TABLE OF CONTENTS

Supporting a Language ...................................... 2
Parallel Print Driver & Listing ............................... 3
Disk Drive Motor Control .................................... 5
Jumpering the Wild Shugart ................................. 6
More Power Supplies ........................................ 7
Direct Input Routine & Listing ............................... 7
Program Storage Above PFM & Listing ...................... 8

REGULAR FEATURES

Editorial ....................................................... 1
Letters .......................................................... 2
Notes from Garland .......................................... 4
Something New .............................................. 7
Here We Go Again!

Exclusive!
What happens when a Xerox copies a Big Board? Why you get a "Worm", of course! That's right! The Xerox 820 is just a Big Board in disguise.

My informed sources say that last fall Xerox bought non-exclusive rights to manufacture a system based on the Big Board. Xerox re-laid out the board (4 layers) so that it would fit in the cabinet, they dedicated the SIO port B as a printer port, and they set up the disk interface (1771) to handle either 5 or 8 inch. Otherwise, it appears to be all Big Board, right down to the 2.5 MHz clock. The system PIO does the same things on both systems, bit for bit, according to Xerox's documentation.

Xerox had 50,000 orders in hand the day they shipped the first 820, and they expect to recoup all their startup costs by the end of this calendar year. What a market for software and hardware developed around the Big Board. I'll say more about the 820 as information comes in. (I'd give my eye teeth to see a schematic and service manual for the 820.)

Picnic
We had a Saturday noon picnic to celebrate our first issue. It turned out that the Saturday we picked conflicted with every party/birthday/outing/etc. for three states around. But Sandy and I and those who came had six hours of very interesting and mellow conversation.

The knowledge, resources, and excitement among the local group members are terrific. I only wish all of you could have joined us.

The First Issue
Despite the speed of the U.S. Snail, a heartening number of readers have actually received issue no. 1. The responses from these lucky folks have made the daily trip out to our mailbox most enjoyable. The comments have included; 'surprised, happy, delighted'.

Though Micro C is a long way from being a success financially, feedback like this tells us that it is successful in other ways. We like doing it and we really appreciate your response.

Sometimes a dream generates momentum of its own. This one has.

Thanks.

David Thompson
Editor & Publisher
Dear Sir,  
July came and July went by, and my mailbox has completely rusted out due to all that drooling.  
Silly me! When I read 'Issue No. 1 will hit the streets during July' I assumed it was July 1981! But now I realize you meant July 1982. I'd better get a stainless steel mailbox or maybe not bother to wait, because the magazine will never get here.  
Maybe it went the way of Mitt's Newsletter, the Digital Group Newsletter, and Processor Technology's "Access."  
I hope not.  
Joe Kish  
758 Yucca Ridge Lane  
San Marcos, CA 92069

Editor's note:  
I called Joe; after all it was the least I could do for his mailbox. And besides, I think it's a great letter! (He did finally receive issue no. 1.)

Sandy and I made a desperate, last ditch effort to get all 500 first issues collated, bound, labeled, sorted and bundled in one afternoon so we could get the first issue in the mail on July 31. We missed the 8 PM deadline at the post office by 15 minutes.  
So the magazine was mailed Monday morning, August 3rd. (So much for hitting the streets in July.)

Someday maybe I'll write a book about starting a users group magazine. I could almost write the book about the first issue, and Murphy would certainly be a leading figure. (For those of you who don't know Murphy, he is the one credited with the first voyage of the Titanic.)

Quote from Murphy:  
If there is no way your plan can fail, you simply don't have all the information.

Dear Editor,  
I bought a bare board version and built it up from scratch. I had to buy about $80.00 worth of parts beyond what I had around. I have it up and running CP/M and am currently working on packaging it in a terminal-type case with a Ball Brothers CRT. The unit is going to be used for text processing and formatting for a friend's photo typesetter. My other computer is an LSI-11 and I also use my H19 with the DEC-20 at work. I think the Big Board is an excellent value and very useful.  
I agree that Frank Gentges' idea about the parallel ports is excellent. That would take care of most of the board's limitations. I think your publication has already been worth the price and I suspect that an active users group with a publication will enhance the usefulness of the hardware significantly.

Doug Faunt  
PO Box 11142A  
Palo Alto CA 94306

Dear David,  
CONGRATULATIONS!!! FANTASTIC!!! You really made it. It looks great and reads great. You are certainly to be congratulated for undertaking such a task that should be helpful to so many.  
I hate to mention that Momma and I are just back from five weeks vacation in the Smokey Mountains in Tennessee. I am about ready to get my feet on the ground again. I hope that I can get back on track to help keep the pipe full of articles for future issues.

Don Retzlaff  
6435 Northwood  
Dallas TX 75225

Editor's note,  
What can I say? Thanks again Don, without you and John Jones and Andrew Beck, and the rest of you who are writing up things for future issues this wouldn't be possible.  
(As for the five whole weeks in the Smokey Mountains, that's just not fair.)

---

Letters continued

Supporting A Language

By David Thompson

Throughout these early months of Micro Cornucopia, I have been looking at commercial and public versions of various languages with the hope of finding a semi-official language for this group.

A common high level language would mean we could pass around source code in something other than assembler. But the language would need to be powerful enough for substantial commercial applications and inexpensive enough that most of the people in the group could afford it.

Letters continued

Plus, it would need to produce fast and compact object code, encourage readable source code, and promote structured programming. (Whew!)  
I am looking seriously at three languages: Forth, Pascal, and C. Of these three, C is presently leading. One reason is that all the versions I have seen have been upwardly compatible with Bell Lab's C.

Versions of C that I'm aware of:  
Small C (Public)  
Small C+ (Public)  
Tiny C ($100)  
CW/C ($75)  
BDSC ($145)  
Supersoft C ($200)  
Whitesmith's C ($600)

(The prices are approximate.)  
Whitesmith's C is a full blown version of the language. In fact, sources tell me that it was created by three fellows who worked on C for Bell Labs. They left Bell in order to develop and market C for the business and scientific community.

I've heard that BDSC is a competent enough subset to be an option for someone writing commercial applications. It has its own users group and publication. All this for $145, such a deal. (Lifeboat is offering discounts on quantity purchases of BDSC.)  
CW/C is an expanded version of Small C with lots of nice utilities, but I don't know if it is ready to do commercial work. However, it still looks like quite a bargain at $75.

Tiny C is the only interpreter in the bunch. It also comes in compiler form for about $300. The only thing I have heard about Tiny C is that it has an excellent manual (and I heard that fourth or fifth hand).

Supersoft's C is new on the market. The ads say that they support 'most' of version 7 Unix. If that includes floating point and pointer arithmetic, then it would be a very credible piece of software, assuming they have taken time to exorcise bugs.

The standard text on C is:  
"The C Programming Language" by Kernighan and Ritchie  
Prentice-Hall
Parallel Print Driver

By John P. Jones
5826 Southwest Ave.
St. Louis, MO 63139

This is a simple parallel printer driver that can be incorporated into any CP/M BIOS.

On first entry, the program initializes PIO port B and the interrupt vector register. The program also modifies the BIOS jump table so that all subsequent calls for list output bypass the initialization routine.

As each character is output to port B, a flag byte is set, indicating that the printer is busy. When the printer is again ready, the PIO does an interrupt. The sole purpose of the interrupt routine is to reset the 'printer busy' flag. The character output routine tests the flag byte and loops until it is reset. When the flag is reset, a character is sent and the flag is again set.

---

ADS

If you want millions to know what you’re doing, buy a page in Byte.

However, if you:
- need help designing a commercial product
- can provide help on a consulting basis
- need to find a source of...
- want to sell that new BB peripheral we’ve all been waiting for

Well then, how about an ad in Micro C?

Space Ads
People laugh when we tell them what our space rates are. They stop laughing when they realize that a 1/3 page ad costs about as much as a sack of groceries.

If you are interested in one of our grocery ads or in something larger or smaller, call or write. We’ll send a rate card and complete details. The advertising deadline is October 15 for issue no. 3, and December 15 for issue no. 4.

Want Ads
For a modest 20 cents per word, you could become famous on a budget. (Please include payment with ad.) Where else could you say WORLD’S GREATEST PROGRAMMER

503-645-3253

for only 80 cents?

So write it down just the way you’d like to see it. Don’t abbrev the pr thing to deth. List the price if possible and any expected shipping delay.

Write or call the editorial office for information.
Notes From Garland, Texas

By David Thompson

Clearing up the screen.

The clear-to-end-of-screen command is CONTROL Q, not CONTROL W as indicated in the documentation.

Bringing up stubborn boards.

A number of people have been contacting Jim and me about problems they are having bringing up boards. One of the most common symptoms is a pattern of two characters on the screen or a screenful of random garbage. Either way, it basically means that the board probably didn’t finish loading the PFM monitor in RAM so it could try to clear the screen.

Jim is going to put together information about what they look for when they troubleshoot boards. Hopefully, I will have that in time for the next issue.

Don’t forget the 90 day guarantee which completely covers defective parts and boards. Plus, he has been doing out-of-warranty or pilot error repairs very reasonably. Most of the time these charges have been between $25 and $50. The maximum so far has been $75 (the board had to be almost completely resoldered, among other things). That’s pretty hard to beat.

Two CP/Ms

I have noticed that some software which runs on one Big Board system will not necessarily run on another. I also noticed that there are two different IDs when CP/M boots.

I called Jim about this and he said that those folks who used the BIOS he sent out with the boards and who didn’t do their own incorporation into CP/M have a version which origins the BIOS at EA00. All the folks who bought CP/M already modified for the Big Board have a BIOS starting at E800. The difference has led to some problems with software which depends on having BIOS in a certain place.

Jim said the ready-to-run version has BIOS shifted down 200H because they thought they needed room to store 256 bytes (a double-density sector) in high memory. Then the data could be moved into low memory in 128 byte chunks and accessed. Jim isn’t sure whether there is going to be a use for this space but he is concerned that we maintain consistency.

According to Jim, it’s easy to make the EA00 BIOS into an E800 BIOS.

Original—RES.(MSIZE-20)*1024
New—RES.(MSIZE-20)*1024-200

Now reassemble the mess and you too can ORG at E800.

By the way, a pretty reliable way to tell which version you have is to look at the ID that’s displayed when you boot CP/M. If it just says “60k CP/M version 2.2” then you probably ORG at EA00. If the prompt includes the words “BIG BOARD” then you already ORG at E800.

The separate BIOS (and monitor etc.) disk Jim is shipping with orders now ORGs at E800. If you would like the latest version rather than reassembling BIOS with the modification above, send Jim a disk and $3.00 for shipping.

4 MHz (Again).

This is an updated version of the 4 MHz mod printed in issue no. 1. This version reportedly does not require special ram. Jim says he has 300ns 4116 working consistently using this mod. The only difference between this one and the previous one is that the CAS and MUXC lines are each moved left one pin on U76 (shift register) so that they change states 50ns earlier. This change means that the system meets the precharge requirements for the slower RAM.

4 MHz Mod Version 2

1. Cut the trace (bottom of the board) to U76 pin 4.
2. Connect the cut trace (MUXC) to U76 pin 3.
3. Cut the trace (bottom of the board) to U76 pin 5.
4. Connect the cut trace (CAS) to U76 pin 4.
5. Remove U96.
6. Connect U97 pin 4 to U96 pin 4.
7. Don’t replace U96.

(continued next page)
**Disk Drive Motor Control**

By David Thompson

---

**Disk AC Control Circuit.**

*If you're tired of listening to your disk drives grind on hour after hour, here's relief.*

The board must have the timer option installed and you must jumper pin 3 to pin 4 and pin 7 to pin 8 on JB2. This supplies the one second interrupt to the Z80. If the Z80 counts all the way to 30 after the most recent disk access then it sends a command to the system PIO to drive the output of U112 pin 2 low.

Terminal 7 on the Big Board power connector is tied to U112 pin 2. This terminal is high (about 4V) when the system is doing a disk access and goes low if there hasn't been any access for 30 seconds.

Simply connect the output of an optically isolated solid state relay between terminal 7 and ground. Then connect the output in series with the AC to the disk drive motors. (But do not connect in series with the drives' DC supply.)

*No UPS to a PO Box?*

Jim Tanner lists his mailing address as a PO Box but he also has a street address that works for both the post office and United Parcel Service. (The ZIP is different.)

**Editor's note:**

To change the baud rate, create F.COM as follows:

1. Enter "DDT" "<CR>"
2. Enter "A100" "<CR>"
3. Enter "MV1,XX" "<CR>"
4. Enter "OUT OC" "<CR>"
5. Enter "JMP 0" "<CR>"
6. Enter "<CR>"
7. Enter "G00" "<CR>"
8. Enter "SAVE 1 F.COM" "<CR>"

This routine sends a single byte (XX) to the channel B baud rate generator. I am working at 9600 baud so I replace XX with 0E. See the Big Board Theory of Operation for other baud rates.

Once you have completed the baud rate program, simply enter "F" "<CR>" from the CP/M prompt to set the baud rate.

**Dis/ Drive Motor Control**

This modification redirects the list device output to serial port B. The default data rate is 300 baud. This patch does not force the Big Board to poll any of the handshake lines on port B. Thus, it has no way of knowing if the printer buffer is full. (May or may not be a problem.) This modification is for those who

---

**CP/M patch for serial printer port.**

This CP/M modification redirects the list device output to serial port B. The default data rate is 300 baud. This patch does not force the Big Board to poll any of the handshake lines on port B. Thus, it has no way of knowing if the printer buffer is full. (May or may not be a problem.) This modification is for those who

---

**This modification redirects the list device output to serial port B.**

The default data rate is 300 baud. This patch does not force the Big Board to poll any of the handshake lines on port B. Thus, it has no way of knowing if the printer buffer is full. (May or may not be a problem.) This modification is for those who

---

**Editor's note:**

To change the baud rate, create F.COM as follows:

1. Enter "DDT" "<CR>"
2. Enter "A100" "<CR>"
3. Enter "MV1,XX" "<CR>"
4. Enter "OUT OC" "<CR>"
5. Enter "JMP 0" "<CR>"
6. Enter "<CR>"
7. Enter "G00" "<CR>"
8. Enter "SAVE 1 F.COM" "<CR>"

This routine sends a single byte (XX) to the channel B baud rate generator. I am working at 9600 baud so I replace XX with 0E. See the Big Board Theory of Operation for other baud rates.

Once you have completed the baud rate program, simply enter "F" "<CR>" from the CP/M prompt to set the baud rate.

---

**No UPS to a PO Box?**

Jim Tanner lists his mailing address as a PO Box but he also has a street address that works for both the post office and United Parcel Service. (The ZIP is different.)

**Jim Tanner**

Digital Research Computers
2702 Industrial Lane
Suite J2
Garland, Texas 75041
Phone 214-271-3538

---

**Micro Cornucopia, Number 2, September 1981**

---
Shugart set a new standard for obscurity when they came out with their SA 801 user's manual. It's not that they don't tell you how to jumper their drives, the only problem is figuring out what they told you. Once you figure it out, don't go back and look at the manual, you'll just get confused again. So on that note, here's what I figured out.

For drive A, jumper only the following: DC, C, DS1 (Drive Select 1), T2, T3, T4, T5, T6, HL, A, B, T1, 800, Y.

For drive B, change DS1 to DS2.

For drive C, change DS1 to DS3, and so on.

For the last 9 months or so, Shugart has been shipping drives with a new circuit board. The new board is completely interchangeable with the old one, but the new one does not use the -5/-15V pin on the DC supply jack (OS). The pin is there but is not connected to anything because the new board does not need -5V.

One way to tell whether you have a new or old style drive is to check the bottom left hand corner on the circuit board. The old drive has a -5V regulator there. On the new one, that corner is pretty empty. Also, the resistance from the -5V pin to ground is infinite on the new boards.

I had one of the new boards but the old documentation so I spent a couple of 'interesting' evenings trying to make sure the -12V I was supplying would be properly turned into -5V on the board. (Oh well, if everyone's documentation were perfect there probably wouldn't be so much need for user groups.)

Note: The following information is from Bill Klevesahl, Shugart's product manager for the SA 800 series.

Test points for both boards.

1,2 Amplified read signal
5,6,7 Ground
10 -Index
11 +Head Load
12 -Index and Sector Pulses
16 +Read Data
25 +Write Protect
26 +Detect Track 0
27 +Step Pulse

Test points on the old board only.
3,4 Differential Read Signal (this signal is now hidden inside the new LSI read chip).
21,24 -Data Separator Timing (there is no longer a pot to adjust this).

Test points on the new board only.
8 +Data Window (for checking FM data separation).

Optional features on the new board.
- Add-trace option TS enables true FM data separation, maintaining synchronization during address marks.
- Add-trace option NFO prevents the head from being forced out past track 0.

BUG

The formatting program listed in issue 1 contains a bug. If the program has a problem accessing a disk in drive B, it reformats the disk in the default drive (A).

Issue 3 will include a revised format program.

Coming Up

Articles you'll be seeing in the future.
- Reverse video cursor
- 5 inch disk interface
- Real time clock routine
- Converting a TV into a real video monitor
- More on the PFM monitor
- Review of 3 assembly language texts
- Bios modifications

Articles we'd love to see.
- Trials and tribulations of bringing up a Big Board
- How you've improved the PFM monitor
- Hard disk interface
- Filling out the second bank with system RAM
- DMA interface
- Double density disk interface
- A graphics display
- A speech generator
- A simple ROM burner
- Interfacing with particular printers etc.
- An in-depth series on CP/M
- Reviews of FIG Forth and Forth 79
- Reviews of BDSC, Whitestock's C, CW/C and Supersoft's C
- Computer consulting using a Big Board
- Reviews on peripherals, keyboard, video monitor, power supply, cabinet, disks, etc.
- Other software reviews. Even if you are just borrowing a copy to evaluate, please let us know how you like it.
- Book reviews

If you are immersed in any of these projects, please share your experience with all of us.
Direct Input Routine

By Andrew P. Beck

Assembly Listing

```
F800  E5  SUBR  PUSH HL  ;SAVE ADDRESS OF HL%
F801  CD06F0  CALL KBDST  ;GET KBD STATUS
F804  B7  OR A  ;IF A=0 DATA AVAILABLE
F805  CA0EF8  JP Z ISDATA  ;JP TO DATA SAVE ROUTINE
F808  E1  POP HL  ;GET ADDRESS BACK
F809  3C  INC A  ;IF A=FF IS NO DATA, MAKE IT 0
F80A  77  LD (HL),A  ;STORE 0 IN HL%
F80B  23  LD (HL),A  ;DD BOTH BYTES
F80C  77  LD (HL),A  ;DD BOTH BYTES
F80D  C9  RET  ;RETURN WITH HL% 0

F80E  CD09F0  ISDATA  CALL KBDIN  ;GET INPUT CHAR INTO A
F811  E1  POP HL  ;GET ADDRESS OF HL% BACK
F812  77  LD (HL),A  ;STORE DATA, LOW ORDER
F813  23  INC HL
F814  3600  LD (HL),0  ;HIGH ORDER = 0
F816  C9  RET  ;RETURN TO BASIC
```

--- Poke the above program into F800+ ---

```
500  SUBR = &HF800
510  DATA &HES,&HCD,&H06,&HF0,&HB7,&HCA,&HOE,&HF8
520  DATA &HE1,&H3C,&H77,&H23,&H77,&HC9,&HCD,&H09,&HF0
530  DATA &HE1,&H77,&H23,&H36,&H00,&HC9
540  FOR I=0 TO 22
550  READ INST
560  POKE SUBR+I,INST
570  NEXT

--- Demonstration routine ---

```
580  HL%=0
590  CALL SUBR (HL%)  ;RETURN WITH HL% = 0
600  IF HL%=0 GOTO 590
610  IF HL%=3 THEN STOP
620  PRINT CHR$(HL%);  ;RETURN TO BASIC
630  GOTO 590
```

This routine makes it possible to do direct input with Microsoft basic. First, a machine language subroutine is poked into an unused area of the system monitor. This subroutine calls the monitor subroutine and the monitor checks to see if an input character is available. If none is available, the HL% is set to zero. If a character is available, it is stored in HL% before a return is executed.

In the demonstration program, a returned character is echoed on the console. If the character is 'C, the demonstration stops.

More Power Supplies

By David Thompson

I just received a catalog from ACDC Electronics and they list a power supply that should power the Big Board and a couple of drives. (Like the Power One, you still have to finesse +12V but that isn’t hard, see Issue no. 1.)

Model ETV801 provides:
- +5V at 9 amps
- 12V at 0.8 amps
- +24V at 4.5 amps peak
Price is $132 (list, single).

They don’t mention how they handle over-current protection, but they do indicate that they only have over-voltage protection on the +5V line unless you specify the -1 option. They don’t say how much extra you pay for the option.

ACDC Electronics
401 Jones Rd
Oceanside, CA 92054

Power/Mate also has an open frame linear with the same specifications as the ACDC model above, but the PowerMate model ED-132AV lists for $120 (single).

Power/Mate
514 S River St
Hackensack, NJ 07601

Something New

DataCast
345 Swett Road
Woodside, CA 94062

I just received issue no. 1 of DataCast and I’m impressed, very impressed. This is a bimonthly magazine for ‘major micro systems and telecommunications,’ ‘Major micro systems’ means CP/M in a business or OEM environment and ‘telecommunications’ means networking.

Jim Warren, guiding force behind the West Coast Computer Faire, is behind this magazine and I suspect it will be around for a long while. Subscriptions are $18 per year (6 issues).

He is starting with a staff of 19 (if you include the mascot, Sir Lick-A-Lot) and it shows. The first issue is 64 pages and about 60 pages of that is copy.

Some first issue articles:
- What is Telidon and Why is AT&T Adopting It?
- Overview of Home Information Services
- A Seminar for Independent CP/M Software Vendors
- Software Documentation Protocols
- An Index to CP/M Software and Vendors

Other Interesting Periodicals

Dr. Dobb’s Journal
PO Box E
Menlo Park, CA 94025

Lifelines
1651 Third Ave
New York, NY 10028

Please let us know about your favorite magazines.

There are numerous times when you want to write a small assembly language program to use as a printer driver or other routine. These small utilities need to reside in high memory so they can operate at the same time as routines which reside in the normal transient program area (starting at 0100H).

Since programs are loaded starting at 0100H, these utilities must load themselves into high memory.

There is a considerable amount of memory available above PFM that is not dedicated to any other use. PFM version 3.3 uses upper memory starting at F000H through F7E6H. The RAM area FF00H through FFCH is used for data storage. This leaves the memory from F7E7H through FEFH and FFC9H through FFFFH available for your use. Not all of this space is really available since future releases of PFM could use some of this space.

I recommend that you limit your programs to the following areas: (FA00H through FEFFH and FFE0H through FFFFH).

Moving the program up

In order for your routine to start out as a normal COM file but wind up in upper memory, it has to do a quick shuffle.

1. When the COM file is executed it is loaded into memory starting at 0100H.
2. Execution starts at 0100H.
3. The first few statements (starting at 0100H) must copy the routine into upper memory.
4. An initialization routine may then be executed.
5. Control is then transferred to the routine or back to PFM.

In order to accomplish all of the above it is necessary to do the following:

1. Write your assembly language routine as follows:
   a. The origin is set at the desired point where your routine is to reside.
   b. Your program must start with a short move routine.

c. An initialize routine usually follows that patches (hooks) your routine into the monitor or PFM.
d. Your routine follows.
e. The last statement defines the length of the program.

2. Assemble your program.
3. Execute DDT and load your HEX file into memory. Typically this is done as follows:

   >A.DDT NAME.HEX

This will load your program into memory at the desired location (example EA00H). The program will not execute.

DDT will print out starting and ending addresses.

NEXT PC/n FAxx FA00

4. Using DDT, move the program from upper memory to 0100H.

MFA00, FAxx, 0100

5. Transfer control back to PFM by typing:

   GO

6. Save the program using the SAVE command.

SAVE 1 NAME.COM

You must save the program in 256 byte blocks. Using '1' will save 256 bytes, '2' would save 512 bytes, etc.

7. The program is now ready for execution as a COM file.

The above procedure may seem long and rather involved but after you have done it a few times you will find it very quick and simple.

---

**Program Storage Above PFM**

By Don Retzlaff

6435 Northwood

Dallas, TX 75225

---

**PFM Monitor Listing (continued from issue no. 1)**
PFM Monitor Listing (continued)

<table>
<thead>
<tr>
<th>Location</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0999</td>
<td>3A30FF</td>
</tr>
<tr>
<td>099A</td>
<td>LD A, (FIFCNT) ; GET INPUT FIFO BYTECOUN</td>
</tr>
</tbody>
</table>
Micro Cornucopia, Number 2, September 1981

(continued on top of page 10)
PFM Monitor Listing (continued)

F55C CD8B 1151 RES 7,A ;SWITCH BACK LOWER 16K OF RAM
F55E D31C 1152 OUT (BITDAT),A ;INTERRUPTS ARE SAFE AGAIN
F560 FB 1153 EI
F561 C1 1154 POP BC
F562 D1 1155 POP DE
F563 E1 1156 POP HL
F564 C9 1157 RET
F565 1158:
F566 1159:
F567 1160:
F568 1161 DUTCH: LD D_E,LEADING
F569 B7 1162 LD A,(DE) ;GET LEAD-IN STATE
F56A C3270F6 1163 JP NZ,MULT ;JUMP IF IN A LEAD-IN
F56B 79 1164 LD A,C ;ELSE PROCESS CHARACTER IN C
F56C 73 1165 INC HL ;AND ADV POINTER TO NEXT COLUMN
F56D 7D 1166 LD A,L ;EXTRACT COLUMN# FROM HL
F56E FE20 1167 CP C,CONTROL-# ;JUMP IF A CONTROL CHARACTER
F570 380F 1168 JR C,CONTROL-# ;JUMP IF A CONTROL CHARACTER
F572 71 1169 INB HL ;ELSE STORE DISPLACABLE CHARACTER
F573 23 1170 LD A,C \* ;AND ADV POINTER TO NEXT COLUMN
F574 7D 1171 LD A,L ;EXTRACT COLUMN# FROM HL
F575 E67F 1172 AND 01111111B ;GET COLUMN# FROM HL
F577 FE50 1173 CP BO ;EXIT IF NOT PAST COLUMN 79
F579 DB 1174 CALL CDE7FS ;CALL RETURN ;ELSE DO AUTOMATIC <CR>
F57C CD42F6 1175 CALL LFEED ;AND LINEFEED
F580 C9 1176 RET
F581 E5 1177
F582 210FF5 1178 LD BC,CTLTAB ;SEARCH FOR CONTROL CHARACTER
F583 010000 1179 LD BC,CCTL253/3 ;HANDLING SUBROUTINE IN TABLE
F584 C640F3 1180 CALL SEARCH
F58B E1 1184 POP HL
F58C C0 1185 RET NZ
F58D C5 1186 PUSH BC
F58E C9 1187 RET
F5BE 1188
F5BF 1F 1189 CTLTAB: DEFB '~ -64
F590 1E 1190 DEFB '=' -64
F591 1B 1191 DEFB '* -64
F592 1A 1192 DEFB 'Z -64
F593 18 1193 DEFB 'X -64
F594 11 1194 DEFB 'Q -64
F595 0D 1195 DEFB 'M -64
F596 OC 1196 DEFB 'L -64
F597 OB 1197 DEFB 'K -64
F598 OA 1198 DEFB ' J -64
F599 09 1199 DEFB 'I -64
F59A 08 1200 DEFB 'H -64
F59B 07 1201 DEFB 'G -64
F59C DCF5 1202
F59D DCF5 1203 DEFW .BELL ;CTL-G IS THE BELL
F59E BE8F 1204 DEFW BAKSPC ;CTL-# IS CURSOR LEFT
F59F C5F5 1205 DEFW TAB ;CTL-# IS CURSOR LEFT
F5A0 C2F6 1206 DEFW LFEED ;CTL-J IS CURSOR DOWN
F5A4 2CF6 1207 DEFW UCSR ;CTL-J IS CURSOR UP
F5A6 C4F5 1208 DEFW FORSPC ;CTL-L IS CURSOR RIGHT

F5E5 1273 PUSH HL
F5F0 1274 LD DE,CRTMEM+1
F5F3 1275 LD BC,24#2128
F5F4 1276 LD (HL),'
F5F8 1277 LDIR ;FILL CRT MEMORY WITH SPACES
F5FA E1 1278 POP HL ;POINT TO HOME CURSOR POSITION
F5FE 1279 LD A,23
F5FD 1280 LD (BASE),A ;MAKE BASE LINE# 23 AND
F600 D314 1281 OUT (SCROLL),A ;STORE IN SCROLL REGISTER
F602 C9 1282 RET
F603 E5 1284 :CRTCREDL: PUSH HL ;SAVE CURSOR POINTER
F604 7D 1286 LD A,L
F605 E7F 1287 AND 01111111B ;GET COLUMN# COMPONENT OF
F607 4F 1288 LD C,A ;CURSOR POINTER INTO C
F608 3E50 1289 LD A,B0 ;CALCULATE HOW MANY CHARS
F60A 91 1290 SUB C ;REMAIN ON CURRENT LINE
F60B 47 1291 LD B,A
F60C CD66F6 1292 CALL CLR ;CLEAR REST OF LINE & HL
F60E E1 1293 POP HL
F610 C9 1294 RET
F612 1295:
F614 E5 1297 CRTCREDL: CALL CRTCREDL :CLEAR REMAINDER OF CURRENT ROW
F615 3A77FF 1298 CRT ;COPY BASE SCREEN ROW# TO C
F618 17 1299 LD C,A
F619 7D 1300 CLRS1: LD A,L
F61A 17 1301 CLRS1: LD A,L
F61B 7C 1302 CLRS1: LD A,L
F61C 17 1303 CLRS1: LD A,L
F61D E61F 1305 AND 00001111B 
F61F B9 1306 CP C ;SEE IF HL IS AT BOTTOM ROW
F620 280B 1307 JR Z,CLRS2- ;AND LEAVE CLEAR LOOP IF SO
F622 CD37F6 1308 CALL DCSR ;ELSE POINT HL TO NEXT ROW DOWN
F625 1260F6 1309 CALL CLRS1- ;AND FILL THAT LINE WITH SPACE
F626 1BEF 1310 JR CLRS1- ;
F62A E1 1312 CLRS2: POP HL ;RESTORE CURSOR POINTER
F62B C9 1313 RET
F62C 1180FF 1314 :UCSR: LD DE,-128 ;SUBTRACT 1 FROM ROW# COMPONENT
F62F 19 1317 ADD HL,DE ;OF CURSOR POINTER IN HL
F630 7C 1318 LD A,H
F631 3E30 1319 CP CTRTBAS ;CHECK FOR UNDERFLOW OF POINTER
F633 0D 1320 RET NC
F634 263B 1321 LD H,CRTTOP-1 ;WRAP CURSOR AROUND MODULO 3K
F636 C9 1322 RET
F637 118000 1323 :DNCSR: LD DE,-128 ;ADD 1 TO ROW# COMPONENT
F63A 17 1324 ADD HL,DE ;OF CURSOR POINTER IN HL
F63B 7C 1325 LD A,H
F63C FE3C 1326 CP CTRTPO ;CHECK FOR OVERFLOW OF POINTER
F63E DB 1327 ADD C
F63F 2630 132E LD A,H
F641 C9 1331 RET
F642 7D 1332 LD A,L
F643 17 1333 RLA
F644 7C 1337 LD A,H
PFM Monitor Listing (continued)

FD61 79 1402 SETCOL: LD A,C ;ARRIVE HERE ON FOURTH CHAR
FD62 D620 1403 SUB * ;OF EBC,*,",ROW,COL SEQUENCE
FD64 D650 1404 SETC2: SUB B0
FD66 30FC 1405 JR NC,SETC2- ;MAKE BURE COL# BETWEEN 0 & 79
FD68 C500 1406 ADD A,B0
FD6A B5 1407 OR L
FD6B 6F 1408 LD L,A ;MERGE IN COL# WITH L
FD6C 99 1410 RET
FD6D CD72F5 1411 M4ST: CALL DISPLAY ;DISPLAY THE CONTROL CHAR
FD6E C9 1412 RET ;PASSED IN C
FD6F 1413
FD70 1414 ;INCLUDE DISKIO.ASM
FD71 1415
FD72 1416
FD73 1417 ;*******************************
FD74 1418 ;DISK INPUT/OUTPUT DRIVER SUBROUTINE PACKAGE
FD75 1419 ;FOR WESTERN DIGITAL 1771 DISK CONTROLLER
FD76 1420
FD77 1421
FD78 1422 ;bullet-proof error recovery added 12-APR-80
FD79 1423
FD7A 1424
FD7B 1425
FD7C 1426
FD7D 1427
FD7E 1428 ;EQUATES FOR DISK CONTROLLER PORTS AND COMMAND CODES
FD7F 1429

---

F71F CDAFB7 1524 WRITE: CALL READY ;CLEAR THE DISK CONTROLLER
F720 C0 1525 RET NZ ;CLEAR IF DRIVE NOT READY
F722 CB77 1526 BIT $ ;WRITE DISK
F723 C0 1527 RET NZ
F725 06AB 1528 LD B,WRTCMD ;EXIT IF DISK WRITE-PROTECTED
F726 1806 1530 JR RDRT- ;
F727 1450
F728 1451
F729 1452
F72A 1453
F72B B81C 1454 IN A,(BITDAT)
F72C 47 1455 LD B,A ;SAVE CURRENT DRIVE SELECT DATA
F72D E6FB 1456 AND 11110000 ;MERGE IN NEW DRIVE UNIT# IN C
F72E B1 1457 OR C
F72F D31C 1458 OUT (BITDAT),A ;TO SELECT THE NEW DISK DRIVE
F6C0 CDAEF7 1459 CALL FORCE ;TEST NEW DRIVE'S READY STATUS
micro cornucopia, number 2, september 1981

(continued on top of page 14)
PFM Monitor Listing  (continued)

1651 ;******************************************************************************
1652 ; STORAGE ALLOCATION FOR 256 BYTE SCRATCH RAM *
1653 ;******************************************************************************
1654 ;
1655 ;******************************************************************************
1656 ;
1657 ;
1658 ;
1659 VECTAB EQU * ; INTERRUPT VECTOR TABLE STARTS
1660
1661 SIDVEC: DEFS 16 ;SPACE FOR 8 VECTORS FOR SID
1662 TOCVEC: DEFS 8 ;SPACE FOR 4 VECTORS FOR CTC
1663 SYSVEC: DEFS 4 ;SPACE FOR 2 VECTORS FOR SYSTEM
1664
1665
1666 ;KEYBOARD DATA INPUT FIFO VARIABLES
1667
1668 FIFO: DEFS 16 ;CONSOLE INPUT FIFO
1669 FIFO1: DEFS 8 ;FIFO DATA COUNTER
1670 FIFO2: DEFS 4 ;FIFO OUTPUT POINTER
1671
1672 ;STACK POINTER SAVE AND LOCAL STACK FOR INTERRUPT ROUTINES
1673 ;
1674 ;
1675 ;STACK POINTER SAVE AND LOCAL STACK FOR INTERRUPT ROUTINES
1676 ;
1677 ;
1678 ;
1679 ;
1680 ;
1681 ; 'SOFTWARE' VECTORS FOR INTERRUPT SERVICE ROUTINES
1682 ;
1683 TIKVEC: DEFS 2 ;1 SEC INTERRUPT ROUTINE VECTOR
1684 PINVEC: DEFS 2 ;PARALLEL CONSOLE INPUT VECTOR
1685 SINEC: DEFS 2 ;SERIAL CONSOLE INPUT VECTOR
1686 ;
1687 ;
1688 ;CLOCK-TIMER INTERRUPT VARIABLES
1689 ;
1690 TIKCNT: DEFS 2 ;BINARY CLOCK TICK COUNTER
1691 DAY: DEFS 1 ;CALENDAR DAY
1692 MONTH: DEFS 1 ; MONTH
1693 YEAR: DEFS 1 ; YEAR
1694 HRS: DEFS 1 ; CLOCK HOURS REGISTER
1695 MINS: DEFS 1 ; MINUTES REGISTER
1696 SECS: DEFS 1 ; SECONDS REGISTER
1697 ;
1698 ;
1699 ;DISK I/O DRIVER VARIABLES
1700 ;
1701 UNIT: DEFS 1 ;CURRENTLY SELECTED DISK#
1702 TRKTAB: DEFS 4 ; DRIVE HEAD POSITION TABLE
1703 SPEED: DEFS 1 ;SEEK SPEED FOR 1771 COMMANDS
1704 RECLN: DEFS 1 ;SECTOR RECORD LENGTH VARIABLE
1705 MOTOR: DEFS 1 ;DRIVE MOTOR TURN-OFF TIMER
1706 TRACK: DEFS 1
1707 SECTOR: DEFS 1

end
SUBSCRIPTION FORM

(It's OK to brag!)

☐ I own a big board (Hooray!)
☐ I don't own a Big Board but am very interested (There's hope)

EXPERTISE
- Software Systems
- Software Applications
- Languages
  1. 
  2. 
  3. 
- Hardware

INTEREST
- Fanatic
- None

Are you willing to be a resource in the areas where your expertise is 4 or 5?
- love to ☐
- probably ☐
- maybe ☐
- no ☐

What are your hardware/software needs now?

In the near future?

What kind of exciting adventure (misadventure) are you working on?

If you get the idea that this document is as interested in enlisting your aid and ideas as it is in getting a subscription, you're right. Lots of people are willing to subscribe, lots of people have ideas - and we'd like to encourage lots of people (especially you) to take an hour or two and put ideas and needs and accomplishments down on paper or disk. Then we can pass them along to others and that's what this journal is all about.

Send me six issues (1 yr.) of MICRO CORNUCOPIA. I understand that I can cancel at any time and receive a refund for the balance of the subscription. (Issue #1 was published in August 1981.)

U.S.  Canada & Mexico  Other Foreign
- $16.00  ☐ $20.00 (U.S. funds)  ☐ $26.00 (U.S. funds)
- $20.00 (1st class mail)  ☐ Back issues, Specify #s  ☐ Back issues, Specify #s
- ☐ Back issues, Specify #s  ☐ $3.00 each (U.S. funds)  ☐ $3.00 each (U.S. funds)
- $3.00 each

NAME ________________________ PHONE (?) ____________
ADDRESS ___________________________________________
CITY __________________________ STATE ________ ZIP _______

☐ Renewal  MICRO CORNUCOPIA  11740 N.W. West Rd  Portland, Oregon  97229
Because of the demand from new subscribers (bless their 
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-