2: let wid_=39: let gap_=1
let lin_=1: let percol_=24
let sf_$="ram1_gazet_dbf"
endproc
```

```
proc def_display
let i$=country$: let i=len(i$)
if i>20: let i$=i$(1 to 20)
  else : let i$=i$+rept(" ",20-i): endif
col_put;i$+"  "+capital$
endproc
```

Example 4

Now suppose you had an address database. You might have different field names than used in the example below. If so then modify as needed. This example prints out our sought after address labels in columns. This is set up for the screen but can be easily altered for printer. To print to paper probably set percol_=9 with a heading and form feed. To print to tractor fee labels set percol=1 and no heading or form feed. Remember to put your address database in RAM.

```
proc def_columns
let col_=3: let wid_=25: let gap_=1
let lin_=6: let percol_=12
let sf_$="ram1_address_dbf"
endproc

proc def_display
col_put;" "+fname$+" "+name$
col_put;"  "+street1$
if street2$<>"":col_put;"  "+street2$: endif
col_put;"  "+city$+", "+state$+"  "
col_put;"        "+zip$
if street2$="":col_put;"": endif
col_put;""
endproc
```

Example 5

To print a handy phone book type list from an address database use the following settings.

```
proc def_columns
let col_=2: let wid_=39: let gap_=1
let lin_=1: let percol_=24
let sf_$="ram1_address_dbf"
endproc

proc def_display
let i$=" "+name$+"   "+fname$: let i=len(i$)
if i>24: let i$=i$(1 to 24)
else : let i$=i$+rept(" ",i-24): endif
col_put;" "+i$+" "+phone$
endproc
```

Play around with other databases too. There are many useful things you can do. Next time more fun with ARCHIVE. Any suggestions are welcomed. Until then, Happy Archiving!




QBOX-USA

24 hours

810-254-9878

300-14400 bps

Now running with a QUBIDE interface and a hard drive!

We're expanding! Lots of new files! New message and file areas will be added soon. Give us a call and let us know what you want to see here. We still have the following areas available:

MESSAGES	FILE AREAS
Local	General
Private	Utilities
International QL	Communications
Minerva	Archivers
Quanta	Pointer Stuff
QBox Sysop	Emulators
Spectrum	Spectrum/2068
Netmail	Private

If you like the 2068 but use a PC, we have a Timex 2068 emulator available for download.

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

QLuMSi 4.80
by Al Feng

QLuMSi/QL.using.**M**'cr~**S**~ft.interface (yes, it's pronounced "clumsy") is a front-end program, an MS-DOS simulation, and a learning tool. **QLuMSi** is a TURBO-compiled program that can be EXEC(_W)'d on any QL which has TK2_EXTensions.

For individuals who move between QDOS and DOS, **QLuMSi** allows for a smoother transition when invoking command line input -- **QLuMSi** will convert the delimiter for you. Either an 'underscore'/'_' (the QDOS's standard delimiter) or a 'period'/'dot'/'.' (the standard DOS delimiter) may be used within filenames.

For individuals whose spouse may not want to use their QL because they use a DOS system at work, this allows them the comfort of the "standard" prompts and commands. For example, typing "quill" at an 'A:\>' prompt will instruct your QL to "EXEC_W flp1_quill" and load the **Quill** program if it is present on the disk in flp1_.

For individuals who are not familiar with "standard" DOS commands, it gives them an opportunity to easily familiarize themselves while allowing them to continue to use the QL programs with which they are already familiar.

Where possible, **QLuMSi** input mimics the MS-DOS command line syntax. Because **QLuMSi** should also be thought of as a front-end program, some of its functions exceed what can be done from the DOS command line.

Every attempt has been made to ensure **QLuMSi** compatibility with the demands of the MINERVA ROM code as well as the variations found in the SMSQ's SBasic code.

PATHs

Now that hard drives are more common, PATHs have been reassigned from ealier versions to closer mimic the DOS standard.

```
C: == win1_    D: == win2_    E: == win3_
H: == win4_    I: == win5_    J: == win6_
K: == win7_    L: == win8_
```

You may notice that the sequence is interrupted, with 'F:' and 'G:' being assigned to 'flp3_' and 'flp4_', respectively.

Microdrive access is only assigned to the '_COMn' (*i.e.*, == common) variation:

```
M: == mdv1_    N: == mdv2_
```

All other versions begin the 'ndk()_' sequence with 'M:'.

Some NETwork comments, etc.

This is a good time to explain why 'ndk()_' is a PATH equivalent.

After some trial-and-error, I have found that 'ndk' is a good abbreviation for **n**(etwork)**d**(is)**k** [*after* fdk/hdk].

When assigning a device name during the NFS_USE declaration, any word which is NOT already used by QDOS (or, extensions) may be chosen.

However, if your system does not have a hard disk (interface), you may use the eight win()_ devices for NETworking.

Some of you may wonder what and why there are provisions for up to eight (8) ndk()_ devices [six with **QLuMSi_COMn**]

The reason that there are eight ndk()_ allowances is because you can, in fact, assign that many designations. This is in keeping with what is apparently the standart QDOS device limit of eight.

This quantity may not have been obvious because the original TK2 manual was so tiny that it was difficult to read (of course, that was not the only reason that it was difficult to read!).

In fact, I ascertained the eight device NETwork capability as a what-if supposition, and then went back and re-read the manual to determine whether this had been documented. Of course, it was.

PEER-TO-PEER NETwork commuication, where the QDOS devices have equal access to one another, becomes viable when faster QDOS devices are attached to one another. I currently have a GOLDCARDed QL set up as a "peer" with a QXL by implementing the following commands in the respective boot programs:

```
      QXL:  NET 1
            FSERVE
            NFS_USE NDK, N2_flp1, N2_flp2_,N2_ram1_,N2_ram2_,
                         N2_ram3_,N2_ram4_,N3_ram1_,N3_mdv1_

GOLD CARD:  NET 2
            FSERVE
            NFS_USE WIN, N1_win1, N1_win2_,N1_win3_,N1_win4_,
                         N1_win5_,N1_ram1_,N1_ram2_,N3_ram1_
```

COMMANDs

Typing "HELP" will display this list of COMMANDs:

CD \	CLS	COPY	DATE
DEL	DIR	FORMAT	FREE
MEM	PRINT	RENAME	STAT
TIME	TYPE	VER	VIEW

'RD \' & 'MD \' are also supported.

However, since 'MD \' accesses the MAKE_DIR keyword which is not available on all systems, **QLuMSi_COMn** is amongst the variations provided for systems lacking the MAKE_DIR keyword.

Typing "HELP ..." (where '...' is the name of a supported command) will reveal more detailed information.

The **COMMAND_COM** program is the same as the "full" **QLuMSi_COM** program except that it lacks the additional HELP beyond the HELP screen which shows the COMMANDs and PATHs which are supported. **COMMAND_COM** has the advantage of being "smaller" than the "full" program.

QLuMSi_EXE is the **QLuMSi_COM** program which begins with a 'C:\>' prompt (*i.e.*, 'win1_').

Similarly, **COMMAND_EXE** is the **COMMAND_COM** program which begins with a 'C:\>' prompt.

FLIST_imp == F(ile)LIST_imp(ort)

QLUSTer and/or **QLAMBer** users will recognize the FLIST_imp as the dynamic file which contains the names of the sub_DIRectories and files on the medium. The medium must not be write-protected.

The FLIST_imp file can be imported into **Quill** or any other word processor or text editor.

Although **QLuMSi** "hides" the FLIST_imp file *(on prior versions, the file was named ' ')*, it is present on the medium.

deFILE_exe

The **deFILE_exe** utility is a truncated version of the **QLAMBer** front end & utility program. It can be run as a stand-alone program and providided to facilitate removal of the ' ' datafile by people who are upgrading from earlier versions of the program.

SELECT_DEVICE: Use the up/down arrow keys to move the shaded, green bar up-or-down and the left/right arrow keys to change the drive number (1-4). Press ENTER to indicate your choice. The program will then go directly to the deFILE "page."

Again, use the arrow (cursor) keys to move the highlighting box over the appropriate filename; and then, press the ENTER key.

Press the '(esc)ape' key to abort the DELETE process or to exit the program.

VIEW [filename]
COPY [filename] LPT1.

Both of these functions now recognize **Quill** files appended with a "_doc" suffix and generate a roughly formatted screen-or-paper output.

If you VIEW [filename] or COPY [filename] LPT1. the "UPDATES_doc" you would see something like this:

vrm1qdf0

&
H
page nnn

QLuMSi 4.80 Updates

Where possible, QLuMSi input mimics the MS-DOS command line syntax. Beca use QLuMSi should also be thought of as a front-end program, some of its f unctions exceed what can be done from the DOS command line. To this end, VI EW and COPY now recognize Quill generated "_doc" files and generate more usa ble output.

VIEW [filename] & COPY [filename] LPT1.

Both of these functions now recognize Quill files appended with a "_doc" suffix and generate a roughly formatted screen-or-paper output.

If you VIEW [filename] or COPY [filename] LPT1. this Updates_doc you would see something like this:

and, so on.

Shortly after the end of the document is located, the VIEWing will halt, or a FORM FEED will be sent to the printer.

Although the output is not what you would get using **Quill** and a **printer_dat**, it will allow you to look at a screen or paper copy of a document within the **QLuMSi** program.

Of course, this might seem to be a bit of a *kluge* for salvaging a corrupted "_doc" file; but, you should be able to salvage all/part of the text of a "corrupt_doc" file(s) whose formatting data has been corrupted:

COPY [corrupt_doc] lpt1.

As before, the non-ASCII characters are "red" when VIEWed and empty spaces when a COPY is sent to the printer.

If the "_doc" suffix is not present, the program will display the file without the rough formatting.

DIR
DIR /W

An implicit '/p'(for "pause") is now presumed. In addition, the standard DIR display now more closely mimics the DOS display:

```
A:\> dir
    Volume in drive A is @ PLATYPUS
    Directory of A:\

Quill          60614 04-02-95 19:33
Archive        52814 04-02-95 19:33
- - - - -
misc ->
turbo ->
printer_dat       85 04-02-95 19:37
GPRINT_PRT       510 04-02-95 19:37
QLAMBer        32330 04-02-95 19:37
QLUSTer        32470 04-02-95 19:37

 Press any key when ready ...
```

The number of files that are displayed on the screen is dependent on whether the "banner" is displayed ("CLS" vs. "SHOW").

If you want the DIRectory of another PATH, then type:

DIR [path_letter]:

To see a DIRectory data displayed in four columns, type:

DIR /W
or, DIR [path_letter]:/W

Extra spaces may return a "Bad command or file name" message.

THE BOTTOM LINE

This remains as an *UPDATE! issue disk* (see back cover).

Previous users [*direct*(**CATUG/ZQA!**)/**EMSoft/UPDATE!**] may acquire an upgrade from me ($5/specify disk size).

HAPPY TRAILS,
AND COMPUTING, TO YOU ...

REPORT ON OAK RIDGE QL SHOW by Frank Davis

I want to take the time to thank all of the many people who took time out of their schedules to come to the QL show (Bob Dyl thinks perhaps show is a misnomer and that it would more accurately be called a "get together"). For the last three years this has been where some of the more serious QL users in North America (and this year more Europeans than in the past) all get together to visit, see what is new for the QL or QDOS compatible systems. It is as much a chance to socialize with people of diverse backgrounds, all held together by the use of a computer that got off to a bumpy start and is just of recent come into its own.

Carol and I loaded down our van with items from UPDATE Magazine as well as hardware and software for Mechanical Affinity (plus a taste of Z88 computers and items) and headed south to Oak Ridge, Tennessee in early June. We knew that Paul Holmgren and his son John would also being doing the same from Indianapolis to meet us there. We arrived there early Friday afternoon, and it was both hot and humid, my van thermometer said 105 degrees fahrenheit. Paul said it was only 97 or 98 on his, either way it was hot and humid. We were soon greeted by Hugh Howie of Ontario, Canada. We are always glad to see him at the shows. Next we ran into Tim Swenson of the QL HACKERS JOURNAL, followed up by the appearance of Bob Dyl of IQLR. From there on we ran across the John Donaldson family, the John Rish family, Don Walterman (QBOX USA), John Impellizzeri (QBOX USA), Jochen Merz of Germany, Urs Konig of Switzerland (Cowo), Tony Firshman of UK (TF Services), Stewart Honeyball of UK (Miracle Systems), Bill Cable and his wife Mary (Wind and Wood Computing), the Al Boehm family of NESQLUG, Ruth Fegley of CATS, Herb Schaaf of CATS, Barry Washington and his wife, also of CATS. I know that I have left some names out, because that is not all who were there. Ah, yes! The hosts of the show, Mel LaVerne and wife, as well as their son Doug LaVerne. Pardon any names left out, recalling names is not my strongest point.

We settled into the motel room and made plans to all get together and go out for Chinese food. This is where I met Mike Simson and his wife, as I offered them a ride over with us to the restaurant. As there were at least three extended vans there, plus cars, we soon formed a good sized caravan to the restaurant. When we got there the restaurant seemed mighty quiet, but they sure perked up when forty some new customers showed up to buy food. It was fun. Here I met a customer of mine, whom I had known for some time by phone and mail, March Renick and his wife. It is always nice to have a face to go with the voice and name.

The next day we headed over to the church where the show was being held. We had done most of the setup the evening before (thanks to all who helped me carry in all of our trade goods, as I had only been up and about for a short while from a large hernia operation). In a short time we were ready to open the doors for business. The fun had begun.

One thing that I really liked about this years show was that we had more wives and girlfriends there than ever before. And a good many of them were primary computer users. Contrary to what some people say, there are a good number of women into computing. I know this from personal experience on the Internet. Bill Cables wife, Mary, took a group picture that I hope to wheedle a copy of for our scrap book of shows, of most or all of the female attendees. I would say that about 25% of our UPDATE readers are women. Maybe it is just IBMs that they don't like?

For Mechanical Affinity and UPDATE Magazine we took up 5 long tables just as you entered the show. How was business? Well, I had hoped for more attendees at the show. In fact there was at least one couple I had met at the motel who had come for the show that never showed up at the church. Several people who had motel reservations never showed. I guess we

had between 45 to 60 people at the show. One thing I could not complain about. Those who did come did a lot of spending. Not everyone spent dollars with us, but those who did came prepared. Thanks to all of our customers. Even Parker Lewis made the drive over for part of the day to visit with us. Hello, Parker!

We purchased some of the new Super Hermes from TF Services to offer to our customers after the show. Not all of the features of the new Super Hermes were active at that time, but by the time you receive this magazine the new chip should be with us to upgrade them to full function. They cost more than we had anticipated, but contain so many new features that they are worth it. More speed, IBM keyboard connection, direct IBM type mouse port, true serial ports, capslock LED kit to name a few. Call me to find out more about them.

The Masterpiece Graphics Card from Miracle Systems was not finished in time for the show, much to the disappointment of Stewart Honeyball as well as the buyers. It is still being worked on at this time. Here at Mechanical Affinity we are really awaiting this board. Most of you should have noticed that RGB monitors are not produced anymore and are very hard to come by. With this card you would do away with the 8301 chips so many of you seem fond of blowing, as well as finally be able to use VGA and SVGA monitors with your QL. I have another selfish reason for wanting to see this card, or something similar to it. To navigate the Internet with a graphical based program will require this. I presently use a terminal program capable of what is called VT100 to access the Internet with a Shell account (UNIX is what the Internet was based on). With this card someone can write a decent way to use a Navigator or Mosaic program to access the Internet with the ability to see the many graphic based pages and features that we are now denied. I do not want to have to purchase an IBM or MAC just to do this. Presently I can see what my QL is missing by using my Amiga to go online, and it is alot!

When the show ended I approached Urs Konig of Cowo and purchased some copies of ARCED, a program I hope to replace Editor and Editor Special Edition with. It seems to be just as good, and a bit cheaper. Some new features and only $30. The other programs seem to no longer be available separately from Digital Precision, so we needed one to replace it for our customers. We still have a few copies left of the Editor and Editor SE, but will not be getting more as far as we know. Check out the special sale flyer in the envelope with this copy of UPDATE for some good deals.

We picked up a lot of new games from Jochen Merz Software. You will soon see them in a new catalog in August. We have also added a lot of new and varied utility programs and many new graphic collections for use in art and DTP programs for your QL.

Bob Dyl tells me that at this time the plans are to have next years show back in Rhode Island. He is seeing if he can get a place where the motel would have a space to meet in. This would be more convenient to those who fly out to shows and have no car to get around in. The show next year would be run by the NESQLUG group, which is primarily out of the area surrounding Boston. They have a fine and extremely computer literate membership and would do a good job of putting on a show.

What did we sell most of at the show? Qubide IDE hard drive interfaces for the QL and Z88 palmtop computers. The Z88s were sold to QL users who mainly want them as a portable for use with their QL computers. With the Link packages it is not too difficult to transfer data back and forth between the Z88 to IBMs, MACs, QLs and to Amigas. With the Qubides many users are finally able to fullfill the need to store vast numbers of programs and files on a hard drive. Those who are waiting for the much rumoured SCSI Interface for the QL are in for a long, long wait for a product that does not seem to be making an appearance. My advice is settle for the Qubide and get a good hard drive. IDE hard drive prices now are at the lowest they've ever been and are easy to set up, as long as you follow the manual and do not get overly creative.

A REVIEW OF BANTER, the Banner Maker, by E.P.Wannum

Recently I discovered a small gem of a program that I had overlooked. This is strange as I often pride myself upon being very up to date on programs for the QL. Ah! It just goes to show that ego can often get ahead of ones true knowledge!

The program I am referring to is Banter by NICK WARD, of the UK. It was released in 1992, and while not without some minor areas that can use some improvement, it is undoubtedly the best banner making program we have for the QL. I hope here to tell you its good and bad points.

First an overview of what the program claims to do, and succeeds at. It uses "user scalable fonts". This means that the fonts can be made larger or smaller by use of the menu, by you the user. You have five sizes of each font to choose from. It will hold upto eight fonts in memory at a time. These are outline fonts, and you can add other outline fonts that you have from other programs, dealers, or from the QUANTA library. It would have been nice for this to have been vector fonts (they have no jaggedness to them) but this would have greatly increased the complexity and cost of the program. Perhaps if Nick Ward sells enough of these programs he will be inspired to do so in an upgrade. Perhaps a Pointer Environment version, also.

The second feature touted for the program is that it will make banners up to 4 pages wide. This of course means with printers that are either 9 pin or 24 pin dot matrix printers. No drivers available for the Canon Bubblejet or the HP Deskjet printers. Of course this, plus printing labels, can be a good reason to keep the old dot matrix printer around a few more years. It is meant for those that are Epson compatible printers. I found that on my Seikosha and QL printers I was actually able to make it a 5 page banner if I set my margins right, and that this was an adequate size for a banner.

You may have upto 5 lines of text on the banner. If you use large size fonts then this will not be possible, but by varying the font sizes you can get some interesting effects on the banner. If the sizes are wrong for the banner to work, the program will let you know.

Speaking of fonts, on all of them you have an Italics option. This was helpful, but it would have been nice to also have some other options, such as bold or shadow. One other point about fonts. You may only have one font in a line, but all five lines, should you desire, can have a different font.

The program allows you to have a border around the banner, or not. It is merely a line around the edge of the banner. Here I would really have liked to have 5 or 6 choices. It is, however, the only QL banner program that allows even one border as a choice.

When you have your proposed banner all set to go, the program has a preview option. This allows you to see how the banner will look before you print it out. The program is rather slow in drawing this out for you, so here is a good time to go for coffee, milk or a soda pop. If you do not like what you see then you can go back and make changes, rather than waste time printing it out, only to see that it is not to your liking. This, while slow, is a good option.

This is not a fast printing program, but then few dot matrix, graphic based programs are. Even with this I have used it for booths and tables for Psychology conferences and found it useful. As I usually take a computer setup with me, I can make banners as I need them, and as many as I have paper for. I have even been found making them up for seminar rooms and other speakers. Until some other (and I have heard of none coming) QL banner program comes along, this is the best that we have. It is better than those in the public domain. This program used to be carried by Dilwyn Jones Computing, but is now only available from Quo Vadis in the UK and from Mechanical Affinity in North America. The price is $39.

Z88BAS.TXT -- BY CHRIS FENN of Las Vegas

(editors note: this was downloaded off the Internet)

Here is a list of Z88 BBC BASIC Commands Statements and functions, I've commented on some to help clarify. Abbreviations are listed in brackets, they can be typed in as such to save memory.

COMMANDS:

AUTO [AU.] line numbering
DELETE [DEL.]
LIST [L.]
LISTO controls the appearance of a
listed program (4 options)
LOAD [LO.]
NEW
OLD recover program after using
the "NEW" command.
RENUMBER [REN.]
TRACE [TR.]

STATEMENTS:

BPUT# [BP.#] output a byte.
CALL [CA.] call machine code
with paremeters
CHAIN [CH.]
CLEAR [CL.]
CLOSE# [CL.#]
CLS
DATA [D.]
DEF define function or procedure.
DIM (supports integer variables)
ELSE [EL.]
END
ENDPROC
FN
FOR [F.]
GOSUB [GOS.]
GOTO [G.]
IF
INPUT [I.]
INPUT LINE
INPUT# input from file.
LET
LOCAL [LOC.]
NEXT [N.]
ON
ON ERROR
OPT assembler options with four different options.
OSCLI operating system command. Allows string expression to be passed.
PRINT [P.] also controls field width, sci. notation,decimal places etc.
PRINT# [P.#] write to file
PROC (PROCEDURE)
PUT output to port
READ

REM
REPEAT [REP.]
REPORT [REPO.] report error
RESTORE [RES.]
RETURN [R.]
RUN does not clear A%-Z%
SAVE [SA.]
SPC print spaces
STEP [S.]
STOP
TAB TAB(X) or TAB(X,Y)
THEN [TH.]
TO
UNTIL [U.]
VDU [V.] output bytes to screen
WIDTH [W.]

FUNCTIONS:

ABS
ACS arc cosine
ASC ASCII value.
ASN arc sine
ATN arc tangent
BGET# [B.#] byte from data file.
CHR$
COS cosine
COUNT [COU.] returns the number of chars. sent to output stream since the last new line.
DEG degrees
EOF#
ERL error line.
ERR error code.
EVAL [EV.] evaluate string. example:A=10:B=20
PRINT EVAL ("A+B")/would return 30.
EXP exponent
EXT# extent of file. Total length of file whose channel number is its argument.
FALSE [FA.]
GET/GET$ wait for key.
HIMEM [H.] first byte of free memory.
INKEY/INKEY$
INSTR
INT
LEFT$
LEN
LN natural logarithem
LOG logarithem
LOMEM [LOM.] lower memory bound.
MID$
OPENIN [OP.] open file for output.
OPENOUT open file for output.
OPENUP [OPENU.] open file for update.
PAGE [PA.] program area. Can be altered to allow several progs. to be in memory at once.
PI
POS horizontal cursor position.

PTR# file pointer.
RAD radians.
RIGHT$
RND with optional paremeters.
SGN
SIN sine
SQR
STR$
STRING$ repeats strings. A$=STRING$(2,"Z88") returns "Z88Z88"
TAN [T.] tangent
TIME [TI.] elapsed time clock
TIME$ EXAMP:Sunday April 3,1988,10:17:30 AM
TOP returns the value of the first free location after the end of the current program.
TRUE
USR call machine code at specified address.Passes the least sig.bytes of the INT. variables A%,B%,C%
D%,E%,H%,L% and F% into corresponding registers of the Z80 on entry.
VAL value of string
VPOS [VP.] vertical cursor position

OPERATORS:

+,*,^,-,/
DIV integer divide
MOD modulo

RELATIONAL OPERATORS:

<,<=,<>,=,>,>=

LOGICAL OPERATORS:

AND [A.]
EOR
NOT
OR

The equivalent of PEEK & POKE on the Z88 is handled by the use of indirection operators using VECTORS. (VEC)
INDIRECTION OPERATORS include:
? byte indirection
! word indirection
$ string indirection

For you math people. Here is the output of a program I found in TI PPC NOTES, it was converted to run on a Tandy Model 100. Hopefully I translated it correctly to run on the Z88.
it checks the math operations of the computer.

RADIX= 2
PRECISION= 32
FPWIDTH= 4.2949673E9
CLOSEST RELATIVE SEPARATION FOUND IS UlpOne= 2.32830644E-10
ADD/SUBTRACT HAS A GUARD DIGIT AS IT SHOULD.

OTHER NOTES:

1/3 returns .333333333 1=1 returns -1

Empty loop to 10000 takes 13.5 secs. to execute.
Empty INTEGER loop to 10000 takes 12.5 secs. to execute.

I've found it to be roughly as fast as a 4.77 MZ PC. but the Z88 is more accurate.

I've had pretty good luck loading M/S BASIC programs & running some programs unchanged. The W/P in the Z88 with its fast global search & replace makes it much easier to convert programs when you have to.

You can route output to the printer at anytime by pressing the square key (Alt.),+,& P. Square -P turns it off. You can send the output to a RAM file by pressing the square key +S, to turn it off would be square key-S.

The screen size in Z88 BBC BASIC is 8 lines by 94 columns. I will later upload screen codes to turn on BOLD,UNDERLINE,FLASH etc. (not supplied in the manual) Graphic commands are not supported from BASIC,such as PLOT,LINE,DRAW but I'm sure someone will be able to access the ROM routines.

Sound (not much) is in the form of PRINT CHR$(7) (bell code).

The only special codes you need to tack on to an ASCII file to load into BASIC are: #B & .J, the S#B routes the stream to the BASIC area, the .J tells it to "JAM" any commands while it's loading. You may also include "NEW" to the sequence to delete the old program first.

The Z88 includes CLI files (Command Line Interpreter). Using special codes you can build batch/macro files that can act on all the features of the Z88, including unattended operations. A few examples:"#"=the square key,"I"=the diamond key,.T> creates output file etc.

You can have many BASIC programs ACTIVE at the same time (memory providing) switching between them by pressing square key"B",or by pressing the index key and choosing which one you want. Actually you can have many documents,programs and worksheets ACTIVE switching between them in a like manner.

I am no expert on BASIC but if there is something you would like me to try on the Z88 I'll try to accomodate.

Z88 File Exchange with Other Computers

NOTICE!!!! Z88 USERS, OR POTENTIAL USERS! Not only is the Z88 a great standalone personal and extremely portable computer, but it can be of great use with most other computers that you may own. If you have a MAC, you can use MACLink to transfer files and data between the two. The same goes for the IBM type computers with the program PCLink. For QL users there is the QLink to do the same thing, as well as some PD or shareware programs to do pretty much the same. To find out how to exchange files between your TS2068, get the back issue of UPDATE or a copy of the Z88 Source Book from UPDATE. If it is an Amiga computer that you wish to exchange data with then you need to have both the PCLink and the AmigaLink software. Now just where do you think you can find all of this software? In North America the ONLY place that has all of these is Mechanical Affinity. Give Frank a call or place an order for one of the above. It will open up a whole new use for your Z88.

Eliad P. Wannum

Z88 External Battery Box

By Tim Swenson

Ever since I got my Z88, I've always been worried about running low on batteries. 20 hours on one set of batteries is nice, but good batteries are not cheap. Almost immediately I picked up an AC adapter for the Z88. This allowed me to work without the worry about going through batteries.

Since I always carried my AC adapter, I've never been too worried about finding a place to plug my Z88 into. That was until I went on one trip. I set up the Z88 on a desk with a lamp next to it. The only outlet close to the desk had the lamp plugged into it. Since I needed to use my modem (which takes up one plug), I was forced to run the Z88 on the internal batteries. Now, the most power hungry part of the Z88 is the serial port, which I was using to connect to the modem. This had me really worried about battery wear.

One neat feature of the Z88 is that the input at the AC adapter input is the same as the input from the internal batteries, 6 volts. I had heard about other Z88 users hooking in larger batteries through the AC adapter port and getting months of use.

I finally decided to build an external battery pack. I could have used a large rechargable 6 volt cell, but I wanted to keep to using standard batteries. To get the 6 volts for the Z88, all I needed was 4 1.5 volt batteries. Most standard batteries (A, AA, C, D) are 1.5 volts. I went with D cells because they were about the biggest I could get and did not cost that much more than C cells.

After looking at various electronic surplus places, I found that good old Radio Shack had exactly what I needed. Basically I needed a battery holder, a box to keep it in, and an adapter plug to fit the Z88. Below is the parts list for this project:

270-627	Experimenter Box (6.25"x3.75"x2")
270-396	D Battery Holder (6 Volt)
274-1569A	Coaxial DC Power Plut (male) 5.5mm OD
	2 Lead wire (same gauge as on an AC adapter)
	3/8" thick Foam Rubber
	4 Screws
	4 Rubber Feet

The battery holder does not fit square in the box, it fits in at a slight angle. The box is plastic with a metal cover. I wanted to call the metal cover the bottom. Since I would be easier to mount the plastic battery holder on the plastic box than the metal plate, I mounted the battery holder upside down in the box with plastic model cement (use lots).

To let the lead wires out, I drilled a small hole near the top of one end of the box (since I put this all together upside down, it looked like I drilled near the bottom of the box.).

The two wires coming from the battery holder are not long enough to reach out of the box, I ran the other wire into the box and attached it to the battery wires. To make a good connection, I twisted the wires together, put some solder on the joint, and wrapped them with electrical tape. To keep the wire from being pulled out of the case, I wrapped some electrical tape on the 2 lead wire so it would not allow the wire to be pulled through the hole.

The length of the lead coming from the box to the plug can be as long as you want. I went with a fairly long lead about 20 inches. I connected the plug to the other end of the 2 lead wire. The Z88 requires that the inner part of the plug is positive (be sure to get this right or you might blow your Z88). I would be usefull to use 2 lead wire with one lead marked (usually with a painted stipe down it's length). The positive lead coming from the battery holder is the red one. I soldered the wires on and then ran some electrical tape around between the two connecting points. I wanted to make sure that I did not get a short in the system.

Once I had this all hooked up I put the batteries in the holder and, using a multimeter, checked to make sure that I was getting 6 volts on the plug. Since the batteries were fresh, I was actually getting about 6.5 volts.

I did not want the batteries to fall out, especially since there were going to be hanging upside down, so I put in some foam rubber to support the batteries and the battery holder. I did not glue the rubber to the metal cover, since some glue will eat foam rubber, plus I did not feel a need to have the rubber mounted.

I found it odd that the box had 4 mounting holes, but came with only two screws. The screws were too tight in the mouting holes, so I bought some smaller screws to make getting to the batteries easier. The screws stuck out beyond the cover and would scratch a table surface, so I got some stick-on rubber feet to prevent this. The metal cover is fairly tight and fits the box well. To make it easier to get the lid off, I created a small notch in the cover with a metal file. I did it just big enough to get my finger nail in and be able to pry the cover off.

Total cost for the project (not including batteries and wire) was under 7 dollars. Since I took my time to get everything right, it took me about 1 hour to build the battery box.

Exactly how many hours I will get out of the 4 D cells, I'm not too sure. I do know it will be far more economical than using lots of AA's.

" 4-5 quick print PF files
TMZ 184 2068 move files
" 224 " mods available
" 232 add a dialer to PRO/FILE

QL hard
TD 3-5 heat probs
TMZ 367 modem port wiring
" 372 QL to clone cable

QL soft
SWN 2-6 tips & explanations
TD 2-3 SuperBASIC explained
" 2-4 tips & such
" 2-5 game mods
" 3-1 Etch-A-Sketch, other
" 3-2 more colors, Gas Guide, JOS, Quill tips
" 3-3 Wind Chill chart, trouble shooting, Quill tip, JOS, fixes, crater
" 3-4 Cumana I/F, Abacus tips
" 3-5 Abacus tips pt2
" 3-6 Easel tips, SuperBasic editor routine
" 4-1 Fractals, Trump card tip
" 4-2 VAL$ emulation, Fractals pt 2, Sound Explorer
" 4-3 2 screens, Fractals pt 3, design windows
" 4-4 Fractals pt 4
" 4-5 " update, ARCHIVE tips
" 4-6 resident PROCs
TMZ 115 short sound demo
" 220 Run-Length-Encoded RLE
" 275 banner
" 277 tips
" 296 Orbit simulator
" 302 File encoder
" 308 TAPE COPY
" 318 wire frame graphics
" 330 Cluster
" 344 turtle graphics
" 349 UDG's
" 364 Draw a QL
" 384 SLOT game
" 391 analyze paycheck
" 410 linked list
" 445 screen peek

QUADRA CHART PATCH soft
SWN 2-6 big printer & Tasman IF
TD 4-2

QUICK MODE for 1000/1500 soft
SWN 5-2 improved FAST?

RANDOM NUMBERS soft
SWN 5-4 2068, send for prog.

RANDOM WALKS soft
SYN 2-6
" 3-2 fix

REAR GUNNER soft game
TSU 1-1 1000

REMEMBER soft game
TD 2-3 2068

REM LINE MAKER soft
SWN 3-2 1000
" 3-6 see MC tutor (MEGA-REM)
" 5-2 2068
" 5-3 " modifier
TSH #10 1000
" #12 REM builder 1000

RENUMBER soft
SYN 2-3 in MC
" 2-5 fix for above
" 2-6 in BASIC
" 3-2 fix for above
TMZ 155 BASIC renumber 1000
TSH #16 BASIC renumber 2068

REPAIRS hard
Sum 2-9 ZX/TS pt 1, 2068 cir
" 2-10 " " 2, 1000 "
" 2-11 " " 3 16K & "
SWN 4-6 pull up res. ZX81
SYN 1-1 ZX80 circuit diagram
" 1-5 16K RAM pack cir diag
" 2-1 " " " " fix
" 3-5 PCB differences
" 3-6 " " fix
" 4-2 transistor, rectifier
TD 1-5 disk drives
" 4-2 1000

RGB CIRCUIT hard (see COLOR)
Sum 3-1 Sears unit
" 4-2 Sears tv
" 4-7 wire 2068
SWN 4-5 adjust 2068 mods
TD 3-5 for 2068
TMZ 107 Sears & wiring
" 257 better mods

ROBOTICS hard soft
SYN 3-4

ROBOT COMPOSER soft
SYN 1-5 ZX80

ROM CALLS soft
SWN 2-1 2068
SYN 3-6 1000

RS232 hard
TMZ 286 convert 2050 to RS 232 port

SCREENER 1000 soft
TSU 1-7 screen display routine

SCREEN soft
SWN 2-4 5 diff CLS routines 1000
" 2-5 bug from 2-4, quick display for 2068
" 3-3 COPY 2068
" 3-4 bug crawling on screen
SYN 1-2 build display file MC ZX80
" 1-3 simulate PAUSE to effect screen
" 1-4 fix for MC build file,
" " simulate PAUSE, scroll
" 1-5 " " fix
" 2-3 inverse all characters
" 2-5 side scrolling
" 2-6 4X char scrolling MC
" 3-1 clean up display
" 4-2 CLS part of screen
TD 2-5 2068 Poly-Scroll
" 3-1 Kaleidoscope, invert scroll
TMZ 34 2068, darken pixels
" 36 inverting screen
" 81 pattern generator 2068
" 147 expand 8X8 pixel block
" 148 " " " "
" 200 COPY corrections
TSH #3 Page 8 1000 page flip
" #17 Screen saves 2068
" #19 Explore Video modes

SEARCH & REPLACE soft
SYN 4-1 MC routines
" 4-2 " " fix

SEMAPHORE soft
SWN 4-3 2068

SERIAL soft
SYN 1-4 ZX80 serial port usage
TD 4-2 serial port driver 2068

SHARP SHOOTER soft game
TSU 1-4 1000

SHELL soft game
TD 2-2 2068, The Old Shell Game
TMZ 163 orig for 1000

SHOOTING RANGE soft game
TSU 1-7 1000

SIGN DESIGNER PATCH soft
SWN 5-1 shrink the printouts

SKETCHIT - G soft
TD 3-4 2068 drawing program

SKI soft game
TD 2-5 2068
" 3-1 fix's

SLOT MACHINE soft game
TD 1-4 2068

SORT soft
Sum 3-8 Unisort 2068
SWN 3-5 1000 Visible Sort, 2068 try various sorts
" 5-5 1000 & 2068, MC
SYN 1-4 ZX80
" 4-2 assorted sorts
TD 1-5 alphabetizer
" 2-2 short bubble sort
TMZ 316 Qwik sort 1000/2068
TSU 1-3 1000, in BASIC
" 1-4 " "
" 1-6 screen soft

SOUND soft
Sum 4-7 2068
TD 1 MC the BEEP
" 2-2 using MC to do, 2068
TSH #8 1000
" #9 1000 corrections

SPACE CRISIS soft game
SYN 3-1

SPACE TAXI soft game
SYN 2-5

SPACE WARP soft game
SYN 2-3 ZX80
" 2-6 8K update

SPEECH RECOGNITION hard soft
SYN 3-6 simple

SPECTRUM hard
Sum 2-8 make a ROM switch
" 2-9 " " " fix & PCB
" 2-11 OM, Tasword & AERCO IF
" 2-12 " " " " fix, convert to 2068 bus
" 3-4 AERCO IF & Spec ROM
SWN 2-4 add Spectrum ROM
TSH #15 bus conversion
" #16 Mterm mods

SPINNING WHEEL soft
TD 3-2 2068

SPREADSHEET soft
SYN 4-1

SQUASHED soft game
TSU 1-4 1000

STAFF soft
SYN 3-6 teach & test

STARBLASTERS soft game
SYN 3-1

STARFLEET soft game
TD 3-5 arcade type

STAR GATE soft game
TSU 1-4 1000

STAR SPANGLED BANNER soft
TMZ 307 flag & music

STATISTICAL TESTS soft
SYN 3-2

STRENGTH soft
SYN 3-2 material stress load
" 3-4 fix for above

SURROUND soft game
SYN 2-2 ZX80 Isolation
TSU 1-2 1000

SWEEPER soft game
SYN 2-3 simple

SYNTAC SUM soft
SWN 4-2 for 1000
SYN 1-4 Syncsum ZX80
" 1-5 " " fix
TD 3-4 2068 CK type
" 4-2 1000 CK Type
" 4-3 2068 revised version
" 4-5 1000
TSH #11 2068, 1000 Linecheck routine

TANK BATTLE soft game
SYN 2-3 ZX80

TAPE INDEX soft
TD 4-5 2068 indexer prog.

TAPE RECOMMENDATION
Sum 4-5
" 4-6 2068 backups
TSH #3 1000 tape Indexer

TARGETS FIVE soft game
Sum 4-3 1000

TAXMAN soft game
SYN 1-6 ZX80
" 2-1 " fix

TAX CALCULATOR soft
TD 2 for 2068
" 4-2

TECH MANUAL
SWN 2-4 MODE POKEs
TMZ 85 corrections
" 270 "

TELECOMMUNICATIONS TIPS
Sum 2-10 term & big printer
" 3-8 MtermII tips
SWN 2-3 Byte-Back RS232
" 3-2 " " " (hard)
" 3-4 MTERM & big printer, other tips
SYN 3-5 Byte back
TD 2-5 MTERM mods
" 3-1 tone dialing
" 3-2 see RLE
" 4-1 Spectrumize MTERM
TMZ 55 ref guide for MTERM

" 85 MTERM & big printer
" 100 loader & tips (poor)
" 109 adding MC
" 130 AERCO & MTERM & printer
" 145 MTERM & Spectrum
" 174 dialer POKEs 2068
" 192 " " "
" 223 " BASIC
" 246 MTERM II tips
" 259 MTERM stuff
TSH #11 ByteBack 8251 chip
" #12 Mterm 1 big printer
" #16 spectrum & Mterm

TERMINAL hard soft
SYN 2-6 convert to serial term.

TIC TAC TOE soft game
SYN 1-2 ZX80
" 1-3 fix for above
" 1-4 " " "
TD 1-5 2068
" 2-6 3D
TSU 1-3 1000, 3D
" 1-7 " fix

TIMEMACHINE PATCH soft
TD 4-5 on AERCO
TMZ 304 2 bugs

TIME BOMB soft
SWN 5-4 1000, Time tutor

TIME ZONES soft
TD 4-2 2068

TIOGA TOADS soft game
SYN 2-1 ZX80

TL$ soft
SYN 1-4 ZX80, 2 articles
" 1-5 " splitting strings

TONE DIALING soft
TD 3-1 2068

TOWER soft game
SYN 1-6 ZX80/ZX81

TOWER OF BRAHMA soft game
SWN 3-4 2068

TRAIN soft game
SYN 2-1

TREASURER HUNTER soft game
TSU 1-5 1000

TREASURE HUNT soft game
TSU 1-3 1000

TRIANGLES soft
TSH 21 simulate modulo

TRY THIS, JUST FOR FUN, SHORTS soft
SYN 1-3 2-5
" 2-6 also fix for 2-4, 5
" 3-1
" 3-2 also fix for 2-6
" 3-4
" 3-5 also fix for 3-2
" 3-6
" 4-1 also fix for 3-6
" 4-2
TD 1-4
" 2-6
" 4-1 1000
TMZ 61
" 104
" 120 game
" 161
" 220
" 295 2068
" 324 "
TSH #4
" #13
" #15
" #16
" #18
" #20
TSU 1-6
SWN 2-3
" 2-4
" 3-1
" 3-6

TS COMMUNIQUE hard soft
TD 3-6 hints & tips

TURBOS soft
TD 2-2 2068 and autos

UDG soft
Sum 2-9 expanded
" 3-4 for the 1000, + hard
" 3-5
" 4-4 Sprites
SWN 3-4 for 1500
SYN 4-2
TD 2-5 Designer graphics
" 3-1 " " son of
TMZ 121 mods for previous item
" 194 prog with DATA lines
TSH #13 In Sync #2

UPPER/LOWER CASE soft
TD 2-4 change from lowercase

USE IT soft
TSU 1-2 plotter, sine wave, add, sub, mult, divide, char listing, bull board,

VDAQ hard soft
SWN 1-5 input & chart analog DATA into 1000

VOTEM 1000 hard
SWN 1-4 control cass. deck

VU-3D soft
SWN 3-5 Tasman IF patch

VU-CALC PATCH soft
Sum 3-11 & J
SWN 3-5 1000 patches
" 4-1 2068 & Tasman IF
" 5-2 to disk
TD 3-2 and big printer
TMZ 34 2068
" 136 " & Tasman
" 142 " & AERCO
" 172 "
TSH #15 big printer
" #16 mods

VU-FILE soft
SWN 5-2 to disk
TMZ 173 tips 1000

WAFADRIVE soft
TD 4-5 tape to wafadrive

WIDGET soft game
SYN 1-2 ZX80
" 1-3 fix for above
" 1-4 " " "
" 1-6 " " "
" 2-5 8K update

WINDOWING soft
SWN 4-2 for 2068
SYN 4-1 on the 1000

WINDOW soft
SWN 5-5 2068
SYN 2-1 MC monitor
TD 4-4 2068

WIND CHIMES soft
SYN 3-6

WORD PUZZLES soft game
Sum 4-1 Solving Jumbled
" 4-3 correction
SWN 2-4 Word Images 2068
" 4-2 Wordsearch
SYN 1-6 ZX80
" 2-5 " mixed sentences
" 3-2 mods to above
TD 3-3 Cube-it
TMZ 56 2068 Word puzzle
" 212 rewrite of Sum 4-1
" 217 words from tele no.s
TSH #14 jumbles from tele no.s
TSU 1-5 Dial a word
" 1-6 Crazy words

WORD WRAP soft
SWN 4-2 2068 utility
SYN 2-6 in BASIC for text

WORLD GEOGRAPHY soft
SWN 5-4 1000, draw maps
" 5-5 " Fixes

WORM soft game
Sum 3-9

W-P soft
Sum 2-8 Tasword & AERCO CPI
" 3-2 " mods
" 3-4 " & A & J, Textwriter to Tasword
" 3-11 2068 Poor Mans
" 4-1 Tasword POKEs
" 4-6 " & Zebra serial port
SWN 1-4 caveman
" 2-4 Tasword II patches, Fat bits for MSCRIPT
" 2-5 customize MSCRIPT
" 2-6 MSCRIPT & Tasman IF
" 3-5 1000 2K text Writer
" 3-6 customize MSCRIPT
" 4-4 EPROMize MSCRIPT
" 4-5 quick print TW II files
" 4-6 " " bug fix
" 5-1 MSCRIPT & JLO CPI, JLO DOS & MSCRIPT
" 5-2 MSCRIPT & JLO CPI paper empty
" 5-3 PROFILE to MSCRIPT
" 5-4 1000/2068 simple WP
" 5-5 fix for above
TD 1-4 Tasword tips
" 2-2 " AERCO patch
" 2-5 " mods
" 3-2 redone Tasword basic
" 3-4 Dohaney mods MSCRIPT

" 3-5 adding to TW II
TMZ 85 AERCO & Tasword
" 95 Tasword II guide
" 101 Tasword " big printer
" 109 Tasword PAPER/BORDER
" 114 Tasword II & A & J, MSCRIPT & Tasman IF
" 118 Tasword big printer
" 141 MSCRIPT
" 155 Tasword II mult copies
" 158 fixes for #114 above
" 204 TEXTWRITER patches
" 235 Tasword II & AERCO CPI
TSH #5 Worm pt 1 1000
" #6 " " 2 "
" #7 " " 3 "
" #8 " MC error, Tasword
" #9 " corrections
" #13 " enhancements, 2068 in basic
" #16 Worm mods

YAHTZEE soft game
SWN 4-5 2068 version of game

ZEUS PATCH soft
TD 3-4 2068
" 4-3 2068 FIX

ZIP PATCH soft
TMZ 317 joystick

ZXLR8 hard soft
SWN 5-1 and the 1500

ZX DESTROYER soft game
SYN 2-4 ZX80/81
" 2-5 fix for both
" 2-6 " " "

ZX MAZE soft game
SYN 3-2

ZX STOCK EXCHANGE soft game
SYN 3-5
" 4-1 fix for above

MECHANICAL AFFINITY
c/o Frank Davis
513 EAST MAIN ST.
PERU, INDIANA 46970
Phone 317-473-8031

MECHANICAL AFFINITY
c/o Paul Holmgren
5231 WILTON WOOD COURT
INDIANAPOLIS, INDIANA 46254
Phone 317-291-6002

ITEMS FOR THE Z88!

Soft Carrying Case for the Z88 . Protect your notebook sized computer from accidents and the elements. This case is also handy for those with the Laser PC3, Compumate 3 or PC4. $10

32K RAM EXPANSION for the Z88, only $33, limited supply.
256K EPROM EXPANSION for the Z88, only $80.
128K RAM EXPANSION for the Z88, only $47, limited supply.
512K RAM EXPANSION for the Z88, only $90.
One Meg RAM EXPANSION for the Z88 only $170.
Z88 to MAC CABLES for only $10.
QLINK , QL to Z88 software, for only $30.
QL to Z88 Cables for only $15.
PC LINK2, IBM to Z88 cable & software, for only $36.
MAC LINK, Macintosh to Z88 cable & software, for only $45.

NEW Z88 COMPUTERS , with new manuals, carrying case, battery powered notebook sized computer. It comes with built-in word processor, database, diary, comm facilities, eprom slot and built-in eprom burner to save programs. Expandable with two RAM slots and one EPROM slot. All of this for $175, new. Used Z88 computers for $115.

* TS2068 UPDATE ISSUE DISKS *

These disks contain a program, suite of programs, or a set of utilities that have either been presented in UPDATE, or sponsored by. This is done in the hopes of encouraging and sponsoring the authoring of TS2068 or Spectrum software by North and South American programmers. They are guaranteed to be worth the money. At times vacant space left on these disks is also filled with particularly appealing Public Domain or Shareware programs for the TS2068 or Spectrum. We also welcome programs from our other readers anywhere in the world. We support at this time Larken and Oliger disk formats for the TS2068. Contact us for other formats. Please let us know what size, interface format and density of disk you wish the program in. Half of the funds received go to the author of the program. Please feel free to offer a program for inclusion in our Issue Disk Series.

---

1) THE JOHN McMICHAEL GRAPHICS COLLECTION- This is a six disk set of IBM clip art graphics converted for use with Print Factory, the desk top publishing program for the TS2068. Can also be used with Pixel Print, by Lemke. They are offered in Larken and in Oliger disk format. Each disk is $7, or $40 for the entire collection. The best ever for the TS2068.

2) LOGICALL PROFESSIONAL V5.2- This is the Auxiliary Operating System and Integrated Software Package that completes your Larken multi-drive system. It allows you to move in and out, and around your system and programs with usually two keystrokes or less. No need to turn off your machine. The best Larken improvement yet. Soon to be available for the Oliger sytem. Authored by Bob Swoger. $15 for the program.

3) THE WIDJUP COLLECTION- This contains some of the most popular programs formerly offered by WIDJUP. They are not Public Domain, all were commercial and given to UPDATE for distribution by Bill Pedersen with his passing. This is a two disk set, and does not contain his CAD program. It has editors, printer drivers, games, TS2068 tutorials, etc. In Oliger or Larken format. Priced at $20.

4) WIDJUP'S CAD PROGRAM- This is a long time favorite for Larken and Oliger users. It will give you professional results from your TS2068 in the area of computer aided design and the development of printed circuit boards. Bill used to even use it to make disk labels and DTP. Let us know which format and whether it is for either an IBM compatible printer or an Olivetti Inkjet printer. The price is $20.

5) OLIGER DISK DRIVE BBS PROGRAM- This will create a single user BBS program, with several message bases, E-mail, and SYSOP chat area. The rest of this disk is chocked full of other programs either by, or enhanced by, Paul Holmgren (program author). In Oliger format only. The price is $20.

6) 24-PIN BIT IMAGE GRAPHICS FOR 24 PIN OR BUBBLE JET PRINTERS- This program, by Larry Crawford, is for Epson emulation modes on your printer. It takes the mystery out of graphics when used with some of the newer printers out there on the market. We have also included some extra software with this one, and all of this for $15. It is available in either Larken or Oliger format disks.

Needless to say we are always interested in a new issue disk we can present here for our readers. If you are out there writing programs, or know someone who is, give us a chance to present your software to the users, and perhaps make a bit of change in the bargain. This way we all win. We make royalty payments twice a year based upon the sales. Sales are not guaranteed, but we do our best!

* QL UPDATE ISSUE DISKS *

These disks contain a program, suite of programs, or a set of utilities that have either been presented in UPDATE, or sponsored by. This is done in the hopes of encouraging and sponsoring the authoring of QL programming by North and South American programmers. They are guaranteed to be worth the money. At times vacant space on these disks is also filled with particularly appealing Public Domain or Shareware programs for the QL. We support all know disk formats for the QL. Please let us know what format you wish yours in when you order. The prices are as listed and do already include postage and handling. Half of the funds received go to the author. Please feel free to offer a program for inclusion in our Issue Disk Series.

---

1) HARTUNG UTILITY ISSUE DISK- Here we have some excellent programs for the QL, such as a stand alone database, an Address and QSO file program, etc. It also gives lots of programming hints and tips for QL programmers. All are in SuperBasic. The Address file can also be used as an inventory program, or use it to print out labels. Both screen and printer output can be alpha-sorted, or done by last name. The price is $15.

2) CABLE ARCHIVE ISSUE DISK- Written by Bill Cable, perhaps Americas best known programmer for Archive. 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. Commercial Quality! The price is $20.

3) QLUSTER 5s109 ISSUE DISK- A great program from Al Feng to provide you with many utilities to handle and unclutter your disks and MDVs (and it now supports sub-directories such as in Level 2 devices such as FLP LEVEL 2 ROM for Trump Cards and Gold Cards). 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 (not emulator) 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 ISSUE DISK- Al Fengs latest issue disk and it is even compatible with the QXL now! 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 now $20.

6) COMMS & COMPRESSION COLLECTION- Eliad Wannums collection of the best P.D. and shareware fax, telecommunications, and file compression and decompression programs available for the QL. This is a four disk set that is extremely useful, and all the disks are fairly loaded. The price is $20.