ELF II NEWSLETTER
Netronics Research

FEBRUARY, 1979 VOL. I NO. 1 PUBLISHED MONTHLY

Greetings
by Tom Pittman

Welcome to the first issue of the Netronics ELF II Newsletter. As many of you have noticed, this first issue has been a long time coming. Two weeks ago I was asked to take over as Editor, so I think we will be on a more regular schedule from now on. Those of you who have been waiting for programs in my mill can (quite properly) take that with a grain or two of salt, but I think writing a column once a month and collecting together some contributed programs should be less fraught with delay.

Speaking of contributions, I want to emphasize that most of the programs in this rag will be simply what you, the readers, contribute. I think we can make it worth your while: any program you contribute that we publish is good for a free year's subscription to the newsletter. From time to time (perhaps as often as once a month) we will feature a "Program of the Month" with a special award ($50 credit toward any Netronics hardware or software). Of course all submitted programs become the property of Netronics (we gotta get something for our money!).

We are looking for programs in three general categories:

1. Minimum ELF II. Programs which run in 256 bytes of RAM, using only the hex keypad/display and/or the CRT for I/O. We will seriously consider clever programs that require only a small amount of extra hardware if it is well documented.
2. Expanded ELF II. Programs in this category are those which depend on the Giant Board monitor and/or its I/O ports, or those which need extra RAM, or those which use the new A/D converter or color video. Again, extra hardware is not precluded if it is well documented.
3. Tiny Basic. Any programs which run in Netronics Tiny Basic are of great interest. If you can fit the program into 4K or 4.3K (one 4K board + 256), that is all the better, but longer programs will certainly not be excluded.

I would like to encourage you to send a machine-readable copy of the program listing, if you can, to prevent my typing from introducing bugs. I prefer paper tape, but cassette is OK. If you send cassette, be sure to send a paper listing also, in case I have trouble reading your cassette. Please address all contributions and letters to

Editor,
%Netronics R&D
333 Litchfield Rd.
New Millford, CT 06776

If you include a Stamped Addressed Envelope ("SASE"), I will return your manuscript (and tape).

I suppose many of my readers will be concerned about objectivity, now that Netronics is paying me on a regular basis. I hope you will believe that as a matter of principle I will never tell you something unless I honestly consider it to be true. If I ever find myself in a position to compare Netronics products to "the competition", I will tell it like it is; perhaps you may suddenly find a new editor writing to you from these pages, but I really do not think George is that kind of person. Anyway, I would not have agreed to this position if I thought that the ELF II were an unworthy product.

Other Clubs

Besides the house organs from the other 1802 dealers, there are two Cosmac-oriented clubs with significant newsletters. Though nobody in this business actually meets their publication schedules (including us!), I think they are doing a good job and should be encouraged.

The most significant, perhaps because they were first, is the **Association of Computer Experimenters** (ACE), based in the Toronto region. They started out with a local club of some 200 members, but because there was nothing else, they quickly acquired a mailing list covering the whole continent. Their newsletter, IPSO FACTO, is consistently chock full of goodies — 40 or so pages of them every time.

A one-man effort in Los Angeles has developed into the second large club, the **Cosmac Users Group** (CUG). Patrick Kelly, the "director", has adopted a unique membership number scheme, which is (I suppose) designed to encourage a sense of camaraderie; I happen to be #277. His newsletter, The 1802 Peripheral, has only seen a few issues so far. I think a national newsletter/club may be much work for one person who is probably trying to hold down a full-time job at the same time. On the other hand, he has published more issues than we have.

There is an independent newsletter directed towards the owners of the VIP, called The VIPER. I have seen only one issue of this, so I cannot comment on its quality.

As far as I know, the only other newsletter of any significance directed to 1802 users is published by Quest Electronics' Cosmac Club. It is called Questdata. It is more professionally done than the independent club newsletters, I suppose because Quest subsidizes the publication. So far, the issues have had more in them than The Peripheral and less than Ipso Facto.

For more information, or to subscribe, write the editors directly:

Bernie Murphy, Ed.
IPSO FACTO ($10/yr)
102 McCrany Street
Oakville, Ontario
CANADA L6H 1H6

Patrick Kelly, Ed.
The 1802 Peripheral ($5/yr)
P.O. Box 7162
Los Angeles, CA
90022

Terry Laudereau, Ed.
The VIPER ($15/yr)
P.O. Box 43
Audubon, PA
19407

Bill Haslacher, Ed.
QUESTDATA ($12/yr)
P.O. Box 4430
Santa Clara, CA
95054

I hope all of you will support the ACE paper, since we cannot possibly do the same job here in this newsletter; however I hope we can achieve here the quality of the others as a minimum. But you can help with your contributions.

Bug Report

Yes, Virginia, there is a bug in Netronics Tiny Basic. And I was so proud of two years of distribution with no known bugs! Well, as the wise King Solomon once said, "Pride goeth before destruction, and a haughty spirit before a fall." Actually, the bug is in the character generator, not Tiny Basic itself. Perhaps next month I will go into the details of how it came about. For now, you need to know how to stop those pesky crashes.

Load in Tiny Basic, then jump to the monitor. If you load Tiny in one page up (i.e. start the cassette reading in at 0100), then your LBR to F000 in the first three bytes of memory will not be clobbered. Now make the following patches:

```
0002 01
0006 10
0007 D0
0C7A 9D
0C7B B0
0C92 EC
0C9A BC
0C9D AC
0CA3 5C
0CAC 5C
```

Notice that if you loaded Tiny in one page up, you will have to make your patches offset by one page also (i.e. at 0102 instead of 0002, and 0D92 instead of 0C92, etc.).

After you have made your patches, make a new cassette tape of Tiny (from the same locations you loaded it into). Use this new tape next time you want to use Tiny for anything.

If you are not using the CRT for display, you do not need to worry about this bug at all.

"Missle" Video Game
by Stewart Rauch III

This program produces a simple target-shoot video game. Memory locations 80 through FF are displayed on the screen as a 16x64 matrix. The top line (80 to 87) is continuously ring shifted to he right. Four targets (1x1 rectangles) placed on this line move across the screen in about seven seconds. When IN is depressed, a "missle" (also a 1x1 element) is launched from the middle of the bottom line of the screen, and rises to the top of the screen in about 1.8 seconds. If the missle hits (coincides with) a target when it reaches the top line, that target disappears. A new missle is launched every time IN is pressed. The object, of course, is to eliminate all four targets in as few shots as possible.

Before loading the program, it is easiest to zero the video field first by entering and running:

```
E2 F8FF A2 F800 B2 73 3007
```

Then load the program, turn load off, and run on. (Leave protect off). The targets will appear and firing can comence. When all the targets have been hit, the program can be restarted. When the program is reinitialized, new targets are added to the beginning of the target line, but old targets remain. By switching run off and on several times, a game with more than four targets can be played.

[Ed note: this program only works for 256-byte memory. The high ends of the registers are not initialized, so it will not work for extended memory as is. However, if you start with a little program like the following, you can set up the necessary registers before executing MISSLE:

```
(0100) F800 B1 B2 B3 B5 B7 BF F80F A1 C00003
```

You can load this program into any place in memory; I just used 0100 because it is available.]

```
Addr. Contents Label Instruction      Comment
 00 F80F       LDI  INT    Initialize pointers to
 02 A1         PLO  R1     Interrupt routine,
 03 F839       LDI  STACK
 05 A2         PLO  R2     stack,
 06 F83A       LDI  MAIN
 08 A3         PLO  R3     main program, and
 09 F866       LDI  SHIFT  target shift subroutine
 0B AF         PLO  RF     entry point.
 0C D3         SEP  R3     Go to main program
 0D 72         LDXA        Restore D
 0E 70         RET         Restore X,P, return
 0F 22    INT  DEC  R2     Entry pt of interrupt
 10 78         SAV         Save X,P
 11 22         DEC  R2
 12 52         STR  R2     Save D
 13 C4         NOP         Waste Time
 14 C4         NOP
 15 C4         NOP
 16 F800       LDI  00     Set DMA pointer
 18 B0         PHI  R0     to 0080, first
 19 F880       LDI  80     address of video data
 1B A0         PLO  R0     block
 1C 80    REFR GLO  R0
 1D E2         SEX  R2
 1E E2         SEX  R2     Refresh line
 1F 20         DEC  R0     7 times
 20 A0         PLO  R0     ...
 21 E2         SEX  R2     Refresh line
 22 20         DEC  R0     7 times
 23 A0         PLO  R0     ...
 24 E2         SEX  R2
 25 20         DEC  R0
```

```
26 A0          PLO R0
27 E2          SEX R2
28 20          DEC R0
29 A0          PLO R0
2A E2          SEX R2
2B 20          DEC R0
2C A0          PLO R0
2D E2          SEX R2
2E 20          DEC R0
2F A0          PLO R0
30 E2          SEX R2
31 20          DEC R0
32 A0          PLO R0
33 3C1C        BN1 REFR  If not done, repeat
35 300D        BR INT-2  If done, return
37 xx
38 xx
39 xx  STACK             Stack bottom
3A E2  MAIN    SEX R2    Make R2 data pointer
3B 69          INP 1     Turn on display
3C F880        LDI 80    Initialize the
3E A5          PLO R5    four
3F F8AA        LDI AA    targets
41 55          STR R5    ...
42 DF  WAIT1   SEP RF    Go to subroutine
43 3F42        BN4 WAIT1 If IN not pressed,
45 F8FB        LDI FB    Otherwise,
47 A7          PLO R7    initialize
48 F801        LDI 01    missle
4A 57          STR R7    ...
4B DF  MISL    SEP RF    Go to subroutine
4C F800        LDI 00    Shift missle up
4E 57          STR R7    one line,
4F 87          GLO R7    ...
50 FF08        SMI 08    making sure not
52 A7          PLO R7    to interfere with
53 07          LDN R7    targets
54 F901        ORI 01    ...
56 57          STR R7    ...
57 87          GLO R7
58 FF83        SMI 83    Is missle on top line?
5A 3A4B        BNZ MISL  No, keep shifting up
5C 07          LDN R7    Yes, delete
5D FAFE        ANI FE    any hit targets
5F 57          STR R7    ...
60 DF  WAIT2   SEP RF    Go to subroutine
61 3760        B4 WAIT2  Wait for IN released
63 3042        BR WAIT1  Wait for IN pressed
65 D3          SEP R3    Return
66 F887 SHIFT  LDI 87    Set DF to
68 A5          PLO R5    last bit of
69 05          LDN R5    target line
6A 76          SHR       (Ring Shift Right)
6B F880        LDI 80    Set pointer to
6D A5          PLO R5    beginning of line
6E F8FF REP    LDI FF    Initialize delay loop
70 A6          PLO R6
71 26  DELAY   DEC. R6   6.8ms Delay
72 86          GLO R6
73 3A71        BNZ DELAY
75 05          LDN R5    Shift byte pointed to
76 76          SHR
77 55          STR R5
78 15          INC R5    Point to next byte
79 85          GLO R5    Check for end of line
7A FB88        XRI 88
7C 3265        BZ SHIFT-1 If done, return
7E 306E        BR REP    If not, repeat
80 00                    Initially this half
.. ..                    of memory is zero.
FF 00                    (video data block)
```

Penny Bandit

REQUIRES 2 4K BOARDS

We do not yet have any contributions in the Tiny Basic department, so I worked up a simple program to simulate a slot machine. I know you can do better than this.

```
10 PRINT "PENNY BANDIT"
20 M=100
100 PRINT "YOU HAVE $";M/100;".";
110 IF M-M/100*100<10 THEN PRINT 0;
120 PRINT M-M/100*100
150 PRINT "PULL HOW HARD";
160 INPUT H
170 IF H<1 THEN GOTO 2100
180 IF H>9 THEN GOTO 2000
200 I=3
210 S=0
220 GOSUB RND(8)*100+1000
230 LET I=I-1
240 IF I>0 THEN GOTO 220
250 PRINT
260 H=H-1
270 IF H>0 THEN GOTO 200
300 IF S=1 THEN H=1
310 IF S=2 THEN H=5
320 IF S=3 THEN H=250
330 IF H>0 PRINT "YOU WIN ";H;" CENTS."
340 M=M+H-1+(H+499)/500
350 IF M>0 THEN GOTO 100
360 PRINT "YOU WENT BUST."
370 END
1000 PRINT "BAR  ";
1010 S=S+1
1020 RETURN
1100 PRINT "BELL ";
1110 RETURN
1200 PRINT "LIME ";
1210 RETURN
1300 PRINT "HORN ";
1310 RETURN
1400 PRINT "DOG  ";
1410 RETURN
1500 PRINT "NUTS ";
1510 RETURN
```

```
1600 PRINT "CAKE ";
1610 RETURN
1700 PRINT "BIKE ";
1710 RETURN
2000 PRINT "NOT SO HARD!"
2010 GOTO 150
2100 PRINT "QUITTER!"
2200 END
```

I leave it to you to figure out the house odds.

The ELF Workshop

One thing I would like to do on a continuing basis is to develop some larger program. You, the readers could also contribute to this program, so that it would benefit from the collective intelligence of all of us. Naturally, to succeed in this kind of group–think we need a project that is interesting to many people. Some ideas I have are,

1. Personal accounting. We might do a whole series of programs to balance your checkbook, plan your budget, figure your income tax (probably too late for this year, but it could be ready in time for next year's taxes), and so on. We could do it in Tiny Basic, or we could define a new interpreter especially suited to this application: the 1802 is particularly good at that.

2. Adventure games. We could develop some super game, perhaps like Star Trek, in which you act out some fantasy, under the general supervision of the computer. With graphics it ought to be spectacular. Maybe something like spacewar for two people. Alternately, you could set it in Middle Earth, with Hobbits, elves, trolls, wizzards, dragons, and the like, to aid or hinder your quest. I think this would have to be done in machine language or some semi–interpreter, but we can try out the ideas in Tiny.

3. Anybody for computer chess? Who knows, if we all pooled our skills, the result might win against Boris. But even if not, we can put the board on the TV screen and have a lot of fun.

4. How about a robot? The IEEE sponsored a robot mouse contest last year. I think an 1802 is an ideal CPU to build into a small robot. Whether we want to try for a mouse in next year's contest (if there is one), or just a robot pet to roam around the house, the project ought to be very educational.

5. Then there is voice input and output. This is not quite so "state of the arty", since there exist commercial packages to do those kinds of things, but those packages do not run on your ELF. Perhaps we can do some innovation.

6. Music is always a possibility. The computer music buffs claim that fast computation is a necessary requirement, so perhaps this is less exciting. But you decide.

7. On a less grand scale, we could try to define a physical system controller, for doing things like controlling your house (watering the lawns, turning the heat on and off, opening the garage door, etc.), or supervising a model railroad, or operating a machine in a factory. I think the same kinds of controls are needed in each application.

Perhaps you have some better ideas. Let me hear from you. I will hold off on getting started until I know which way you want to go. Remember, we will try to do something that will really work when we finish! Along the way there should be plenty to learn for both beginners and the more experienced.

NETRONICS RESEARCH
AND DEVELOPMENT LIMITED
333 LITCHFIELD ROAD (RTE. 202)
NEW MILFORD, CONNECTICUT 06776

NEW PRODUCTS
Text Editor
Assembler
Disassembler
Video Display Board

PRINTED MATTER