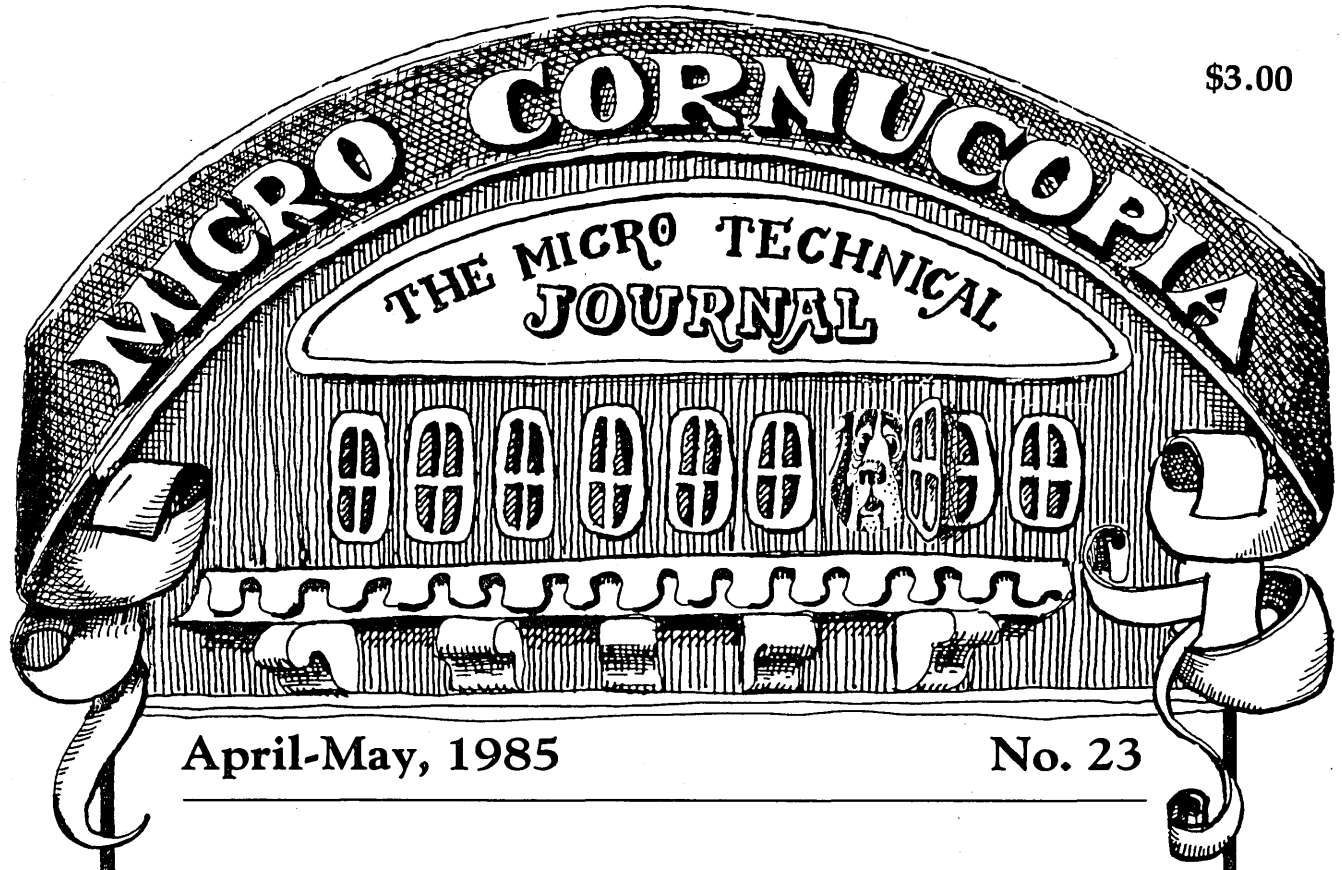
$3.00



April-May, 1985 No. 23

TABLE OF CONTENTS

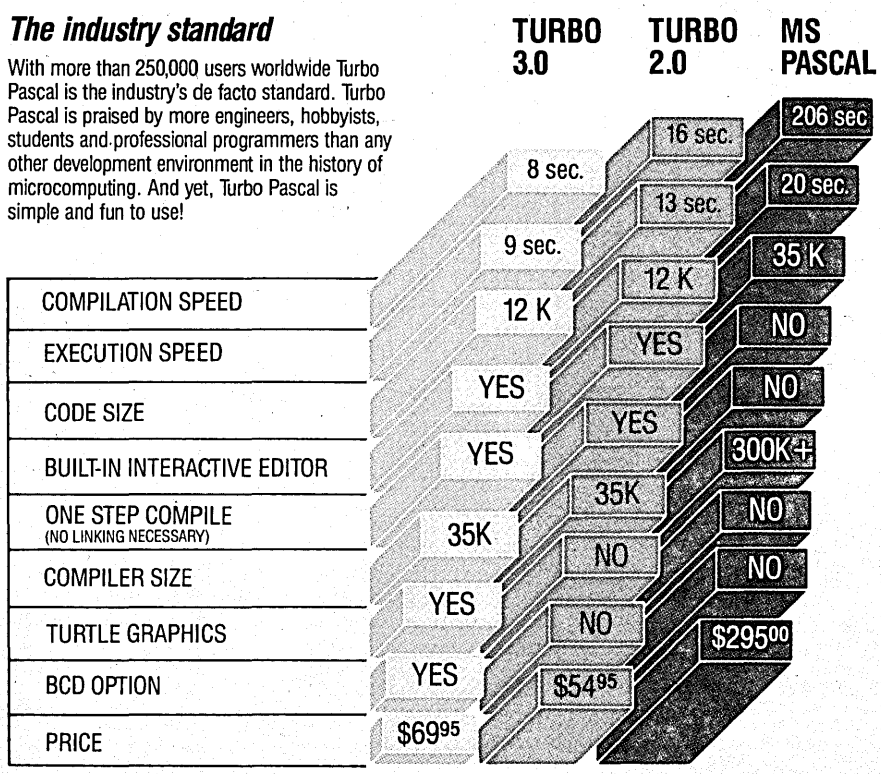
Regular Features

Future Tense

They said it couldn't be done. Borland Did It. Turbo Pascal 3.0

The industry standard

With more than 250,000 users worldwide Turbo Pascal is the industry's de facto standard. Turbo Pascal is praised by more engineers, hobbyists, students and professional programmers than any other development environment in the history of microcomputing. And yet, Turbo Pascal is simple and fun to use!



The best just got better: Introducing Turbo Pascal 3.0

We just added a whole range of exciting new features to Turbo Pascal:

- First, the world's fastest Pascal compiler just got faster. Turbo Pascal 3.0 compiles twice as fast as Turbo Pascal 2.0! No kidding.
- Then, we totally rewrote the file I/O system, and we also now support I/O redirection.
- For the IBM PC versions, we've even added "turtle graphics" and full tree directory support.
- For all 16 Bit versions, we now offer two additional options: 8087 math coprocessor support for intensive calculations and Binary Coded Decimals (BCD) for business applications.
- And much much more.

The Critics' Choice.

Jeff Duntemann, PC Magazine: *"Language deal of the century . . . Turbo Pascal: It introduces a new programming environment and runs like magic."*

Dave Garland, Popular Computing: *"Most Pascal compilers barely fit on a disk, but Turbo Pascal packs an editor, compiler, linker, and run-time library into just 29K bytes of random-access memory."*

Jerry Pournelle, BYTE: *"What I think the computer industry is headed for: well documented, standard, plenty of good features, and a reasonable price."*

Portability

Turbo Pascal is available today for most computers running PC DOS, MS DOS, CP/M 80 or CP/M 86. A XENIX version of Turbo Pascal will soon be announced, and before the end of the year, Turbo Pascal will be running on most 68000 based microcomputers.

An Offer You Can't Refuse

Until June 1st, 1985, you can get Turbo Pascal 3.0 for only $69.95. Turbo Pascal 3.0, equipped with either the BCD or 8087 options, is available for an additional $39.95 or Turbo Pascal 3.0 with both options for only $124.95. As a matter of fact, if you own a 16 Bit computer and are serious about programming, you might as well get both options right away and save almost $25.

Update policy

As always, our first commitment is to our customers. You built Borland and we will always honor your support.

So, to make your upgrade to the exciting new version of Turbo Pascal 3.0 easy, we will accept your original Turbo Pascal *disk* (in a bend-proof container) for a trade-in credit of $39.95 and your Turbo87 original *disk* for $59.95. This trade-in credit may only be applied toward the purchase of Turbo Pascal 3.0 and its additional BCD and 8087 options (trade-in offer is only valid directly through Borland and until June 1st, 1985).

(*) Benchmark run on an IBM PC using MS Pascal version 3.2 and the DOS linker version 2.6. The 179 line program used is the "Gauss-Seidel" program out of Alan R. Miller's book: *Pascal programs for scientists and engineers* (Sybex, page 128) with a 3 dimensional non-singular matrix and a relaxation coefficient of 1.0.

NOT COPY-PROTECTED

TURBOPASCAL

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by Credit Card call (800) 255-8008, CA (800) 742-1133

Carefully Describe your Computer System!

Mine is: ___ 8 bit ___ 16 bit
I Use: ___ PC-DOS ___ MS-DOS
___ CP/M 80 ___ CP/M 86

My computers' name/model is:

The disk size I use is:
☐ 5¼" ☐ 8" ☐ 3½"

For update: original Turbo disk must accompany order

YES! I want the Best!

Please send:		Quantity
Pascal 3.0	$ 69.95	
Pascal w/8087	$109.90	
Pascal w/BCD	$109.90	
Pascal w/8087 & BCD	$124.95 (SAVE $24.90)	

*These prices include shipping to all U.S. cities. All foreign orders add $10 per product ordered.

Trade-in Credit Claimed:

Amount: (CA 6% tax)

Payment: VISA MC BankDraft Check

Credit Card Expir. Date: ___/___

Name on Card:

Card #:

Name:

Shipping Address:

City: Zip:

State:

Telephone:

COD's and Purchase Orders WILL NOT be accepted by Borland. California residents: add 6% sales tax. Outside USA: add $10 and make payment by bank draft, payable in US dollars drawn on a US bank.

8

BORLAND INTERNATIONAL
Software's Newest Direction
4113 Scotts Valley Drive
Scotts Valley, California 95066
TELEX: 172373

MICRO CORNUCOPIA
P.O. Box 223
Bend, Oregon 97709
503-382-5060 Orders Only
503-382-8048 Tech. 9 - Noon

Editor & Publisher
David J. Thompson

Operations Manager
David Pogue

Assistant Editors
Rebecca Ozrelic Gary Entsminger

Accounting
Sandra Thompson

Graphic Design
Craig Lannes Michael Odell

Technical Department
Dana Cotant Eric Roby
Bruce Berryhill Laine Stump

Advertising Director
Alice Holbrow

Staff Assistants
Dorcas Dsenis Tracey Braas
Cary Gatton Carla Miller
Laura Pendley

Secretary
Christina Donaldson

Typography
Patti Morris & Martin White
Irish Setter

MICRO CORNUCOPIA supports systems programming languages and builders of single board and S-100 systems.

Application to mail at second class postage rates is pending at Bend, OR 97701. Published bimonthly by Micro Cornucopia Inc., 155 NW Hawthorne, Bend, OR 97701. Postmaster: Send address change to Micro Cornucopia Inc., PO Box 223, Bend, OR 97709.

SUBSCRIPTION RATES:

1 yr. (2nd Class)	$16.00
1 yr. (Canada & Mexico)	$22.00
1 yr. (Other foreign)	$30.00

Make all orders payable in U.S. funds on a U.S. bank, please.

ADVERTISING RATES: Available on request (call Alice Holbrow).

CHANGE OF ADDRESS: Please send your old label and new address.

SOFTWARE, HARDWARE, AND BOOK VENDORS: We would very much like to review your software, Single Board, and S-100 products. Send materials to the Review Department, Micro Cornucopia.

ISSN 0747-587X

MICRO CORNUCOPIA

April-May 1985 The Micro Technical Journal No. 23

April Fool

SOG IV

If you saw the list of folks who have already signed up for the SOG this year, you'd be surprised (we're certainly surprised).

Ezra Shapiro, West Coast technical editor for Byte, will be our keynote speaker. His talk will be titled "The Demise of Innovation." Ezra will be sharing with us his very definite ideas about the 'me-too' designs.

Slicer will be having a board construction party again (they'll throw in a bare expansion board free if you build one this year), plus they'll be bringing all the rest of their new products to demonstrate and sell. Call them to reserve a board.

The Ampro folks will be bringing up their Little Board and they are going to be holding a construction class (at special prices). In fact, you'll be able to put together a complete Little Board system complete with Integrand cabinet and drives, right at the SOG. they'll have something new to demonstrate. The Little Board folks will also be giving a talk on designing with the Z80. (Actually the Ampro folks think they are giving it.)

Integrand will be here displaying their Little Board cabinet (if you're surprised, go back and reread the previous paragraph). They'll also be doing a presentation on cabinet design and system cooling.

We've invited Philippe Kahn back, this time to talk about Modula-2 (Turbo style), and to toot his own horn (he plays a mean sax). We're still waiting to hear if he can come.

Speaking Of Music

In fact, we have a very musical group working at Micro C. Gary and Craig play guitar (semi-professionally); Bruce plays the flute; Jean (his other half) plays fiddle, and I have one of those fancy Yamaha synthesizers (the DX7), so I can pretend to be just about anything (even an editor).

My little MSX computer also contains a pretty decent synthesizer (rhythm anyone?), so we should be be able to come up with something interesting.

In fact, we will have a jam session Thursday evening after the Kickoff Cookout. You are all encouraged to bring your woodwinds, bongos, strings, horns, synthesizers, kazoos, speech generators, Integrands, tin ears, everything . . . . If we come up with something that even approximates music, maybe we'll have a lunch-time performance on Saturday.

White Water Rafting

This year we'll have the 2½ hour raft trip (just like last year), plus an all-new, all-day affair. The all-day white water trip will feature a total of 4½ hours on the McKenzie River (not counting the one-hour lunch on the bank). There will be an incredibly beautiful 1½ hour van ride to the starting point through some of the most beautiful alpine terrain in the Northwest.

Reservations

There are limitations on the number of people we can accommodate in the dorm (there are only 25 double rooms). The rafting trips shouldn't be any problem because the guides will run as many groups as necessary. But please call us to verify space if you want to stay in the dorm.

Trenton

The 10th annual Trenton Computer Festival will be held April 20 and 21 at Trenton State College, Trenton NJ. They say it's the oldest computer fair in the country. I've also heard they have one of the best flea markets going (a whole parking lot full of tables and tractor trailers). When I mentioned that I would be attending, Sol Libes suggested that I come in a truck.

(Continued on Page 77)

LETTERS

Go Winnie!

I have enjoyed your publication very much during the past year. In fact, I recently changed my subscription to first class and submitted two gift subscriptions.

I have a comment on the closing paragraph of your Kaypro column for October.

We have had three Kaypro 10s with this same 'status 02' error over the last six months. Each time the error occurred, the hard disk was not up to speed when the Kaypro 10 tried to read it. In all three cases the problem stemmed from the hard disk controller board. After the boards were replaced, the 'status 02' error problem was resolved.

Stanley D. Watson
4745 Convention Street
Baton Rouge LA 70806

Editor's note:

I also got a call from a local dealer about this problem. He had contacted two of the drive manufacturers and found out that some of the drives Kaypro is using are the low current versions which start very slowly. (In fact, they often won't start at all if the head gets left somewhere other than in the safety position and is resting on the disk.)

He has found that if he lubricates the winchester's stepper arm with a little silicon spray, the head will return to the safety position between data fetches and the drive will start properly the next time the system is turned on.

ZCHESS & Ticks

I certainly wouldn't begin to complain about the value of your users disks. But I would like to point out a bug or two.

First, on ZCHESS from K3 there have been several occasions when the computer playing black thinks it's entitled to move two pawns two spaces at once from other than the second rank. This wouldn't bother me so much except that it won't let me cheat in the same way.

Now, there's certainly twelve bucks worth of other stuff on that disk, but it would be nice if the main attraction for me didn't have that bug. (The CRC checks out, by the way, although I guess there could be some obscure memory problem in my machine causing this slip.) Anyway, I wonder if there's a version of ZCHESS out there somewhere without the bug?

The other thing I wonder about is that the version of ZCPR on your second release of it (haven't tried the first) puts a "' at the end of each comment it makes, including each listing in a directory. Minor little mark, and it's so damned useful that it's certainly overlookable. But what the heck is it doing there? And can the thing be exorcised? (By the way, I'm running it on an '83 II.)

Whitney Blauvelt
505 E. Denny Way #507
Seattle WA 98122

Editor's note:

Due to popular demand we changed ZCPR to display special characters (the normal CCP doesn't let them reach the screen, but people wanted to display them and so we . . .). Anyway, that little tick you see on the screen is what the Kaypro displays when it sees a null (0) byte. Every other system I've seen simply discards the null, but not the Kaypro (there are a number of situations where the null can be very irritating). So it is not without malice of forethought that our Pro-Monitor ROMs throw away nulls.

Pro-8 Installation

Someone once said that the devil dwells in the details. I found this to be true when I recently did the Pro-8/Plus-4 upgrade on my Kaypro II, installing two Mitsubishi half-height, quad density drives (4853) in place of the original 'A' drive. I would like to pass along what I learned about some of the details of this upgrade.

The existing screws holding the Tandon drives require a 7/64" allen wrench. This is a size not found in most of the common allen wrench sets.

The screws from the Tandon do not fit the Mitsubishis. You need 3m x 1m socket head cap screws which are not available in most hardware stores.

To install two half-height drives in place of one full-height drive the entire drive enclosure must be removed from the computer case and new holes drilled in it. This is not particularly difficult, but laying out the new holes is tedious. This wouldn't even need to be done if only one new drive were being installed.

The power connector for the drives is not sold in Radio Shack or in most electronic supply houses. Nor could I find it in most computer stores which claim to have a service department.

The Micro C Plus-4 Decoder Board does not go on the obvious way. The numbers on the decoder board must correspond with those on the main board, which means that the decoder board will cantilever out over the drive enclosure. When the cover is put back on, everything shorts out. This probably never happens to the Micro C folks who never put their covers back on. The problem can be fixed up with some electrical tape and cardboard, but it would be better if the designer went back to the drawing board.

Neither the problems nor their solutions were very profound, but resolving them took about three quarters of the time needed for the entire installation. Everything is working nicely now.

Walter B. Whitcher
319 E. 24th Street
New York NY 10010

Editor's note:

You are right about our covers. They come off when the machine comes out of the box and they usually stay off. Forever. Now, we are using a new low profile connector (the part that hangs out over the drives) so folks' circuits shouldn't be so short any longer.

Xerox 820 Info

I've recently 'discovered' Micro Cornucopia and I am enjoying it. It's nice to see a magazine that gets down to the basics. I wrote the Apple Cart column in Creative Computing for three years when an author could still do some of the things you do in your publication. Now, if you aren't reviewing someone else's software, no one wants your articles.

The parts of Micro C that I am most interested in are those about the original 820 CPU. I was the senior electrical manufacturing engineer on the program from its beginning. From that perspective I'd like to make a few comments.

A technical reference manual called "Software Development Guide" was published to complete the only detailed source of 820 documentation. The book includes all the software and hardware information available for the 820 board.

LETTERS

All the schematics for PWA etches 1 and 2 are there, as are the listings for versions 1.0 and 2.0 of the monitor ROMs. There is a wealth of other information including programming data for the SIOs, PIOs, the CTC and the FDC chips. Perhaps you could get permission to reprint this as a project for Micro C magazine.

About disk drives, software, and other things. Most people know that the 820 was designed for single-sided 5¼" drives. Later, the double-sided drives were added and so were single-sided 8" drives. With the etch 1 CPU, it was not directly possible to use DS 8" drives. For DS 5¼" drives, version 3.0 of the CP/M operation system was issued. This allowed the user to initialize and copy on two-sided media.

In the process of developing test procedures and equipment for the two-sided drives, I discovered that the software was really looking at four separate single-sided drives. Once a diskette was initialized with the 3.0 INIT program, I used my DS system as four drives in the sequence A,C and B,D; worked fine, too. I could get the equivalent of four single-sided drives in a two-drive package. Otherwise, the original system was capable of four single-sided drives anyway. The user would only need to properly wire and encode the drive block on the individual drives. Shugart's SA400 manuals show how to do this.

Chuck Carpenter
3714 Bishop Hill Dr.
Carrollton TX 75007

Fixing The SBASIC Sorts

In Jack Rodenhi's SBASIC column (Issue 19) there seems to be some confusion about SBASIC's rules governing recursion. Rodenhi states, "In the manual's example program on recursion, X2 seems equivalent to our variable ListSize. In the discussion of their program, they claim that X2 would not be duplicated when P3 is called from P4." Jack is correct that ListSize is equivalent to X2 and that it is indeed duplicated. However, he is incorrect in saying that this corresponds to P4's calling P3 on line 20 of the manual's example program.

In SSSORT (Jack's procedure which contains ListSize), it is SSSORT itself which calls its nested procedure READRECORDS (causing ListSize to be treated recursively). The corresponding call is on line 26 of the manual's example program, where P2 calls P3. There it clearly shows that X2 will be duplicated.

Although the example program correctly indicates how SBASIC's recursion occurs, the manual does not. (So what else is new?) Recursion occurs when a procedure calls itself or when one nested procedure calls another nested procedure (including the situation where a nested procedure calls the unnested procedure which contains it). Recursion will also occur when a nested procedure is called by the procedure which contains it. (Anyone for Peter Piper?)

Also, arrays are treated differently than variables in recursive calls because arrays are not duplicated. (If the call causes the DIM statement to re-execute, however, the array may be reinitialized.)

This provides at least four ways for Jack to fix his program:

1. Procedures (such as READRECORDS) which change one of the variables he describes as "global in the procedure SSSORT" could be rewritten as subroutines.

2. Those same procedures could instead be rewritten as functions which return the value of the changed variable (in use this might look like "ListSize = READRECORDS (ListSize(0)").

3. The variables in question (such as ListSize) could instead be declared as one-element integer arrays (e.g., ListSize (0)).

4. Those same variables could be made global to SSORT by moving their declaration outside of the procedure (of course, that would make them global to the rest of the program as well).

Richard Levine
3105 Meadow Grove Dr.
San Diego CA 92110

Run CP/M On An IBM?

I need your help with a CP/M problem. I haven't been able to locate any products that will do what I want.

I am writing software in Turbo Pascal under MS-DOS. I want to port over finished software to run on CP/M machines. For this conversion, are there coprocessor boards I can add to my PC-compatible system so it will look like a CP/M system to a CP/M Turbo Pascal compiler? Are there any products for the PC that emulate the Z80 or 8080 to do what I want?

Please supply the names and addresses of any suppliers of suitable boards and emulators. Many thanks for your help.

Frank J. Mihm
Box 622
Aptos CA 95001

Editor's note:

What a refreshing request. For a while there, I thought everyone was going in the other direction. OK, there are several clones (or semi-clones) that say they can run both: Rainbow, Chameleon, and, I believe, one of the Heath systems. However, I am not aware of any plug-in boards that will handle this. Any suggestions, anyone?

The obvious option would be to get a Kaypro and transfer the software over serially. Kaypro also has a utility that is supposed to be able to read MS-DOS disks and transfer files to CP/M. So far, we haven't been able to get it to work. What are you working on, by the way?

Speed-Up Problem

I purchased a Pro-Set II so I could upgrade my 1983 Kaypro II to 5MHz (and then do a program for the Palm Beach KUG). In the process, I learned a few things.

First, my Kaypro II is really a 4 inside so I had to exchange the Pro-Monitor II for a Pro-Monitor 4. Second, the Z80B costs between $12 and $45 depending on where you buy. Third, I'm still having a problem because at 5MHz the system runs for about a minute and a half and then locks up.

Gene Klein
12775 Builford Circle
West Palm Beach FL 33414

Editor's note:

You are not the only person who has a II with the heart of a 4. Anyone with an older Kaypro II (pre-84—which means it has the standard full-width Tandon drives and no graphics) should take the top off his or her Kaypro II and read the paper stuck to the top of the monitor ROM (there are two chips with paper on top—the one nearest the front

(Continued on Page 64)

Automatic Disk Re-logging With CP/M 2.2

By Clark A. Calkins

1907 Alvarado Ave.
Walnut Creek CA 94596

If you've felt like resetting the parity bit on the person who installed "BDOS ERROR ON A: DISK R/O" in your computer's vocabulary, then you're in the right spot. In this article we'll cover why CP/M makes a disk "read only." Plus, you'll find out what you're really telling the operating system when you hit ctrl-C. Doing this mod is (almost) never having to say "warm boot."

I'll show you how to modify the source of CP/M (see reference 1) rather than asking you to patch bits and pieces of the executable code.

CP/M Is Smart

When the CP/M V2.2 operating system first accesses a disk, it scans the directory and makes a map showing which spaces (or blocks) on the disk contain data and which spaces are available for use. This map is called a bit map and is updated when space is allocated (a file is created or expanded) or when space is released (a file is erased).

The map is kept in memory, so unless CP/M does some checking, you could destroy data simply by changing disks. You see, when CP/M writes data to the new disk, it writes the data into the areas that the bit map says are available. (Of course, these areas might not be available on the new disk, so it would be writing over valid data.)

To prevent this, CP/M keeps a condensed record of the directory (actually, this is a series of checksum bytes) in order to tell if the disk directory has changed. When this occurs, CP/M will tag this disk as 'read only,' so you can't write to it. Nice, huh? The only problem is that you get a "BDOS ERROR ON A: DISK R/O."

Of course, this means you need to type ctrl-C to let CP/M know the disk has been changed—after CP/M just told you that the disk had changed. (Oh well, nobody said operating systems had to be both smart and helpful.)

Making CP/M Helpful

The modifications described below will force CP/M to update the bit map (and the directory check) whenever a disk is changed, as long as the update

(Continued on Page 6)

Figure 1 - Changes to the CCP

```
          ;
          ;   Process command line here.
          ;
          CMMND2  MVI     C,41        ;Reset the disk re-log vector.
                  CALL    ENTRY
                  LXI     D,TBUFF
          ...
          ...
          UNKWN8  MOV     A,B         ;Now store the character count.
                  STA     TBUFF
                  CALL    CRLF        ;Clean up the screen.
                  CALL    STDDMA      ;Set standard transfer address.
                  CALL    SETCRDV     ;Reset current drive.
                  MVI     C,41        ;Reset the disk re-log vector.
                  CALL    ENTRY
                  CALL    TBASE       ;And execute the program.
```

Figure 2 - Clearing the Re-log Vector

```
;
;   BDOS function jump table.
;
NFUNCTNS EQU          42        ;Set function count limit.
;
FUNCTNS DW      WBOOT,GETCON,OUTCON,GETRSR,PUNCH,LIST,DIRCIO,GETIOB
        DW      SETIOB,PRTSTR,RDBUFF,GETCSTS,GETVER,RSTDSK,SETDSK,OPENFIL
        DW      CLOSEFIL,GETFST,GETNXT,DELFILE,READSEQ,WRTSEQ,FCREATE
        DW      RENFILE,GETLOG,GETCRNT,PUTDMA,GETALOC,WRTPRTD,GETROV,SETATTR
        DW      GETPARM,GETUSER,RDRANDOM,WTRANDOM,FILESIZE,SETRAN,LOGOFF,RTN
        DW      RTN,WTSPECL,CLRLOG
```

Figure 3 - Checking the Disk Directory

```
          ;   Routine to set or compare the directory checksum byte. If
          ; (C)=0FFH, then this will set the checksum byte. Else the byte
          ; will be checked. If the check fails (the disk has been changed),
          ; then this will be write protected.
          ;
          ; Note, changes added here will not write protect the disk unless
          ; the re-log vector says this cannot be re-logged (a file is open).
          ;
        CHECKDIR:LHLD   CKSUMTBL
                 XCHG
                 LHLD   ALLOC1
                 CALL   SUBHL
                 RNC                ;Ok if (cksumtbl) > (alloc1), so return.
                 PUSH   B
                 CALL   CHECKSUM    ;Else compute checksum.
                 LHLD   CHKVECT     ;Get address of checksum table.
                 XCHG
                 LHLD   CKSUMTBL
                 DAD    D           ;Set (HL) to point to byte for this drive.
                 POP    B
                 INR    C           ;Set or check?
                 JZ     CHKDIR1
                 CMP    M           ;Check them.
                 RZ                 ;Return if they are the same.
                 CALL   MOREFLS     ;Not the same, do we care?
                 RNC
                 LHLD   RELOG       ;Can we re-log in this drive?
                 CALL   GETWPRT+3   ;Get bit for this drive.
                 JNZ    WRTPRT      ;Nope, set write protect bit then.
                 STC                ;Okay, set carry saying that this disk must
                 JMP    STFILPOS    ;be re-logged. Also reset pointer to start.
        CHKDIR1  MOV    M,A         ;Just set he byte.
                 RET
```

Figure 4 - Getting the Next File Entry From the Directory

```
;   Move on to the next next file position within the current
; directory buffer. If no more exist, set pointer to 0FFFFH
; and the calling routine will check for this. Enter with (C)
; equal to 0FFH to cause the checksum byte to be set, else we
; will check this disk and set write protect if checksums are
; not the same (applies only if another directory sector must
; be read).
;
; Modifications here will re-log the current disk if a change has
; been made and it is safe to do so.
;
NXENTRY:LHLD    DIRSIZE   ;Get directory entry size limit.
        XCHG
        LHLD    FILEPOS   ;Get current count.
        INX     H         ;Go on to the next one.
        SHLD    FILEPOS
        CALL    SUBHL     ;(HL)=(DIRSIZE)-(FILEPOS)
        JNC     NXENT1    ;Is there more room left?
        JMP     STFILPOS  ;No. Set this flag and return.
NXENT1  LDA     FILPOS    ;Get file position within this directory.
        ANI     03H       ;Only look within this sector (only 4 entries fit)
        MVI     B,5       ;Convert to relative position (32 bytes each).
NXENT2  ADD     A         ;Note that this is not efficient code.
        DCR     B         ;5 'ADD A's would be better.
        JNZ     NXENT2
        STA     FCBPOS    ;Save it as position of fcb.
        ORA     A
        RNZ               ;Return if we are within buffer.
        PUSH    B
        CALL    TRKSEC    ;We need the next directory sector.
        CALL    DIRREAD
        POP     B
        CALL    CHECKDIR  ;Check this directory segment.
        RNC               ;Just return if no more to do.
        PUSH    B         ;Wait, we must re-log this drive and then
        CALL    BITMAP    ;start from the beginning again.
        POP     B
        JMP     NXENTRY
```

Figure 5 - Opening a File For Reading or Writing

```
;   Open a file (name specified in fcb).
;
; This will mark the drive to prevent re-logging at a later time.
;
OPENIT  MVI     C,15      ;Compare the first 15 bytes.
        CALL    FINDFST   ;Get the first one in directory.
        CALL    CKFILPOS  ;Any at all?
        RZ
OPENIT1 LHLD    RELOG     ;Set flag to prevent subsequent re-logging
        MOV     B,H       ;of this drive. We now call it unsafe to
        MOV     C,L       ;change.
        CALL    SETBIT
        SHLD    RELOG
        CALL    SETEXT    ;Point to extent byte within users fcb.
        MOV     A,M       ;And get it.
        PUSH    PSW       ;Save it and address.
        PUSH    H
        CALL    FCB2HL    ;Point to fcb in directory.
        XCHG
        LHLD    PARMS     ;This is the users copy.
        MVI     C,32      ;Move it into users space.
        PUSH    D
        CALL    DE2HL
        CALL    SETS2B7   ;Set bit 7 in 'S2' byte (unmodified).
        POP     D         ;Now get the extent byte from this fcb.
```

(Listing continued)

(continued from page 4)

can be done safely.

In the following figures, it is unsafe to alter the tables only while a file is open. Therefore, application programs (Dbase II, WordStar, SuperCalc, etc.) are protected, while still allowing maximum flexibility on disk changes.

You can enter the modifications with any good editor (we all have our favorites). Check the listed routines against your source code to be sure you understand which lines have been inserted or changed. Then re-assemble the source and combine it with the BIOS code. Finally, you can SYSGEN a new disk.

Changes To The CCP

The console command processor (CCP) needs to be modified to clear the disk re-log vector after each command and prior to the execution of a user's program. I've added a new BDOS function (#41) for doing this. If you are using ZCPR or other new 'front end,' then these changes must be placed at the appropriate locations within the code in Figure 1. Consult local talent if you are not sure what to do (or where to do it).

Changes To The BDOS Function Jump Table

An added function (#41) in Figure 2 allows the re-log vector to be cleared without having to reset all the disks.

Changes To The Routine CHECKDIR

The routine in Figure 3 is used to check the disk directory against the stored check value. If the disk needs to be re-logged, the carry flag is set. (The disk will be marked 'read only' if the check values don't match and it is not OK to re-log the disk.) Note that changes are marked with a vertical line in the left column.

Changes To Routine NXENTRY

Figure 4 shows a routine which is called to get the next file entry from the directory. It will check to see if a disk has been changed by calling the routine in Fig. 3 (if a new segment is to be read from the disk). The modifications here will cause the disk bit map to be re-computed if a disk change is detected and the disk is not write-protected.

Changes To The OPENIT Routine

The routine in Figure 5 will open a file for reading or writing. When this occurs successfully, we will mark this drive so it cannot be re-logged in the event of a disk change. Note that there is no way to determine whether the file has been closed (making it safe once again for re-logging). So, we have to issue a reset command when we close a file.

Disk Reset Function

This routine (Figure 6) resets all disks. The result is that disk A is logged in and all other disks are write-enabled. The routine clears the disk re-log vector if it was set.

Added Routine CLRLOG

The routine in Figure 7 has been added to clear the re-log vector. It allows the CCP to reset this without forcing the system to re-read all the disks. Use this when you know that changing the disks is safe. Locate this routine after the RSTDSK routine in Figure 6.

Storage Area Changes

To accommodate the extra storage space needed, the changes shown in Figure 8 are necessary, making the BDOS 36 bytes longer. If it is critical that the BDOS remain the same size (as it usually is), some space must be found. Have no fear, there is plenty of room for improvement.

Figure 5 - Listing continued

```
            LXI     H,12
            DAD     D
            MOV     C,M         ;into (C).
            LXI     H,15        ;Now get the record count byte into (B).
            DAD     D
            MOV     B,M
            POP     H           ;keep the same extent as the user had originally.
            POP     PSW
            MOV     M,A
            MOV     A,C         ;Is this the same as in the directory fcb?
            CMP     M
            MOV     A,B         ;If yes, then use the same record count.
            JZ      OPENIT2
            MVI     A,0         ;If the user specified an extent greater than
            JC      OPENIT2     ;the one in the directory, then set count to 0.
            MVI     A,128       ;Otherwise set to maximum.
OPENIT2     LHLD    PARAMS      ;Set record count in users fcb to (A).
            LXI     D,15
            DAD     D           ;Compute relative position.
            MOV     M,A         ;And set the record count.
            RET
```

Figure 6 - Resetting the Disks

```
;
;   Function to reset the disk system.
;
; Changed to also clear the relog vector.
;
RSTDSK  CALL    CLRLOG      ;Clear the re-log vector (returns HL=0)
        SHLD    WRTPRT      ;Clear write protect vector.
        SHLD    LOGIN       ;Also the disk log in vector.
        STA     ACTIVE      ;Select drive A:.
        LXI     H,TBUFF     ;Setup default DMA address.
        SHLD    USERDMA
        CALL    DEFDMA
        JMP     LOGINDRV    ;Now log in drive A:.
```

Figure 7 - Clearing the Re-log Vector (again)

```
;
;   Function #41 to re-log all disks.
;
CLRLOG  XRA     A           ;Set (A) and (HL) to zero also.
        MOV     H,A
        MOV     L,A
        SHLD    RELOG       ;Clear the relog vector.
        RET
```

The following entry points end with a jump instruction and may simply be moved in front of the destination, thus saving the three bytes for the jump. The code for these is not shown here, but upon examination, the section of code to move should be apparent.

```
SETDSK
OPENFIL
CLOSEFIL
GETNXT
DELFILE
READSEQ
WRTSEQ
FCREATE
```

It should go without saying that any experimentation must be done on a spare disk, and the resulting system should be tested thoroughly with all common applications before converting all of your disks. I have tested this for a few months now, but I still have not changed my business disks. I certainly don't want to chance a problem with my main inventory and expense record disk. The IRS just wouldn't buy it!

Editor's note: Clark Calkins wrote the utility to generate source from your CP/M system image. See the following References for information on how to get this package.

References

1. *SCG22, A Source Code Generator For CP/M V2.2.* This will generate source code complete with labels and comments for the CP/M 2.2 operating system. Available from C.C. Software, 1907 Alvarado Ave., Walnut Creek CA 94596 for $45 plus $1 shipping ($2.50 foreign) and 6.5% sales tax for those lucky enough to live in California.

2. *CP/M Plus, User's Guide,* Digital Research, Pacific Grove, California. Refer to Section 2 for disk change information.

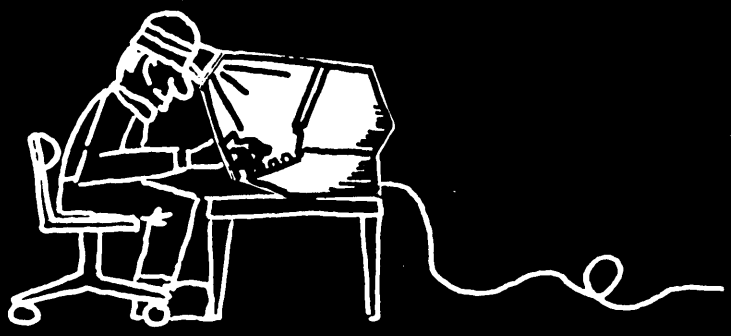
3. *CP/M Plus, Programmer's Guide,* Digital Research, Pacific Grove, California. Contains a good discussion of how the CCP and the BDOS work together in Section 1.

■ ■ ■

Figure 8 - BDOS Data Storage Pool

```
EMPTYFCB:DB     0E5H        ;Empty directory segment indicator.
WRTPRT   DW     0           ;Write protect status for all 16 drives.
LOGIN    DW     0           ;Drive active word (1 bit per drive).
RELOG    DW     0           ;Drive re-log disabled vector.
USERDMA  DW     080H        ;User's DMA address (defaults to 80 h).
```

HAS CP/M® LEFT YOU IN THE DARK?



With the MRS/OS Source Code, you can see the light.

If you own a CP/M compatible operating system, you've had to put up with the mistakes and quirks of someone else's programming. Until now. Now you can see the light with MRS/OS. In fact, MRS is a full operating system designed to replace CP/M 2.2 or CDOS and it comes with complete source code. MRS is designed for Z80 processors, runs CP/M software, and can interface directly to a CP/M BIOS, saving you a lot of sysgen time.

With MRS, you get more than what you pay for. For under forty dollars you receive fully commented source code for standard and extended BDOS functions, a sample BIOS, our all-in-one utility package and a 130 page manual.

So if you're tired of being in the dark with some other guy's program, here's the answer to your prayers.

$39.95 (includes shipping & handling in N. America; overseas add $12) Mass. orders include 5% sales tax

Order by phone 24 hours a day!

(617) 478-3102

Tech. inquiries: 7:30pm-10:30pm EST (Wed. ONLY)

OCCO inc.

CP/M is a registered trademark of Digital Research Corp.
CDOS is a registered trademark of Cromemco Corp.

SPECIAL FREE OFFER TO 8" DISK USERS!

ACT NOW and receive our ultra-fast, flexible disk format program, **FREE!** FORMAT can format a 77 track 8" disk in just 25 seconds and provides you with four of the most common disk formats used with CP/M based systems:

- 8" SSSD 128 byte sectors, 26 sectors/track, 76 tracks
- 8" SSDD 256 byte sectors, 26 sectors/track, 76 tracks
- 5.25" SSDD 256 byte sectors, 18 sectors/track, 40 tracks
- 5.25" SSDD 512 byte sectors, 10 sectors/track, 40 tracks

YES! Please send me ____ **MRS/OS Source Code(s)** for just $39.95 each on ☐ 8" SSSD disk or ☐ 5¼" Kaypro format disk. If I act **right now,** I'll receive a FREE **FORMAT** program included on the disk (8" disk users ONLY).

☐ VISA/Mastercard # _____ exp. date _____ (circle one)

☐ Check

☐ Please send more information

Name _____

Address _____

City _____ **State** _____ **Zip** _____

Phone _____

Send to: **OCCO, Inc.**
28 Claflin St., Milford, MA 01757

The S-100 Bus

By Dave Hardy

736 Notre Dame
Grosse Pointe MI 48203

As we will see in future 'S-100 Bus' columns, the IEEE-696 (S-100) bus can do things that most other computer busses cannot. But before we get into such exciting stuff as multi-processing, TMA, and Master-Slave operating procedures, we have to be familiar with the simpler operations of the S-100 bus. Probably the easiest thing we can do to become familiar with the S-100 bus is add some simple I/O circuits. This month's column will teach you how to do that.

Some Simple I/O Circuits

Unlike most other types of small computers, an S-100 machine offers the great advantage of (almost) painless I/O expansion. If the user needs another printer port or a digitizer input, he needs only to plug an additional board into the S-100 bus, interface his software, connect the wires, and go.

S-100 boards are available in several flavors to do just about anything a computer board can do. In fact, if the user is incredibly cheap (or frugal, which is how I refer to myself) he can even build his own I/O boards with a little pain, a little grief, and very few bucks.

In fact, most S-100 users don't know how easy it is to add additional I/O to their machines. The circuits shown in Figures 1 and 2 illustrate some simple 'bare-bones' parallel input and output circuits that can be added to any S-100 machine for less than $50 (including the cost of a prototype board and all parts). Of course, the prototype card would have room for a lot more than a single I/O port. A resourceful builder could probably cram about 50 of these ports onto a single board (and then he could share the other 49 with his friends and neighbors).

If you are confused by the circuits in Figures 1 and 2, don't worry. If you have any experience with TTL circuits, the function of these circuits should be readily apparent. If not, at the end of this column, I will recommend some books you can buy to transform yourself into an S-100 wizard in just a few hundred pages. (Of course, it would also help to read future issues of Micro C.)

Now for the hard stuff. Following is probably the world's shortest introduction to S-100 board design (at least, for I/O ports).

Figure 1 - Parallel Input

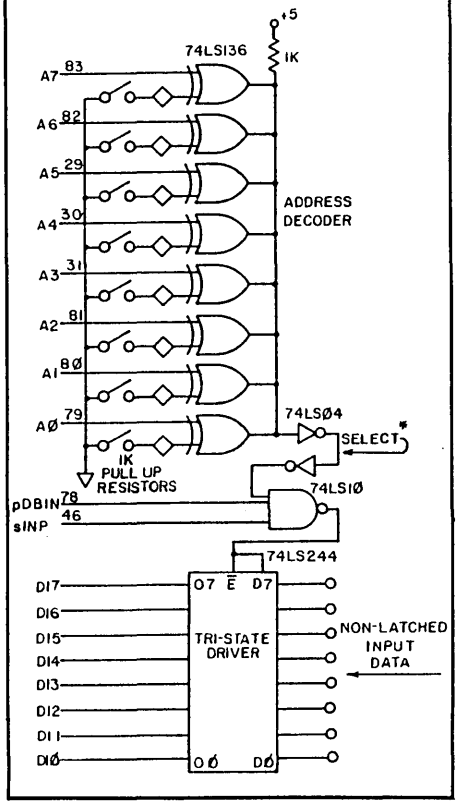
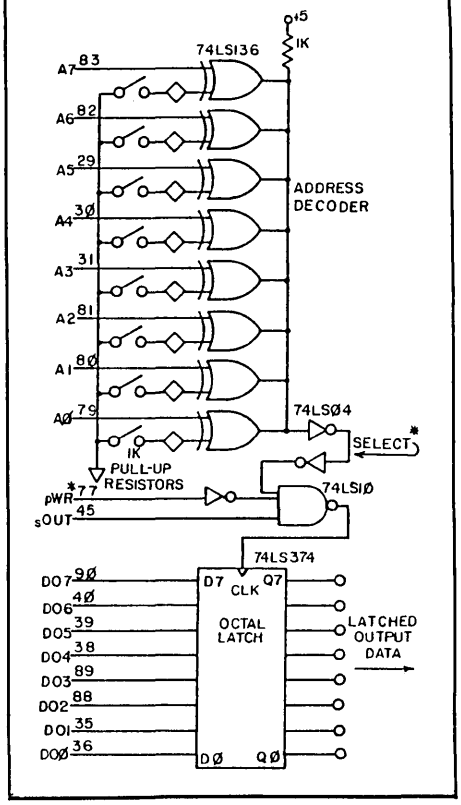


Figure 2 - Parallel Output



I/O Addressing

In order to send data between a peripheral device like a printer or a bank of LEDs and the S-100 bus, an interface must monitor several of the S-100 lines to determine when the bus is ready to input or output a piece of data. In addition, the interface must also decide if the data about to be transferred is to be transferred by it, or by some other interface. The interface does this by reading the S-100 bus address lines to see if its own unique I/O port number (or group of port numbers) is being addressed (see the ADDRESS DECODER section of Figure 1). This procedure is called "Address Decoding."

If the number it reads doesn't match its own address (port) number, then it does nothing. If the number it reads does match its own address, then it assumes that the S-100 bus might want to talk to it. The address match causes the SELECT* line (shown in Figure 1) to go true (that is, it goes to a logic 0, since SELECT* is a "low true" signal), which enables the rest of the interface.

Still More Decoding

After determining that the I/O address is proper, a few other things still need to be determined by the interface. First, all the address decoder does is read the address lines and tell the interface when they match. In the S-100 bus, I/O addressing and memory addressing are both done on the same lines, so the next thing that must be determined is whether the address the decoder is seeing is an I/O address or a memory address. This is done by looking at the sINP and sOUT lines, which indicate if the read or write operation about to take place is an I/O operation, or a memory operation.

Once the interface has decided that the bus operation about to take place is at its own address, and that it is an I/O operation, it then monitors the pWR* (for output) or pDBIN (for input) line to determine exactly when to transfer the data from or to the bus.

The following are definitions for the S-100 bus lines we'll be using. These definitions should help you understand what is happening in the examples.

(Continued on Page 11)

Now there is a choice!

- ☐ Does your workload demand another printer, but your budget says no to another system?
- ☐ Do your print jobs take hours of your system's time?
- ☐ Do you have a system dedicated to bulk mailing?

The PHOENIX INTELLIGENT PRINTER BUFFER was designed to solve these problems easily and effectively. Anyone can install and use the Phoenix without modifying software; it's as simple as plugging it in and typing your instructions.

The Phoenix Buffer runs two printers from a single input, comes with up to 512K of RAM, and has a built-in merge feature which will allow an average bulk mailing of 900 personalized letters in only 64K of RAM. When ordering, select either the centronics parallel or the RS-232 serial version.

Additional features include:

- ★ MULTIPLE COPIES (1-256).
- ★ ABORT one or all jobs for either printer.
- ★ PAUSE between jobs.
- ★ RE-START a job (i.e., to align paper).
- ★ 1 YEAR factory warranty.

64K: $349

256K: $649

512K: $849

SEND ORDERS TO:

PHOENIX ELECTRONIC DEVELOPMENT CORPORATION
618 Venice Boulevard
Marina Del Rey, CA 90291
(213) 827-2511

DEALERS AND DISTRIBUTORS WELCOME!

(continued from page 9)

Line Descriptions

pDBIN (processor Data Bus IN) pin 78, active high—A generalized read strobe, asserted for memory read, I/O read, and interrupt acknowledge cycles. Used to enable a slave's data output bus drivers to gate data onto the S-100 bus.

pWR* (processor WRite) pin 77, active low—A generalized write strobe, asserted for memory and I/O write cycles. Indicates to slave that the data output bus contains valid data.

sINP (status INPut) pin 46, active high—Active when S-100 bus is executing an input cycle and reading from an I/O port address.

sOUT (status OUTput) pin 45, active high—Active when S-100 bus is executing an output cycle and writing data to an I/O port address.

Although these definitions are not complete, and are not strictly in agreement with the IEEE-696 standard, they should be adequate for most simple S-100 interfacing projects.

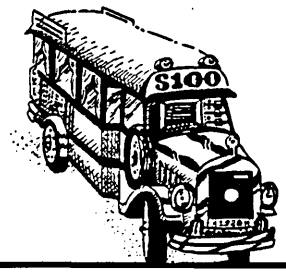
Homework

If you want to know more about the S-100 (IEEE-696) bus, I would recommend the following book: "Interfacing to S-100/IEEE-696 Microcomputers" by Sol Libes and Mark Garetz (Osborne/McGraw-Hill 1981, ISBN 0-931988-37-3). You might also want to get a copy of IEEE Task 696 (The S-100/IEEE-696 standard) directly from the IEEE, although the preliminary standard is contained in the book mentioned above.

Next Time

The next "S-100 Bus" will demonstrate some ways to see what your machine's bus is doing, introduce the IEEE-696 concept of multiple processing, and answer some reader questions about S-100 system expansion and troubleshooting.

■ ■ ■



In The Public Domain

By Sol Libes

Box 1192
Mountainside NJ 07092

In my last column I gave an introduction to public domain software which included its history and some background on the organizations which publish and distribute PDS. In this column I'll discuss the high level languages available in the public domain.

Latest SIG/M Releases

Before launching into the column, however, I've listed the latest SIG/M disk releases below (each disk is referred to as a volume):

Volume—Description
210—CP/M-86 Utilities (SQ, USQ, CRC, etc.)
211—CP/M-86 Kermit Communications package
212—CP/M+ Utilities, Kaypro multi-disk formats, C programs, etc.
213—REC (Regular Expression Compiler) Vol. 1
214—REC Vol. 2
215—REC Vol. 3
216—CP/M-86 MODEM package
217—CP/M-80 emulator for CP/M-86, dBase-II Toolkit, BBS security system, CP/M-86 MODEM overlays

Language Software In The Public Domain

BASIC holds a particular fascination for me. I still remember the day, back in 1975, when I placed a paper tape containing MITS Basic (later to become Microsoft Basic) in my old teletype (that was when we thought 10 characters per second was fast), toggled in a little loader program with the switches and lights on the front panel of my Altair 8800 computer (it contained all of 4K of memory and ROM memory was unheard of) and waited several minutes for BASIC to load and execute. After several tries I jumped for joy when the word 'READY" appeared on the paper (who had a CRT in those days?).

It was the first high level language I'd ever seen, and its level of intelligence amazed me. Always being curious about how things worked, I immediately wanted to know about BASIC. But I was always frustrated because the versions I used were only available in machine code form, so decoding them was a horrendous task. Thus, when I saw versions of BASIC including the full source code in the PDS software libraries, I was in a state of delight. If you are as interested as I am in how things work, or would like to create your own personal version of BASIC, get a copy of these programs and dig right in!

Eight Is Enough

I count eight different versions of BASIC in the public domain, and most (not all) include the source code. It is probably best to start with the smallest version of BASIC. It is on volume 11 of the CPMUG library and is called "TINIDISK." Many people refer to this as "Tiny BASIC." It is a version of Wang's Palo Alto BASIC originally published in Dr. Dobb's Journal. The disk contains .ASM and .COM versions as well as full instructions in its use in a .DOC file. The .COM file is only 3Kbyte in size. This version of BASIC may be lacking the power of more recent versions, but its small size makes it easier to study the structure of a high level language. Also on the disk you will find a version of Star Trek (it takes up 6K bytes) that runs under TINIDISK.

BASIC/5

A more elaborate version of BASIC in the CPMUG library is a disk version of Processor Technology's "BASIC/5." This was written to run under CP/M V1.3 and occupies about 8K of memory. PT manufactured S-100 systems that were primarily cassette-based, and when the company went out of business much of its software was put in the public domain and enhanced for disk operation.

Tarbell BASIC

CPMUG volumes 31 and 32 contain an even more advanced version of BASIC called Tarbell BASIC. Tarbell is another S-100 manufacturer, still in business, which has put much of its software into the public domain. This version still has bugs in it and its editor leaves something to be desired. However, it does have some very powerful features such as WHILE . . . .WEND.

LL-BASIC

Probably the most popular public domain BASIC interpreter is the version of Lawrence Livermore BASIC (usually referred to as "LL-BASIC") found in CPMUG volumes 2 and 10. This was written at Lawrence Livermore Labs in California using public funding and hence is in the public domain.

EBASIC

Another publicly funded BASIC project was EBASIC written by Gordon Eubanks. EBASIC is a semi-compiler with a runtime interpreter. Gordon later created enhanced commercial versions of EBASIC sold under the names CBASIC, CBASIC-2 and CB-80.

EBASIC is the most powerful of the PDS BASICs, and is found on SIG/M volume 26 with a help file on volume 14 (you will need HELP.COM on volume 13). The original source code, written in PL/M for the compiler and runtime interpreter, is on CPMUG volumes 29 and 30. Also, volume 53 of CPMUG contains EBASIC floating point routines. EBASIC is a structured version of BASIC in that it does not require line numbers except in statements that are targets for GOTOs and GOSUBs.

An EBASIC compiler written in the Forth-83 language is on SIG/M volumes 204 (CP/M-80 version) and 205 (CP/M-86 version).

Pascal, C And The Like

Several versions of Pascal are available in the public domain, with a Pascal help file found on SIG/M volume 14.

First is the Pascal compiler on SIG/M volume 8. This disk includes a .COM version as well as the .ASM modules and documentation and is written in Pascal, thus providing an excellent example of how to use the language. This Pascal is different than UCSD Pascal in that it generates 8080 object code linked to a runtime library.

SIG/M volume 82 contains the JRT Pascal compiler. This program was first sold commercially, became quite popular, and developed a loyal following. Unfortunately, JRT was a poorly managed company and is no longer in business.

SIG/M volume 162 contains a concurrent Pascal compiler called PASCAL-S.

(Continued next page)

(continued from page 13)

C

Only one version of the C language is available so far. This is a version of small-C, which appeared several years ago in Dr. Dobb's Journal, implemented for CP/M-86. The complete source code and documentation are included for anyone interested in translating it to 8080 or Z80 code.

FOCAL

FOCAL is another procedure-oriented language similar to BASIC. It was first introduced by Digital Equipment Corporation for the early PDP-8 minicomputers. This program is found on CPMUG volume 16.

ALGOL/M

ALGOL/M, an Algol-60 implementation, is on CPMUG volume 28. It includes full documentation, as well as a set of test and demo programs. An Algol help file is contained on SIG/M volume 14.

RATFOR

CPMUG volume 24 contains a version of RATFOR (RATional FORtran), a preprocessor for Fortran source programs. It permits structures such as IF..ELSE, WHILE, REPEAT UNTIL, FOR..NEXT, BREAK and INCLUDE and generates standard Fortran statements. A Z80 version of RATFOR, which runs faster, is on CPMUG volume 49 along with the full source code, documentation, and demo programs. The programs produced with RATFOR must be compiled with Microsoft's Fortran-80 compiler.

FORTH And Other Threaded Languages

Threaded languages employ a block structure allowing the user to define extensions to the language. The versions in the public domain are all interactive, use RPN (Reverse Polish Notation), and generate fast, compact object. For these reasons these languages are very popular with micro hackers and have produced many versions in the public domain.

FORTH11

There are six versions of FORTH. FORTH11, which is actually Fig-FORTH V1.1, is found on SIG/M volume 13. Although documentation is included, the author recommends that the user obtain a copy of the Fig-FORTH manuals first. SIG/M volume 70 contains an upgraded version of this program called FORTH110, in source form. FORTH130 is another version intended primarily for Apple-CP/M systems, but will run on many other systems, and is on SIG/M volume 116.

SMFORTH

SMFORTH is a minimal version of FORTH found on SIG/M volume 67. The source code version is provided without documentation. F83, another popular 8080 version of FORTH with documentation, is on SIG/M volume 154. An enhanced version comes on volume 204, and a 68000 version is located on volume 205 along with a BASIC Compiler written in F83. SIG/M volume 150 contains a version of FORTH for Z8000-based machines. *(Editor's note: Micro C's Bruce Berryhill is a FORTH fanatic, and he couldn't stand the idea that we didn't have a complete FORTH 83—since it has a built-in editor and all—on a Kaypro disk. So now we have one.)*

STOIC

STOIC, a FORTH-like language, is included on CPMUG volume 23, and for many applications produces even more compact code than FORTH. PISTOL, written in BDS-C, evolved from FORTH and STOIC and is located on SIG/M volume 59.

LISP

LISP is the most popular language used in writing AI (Artificial Intelligence) programs. It is based on list processing. The versions in the public domain are limited implementations but will give the user a good feel for the language.

SIG/M volume 71 contains a version written in Pascal. However, no documentation is included. INITLISP, on SIG/M volume 148, is also written in Pascal and includes documentation. XLISP, complete with documentation and examples, comes on SIG/M volume 118 with a CP/M-86 version on SIG/M volume 153.

Another example of LISP is on SIG/M volume 166 and is written in the REC language, described later. Several LISP program examples are included on the disk.

Other Languages

ACTOR is a string processing language found on CPMUG volume 4. It includes a complete manual and sample programs.

PILOT is used to write CAI (Computer Assisted Instruction) programs, and is located on CPMUG volume 12. The documentation was published in Dr. Dobb's Journal in April and May 1977.

SAM-76, on CPMUG volume 34, is a very powerful macro and string processing language that can be extended somewhat.

SIG/M volume 43 includes PIDGIN, a systems programming language useful for writing a compiler, and TINCMP, a special purpose compiler written in PIDGIN.

META is a CP/M-86 systems programming language and a version of TINCMP written with META as an example. Both are located on SIG/M volume 159. The documentation, however, is in Dr. Dobb's Journal of August '81. Versions of META written in FORTH-83 are on SIG/M volumes 204 (CP/M-80) and 205 (CP/M-86). An 8080 version of META4 is on SIG/M volume 208.

REC (Regular Expression Compiler) along with a very comprehensive subroutine library, is located on SIG/M volumes 213, 214, and 215 (an earlier version comes on volumes 164, 165, 166, and 167). REC includes translators to convert programs written in REC into other languages such as C.

PLO, a Wirth PL/O compiler, and some sample programs are on SIG/M volume 163.

And, as if all this weren't enough, there is a version of ye olde COBOL on SIG/M volume 209. *(Editor's note: Don't let this get out—someone might try to use it!)*

In Conclusion

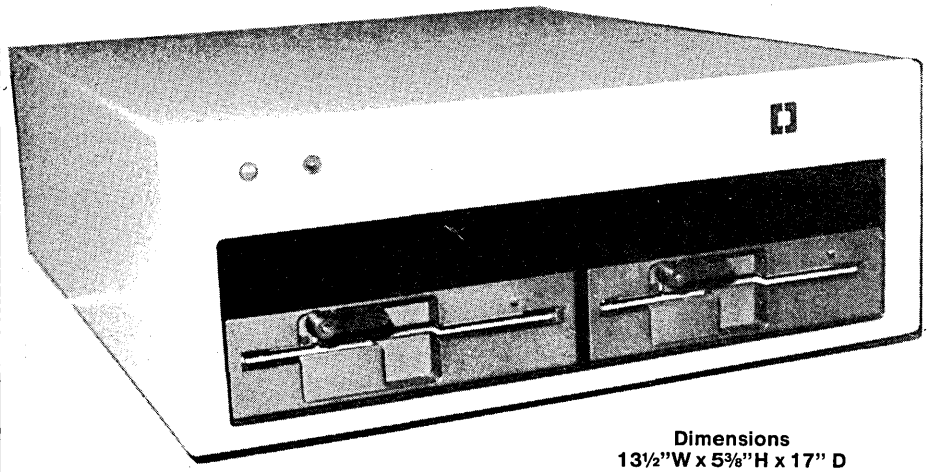
I think there are enough languages in the public domain to keep a computer language addict in a state of euphoria for months. In all, I count nearly 40 language interpreters and compilers. It is

interesting to note that the PC/MS-DOS public domain libraries contain fewer than 10 high level languages, even though the amount of overall software is greater.

I suggest you obtain copies of the CP MUG, SIG/M and PC/BLUE (for those interested in MS-DOS) printed catalogs (see below). Also, both SIG/M and PC/BLUE furnish an information disk (labeled "Volume 0") which contains an up-to-date listing of all the software in their libraries along with a program to help you find which particular volume contains the program you want. The disk also contains a listing of the clubs across the country that distribute the disks.

Order the SIG/M and/or PC/BLUE printed catalogs (each $3, $4 foreign) or Volume-0 information disk ($7, $9 foreign) from: SIG/M-PC/BLUE, Box 97, Iselin NJ 08830. The CPMUG printed catalogs can be ordered from the New York Amateur Computer Club, Box 106, Church Street Station, NY NY 10008, or call their hot line, (212) 864-4595. The CPMUG printed catalog comes in several volumes, and you will have to call or write for the price.

■ ■ ■



Dimensions
13½"W x 5⅜"H x 17" D

FULL 90 DAY PARTS & LABOR WARRANTY

Colonial Data
SB-80-II

COMPLETE SYSTEMS!
4 MHZ Z80A CPU
64K RAM (EXPAND TO 128K)
DUAL 5¼" DISK DRIVES
2 SERIAL PORTS
1 PARALLEL PORT

MODEL IIA - 2/380K DSDD DRIVES	**$ 895.00**
MODEL IIB - 2/780K DSQD DRIVES	**$ 995.00**
MODEL IIC - 10 Meg H/Disk w/380K Floppy	**$1895.00**
MODEL IID - 10 Meg H/Disk w/780K Floppy	**$1995.00**

* **SB80-II** (Board Only) Wired and Tested **$ 298.00**

SB80-II SYSTEM OPTIONS	**PRICE**
All Models:	
Additional 2 Serial Ports (installed)	**$ 60.00**
Additional 64K Memory (installed)	**$ 60.00**
CP/M 2.2 Operating System w/Manual	**$ 75.00**
CP/M 3.0 Operating System w/Manual	**$275.00**
Models IIA and IIB only	
SASI Interface Option (installed)	**$ 35.00**
XEBEC Hard Disk Controller	**$290.00**

PICK YOUR SYSTEM! CHOOSE YOUR OPTIONS!

The Original "Better Board" SB80 from Colonial Data has just gotten BETTER YET! More on-board features like SASI interface circuitry, 4 Channel Counter Timer, Memory Expansion to 128K and MORE! All made to run the most popular programs under the CP/M Operating System.

Z80™ — ZILOG
CP/M™ — DIGITAL RESEARCH (Calif.)

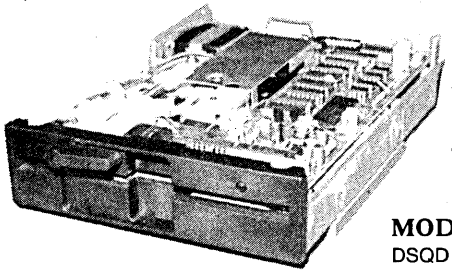
Colonial Data Cabinet Kits (Build Your Own Systems!)

MODEL CKP1
Metal Cabinet will accept Single Boards up to 12"x16" complete with fan, RFI filter with power switch, front panel and mounting for two 5¼" half-height floppies. Pre-punched for 4 DB25 and 1-50 pin. Includes 60 watt power supply with ±12V, + 5V. **$225.00**

MODEL CKP2
Same as Model CKP1, but includes 90 Watt Power Supply to power half-height Winchester 5¼" hard disk and single 5¼" half-height floppy. **$425.00**

MODEL CKX
Cabinet only with fan, RFI filter, power switch, prepunched connector holes. No Power Supply. **$139.00**

Shugart 5¼" Disk Drives



MODEL 455
DSDD Floppy — **$195.00**
Over 380K Capacity — 2 For **$370.00**

MODEL 465
DSQD Floppy — **$225.00**
Over 780K Capacity — 2 For **$430.00**

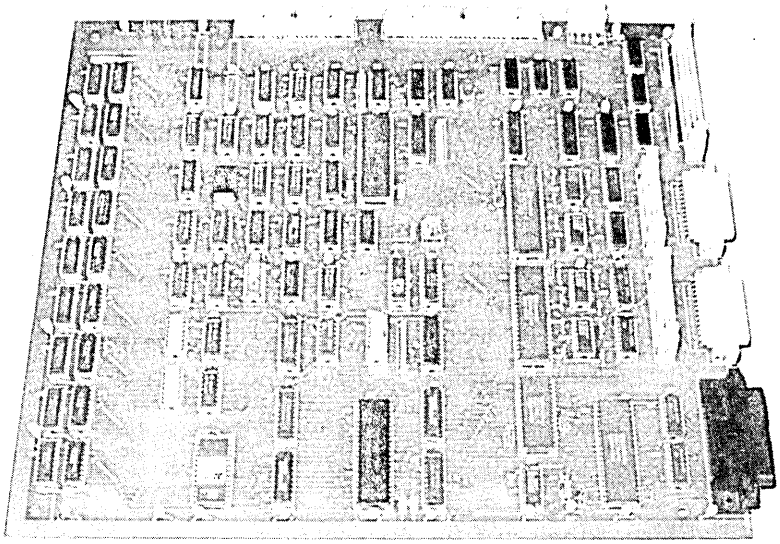
O.E.M. Quantity Inquiries Invited! Call or Write for Details!

HOW TO ORDER:
Call or Write Colonial Data—Orders paid with bank card or cashier's check are shipped within 3 working days. Allow 3 weeks for Personal Checks.
Shipping: Add 2% (up to a maximum of $20.00) for UPS (Brown) shipping within the United States.

Colonial Data Services Corp.
80 Pickett District Road,
New Milford, Conn. 06776
Telephone (203) **355-3178**

Colonial Data SB-80-II

The "Better Board" Gets BETTER!



$298.00
ASSEMBLED & TESTED
NOT A KIT!

Standard Features Include:
- 4MHz Z80A Processor
- 64K Memory (Expands to 128K)
- Supports 8" and 5¼" drives simultaneously
- Auto Density Selection (single/double/quad)
- 2 Serial Ports/1 Centronics Parallel Port

For the past 4 years, Colonial Data has been supplying thousands of its original SB80 "Better Board" computers to the O.E.M. market. Now, the "Better Board" is even BETTER with the introduction of the **SB80-II.** More Standard Features at a NEW LOWER PRICE!

SIZE: 12" x 13"
POWER: +12V, −12V, +5V
WARRANTY: 90 days Parts and Labor

■ 4MHz Z80A CPU WITH NO WAIT STATES
Enhanced **BIOS** uses mode 2 interrupts.

■ 64K MEMORY STANDARD (EXPAND TO 128K)
Parity checked RAM utilizes 4164
Up to 8K of EPROM (4K is standard)
Accepts Pin-compatible EPROMS from 2716 thru 2764

■ NUMEROUS FLOPPY DISK STORAGE OPTIONS
Uses the advanced NEC765A controller chip
Supports 8" and 5¼" drives simultaneously
Automatic Density Selection (Single/Double/Quad)

■ 4 SERIAL I/O PORTS (2 STANDARD)
Software selectable baud rates to 19,200
Uses Z80 SIO/0 and SMC 8116 baud rate generator.

■ Z80A PIO PARALLEL PRINTER INTERFACE
Centronics compatible printer interface is Standard!

■ SASI HARD DISK INTERFACE CIRCUITRY ON-BOARD!
SASI interface provides access to hard disk.
Compatible with XEBEC and other controllers.
Just populate with TTL and add connector.

■ 4 CHANNEL COUNTER TIMER-STANDARD!
Allows custom software applications.
User accessible 125Hz interrupt.
Provides system date/time clock.
4th Channel used for the NEC 765 interrupt.

■ CP/M 2.2 BIOS ENHANCEMENTS
Allows flexible device assignments, baud rate selection, automatic density detection, a system date and time clock and hard disk support.

■ OPTIONAL CP/M 3.0 (CP/M PLUS)
Extensive disk buffering speeds system throughout.

Z80™ — ZILOG **CP/M™ — DIGITAL RESEARCH** (Calif.)

SB80-II SYSTEM OPTIONS:	PRICE
ADDITIONAL 2 SERIAL PORT (INSTALLED)	$ 60.00
ADDITIONAL 64K MEMORY (INSTALLED)	$ 60.00
SASI INTERFACE OPTION (INSTALLED)	$ 35.00
XEBEC HARD DISK CONTROLLER	$290.00
CP/M 2.2 OPERATING SYS W/MANUAL	$ 75.00
CP/M 3.0 OPERATING SYS W/MANUAL	$275.00

HOW TO ORDER:
Call or Write Colonial Data—Orders paid with bank card or cashier's check are shipped within 3 working days. Allow 3 weeks for Personal Checks.
Shipping: Add $5.00 for UPS Shipping (Brown) Within the United States.

MasterCard VISA

Colonial Data Services Corp.

80 Pickett District Road,
New Milford, Conn. 06776
Telephone (203) **355-3178**

Interrupt Driven Serial Print Driver

By D.L. Carlyle

1107 Horseshoe Lane
Blacksburg VA 24060

With the recent addition of a Dynadisk to my BBI, I found myself printless. That is, the printer which used to receive data via parallel port B was left with a dangling IDC connector and no header to plug it into. Since Mr. Data (or perhaps Mr. Oki) was good enough to provide both a parallel and serial interface, I purchased a Z80A Dart (and the rest of the serial I/O components) and implemented the SIO option.

Figure 1 shows a segment of .MAC source for a serial output device connected to serial port B (the A port does not support vectored interrupts). Figure 2 is a brief description of the required SIO option jumpering. Figure 3, as if you haven't seen it often enough, is a step-by-step description of how to modify and install the SIO driver into your CBIOS, and how to install the new CBIOS into your system.

I was unable to find a Z80B DART, although I'm sure I've seen them advertised somewhere. The Z80A DART has worked reliably on my 5MHz BB for the last few weeks, however. (The Z80A DART is identical to the Z80A SIO/0 except that the DART does not support synchronous communications. In this application you can use either.)

A word of caution: I replaced the parallel print driver with the following SIO driver which caused the CBIOS to grow by about 12 or 13H bytes. When I overlayed it onto the rest of the CP/M system and SYSGENed it back onto the diskette, I found that on reboot, the tail of the new CBIOS was bumping into track two, where the directory lives. In addition, the ASCII string comprising the sign-on message (which uses a '$' as an end of message semaphore) became several hundred bytes long, making for a very exciting sign-on message.

The remedy for this was to edit out every non-essential character in every ASCII string for such things as "drive not ready" and so forth. This shrunk (sounds like a programming term . . .) the CBIOS back to its original size and eliminated the alphabet soup sign-on. Incidentally, my CBIOS orgs at EA00. Those who org at E800 might not have this problem.

■■■

Figure 1 - List Device Driver For The SIO B With Interrupts Added

```
;
;       SIO B EQUATES
;
SBDATA  EQU     05H             ;SIO B DATA PORT
SBCNTRL EQU     07H             ;SIO B CONTROL PORT
BAUDGN  EQU     0CH             ;BAUD GEN FOR SIO B
BAUD    EQU     07H             ;07H=1200 BAUD
INITCNT EQU     0BH             ;LENGTH OF INIT TABLE
RTBEI   EQU     00101000B       ;RST XMT BUFR EMPTY INTRPT
;
;       INITIALIZATION BYTE TABLE
;
INITBT: DEFB    00011000B       ;CHANNEL RESET
        DEFB    04H             ;POINT TO WR4
        DEFB    01000101B       ;16X CLK, 1 STP BIT, ODD PARITY
        DEFB    03H             ;POINT TO WR3
        DEFB    00100000B       ;AUTO ENA FOR CTS-BUSY HANDSHAKE
        DEFB    05H             ;POINT TO WR5
        DEFB    00101000B       ;7 BITS, TX ENABLE
        DEFB    02H             ;POINT TO WR2
        DEFB    00H             ;SIO B INT VECTOR IS (FF)00
        DEFB    01H             ;POINT TO WR1
        DEFB    00000010B       ;RX DISAB, TX INT ENABLE
;
INTPND: DEFS    1               ;INT PENDING FLAG
;
;       INITIALIZE BAUD RATE GEN FOR SIO B
;
;
OPNPRT  EQU     $
        LD      A,BAUD          ;SETUP FOR 1200 BAUD
        OUT     (BAUDGN),A      ;SEND TO SIO B BAUD GEN
;
;       INITIALIZE SIO B FOR TRANSMIT TO PRINTER
;
INIT:   PUSH    HL
        PUSH    BC
        LD      HL,INITBT       ;POINT TO FIRST CONFIG BYTE
        LD      C,INITCNT       ;MOVE RIGHT NO. OF BYTES
INITA:  LD      A,(HL)
        OUT     (SBCNTRL),A     ;INIT SIO B FROM INITBYT TABLE
        INC     HL
        DEC     C
        JR      NZ,INITA
        LD      HL,SIOINT       ;INTERRUPT DEST ADDRESS
        LD      (0FF00H),HL     ;STORE AT VECTOR
        LD      HL,PRTCHR       ;CPM ENTRY
        LD      (OVECTR+4),HL   ;ALL SUBSEQUENT ENTRIES SKIP INIT
        POP     HL
        POP     BC
        LD      A,0FFH          ;NON-ZERO TO A
        LD      (INTPND),A      ;DECLARE INT NOT PENDING
;
;
PRTCHR: LD      A,(INTPND)      ;INT PENDING
        OR      A
        JR      Z,PRTCHR        ;IF YES WAIT TILL NOT
        XOR     A               ;ZERO A
        LD      (INTPND),A      ;SET PENDING BACK
        LD      A,C             ;SET CHAR
        OUT     (SBDATA),A      ;SEND CHAR TO SIO B DATA PORT
        RET
;
;       SIO B INTERRUPT HANDLER
;
SIOINT: PUSH    AF
        LD      A,0FFH          ;ANY NON-ZERO OK
        LD      (INTPND),A
        LD      A,RTBEI         ;MAKE SURE A STRAGGLING
        OUT     (SBCNTRL),A     ;INTERRUPT DOESN'T GET YOU
                                ;AFTER LAST CHAR IS SENT
        POP     AF
        EI
        RETI
;
```

Figure 2 - SIO B To Printer Wiring Table

Note: The J3 header pins are not designated with odd pins in one row and even pins in the other; the first row has pins 1 through 13 while the second row has pins 14 thru 26.

SIO B HEADER J3	PRINTER (typical)
J3-2 (TxDb)	3 (RD)
J3-5 (CTS)	11 (SSD)*
J3-1&7 (GNDS)	1&7 (GNDS)

*SSD=Supervisory Send Data; equivalent to NOT BUSY.

Now jumper pin 7 to 8 and pin 19 to 20 on JB-5. do whatever you have to do to make your printer accept serial data (I had to connect DTR-pin 20 to DSR-pin 6 on the Okidata) and the hardware steps are complete.

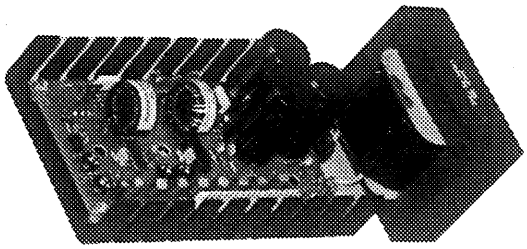
Figure 3 - Creating And Installing a Modified CBIOS

```
1. Create a .HEX file of the new, full CBIOS:

     a) Assemble  CBIOSn.MAC after edits where n is the  revision
        level of the CBIOS:

          M80  CBIOSn,CBIOSn=CBIOSn creates a .PRN and .REL  file
          from a .MAC file.

     b) Link the new CBIOS to create a .HEX file:

          L80  CBIOSn,CBIOSn/N/X/E.  The  message  "Origin  above
          loader memory,  move anyway?  (Yes or No)" will appear.
          Type Y.

2. SYSGEN

     a) Enter A as source drive to bring in a CP/M system image @
        900H.  When  the destination drive prompt comes  up,  hit
        return to leave the system image in memory.

3. SAVE 34 CPM64.COM

     a) This saves 34 (base 10) 256 byte segments on disk.

4. DDT

     ICPM64.COM
     R (Read an image of the SAVEd CP/M into the TPA)
     ICBIOSn.HEX
     I3580 (Overlay a copy of the new CBIOS over the old CBIOS)

          Note:  3580  is a load offset used to  overlay
          the  new CBIOS at the correct memory  address.
          If your CBIOS orgs at other than EA00, use the
          DDT hex command to find the correct offset:

          h1F80,org of your CBIOS (ie,EA00):  the  second
          set  of  four  digits disolayed by DDT  is  the
          offset.

     Use  DDT  Disolay  to  make  sure  that  the  new  CBIOS  has
     overlayed the old CBIOS at the correct address.

     ^C (Warm Boot)

5. SYSGEN

     a) Hit return in response to the source drive prompt.  Hit A
        in  response to the destination drive prompt to  write  a
        bootable copy of CP/M on disk.


The codes for selecting baud rates other than 1200 are listed  in
the  documentation package that came with your BB.  Edit the .MAC
source accordingly,  follow the steps listed above and you should
be up and running with a fully interrupt driven print routine.
```

FERGUSON ENGINEERING

817-640-0207

820-1 Schematics, 18" x 24" set of six, very legible	$ 18.00
STD BUS connector for BBII	12.50
Crydom S312 solid state relay, 120 VAC,2 A, 0 volt switch	10.00
Complete 820-1 Documentation package	15.00
820-1 CBIOS source, ROM monitor source, and 5.25"—or—8" formatter all in source code, M80* compatible, 8" SSSD	35.00
820-1 dual drive disk cable for two 5.25" drives w/power conn	20.00
820-1 dual drive disk cable kit for eight inch drives	45.00
Composite video adapter kit for 820-1, produces true RS-170 campatible video	25.00
Serial console monitor program for the 820-1, gives your 820-1 the PFM style serial console, M80 source, 8" SSSD	35.00
820-1 power connector w/pins	4.00
820-1 / BIGBOARD-1 ASCII keyboard, w/cable, NEW!	95.00
BIGBOARD-1/11 dual-drive 8" disk cable, 17" long	35.00
Assembled and tested Xerox 820-1 computer boards	125.00
Bigboard dual 8" drive cable	30.00
Bigboard serial I/O cable	15.00
AMP 8" drive AC connector w/pins	.75
AMP 8" drive DC power connector w/pins	1.50
AMP 5.25" drive DC power connector w/pins	1.10
CP/M Primer (Xerox CP/M handbook)	7.50



IMPROVED!

UNINTERRUPTABLE POWER SUPPLY

UPS 5-200 **$195.00**

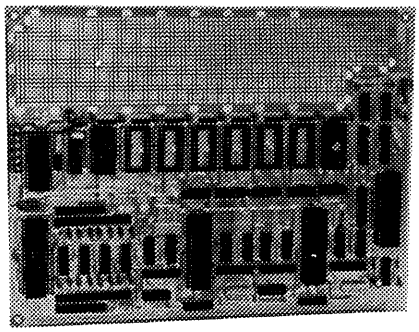
+5V @ 8.5 AMPS, +12V @ 7.5 AMPS — 12V @ .5 AMP
TWO +24 VOLT OUTPUTS @ 1.5 AMPS EACH (2A PEAK)

This supply is designed to power entire computer systems with Winchester and floppy disk subsystems. When the AC power fails, power can be maintained by a single 24V battery. The switchover time from AC to battery is zero, power is never interrupted.

24V 4.5 AH Panasonic LCR series battery $65.00

- High Reliability (1 year limited warranty)
- High Efficiency (75% typ)
- New design is very quiet.
- Quick response OVP
- Fold-Back current limiting
- Drop-out immunity for up to 250 mS (without battery)
- Uninterruptable (with addition of battery pack)
- Built-in battery charger
- Very Small 3.5"H, 10.0"L, 4.25"W

GENERAL PURPOSE INDUSTRIAL CONTROLLER



LB-1

$75.00

Bare Board & Doc

- Uses Z80,Z80A, or Z80B family of components
- Dual RS232 serial I/O W/Current loop option on the "A" channel (both strappable as terminal or modem in the RS-232 mode)
- 16 programmable Baud rates on each serial channel
- 32 bits of buffered parallel I/O assignable on 4 bit boundaries, W/Strappable Handshake Polarity (2xZ80-PIO)
- 4 Channels of counter-timer, 4 inputs, 3 outputs
- Accommodates 16K mix of RAM or ROM or EEPROM on 2K boundaries in 2716 or 2016 or Xicor EEPROM
- 2.3" x 10" sea of holes prototyping area on .1" grid
- All I/O is mode 2 interrupt compatible
- Watchdog timer for long-term unattended operation

ROM monitor and source listing ..... $ 35.00
Assembled and tested 4 Mhz LB-1 ..... 250.00

FERGUSON ENGINEERING
817-640-0207
P.O. BOX 300085 ARLINGTON, TEXAS 76010

TERMS: Master Card/VISA add 3%, money order, certified check, UPS cash COD. Allow 2 weeks for personal or company checks to clear. Shipping/handling extra. Texas order add 5% tax.

The Xerox Column

By Mitchell Mlinar

1013 W 210th
Torrance CA 90502

Being wrong can be a wonderful experience. I am referring to the article in issue 21 (December) in which I said there was no manual available for the 820-II. The deluge of phone calls and the truss I have recently purchased for my mailman say otherwise. So, despite my homework (I did know of an internal manual, but not that it had been released), I was wrong. Here is the best phone number and re-order number I could come up with: 1-800-527-1922, ask for the Xerox 820-II Technical Manual, Reorder #9R80758.

It turns out that all the mail I received was from -II users who are also Xerox employees. And I don't feel bad at all about being wrong because ANY mail is better than no mail. Anyway, there are a lot more -II users out there than I had expected, so I will try to share space a little more in the future between the -I and -II.

Xerox 820-II Technical Manual

The -II manual is definitely better than the -I version. (A generous X'er donated a spare manual to the "Let's Straighten Mitch Out" club, so I now have one.) It is 400 pages of VERY small print in a 3-ring binder. For once, being near-sighted actually helped, since I could read the manual when I held it 2 inches away. You 20/20 types will need a magnifying glass. Make no mistake, though. The print may be small, but the copy quality is unsurpassed.

The manual starts with a description of the 820-II and the 16/8 (the 16-bit upgrade for the -II), along with a theory of operation. After that come Zilog reprints for the Z80, SIO, PIO, and CTC. All schematics are included, which might interest those who have the 16/8, or are thinking about building their own 8086 card. And that's just the first part of the manual!

A complete description of software follows the hardware. I finally know what those monitor software 'hooks' are for (I wondered about them last time). They are used to allow processing during I/O waiting time. The manual contains the source for the BIOS and monitor (V4.01) as well as a discussion of peripherals such as printers and keyboards. It also has a reprint of the Western Digital floppy controller data.

There is a lot of information in the -II manual that would help the -I owners though I would recommend buying the -I manual first.

ROM 5.0?

Xerox has done it again. Several readers have told me that my information on the -II in the December issue was wrong. It turns out they have a new ROM—version 5.0—now being shipped from the surplus outlet in Dallas. As soon as I find out what the differences are, I will let you know.

SWP Vrs Emerald Microware

There are two big decisions to make when you have a single density 820-I: do you go double density, and if so, which package do you get (I counted four in a recent Micro C). A real dilemma can occur if you have scraped to buy a board from Dallas (or BG Micro) for $50, only to find that complete double density packages cost three times as much! In an earlier column, I reviewed the SWP hardware for 8" systems. Another complete DD package is also available from Emerald Microware.

The Emerald Microware package I am reviewing, which includes the XPRO ROM set and the X120 DD controller board, comes completely assembled for $180.

A kit sells for a little less and assembly instructions are thorough. I like the comment up front which warns kit buyers that they need some proficiency in soldering.

It mentions that those who feel unsure should send the package back for a refund. You don't need much technical skill, but you should have access to an oscilloscope for the final alignment. Brian Garrison of Emerald Microware offers technical support over the phone if you have problems.

Since you really need the ROMs to run the package, the price difference between the kit and the assembled and tested version is only $15 (unless you like "electronic safaris" to find parts for your board). Also, I recommend you request the ROMs in a set of 2716s instead of a single 2732; no modifications are required to your board if you use 2716s.

Even if you buy the assembled version, you need to do some soldering; a small 14-pin chip adjacent to the disk controller 40-pin socket must be removed. Just use solder wick on the underside of the 820-I board, and the chip should pull right out. If that doesn't work, clip all the pins on the top side and remove each pin separately with a soldering iron and needle nose pliers. You will still need to solder wick afterwards to clear out the 14 holes.

I have tested the board on a 5.25" system with single-sided drives and it runs quite well. I have even run my single density 8" system on it. However, one thing the ad in Micro C does not mention is that in order to run 8" double density you need a 4 or 5 MHz clock.

Brian details a 5MHz upgrade which is much easier than the standard 4MHz upgrade.

Unfortunately, I have pre-stuffed boards from Dallas. Even though I'm technically inclined, I will not remove three 40-pin chips from a multi-layer board. (Not while the board is still working, that is.) If you have a higher clock rate -I or socketed chips, then running DD on an 8" system should not be a problem.

The Verdict:

I have two double density packages working on 820-Is: SWP and Emerald Microware. SWP has an excellent package for 8" systems. Emerald Microware is a comparable package for 5.25" users. In fact, the SWP package was designed for 8" and later modified for 5.25."

In comparison, unless you increase your -I's clock rate, the Emerald Microware package is limited to single density on 8". However, if you speed up your system you can run both 5.25" and 8" double density together.

There are other packages out there I will never get a chance to try, but of these two, I recommend SWP for 8" users and Emerald Microware for 5.25" users.

Turn Off That Screen!

Back in last year's August issue, Jim

(Continued on Page 23)

XEROX COLUMN

(continued from page 21)

Mayhugh added a little hardware and software to shut off the screen. It turns out that you really don't need any hardware.

On the system control port of the 820-I at output port 1CH (which handles drive select, ROM/RAM select, keyboard status, and the unimplemented bell output), there is a little-known line called Display Character Set (bit 6). This bit is set low during initialization and is never touched again. So only the lower half of the 2716 character ROM is used. What about the other half? There is nothing in the other half. And that is exactly what happens to your display if you set bit 6—everything disappears.

I have modified my monitor (which has a software real time clock) to turn off the display after 10 minutes without video activity; it simply sets bit 6 of the system port. The next character sent to the screen resets the bit and the display reappears. Except for the small software overhead, the hardware could not be cheaper. You can use this with both the 1.x and 2.x ROMs.

820 Focal Point

Many people write to me about their projects, problems, and suggestions for future articles. As a columnist, my goal is to disseminate as much of that information, as well as my own experience, as I can.

Unfortunately, the mail bag is getting pretty big and I am not very good at answering letters. On the other hand, those who manage to get me on the phone spend a lot of money because I like to talk. However, that is all going to change.

Before the next article appears, I will have moved to a larger house, and my 20-Mbyte system will become a 24-hour RQP/M system with everything I can get for the 820-I, 820-II, and Big Board I.

With the BBS, you can expect an answer from me in, at most, two days. Further, questions you ask could be answered by anyone. There are plenty of knowledgeable 820 users out there. Of course, you can always wait until the SOG. I am going to be there, and I hope you will, too. We will have a meeting for the 820 users. (Until next time, keep those heads clean.)

■ ■ ■

C'ing Clearly

By Fred Scacchitti

25 Glenview Lane
Rochester, NY 14609

In the August '84 Issue of Micro C (Issue 19), Tony Ozrelic treated Small-C a little harshly. He attacked the version on Micro C's User Disk #17 without realizing that Small-C has changed.

When I received that same Small-C disk a little over a year ago, I went through much the same pain and agony that Tony described. I don't know why I didn't give up on it, I guess there's just something about having your very own C compiler that you can hack, change, modify, and upgrade. The fact that it is in the public domain didn't hurt.

Before I go any further I must give one heck of a lot of credit to Ron Cain who wrote Version I, J. E. Hendrix for the massive upgrade to Version II—and Glen Fisher, Bill Danielson and Bill Randle for their work on the runtime module. I have worked with a variety of C compilers. Currently I'm using a Xerox 820-II running 4 Mhz. (Soon to be upgraded to a 16/8 - I hope!)

Now, On With The Story

The first thing I did was upgrade the compiler for use with Microsoft's M80/L80 package. (I agree with Tony that every compiler needs a macro assembler and linking loader.)

Then I began to build up library functions as well as documentation for my version of the compiler. In May & June of '84 Dr. Dobbs Journal published Jim Hendrix's latest library, so I purchased his latest software along with his "Small C Handbook." It's available from Reston Publishing. (Micro C will try to stock this book.)

Jim Hendrix and I differ somewhat on our philosophy for the runtime and I/O library. My package is optimized for a CP/M environment, he developed his with a significant UNIX flavor. I guess it's all a matter of taste.

Using Jim's compiler and library as guides I went to work on the package. The result is a reasonably fast compiler which produces small object files, and a library of over 100 C functions.

The compiler disk contains all the files you need to make the compiler do its thing, the compiler source, twelve program source files,and a step by step pro-

(continued on page 27)

Figure 1 - Time.c Program

```
/

 *
** time.c   Time set/display Program
**
**
**
**      Written in Small-C  (Version 2.09)
**
**      Time Utility for the Xerox 820
**
**      time p   --->  prints current time and date
**      time s   --->  enter time set routine
**
** This program is designed to be used with an auxiliary printf routine
** (prntf) to minimize program size. (printf may be substituted for prntf)
**                                  (fas)
**
*/

#include <stdio.h>

#define CLEAR 26
#define BACKSP 8
#define SPACE 32
#define VERTAB 11
#define LINSIZ 18
/*#define MEM 65369             pointer to first time byte for the 820-I   */
#define MEM 65366           /*  pointer to first time byte for the 820-II  */


char array[LINSIZ];
int num[6];
int lolim[] = { 1, 1, 84, 0, 0, 0};
int hilim[] = {31, 12, 99, 23, 59, 59};


main(argc,argv) int argc, *argv; {

char *entry;

argv++;     /* point to option */
entry = *argv;  /* set up 2nd indirect pointer   *entry = **argv */

   switch (*entry) {
      case 'P':
        if(!gudtim(gettim())) settim();
        if(gudtim()) prntim();
        break;
      case 'S':
        if(settim()) prntim();
        break;
      default:
        argc = 0;
        break;
   }
   if (argc != 2){
    prntf(" usage: time p <CR> prints time and date\n");
    prntf("        time s <CR> to set time and date\n");
   }
}

    prntim(){   /* Routine to print time and date */

    char *clkmem;

       putchar(VERTAB);

       clkmem = MEM;
       clkmem++;
        switch (*clkmem){
        case 1:
            prntf("January");
            break;
```

(Listing continued on page 29)

Only $95 with FULL SOURCE CODE!

Q/C

"... an incredible learning tool." *Byte*

For only $95, Q/C is a ready-to-use C compiler for CP/M with complete source code. Here's what *BYTE* (May 1984) said: "Q/C ... has a portable library and produces good code quality. If you want to learn compiler construction techniques or modify the standard language, Q/C is the obvious choice."

- Source code for compiler and over 75 library functions.
- Strong support for assembly language and ROMs.
- No license fees for object code.
- Z80 version takes advantage of Z80 instructions.
- Q/C is standard. Good portability to UNIX.

Q/C has casts, typedef, sizeof, structure initialization, and function typing. It is compatible with UNIX Version 7 C, but doesn't support long integers, float, parameterized #defines, or bit fields. Call about our new products: Q/C profiler, Z80 code optimizer, and Z80 assembler and virtual linker, all with full source code!

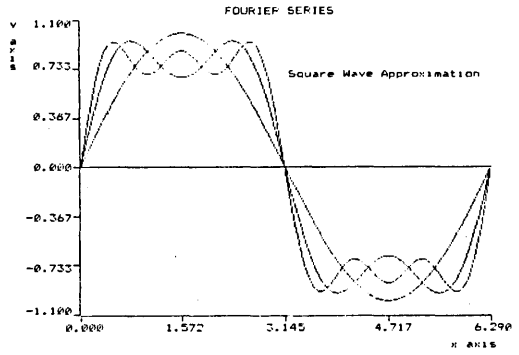
THE CODE WORKS
5266 Hollister, Suite 224
Santa Barbara, CA 93111
(805) 683-1585

Q/C, CP/M, Z80, and UNIX are trademarks of Quality Computer Systems, Digital Research, Zilog, Inc., and Bell Laboratories respectively.

GRAF 2.0 *$29.95*

GRAF 2.0 allows you to create amazing graphics on your dot-matrix printer. Features include:

* Automatic bar chart and line graph generation
* Automatic scaling and labeling of both axes
* Ability to plot floating-point data obtained from most spreadsheets, word processors, or languages.
* Extensive, 60-page illustrated User's Manual includes examples showing how to do graphics from SuperCalc, dBASE II, MBASIC, and Turbo Pascal.



System Requirements Any Z80 computer running 60k CP/M and driving an Epson, Gemini, or C. Itoh dot-matrix printer.

MSC Microcomputer Systems Consultants

301 North Harrison Street | CN 5279, Suite 228 | Princeton, New Jersey 08540

Terms Send check or money order for $29.95 + $5.00 s/h to MSC at the address above. You MUST state your computer and printer make and model. NJ residents add 6% tax.

The following are trademarks or registered trademarks of the indicated companies: CP/M – Digital Research, MBASIC – Microsoft, SuperCalc – Sorcim, Turbo Pascal – Borland International, dBASE II – Ashton-Tate, Z80 – Zilog, Keypro – Keypro Corporation

New!

THE LITTLE BOARD®

OEM — INDUSTRIAL — SCIENTIFIC

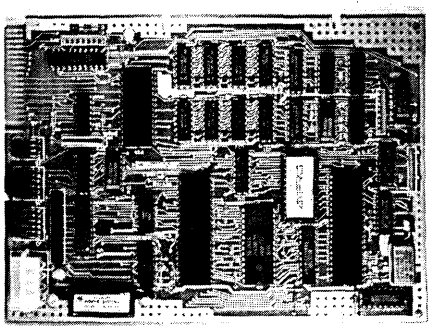
SECOND GENERATION SINGLE BOARD COMPUTER KIT!

4 MHZ Z80* CPU! 64K RAM! DOUBLE DENSITY!

FREE CP/M* 2.2!!! A $139 VALUE! A FREE 5-1/4 IN. CP/M 2.2 DISKETTE IS INCLUDED WITH EACH KIT.

A. & T. UNITS $319

PRICE CUT!



MINI-SIZE: ONLY 5-3/4 x 7-3/4 INCHES

$259.95 (COMPLETE KIT)

DISK DRIVE SPECIAL:
NATIONAL (PANASONIC) J551 DS-DD 5-1/4 IN. 40 TRACKS PER SIDE — $125 EACH (HALF HEIGHT) (48TPI) 2 FOR $240
MITSUBISHI #M4853 DS-DD 5-1/4 IN. 80 TRACKS PER SIDE — $130 EA (HALF HEIGHT) (96 TPI) 2 FOR $245
BOTH DRIVES WORK FINE WITH THE LITTLE BOARD!

FULLY SOCKETED! PERFECT MATE TO OUR ZRT-80 TERMINAL BOARD. THROUGH SPECIAL ARRANGEMENT WITH AMPRO COMPUTERS, WE ARE PLEASED TO OFFER THEIR LITTLE BOARD* IN KIT FORM.

FEATURES:

- 4 MHZ Z80 CPU!
- DOUBLE DENSITY (5-1/4 IN.) FLOPPY CONTROLLER
- 64K DYNAMIC RAM!
- CENTRONICS STYLE PARALLEL PRINTER PORT
- USES +5VDC @ .75 A. AND +12VDC @ 50MA
- TWO RS232 SERIAL PORTS
- SAME SIZE AS A MINI FLOPPY
- 2732 BOOT EPROM

Digital Research Computers
(OF TEXAS)
P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 225-2309

TERMS: Shipments will be made approximately two weeks after we receive your order. VISA, MC, cash accepted. We will accept COD's with a $75 deposit. Balance UPS COD. Add $4.00 shipping.
USA AND CANADA ONLY

ALL SALES SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. FREE COPY UPON REQUEST.

Z80 TM OF ZILOG. CP/M TM OF D.R.I.

(continued from page 25)

cedure for putting together a C Compiler Disk. This package is available as a user disk. (Editor's note, actually it's two disks. The compiler fills one—and the library, which normally fits on two disks, has been squeezed onto another.)

Recommendations

1. Purchase "The Small C Handbook."
2. Obtain reprints or back issues of Dr. Dobbs which deal with the Small-C Compilers.
3. Read over the document files that come on the disk with the compiler.
4. Experiment.

If you don't have M80 and L80, take heart. I recently received a Z80 version of Small C, but haven't had a chance to look it over yet. In the near future I hope to have a version available which will be compatible with the CWA Assembler/Linker/Librarian. (It's available from the Code Works for $35.)

I got hooked on C almost as soon as I started using it. Small C is an excellent way to get started and can be used very effectively. There has been a lot done with it, and I believe it will grow to be competitive with many of the commercial compilers. (Most of them began as Small-C.)

TIME.C

I wrote a time.c program for the 820/820-II. It is a good example of some of the features of Small C and shows how to interface to system memory locations. (The date and time values.) This program compiles, assembles, and links to produce an object file that's less then 4k. I use a modified printf() routine called prntf(). You can substitute printf() for prntf(). See Figure 1.

Editor's note: Fred very graciously agreed to do this column at the last minute after Tony bowed out. Fred's strong interest in Small-C is a big plus in the Micro C community. I'd like to see the Micro C community continue to expand this famous compiler. How about you?

□□□

C'ing CLEARLY LISTING

(continued from page 25)

```
        case 2:
            prntf("February");
            break;
        case 3:
            prntf("March");
            break;
        case 4:
            prntf("April");
            break;
        case 5:
            prntf("May");
            break;
        case 6:
            prntf("June");
            break;
        case 7:
            prntf("July");
            break;
        case 8:
            prntf("August");
            break;
        case 9:
            prntf("September");
            break;
        case 10:
            prntf("October");
            break;
        case 11:
            prntf("November");
            break;
        case 12:
            prntf("December");
            break;
        default:
        abort('?');
        }

        clkmem--;
       prntf(" %d, 19%d  %02d:%02d:%02d",*clkmem++,*++clkmem,*++clkmem,
                          *++clkmem,*++clkmem);

}

settim(){   /* Routine to set time and date */

int i;
char c, *clkmem;

    clkmem = MEM;
/* Display the time set header */

    putchar(CLEAR);
    prntf("Enter the time and date as dd/mm/yy hh:mm:ss\n");
    prntf("\n eg. for 1:45 P.M. on August 12, 1984\n");
    prntf("     Enter ---> 12/08/84 13:45:00\n");
    prntf("                dd/mm/yy hh:mm:ss");
    prntf("\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b");

/* Get the input string */
    i = -1;
    while (((c = getchar()) != CR) && i++ < 17)
        if (c == BACKSP)
            if (i == 0){
                i = -1;
                putchar(SPACE);}
            else
                i = i - 2;
        else
            array[i] = c;

/* Convert the characters to integer values */

    for (i = 0; i <= 18; i = i + 3)
        num[i/3] = ((array[i] - '0') * 10) + (array[i+1] - '0');

    prntf("\n\n");

/* Store the values to memory if they're valid */

    if (gudtim()) {
        for (i = 0; i <=5; i++)
            *clkmem++ = num[i];
        return(TRUE);
    }else{
        prntf(" Incorrect data entered - time not updated");
        return(FALSE);
    }
}

gudtim(){

int i;

   for(i = 0; i <= 5;i++)
    if (num[i] < lolim[i] || num[i] > hilim[i]) return(FALSE);
   return(TRUE);
}

gettim(){  /* Routine to get time data from memory to an array  */

int i;
char *clkmem;

    clkmem = MEM;

    for (i = 0; i <=5; i++)
            num[i] = *clkmem++;
}
```

End of Listing

SLICER Special April Prices!

Slicer Single Board Computer

	April Specials!
Assembled and Tested 8 Mhz	$895!
Full Kit	$715!
Easy Kit	$400!
Bare Board	$125!
New!! DOS KIT includes Bare Board, CPU and choice of MS-DOS or CP/M-86	$295!

The Slicer System Expansion Board

	April Specials!
Assembled and Tested	$650!
Full Kit	$450!
Memory Board Kit	$300!
3-Port Kit	$200!
Bare Board	$75!

The Slicer PC Expansion Board

Assembled and Tested	$600
Full Kit	$550
Easy Kit	$400
Not-So-Bare Board	$200

New!!

Western Digital 1002-SHD Hard Disk Controller

Assembled and Tested ..... Only $200!

New!!

The μ Slicer 188

Assembled and Tested ..... Only $700!

Slicer Enclosure System

Enclosure (alone)	$125
with 135W Power Supply	$245
with P.S. and two 80 track 5¼" Disk Drives	$695
10MB (formatted) 1/2 height 5¼" Hard Disk	$700
PC Compatible Keyboard	$150
PC Compatible Amber Monitor	$175

New!!

Super Enclosure System

Enclosure, Power Supply, 1/2 height
80 track floppy, 10 MB (formatted)
5¼" Hard Disk, Western Digital
1002 SHD Controller
All for Only ..... $1395!

New!! 8087 Math Co-Processor Board

(Requires removel of CPU socket)
This board requires a C-Step CPU and an 8 Mhz 8087. Kit with all parts except CPU and 8087 ..... $300!
8 Mhz 8087 ..... Call

Try our New SLICER Bulletin Board System 300/1200 Baud at (612) 788-5909

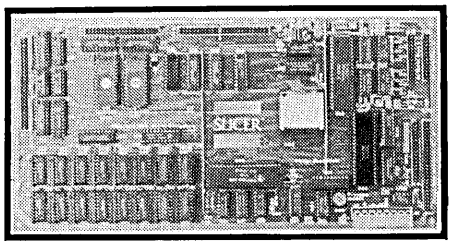
SLICER™

SLICER COMPUTERS INC.
2543 Marshall Street N.E.
Minneapolis, MN 55418
(612) 788-9481

PAYMENT METHOD: Mastercard, Visa, check, money order, or COD (certified funds or cash). Please allow 4 to 6 weeks for delivery ARO. Minnesota residents please add 6% Minnesota sales tax. Normal shipping within the continental USA (via standard U.P.S. service) is now included on all boards and software.

SLICER . . . Growing to meet your needs!

The Slicer

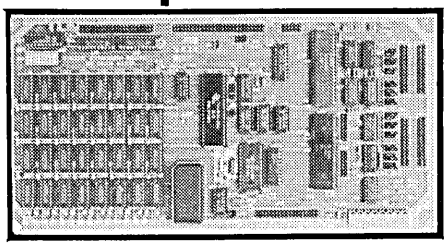


REAL 16-BIT POWER ON A SINGLE BOARD
Featuring the Intel 80186

- Complete 8 MHz 16-bit microprocessor on 6"x12" board
- 256K RAM, plus up to 64K EPROM
- SASI port for hard disk controller
- Two full function RS232C serial ports with individually programmed transmission rates 50 to 38.4K baud
- Software compatibility with the 8086 and 8088
- 8K of EPROM contains drivers for peripherals, commands for hardware checkout and software testing
- Software supports most types and sizes of disk drives
- Source for monitor included on disk
- Bios supports Xebec 1410 and Western Digital WD 1002 SHD controller for hard disks

Fully assembled and tested only . . . . April Special $895

The Slicer System Expansion Board



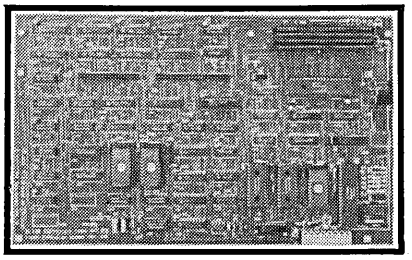
FOR EXPANDED MEMORY, ADDITIONAL PORTS, AND REAL TIME CLOCK

- Up to 256K additional dynamic RAM
- 2 RS232C asynchronous ports with baud rates to 38.4K for serial communication
- 2 additional serial ports for asynchronous RS232C or synchronous communication (Zilog 8530 SCC)
- Real Time Clock (with battery backup) for continuous timekeeping
- Centronics type parallel printer port

Fully assembled and tested only . . . . April Special $650

The Slicer PC Expansion Board

New!!



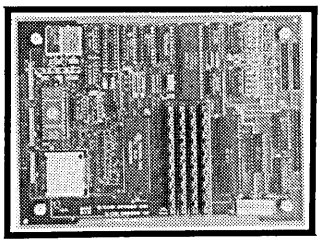
GIVES YOUR SLICER HIGH PERFORMANCE VIDEO CAPABILITY

- IBM compatible monochrome video
- Video memory provides 4 pages of text or special graphics capability
- 2 IBM type card slots for color video, I/O expansion, etc.
- IBM type keyboard port

Fully assembled and tested only $600

The μ Slicer 188

New!!



REAL 8/16-BIT POWER ON A SINGLE BOARD
Featuring the Intel 80188

- Complete microcomputer on high quality 5-3/4" x 7-3/4" multilayer board
- 256K RAM, plus up to 8K EPROM
- SASI port for hard disk controller
- Two full function RS232C serial ports with individually programmed transmission rates 50 to 38.4K baud
- Software compatibility with the 8086 and 8088
- 8K of EPROM contains drivers for peripherals, commands for hardware checkout and software testing
- Software supports 5-1/4" and 3-1/2" disk drives
- Source for monitor included on disk
- Bios supports Xebec 1410 and Western Digital WD 1002 SHD controller for hard disks

Fully assembled and tested only $700

SLICER™

SLICER COMPUTERS INC.
2543 Marshall Street N.E.
Minneapolis, MN 55418
(612) 788-9481

All products are available in several kit forms:

Operating Systems are:
CP/M-86 by Digital Research, Inc. $85
MS-DOS by Microsoft Corporation $175

MasterCard, Visa, Check, Money Order or C.O.D.
Allow four weeks for delivery. Prices subject to change without notice.

The Slicer Column

By Laine Stump

Micro C Staff

Dave, this column isn't quite the trip on the data bus to the CPU that the last column was. It is intended more for people who don't have a 16-bit machine yet, or who have just gotten one and have been paranoid about 8086 assembly language. I hope it will be read by a lot of 8-bitters who are scared of big bad 86. (Now, life in the fast Laine . . .)

Well, you finally did it. You just dropped your entire food budget for the spring of '85 on a cute little 16-bit machine that should really kick the diodes off your Z80. You bring it home, proudly set it up on the workbench, turn it on, and suddenly remember that you don't know a thing about 8086 assembly language. Your hands tremble as paranoia sets in. You muffle a low scream and reach for the CP/M-86 manual. HELP, MR. WIZARD, HEEELLLLLLPPPP!!!!

Eightysixophobia

Many people are initially scared by the apparent complexity of 8086 assembly language compared to Z80. It really is nothing to be frightened of. If you are used to writing in 8080 code, all you have to do is change the name of a few registers and add a CSEG or DSEG here and there.

Even the interface to the operating system is very similar. If you have been using Z80 code you will have to replace all your LDs with MOVs (I still forget quite frequently) and give up some of those old tricks with the alternate registers, but most of the same old instructions are still there (plus quite a number of new ones). Only the names have changed.

To help along those souls who are still troubled by the idea of learning a 'whole new language,' I have written a short CP/M-86 assembly language program (Figure 1) that shows the similarities between 8086 and Z80 programming, while also giving examples of some of the more powerful 8086 instructions that have no direct Z80 counterpart.

More, You Say?

MORE doesn't do anything complex; it just types a file to the console a page at a time. After each screenful, the message '– More –' is displayed. The program will not display another page until you type a key. This is similar to the program of the same name that comes with MS-DOS and UNIX.

This isn't much of a job, but it is enough to show how to use CP/M-86 to open a file, read from the file, output characters to the screen, and input characters from the console. MORE can be assembled with the ASM86 assembler included with CP/M-86.

I will examine each section of the file, pointing out unique features along the way.

Constants, Labels, And Variables

The first section of the program defines some frequently used constants. Although Z80 assembly language treats constants, labels, and variables identically, ASM86 thinks they are totally separate beasts. This becomes apparent when you discover that the instruction:

```
MOV     DX,ESC
```

has two different meanings depending on how ESC is defined. If ESC is a constant (defined with EQU), DX is loaded with the value ESC. If it is a label (defined by following it with ':') or a variable (defined by placing it at the beginning of a DW, DB, RW, etc.), DX is loaded with the contents of memory location ESC. To MOV the value ESC into DX if ESC is a label or a variable, you must use:

```
MOV   DX,offset ESC
```

OFFSET instructs the assembler to use a variable's 'offset address' within its memory segment as a constant. I know all this seems confusing, but it really does work out nicely, since most of the time you use the contents of a variable, not its address. Think about it for a minute, and you'll see that this is how constants and variables are handled in most high level languages. See now, if ASM86 is inconsistent, then so is Niklaus Wirth. (I'm not telling him, are you?)

CodeMacros

The next section of MORE uses ASM86's CodeMacro facilities to define a new instruction called 'BDOS'. This instruction takes the form:

```
BDOS    FTNNUM
```

where FTNNUM is the number of the BDOS function you want to use. This 'macro instruction' expands into the following instructions:

```
MOV  CL,FTNNUM  ;put funct. # in CL
INT  224                ;call BDOS
```

This is how BDOS is accessed in CP/M-86. Any arguments that need to be sent to CP/M-86 are placed in the DX register. This is very similar to CP/M-80 where arguments are placed in the DE register and BDOS is called with:

```
LD      C,FTNNUM
CALL    5
```

All the function numbers supported under CP/M-80 perform the same operation under CP/M-86. There are also several new functions that I may discuss at some later time.

Where Am I ?

The next interesting thing to occur in the program doesn't. There is a CSEG statement to tell the assembler we are in the code section of the program, but where is the ORG statement? It turns out that CP/M-86 programs start at location 0 in the code segment, and ORG 0 is assumed by the assembler. Makes things easy, doesn't it?

The Stack

Most well written programs start by setting the stack pointer. This one doesn't. I chose not to set up my own stack because MORE doesn't use much stack space and SS:SP already points to a 96-byte stack in the CCP when the program begins. This avoids having to discuss all the clutter that occurs when setting up a different stack pointer. The clutter (trick code) is necessary because an interrupt occuring after the Stack Segment (SS) register has been set and before the Stack Pointer (SP) has been set could create catastrophic results. This is all explained in Appendix B of the CP/M-86 manual.

The Body

There are several instructions in the body of the program that deserve a close look. Notice, for instance, the CMP instruction that compares the contents of memory with a constant value. No more

clobbering the accumulator every time you want to compare two values!! Most of the 8086 instructions offer greater flexibility than those of the Z80. Any of the general registers can be used for arithmetic operations; you can even directly use a memory location as an accumulator. The only constraint placed on MOV and the arithmetic instructions is that at least one operand of the instruction must be either a general register (no segment registers) or a constant.

BDOS calls are made in several places using the BDOS instruction we defined earlier. Notice that when an address is needed (for instance the address of FCB, the File Control Block), it is placed in DX by using the 'offset' directive. It is sometimes useful to think of offset as meaning 'the address of.'

The TYPE128 routine contains two helpful instructions. The first is the LODS instruction. LODS moves a byte or word (depending on the operand type) from memory pointed to by register SI (Source Index) to register AL or AX. SI is then automatically incremented or decremented by the proper amount (1 for byte, 2 for word). There is no direct counterpart to this instruction on the Z80. The LDI instruction is close, but it moves only bytes and only from memory to memory (memory to memory moves are done on the 8086 with the MOVS instruction).

Another instruction in TYPE128 is LOOP, which decrements register CX and jumps to the designated address if CX is not yet 0. This is the same as the Z80's DJNZ instruction except that it uses a 16-bit register (CX) and therefore can control the repetition of loops that need to repeat up to 65,536 times.

Farther down in the listing is the PAGE routine that waits for a key to be typed and resets the line count. Another feature of the 8086 shows up here—LINECT can be directly loaded with a constant value, without using a register.

Data

Following the PAGE routine is the data section of the program, indicated by the DSEG directive. You notice I have put ORG statements here. This is because CP/M-86 initializes the first 100h bytes of the program's data segment to look just like the base page of memory in CP/M-80. This means that the two default File Control Blocks (FCBs) are placed at 5Ch and 6Ch, the command line trailer is placed at 80h, and the Disk Memory Access (DMA) address is initialized to address 80h. If I were not going to use any of the data provided by CP/M, I could just start the program's data at location 0; since I do use the data from CP/M, I start the program's data at 100h, just above the base page.

All data looks nearly the same in any assembly language, so there isn't much more about the data section that is worthy of mention. One thing you should notice is the RB directive at LINECT. RB means 'Reserve Bytes' and is identical to the Z80 DS (Define Storage).

Also notice that none of the labels (actually they are 'variables' according to the ASM86 manual) in the data section is followed by ':'. Doing so generates an error message from ASM86. ASM86 is very strict about where colons are placed. Labels MUST be followed by a colon; variables and constants MUST NOT be followed by a colon. Contrast this to the M80 assembler where constants (equates) MUST NOT be followed with colon, and variables and labels MUST. Now try working with both systems at the same time and see how long it takes before confusion subsides.

The <ESC>B0 and <ESC>C0, by the way, turn video highlighting on and off on my souped-up Big Board. Unless you are using a new Kaypro (or a Big Board with my super-duper ROM) as a terminal, you will want to change or remove these.

Object Code Size

I pointed out earlier that the more versatile moving and arithmetic routines let you do the same operation with fewer instructions. An interesting sidelight to this is that the single 8086 instruction:

```
MOV  LINECT,1    ;C6 06 00 01 01
```

takes 5 bytes, exactly the same number of bytes it would take to do the operation on the Z80 with:

```
LD  A,1          ;3E 01
LD  (LINECT),A   ;32 00 01
```

It appears that these more powerful instructions don't directly lead to a smaller object file. But let's assume that the value in the A register must be preserved. This does not change the size of the code for the 8086, since it doesn't use any registers. It does change the size for the Z80, however, since now we must do the following:

```
PUSH    AF              ;F5
LD      A,1             ;3E 01
LD      (LINECT),A      ;32 00 01
POP     AF              ;F1
```

Now it's 7 bytes! (and it requires four more memory accesses). Of course, most Z80 programmers avoid keeping permanent values in the A register, so the extra code wouldn't be needed very often. Even so, this forces us to keep one less value in a register, leading to more memory accesses and slower execution (along with longer instructions somewhere down the line).

8080 Translations

All this talk of object code length and program efficiency brings to mind another subject: converting 8080 assembly language programs to 8086 assembly language with XLT86 (a source code translator available from Digital Research). Let's see what happens to a typical sequence of 8080 code when translated to 8086. The following two statements check to see if register DE contains a 0 (a very common need):

```
LD    A,D    ;7A
OR    E      ;B3
```

When XLT86 translates this to 8086 code we get:

```
MOV   AL,DH   ;8A C6
OR    AL,DL   ;0A C2
```

The object code is 100 percent larger! Now you know why all the public domain programs for the 8086 are so big—most of them are just 8080 programs worked over with XLT86.

The extra code size can be avoided, however. A good way to check for DX=0 in 8086 assembly language is:

```
OR      DX,DX   ;0BD2
```

(Continued next page)

(continued from page 33)

Back to two bytes again. See what a little human intervention can do?

Now let's look at execution times of the real 8086 code versus the 'rehashed 8080' code. According to my 8086 micro-reference chart, OR DX,DX executes in three clock cycles. Its double instruction counterpart takes five cycles. Not only is the code twice as large, but it also takes nearly twice as long to execute! I must admit, however, that it is still faster than a Z80 running at the same clock speed (it takes the Z80 eight cycles).

These may all seem like nit-picky, minor little points, and since the program in question is just XDIR or something, program size and speed are rather inconsequential. But if you are considering moving over a real, professional package from 8080 by using XLT86, don't consider it. Forget it. Translate it by hand; you will be much happier in the end.

I Know, I Know

I said last time I would talk about the Signetics 2681 UART chip. I still will, but not this month. Next time. It's just that I got a request for this topic, and it's so rare that I get a request to do anything (other than be quiet and go away) that I just had to do it.

I should also have a firsthand report on the Slicer PC board by next time, along with notes on a C compiler or two. I spent an evening toying with Small C and found that it is a great learning tool, but lacks some important features (struct, for example) that I can't do without (I've been spoiled by UNIX C). I have heard that The Code Works is coming out with Q/C for MS-DOS. If it is as inexpensive as the Z80 version it might be worthwhile.

Speaking of requests—I have finally received some responses to my column (a whole two letters and a comment on a renewal slip). Just to encourage more of the same, I am going to give you my address at school. I don't give this to just anyone, so you should feel privileged. (You Kaypro people turn the page now! This isn't for you.) I must warn you that I will only be at this address until June 15, when I graduate (cross your fingers everyone). After that, I may be anywhere.

Please send any suggestions for future columns, public domain software, care packages (no peanut butter cookies, thank you), products to review (no data bases or Apple products), donations, hot software tips, bribes, threats, citations, phone numbers (brunettes only), death threats, and fan letters to:

Laine Stump
1819 S. Rouse
Bozeman, MT 59715

And, yes, we do have a lot of clowns up here.

■ ■ ■

The Kaypro Column: An Adventure In Trouble Shooting

By Dr. Ralph E. Chatham

8527 Betterton Ct.
Vienna VA 22180

It was on a Monday that my Kaypro II died. This was the same trusty machine that had worked perfectly for over a year and a half. I had hauled it across country and had wheeled it on a little cart through the halls of the Pentagon. This was my good friend, who corrected my spelling without making snide remarks and with whom I had struggled through the first release of Perfect Software. Imagine my dismay when my writing companion first began to slow down and then refused even to read a disk.

Up to that time, my experience with computer hardware had been limited to learning what an op-amp was, and to knowing which end of a soldering iron to hold. But with my little ohmmeter and a lot of advice, I learned more about my Kaypro in six days than I had in the previous year and a half.

About a year ago I subscribed to Micro Cornucopia. I wasn't sure I would understand enough of it to make it worthwhile, but I had hopes for the 5MHz mod and an interest in the schematic and theory of operation package. As each Micro C came, I found that I understood about half of its contents, was thoroughly baffled by the rest, but enjoyed everything.

My Kaypro had had some trouble reading a few disks a week or so earlier, but I had attributed it to the 5MHz speed up.

Monday

On that dreadful day, I hauled the Kaypro into my office and was trying a new program to change the tab spacing default. I had written half a copy of PW.SWP to my B: drive by mistake. Every time I tried to run Perfect Writer I got nasty notes about a bad swap file until, by accident, I started from disk A:. Then the disk accesses began to get very slow. Swapping crept along, and the return to menu took what felt like forever. Things seemed to get even slower as the system tried to access the disks to format and then print my file.

"Well," I thought, "there must be something else wrong with my newly generated copy of Perfect Writer." So I removed the disk, put the old copy in, and pushed the reset button. Drive A: just spun. I tried every other disk I had in the office, and all drive A: would do was light up and turn, while the screen kept saying, "Please put your diskette in drive A."

There are few things more frustrating than a computer that can't even read its own system. I called my computer store. A recording said they were all busy, but they would call back as soon as they could.

On My Own

In the face of such prompt dealer support my first step was to open up the case and look for any lose sockets or cables—anything that might have wiggled during transit that shouldn't have. Nothing.

Next, I suspected the 5MHz mod I had just installed. Maybe if I took it completely out that would fix it. The Radio Shack I called had never heard of those chips whose pins I had bent and soldered jumpers to, but they suggested an electronics supply store which carried things like that.

I remembered a note in Micro C about a transistor that might be needed to sharpen up the rise time of the timing pulses. A call to Dana at Micro C was next on the agenda. He assured me that if the 5MHz mod had worked once, then the transistor was not the problem. He suggested I look for loose things.

He also suggested I swap drives, and explained how: remove them, change the shunts, and be sure that the blue terminator is on the end of the line. "If that fixes it," he said, "the original A: drive is bad, and you might as well buy another one since it would cost more to fix than to replace."

That night I pulled out all my back copies of Micro C and re-read the Kaypro columns. I found a discussion of removing drives and swapping them that was a bit more detailed than my memory of Dana's directions. I tried it. The 'new' drive A: just spun. I put the drives back and systematically undid every trace of the 5MHz mod, testing at each stage. Drive A: hummed, and the monitor blinked its cursor. I stared at the Micro C schematic and the theory of operation that had come with it. I went to bed.

Tuesday

Once again at the office (in between answering questions from Congress, and defending my next year's budget) I called Dana again. I mentioned the previous problems with the printer. He had another idea: some printers ground pin 18 of the parallel port, but some put 5 volts across it.

The older Kaypros (mine was one of the very first with horizontal drives and the Perfect Software package) had a 10 Ohm resistor from pin 18 to ground. This resistor, R21, had burned out on him once and pulled down the rest of the power supply, causing funny things to happen to the drives, one of the systems most sensitive to bad power regulation. "Try cutting out R21," he suggested. The newer Kaypros deleted it altogether. This suggestion had all the elements of the right solution.

I drove home joyously, and just before a family outing, I snipped the wire to R21.

"Please insert your disk into drive A:." Flash, flash, flash.

I was poor company that evening.

Wednesday

Back to Congressional statements and telephone calls. I called Dana again and was lamenting my lack of an oscilloscope when he began telling me some of the virtues of a multimeter. The 4MHz signal, for example, will show on a dc voltmeter as somewhere between 1 and 5 volts. One can trace it from its source to the drive and insure that it gets that input. He explained that there was good reason to trace the 4MHz line, since it is important to the proper timing of the drives.

Then I called Kaypro, but found the support number is just for software. Only dealers can get technical support by phone. After listening to me spout chip numbers I had discovered only that morning, the person I spoke to was kind enough to transfer my call to the hardware department. They were busy, but promised to call me back. (Where had I heard that before?)

That evening I called back Kaypro technical and talked to a man who had never seen a Kaypro as old as mine. Pretending to be a dealer, but without actually lying to him, I mentioned all the

(Continued on Page 37)

Create graphics masterpieces with...

Rembrandt

Complete Business Graphics Toolkit™

NEED GRAPHICS? You don't need a new computer. You need REMBRANDT. The software package that unleashes all the graphics power built into your Kaypro Computer.

Until now, accessing Kaypro graphics required advanced programming efforts. Now the REMBRANDT Business Graphics Toolkit gives you three easy-to-use tools that allow even the most inexperienced user to quickly master Kaypro graphics.

KGRAPH™ enables quick and easy creation of business graphics including horizontal and vertical bar charts, pie charts and xy plots (scatter-graphs) — KGRAPH uses hand entered data or reads numerical data from just about any source including dBase II, spreadsheet, Mbasic and Wordstar files.

KBOARD™ is the full-screen graphics editor for your Kaypro computer. Create graphic screens, save and recall them to and from disk. Layout forms, design logos, draw pictures. It's easy and fun to use!

KBRIEF™ produces electronic on-screen "slide shows" with absolutely no programming required! KGRAPH and KBOARD files are easily sequenced using nine special effects!

REMBRANDT is complete with printer routines so graphics can be reproduced on virtually every dot-matrix or daisy-wheel printer.

REMBRANDT, the Complete Business Graphics Toolkit costs just $79.95.
A demonstration disk is just $5.00 applicable to the purchase price.
See your Kaypro dealer for a demonstration.
For Kaypro 2-84, 2X, 4-84, 4E, 4X, 10, 12X and Robie.

Make your KAYPRO computer IBM-PC compatible for $29.95!

READ, WRITE and FORMAT more than 25 different types of disks (including IBM PC-DOS/MS-DOS) with

Media Master™

Now available for the IBM-PC 75 formats just $39.95

Are you tired of trying to find your favorite software package in Kaypro format? Would you like to use your Kaypro generated Wordstar files, dBase II data and spreadsheet files on the IBM-PC at work (and vice-versa)? Do you want to trade public domain software with a friend who owns an Osborne?

MEDIA MASTER gives your Kaypro instant access to program and data files in over 25 disk formats including:

Osborne SD & DD	LNW-80	Cromemco w/Int'l Term	Xerox 820 I SD
IBM PC-DOS 1.0 & up	TRS-80 with Omikron CP/M	Cromemco CDOS SSDD	Xerox 820 II DD
IBM PC-DOS 2.0 & up	TRS-80 III w/Memory Merchant	Cromemco CDOS SSSD	Zenith Z90
IBM CP/M-86	TRS-80 IV with CP/M+	TI Professional CP/M-86	DEC VT180
Morrow MD2	Heath Z100	Actrix	NEC PC-8001A
Systel II	Heath w/Magnolia CP/M	Lobo Max-80	Kaypro II

Now available for Kaypro 2 (and II): All other models soon!

ORDERING INFORMATION: Include $3 per order for postage/handling. Overseas airmail add $10.
California residents add 6% tax (LA County, add 6.5%).
To place COD or credit card orders,

VISA MasterCard

Call TOLL FREE 24 hours: 800-824-7888 (Ask for Operator 409)
Alaska, Hawaii: 800-824-7919 (Ask for Operator 409)
Technical questions: call (818) 716-1655

For more information, a free brochure (including sample printouts and reviews), or to order, contact:

formerly
DG/SYSTEMS

TECHNOLOGIES, INC.

22458 Ventura Blvd., Suite E
Woodland Hills, CA 91364

All programs also available for OSBORNE and DEC RAINBOW computers. Dealer inquiries invited.

(continued from page 35)

things I had done, concluding that I could "Easter-egg" parts, but would rather have some rationale to my approach.

"Why don't you swap boards with a good one?" he asked. My identity as a dealer became rather shaky when I had to admit I had no spare boards. ("I'm a rather small outfit.") He suggested I try a blank formatted disk. I already had. Then he suggested it could be the disk controller chip. I thanked him and wandered down to the basement to try tracing 4MHz with a dc ammeter as Dana had suggested.

In the process of tracing, I realized there were buffers in the circuits. Many of them inverted signals from one source or another before they went to the disks. I could easily measure the input and output to see if the inverters did the right thing.

I made some measurable progress that night as I reread the theory of operation and studied the schematic. I went to bed a bit more hopeful than the night before.

Thursday

I now had a list of about eight chips that were connected to the disk drives and might be doing bad things. My problems had to be in one of them. I tried to find which of these chips were available. Once again I heard the dreadful words: "Are you a dealer?" or "What company are you with?" I learned quickly that nobody who had the parts I needed sold to people, only to firms. I finally told them I worked for DARPA. "How do you spell that?" they asked. I was obliged, but failed to mention that it means Defense Advanced Research Projects Agency, and I was a program manager, not a computer technician.

Schwebbers had a minimum. They referred me to Capital Radio, which suggested Pioneer of Gaithersburg, to Tinker, to Evers, to Chance. Even so, nobody had ever heard of a 1793 chip.

Back To The Dealer

I began to speculate that the only place that had the parts I needed was a dealer. So I called my dealer again and got the same recorded message that they would get back to me as soon as possible. I told the recording machine that I did not consider three days as soon as possible in anybody's time scale.

I tried another dealer. He would have helped me, but he didn't have another board himself to swap with me. In any case, he didn't do troubleshooting by chip, but by replacing boards and sending the defective one back to Kaypro to repair.

The Xerox 820 is similar to the Kaypro, according to the Micro C articles; I figured maybe they could find a disk controller for me. After a call to a Xerox dealer who referred me to a central repair facility, I found that even they send their whole boards away for repairs. "Nobody repairs at the chip level," they said. "We will be able to do Kaypros in a few months, however, and it will cost $75 plus parts."

A final call to Micro C: Dana tells me the Floppy Disk Controller occasionally has a prefix FDC, and they were made mostly by Western Digital. Byte advertisers sell them, he says. The data separator also sometimes goes by the name of FDC. I didn't have a copy of Byte, but armed with the FDC designator, I made another round of calls to the local suppliers. I could get an SMC 1793-02 in "plastic, $44; ceramic, $76.90; or surdip (?), $54.30 each." (Is 'surdip' anything like sheep dip?) But as luck would have it, they were out of stock and had a backlog of 26 weeks for delivery. Moreover, they had a minimum, and was I a dealer, anyway? . . .

Kaypro Calls Back

Back to answering Congress (with pencil and paper, and no Perfect Writer). Then came a call from Kaypro technical support; my message had not been lost. It just took some time to get to it. "Sounds like the problem is on the board," she said. The metal angle support bracket next to the video board had occasionally picked up stray currents from the field of the CRT and could mess up the drives. Kaypro had been using plastic brackets there for some time now.

She suggested I first remove the metal bracket. After that, I should try to trace the baud rate chip and related ones, then check for shortings and opens in the cables, and finally try another data cable and check for more shorts.

I felt pretty good when I finally went to bed that night. I hadn't fixed anything, but I had eliminated all but two chips from the possible bad ones, since every line driver and hex inverter worked fine. Five volts or ground applied to the input had the correct effect on the output.

(Beware, however, if you check the unused portions of some chips. Some inputs are hard grounded, and trying to put a wire with 5 Volts onto that pin will get you unpleasant sparks and possibly damage your power supply. I was lucky and only got the sparks. Another note for those who are as ignorant as I: two of the pins, usually on a corner or two, are used for power and ground. They are not mentioned on the schematic.)

Friday

On the bus that morning I reviewed what I had done. My problem either had to be in the floppy disk controller (FDC 1793) or the data separator (FDC 9216). Everything else either clearly did what it was supposed to do, or when replaced did not improve anything.

I called a Micro C advertiser (Microprocessors Unlimited), and although they didn't have the parts, they looked up the distributor in my area. It was someone I had already called, but when I told them that they were supposed to have it, they checked again. Sure enough, the 9216 was in stock for $11.50, and a 1793 was in Florida for $33.75, and because the order totaled more than $25, I didn't even have to be a company! I ordered both.

Later that morning I got a call from my dealer who gave me the phone number of his computer fixer. A few hours later, I was talking to Dave Means, describing my symptoms, and offering my conclusion that they were probably due to the disk controller.

"Oh yes," said Dave. "How about the data separator. In my experience, whenever this happens, nine times out of ten it is the date separator. The 1793 is less likely." If he had told me that on Monday, he would have gotten my $30 for the first hour, plus parts, but by then he and I agreed it would be a waste of my money. I had already found the problem

(Continued on Page 39)

(continued from page 37)

("and, by the way, did I know of a source of data separator chips?").

Friday, A Week

The data separator arrived! I never thought I would be so pleased with the whirr-chunk of my noisy drive A: as it ran through its loading paces and gave me the A>. "My problems are over!" I thought. Then I tried a disk in the B: drive. Whirr-whirr-whirr, but no chunk.

I looked at the back of the drive enclosure and saw that the drive select shunt which was supposed to tell the drive that it was drive B: had been knocked loose. Out came the drives, in went the shunt, and back went the drives. I finally got the raggedy whirr-chunk from both drives.

Metamorphosis

So here I am and things will never be the same. I have no (well hardly any) fear of the inside of the Kaypro. And I can never again just assume it is my fault when something doesn't work. My trusty Kaypro isn't so trusty anymore. When a disk fails, is it a faulty disk, or is my drive subtly out of order due to my messing with it? And I still don't know what caused the data separator to fail in the first place.

(Editor's note: Kaypro got a huge batch of flaky SMC data separators, and for several months, many systems were dying before they could get off the dealers' shelves. I assume that Ralph's system got one of the flaky chips. His next problem will probably be with the Tandon drives, but replacement with good quality half-wide Japanese drives is easy and quite inexpensive.)

I do, however, have a great feeling of achievement over the all-too-animate object. Moreover, I re-learned some rules of trouble-shooting:

1. First, you can do a lot with very simple tools: screwdrivers, a few alligator clips, bits of wire and a cheap meter. There is even some advantage to using such tools; it makes you think about the problem. (Those of you who used to use slide rules but now use calculators may recognize the phenomenon.)

2. Keep trying. There were two fronts to my effort. One was the daytime effort, confined to phone calls to dealers, parts suppliers, non-linear, and of course, Micro C. The night shift was trial and error with screw driver, multimeter, and schematic. Both fronts were necessary to my developing a final solution.

3. Get out the telephone and call everybody you can think of. Some of them may be a big help. (Micro C only takes technical calls between 9 a.m. and noon Pacific time.)

4. Read everything you can get hold of.

5. Suspect things that have changed recently on your machine.

6. Test everything as you do it. Change one thing and then test. Otherwise you don't know what did what.

7. Don't be afraid of trial and error. Probe test points. When something changes, try again.

8. Try to make an exhaustive list of the parts possibly affecting the problem area, and then cross out the ones that definitely or probably don't contribute to the difficulties.

9. Beware of damaging things while working. I was strongly tempted to fiddle with drive alignment even though I knew I had no way of aligning them. Fortunately, the alignment setting screws were hard to move, and that provided enough resistance to keep me from more than just half heartedly twisting at them. I was also lucky that I had not dislodged the drive select shunt on drive A: (as I had on B:). Otherwise, I'd never have known that the data separator had fixed the problem.

10. Finally, when in doubt, don't just sit there—MEASURE something.

■ ■ ■

Do you own a CP/M-in-a-lunchbox computer? If so...

Summon ZEUS!

"Big enough to break the coffee table. . . I love it!"

"Thanks for another unbelievable issue, this one wore out two ribbons." "My MX-80 is still panting. Wow, 131 pages!"

"Including software is an excellent idea."

"Utterly dazzling"

"I hear nothing but rave reviews of the Oracle."

"Easily the best microcomputer newsletter in North America."

"I have never read a more readable or more useful (or printed a bigger) periodical on microcomputers."

"My wife and I are programmers (5 and 17 years respectively) and are not easily impressed. but the Oracle of Zeus has left us slack-jawed!"

"Once again I was overwhelmed at the quantity and quality in the Oracle."

"I learned more from the Oracle than from some college courses I've taken."

Read the Oracle of Zeus! It's the best CP/M oriented newsletter in North America--but don't take our word for it, read what our readers have said.

The Oracle is the newsletter of the Zorba Equipment Users' Society

Published bimonthly on disk and now available in KAYPRO 4 format. Volume 1 (1984) was more than 600 pages crammed with useful information and ready to run programs. In it you'll find reviews that really tell you how to separate the roadrunners from the turkeys, programs or hardware--reviews of up to 20 pages each. And...

FORUM (letters): questions answered and experiences shared. REGULAR COLUMNS: hardware, telecommunications, funnies etc. HARDWARE REVIEWS: modems, printers, RAM disks, hard disks...SOFTWARE REVIEWS: languages, editors, utilities... you name it!

Plus:

interfacing information, tips on how to get the most out of your programs (dBASE, WordStar etc.) and hardware, book reviews, FREE classified ads and subscriber discounts.

Subscribe to the Oracle, it's like a great bulletin board without the phone bills!
Trial issue: $5. Volume 1 (6 issues): $20. 1985 Subscription: $25

ZEUSLIB: Now in Kaypro 4 format and on 8-inch disks

We have compiled 15 volumes (388k each) of the best public domain software you can find anywhere. Many of the files are compressed, so some of our disks have more than half a megabyte on them, yet they only cost $5 each ($12 on two 8" disks). Free catalog on request.

ZEUS, P.O. Box 40165, San Francisco, CA 94140

The MicroSphere 256K RAM Disk

By Sven Erlandsson

3268 Leon Brisebois
Ile Bizard Quebec Canada H9C 1W2

When I discovered that installing a RAM disk could prevent a lot of wear and tear on my disk drives, I decided to invest in Microsphere's 256K version.

The installation was simple. I removed U52 (74LS24) and replaced it with a jumper block (included with the shipment). Then I mounted the power supply plug on a rear vent opening, and connected the grabbers to +5V and 0V(ground). No soldering was required.

The parallel port is normally a one-way port, sending data to the printer, and U52 provides buffers on the signal lines between the Z80 PIO and the printer. The RAM disk requires two-way communication (write and read), and the buffer (U52) has to be removed for direct contact between the PIO and the RAM disk.

Once the short, sturdy parallel ribbon cable and the power supply cable are connected between the Kaypro and the RAM disk, everything is ready for operation. The ordinary printer cable, formerly connecting the Kaypro and the printer, now connects to a second parallel connector on the RAM disk enclosure.

Software Installation

The RAM disk comes with a floppy containing initialization programs, but the CP/M system first has to be changed from a 64K to a 63K system. MOVCPM.COM handles the change, but ZCPR gets lost in the process. 4INSTALL.SUB on Micro C's K22 disk reinstalls ZCPR on the 63K system for the RAM disk.

Immediately after setting up the 63K CP/M system, I tested the RAM disk by initializing it with the DYNA42.COM program.

The RAM disk works exactly like a floppy disk drive, but much faster, and without the grinding sound of the old disk drives. DYNA42 provides a 215K drive, while DYNA42S provides a 191K drive. The latter is the way to go if you plan to use the diskcopy program for copying the RAM disk to a 191K floppy.

With a 'DIR F:' command, one can toggle the RAM disk to be either drive A: or drive E: (conversely, the real A: drive becomes E: or A:). On power-up initialization, the RAM disk is E:. A diskcopy program is included, which will copy the complete disk A: to E: (track by track), but you can also do file by file copying using PIP.

The 32K printer buffer part of the RAM disk works beautifully, and keeps the printer chugging along even during read and write activities.

Problems

Initially I had problems getting some programs to run on the RAM disk. Perfect Writer hung up just after sign-on and the only way out was to reset the system.

Perfect Calc printed a distorted matrix on the screen (starting with row 1, but with column 'c', rather than 'a') and went to a warm boot as soon as I asked for a datafile.

SBASIC generated a lot of error messages when displaying the 'print' file, and the compilations were aborted.

A Little Sherlocking

Trying to figure out the bugs listed above, I did the tests below:

I replaced the MOSTEC Z80 PIO with a ZILOG Z80A PIO (the shop didn't have any Z80B in stock).

I moved 'MUX' to pin 3 of U66 and 'CAS' to pin 4 (according to a recommendation in Micro C #12).

I exchanged U33, U34, U39, and U48 to faster chips (good for speed-ups, according to Micro C #18).

I installed ZCPR to work on the RAM disk setup. (I was lucky to find 'ZCPR4S.HEX' fitting the 63K CP/M system on Micro C's K22 disk. Otherwise I would have had to get hold of a macro-assembler.)

I temporarily installed a separate 5V power supply for the RAM disk.

A Pattern Emerges

I discovered some interesting things during my testing.

First, the problems with Perfect Writer and Perfect Calc repeated themselves time after time. CRC consistently gave the same 'Mismatch' result each time a floppy had been copied over to the RAM disk.

If 'PC.COM' was moved to another location on the RAM disk, it worked perfectly. However, the copy program used with its 'verification' option didn't give any error messages. 2DISKTST.COM didn't find anything wrong with the RAM disk drive.

I noticed that the problems occurred with the program file in 'Memory Bank 1' of the RAM disk. I used the 'COMPARE' program (from MICRO C's K2 disk) to find out where the file differed from the original on the floppy, and I used 'XAMN' (from my original System-Disk) to study the differences. I found that if I corrected the faulty bytes (using XAMN), they would remain correct for a while, but then change back to the same faulty value. That's why the verification program and the drive test didn't find errors.

I See The Light

Byte #20H on track #16, sector#26 contained 28 hex rather than the 2A hex it was supposed to contain. Byte #20H on track #21, sector#22 contained FD hex rather than the FF hex it was supposed to contain. With this information in hand, I called Microsphere.

They were very responsive. Within minutes I learned that the same memory chip contained both bits. I also learned about the memory bank layout, and that I should replace either memory chip #6 or #2 in the second memory bank.

Although Microsphere mailed me a new RAM chip, I couldn't resist a quick trip to the local parts house for a Mostek 4264.

Now everything's working perfectly. The RAM disk is more than fulfilling my expectations, and I had a great time doing the detective work. In the process, I learned to appreciate the Micro C series of well supported Kaypro disks. And now that I have 256K, I'm ready to speed up to 5MHz.

■ ■ ■

SUPERSPEED

FOR YOUR KAYPRO

Electronic RAM disk and Printer Buffer Combination—the ultimate add-on for Kaypro II, 4, 2-84, 4-84 and 2X Computers.

Incredible speed and efficiency are offered using MicroSphere's dual operation Electronic RAM disk and printer buffer. No longer will you need to wait for your Kaypro to slowly finish mundane tasks such as running a printer or waiting for floppy drives to turn when you have better things to do.

RAM disk size	Configuration	price
256K	215K RAM disk 32K printer buffer	~~$425.00~~ $395.00
512K	470K RAM disk 32K printer buffer	~~650.00~~ 545.00
512K	430K RAM disk 64K printer buffer	~~650.00~~ 545.00
1mb	946K RAM disk 64K printer buffer	~~1,200.00~~ 995.00

U.S. add $5.00 shipping
International add $15.00 shipping

When ordering, please include the model of Kaypro to be used. Boards can be upgraded should yours need change.

Typical speed increases you can expect to see using MicroSphere's RAM disk:

	4MHz Kaypro Floppy Disk	RAM Disk
Recalc 14K Perfect Calc	9:31.25	1:17.78
Load LADDER.COM	9.38*	2.12*
Load Printer Buffer 20k file, 11 pages, 2586 words, using PIP to the LST device		24.61*

*Time in seconds

Comes complete with cabinet, cables, software and connectors.

TLC LOGO for Kaypro Computers. Easy and exciting language for all ages.
TLC LOGO is an exceptionally complete logo with vectors, multiple turtles, full floating point decimals and extremely fast program execution.
TWO versions offered:
STANDARD version using only internal graphics of Kaypro 2-84, 4-84, 2X and 10; no hardware additions or modifications are required; 16,000 pixel resolution.
DELUXE version for all Kaypro Computers that have the MicroSphere Color Graphics board. Features 16 colors, 32 sprites, 49,000 pixel resolution and utilities such as screen dump to printer or disk.

Time comparison of 3 common Logo programs currently offered:

	DR LOGO 320k IBM PC	TLC LOGO 64k Z80	Apple LOGO 64k Apple IIe
Circle test	10 seconds	3 seconds	22 seconds
PolySpiral1	17	4	11
PolySpiral2	out of stack	7	out of stack
Square Test	27	10	41
Four Bugs (req. 4 turtles)	78	6	N/A

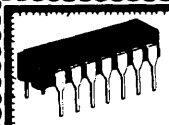
Times provided by The Lisp Company . . . (note: out of stack indicates inadequate implementation of "tail recursion") DR LOGO is copyright Digital Research Company, Apple Logo is copyright Apple Computer Company, and TLC Logo is copyright the Lisp Company.
Standard Version of TLC Logo..... **99.95**
Deluxe Version of TLC Logo..... **129.95**
Special: Color Board and Deluxe TLC Logo..... ~~299.95~~ **199.95**

A NEW DIMENSION FOR KAYPRO COMPUTERS: Color Graphics Board. Features 16 colors, 32 sprites, 256x192 bit mapped graphics. 16K of RAM on the color board itself allows creation of graphics without losing internal memory of the Kaypro.
Software includes 3 editors, drivers and routines to access graphic system. Utilities include screen dump to disk and printer. Dual screen operation features internal Kaypro screen for text and commands, external graphics screen for results. A TV set can be used with addition of RF Modulator.
Color board/Kaypro II, 4, 2-84, 2X, 4-84, 10 & Robie, ~~245.00~~ **145.00**

INSTANT GRAPHER 2.0 (For use with Color Graphics Board) Creates bar charts, stacked bar charts, hi/low, line graphs from keyboard, Perfect Calc, CalcStar or text files. Single and Double size dumpscreen to printer, each color prints a different pattern on a standard dot matrix printer..... ~~$50.00~~ **$40.00**

MicroSphere, Inc.

P.O. Box 1221
Bend, Oregon 97709
503-388-1194
9-5 Pacific Time
Dealer inquiries invited.



64K Memory Chips!
150 nanosecond
Only $1.99 each!

4116 200ns
ONLY $.55

ADVENTURE GAMES BOOK

Written by Kim Schuette over 300 pages of maps, listings and general hints on almost every computer game on the market. Like **Zork I, II, & III, Adventure, Ultima , Secret Agent, Dark Crystal,** etc. - Over 75 in all.
* * only **$16.95!** * *

Z80A Support

Z80A DMA	**$3.95**
Z80A CTC	**$2.50**
Z80A PIO	**3.50**

Qume Trak 242 - 8" half-ht. double-sided double-density
$349.00 ($659.00 List)

TERMINAL CITY

Televideo 921	$495.00
Televideo 925E	$669.00
Televideo 950	$935.00
Televideo 970	$1129.00
Wyse 50	$545.00
Wyse 75	$739.00
Qume QVT 108	$695.00
ADDS Viewpoint - your're choice - A1, A2, or 3A+	$569.00

HARD DISK SPECIALS

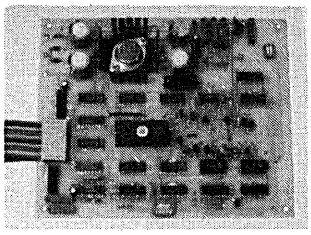
Miniscribe 5¼" 10 Megabyte Half Height. (As used in the Eagle PC and Televideo PC)
1 Year Parts & Labor!
12.8 Megabytes unformatted with an 18 msec. track-to-track access time. This is the 10 meg used by IBM for the PC-XT version. Standard ST-506 interface. **IN STOCK** ~~$699.00~~
Sale Price $539.00

ORANGE COUNTY COMPUTER

(714) 895-5033
5122 Bolsa Avenue, #108
Huntington Beach, CA 92649

CP/M EPROM PROGRAMMING SYSTEM

2708
2758
2716
2732
2764



2516
2732A
27128
27256
27CXX

- STAND ALONE BOARD - ELECTRONIC SWITCHING OF EPROM TYPES
- USES 24 VOLT XFMR FOR POWER - ALL SUPPLIES/TIMING ON BOARD
- DESIGNED WITH EASY TO GET PARTS - COMPREHENSIVE 30 PG MANUAL
- WORKS WITH ANY CP/M SYSTEM - SOFTWARE INSTALL UTILITY

* * INTERFACE TWO WAYS * *

1 - ONE 8 BIT INPUT PORT AND ONE 8 BIT OUTPUT PORT - 16 WIRES AND A GROUND - NO HANDSHAKE LINES REQUIRED.
2 - CENTRONICS PRINTER PORT - 8 OUTPUT DATA BITS AND ONE INPUT DATA BIT (BUSY LINE) - 9 WIRES AND A GROUND.

* * CONTROL PROGRAM COMMANDS * *

- PROGRAM EPROM(S) FROM DISK
- SAVE EPROM(S) TO DISK
- READ DISK FILE INTO RAM
- PROGRAM EPROM(S) FROM RAM
- READ EPROM(S) INTO RAM
- COMPARE EPROM WITH RAM
- VERIFY EPROM IS ERASED
- COPY EPROM
- DISPLAY/MODIFY RAM - (MONITOR MODE) WITH 11 SUB COMMANDS FILL-DUMP-XFER-EXAMINE-MODIFY-BIAS-PROGRAM-VERIFY, ETC.)

BARE P.C. BOARD WITH COMPLETE DOCUMENTATION AND SOFTWARE ON 8" SINGLE DENSITY DISKETTE $69
(ABOVE WITH COMPLETE PARTS KIT - $169)(A&T - $189)
SOFTWARE AVAILABLE FOR OSBORNE, KAYPRO AND OTHER 5 1/4 FMTS

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL
ANDRATECH
P.O. BOX 222
MILFORD, OHIO 45150
(513) 752-7218

CALL OR WRITE FOR MORE INFORMATION -- ADD $3.00 FOR SHIPPING
OHIO RES. ADD 5.5% TAX -- VISA/M.C. ACCEPTED -- $3.00 FOR COD

Do It Yourself Smart Video Controller

By Roger C. Linger

Linger Enterprises, Inc.
P.O. Box 5783
Newport News VA 23605
804-244-3874

Editor's note: Many of the newer, smaller, computer boards don't contain terminal hardware. These boards are very compact and portable, but if you're stuck hauling around a monstrous terminal so you can see what you've done, then you haven't gained much.

You can build the following smart terminal and package it just about any way you wish, even in the same cabinet as the computer. Then, by golly, you'll have all the benefits of a Big Board. Plus a few.

Some terminals work well with some computers and some software, while other terminals have problems. Your computer's designer (or your text editor's author) may well have used this terminal with his product. If you are at all concerned about compatibility you might do a little checking before doing this project.

As computer hardware and software technology increases, more and more capability is required of video terminals. Terminals need to be flexible in operation, upgradeable, easy to configure (this does not mean dip-switches on the circuit board), able to process all data without the need of handshaking, and capable of enhanced visual and communications functions.

Here's a 16-chip circuit that can be implemented into an existing "bare bones" terminal or used separately with a keyboard and monitor to give the video and communications capabilities supported by many popular software packages. It will directly interface to the Xerox 820, Big Board, Little Board, and Slicer, or any other computer offering console operation through an RS-232C compatible port.

Only 700mA at 5V and about 20mA of + and -12V are required to run this circuit.

The 65/9028 VT

The 65/9028 VT (Video Terminal) functions as the communications and video controller circuitry for a stand-alone video terminal. There are three terminal emulations within its firmware (monitor ROM). These are ANSI (American National Standards Institute) X3.64-1979, Heath H19, and Lear Siegler ADM-3.

ANSI X3.64 specifies reliable control sequences for almost any function of video terminals. Because of this and its growing acceptance by applications programmers, ANSI was chosen as the default mode of operation for the 65/9028 VT.

While the ADM-3 emulation basically performs as a "bare bones" terminal, many attributes and functions are available while under control of the ANSI or H19 emulation. These include: access of a non-scrolling status line, programming of soft (programmable) keys, control of visual attributes and graphics, screen editing and erasing functions, full cursor control, etc.

Handshaking protocol, XON/XOFF, is provided but typically not required since a Video Terminal Logic Controller frees the on-board processor of the time-consuming video tasks, thus allowing it more time to service incoming data.

Set-ups

An on-screen set-up method is used to configure this terminal. When the set-up function is selected, the 24th (bottom) video row becomes non-scrolling and displays a bit pattern of 1s and 0s. You select attributes and operating parameters setting these bits.

For example, the logical state of the first (left-most) bit determines whether the terminal is to perform smooth or jump scrolling ('1' selects smooth scroll, '0' is jump scroll). All selections are stored in battery backed-up memory so they remain during power-down. Figure 1 describes the set-up selections.

7 Vrs. 8-bit Keyboards

Figure 2 lists ten functions that can be performed only from the keyboard, thus preventing inadvertent alteration of system status by on-line equipment.

Either a 7-bit or 8-bit code can be used to call local functions. If your keyboard generates 7-bit characters, a 'Null' (ctrl-@) entry alerts the on-board processor that a local function request is in progress. This special 'lead-in' character is required to keep the following character from being sent to the computer. If you

(Continued on Page 47)

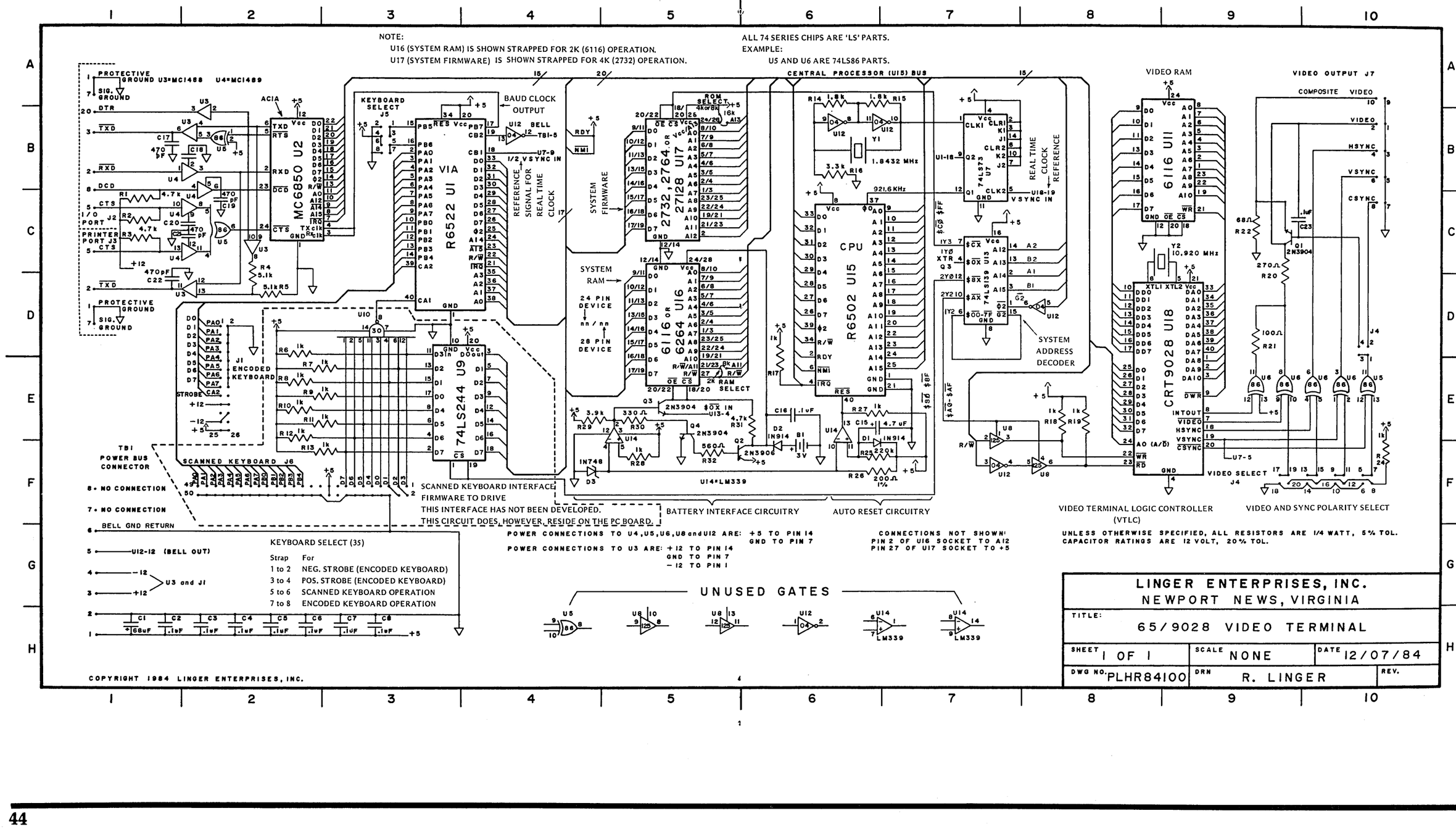
Linger Figure 1 - Set-Up Selections

```
Jump or Smooth scroll
System Status display (on or off)
XON/XOFF protocol (on or off)
Underline or Block cursor
Solid or Blinking cursor
15 I/O Port baud rates (50-9600)
15 Printer Port baud rates (50-9600)
ANSI, H19 or ADM-3 term. emulation
Line Feed on receipt of <cr> (on/off)
<cr> on receipt of Line Feed (on/off)
Inverse or Normal Video Screen
I/O Port data formatting
Printer Port data formatting
Full or Half duplex
Online or Local
60 Hz or 50 Hz Vertical Sync
```

Linger Figure 2 - Local Keyboard Functions

7 bit keybd	8 bit keybd	
NULL 1	B1	On-Screen Set-Up
NULL 2	B2	Display System Status on non-scrolling row.
NULL 3	B3	Transmit through Printer Port screen contents from top to cursor.
NULL 4	B4	Transmit through I/O Port screen contents from top to cursor.
NULL 5	B5	Display control characters.
NULL 6	B6	Display in hexidecimal.
NULL 7	B7	Store first 12 rows of screen for later use.
NULL 8	B8	Recall previously stored screen.
NULL 9	B9	Local/Online operation.
NULL 0	B0	Configure Programmable Keys.*

* Each programmable key can contain a string of up to 32 characters which can be programmed from the keyboard or host computer. Ten keys (hexidecimal C0-C9) can be used.


NOTE:
U16 (SYSTEM RAM) IS SHOWN STRAPPED FOR 2K (6116) OPERATION.
U17 (SYSTEM FIRMWARE) IS SHOWN STRAPPED FOR 4K (2732) OPERATION.
ALL 74 SERIES CHIPS ARE 'LS' PARTS.
EXAMPLE:
U5 AND U6 ARE 74LS86 PARTS.
CENTRAL PROCESSOR (U15) BUS
VIDEO RAM
VIDEO OUTPUT J7
COMPOSITE VIDEO
PROTECTIVE GROUND U3=MC1488 U4=MC1489
SIG. GROUND
ACIA
MC6850 U2
R6522 U1
VIA
KEYBOARD SELECT J5
BAUD CLOCK OUTPUT
BELL
I/O PORT J2
PRINTER PORT J3
J1 ENCODED KEYBOARD
SCANNED KEYBOARD J6
SYSTEM FIRMWARE
2732,2764 OR 27128 U17
ROM SELECT 4K OR 8K OR 16K
SYSTEM RAM
24 PIN DEVICE
28 PIN DEVICE
6116 OR 6264 U16
2K RAM SELECT
74LS244 U9
R6502 U15
CPU
921.6KHz
1.8432 MHz
REAL TIME CLOCK REFERENCE
74LS373 U7
74LS139 U13
SYSTEM ADDRESS DECODER
6116 U11
74LS28 U18
CRT9028 U19
VIDEO SELECT J4
TB1 POWER BUS CONNECTOR
8 - NO CONNECTION
7 - NO CONNECTION
6 - BELL GND RETURN
5 - U12-12 (BELL OUT)
4 - -12
3 - +12
U3 and J1
KEYBOARD SELECT (35)
Strap For
1 to 2 NEG. STROBE (ENCODED KEYBOARD)
3 to 4 POS. STROBE (ENCODED KEYBOARD)
5 to 6 SCANNED KEYBOARD OPERATION
7 to 8 ENCODED KEYBOARD OPERATION
SCANNED KEYBOARD INTERFACE FIRMWARE TO DRIVE THIS INTERFACE HAS NOT BEEN DEVELOPED. THIS CIRCUIT DOES, HOWEVER, RESIDE ON THE PC BOARD.
BATTERY INTERFACE CIRCUITRY
AUTO RESET CIRCUITRY
VIDEO TERMINAL LOGIC CONTROLLER (VTLC)
VIDEO AND SYNC POLARITY SELECT
POWER CONNECTIONS TO U4,U5,U6,U8 and U12 ARE: +5 TO PIN 14 GND TO PIN 7
POWER CONNECTIONS TO U3 ARE: +12 TO PIN 14 GND TO PIN 7 -12 TO PIN 1
CONNECTIONS NOT SHOWN: PIN 2 OF U16 SOCKET TO A12 PIN 27 OF U17 SOCKET TO +5
UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE 1/4 WATT, 5% TOL. CAPACITOR RATINGS ARE 12 VOLT, 20% TOL.
UNUSED GATES
U14=LM339
LINGER ENTERPRISES, INC.
NEWPORT NEWS, VIRGINIA
TITLE: 65/9028 VIDEO TERMINAL
SHEET 1 OF 1
SCALE NONE
DATE 12/07/84
DWG NO. PLHR84100
DRN R. LINGER
REV.
COPYRIGHT 1984 LINGER ENTERPRISES, INC.

NOW IS THE TIME

KAYPRO 4-84

At last, the real time clock in the 4-84 IS good for something — the complete TIMEPRO system is available as single programs, or as a complete set:

SEETIME.COM continuously displays the time and date (in either standard or military time) on the 25th line of the screen — all the time you use your Kaypro. Does screen dump.

TD.COM time and date stamps your disk files. Automatically detects which files have been added or modified since TD was last run and updates the time and date only for those files. Does automatic backup of updated files.

APPT.COM maintains calendar of appointments and other reminders for up to 20 people. Includes archiving capabilities and several utilities.

★NEW **LOG.COM** designed mainly to aid Kaypro users in complying with the new IRS requirements regulating computer use. Can be used for tracking time spent doing **any** type activity. An easy and convenient method of time keeping!

KAYPRO 2-84, 2X-84

As you probably know, the main computer board of these two systems is identical to the 4-84, except that Kaypro left out the components for the real time clock and modem. We have the forgotten real time clock components, complete with instructions, ready to install.

KAYPRO II, 4 (Pre '84)

Time and speed are yours with our easily installed SPEEDPRO-5RTC 5 megahertz speedup board with real time clock. Shipped complete with the full TIMEPRO series of 12 programs — a whole disk full (192K) of software.

All packages are accompanied by programs for setting the real time clock and by source code in several computer languages to demonstrate the use of the real time clock in your own programs.

KAYPRO 10

Now TIMEPRO can be yours too! Available now with clock kit. TIMEPRO software is fully compatible with Z TIME-1 calendar/clock kit from Kenmore.

THE BOTTOM LINES

SEETIME.COM	$ 24.95
TD.COM	$ 24.95
APPT.COM	$ 24.95
ALL ABOVE, PLUS SEVERAL ADDITIONAL PROGRAMS	$ 64.95
LOG.COM	$ 24.95
TIMEPRO DOCUMENTS ON DISK (credited toward complete TIMEPRO system)	$ 9.95
2/2X/10 REAL TIME CLOCK KIT	$ 49.95
CLOCK KIT & TIMEPRO	$104.95
SPEEDPRO-5RTC & TIMEPRO	$129.95
SPEEDPRO-5+ SPEEDUP BOARD	$ 89.95
SHIPPING ..... Software	$ 2.50
Hardware	$ 4.00

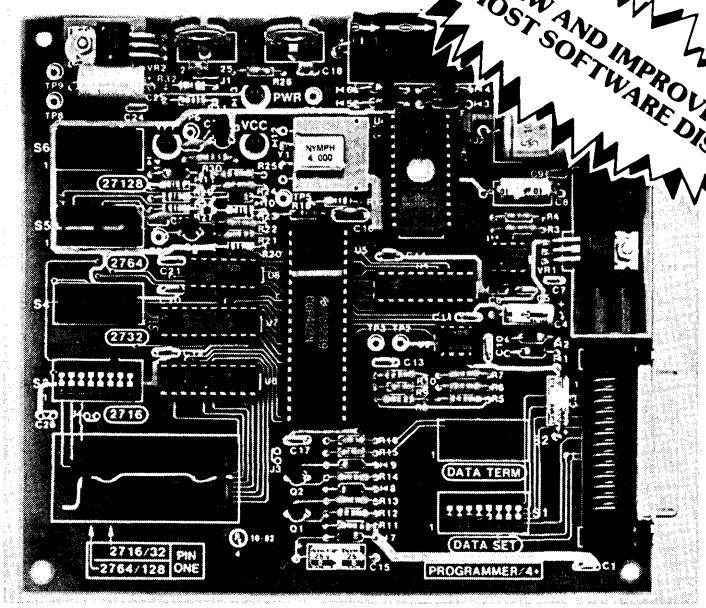
M/C, Visa, COD, Check

DIGITAL *dynamics*

Digital Dynamics (206) 627-0797
P.O. Box 5653 Tacoma, WA 98405

PROGRAMMER/4+

NEW AND IMPROVED HOST SOFTWARE DISK



A LOW COST ALTERNATIVE TO EPROM PROGRAMMING

Reads and programs 2716, 2732, 2764, and 27128 EPROMS.
Reads 2-16K ROMS.
Direct connect to any RS232C terminal or computer.
Plug selectable as either a data set or data terminal.
All voltages made on board, (no power supplies needed).
(User supplies power Xformer, 25.2 to 30 VAC C.T.1 Amp.).
Power electronically switched, (can't damage EPROMS).
Zero insertion force socket for EPROM.
Programs, verifies, and dumps in both ASCII and hex.
Edit buffer (like DDT).
Saves hex and/or image files to and from disk.
Saves or loads all or partial buffer.
Completely menu driven for ease of operation.
Commands of Test, Read, Display, Save, Load, Program and more.
Check sum calculation.
All software on disk including well commented source code.
Detailed owners manual including schematic
All chips socketed.
Not a kit! Completely built and tested.
48 hour dynamic burn-in and test before shipment.
90 day limited warranty on parts and workmanship.
24 hour return policy on repairs.
Delivery from stock.
NOTICE TO PREVIOUS CUSTOMERS: Send us your old disk and a suitable S.A.S.E. for your free copy of our new software, ver. 1.6.
PROGRAMMER 4+ WITH OWNERS MANUAL AND DISK. $199.95

Order from **peripheo**
1659 Scott Blvd., Suite 1
Santa Clara, CA 95050
(408) 244-5214

VISA and MASTERCARD telephone orders welcome.

Please specify Disk format
CP/M 8" IBM format, KAYPRO II, XEROX 820, OSBORNE I, others.

Please specify method of shipment, UPS or Postal Service.
California residents add 6% Sales Tax. Dealer Inquiries invited.

(continued from page 43)

have an 8-bit keyboard that can output characters between B0 through B9 hex, the extra lead-in character is not required.

Even if you have a 7-bit keyboard (as opposed to a 2-bit model) you can add a switch to force the eighth bit high (connect to +5V to the high bit line through a 1K resistor). Then characters 0 through 9 appear to the terminal as hexidecimal B0 through B9.

Connect the switch between the high bit line and ground. Opening the switch causes the entire keyboard set to be shifted up 128 decimal since the most significant bit is being held high.

The Display

The screen display contains 80 characters by 24 rows. When the 24th row is non-scrolling, the top 23 rows are used for text display. All 24 rows are used for text display when the 24th row is set to scroll.

Character Display

This terminal can display 128 characters, plus simple graphics. The 128 characters include: special characters for the 32 control codes, upper case, and lower case (with descenders). There is a 7 by 10 matrix for each character.

Character attributes include normal, reverse, intensified, underlined, and blank. You can also select a normal or reversed screen.

Circuit Description

Data from the RS-232C ports, J2 and J3, is routed through voltage level converters U3 and U4 to the ACIA (Asynchronous Communications Interface Adapter), U2. The ACIA is a smart interface between the processor and the serial world. (Editor's note: ACIA is the 6502 world's name for an SIO.)

The ACIA gets its instructions (baud rate, bits per character, and interrupt info) from the processor.

The baud rate clock comes from pin 17 of the VIA (Versatile Interface Adapter), U1. The VIA also performs clock interrupts, keyboard interfacing, and tone generation.

The DTR (Data Terminal Ready) pin of J2 is set 'high' when the terminal is ready. The computer can check this handshake line to see if it is OK to send data to the terminal.

Meanwhile, the terminal checks the CTS (Clear To Send) line on J2 or J3 when it is ready to send characters. If this handshake line is high, then the terminal knows it is OK to send data to the computer (or printer or whatever else is connected to the port).

CPU

The CPU, or processor, has overall charge of the system. It is supported by U13 (decoder), U16 (RAM), and U17 (ROM).

The scratch pad memory, processor stack, data buffers, set-up variables, and screen and function key information all reside in RAM. The RAM is contained in a 2K by 8 (2048 byte) CMOS device and is upgradeable to 8K by 8. The ROM (Read Only Memory) is a 2732 (4096 byte) device.

The Battery Circuit

The battery circuit preserves the contents of U16 when the terminal is turned off. This circuit detects a power-up or power-down transition and switches in or out the system battery. Zener diode D3 and resistor R26 generate the reference voltage for this circuit.

Pin 3 of voltage comparator U14 is high impedance (open circuit) when the +5V bus exceeds 4.60V (the power is on). When the 5V bus is below 4.60V (power-down time) U14 pin 3 will short to ground.

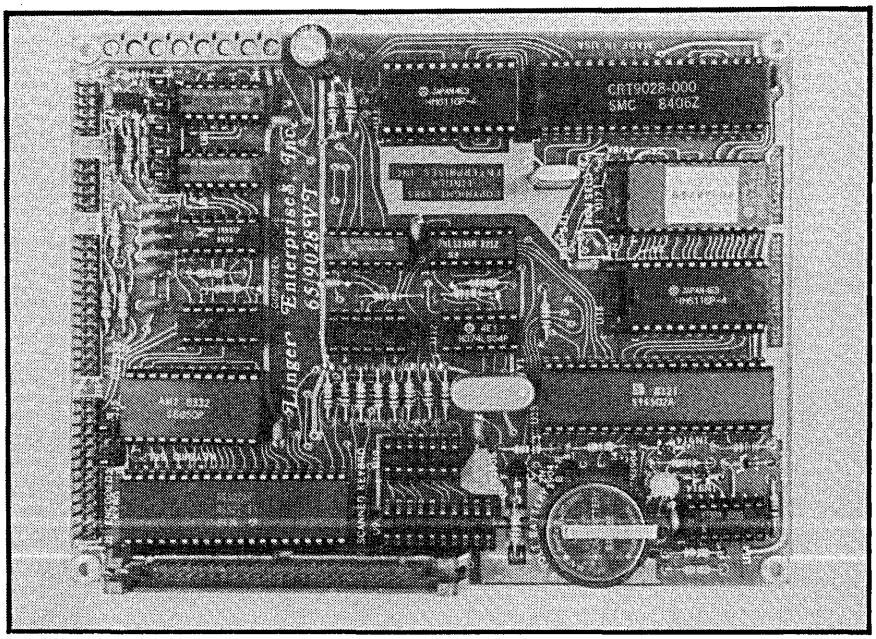
When U14 pin 3 is high impedance, transistors Q2, Q3, and Q4 are turned on by the current through R29 and R30. In this condition, Q2 supplies +5V, and Q3 allows the processor to select U16 for read or write operations.

Two seconds after power-up, capacitor C15 charges enough to cause U14 pin 13 to turn off so that R27 can pull the processor's reset pin (40) high. The diode tied to the comparator side of the capacitor discharges it when power is removed.

The Video Terminal Logic Controller

The VTLC (Video Terminal Logic Controller), U18, is the reason this terminal is so simple. This device provides all video timing, character and graphics generation and display, visual attribute logic, and data transfers between the CPU and Video RAM, U11. The horizontal sync frequency supplied by the VTLC is 15.75KHz. It can generate either 60Hz or 50Hz vertical sync.

(Continued next page)



Top view of terminal circuit board

(continued from page 47)

Data transfers to and from the CPU are handled through the processor data bus, D0 through D7. The VTLC READ and WRITE pins determine the data direction. Resistors R18 and R19 keep these pins biased high (and data bus drivers in a high impedance state) when the processor isn't talking to the VTLC.

You can address the VTLC's internal registers by using its A/D input, pin 24. Select a data register by pulling pin 24 high, pulling the WRITE input low, and then writing the register address on the data bus. The addressed register will be available for the next read or write operation (the A/D pin must be low during the read or write).

All data transfers between the Video RAM and VTLC are performed video blanking, thereby giving a totally flicker-free display.

Attributes

You can select an attribute for each character. When the VTLC writes a byte to Video RAM, the first 7 bits of that byte represent the ASCII character to be displayed. The state of the 8th bit indicates whether or not the character has a special attribute. If it has, the current attribute will be used when the character is displayed.

Video Out

You select the polarity of video and sync signals by strapping the exclusive-OR video buffers, U5 and U6. Strapping an input low gives true output data, while strapping it high gives inverted output data.

Scanned Keyboard

The hardware can handle scanned (undecoded) keyboards. U1, U9, and U10, resistors R6 through R12, and keyboard header J6 provide the scanning interface. However, many of these components are not installed in our board, because our ROM does not support the scanning.

Building The 65/9028 VT

Here are a few tips that will help if you wish to wire-wrap your own version:

Incorporate more bypassing on your wire-wrap board than is listed here. A bypass capacitor of 0.1 uF per I.C. socket is not too much. These capacitors should be soldered directly to the appropriate socket pins on the underside of the board.

You may not want to incorporate all or any of the personality jumpers on your board. If so, tie pin 16 of U1 to ground for encoded keyboard operation. Tie pin 15 of U1 to ground if your keyboard supplies a negative strobe, or tie it high for positive strobe operation.

Hard To Get Parts

Listed below are the hard to get parts and their source. You can probably find a

local distributor for the first two items listed.

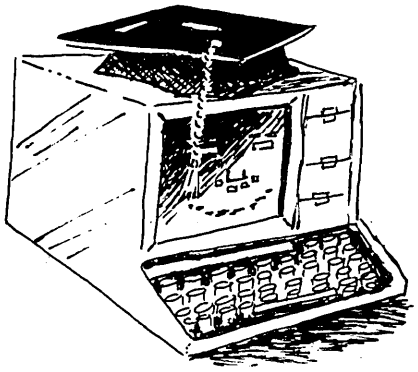
1. CRT9028-000—Standard Microsystems Corp., 35 Marcus Blvd., Hauppauge NY 11788.
2. 10.920 MHz crystal (HC-18/U case, 10pF load)—Seiko Instruments, 2990 W. Lomita Blvd., Torrance CA 90505.
3. 1N748 diode, 200 Ohm 1% resistor, Lithium battery and holder, 8 pole P.C.B. Terminal Strip—Digi-Key Corp., phone 1-800-344-4539.

Packages Available

There are four packages available from the author. The ROM with manual is $44. The PC board (only) is $39. Complete kit will manual, board, parts, and sockets is $219. Assembled and tested board is $259.

We've got some serial 67-key keyboards that provide the ten programmable function keys used by the 65/9028 VT. They can be converted to parallel for $3 worth of parts. Keyboard and schematic for the conversion cost $15.

■ ■ ■



CP/M 86

8" CP/M-86 Disk $15.00 each

DISK 86-1 — Disk Utilities

D.CMD/A86, SD.CMD/A86, XDIR.CMD/A86: Three extended directory programs. Each does it differently, so we included all three.

FILE-EXT.CMD/A86: Disk status program with good display format.

PAGE. CMD/A86: A text paging program. Displays 24 lines at a time.

PRINT.CMD/A86: File printing routine. Puts a header at the top of each page along with page number and file name.

MUCHTEXT.CMD/A86: Counts words and lines in a text file.

ERQ.CMD/A86: Selective file erase program. Displays all selected files and then asks you one at a time for a Y/N.

INUSE.CMD/A86: Prints "In Use" on your terminal and asks for a password. It will not release the console until you enter the password.

FINDBAD.CMD/A867. Finds and collects bad sectors on a disk. If there are no bad sectors, information on the disk is unaltered.

Disk 86-2 — DU and Modem Programs

DU-V75.CMD/A86/DOC: This is the popular disk utility from CP/M 80. It lets you read, write, and modify disk sectors.

MODEM4.CMD/A86: This is a modem program set up for the Slicer. This program includes a built-in help file.

MODEM7SL.CMD/A86/DOC: No modem disk would be complete without this standard. This is modem7 set up for the Slicer. It displays a menu when it is called.

Disk 86-3 — Small C

C86.CMD: This is the original Small C compiler which appeared in Dr Dobbs Journal in 1980. It runs under CPM-86 and generates 8086 source for the ASM86 assembler.

C86. COM: This is the C86 compiler which runs under CPM-80. This 8080 program produces 8086 assembly language.

C86LIB.A86: This is the C86 I/O library.

SMALLC86.DOC: Documentation on Small C.

C?????.C: Source of the C86 compiler.

DISK 86-4 — IBM Mainframe Interchange/ RESOURCE 8086

XBIOS.A86: A new BIOS that supports a real time clock.

RES86.CMD: A disk management program for transfering files between CP/M-86 and IBM 374X mainframe environments.

SDI86.CMD: An 8086 version of the RESOURCE disassembler.

DISK 86-5&6 — FIG Forth

Disks 5 and 6 are a complete two disk set of FIG Forth 83.

F83.CMD: The standard Fig Forth 83.

META86.CMD: The Forth compiler.

More ROMS: Fast monitor ROMs for speed freaks and our famous 'better than Texas' character ROM (V2.3) for screen freaks.

Fast Monitor ROM BB1 ..... $29.95

Deluxe Character ROM BB1 ..... $29.95

BB II DRIVE INTERFACE

For 5¼" and 8" Drives

Andy Bakkers is making this special software package available through Micro C. Complete source, HEX, & documentation files on an 8" SS SD disk. Also outlines on disk the hardware changes needed.

$29.95

Micro Cornucopia

P.O. Box 223
Bend, OR 97709
503-382-8048
9-5 Pacific Time
Monday-Friday

You Can't Afford Not To Have A(rchivist)!

A(rchivist) gives you total control over all of your disk file needs in one convenient program. Just some of the features include:

- Backup files that are larger than your floppy disk can hold (e.g., copy a 1 megabyte file to 250 kilobyte floppies). Try that with PIP!
- True copy of random files (virtual vs. actual).
- Copy between user areas.
- Start-end command lets you copy contiguous files in a directory.
- Ambiguous renaming of files.
- DIR, STAT, ERA, TYPE, LIST, and VERIFY.
- Copy or erase with the following flags off or on:
  - **e** (exists) do not overwrite if file already exists.
  - **r** include system file if they exist.
  - **w** write over a read-only file.
  - **s** sort the files before writing.
  - **v** do or don't verify the copy.
- Commands are available to set the printer tabs, lines per page, and line width for program listings.

"I don't think we need to run any back-up today."

- There is even a batch mode for batch processing of file maintenance, including a pause and reset command for changing disks in the middle of a batch job if necessary.
- Complete user's manual included.

A(rchivist) requires CP/M80, version 2.0 or later using a Z80 CPU (it will not run on an 8080) and an 80x24 cursor-addressable terminal. To order, call or write:

New Lower Price
~~$150.00~~
$49.95

Ecosoft Inc.
6413 N. College Avenue
Indianapolis, IN 46220
(317) 255-6476

VISA MasterCard AMERICAN EXPRESS

NEW! RELOCATABLE Z-80 MACRO ASSEMBLER FROM MITEK

It's a real bargain! Here's why:

- Only $49.95 plus shipping
- 8080 to Z-80 Source Code Converter
- Generates Microsoft compatible REL files or INTEL compatible hex files
- Compatible with Digital Research macro assemblers MAC & RMAC
- Generates Digital Research compatible SYM files
- Full Zilog mnemonics
- INCLUDE and MACLIB files
- Conditional assembly
- Separate data,program, common and absolute program spaces
- Customize the Macro Assembler to your requirements with installation program
- Cross-reference Generation
- Z-80 Linker and Library Manager for Microsoft compatible REL files available as a total package with Macro Assembler for only $95.00 plus shipping
- Manual only is $15

TO ORDER, CALL TOLL FREE: 1-800-367-5134, ext. 804
For information or technical assistance: 1-808-623-6361

Specify desired 5¼" or 8" soft-sectored format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed. Include $5 for postage and handling.

MITEK P.O. Box 2151 Honolulu, HI 96805

Z-80 is a trademark of Zilog, Inc. MAC, RMAC, and ZSID are trademarks of Digital Research, Inc.

Inline Object Code In Turbo Pascal (CRC For BBII)

By Cliff Nunnery

313 Vaughn Street
Fort Walton Beach FL 32458

This may seem trivial to some, but my training as a fighter pilot didn't include CRC checking. I did loads of instrument and proficiency checks, but those didn't help much in getting my BBII to come up after I'd changed a few bytes in the monitor EPROM. The obvious solution was to remove the CRC check from the monitor, but instead, I chose to try my hand at a little inline code in Turbo Pascal.

The instruction at "INITI:" in the monitor routine carries the comment ";LOOP FOREVER IF BAD CRC." Changing code in the monitor EPROM changes the CRC result and, sure enough, it loops forever. The volumes of documentation supplied with the BBII don't mention this problem. (Inside joke.)

The Solution

My CRCGEN routine stops two bytes from the end of the EPROM. Then the contents of the H register are loaded into the next byte, and the contents of the L register are loaded into the last byte. These values will return HL = 0 when the ROM's CRCGEN routine is called during initialization.

First, move an image of the new ROM code to 9000H, then use my modified version of CRCGEN to check this code from 90000H to 9FFDH. Load the contents of the H register into 9FFEH, and the contents of the L register into 9FFFH. This code can then be used by PROG32.COM (User Disk #14) or moved to 0100H and saved to a .COM file for use by PRORAM.COM on the same disk.

Figure 1 shows how I used a combination of Turbo Pascal and object code (with assembly language mnemonics as comments). The program modifies the last two bytes of code in a file, ROM-.COM, and writes it to disk with the same file name. The modified code will then pass the CRC check during BBII initialization. This should save someone some time.

I sent Cal-Tex a 2732 EPROM about a year ago with a request for a de-bugged monitor program. I followed up with a letter two months later. I have yet to get a reply. Besides that, they kept the EPROM. What gives, guys? Even a super computer needs a little support!

■ ■ ■

```
                Pascal Procedure For Checking The BBII

SET CRC

Program SetCRC;  {Program to set CRC in last two bytes of
                  BBII monitor file, ROM.COM
                  File ROM.COM must be on same disk as
                  this program}

Var
  RomArray : Array[1..$1200] of Byte Absolute $9000;
  RomFile  : File;
  InCh     : Char;

Begin
  Writeln;
  Writeln('Program will modify CRC bytes in File: ROM.COM');
  Writeln('Proceed ?     <Y / N>');
  Writeln;
    Repeat
      Read(Kbd, InCh)
    Until
      UpCase(InCh) In ['Y', 'N'];
    If UpCase(InCh) = 'N' then Halt;
  Assign(RomFile,'ROM.COM');
  ($I-) Reset(RomFile) (I+);
    If IOResult <> 0 then
      Begin
        Writeln('File: ROM.COM not found');
        Halt;
      End;
  BlockRead(RomFile, RomArray, 32);
  InLine ($11/$00/$90/  {       LD    DE,9000   ;1ST BYTE OF ROM IMAGE}
          $21/$FE/$0F/  {       LD    HL,0FFEH  ;NO. OF BYTES TO CHK }
          $19/          {       ADD   HL,DE     }
          $E5/          {       PUSH  HL        }
          $21/$FF/$FF/  {CRCGEN:LD    HL,0FFFFH }
          $06/$08/      {CRCG1: LD    B,8}
          $1A/          {       LD    A,(DE)    }
          $4F/          {       LD    C,A       }
          $79/          {CRCG2: LD    A,C       }
          $AC/          {       XOR   H         }
          $07/          {       RLCA            }
          $ED/$6A/      {       ADC   HL,HL     }
          $0F/          {       RRCA            }
          $30/$08/      {       JR    NC,CRCG3  }
          $7C/          {       LD    A,H       }
          $EE/$10/      {       XOR   10H       }
          $67/          {       LD    H,A       }
          $7D/          {       LD    A,L       }
          $EE/$20/      {       XOR   20H       }
          $6F/          {       LD    L,A       }
          $CB/$11/      {CRCG3: RL    C         }
          $10/$EC/      {       DJNZ  CRCG2     }
          $C1/          {       POP   B,C       }
          $C5/          {       PUSH  BC        }
          $13/          {       INC   DE        }
          $78/          {       LD    A,B       }
          $BA/          {       CP    D         }
          $20/$E1/      {       JR    NZ,CRCG1 ;TEST FOR ENDING PG}
          $79/          {       LD    A,C       }
          $BB/          {       CP    E         }
          $20/$DD/      {       JR    NZ,CRCG1 ;TEST FOR ENDING BYTE}
          $7C/          {       LD    A,H       }
          $12/          {       LD    (DE),A ;STORE H REG HERE}
          $13/          {       INC   DE        }
          $7D/          {       LD    A,L ;STORE L IN LAST BYTE}
          $12/          {       LD    (DE),A    }
          $C1);         {       POP   BC        }
  Close(RomFile);
  Rewrite(RomFile);
  BlockWrite(RomFile, RomArray, 32);
  Close(RomFile);
  Writeln;
  Writeln('       CRC bytes modified in file.');
  Writeln('       Code should now pass CRC check');
End.
```

Pascal Procedures

By John P. Jones

6245 Columbia Ave.
St. Louis MO 63139

While talking to Dave Thompson a couple of months ago, I learned that both the FORTH and C columnists would be doing 'typewriter' or keyboard to printer programs for Issue 22. I figured I might as well do one, too, but for Issue 23.

I normally get my copy of Micro C one or two weeks after I submit my column for the following issue, so there was no chance of seeing what Tony and Arne had done before writing this. I didn't want to get into a 'me too' kind of situation anyway, so I started from scratch by analyzing what a typewriter program needs.

Basic Requirements

The simplest form is similar to the 'T' command on the original Xerox 820. In this mode, keyboard input is sent verbatim to the printer. See Figure 1 for an example. This minimum program is useful, but not very.

A full-featured typewriter program is basically a one line text editor and should have at least:

1. Variable line width (right margin)
2. TAB stops (fixed)
3. Append characters to input
4. Insert characters in input
5. Display current input
6. Delete characters from input
7. Move cursor back and forth on input line

These features are desirable but not necessary:

8. Input line wider than display width
9. Large scale cursor movement—beginning/end of word/line
10. Large scale delete—word/end of line
11. Status indicators—ruler line, current cursor pos and text length
12. Printer configuration
13. Continuously variable tab stops
14. Imbedded printer controls for boldface, underline, etc.

The program in Figure 2 implements all of these features except 13, 14, cursor movement by word, and word delete. The code is not confined to 'standard' Pascal.

Much of the efficiency of Turbo Pascal is due to its wide variety of built-in features, many of which were used in the program. The Turbo specific features used are indicated in the comments for those who may need to translate for another compiler. The program requires an X-Y cursor addressable terminal or terminal emulator as in the Big Board or Kaypro.

Figure 1 - Bare Bones Program

```
{$c- Do not interpret control char input }
PROGRAM echo_to_printer;
{ Assumes typewriter connected as CP/M LST: device }

VAR
  ch, ctl_z : char;

BEGIN
  ctl_z := chr(26);
  REPEAT
    read (kbd, ch);
    if ch <> ctl_z then write(lst, ch);
  UNTIL ch = ctl_z;
  writeln(lst);
END.
```

Figure 2 - Full-featured Program

```
{$u-}   { no user interrupts }
{$c-}   { no ctl char interpretation }

PROGRAM typist;
{ Author John P. Jones - January, 1985 }
{ Written for Turbo Pascal, CP/M-80 but should run without
  change on other operating systems }

CONST
  rulerline = 15;  { screen display positions for ruler, }
  textline = 14;   { input text and prompt/status line.  }
  promptline = 23;

  tab = ^i;        { Control character constants, use Turbo }
  bs = ^h;         { specific representation of character   }
  move_rt = ^d;    { constants.  For other compilers will   }
  move_lft = ^s;   { have to declare as VAR's and assign.   }
  del_lft = #127;
  del_crnt = ^g;
  del_eol = ^y;
  move_strt = ^a;
  move_end = ^f;
  form_fd = ^l;
  cr = ^m;
  stop_char = ^z;

  display_len = 79;  { screen display width -1 }

TYPE
  anystr = STRING[255];

VAR
  ch : char;
  linelen : integer;
  tabgap, shift_amt : integer;
  ruler, txtline : anystr;
  help : boolean;
  position : integer;  { position in input text }
  crsr_pos : integer;  { position on display line }


PROCEDURE write_ruler;  { display ruler line below text input line }
BEGIN
  lowvideo;      { Turbo specific }
  gotoxy(1,rulerline);  { screen coordinates start @ 1 for Turbo }
  write(copy(ruler,shift_amt+1,display_len));

 { signal text beyond display if necessary }
 IF (length(txtline) > display_len) AND (position < length(txtline))
   THEN write('+');
  clreol;        { Turbo specific }
  highvideo;     { Turbo specific }
END;

PROCEDURE clr_prompt;    { clear prompt/status line, }
BEGIN                    { leave cursor at beginning of line }
  gotoxy(1,promptline);
  clreol;
END;

PROCEDURE beep;          { Wake up! or I don't understand! }
BEGIN
  write(^g);
END;

PROCEDURE prompt (msg:anystr); { display new prompt, ready for input }
BEGIN
  clr_prompt;
  write(msg);
END;

PROCEDURE config;

VAR i,j : integer;
  ctl : ARRAY[1..10] OF char;
BEGIN

  REPEAT      { get print line length in proper range }
    beep;
    prompt('Characters per print line (1-132)? ');
    read(linelen);
  UNTIL linelen IN [1..132];
```

(Listing continued)

How It Works

The main program first calls INIT, which clears the screen, highlights the prompt/status line, and calls CONFIG. CONFIG prompts for and gets values for print line length and tab spacing, then prints the ruler line. A printer initialization string of up to ten characters is then requested. Special printer modes can be set up at this time.

For instance, the string ESC>+ 'B'+ctrl-B sets up my printer for 12 CPI. The INIT string cannot contain a CR> code, so if you need it, change the program to use another character to terminate input. Finally, CONFIG will display a brief help menu on request.

The major routine in the program is INPUTSTR. When called, input is initialized, and keystrokes are input and processed in a REPEAT loop. INPUTSTR maintains three pointers: POSITION indicates where in the input string characters are to be inserted, CRSR_POS is the corresponding location on the screen display line, and SHIFT_AMT is the magnitude of horizontal scroll needed to keep the cursor on the display.

INPUTSTR contains two routines. SET_CURSOR first saves the current horizontal scroll status, then calculates the new cursor position and scroll amount based on entry position and current length of input. The cursor always points to the character position following POSITION except when the string is full and the pointer is at the end of the string. Since ORD(TRUE) = 1 and ORD (FALSE) = 0, adding the value ORD(POSITION <LINELEN) to the cursor position takes care of this. If there has been a change in scroll status, both the ruler and display lines are refreshed.

If the input is full, INSERT_CH will delete the last character of the input string before inserting the new character.

A CASE statement handles all character processing, printable characters are inserted, valid control characters act as discussed below, and the bell is sounded for all other input.

(Continued on Page 55)

(Listing continued)

```
REPEAT      { get tab spacing, must be < print line length }
  prompt('Tab spacing? ');
  beep;
  read(tabgap);
UNTIL tabgap <= linelen;

ruler := 'L';              { build ruler line based on line length }
FOR i := 2 TO linelen-1 DO { and tab spacing }
  IF i MOD tabgap = 0
    THEN ruler := ruler + '|'
    ELSE ruler := ruler + '-';
ruler := ruler+'R';

{ set up printer for special modes: other pitch, bold face, etc. }
  prompt('Enter printer initialization string (<CR> ends) : ');
  i := 1;
  REPEAT
    read(kbd,ch);
    IF ch <> cr       { <CR> signals end of input }
      THEN BEGIN
             ctl[i] := ch;
             IF ch < ' '    { control char? }
               THEN write('^',chr(ord(ch)+64))  { if ctl char, display in
                                                  CP/M standard ^ch format }
               ELSE write(ch);
             write(' ');    { space between chars on display }
             i := succ(i);
           END;
  UNTIL (ch = cr) OR (i > 10);     { <CR> ends input }

  IF i>1          { if any input, send to printer }
    THEN
      FOR j := 1 TO pred(i) DO
        write(lst,ctl[j]);

  prompt('Do you want the help menu? ');     { want to clutter up screen? }
  read(kbd, ch);
  help := ch IN['Y','y'];

  clr_prompt;
  write_ruler;

  IF help
    THEN      { if help menu requested, print it on screen }
      BEGIN
        lowvideo;
        gotoxy(1,17);    { output lines split so listing not too wide }
        write('Move left = ^S,<BS>  Move right = ^D    ');
        writeln('Delete current = ^G   Print line = <CR>');
        write('Move begin = ^A      Move end = ^F      ');
        writeln('Delete prev. = DEL    Space to | = TAB');
        write('Eject page (if text empty)  = ^L        ');
        writeln('Del to end line = ^Y  Exit = ^Z');
        highvideo;
      END;
END;
```

```
PROCEDURE init;  { initialize status line display and globals }

VAR i : integer;
BEGIN
  clrscr;
  lowvideo;

  gotoxy(1,promptline -1);  { outline prompt/status line with -'s }
  FOR i := 1 TO display_len DO
    write('_');
  gotoxy(1,promptline +1);
  FOR i := 1 TO display_len DO
    write('-');

  shift_amt := 0;
  config;                   { get global vars, setup ruler, etc. }
  highvideo;
END;

PROCEDURE inputstr(VAR text_str     : anystr;
                   VAR tc     : char    );

VAR
  ch : char;
  remainder : integer; { amt of display space remaining }

PROCEDURE set_cursor;  { Where are we now ? }

VAR
  last_shift : integer;

BEGIN
  last_shift := shift_amt;  { keep track of horizontal scroll status }

  IF position <= (display_len)  { within unscrolled display area? }
    THEN BEGIN
{ yes, reset scroll and bump cursor unless at max input }
           shift_amt := 0;
           crsr_pos := position + ord(position<linelen);
         END

    ELSE BEGIN
{ calculate scroll offset and set cursor based on scroll amt }
           shift_amt := position - display_len;
           crsr_pos := position + ord(position<linelen) - shift_amt;
         END;

  IF shift_amt <> last_shift  { change in scroll status? }
    THEN
      BEGIN
        write_ruler;  { refresh ruler line and rewrite entire text display }
        gotoxy(1,textline);
        write(copy(text_str,shift_amt+1,display_len));
      END;

  remainder := display_len - crsr_pos +1; { calculate remaining display space }
END;
```

(Listing continued)

(continued from page 53)

Using TYPIST

After you have answered the setup questions, the screen will display a 'ruler' line on screen line 15. The ruler shows tab stops as ' ', left margin with 'L', and the right margin with 'R'. If there is entered text beyond the right side of the display, the last ruler position will show '+'.

Text entry is all on line 14. As each character is input, it is displayed on the screen, the cursor is moved, and the status line is updated.

The cursor can be moved forward with ctrl-D, back with ctrl-S or <BS>, to the end of the entered text with ctrl-F, and to the beginning of the line with ctrl-A. <DEL> deletes to the left of the cursor, ctrl-G deletes under the cursor and ctrl- Y deletes to end of line. <TAB> (ctrl-I) inserts spaces to the next tab stop.

If an input line length greater than the display width has been specified, the display is automatically scrolled left or right to keep the cursor on the screen.

Line entry is terminated with <CR>, which causes the input text to be printed on the LST: device. If the input line is empty, the page can be ejected with ctrl-L (form feed). Finally, the program is ended with ctrl-Z, which also prints the current line.

Customization

If you don't like the choice of control characters, change the values in the constants. Display configurations other than 80 x 24 are accommodated with the constants DISPLAY_LEN, RULERLINE, TEXTLINE, and PROMPTLINE. DISPLAY_LEN should be set to one less than the actual display width, and the line positions to whatever you like. The maximum input length of 132 is hard-coded in CONFIG but can be changed up to a maximum of 255. If your typewriter is not the CP/M LST: device, the WRITELN statement in the main program will have to be changed.

Final Thoughts

I am sure many of you realize that if you are running Turbo Pascal, this program is totally unnecessary! Turbo has a full-screen text editor which handles in-

(Continued on Page 57)

(Listing continued)

```
PROCEDURE insrt_ch (ch : char);    { insert char into input string }
BEGIN

{ if line buffer full, delete last char of text }
  IF length(text_str) = linelen
    THEN
      delete(text_str,linelen,1);

{ bump char pointer and plug in char }
  position := position + 1;
  insert(ch,text_str,position);
END;


BEGIN       { inputstr }

  txtline := '';        { clear input }

  position := 0;        { reset pointers }
  shift_amt := 0;

  clr_prompt;           { status line labels }
  lowvideo;
  write('Cursor:      Input text length: ');
  highvideo;
  gotoxy(1,textline);   { clear possible previous display }
  clreol;


  REPEAT
    set_cursor;    { calculate display pos and refresh display to right }
    gotoxy(crsr_pos,textline);
    write(copy(text_str,succ(position),remainder));

    gotoxy(9,promptline);  { update status line }
    write(position+ord(position<linelen): 3);
    gotoxy(33,promptline);
    write(length(text_str): 3);

    gotoxy(crsr_pos,textline);  { return to input display pos }
    read(kbd,ch);               { input from kbd device needs no <CR> }

    CASE ch OF
      ' '..'~'  : IF position < linelen  { printable and within text? }
                    THEN
                      BEGIN          { yes }
                        insrt_ch(ch); { plug it in & refresh to right }
                        write(copy(text_str,position,remainder));
                      END
                  ELSE IF position=linelen  { at max input pos ? }
                         THEN
                           BEGIN  { yes, show char, then replace ch in input }
                             write(ch);
                             text_str[length(text_str)] := ch;
                           END;
      bs, move_lft: IF position > 0  { back up if anywhere to go }
                      THEN
                        position := position - 1;

      move_rt   : IF position < length(text_str) { move fwd if we can }
                    THEN
                      position := position + 1;

      move_strt : position := 0;   { reset to start of text }

      move_end  : position := length(text_str);   { set to end of text }

      del_crnt  : IF position < length(text_str)
                  { delete under cursor and refresh display }
                    THEN
                      BEGIN
                        delete(text_str,position + 1,1);
                        { refresh right, kill previously displayed last ch }
                        write(copy(text_str,position + 1,remainder),' ');
                      END;

      del_lft   : IF position > 0
                    { delete @ current entry pos if is text }
                    THEN
                      BEGIN
                        delete(text_str,position,1);
                        IF position=linelen
                          THEN write(' ')   { if @ max, just write blank }
                          ELSE
                          { else backspace, refresh right, space }
                            write(^h,copy(text_str,position,remainder),' ');
                        position := position - 1;
                      END;


      del_eol   : BEGIN      { delete to end-of-line }
                    clreol;
                    delete(text_str,position + 1,linelen);
                  END;
```

(Listing continued)

CALL WREN AND SAVE!

WORKSTATION

Computertone by Wren
- Z-80 4MHZ
- 5¼, 8 or hard drives
- Uses BBII CPU
- CPM, CPM86, MS DOS
- Defineable keys

Starting at $1955.00

DISKS:

SAVE!

5¼	SSDD	Dysan	$2.30	Verbatim	$2.70
5¼	DSDD	Dysan	2.95	Verbatim	3.75
8	SSDD	Dysan	3.10	Verbatim	3.40
8	DSDD	Dysan	3.50	Verbatim	3.80

Disk Mailers 10 for $5.00, 100 for $40.00

PRINTERS:

SAVE! OKI

182	P	IBM	8.5 Carr. 120 CPS	$ 229.
182	S	SER	8.5 Carr. 120 CPS	269.
82	A	S & P	8.5 Carr. 120 CPS	287.
83	A	S & P	13.6 Carr. 120 CPS	529.
84	P	P	13.6 Carr. 200 CPS	669.
84	S	S	13.6 Carr. 200 CPS	770.
92	P	P	8.5 Carr. 160 CPS	379.
92	S	S	8.5 Carr. 160 CPS	449.

DRIVES:

SPECIAL PRICES:

Shugart	810	$300.	Tandon 100-1	$ 155.
Shugart	860	310.	Tandon 100-2	199.
Tandon	848-1E	282.	Tandon 100-4	125. (when available)
MPI	51	94.	Tandon 848-2E	375.

CALL TOLL FREE 1-800-543-WREN

except in Ohio, 513-931-7160

Minimum Order $30.00

MasterCard VISA

WREN

880 Reynard
Cincinnati, Ohio 45231

"BMON"

Software In-Circuit Emulator

Links your CP/M computer with any Z80 based computer or controller that you may develop. All that is needed is BMON, 12K of ROM space, and a handshakeable bi-directable I/O port (either RS232 or Parallel).

Features:

—Full program development debugger with Breakpoints, Snaps, Stops, & Waits.

—Single Step program execution.

—Download file from CP/M system to development Ram.

—Upload Memory from development RAM to CP/M disk.

—Two versions: Master BMON runs in your CP/M system, Slave BMON runs in your target system.

Note: Requires Microsoft's M80 & L80 assembler & linker to setup Slave BMON.

8" SSSD Disk containing Master BMON, Slave BMON, CONSOL, BMONIO, CONSOLIO, and Users Manual $49.95

Shipped Via prepaid UPS
—No COD or P.O. Box—
Check or Money Order to:

Barnes Research & Development
750 W. Ventura St.
Altadena, CA 91101
(818) 794-1244

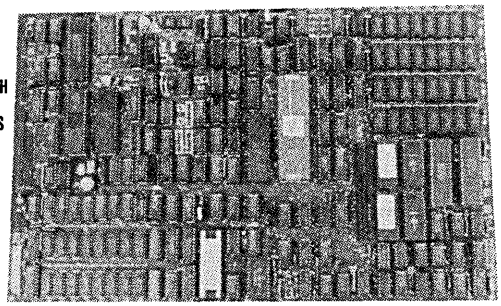
CP/M is a trademark of Digital Research Inc.
M80 & L80 are trademarks of Microsoft Inc.

FOR THE **BEST** OF US . . .

THE CYPHER T.M.

A COMPLETE 68000 & Z 80 SINGLE BOARD COMPUTER SYSTEM WITH ULTRA-HIGH-RES GRAPHICS!!

FREE 68000 FORTH AND CYPHER-DOS



FREE 68000 FORTH AND CYPHER-DOS

- 68000 & Z80 DUAL PROCESSORS (BEST OF BOTH WORLDS!).
- 256K to 1 MEGABYTE MEMORY. (4164 OR 41256 DRAM).
- DOUBLE DENSITY FLOPPY DISK CONTROLLER (8", 5¼" OR 3½" WD 2793).
- DMA CONTROLLER FOR FAST IMAGE TRANSFERS TO/FROM VIDEO MEMORY. (INT 8237).
- 2 RS232 SERIAL PORTS (ZSIO).
- 24 BIT ADDRESS MANAGEMENT FOR Z80.
- 4 LAYER P.C.B. (9¼" x 14¾").
- RUNS CP-M-80 2.2, CP-M-80 3.0, CP-M-68K
- ULTRA HIGH RESOLUTION GRAPHICS, 128K, PROGRAMMABLE UP TO 1024 x 1024 RESOLUTION (NEC 7220, GREAT FOR CAD SYSTEMS!).
- REAL TIME CLOCK (MULTITASKING CAPABILITY!).
- TWO CHANNELS OF D/A AND A/D, 12 BIT RESOLUTION (MUSIC! ROBOTICS! LAB WORK!).
- 16K TO 64K MONITOR EPROM.
- 4K TO 64K STATIC RAM.
- PROGRAMMABLE BAUD RATE GENERATOR.
- PARALLEL ASCII KEYBOARD INPUT.
- FULL 68000 EXPANSION BUS (60 PIN HEADER, BUFFERED).

- MANUAL $20.00
- BARE BOARD, EPROMS, Z80 BIOS, 68000 BIOS, Z80 MONITOR, 68000 MONITOR AND CYPHER DOS $399.95
- MINIMUM OPTIONED "CYPHER" WITH 68000/Z80, SERIAL I/O, 128K DRAM, 4K S RAM AND DISK CONTROLLER, ASSEMBLED AND TESTED $1,099.95
  PARTS KIT $999.95

- COMPLETE "CYPHER" WITH 256 K DRAM, 128 K VIDEO DRAM, NEC 7220, REAL TIME CLOCK, A/D D/A DISC CONTROLLER, SERIAL I/O, ASSEMBLED AND TESTED $1,499.95
  PARTS KIT $1,379.95
- KEYBOARD $94.95
- SWITCHING POWER SUPPLY $154.95
- CASE $94.95
- HARD DISC HOST INTERFACE PLUG-IN CARD $150.00

* EXPANDED MEMORY OPTIONS - INQUIRE *

VISA

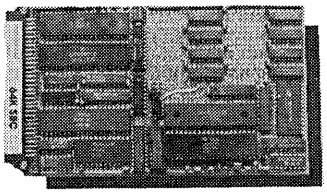
SHIPPING CHARGES:
ALL PRICES ARE IN U.S. DOLLARS
SHIPPING: 8% SHIPPING, EXCESS WILL BE REFUNDED

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.
CP-M IS A TRADEMARK OF DIGITAL RESEARCH.

MOTOROLA INTEL

MOTEL COMPUTERS LIMITED
174 BETTY ANN DRIVE, WILLOWDALE, TORONTO, ONTARIO, CANADA M2N 1X6
(416) 221-2340

New 64K SBC



Only

$375.

4"x6"

- **Requires no terminal. Includes Video Controller and CP/M® 2.2**
- **Runs any size floppy drive.**
- **Other models include Hard Disk Controller, CP/M® 3.0, 128K or 256K RAM, and 8088**

64K SBC includes:
- 6MHz Z80B®
- Video Controller
- 2 Serial Ports
- 4 Parallel Ports
- I/O Expansion
- Source Code and Drivers included
- CP/M® 2.2

Call our Toronto office today.
(416) 745-7214

Or write: Megatel
1051 Clinton St.,
Buffalo, N.Y.
14206

CP/M is a registered trademark of Digital Research Inc.
Z80B is a registered trademark of Zilog Inc.

Substantial OEM Discounts Available

(continued from page 55)

put lines of up to 127 characters. You need only enter the text into a file with the Turbo editor, then PIP it to your printer. The Turbo editor even lets you imbed control characters in your text (ctrl-P command) so that any special printer functions you have can also be used. The methods in the program could be useful for other applications, such as data entry.

Next Time

There are a lot of useful Pascal programs in the public domain (see Pascal Procedures in Micro C, Issue 22). Next time we'll look at some of the things you have to watch out for while translating between Pascal dialects (dialectic translationism?).

■ ■ ■

(Listing continued)

```
        tab       : IF position < linelen
                    { insert spaces to tab stop or end of text }
                      THEN
                        REPEAT
                          insrt_ch(' ');
                          write(' ');
                        UNTIL (position=linelen) OR (succ(position) MOD tabgap = 0) ;

        form_fd : IF length(text_str) = 0  { if line empty, eject page }
                    THEN
                      write(lst,form_fd);

        ELSE IF NOT(ch IN [ch,stop_char])    { invalid input, dummy! }
             THEN beep;
      END;  {of case}

    UNTIL ch IN [cr,stop_char];  { both = EOL }
    to := ch;
  END;      { inputstr }



  BEGIN { typist }
    init;
    REPEAT
      inputstr(txtline,ch);  { get input }
      writeln(lst,txtline);  { ship it out }
    UNTIL ch=stop_char;      { until get ^Z }
  END.
```

End of Listing

FORTHwords

By Arne A. Henden

7415 Leahy Road
New Carrollton MD 20784
(301) 552-1295

This has been a hectic two months for me. I finally defended my dissertation (after years and years), and am madly revising and making figures, trying to turn it in on time. Beware! Don't go for your doctorate unless you really want it!

Glenn Dixon recently sent me a letter I want to share with you. In the letter, Glenn analyzes FORTH, points out its weaknesses, and suggests improvements. He is a strong advocate of the language, but feels it is time to change. Rather than reprint his entire 4-page letter, I've taken excerpts and allowed room for a reply.

The Letter

" . . . When you look at FORTH columns in magazines (not just Micro C, but any magazine still brave enough to have a FORTH column) they are talking about how to make a menu, or how to swap an integer with a double precision integer on the stack.

"Good grief! Stuff like this should be so trivial that it doesn't even need mentioning. It is trivial in other languages, but not in FORTH. We're finally settling on a sort of standard for floating point numbers, but still can't decide whether to keep them on a separate stack or not.

"What happens when FORTH finally gets all the data types it needs (string, character, multi-dimensional string, etc., etc.)? Are we going to make a separate stack for each one? Are we going to have a + for integers, a 2+ for double precision, an M+ for matrices, a C+ for complex numbers (whoops, that's for bytes), and a $+ for strings?

"What words do we use when we want to pop a floating point number from the floating point stack, use it as a scalar on an integer matrix, then store its ASCII equivalent on the string stack? Can you imagine the garbage that would come forth from FORTH (no pun intended)?

"No, there was a time when FORTH made a lot of sense (back when memory wasn't cheap) but unless something changes, it will die a slow and ignominious death (it has already started).

"Here are some suggestions to one who may be brave enough to go against the FORTH purists and make something really useful:

1. Start from scratch. Hide the FORTH standards. Don't even keep them in the same room with you.

2. Separate the address list from the headers and data, and the primitives from both. The headers have no business being embedded in the address lists. Variables should have dynamic space available.

3. Where possible, implement NEXT in hardware. A simple circuit could be devised that searches for the next primitive while the CPU is executing the last one. This implies an address list in an auxiliary memory.

4. When deciding whether or not to increase complexity and ease of use at the expense of efficiency, go for the increase. This will kill FORTH purists, but make the language usable. A utility can be provided to pull out unnecessary error handling and other overhead once the program is debugged. By the same token, let the user start with a full blown system with all data types and utilities, and let him trim excess, instead of having him start with bare bones and add meat.

5. Get some real debugging tools—word definitions and comments in a random file, menus to display words by category, helps, real time stack display, etc.

6. It is too late to make a significant contribution with applications such as word processing, spreadsheet, data base management (in some areas), so go after the applications where FORTH can shine: scientific number crunching, artificial intelligence, communications, extremely large data base management, really smart real time control, etc.

7. Provide a programming environment that is comfortable. I'm sick and tired of hearing that the drawbacks of FORTH are inherent in the language. They are inherent in purists' heads.

"The only way FORTH can do what it is really able to do will be when it is released from the customs and conventions that bind it down and is made into a real, honest, mature programming language."

My Reply

Glenn's letter was valuable to me for two reasons: not only did he point out bad features, but he also gave suggestions to improve them. I'm happy to receive any letter, complimentary or not, but ones that offer solutions are my favorites.

I have never considered myself a FORTH fanatic. I developed UNIFORTH because of my interests as an astronomer. When you have to do consulting on a dozen processors and work with companies developing single-board computers based on state-of-the-art hardware, you need a consistent language that can be quickly ported to a new processor. UNIFORTH is the perfect language for my applications, but that is not to say it is the language of choice to write a word processor, spreadsheet program, or typing tutorial.

FORTH's strong points are: it is extremely compact; it is faster than any interpreted language; it provides an interactive environment ideal for program development; it is usually the first language available on any processor, so you can stay up with the times; it is extensible, so you can add new data types and other language extensions with ease; and it is cheap.

Vendor Blues

From the standpoint of a vendor, the major problem with FORTH is that it is readily available in the public domain. Vendors must act as "value added" merchants, providing more features than available in the freeware. Such features include floating point, strings, video editors, assemblers, documentation, and support. The nature of FORTH implies that these features are given in full source code. You don't get the source code for dBASE II or Turbo Pascal! Therefore, vendors are often possessive.

FORTH must be sold by merchants if the language is to be adopted by businesses, as they need the support of an established firm. At the same time, vendors are constantly asked why the customer should pay $100 or more for UNIFORTH when he can get FORTH-83 free. You can't have both worlds.

Is FORTH Dying?

Getting back to Glenn's letter, I completely disagree with him when he says that FORTH is dying out. I see more and more customers using FORTH, with the

variety of applications growing daily.

The purpose of this column is to present a library of applications and to teach some of the fine points of the language. Utilities such as forms generation and plotting packages written in FORTH have not been presented in any magazine. In fact, I have only seen such utilities in BASIC: other languages such as Pascal and FORTRAN require you to reinvent the wheel.

Inherent Problems

The problem with new data types is inherent (there's that word!) in FORTH if you want extensibility. If you create a '+' word that works with integers and floating point numbers, then it won't work with that new data type you just created. That is why each new data type has a separate (but consistent) set of operators.

The reason for the controversy on multiple stacks is due to speed and hardware considerations. The 8087 is a stack-oriented floating point processor, and should be used as such for optimum speed. The 68881 and 32081 are not. You can't have compatibility and speed at the same time, and so differences such as these will remain.

You can't hide standards. Just as OEMs won't use a hardware chip that isn't second-sourced, the commercial world won't use a language that is unique to one vendor or one CPU. There are certain applications where speed or compactness is important, and specialized forms of FORTH would be valuable. I don't like FORTH-83, but that is what is being demanded by the consumer, so UNIFORTH will support that standard in the near future.

Symbol table, data, and machine code separation is a feature that is useful on many processors. The 8088 is a prime example, where segmentation permits more than 64K bytes of memory to be accessed quickly if such separation is performed.

Implementing NEXT in hardware is not a software issue. I will discuss FORTH engines in the next column. Creating a CPU that uses a language as its instruction set creates the fastest possible execution of that language. The Pascal Engine on the S-100 bus and the Lillith machine for Modula-2 are two such examples. The new FORTH machines beat the socks off many mainframe computers for certain dedicated tasks.

Looking Ahead

I agree that increasing user friendliness is important, but memory is expensive. My feeling is that the FORTH implementation needs to be solid (no bugs) and the word names consistent so that programming is easy. A good debugging utility can handle most error checking during program development without slowing down execution of the final version. I looked at adding error checking that could be removed once the program was debugged, but I don't think it's possible without either doubling the source code size or slowing the system down by a factor of two.

Much of what Glenn says is true. My company is drastically revising UNIFORTH, not only to support FORTH-83, but also with a radical marketing departure. Release is anticipated in April. We will continue this experiment for six months before making any final decisions, so place your vote now!

Coming Next Time

Next issue I'll finally get around to reviewing 'Thinking FORTH.' If you are at all interested in FORTH, this book is highly recommended. Also, I've received a new single-board FORTH computer and have some other FORTH-engine news. Keep in touch—don't put off writing suggestions and comments!

Editor's note: Bruce has finished FORTH-83 for the Kaypro, so check out disk K32.

■ ■ ■

An Inspired Turbo Tutor

Review By Gary Entsminger

Micro C Staff

In many companies, documentation is done by an apprentice scribe who has been relegated to some obscure, cobwebbed corner. At Borland International, home of Turbo Pascal, that's obviously not the case, since Frank Borland himself wrote "Turbo Tutor." I think you'll like him.

"I live up in the mountains with my family, my dogs, and my burro, Lotus. It is peaceful here. Just the kind of place where a person can sit around and think and put things into proper perspective. Once in a while, when I really want to think through a problem, I take Lotus and hike a few miles from home. It was while camping under the stars that I got the idea to write Turbo Pascal."

How can you doubt the word of a burro driver who writes with an informal wit that's as fun as it is informative.

"It will help you in your learning to have a quiet and comfortable spot to place your computer. There should be room around it for you to place this book and your Reference Manual, and enough light for you to read them easily, as well. Learning anything new is hard enough; you should stack as many things in your favor as possible."

Dissecting A Tutor

He divides "Turbo Tutor" into three parts:

1. Turbo For The Absolute Novice
2. Turbo For The Programmer
3. Advanced Topics

Part 1 covers all those little (and big) details that other language manuals ignore. These are the start-up, get-into-it sorts of things that befuddle first-time programmers.

1. Start your Turbo Pascal program by following the instructions in your reference manual. Press Y for messages.

2. Press W and answer MYNAME when asked for the name of your workfile. (Don't forget to press RETURN.)

Any seriously interested novice should be able to learn programming with Turbo Tutor.

The Programmer's Guide

Part 2 is called the Programmer's Guide, and glides topic-by-topic through Pascal. This is where computer literates (though unfamiliar with Pascal) can jump in. The topics are clearly separated into short (4-17 page) chapters and include:

The Basics Of Pascal
Program Structure
Predefined Data Types
Control Structures
Procedures And Functions
Declared Scalar Types
Arrays
Strings
Records
Sets
Pointers And Dynamic Allocation
Files

Pointing And Filing

The construction of a 'Trek' game, with many ships, stars, and bases, helps tie these topics together. For example, if the number of ships in existence during the game varies, you might create a variable "Ships" which equals an array-[1..MaxShips] of (Ptr)AShip.

The pointer, (Ptr)AShip, is an address, and points to records of type, AShip. In order to point the pointer (to assign an address to it), use the predeclared procedure "New", which allocates the necessary amount of memory and sets the pointer to the appropriate address. "New" makes sure that the address the pointer contains represents an area of memory not being used by anything else. Pascal worries about those details for you.

When the game ends, as it sooner or later must, you can save your ship info in a file.

Assuming you have already created a type "ShipArray" which equals an array[1..ShipMax] of AShip, and Ship is a variable of type "ShipArray," save Ship by writing:

```
Assign(ShipFile,'SHIPS.DAT');
Rewrite(ShipFile);
Write(ShipFile,Ship);
Close(ShipFile);
```

"Simple, isn't it?" encourages Borland. "With a single Write statement, you have sent all the ship records out to the disk."

But why can an array be written to disk without a loop?

"Because Pascal differs from BASIC, FORTRAN, and several other languages in that it writes data files as binary files, rather than text files. Pascal stores a two-byte integer value as a two-byte integer value, not as a character string several

bytes long, with spaces and carriage returns as delimiters. This results in smaller data files and faster disk I/O. This also allows you to do things like read in or write out an entire array with a single command, rather than having to use intricate loops and heavily formatted I/O statements."

If you're not already an expert Pascal programmer, you'll learn a few things from "Turbo Tutor." And if you are, you still might.

Turbo And Standard Pascal

Turbo is based on Standard Pascal, and Frank and his co-writer (who writes most, if not all, of Parts 2 and 3) are good about pointing out the differences between Turbo and Standard Pascal. A brief history of Pascal pinpoints some of the reasons for Pascal creator Niklaus Wirth's omissions from the language.

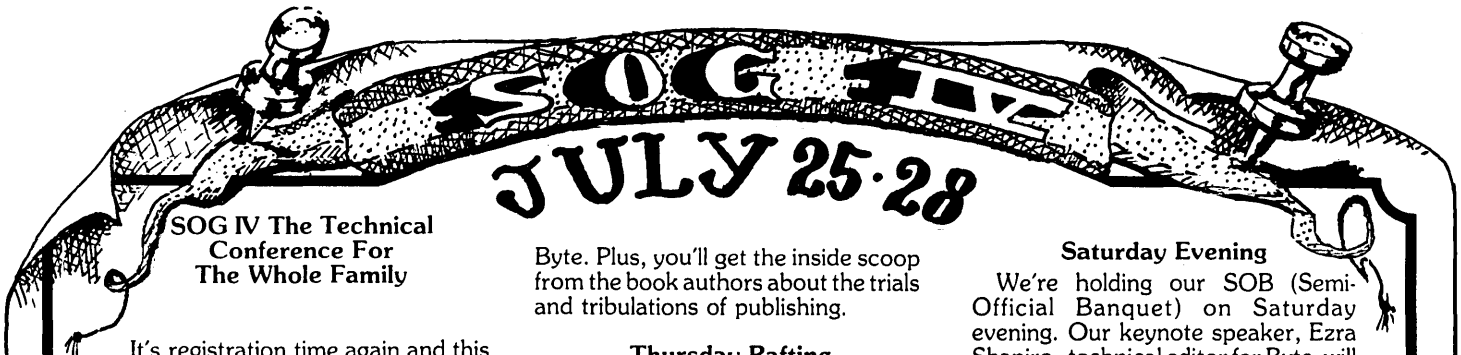
"When Niklaus Wirth designed Pascal, he did so in a punched-card/mag tape/mainframe environment, where fixed-length data was the rule. At least, that's probably the reason he was satisfied to store a string as an array[1..n] of char. At any rate, Standard Pascal does not (currently) have a predefined data type for strings."

But Where's The Index?

"Turbo Tutor," like other Borland products, is an exceptional value ($30 with a disk of programs) with few shortcomings. In particular, there's no index (Dear Frank, for a small fee, I'll gladly . .), and I hate rummaging through text for info I need now. This weakens Tutor's usefulness as a reference, but shouldn't create too many problems for students. Future editions, I hope, will correct this deficiency.

"Turbo Tutor" takes its place just behind Leo Brodie's "Starting Forth" as one of the finest programming language introductions I've read. Other language tutorials can learn much from Brodie's and Borland's informal, informative style. "Turbo Tutor," the Turbo Pascal Reference Manual, the Turbo Pascal Compiler, and your system are really all you need to learn good Pascal programming.

■ ■ ■



SOG IV The Technical Conference For The Whole Family

It's registration time again and this year's Semi-Official Get-together will be grander than ever. Like last year, there will be two days of top notch technical presentations sandwiched between two days of fun. This year, however, there will be more things to build, more sessions to attend (not that the schedule wasn't full last year) and special events Friday and Saturday for non-technical family members.

SOG IV is being held at Central Oregon Community College in Bend, Oregon. The dates are Thursday, July 25 through Sunday, July 28.

Free Conferences

Friday and Saturday: Two full days of free technical conferences, demonstrations, workshops, and forums (plus the new-product displays and swap meet). You'll have a great opportunity to exchange ideas with the Micro C editorial and technical staff, columnists, as well as other leaders in the micro industry. You'll get to talk with the main folks from Slicer, Ampro, Integrand, and Byte. Plus, you'll get the inside scoop from the book authors about the trials and tribulations of publishing.

Thursday Rafting

Once again we are kicking off the SOG with whitewater rafting followed by the Kickoff Cookout. If you are interested in safe thrills then sign up for one of these professionally guided trips.

The all day trip includes transportation from the college, box lunch, 4½ hours of whitewater (up to class 4), and the Kickoff Cookout. The Mackenzie River is famous for its whitewater and the road to the river winds along alpine wilderness.

The 2½ hour trip includes transportation from the college, 1½ hours on the river (up to class 3), and the Kickoff Cookout.

Or, you can choose to attend only the Kickoff Cookout (with the victorious rafters).

Thursday Evening

Following the Kickoff Cookout we'll adjourn to the college for a musical jam session. Bring your instruments and tin ears (or at least stop by for a laugh).

Saturday Evening

We're holding our SOB (Semi-Official Banquet) on Saturday evening. Our keynote speaker, Ezra Shapiro - technical editor for Byte, will follow the food (not just desserts).

Transportation

The nearest commercial airports are Redmond - RDM (15 miles), Eugene (120 miles), and Portland (165 miles). Shuttles run between Bend and the Redmond and Portland airports. Trailways Bus Lines also serves Bend.

If you need travel information, call Bend Travel, 503-388-3824 (they are really helpful folks).

Finally

If you haven't SOGged, then you haven't sogged. So don't miss this year's extravaganza. You'll have a lot of fun and learn a lot without getting soaked (unless you raft, of course).

If you're even considering coming, get this form filled out and in. We'll send you a free packet of information about the area. Also, if you're interested in staying in the dorm (holds two per room) you'll need to call Micro C to verify your reservation. We're limited to 25 rooms and they'll probably go fast.

REGISTRATION FORM

☐ Please send me the **FREE** SOG IV info pkg.

☐ I plan to attend the **FREE** conferences and activities

_____ Number of people in my party

Please list any special (non-computer) interests you have, i.e. fishing, hiking, rock climbing, sky diving, etc. _____

_____

_____

		adult	und.12	number	total
Thurs.	All day raft trip	$65	$60		
Thurs.	2½ hr. raft trip	$25	$20		
Thurs.	Cookout only	$10	$7		
Saturday	Dinner	$15	$8		
Dorms	Double room only For the nights of 7/24-7/27	$75			
Food Ticket 3 meals on Friday Breakfast & Lunch on Saturday All you can eat at COCC Cafet.!!		$20	$12.50		
SOG IV T-Shirt Indicate sizes: S ☐ M ☐ L ☐ XL ☐		$6.50			
				TOTAL $	

Make check or money order payable to: **Micro Cornucopia**

☐ VISA Card # _____-_____-_____-_____

☐ Signature _____ Exp. date _____

Name _____ Ph. # ( ) _____

Address _____

City _____ State _____ Zip _____

MICRO CORNUCOPIA
P.O. Box 223
Bend, Oregon 97709
503-382-5060

New Strokes For KSTROKES: Novice Programming

By David L. Hawkins

751 Page St. Apt. 2
San Francisco CA 94117

The sequence of events went like this: my wife and I bought a Kaypro 2X, read the copy of Profiles that came with it, subscribed to Micro Cornucopia, and ordered one of Micro C's Kaypro users disks.

When I received my copy of KSTROKES on disk #24, I was frustrated because it wouldn't do what I wanted. I could find no way to convert a non-shift character into a tilde ~. The tilde is used: (1) to indicate the beginning and end of a special function command, and (2) to invoke an already defined special function. I tried different combinations of KSTROKES and CONFIG, but nothing worked.

The reason I wanted a tilde is because I am a very bad typist, and I don't like using the shift key any more than I have to. For my convenience, I wanted to define number keys with special functions and didn't want to be constantly shifting to get a tilde (to end the definition). So what was the solution?

Discovering Assembly Language

Micro C had mentioned "Soul of CP/M" by Mitchell Waite and Robert Lafore. What a find! At last a book in assembly language I could understand. Those of you who have been using assembly language for a while may take for granted such simple matters as entering a program, but those listings had always baffled me. (Actually, when I got my first copy of Micro C, I almost gave up ever understanding what was going on.)

First, I learned that .ASM files were readable. (I had thought that an .ASM file was like a .COM file and couldn't be read with a word processing program.) So I opened KSTROKES.ASM (with WordStar), and found that I needed to make only two changes to get exactly what I wanted.

Since I have a Kaypro 2X, or 284, I had to change the BDOS location. Fortunately, this is easy since the changes are in lines 7 through 9. I just had to change the TRUE to FALSE for old 2 or 4, and change the FALSE to TRUE for new 2 or 4. (See Figure 1.)

The other change was in line 537. Since the period is the last one on the chart of changeable keys, I chose it to make my tilde. That way, if I ever change the keys I can change it or leave it a tilde. So, while in non-document mode, I altered the 017h to a 07Eh. Actually, I went ahead and altered all the keys, lines 535 through 537, but that was just for fun. (See Figure 2.)

I assembled my new KSTROKES

(Continued on Page 65)

Figure 1 - Lines 1 - 9 KSTROKES.ASM

```
;
FALSE   EQU     0
TRUE    EQU     NOT FALSE
;
KAYPRO2 EQU     TRUE    ; TRUE, IF YOU HAVE AN OLD 2 OR 4
KAY484  EQU     FALSE   ; TRUE, IF YOU HAVE A NEW 2 OR 4
KAYTEN  EQU     FALSE   ; TRUE, IF YOU HAVE A 10
```

Figure 2 - Lines 535 - 537 KSTROKES.ASM

```
NUKEYS: DB      00BH, 00AH, 008H, 00CH
        DB      01AH, 018H, 006H, 004H, 005H, 001H, 013H
        DB      011H, 00FH, 00BH, 019H, 014H, 007H, 017H
```

A Low Cost EPROM Eraser

By Christian Phaneuf

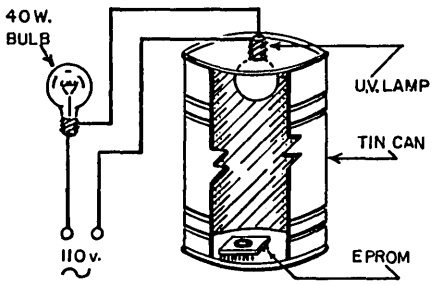
972 Guillaume Boisset
Cap-Rouge Quebec Canada G1Y 3E4

Here is a low cost homemade EPROM eraser for those who don't want to spend the $90 or more required for a good commercial unit. Of course, it can't brag of such features as a digital timer or a low friction ball bearing drawer for easy introduction of ICs in the erasing room, but it erases EPROMS pretty well.

A Light Can

First, get an ultraviolet lamp. I used a GE model 64 S11, but any ultraviolet bulb can be used. Next, you'll need a tin can measuring approximately 4" high by 4" in diameter (the kind used for roasted nuts is perfect).

Place the EPROM to be erased at the bottom of the can, and insert the bulb down into the can until it is about 1/2" above the EPROM's window. A 20-minute exposure is enough to completely erase a 2716 EPROM. Different bulbs, however, yield different levels of radiation, so it's a good idea to check for complete erasure and adjust the time accordingly.



The wiring must be done according to Figure 1, with a 40 Watt light bulb in series with the ultraviolet lamp. This serves as an important ballast. Failure to do this could cause the lamp to shatter. Also, be sure to shield your eyes from the ultraviolet rays (they could be damaged).

■ ■ ■

LETTERS

is the monitor ROM) before ordering a Pro-Monitor.

If the paper has 81-149 written on it (it may have a letter also), then you really have a II. If the paper has 81-232 on it, then you have a 4 board, and you don't have to change monitors to speed up your system to 5MHz (although disk accesses are faster with the Pro-Monitor 4).

The lockup problem could be the Z80 SIO or it could be that your RAM isn't quite making it. Try getting a Z80A SIO or Z80B SIO (the I0 version). The keyboard connects to the processor through the SIO, so if the SIO quits, it makes the system look like it's locking up. RAM is usually not a problem if you do the CAS and MUX change on U66. (The MUX line usually comes from U66 pin 4. Tie that line to U66 pin 3. The CAS line is usually tied to U66 pin 5. Tie it to U66 pin 4.)

If these changes don't fix the problem, then you might have gotten a marginal processor, or you might have a slow Z80 PIO. In either case, try the Z80A PIO as a substitute. Going to 5MHz shouldn't force you to add a fan unless a part is marginal. Use an ice cube in a plastic bag and see if cooling one of the above parts helps. If so, then that's probably your problem.

We have Z80B processors for $12 each postpaid. We don't advertise them because we aren't competitive with the big parts houses, but if you need some, we've got them.

PX-8 Support

I'd like to see you expand your coverage to include the new Epson Geneva PX-8 portable CP/M-80 computer. We're impressed with the one we have because it has a subset of WordStar. It works well and holds more than ten pages of text without add-on memory.

It seems to us that the Geneva fits your editorial purpose as a "single board" computer (no room for much else!), and by virtue of its CP/M operating system, it should be of interest to S-100 users.

Roger W. Brucker
21 Murray Hill Drive
Dayton OH 45403

RAM Disk & EPROM Programmer

I have purchased a Microsphere 512K RAM disk and am very pleased. It reduces the editing time necessary for working my MICROPAS program by one third and reduces the processing time by half or more. I think it is a great tool. (Now all I need to do is team this up with an Arithmetical Processor.) I experienced a few problems when using a twisted pair 26 conductor cable when connecting it up; only a short flat cable seems to work properly.

I tried to make the EPROM programmer. I can get it to read and verify EPROMS but can't get it to program one. I don't have a good 'scope so I'm lost trying to trace the problem. Any tips?

Eric J. Torney, Architect
7 Hart Street
San Rafael CA 94901

Recovering From Bad Edits

Your (our!) magazine is, as ever, fascinating and helpful. Perhaps the following will help others.

To recover after a crash with Perfect Writer or WordStar, I create a disk file of the dumped memory or of a non-editable disk file, e.g., PW.SWP. To write to disk, I use a very nice program, SPOOL, which diverts one device output to another, e.g., console to disk. It is available from the Smartkey people—SRT, 3757 Wilshire Blvd, Los Angeles CA 90010, and Nick Hammond (author of Smartkey), 16 Coles P1, Torrens, ACT, Australia.

Of course, I want only the ASCII part. Problem: How to strip the HEX. On long files—a 64K PW.SWP becomes about 300K—the column erase of WordStar runs the risk of generating a disk full error. The MBASIC listing in Tony Ozrelic's last column suggested an MBASIC answer to a friend (see Figure 1).

Still better would be a program that could be loaded into memory to strip the HEX from what is going to disk. Can someone help?

Michael Stocker
Philosophy Department
Oberlin College
Oberlin OH 44074

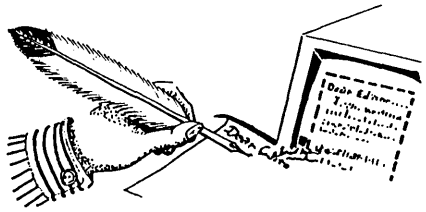
Software Copy Protection

I recently purchased 'PERT and Critical Path Techniques' from Lionheart Publishers. It seemed to be a good buy at $75. However, when it arrived, I discovered it was copy-protected and 'official back-ups' were available for an additional $25.

It seems underhanded to me not to put all your cards on the table before making a deal. Software companies which insist on this after-the-fact disclosure encourage the development and dissemination of sophisticated copy-protection-scheme breaker programs, and indirectly aid the less sophisticated pirates by the availability of the code-breaking software.

I don't mind the reasonably priced copies required to have a working back-up, but I do mind not being told up front!

Philip E. Burke
P.O. Box 90864
Nashville TN 37209-0864



■ ■ ■

Figure 1 - Stocker Listing

```
10 REM MBASIC PROGRAM TO STRIP HEX FROM DUMPX, SAVED DDT ETC
20 REM LEAVING ONLY THE ASCII
30 REM WRITTEN BY EROL MARTIN, 5 JAN 1985
100 INPUT "INPUT FILE (USE CAPITAL LETTERS): ", F$
110 OPEN "I", #1, F$
120 INPUT "OUTPUT FILE (USE CAPITAL LETTERS): ", F$
130 OPEN "O", #2, F$
140 IF EOF(1) THEN 210
150 LINE INPUT #1, L$
160 M$ = RIGHT$(L$,17)
170 N$ = LEFT$(M$,8)
180 PRINT #2, N$;
190 PRINT #2, RIGHT$(M$,8)
200 GOTO 140
210 END
```

Pascal And C Compilers Benchmarked

By Luis Basto

12707 Poquoson Dr.
Austin TX 78759

When the Turbo Pascal compiler was introduced, I felt the ad claims were exaggerated and suspected the benchmark times shown were derived from a system running 6MHz with a 256K RAM disk. I wanted to find out how fast it really was and compare it with some high level language compilers.

I used the "Sieve of Eratosthenes" program from BYTE, Jan. 1983, as a starting benchmark. Turbo can do its compilation entirely in memory (the fastest way, but there is a limit to program size) and to disk.

Compilation in memory takes less than a second, but for comparison purposes I compiled it to a disk. I used a stop watch with hundredths of a second resolution. Timing started when the RETURN key was hit, and ten runs were made to average out my inaccuracies. I calibrated myself with a time-response machine which measured the time it took me to react to a flashing light (about 1/10 second). So the results should be accurate to within 1/10 of a second.

Figure 1 - Comparison of Times

	Turbo Pascal	Pascal-Z V4.0	C/80 V3.0	Aztec C V2.0
Compile Time	10.53	1:36.16	1:25.72	1:21.50
Run Time	27.16	2:29.40	28.22	41.10
Size of .COM file	8K	4K	12K	17K

The Competition

One of the compilers in my test was Pascal-Z from Ithaca Intersystems. This Pascal has been quite popular and has one of the largest public domain libraries I know of.

I also compared Turbo with two C compilers. Turbo's bit manipulation functions and built-in BDOS and BIOS calls make C and assembly language programming pretty much unnecessary. (Of course, if speed were critical you might be forced to slug it out in assembler.)

I timed the Turbo compiles from the Turbo menu, then added the compiler's load time (6.26 seconds). With the other compilers, I let EX14 (the public domain submit utility) direct the compiling, assembling, and linking (as needed). That way there were no delays for typing.

The benchmarks were taken right out of BYTE with no modifications. All four compilers ran without any burps.

The times in Figure 1 were obtained on a Kaypro II running at 4MHz.

The smallest object file size that Turbo generates is 8K, even if you have a one line program. Turbo does not generate any extraneous disk files, while Pascal-Z and the C compilers create anywhere from 3K to 10K of relocatable assembly source files. These extra files are necessary since these compilers have separate assembly and link steps. Of course the separate assembly means that you have your choice of assemblers. (And Pascal-Z produces ROMable code.) So, these other compilers are more powerful than Turbo in these respects.

Intermediate Files

If you are writing large programs these extra files could create disk space problems. The one pass compilation and single object file output of Turbo certainly account for its short compile times.

To see if Turbo generates (and then erases) some intermediate files, I filled a disk with 170K of files and then compiled the Microcalc example which came with Turbo. MC.COM is 21K long, so if any extra files were created they would have overfilled the 191K disk.

Well, MC.COM was generated, and 0K was left on the disk. So much for the possibility of "erased files."

Installation

The editor is a little difficult to set up. The Kaypro selection from the terminal configuration menu does not understand Kaypro's cursor keys, which I found indispensable in editing. It also ignores the TAB and DEL keys. You have to remember that WordStar's commands can be a curse or a blessing. You can overcome this by using the CONFIG program to reconfigure the cursor keys. Then use the command configuration to re-define any edit key you want.

Conclusion

I like what I see in Turbo Pascal so far. It is a fast and very complete language. I must also give Borland International credit for shipping Turbo quickly. (No more JRT blues.) That seems to be a rare bird in the micro world.

■ ■ ■

KSTROKES

(continued from page 63)

.ASM with ASM, and used LOAD to create a .COM file. I ran it, and it worked like a charm.

Why Not CONFIG?

Why didn't I disable the new keys part of the KSTROKES program and use CONFIG? Because I like KSTROKES' flexibility. My wife uses a different number pad configuration on both of her editing disks. This way I can PIP the new KSTROKES.COM (renamed K1.COM) back and forth as we experiment with our different tastes in pad configurations.

On my games disk I can have one version of KSTROKES just for playing CASTLE. A 'tilde 1' calls up MBASIC CASTLE, while the number pad can give me some of the the other commands I use often. Since CASTLE uses directions a lot, I can have the '8' be North, the '4' be West, etc. I may not be getting better as a typist but I'm sure having fun learning how to use our Kaypro.

■ ■ ■

Running In CP/M's TPA

By Tom Geldner

3746 29th St.
San Diego CA 92104

TPA, CP/M, CCP, BIOS, BDOS, boot sector, system tracks, I/O, file control block, transient programs—these are just a sample of the alphabet soup that bedevils anyone trying to understand CP/M. Is there any hope for someone who is completely overwhelmed by the "CP/M Interface Guide" from Digital Research?

Actually, yes there is. This article and those that follow will cover CP/M and the ways that your high level programs can take advantage of this operating system's features (and how they can avoid the problems). This time, I'll take a look at how two compilers, SBASIC and Turbo Pascal, use the space they have available in the transient program area (TPA), the part of memory that isn't being used by the monitor or the operating system.

Transient Program Area

Do all Micro C readers know what CP/M's transient program area is all about? No? OK, here goes possibly redundant explanation #1 (technoids, stick with me).

Your CP/M operating system is divided into five distinct parts identified with mystical acronyms. Each part accomplishes a unique set of functions and occupies its own fixed area of your computer's memory (RAM). Taken in order from the bottom of memory to the top (0000H to FFFFH in a 64K system), they are the Base Page, TPA, CCP, BDOS, and BIOS.

The Base Page (0000H to 0100H) contains little bits of interesting and not so interesting information, such as where the BIOS and BDOS are located and command line information (the name of the program you want to run).

The TPA is where CP/M intends most conventional applications programs to run. The CCP (Console Command Processor) is sort of a program within CP/M that handles input from the command line (your A prompt), and is responsible for loading programs into the TPA and transferring control to them. (It's also where commands like TYPE, ERA, SAVE, etc. hide out.)

The BDOS (basic disk operating system) is the brains of CP/M and is used by applications programs to do most of the things related to system operations (disk file and console activity).

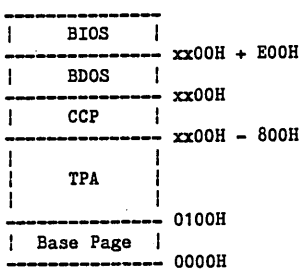
Finally, the BIOS (basic input/output system) is the part of CP/M that is customized for the hardware you are running on. The BDOS and the CCP remain the same; only the BIOS needs to be changed when bringing up CP/M on a new computer. The BIOS tells the BDOS where to find the computer's hardware (CRT, keyboard, printer ports, disk drives, etc.).

The standard routine is for a program to call the BDOS when it wants a character from the keyboard. The BDOS then calls the BIOS, asking for a keyboard character. (It is becoming increasingly common for programs to bypass the BDOS on certain functions and call the BIOS directly, although the hairs on many professional programmers' heads bristle at even the thought of this trend.)

Memory Map

Everyone has probably seen the little memory map below at one time or another. It's still a good way of illustrating the innards of your CP/M system. Note that the memory addresses are in 'offsets' from a known location, specifically, the beginning of the BDOS. The CCP is shown starting at the BDOS - 800H,and the BIOS is shown starting at the BDOS + E00H.

The specific values for your computer system are dependent on two things: the amount of total system RAM (nowadays, almost always 64K), and the size of the BIOS (large for systems with graphics, terminal emulation, hard disks, etc., or small for simpler systems or systems with a lot of hardware control in ROM).



The part we're interested in is the TPA, that area in memory which starts at address 100H and ends at the bottom of the CCP. When you run a program (one with a .COM extension), it will normally load at 100H and occupy RAM memory. How much RAM it uses depends on the program itself.

Word processors, spreadsheets, and other programs that need extensive amounts of RAM for file buffering will occupy as much RAM as is available. Since the CCP is used normally only at system level (the A prompt), some programs will overwrite the CCP temporarily to gain a little bit of extra space (2K or 2048 bytes), using all the memory between 100H and the bottom of the BDOS.

The BDOS is almost never overwritten, and the BIOS is absolutely never overwritten by a conventional program, since doing either destroys the 'guts' of CP/M. For programs like WordStar or Perfect Writer, as more RAM becomes available, less time is spent on disk accesses, and things like search and replace occur faster.

Memory Size Adjustment

Since the amount of available TPA varies from system to system, one has to be careful to take this variation into account when developing applications. Let's take the simple SBASIC program below that fills an array with 0s.

```
Example 1
-----------------------------
VAR x = INTEGER

DIM BYTE sample.array(49152)

FOR x = 0 TO 49151
  sample.array(x) = 0
NEXT x
-----------------------------
```

When this program is compiled, it takes up about 3K of memory. Add the 48K (49,152) occupied by the array and we come up with about 51K of memory used when the program is actually run. 51K will fit into the TPA of most CP/M systems. (Note that the DIM statement itself is the primary determinant of how much memory is used, since the code size is so small.)

Suppose we increase the array size to 57,344 (56K). Now our program will fit into a TPA of 59K or larger. Oops! We've just crashed on a Kaypro 10. The solution? Run-time memory sizing! In this case SBASIC out-does Turbo.

(Continued next page)

(continued from page 67)

There are two forms of run-time memory sizing. The first concerns the ability of the program itself to actually run, and the second determines how much memory the program can access.

Where Turbo Crashes

With Turbo Pascal, any .COM file produced expects the TPA to always be what it was at the time of compilation. This is because Turbo places its program variables and other goodies at the bottom of the BDOS, growing downward from a fixed position at the time of compilation. If we compile a program that has its BDOS starting at D300H, and we switch to a system with a smaller TPA (its BDOS starts at C300H), the Turbo produced .COM file will overwrite the BDOS and crash the system.

The solution for Turbo users is a menu option that allows the user to 'fake' a smaller system size by setting the ending address for compiled code to something less than that of the machine on which the program is being compiled.

If we compile a program on an old Kaypro 2, and want to make certain that it will run on a Kaypro 10, we can set the ending address to that of the K10's system size. In fact, to make certain that our Turbo program runs on a wide variety of systems, we would normally set our system size down to something like 48K or even 40K!

A Turbo .COM program can run on a system larger than that on which it was compiled, but the additional memory is not conveniently available for program use (it requires some fancy steps using special memory arrays). This is sad news if we're writing a word processor for commercial distribution that we'd like to run as fast as possible (with a minimum amount of disk accesses) on as many systems as possible.

A far more convenient solution is to have the .COM program itself determine at run-time how large the system is, and adjust accordingly. This is exactly what SBASIC can do! A .COM file produced under SBASIC determines where the BDOS is and then uses whatever memory is available. So it doesn't matter what system size we have used for compiling; the .COM file produced will work under a larger or a smaller system, and it won't crash simply because the BDOS is at too low an address. (It may run into a lack of memory if the program is too large and the system too small to contain all the code AND the variables, but this will happen to any program!)

Now we need a way to take advantage of SBASIC's ability to run under different system sizes and relate that to making use of all the memory available.

Dynamic Array Sizing

Dynamic arrays are those that can change size depending on the amount of memory available. Let's change our program in Example 1 a little bit to illustrate this.

```
Dynamic array sizing

VAR x = INTEGER
VAR max.memory = REAL

max.memory = FRE(0)
IF max.memory<0 THEN \
 max.memory = ABS(max.memory)+32767

DIM BYTE sample.array(max.memory)

BEGIN
  FOR x = 0 TO max.memory-1
    sample.array(x) = 0
  NEXT
  REM *** more program code ***
END
```

Here we have used the SBASIC FRE function to determine how much room there is in memory for our array before we declare the array size and, importantly, before we have declared any local blocks, procedures, or functions. (According to your handy-not-so-dandy SBASIC manual, FRE(I) returns the amount of free memory by comparing the 8080 stack and the procedure / function stack when I is false or equal to 0. FRE(I) can also be used to determine the amount of available disk space on the currently logged drive by setting I true or equal to -1.)

You may be wondering about declaring the variable 'max.memory' as type REAL and the inclusion of the IF . . . THEN statement. The reasons are due to some quirks of SBASIC.

SBASIC variables of type INTEGER are defined as having a range of -32767 to +32767. (Since SBASIC stores INTEGERS as 16-bit values, those 16-bit integers from 0000H to 7FFFH are considered positive and those from 8000H to FFFFH are negative for calculation purposes.) DIMing the array in terms of an INTEGER variable will be acceptable only if the memory available falls in the range of 0 - 32767. This is because the FRE function generates an INTEGER result. If more is available, the DIM statement will not understand what SBASIC considers to be a negative value.

Using type REAL as our DIM SIZE variable and converting any negative INTEGER results to positive REAL results ensures that we gain the true maximum amount available! This is the purpose of the IF . . . THEN statement. If the FRE(0) generates a negative result, we add 32,767 to it. If not, we leave it alone. (While it would be nice to add 7FFFH directly, due to an SBASIC restriction, only decimal constants can be added to REAL variables.)

Determining System Size

How do most programs determine how much memory they have available? Well, the Base Page of your CP/M system contains the memory address of the start of the executable code portion of the BDOS (the BDOS entry point) at memory locations 0006H and 0007H. The value at 0007H is the most significant byte of the address and the value at 0006H is the least significant (that is, if 0007H contains a D4H and 0006H contains a 06H, then the address itself is D406H or 53278 in decimal). Usually, but not always, this address is 6 bytes higher than the physical start of the BDOS itself. This is because Digital Research uses the first six bytes of the BDOS for your CP/M serial number with the working code beginning immediately thereafter.

To determine the TPA size, most programs simply ignore the least significant byte at address 0006H and assume that the physical start of the BDOS is always on an even page of memory (xx00H). Figure 1 shows a simple SBASIC program that prints your TPA space in both HEX notation and a decimal value.

The first statement looks at the byte value in memory location 7 (most significant), subtracts 9H from it (the offsets for the CCP and the start of the TPA at 100H), then multiplies the result by 100H. The second statement converts the integer value into a decimal one and, in a roundabout way, also illustrates an-

other quirk (or feature) of SBASIC.

The compiler treats constants as type REAL unless they are entered as HEX values. In this case, since the first value encountered in the statement is type REAL, SBASIC will convert the result of the entire statement into type REAL! Contrast this with Example 2 above where we had to declare a REAL variable in order to convert the INTEGER value to a positive number we could use. Here, SBASIC does it for you.

Biography

Tom Geldner is director of marketing and a general partner of Xpert Software in San Diego, producers of XtraKey and XScreen (assembly language keyboard redefinition and screen dump utility programs for CP/M computers). He has been using SBASIC for business and general purpose applications since its introduction with the first Kaypro 2s. And yes, he is even learning Turbo Pascal.

■ ■ ■

FREE CATALOG

KAYPRO USERS DISKS
for Kaypro II, 4 and 10 $12.00 each

KayPro Disk K1
Modem software

KayPro Disk K2
Utilities

Really oodles of spiffy little (and big) programs to help you get full use of your KayPro.

ZESOURCE.COM: A true Zilog format disassembler for 8080 and Z80 object (.COM) files. Now you can turn .COM files into . MAC files.

UNERA.COM: Simply enter "UNERA" followed by the name of the file you just erased and presto, the erased file is back! A lifesaver.

FINDBD54.COM: Checks an entire disk, reports bad sectors, and then creates a special file containing those sectors. You save a bundle on disks.

CAT2: This a group of programs which create and maintain a single directory of all the programs you have on all your disks. Even keeps track of which programs are backed up and which aren't.

UNSPOOL.COM: Use your KayPro II and print files at the same time. Doesn't slow down system response!

DUMPX, DU-77, COMPARE, SUPERSUB, FORMFEED, DIR-DUMP, . . . and all have documentation on disk.

KayPro Disk K3
Games

KayPro Disk K4
Adventure

This disk contains one 191K game, Adventure. ADV.COM: This is the latest, greatest, most cussed adventure ever devised by half-mortals. This is the 550-point version so the cave is greatly expanded and the creatures are much smarter.

KayPro Disk K5
MX-80 Graphics

KayPro Disk K6
Word Processing Utilities

KayPro Disk K7
Small C Version 2 Compiler

KayPro Disk K8
Small C Version 2 Source

KayPro Disk K9
ZCPR

KayPro Disk K10
Assemblers

KayPro Disk K11
Library & Checkbook Programs

CHECKS: This has been a very popular group of programs. Categorizes checks so you can keep track which are tax deductible and which get charged to which projects. Includes source and excellent example check files. Very powerful.

LIBR: This is a complete set of library routines which let you group files into a single file called a library. Then CP/M sees them as a single program, but with the library routines, you can list them out separately, run them separately, or divide them up again. Almost like a unix environment.

DISPLAY, VLIST, PGLST: Additional screen and print utilities.

KayPro Disk K12
FORTH

KayPro Disk K13
Source of fig-FORTH

KayPro Disk K14
Smartmodem Programs

KayPro Disk K15
Hard Disk Utilities

KayPro Disk K16
Pascal Compiler

KayPro Disk K17
Z80 Tools

KayPro Disk K18
System Diagnostics

Just as we finished editing the routines on this disk, we received a copy of KayPro's diagnostic disk. The memory test and drive exercise, routines on this disk are more powerful than KayPro's versions. (Plus, it's only $12!) Setup for KayPro II and 4.

KayPro Disk K19
Prowriter Graphics

Kaypro Disk K20
Color Graphics Routines

Kaypro Disk K21
SBASIC Routines & Screen Dump

SBASIC: Finally a disk of SBASIC software. There are some good examples of structured programming on this disk (including one program written both ways so you can see the difference).

SCREEN DUMP: This is a screen dump for all Kaypros new and old. You can buy a similar package elsewhere for $60.

Kaypro Disk K22
ZCPR (Again)

This disk is filled with ZPCR files. You get ZPCR for the Kaypro II, Kaypro 4, and the Kaypro 10. This version is fixed so that you can pass control characters (such as cntl-P) to the system and you can choose to have it recognize the semi-colon for drive select (as well as the colon). So you can enter "B;" or "B:" to select drive B. Super neat!

ZPCR, for those of you who don't know, makes CP/M a lot friendlier. It searches drive A for any .COM file it doesn't find on the current drive, the TYPE command scrolls text 24 lines at a time, and a new LIST command outputs a file to the printer.

Kaypro Disk K23
Fast Terminal Software & New BYE

Kaypro Disk K24
MBASIC Games & Keyboard Translator

We sifted through many, many games before coming up with these gems. All will work on any Kaypro and all come in MBASIC source.

USOPEN shows you the fairway on the screen. You select the club and direction for each stroke. After you reach the green the display shifts to show details of the green and flag. For one to four players.

DUCK is an offshoot of aliens (pardon the pun). Hunter tries to shoot down ducks while ducks try to bomb the hunter. (Much fairer than real life.)

CASTLE is an adventure in which you select your attributes (strength, dexterity, and intelligence) and you get to purchase arms and protection. Great documentation and very interesting game.

KSTROKES is a keyboard translator similar to Smartkey. Bill Forbes did an excellent job creating this program. You can create and save translation files on disk. The program even includes a table which generates WordStar commands from the Kaypro's keypad! You can define 8 keystrokes at up to 63 characters each.

Kaypro Disk K25
Z80 Macro Assembler

Kaypro Disk K26
EPROM Programmer & Character Editor

Kaypro Disk K27
Typing Tutor

A complete typing tutor for beginners and experts. Written in Australia, it comes complete with source. This was customized for Kaypro II, 4 and 10 by Barry Cole of WLAKUG.

The documentation says you can learn to touch type in 8 hours (probably a little longer for mortals).

Kaypro Users Disk K28
Modem 730

Kaypro Disk K29
Turbo Pascal Games I
With Source

Kaypro Disk K30
Turbo Pascal Games II
With Source

Kaypro Disk K31
Turbo Bulletin Board
Complete Bulletin Board Package For Only $12!

Kaypro Disk K32
Forth-83
Much Fancier Forth!

DISK $12.00 ea.
postage paid

Write or call
for a free catalog
(also included with each order from)

MICRO CORNUCOPIA

P.O. Box 223
Bend, OR 97709
(503) 382-5060
9-5 PST Monday-Friday

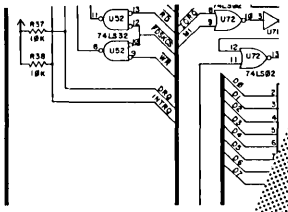
PRICELESS

UPGRADES

Schematic Packages
Finally, a complete schematic for your portable Kaypro, logically laid out on a single 24" by 36" sheet, plus a very complete illustrated Theory of Operation that's keyed to the schematic. You'll get detail information on your processor board that's available nowhere else.
For instance, those of you with the 10 and new 84 systems get a thorough rundown on your video section complete with sample video control programs in assembly language and Pascal. Of course, all packages contain serial and parallel port details and programming examples as well as complete coverage of the processor, clock, I/O, and disk controller (information that is not even available in Kaypro's own Dealer Service Manual).

Kaypro Schematic Packages
Kaypro II & 4 (pre-84) $20
Kaypro 10 (pre-84) $20
Kaypro 84 series (II & 4) $20
All prices include postage.

Special Offer!
Your First Schematic Package (If you also order subscription) $10



ROM'S from Micro Cornucopia

Pro-8 Version 2 for Kaypro 4

NEW!!

Guess what we've just upgraded. The venerable Pro-8 now has a version 2 and it's really neat! The new features include:

1. Screen dump with selectable dump character.
2. Select slow or fast step rate for each drive.
3. ROM that automatically figures out what type of drive you are using.
4. No more delays when using both single and double sided disks.
5. And, it's still only $49.95 for ROM, disk, and manual!

Of course, you get all the original Pro-8 features such as: user selectable cursor (blinking or not), ignores nulls, and your choice of 1-4 drives of the 191K, 390K, and 784K variety. (Use of 3 or 4 drives requires drive decoder.)

Installation requires no cuts or jumpers. The ROM simply plugs into a Kaypro 4-83 (or II-83 with a Kaypro 4 processor board), If you have a Kaypro II with the original II processor board then you must do the II to 4 upgrade. See issue #22 for details. Your Kaypro II has the original II board if the monitor ROM (a 20-pin chip with paper stuck to its top) is marked with 81-149. The 4 ROM is marked 81-232.

If you already have a Pro-8, you can upgrade to version 2 for half price. Call or write for details.

Pro-884 Monitor

NEW!!

The long-awaited PRO-8 ROM for the Kaypro 2-84 and 4-84s is ready! You have not saved your nickels and dimes in vain!

The PRO-884 ROM does everything your old Kaypro ROM does, plus it:

1. Gives you 784k bytes of storage per disk (with quad-density 80-track, double-sided drives).
2. Uses any combination of Kaypro II, IV, or quad-density drives.
3. Includes software to let you format and copy II, IV, or quad-density disks.
4. Comes with complete printed instructions on ROM installation and tips on installing quad-density drives.
5. Runs up to four drives with the Plus-4 Decoder Board.

Installation of ROM takes just five minutes and requires no cuts or jumpers.

Pro-884 Max Monitor

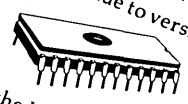
NEW!!

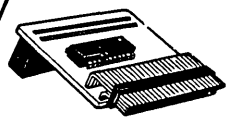
The ROM that does everything but wash the dishes is here! There's so much stuff in this ROM, even we didn't think it would fit!

The PRO-884-MAX ROM does everything the PRO-884 does, PLUS IT:

1. Emulates a DEC VT-52 or Heath H-19 video terminal (both the display and keyboard).
2. Does FASTER screen output.
3. Has ZCPR1 in ROM, eliminating the need for a system disk in drive A: during a warm boot.
4. Dumps the screen to your printer at the touch of a single (user definable) key.
5. Sends characters going to the screen to your printer regardless of the program you are running.
6. Displays the time on the Kaypro's status line (Kaypro 4-84s only). Because of battery backup, the time does not have to be re-entered each time you power up.
7. Inserts the time and date into text while you are running your favorite text editor (4-84 only).

Prices:
Pro-8 Version 2 Pkg. 49.95
Pro-884 Pkg. 59.95
Pro-884 Max Pkg. 79.95





PLUS-4 Decoder Board

With this nifty little plug-in board, your Pro-8 ROM can access up to four 5¼" drives. You just plug a four-drive 34-pin cable into this board and you can add up to two additional drives.
Now you can run any mix of 191K, 390K, and 784K drives as drives A, B, C, and D. You can run your original drives as A and B then add 380K or 784K drives outboard as C and D. You can even run four half-wides inside your original Kaypro!
The Plus-4 Decorder Board for only $39.95
Watch for 4-84 and 10-84 compatible ROMs coming soon.

SPECIAL PRO-884 NOTE:

The Pro-884s are sensitive to the version of CP/M you are running.

1. Neither the Pro-884 nor the Pro-884 Max will run on CP/M 2.2U. However, if you can locate a CP/M 2.2F or 2.2G system disk (your dealer should have a copy) you should be able to run our 884 monitors. (Don't try to boot F or G before you chunge monitors.)

2. There are two distinct versions of CP/M 2.2G. Only the Pro-884 Max is sensitive to the version of 2.2G you have - it's the ZCPR in ROM that's the problem. (If you have CP/M 2.2F then you have a Normal CP/M.) So, before ordering the Max, boot up your original system disk and read the sign-on. If it's CP/M 2.2G then we need to know whether it is the high (normal) version or the low (minus) version.

To determine your G version (you'll become a G Whiz!):

A DDT cr
**L5 cr*
(ddt's response)

The first line of the response will be a JMP D600 or a JMP D800. The JMP D600 means that you have a low (minus) version, and the JMP D800 means that it's a normal version. When you order your Pro-884 Max, be sure to specify whether you want the normal Max or the minus Max. Otherwise, we'll just guess that you need the normal Max.

On Your Own

By David Thompson

The last two "On Your Own" columns have pretty much written themselves, so I didn't think too much about the column while taking a final pass through this issue. Then I realized that I had neither copy nor idea. (You wonder what keeps editors from relaxing too much?)

Anyway, I started thumbing through back issues of Micro C looking for an idea when I spotted the ad for RP/M, a CP/M replacement. I noticed the ad ran in issue #18 (page 43), but then it had disappeared. That could mean a number of things, all of them interesting. Plus, the company (microMethods) is based in Warrenton Oregon, a sawmill and fishing hamlet on the Oregon coast. What a neat setting for a high-tech business. I picked up the phone.

microMethods

Jack Dennon is a soft spoken, relaxed sounding person, but he is intensely involved in microMethods. It is his full time job.

Actually, microMethods is two businesses. It designs computer controlled equipment for sawmills, and it publishes RP/M.

On the sawmill end, Jack has a partner, Bob Cameron. Together they designed a computerized head rig (the special saws and carriage that slice logs into boards) that efficiently handles small logs. Normally a person (sawyer) has to decide how to cut each log so that the resulting boards will bring the most income. The sawyer normally has to manually control the saw and carriage. Training time for this job is often a year or more. With the computerized system, all the sawyer has to do is feed the logs into the head rig; the computer takes care of the rest, measuring the size of the log and making the cuts.

On the RP/M end, Jack did his basic research while writing the book "CP/M Revealed." But first let's get a little background from Jack.

History:

"I quit Boeing in 1974, because I thought I knew how to design touchtone phones. After I had spent all my savings with nothing to show for it, I moved to the Oregon Coast to be near my folks.

"I moved because I thought I could use computers in mills to handle process control and lumber sorting. I wound up working at a mill in Warrenton for two years, and during that time came up with a lumber sorter.

"After the two years, I started my own business, and my first customer was the Warrenton mill. That one customer was enough to sustain me while I wrote the book."

CP/M Revealed

"I spent 3 or 4 months bashing out text and then shipped if off to Hayden. When they got it, the reviewer riddled it, but most of his criticisms turned out to be valid. (That first pass was about CP/M 1.4.) I wound up spending a year upgrading it to 2.2.

"While I was rewriting, I decided to include lots of example programs—an UNERASE program, a COMMON program. You know, lots of meat. Many of the programs were examples of interesting techniques—such as using standard CP/M system calls to unerase a file.

" The book was very successful, selling 20,000 copies in the first 12 months. Released in November 1982, it sold well for a year and a half, but since then, sales have dropped to almost nothing. B. Dalton still stocks it but that's about it. MT Publishing (the German software house that now owns Dr Dobbs) bought rights to print "CP/M Revealed" in German, but so far nothing has happened.

"The royalties made it worth writing the book, but more importantly, the book lead to the creation of RP/M, which generated even more income. The book earned between $15,000 and $20,000. RP/M has generated more than $20K. Most of that income has been through OEM sales.

"The hardest part of writing the book was the research. And it was the research that made it obvious that CP/M should be redesigned."

Anticipation

Jack mentioned that timing is the key to book publishing. He feels that "CP/M Revealed" would have been a tremendous seller if it had been released at the same time that Rodney Zaks' book hit the market. As it was, he caught just the tail end.

"If you do a book, try to anticipate what will be needed. Isaac Asimov already has a book out on Haley's comet. It takes a long time to write a book—then it takes a year to get it on the market."

About the time the book hit its stride, the lumber market went to pot. Jack's bread and butter sawmill declared bankruptcy, and other mills in the area were also curtailing spending. So he lived off book royalties and worked on RP/M.

RP/M Sales

Lately, sales of RP/M have been dropping, and he notes that about the time Microsystems died (his primary advertising spot), sales of RP/M really slowed down. The death of Microsystems and the market's abrupt swing to IBM haven't helped cash flow.

"I was selling 2 or 3 copies per week (to individuals), now it is 1 a month. Fortunately OEM sales are still pretty strong."

Sawmill Business Returns

Fortunately the sawmill business picked up again just as the book royalties and the software were dropping off, though it's not quite as lucrative as it was.

"I used to charge $60 per hour; now I have come down to $50 an hour for programming time and $35 an hour for on-site work. Over the years I have worked in only 4 mills, but each project has been 6 months to a year. They are good projects when you can get them."

Jack does his work in 8080 assembly language using the ASM assembler. He and his partner used to use a paper tape and teletype. When CP/M came along, they began using Altairs. Now they are using 8085 based multi-bus systems.

The Next Software Project

Meanwhile Jack has started on another software project. He is working on a focal interpreter for the IBM PC. Focal was originally used for doing calculations on a DEC PDP-8. The original 8 had only 4K of memory so focal is a very small language. It supports floating point numbers, but it lacks string and file handling. It is line numbered somewhat like BASIC.

Jack's Suggestions For Entrepreneurs

1. Have two or more things going at

the same time. Hopefully one will be hot while the other is not.

2. Find a need.

3. Your customer has to feel he is getting a bargain or must have some compelling reason to purchase.

4. If you have a mail order business, plan on spending more than you expect. If you advertise, for instance, you have to place a large enough ad and run it long enough to give your product a real chance. If you don't, you are wasting your money. Four issues in a row is a minimum. (When you read someone else's ad you're thinking about the product, not the cost of the space.)

Meet Jack At The SOG

Jack will be speaking on "An Insider's view of CP/M" at the SOG (it was a very worthwhile phone call). Come and you can ask him about CP/M and RP/M and focal and sawmills and . . . In the meantime if you're interested in a new headrig for your stud mill, or RPG (it has a neat bunch of extensions), you can reach Jack at 118 SW 1st St, Warrenton, OR 97146.

■ ■ ■

TECHNICAL TIPS

Anchor Signalman Mark XII Fix

I recently bought an Anchor Signalman Mark XII modem. Great product! It appears to be 100 percent Smartmodem compatible (haven't found any incompatibilities yet), and works with the BBI users disk #16 SMODEM software. However, when I initially connected it to the phone line, it wouldn't work.

It conversed with the BBI OK, and an over-the-phone-line check with another modem confirmed that the Signalman was transmitting. Unfortunately, the BB display indicated that it was receiving garbage, whether the remote modem was transmitting info or not.

Plugging in a telephone in parallel with the Signalman, I could hear distorted audio from a local radio station. In the amateur radio game, we call this RFI (radio frequency interference). Evidently, RF from the local station was being picked up on the phone wires, then was rectified (turned into audio, loosely speaking) inside the modem, where it interfered with the desired received signal.

The solution: Run a .001 mFd disc ceramic capacitor from each side of the phone line to a convenient earth ground. If you don't have an earth ground, you might try a chassis ground and see if that works. 1000V caps are cheap. Don't use .01 mFd caps. They will cause a slight AC hum on the phone.

I asked the phone company to check out the lines. I hooked and unhooked the caps while the technician monitored the line, and received a clean bill of health using the caps (modem disconnected). No problems have surfaced after using this setup for a couple of months.

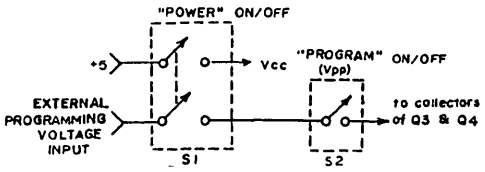
Bob Ghormley, KOBV
5800 Jones P1 NW
Albuquerque NM 87120

Improving A Kaypro EPROM Programmer

In Issue #18, Mr. Bardarson's article on an EPROM programmer for the Kaypro was most informative. I'd like to suggest an improvement. There's a simple way to make power up and power down sequencing idiot resistant. (After all, nothing is idiot proof!)

The data sheet on the 2716 from Applied Micro-Systems states that Vcc must be applied before Vpp or at the same time. The rules for power down are the same: Vcc must be shut off after Vpp or at the same time.

If we use a double pole switch for Vcc, one pole switches Vcc to the EPROM ZIF socket and the other pole is in series with the Vpp.



Richard Bugg
2703 N.W. 20
Oklahoma City OK 73107

A Little Less Drive In An 820-I

I would like to share some info relating to the 820-I and II. The October issue of Micro C carried a letter by A.W. Gustafsson in which he described a problem with a vertical line appearing on the video monitor.

I experienced the same problem using a Motorola DS-3000. It was solved by installing a 2K carbon pot across the horizontal drive and ground (slider to the monitor). It may be that there is too much drive for some monitors.

I also installed the disk drive modification explained in Issue #19 by Mitch Mlinar. I didn't like the idea of bending out pin 23 of the controller, so I isolated the pin by cutting the bus to it on the foil side of the board. You also have to cut the trace to pin 14 of U119 on the component side. This is the trace which is at a 45 degree angle. It's tied to pin 23 of the controller, and 5 volts on U119. The 5 volts required by U119 will be supplied by the trace on the foil side.

Anthony J. Gasbarre
23 Centre Street
Sullivan NH 03445

Adventure Words

Desperate for a competitive edge over ADVENTURE, one night I decided to do some exploring through my BB-I's RAM. Sure enough, after ending the game, the look-up table containing the game's vocabulary was right before my eyes. In alphabetical order, even!

To make a permanent copy of the word list, follow this procedure:

1. After quitting a game, reset the computer.
2. While in PFM, copy the list to 100H using the command 'C273,C6E,100'.
3. Boot CP/M.
4. Save the words to a file by typing 'SAVE 10 WORDS.ADV'.

The words can now be easily accessed with a BASIC program. Open WORDS.ADV as a random file with a field length of 6 bytes.

Purists needn't worry—no real secrets are given away. The main advantage to this is that you save time by knowing what words NOT to try. Be forewarned that indiscriminate use of certain words can be hazardous to your health.

Hal Vikks
Address withheld by request

Modifying An 820-I To Use A BBI Monitor

As a user of both BBI and Xerox 820-I for general hacking around, I soon learned that the 820-I monitor was a liability. Software that places any kind of routine above CP/M will probably overlay the monitor and crash unless you reconfigure to avoid the big 820 monitor. Examples of these kinds of software include: Dyna Disk, Graphtec, Scrndump, and many others.

The steps below take approximately two hours, and modify your 820-I to use the BBI monitor, thus circumventing compatibility problems.

1. Open trace (solder side) connecting to pin 1 of U107.
2. Add a pullup resistor (1000 Ohms) from pin 16 of U107 to pin 1 of U107 (solder side).
3. Open trace to U92 pin 19 (solder side).
4. Add a jumper between pins 19 and

TECHNICAL TIPS

21 of U92 (solder side).

5. Jumper pin 4 of U104 to your AC motor control. I recommend the unit described in Micro C, Issue 8, page 25 and Issue 10, page 32.

6. Install the stepping rate modification described in Micro C, Issue 19, pages 34 and 35.

7. Install an IC socket in location U117.

8. Bend out straight or cut off pins 1, 2, 3, 4, 12, and 13 of a 7406 before installing in U117.

9. Install a 16-pin IC socket in the spare location (between U34 and U36).

10. Connect the following jumpers on the solder side between U117 and this new IC socket.

U117	SPARE
1	13
13	14
3	15
4	1
12	2

11. Jumper pin 12 of the spare to pin 7 of U34 (solder side). NOTE: Pins 3 and 4 of the spare IC are available to implement DVSEL for drives C and D, if needed.

12. Jumper pin 10 to pin 11 on U105 (solder side).

13. Jumper J2 pin 8 to J2 pin 19 (solder side).

14. Remove R65 (150 Ohm) and replace it with a 220 Ohm resistor. Add a 330 Ohm resistor from the data side of R65 to ground.

15. Install a 7445 in the spare IC socket.

16. Remove the Xerox ROMs from locations U63 and U64.

17. Install a BBI monitor ROM in U64.

18. Power up, hit 'B' < cr> and PFM 3.3 will sign on.

D. L. Hedin
4210 Morris Road
Hatboro PA 19040

Kaypro Cursor Mods: The Assembly Language Version

Enclosed you will find a disk with a very small program on it called Flashoff.com. I realize it isn't much, but some readers with Kaypros might find it helpful. It simply stops the cursor from blinking on the 1984 2s, 4s, and 10s.

Kent R. Mason
4009 NW 24th Street
Oklahoma City OK 73107

Kaypro Cursor Mods: The Turbo Version

I recently stumbled upon a way to change the cursor on my Kaypro 4-84.

Send 10 (decimal) to port 28 to select the cursor blink rate register. Then send the number that selects the cursor to port 29. Not only does this register control the cursor blink rate, but it also controls the vertical size of the cursor.

I have included a sample program written in Turbo Pascal. Note that there are ranges of numbers that turn the cursor off. These ranges occur between the ranges where the cursor appears in the upper right corner of the screen.

Bill Tuck
Rt. 1 Box 222
Alberta VA 23821

■ ■ ■

Kaypro Cursor Mod - Turbo Version

```
var
   data : integer;
   key : char;

begin
data := 0;
clrscr;
port[28] := 10; port[29] := data;
gotoxy(1,1); writeln('1 -> decrement cursor data');
             writeln('2 -> increment cursor data');
             writeln('* -> exit to system');
             writeln;
      writeln('data sent to port 29D to create present cursor -> ');
repeat begin;
       gotoxy(50,5); write(' ',data,'    ');
       gotoxy(80,1);
       read(kbd,key);
       case key of
       '1' : data := data - 1;
       '2' : data := data + 1;
       end;
       port[28] := 10; port[29] := data;
end; until key = '*';
end.
```

Kaypro Cursor Mod - Assembly Version

```
;
org     100h    ; start program at 100 hex
mvi     a,10    ; put 10(decimal) in A registar
out     28      ; send value in A registar (10) to cursor registar
mvi     a,0     ; put 0 in A registar
out     29      ; send value to cursor registar
mvi     c,0     ; set up bdos for warm boot
call    5       ; call bdos
end             ; stop program
```

More Goodies From Micro Cornucopia

BB I, BB II, and XEROX 820 USERS DISKS

8" Users Disks $15.00 each

The following are full 8" disks of software. Each program has a .DOC (documentation) file and many come with source.

USERS DISK #1
1-Two fast disk copiers
2-The manual for Small C+
3-Crowe Z80 Assembler
4-Two disk formatters
5-Modem 7
6-Othello
7-Serial print routine-Port B

USERS DISK #2
1-Two single disk drive copy programs, both with source
2-Crowe Z80 Assembler source
3-New Crowe.COM file, debugged version
4-New CBIOS with parallel print driver & other extensions for CP/M 1.4 & 2.2
5-Disk mapper with source

USERS DISK #3
1-EPROM burning software for BB 1
2-Reset bit 7 (unWordStar a file)
3-Disk file CRC checker
4-New fast copy program & source
5-DU77, disk inspector/editor
6-FINDBAD, isolates bad disk sectors
7-Print fancy page headings

USERS DISK #4
1-CBIOS, custom bios for Tandon drives
2-ZCPR, dynamite CCP checks drive A for missing .COM files; improved commands
3-ZCPRBLOC, identifies CCP location

USERS DISK #5
1-CAT, disk cataloging routines
2-Modem 7 for Port A
3-Modem 7 for Port B
4-PACMAN, the arcade game
5-FAST, buffers the disk to speed up assemblies
6-NOLOCK, removes BB 1 shift lock
7-VERIFY, cleanup & verify a flaky disk
8-DUMPX, enhanced for BB 1
9-UNLOAD, create .HEX file from .COM file

USERS DISK #6
1-REZ, 8080/Z80 disassembler, TDL mnemonics
2-PRINTPRN, prints Crowe listings
3-RUNPAC, run-time utility package for 8080 assembly language programs. Has 51 functions. Includes source which assembles under ASM.

USERS DISK #7
1-CHNGPFM, PFM monitor mods
2-TERM, terminal routines let you set up BB as simple terminal, as a file receiver, or as a file sender
3-Checkbook balancing package
4-Disk Utilities - copy to memory, from memory, and dump.

USERS DISK #8
1-BDSCIO, custom BDSC I/O for BB 1 (both .h and .c)
2-YAM, Yet Another Modem program in source & .COM form. Turns BB into paging intelligent terminal, complete with printer interface, baud rates to 9600.
3-ROFF, text formatter
4-SIGNS, prints large block letters

USERS DISK #9
1-ADVENTURE, expanded 550 pt version
2-Keyboard translation program
3-CBIOS, serial & parallel printer interface
4-EPROM programming package for BB II, for 2732s only

USERS DISK #10 - Lots of Disk Utilities
1-REBOOT, sets up the CP/M auto load
2-SWEEP, directory/file transer routine
3-A, Lets BB I recognize a double sided drive as one drive with 494K of usable space
4-FIX, super disk utility, does everything, much easier to use than DU77
5-Compare files routine
6-UNERA, retrieve erased files
7-FIND, check all drives on system for a file
8-MENU, menu program for CP/M
9-NEWCAT, enhanced disk catalog program
10-Single drive copy program that does track by track copies rather than file by file

USERS DISK #11 - Printer Utilities
1-Microline 92 printer routine
2-Graphics display package for MX-80 with Graftrax, very fancy
3-Epson MX80 setup for BB 1 with 59.5K CP/M
4-Epson MX8 setup for any CP/M, lets you set print modes.
5-Micro Tek print driver, Ports A & B

USERS DISK #12 - Games for BB I
1-ALIENS, a fast, exciting arcade game
2-ZCHESS, chess with a 1-6 level look ahead
3-MASTERMIND, match wits with the computer
4-BIO, Biorhythm charts complete with graphics on the BB I
5-LIFE, so fast it's real animation!
6-CRAPS, see how much you'd lose in Vegas
7-WUMPUS, a caver's delight, kill the Wumpus or be killed
8-PRESSUP, similar to Othello
9-Games, 7 games in one program, includes blackjack, maze and animal

USERS DISK #13 - General Utilities, BB 1
1-ZZSOURCE, disassembles to real Zilog mnemonics
2-EX14, superset of submit or supersub
3-MOVPATCH, lets you use MOVECPM on other copies of CP/M
4-XMON, 3K expanded BB I monitor, use in ROM or as overlay .
5-CURSOR, prompts you for cursor char you want
6-UMPIRE, very fancy RAM test
7-ZSIDFIX, display improvement for ZSID
8-PIPPAT, modify PIP so you can reset system from within PIP
9-@, Lets you use the BB as a calculator, including HEX
10-SORT, sort package written in C80.

USERS DISK #14 - BB II Software
1-PRO32, latest 2732 reader & programmer
2-SMODEM2, lets BB II talk to Hayes Smartmodem
3-GRAFDEMO, demonstrates BB II graphics (in BASIC)
4-ATTRTEST, demonstrates BB II graphics (in JRT Pascal)
5-INITSIO, initializes port B for 300 or 1200 baud
6-MENU, displays menu of .COM files, enter number to run file
7-SETCLK, sets realtime clock built into BB II
8-PRINT2, modified print which accesses BB II clock
9-BOX, draws a thin line box on screen determined by HL and BC
10-ALIENS, space invaders arcade game
11-LISTSET, printer interface, auto-enables RTS, ignores DCD.

USERS DISK #15 - Word Processing
1-EDIT, very fancy line editor similar to EX (Unix). Includes help menu, programmable key, and full manual on disk.
2-TED, simple minded line editor, easy to learn & use. Very fast.
3-TTYPE, typing training program written in BASIC
4-TINYPLAN, very simple-minded spreadsheet. Whets your appetite for a fancy one.
5-C80 Text Utilities
6-CHOP, cuts off file after N bytes
7-ENTAB, replace spaces with tabs where possible
8-MS, double or triple spaces a file to output
9-RTW, removes trailing spaces from file
10-TRUNC, truncates each line to specified length
11-WRAP, wraps at column 80, plus pretty pretty printing, page #s . . .

USERS DISK #16 - BB I Modem Software
1-RCPM27, list of U.S. bulletin boards
2-SMODEM, interfaces BB I with Hayes Smartmodem
3-PLINK66, easy to use with non-CP/M host, for port A
4-BBPAT, menu selection of BAUD rate, bits/char, parity, & stop bits
5-MODEM 7+, Modem 7 plus BBPAT, lets you talk to anything from port A

USERS DISK #17 - Small C version 2
SMALLC2, this substantially expanded version of Small C now includes for, goto, label, switch (case); external declarations; new preprocessor commands; expanded I/O includes redirection; initializers; plus 12 new expressions. The I/O and runtime libraries have been greatly expanded (including printf). Source & documentation on one full disk.

USERS DISK #18 - FORTH
IFORTH, this is Idaho FORTH which can be burned into ROM or loaded from disk. It replaces the PFM monitor & handles all the monitor functions. See issue #11 FORTH column for more info about IFORTH and this disk.

USERS DISK #19 - BB I Double Density
New BB I Monitor, BIOS, character ROM, Winchester Interface, ZCPR, and formatter from Trevor Marshall. See BB I expansion article in Issue #11.

USERS DISK #20 - Assemblers
CROWEASM: This is the Crowe assembler modified so that it runs on any CP/M system (including the BB I, BB II, Xerox . . .). Includes .COM .Z80 and .DOC files.
LASM: This assembler is similar to the ASM that comes with CP/M except that it can link files at assembly time.
PRINTPRN: Print routine for CROWEASM .PRN files.
LIBRARY: Utilities which let you combine many files into one, then you can run, type, or extrace any file within the larger system.

USERS DISK #21 - Winchester Utilities
BACKUP: Helps you back-up the winchester onto multiple floppies. Creates a catalog of the files on each disk and includes the date of the latest backup. Will not back-up an unchanged file more than once. Plus many more super features.
FLOPCOPY: Lets you make floppy copies (with only one floppy drive) by using the winchester as a buffer.
BIGBURST: Backs up a very large winchester file onto multiple floppies. Joins the copies to recreate the original file.
MULTCOPY: Use this like PIP but it prompts you to change disks. Accepts ambiguous file names.
MDIR: Displays files in all user areas on selected drive. Many features.
MAKE, MOVE: PIP-like utilities that make it easy to move files between user areas.
SWEEP: The famous disk cleanup and transfer routine that does just about everything you can do with TYPE, ERA, DIR, and PIP.
UNSQ: This is the latest, greatest file unsqueezer. Enter UNSQ *.* and it will check every file on the disk. All squeezed files will be unsqueezed.

USERS DISK #22 - Pascal Compiler
This is a real Pascal compiler. It supports only a subset of the language (no records, pointers, booleans, reals or complex) but it generates a real .COM file. Everything is on this disk: the compiler, its source, example programs and documentation.

USERS DISK #23 - Xerox Utilities
This disk contains Xerox specific utilities including a screen dump fromWayne Sugai(with source); modifications for the SWP package including ZCPR, a new monitor, and a clock/calendar from Mitch Mlinar; and Jim Mayhugh's new monitor (see issue 19). A very special disk for Xeroxers.

USERS DISK #24 - Prowriter Graphics
This is a complete Prowriter printer graphics package written by the same Micro C subscriber who wrote the MX-80 graphics package. Plot points, lines, circles, boxes, and more. Examples, documentation.

USERS DISK #25 - Z80 Macro Assembler
This is a real Z80 macro assembler! Syntax closely follows RMAC and MAC. Also includes pseudo-ops to support conditional assembly etc. No phase or relocatable code.

USERS DISK #26 - BBII CP/M 3.0 Banked BIOS/ Winchester Support
CP/M 3.0 Banked BIOS implementation for the BBI. Roy Epperson's software to support the Adaptec ACB-4000 SCSI and the Rodime R204 5" Winchester on the BBII (see issue #19). Plus more Winchester programs.

USERS DISK #27 - BYE Remote CP/M System
BYE programs to run your BBI, BBII, or XEROX 820-I as a remote CP/M system using a Hayes Smartmodem compatible modem. Includes programs to allow restricted access.

USERS DISK #28 - VFILER and Extended Single Density
VFILER is a screen-oriented file manipulation utility, similar to SWEEP, CLEAN, and DISK. Also, Larry Blunk's documentation and software for implementing extended single density (334K) on eight inch disks.

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709

"You really think you're going to get all those bargains in a suitcase?" he asked.

Well, I'll do the best I can, and let UPS handle the rest. While I'm there I'll be giving a talk entitled "The Big Board, Xerox, and Kaypro: Early Single Board Clones." Plus, I'm going to have a small booth in the commercial display area. David Chasen and his Trenton Kaypro Users Group will be helping me man the booth (thanks, David). If you're in the area, stop by and say hi.

Definitely April

By the time you read this it will be April. If you see anything at all strange in this issue . . .

On the other hand, if you don't see anything strange in this issue, maybe you should be reading something a little more humorous (the federal tax code, perhaps).

Gary Entsminger

Every magazine should have its Gary. We finally do and I'm tickled. Gary is so excited about this and that and the other—that I'm having trouble constraining him. (Actually I'm having trouble constraining him because I'm excited, too.)

Anyway, Gary has suggested two new projects for Micro C: "The Last Page" and "Tidbits."

The "Tidbits" section gets all those interesting new bits and pieces that I haven't really had a spot for in the past, and it should make interesting reading. In fact, Gary just went dashing by waving a press release with "Borland" emblazoned across the top. I guess I'll have to take a short break here, and see what Philippe is up to. Be back in a couple.

OK, Gary will tell you about Borland's new packages for the Epson (oops, I'm letting it out), and he'll also explain "The Last Page" on the last page.

Remex Drives

In the last issue I reported the deal on Remex drives. Well, I can't resist a bargain myself, so I bought six of them for $300.

I think I know why the Remex drives are so cheap. They aren't very good drives. Out of the six, two are still running fairly well. (One was DOA, and three more died within days.) I understand these drives were left over from the Zorba debacle, and these could well have contributed to Zorba's demise.

Bruce and I took a very close look at the units to see if we could figure out why they have trouble reading track 0, why they are noisy, and why other drives have trouble reading disks formatted or written by these drives.

Well, first we checked out alignment. That seemed to be within reason. Then we discovered that the head assembly didn't run smoothly over the rails (it really dragged), and that the stepper motor has a long, unsupported shaft. Between the shaft length and the drag on the head, there's little chance the motor could consistently hit the track properly.

Also, they appear to be very heat sensitive. When they're cold, they won't boot; when hot, sometimes the spindle motor loses its speed control (it sounds like you've put your foot down on a sewing machine control). I can't really blame the motor for the overspeeding; the oscillator that drives the motor was no doubt going wild.

Anyway, if you have any Remex drives, try lightly lubing the rods that the head assembly runs on. Use silicon lube or WD-40.

If you haven't bought your drives yet, then check out the prices on the Mitsubishis and Shugarts. I've seen Shugarts for $109 each, and they are really quiet in our Kaypro. (Shugart is no longer in business, by the way. Another tombstone has replaced a cornerstone.)

Florida

I spent two very interesting weeks in central Florida this last January. During that time I met with the officers of the Central Florida Computer Society. Founded in 1977, it's the oldest computer society in Florida.

Anyway, these folks are old S-100 hands who have also dabbled in other systems like Kaypro and Osborne. Many of them still have at least one S-100 system, but I wonder now if they're keeping them because they use them, because they're antiques, or because they're too monstrous to move. It turns out that it's a bit of all three.

I expected to hear strong appeals to keep Micro C a CP/M only magazine. I heard a little bit of that, but it was very muted, almost apologetic.

What I did hear was that folks hated IBM, but they were buying them, and the purchases seemed to be within the last few months. Often the reason was the company they worked for, or the need for a specific piece of software that wasn't available anywhere else. They were very practical reasons, but they went against the grain.

"My (CompuPro/Kaypro/Morrow . . .) outperforms the IBM, but I needed to have a PC at home so I could (use programs that I use at work/interchange data with the office system/do a specific graphics project)." Even after Comdex, I wasn't ready to hear a bunch of diehard CP/M freaks talk openly about going over to the other side.

One Friday evening I attended an informal officers' meeting. At that meeting I asked each person to pretend he were in my shoes. Where would Micro C be heading if he were at the helm?

One stated that he wouldn't be caught dead in my shoes (I guess he knows what the hours are like). He doesn't like IBM because they're slow, but he is buying a second XT because it is cheaper than adding on to his S-100 board.

General Suggestions:

1. Add an S-100 column.
2. Continue very strong support of the Kaypro and Big Board.
3. Provide regular information on Turbo Pascal, C, and FORTH.
4. Increase information on public domain software.

Well, this is the easy part. We're already doing these. Some other suggestions include:

5. Micro C should carry more hardware design and construction projects. These projects should include controllers, I/O interfaces, printer buffers, terminals, local area networks, battery-based power supplies, and S-100 boards, to name a few.
6. Micro C should take close looks at the insides of different operating systems, starting with CP/M, CP/M 86, CP/M 68K, and then move on to some of the new single and multi-user systems.
7. Micro C should add a regular feature aimed at beginners. Subjects would include hardware, assembly language programming, and CP/M 80.

Great ideas. I'm looking forward to these articles already, so if you are working on anything resembling these, keep us in mind. In fact, give us a call or drop

(Continued next page)

(continued from page 77)

us a card if you have an idea. Direct your inquiry to Gary or Becky (or even me).

Be sure to let us know what disk you want, as well as your shirt size when you submit your article or tech tip. We'll send the disk right away and the shirt if it runs. (Actually, our shirts aren't supposed to run, but you never know. It all comes out in the wash.)

Surplus Information Wanted

I'm interested in the latest info from the surplus folks. It doesn't have to be big surplus folks (skinny, short types are OK, too), but there are a lot of little batches and big batches of super deals that people just stumble onto. These deals could go into the "Tidbits" column, or they might get their own "Surplus Tips" section.

If you find something incredibly incredible, drop us a note or call with the particulars on the product, the price, the address/phone number, and how to get it.

Xerox

Those of you who have been following this column for the past few years know that Xerox Corp. and I (actually Xerox and everybody, it turns out) have had a sort of standoffish relationship.

I had called them about supporting the 820. One department said OK, but engineering treated me like I was the Boston Mangler (without even seeing one of the boards I'd modified).

A couple of issues ago, Mitch Mlinar mentioned that Xerox had dropped the 820-II. That produced an immediate call from Xerox marketing (East Coast) saying they were still selling the 820-II, and would I please straighten things out for our readers.

"Sure," I said.

When he called back to make sure I'd done it, I started to wonder if maybe there wasn't more to the story than just a simple misunderstanding.

Now I'm beginning to hear some other interesting things. Computer marketing is on the East Coast because it's no longer in Dallas (seems simple enough). In Dallas they say that the 820-II is not a current product. Obviously the Southwest mouth knoweth not what the East mouth sayeth.

So, I called the local Xerox dealer and tried to order an 820-II. The lady I spoke with sounded surprised that I wanted an 820-II.

"Are you sure you want an 820-II?"

"Yes," I said, trying my best to sound casual—it's not every day I order a dinosaur by phone.

"You wouldn't consider a used one, would you?" Her voice quavered. "We have a number of customers who would love to sell theirs."

"No, I want a new one." I said, determined to get to the bottom of this.

"I'll have to call our main store in Eugene, and see if they can still order it."

Later that afternoon, I got a call.

"We can't get the 820, but we can probably order the 8/16. It's an upgraded version of the 820."

I resisted the temptation to ask if the 8/16 were PC compatible (it isn't), and then mentioned I had placed the order to check up on Xerox.

Her general feeling was that it wouldn't make much difference whether or not Xerox still stocked the 820 (or the 8/16); there just wasn't much demand. (No question, it's all very simple.)

ATs And Drives

Kaypro is coming out with a new AT compatible (it's being FCC tested now). I wonder if that means it will have flaky winchesters (IBM's are).

I understand that IBM refused to pay IMI $360 per 10 meg drive (they insisted that IMI reduce their price to $280 per drive to match the price they could get from Japan), so IMI simply closed its doors when the contract with IBM expired. It turned out that something like 80% of the Japanese drives have been dying within a few hours.

So Old Blue has egg foo yong on its face and ATs in its warehouse. Up until now IBM has been pretty much invulnerable in the business community because it has such a reputation for quality. That reputation is getting nailed by this drive problem (IBM suddenly looks a lot less invulnerable).

Not only has its reputation for quality been nailed, but other companies are suddenly very leery of IBM contracts. They've realized that they are literally betting their companies when they sign on the old dotted.

DTACK Grounded

I don't know why DTACK Grounded isn't D'TACK Sharp, but then I don't know why Dr. Dobb's doesn't make house calls. Anyway, DTACK Grounded is an interesting gossip rag that covers just about everything in the industry: IBM, Apple, Macintosh, etc. It's sort of a stream of consciousness thing.

I just received my first copy and have read it cover to cover—excuse me—front page to back page, there are no covers. This 28-page piece is probably what people first expected us to produce for the Big Board—reduced dot matrix, quick-printed, and stapled in the corner. It's not fancy, but it's fun all the same. Some of the material isn't very intelligible or very useful, but the rest of it is pretty interesting.

For instance, the following:

"Remember Jean Claude Cornet, whom Intel yanked back from France to fix the 186 mask? He is working on the 386, a 32-bit micro which will maintain upward compatibility with the 8088, 8086, 186, and 286. How this will get pulled off is going to be interesting to see: software compatibility requires a 64K segmented architecture ... Segmented architectures STINK when attempting to work with multimegabyte memories. G. Gordon Bell, DEC's former VP of Engineering, has repeatedly asserted that the biggest mistake a computer architect can make is to use too small a linear address space. Intel has not yet ventured beyond 64K—and neither does the Zilog 80000 (five zeros)—honest!."

I'm not sure I count five zeros in 80000, but I am sure I'll be mentioning more from DTACK. Also, I agree completely with their feelings about segmented addressing. Intel made a bad boo-boo when it limited itself to 16-bit registers in the 8086, and now its future products are locked in by the need for compatibility. (Their 8080 was also limited by the architecture of the 8008.)

DTACK Grounded
1415 E. McFadden, Ste. F
Santa Ana CA 92705
$15 for 10 issues, U.S.

Turbo Tutor

In every training package that's come across my desk (or through my computer) there has been one major flaw. The

flaw is that they haven't exorcised the steps. (But steps are good exorcise, you say.)

The problem with steps is that they become impossible hurdles for people who can't quite pull themselves over the top. What these packages need are ramps, steadily rising paths that take students from 0 to wherever they're going. No steps.

Well, I have yet to find a Pascal course that wasn't full of steps. I learned Pascal at Tektronix, and that course had all kinds of steps. The biggest step was logging onto the time-share system (and then there was the impossible text editor and the two-day wait for a printout). Finally, the instructor and the book added more than a few steps of their own.

When Micro C staffers took Pascal at the local college, they faced a lot of steps even though they got permission to use the Kaypros in the office. I wanted them to learn Pascal before they learned the negative mind set that comes free with BASIC, but the class was really aimed at people with some programming (BASIC) experience.

Anyway, to make a long story short, I looked very closely at the Turbo Tutor when it arrived from Borland International. With this package, Borland had a chance to teach Pascal in a very defined environment. They knew the version of Pascal they were teaching, and they knew the editor people would be using.

And they did it right! No steps! This package should teach you Pascal if you're at all interested (and it's so well written that you'll be interested). This disk and book contain a very easy ramp that starts at 0 and goes a very long way. You simply step onto the ramp at the level that's comfortable and you're off. I highly recommend Turbo Tutor for people who have never programmed before, as well as for people who are old hats at the language. Borland gets a 10+ on this one. (Before I steal any more of his thunder, see Gary's detailed look at the Turbo Tutor in this issue.)

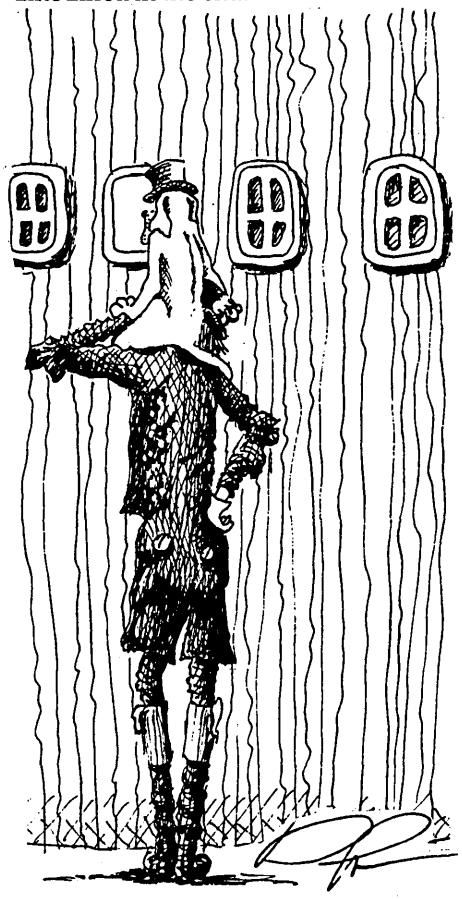
Computers And Electronics Folds

Another cornerstone of the computer community has bitten the dust. Just over 10 years after they announced the Altair 8800, the first affordable computer kit, Computers and Electronics ceased publication. The magazine was called Popular Electronics during the Altair period.

On February 19, Alice called Ziff-Davis (recognize the name?) to get the ad rates and deadlines for Computers and Electronics. What she got was the grand shuffle through the advertising department, but no information. Then Dave Pogue tried calling, and he wound up in the editorial section (advertising was not available). No one knew anything about deadlines or rates.

Finally we got a call back from advertising. The caller indicated she had just found out the magazine had been shut down. The April issue will be its last.

We quickly ran down to the library to see what had happened to the pub. What we found was a mere shadow of its former self. Advertising was noticeably scarce (at $12,000 for a full page, I'd be scarce, too), and most of the articles were shallow reviews. I guess we should have guessed it was coming. Another magazine ziffed in the end.



David J. Thompson
Editor & Publisher

■ ■ ■

WANT ADS

The following folks are reaching you for only 20 cents per word. If you would like to reach the same audience, send your words and 20 cents for each to Micro Cornucopia.

Big Board II system. BBII in Ferguson enclosure with super duty power supply, Siemens 8" DSDD drive, 10 MEG RODIME Hard Disk (new) & XEBEC. Amdek 12" amber monitor with cherry keyboard in enclosure. CP/M 2.2, FRIDAY! + users disk #21 utilities. $2295. Verbatim Datalife Diskettes 8" SS/SD-$25/box, 8" DS/DD-$34/Box. Wade Noxon, Players Computer Systems, 10014 Rodney Parham Road, Little Rock, AR 72207. (501) 225-3908.

Ferguson 256K RAM Board for Xerox 820-11 BBI. Assembled, tested and burned in. Personally checked out by Jim Ferguson. Asking $220 OBO, Tom Coyle, 1848 Roseglen Avenue, San Pedro, CA 90731. (213) 616-5876 or (213) 831-0083.

Disk Service Manual, disk drive tutorial, printer & plotter manual, copier manual, computer phreaking!! Much more! FREE information. Consumertronics Co., Attn: Computers, 2011 Crescent, Alamogordo, NM 88310.

HP 3000 Users: Vitamin Kay is a terminal emulator which will run VPLUS applications using your Kaypro (any model). Supports all features of our MuNet product plus block mode and simplified upload/download. Only $100. Line drawing set EPROM is $25. MuNet customers send your master disk for full credit towards purchase of Vitamin Kay. IBM PC version available soon. Write for details. Attn: John Beckett, Computer Service Dept., Southern College, Box 370, Collegedale, TN 37315.

Simple Simple Simple—Here's a data file simple enough to be truly useful for general notes, receipts, daily reminders, schedules or almost any information you need at your finger tips. Bigger and better'n a 3 x 5 card. Full screen editing. No menus, no complicated formats. Just two words right from the system prompt tells the computer what you want, and there it is on the screen! Change it, re-arrange it or print it. Quick and easy. It's the simplest data file in town! Diskette and manual only $34.95 plus $2 S & H. Calif. add 6% tax. CP/M-80 only. Specify computer disk format. Many CP/M-80 formats available. VISA/MC OK. MYSOFT COMPUTER SOFTWARE, P. O. Box 417, Arroyo Grande, CA 93420. (805) 481-3992 or (805) 481-5687.

Data Analysis, Statistics, Matrix Algebra. Antana is a package of programs for statistical examination of data and for general matrix algebra. There are several modules that work together to manipulate and edit data and to perform univariate and multivariate analyses including multiple regression, polynomial regression, box plots, t-tests, r and q mode factor analysis, non-metric multidimensional scaling, cluster analysis, discriminant functions and others. Matrix programs include matrix divide, matrix invert, spectral decomposition of general real matrix, singular value decomposition. Several graphics utilities are provided with source in C and Pascal so that they can be customized for your terminal or printer. This package was written to be modular, simple, and to extract as much power as possible from microcomputers (e.g. non-metric scaling will handle 40 objects on a Kaypro II). We use it for both university teaching and for research. Available for CP/M-80 in several 5.25" formats and for MS-DOS, the price is $200, manual only $25 (refundable with purchase.) Dundee Software, 1080 Cypress Road, Bosque Farms, NM 87068. Phone 505-869-3595 evenings and weekends.

Xerox 820-I Stuff: completely populated working board purchased new from Xerox but never used or tested; new blank PC board, ROMs, schematics & data from B.G. Micro; new keyboard & case never used; Xerox AC cable plus drive cable for two 5.25"; Xerox software development manual plus Xerox troubleshooting manual for 820-I & II. Everything for $260 shipped. Steve Scriba, Box 399, FPO Seattle, 98761.

Xerox 820-1 with power supply, 12" monitor, cherry keyboard, Seimens FDD-200 115V 8" drive including power supply and diskettes. Comes with CP/M and documentation. No case. Duane Wilson (714) 796-1518. 24715 Lawton Avenue, Loma Linda, CA 92354.

Keyboards for Computer Builders—83 keys, full ASCII; upper/lower case, all control characters, numeric pad, CAPS-LOCK, REPEAT, self test! Brand new, hundreds sold already to builders of Apples, Big Boards, Xerox 820s. Parallel output, positive TTL logic, strobe. Uses only 106mA of +5 volts. Custom case available. 90 day warranty unmodified. Keyboard $35. Documentation (21 pgs.)/cable package $5. Spare custom CPU/ROM $4. UPS included. Call/SASE for detailed spec sheet. Electrovalue Industrial Inc., Box 376-MC, Morris Plains, NJ 07950. (201) 267-1117.

Xerox 820-II, Dual 8" Disk Drives. Diablo 630 Printer. Many Extras. $3700/Offer. (304) 725-6122.

SUPERB MAILING LIST PROGRAM stores and manages names and addresses that can be revised at any time. Its size is limited only by the disk storage available. The address labels may be code selected and printed in five different formats on your computer paper or on label rolls. In addition to the name and address fields there are four additional fields in each record for telephone number, date, and two amount fields if desired. At any time the entire roster may be printed out. For CP/M 2.2 based systems with two disk drives and printer capable of 132 columns for maximum usage. Terminal installation program module included. Supplied on 8" SSSD, 5.25" Kaypro and many others (please write). Special introductory offer by ABLE DATA SOFTWARE INC. PO Box 86923, Station C, North Vancouver, BC V7L 4P6. Only USA $19.95 postpaid check or money order.

Keyboards for Computer Builders 83 keys, full ASCII, upper/lower case, all control characters, numeric pad, CAPS-LOCK, Repeat, self test! Brand new, hundreds sold already to builders of Apples, BBs, Xerox 820s. Parallel output, positive TTL logic, strobe. Uses only 106ma of +5V. Custom case available. 90 day warranty unmodified. Keyboard $35. Documentation (21 pages)/cable package $5. Spare custom CPU/ROM $4. UPS included. Call/SASE for detailed spec sheet. Electrovalue Industrial, Inc., Box 376-MC, Morris Plains, NJ 07950. (201) 267-1117.

BBI, All Options, two 8" Siemans FD-100s, Hall Effect keyboard, 12" green screen "Le Monitor" monitor, several user disks. BBI main clock out. $200. 808-733-7855, 1263 Hudson Circle, Honolulu, HI 96819.

Information wanted on the MicroPro 'I/O Master' interface board or similar and hardware, connections, etc. to interface a Qume Q30 Daisy Wheel printer to the Big Board using WordStar. The printer uses an interface similar to Diablo Hy-Type 11 1300 Series and NEC Spinwriter 5500D (I believe) and uses a 24 bit parallel interface. Please write Ray Evans, P.O. Box 36, Iluka, N.S.W. 2460 Australia.

Bulletin Board Software for the Kaypro 2, 4, or 10. K-NET84 (tm) is a full featured message exchange system under modem control. Allows for both ASCII and binary file transfers between two computers over modem using XMODEM protocol. SECURE. Easy to install, use, and maintain. Comes with 50+ page user manual, telephone support from the author. For more information call or write: DATACOM SYSTEMS, INC., P. O. Box 115, Blue Ridge, GA 30513. (404) 632-2676. Kaypro is a (tm) of The Kaypro Corp.

Tandon Disk Drive Latch for Kaypro II, IV. Latch broken on your Tandon 100-1 or 100-2 5" disk drive? Replace the weak plastic latch pivot block with our improved aluminum piece machined from solid bar stock. Aluminum pivot block, 2 stainless steel pivot pins and instructions for only $15. Add 6% tax in California. Elmo Enterprises, 9955 Wild Grape Drive, San Diego, California 92131. (619) 271-1225.

Teeny-Weeny Basic—A 1K BASIC interpreter/editor. How much power can fit into 1K? A LOT! Full integer arithmetic functions with 26 variables, random function, single-dimension array, parenthesis nesting, string variable I/O, abbreviated commands, multiple statement lines, error handling. PRINT, INPUT, IF, GOTO, LET, RUN, LIST, SYSTEM. External LOAD and SAVE. Full documentation. Sample programs show TWB's power. Source available. Convinced? Try it! Just $17.50 postpaid (Texas add 5.25%). Specify 8" SSSD or Kaypro SSDD. Glenn McEowen, 3801 Glenmont Dr., Fort Worth, TX 76133.

Public Domain UG Software Rental: CP/M UG Vol 1-92 on 46 8" Flippies $45, SIG/M UG Vol 1-209 on 100 8" Flippies, $99.50, PICONET Vol 1-34 on 17 8" Flippies $25, Pascal-Z UG Vol 1-25 13 8" Flippies $25, UG Games 20 Vols of the best ones $25, UG Modem 20 Vols of the best $25, UG Business 20 Vols of the best $25, UG Utilities 10 Vols of the best $25. Rental is for 7 days after receipt with 3 more days grace for return. Credit cards accepted (preferred). 5" disk formats, 170 available. Downloading-disk format conversions. Call. User Group Software Automatic Update Service, $7.50 per 2 volume set PP.619-727-1015 24 hrs. 619-941-0925 info. 9-5. P.J.'s National Public Domain Software Center, 1533 Avohill, Vista, CA 92083.

HPLOT is a plotter emulation program that works with your dot-addressable printer to give quality graphic images without special hardware or programming. Its syntax is compatible with HPGL, the powerful graphics language used by Hewlett-Packard plotters. HPLOT provides full plotter emulation: plot absolute or relative, with user-defined scaling; create labels using characters of any size, slant, or direction; exercise control of windowing, line types and symbol mode. HPLOT is faster than comparable products because it uses pointer indirection rather than array references. No minimum memory requirement—it automatically uses disk buffers when necessary. Plots are saved on disk in minimum space, and may be used as the starting point for new images. HPLOT also boasts the ability to create images in sizes from 11 by 14 to 3.5 by 48 inches. You may use any language or text editor to create an input file for HPLOT. HPLOT is currently available on 8" SSSD for Z80-based CP/M 2.2 systems with Okidata printers; versions for Epson/Gemini-compatible printers are under development, as is 5.25" distribution (please write). $49.95 postpaid check or money order. Ordinate Solutions, 614 Beech St., Oberlin, OH 44074.

Ready To Use Files

By Phil Emery

Micro C
Filing Clerk

Bobcat3 Disk Catalog Program. Finds lost computer files. Allows individual file titles, hard disks, user numbers/sub-directories. Wildcard finds. Seven report formats. 8" SSSD, popular 5 1/4 CP/M and PC/MS-DOS. $49.95 US$ plus $3.00 P&H. Visa, Mastercard, or check. R&L MicroServices Inc., Box 15955, Stn F, Ottawa, ON, K2C 3S8 (613) 225-7904.

CBASIC Users Group—newsletter with membership, $18/yr (US & Canada), $28 other. For info send SASE to Babel/CBNews c/o Ric Allan, PO Box 40690, Cincinnati, Ohio 45240-0690. Sample and back issues available for $2.

8" Drive Cleaning Kits—12 cleaning disks and carrier jacket made by Datalife, regular price, $29.95. Close out price just $6.00 each plus $2.00 postage. Limit 2. P.D. Software, 1533 Avohill, Vista, CA 92083.

Big Board II, power supply, 2 8" Shugart disk drives in heavy duty steel case. 15" monitor and large W/P keyboard. $800. Call John Wytsma (207) 725-4209.

Used Heavy Duty Shugart 800-2 single drives in aluminum cabinets with power one supply, lighted switch, surge protection. BB-I will fit case. $75. plus shipping. (4) Recycled SSSD 8" disks $.50 each verified. Minimum 20. Mail orders only. Write D. Ayres, 308 Maple Street, Ypsilanti, MI 48197. Trades possible.

Keytronic keyboards. New. ASCII encoded $10 each. Two for $18. 300/1200 autodial autoanswer Hayes compatible smartmodem. $199. B.W. Systems, Box 9791, Austin, TX 78766. (512) 255-8350.

■ ■ ■

There are many products now on the market which claim to handle files, or to even create file handlers.

The packages which create file handlers ask you to set up the entry screen and the output forms, and then they create the programs with which you enter and access information.

Other packages, like dCEASEd II and OffBASE, let you write the code yourself, from scratch if you wish. These packages require more work, but they are much more flexible in the hands of an expert, and the routines create the programs automatically.

No matter which way you create the file handler, though, there is still a problem. You have no file. When you call to tell them that they forgot to include the file with your file handler, they kind of snicker understandingly (so you know who they'll be talking about over coffee).

Actually, it's they who have a problem. The sudden demand for file handlers like dCEASEd II has created a tremendous shortage of files—files of all types.

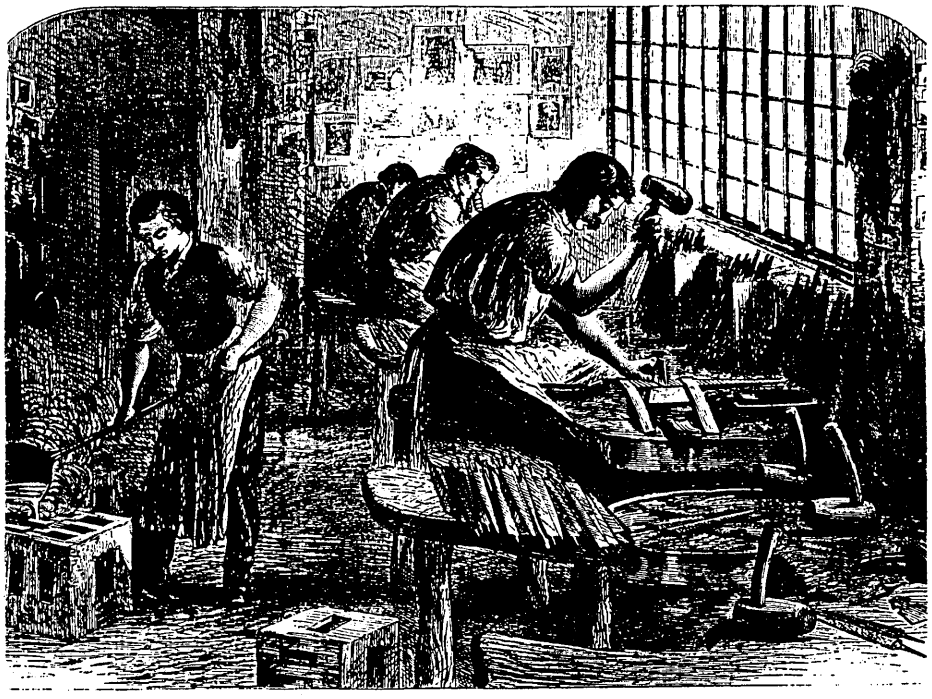
In the early days, there were still enough files around for people to pretty much have their choice. Some of the more popular included: flat files, pillar files, square files, cant files (as opposed to won't files), and equaling files.

Unfortunately, all files weren't created equal, and soon companies found themselves going through the round files looking for something that might satisfy an abrasive customer.

The Cutting Edge Of Technology

Finally, though, one company, Fashion Pate, recognized the problem and set up a major manufacturing facility. Fashion Pate was a natural to take on this project because it was well known for its incredibly hard software and its rigid customer support.

■ ■ ■



Employees hammer out files in Fashion Pate's new production site. An example of ever changing technology, this facility is a converted IC design lab. Japanese businessmen are already converting their IC facilities in order to keep up with the U.S.

The Culture Corner

Confusing at Catastrophy Manor

Alas, it has been yet another sleepless month here at Castastrophy Manor. My faithful Z80, Beulah, began spitting out 'not ready' errors two weeks ago and my in-house technician didn't discover until yesterday that I had not put a disk in the drive.

During that time I was forced to use Zimblefield J. Rothschild, the Cray 1 that normally monitors the odor level from the kitty box. What a frustrating machine! I certainly wish someone would enlighten the Cray people on how to design a proper keyboard. The left shift key is positioned at least a sixteenth of an inch from the standard position on the Selectric! I call their customer service department to complain but it doesn't do any good. This is the fifth Cray they've given me, and they still haven't gotten the left shift right.

Worthless Software

Now that my latest novel, 'Stumblefeet,' is completed, I have refocused my attention on eating, and insulting anyone who markets a product I have not already recommended in my column, along with those who program in anything other than Pascal or Modula-2.

This month I received a program for the IBM PC that guarantees to find enough tax loopholes to reduce your income tax to nothing. Every year. How ludicrous. I have never seen that in my column — I even went back and looked (plus, I paid taxes last year, so I must not have reviewed it.) To make things even more ridiculous, it is written in C. And they expect me to open the box!? Come now fellows, how much effort do you expect me to make?

High Technology

I have been saying for years that the only way for a computer company to be successful is to design a Timex-Sinclair compatible S-100 board. I must have said that to at least 50 people. Nobody listened. But finally, Say Co. Computers came out with exactly what I have been waiting for, and it is truly a tribute to high technology. It can add 200 numbers (some of them large) in under a second, it has a real time display, and it is water resistant.

I am considering using one here at Catastrophy Manor to replace the Crays if something isn't done about that appalling keyboard.

Free Poursmelle Software

A while back I was thumbing through the truckloads of mail, free copies of Burpo Pascal, and free Honkubro hardware that all us famous overweight computer columnists get, when I found a request for another incredible Modula-2 Star Trek game. Since I don't want to write another Star Trek game, and I don't really pay any attention to my mail anyway, I decided to write a Pascal-to-Lisp translator.

I have been laboring over it for several months, and it is finally done. I was planning on selling it for $99.95 through The Softhead Foolworks, but due to a momentary affliction of divine benevolence, I have decided to publish it here in the hallowed pages of OVERBYTE.

This translator avoids all the usual problems of converting infix to prefix notation and of moving from the domain of a sequential language to that of a procedural language. In fact, my translator is very unusual because its output precisely mimics the original Pascal (the process is known as LISP Sync).

My son Smartalex doesn't think that anyone WANTS to convert Pascal to Lisp (but then he thinks that the 68000 is more powerful than the 6502).

By Verry Poursmelle
(As compiled By Laine Stump)

Tidbits ("Choice morsels, as of food or gossip")

By Gary Entsminger

Micro C Staff
Future Tense Editor

Part of the daily routine at Micro C is sifting through a lot of press releases. By reading between the lines we get hints of what's going on in the computer industry. Sometimes what's going on is pretty funny, and, obviously, we don't believe everything we read (ALL those products just can't be the latest and greatest!), but we do from time to time encounter information we think you might find interesting.

You're already getting some of that info via Dave's editorial, the Technical Tips, letters to the editor, and reviews. So "Tidbits" will be another way we pass information along to you. As usual, please let us know what you're doing, and how we're doing.

Geneva PX-8 Gets Turbo

Borland International and Epson Corporation have announced the signing of a joint marketing agreement giving Epson the right to distribute all Epson-compatible Borland software products.

This means Borland's Turbo Pascal Series will be distributed and supported by Epson for their Geneva PX-8 portable and their QX-16 (IBM PC-compatible) microcomputers. The Turbo Series includes Turbo Pascal, Turbo Toolbox, Turbo Tutor, and Turbo Graphix Toolbox. Epson also plans to support and distribute other Borland software products this year.

Imagine: a lap computer that provides a Turbo Pascal development environment. This could be a giant step for the little guys.

Kaypro And Xerox Surplus

Some good buys for your Kaypro from Sabet Electronics:

9" Green Monitors	$45.00
Keyboards	$45.00
Wiring Harness	$4.00
81 Series Roms	$10.00
K2 Motherboard, w/sockets	$27.50
K10 Motherboard, w/sockets	$37.50

Sabet also has a good deal on a Xerox 820-I single board computer. Z80 CPU, 64K RAM, 80 x 24 video display, floppy disk cont., I/O ports, runs CP/M* 2.2! Fully tested ..... $65.

Sabet Electronics
13650 Floyd Rd. Ste. 104
Dallas TX 75243
214-783-4950

More Xerox

Xerox Manufacturing Outlet is advertising 820 boards (used as-is) for $50, and low profile keyboards for the 820-II for $25.

These sound like good deals, but we haven't been able to reach the order department by phone. Bruce spent an afternoon dialing and listening to a busy signal. Hmmm.

If you get through, let us know.

214-960-3367
1301 Ridgeview Dr. MS 503
Lewisville TX 75067

Z System (ZCPR3 + ZRDOS)

Echelon, Inc. has announced high performance 8-bit sets of software called the Z System. Z is operating-system downward compatible with CP/M-80, and works with existing Zilog Z80 and planned Z800 microprocessors. Z comes in three versions. The Z800 version permits easy 16-bit MS-DOS program migration to 8-bit Z.

Z was produced by combining ZCPR3 and ZRDOS, and replaces CP/M while adding several significant enhancements: auto-login of changed disks, file copy archiving, password write-protected files, and more.

Retail prices start at $39, but generous discounts are available for volume users, original equipment manufacturers, and value-added resellers.

For more info contact:

Echelon, Inc.
101 First Street
Los Altos CA 94022
415-948-3820

MSX

It looks like the Japanese MSX is about to descend on the American market. Over a dozen MSX computers are now on the market with prices generally between $200 and $400. Most models have built-in radio frequency modulators and composite video output, which allows the use of an ordinary television set for display. And MSX-standard computers share common interfaces. Programs and hardware designed for one MSX should be compatible with all MSX.

MSX has been big for a while in Japan (half the computers there are MSX) where it's been primarily used to generate sound effects, and for BASIC programming and game-playing.

If you're interested in exploring this new arena, you might check out the Yamaha YIS503, with 2 slots and 32K bytes of RAM for around $270, or the Casio stripped-down model which looks like a keyboard, plugs into a television set, and retails for around $80.

Canon, Sanyo, Pioneer, Hitachi, Fujitsu, and Mitsubishi also sell MSX computers. Looks like everybody's getting into the act.

Happy playing.

ACNAP2 Electronic Circuit Analysis Program

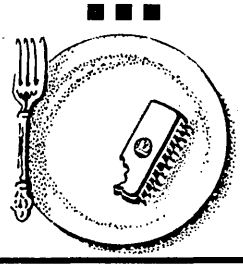
BV Engineering has upgraded its popular ACNAP1 program which is now available for 121 different computer systems running CP/M-80 and MS-DOS.

ACNAP Ver. 2 is a general purpose AC Network Analysis Program which analyzes active and passive electronic circuits consisting of resistors, capacitors, inductors, transistors, and operational amplifiers. Circuits up to 200 components and 30 nodes may be quickly analyzed in a single pass and larger circuits in multiple passes.

ACNAP contains a built-in full feature circuit editor which supports addition, deletion, and changes of components, tolerances, and node connections. Cost is $72.95.

For more info contact:
BV Engineering
2200 Business Way, Suite 207
Riverside CA 92519
714-781-0252

■ ■ ■



The Epson PX-8, CP/M In Your Lap

Review By Gary Entsminger

Micro C Staff

Epson's PX-8 Geneva is the most recent addition to the "Lap Computer" market. Although more expensive than Radio Shack's and NEC's portables, it does more, and runs CP/M.

The $995 retail price includes computer, standard keyboard, microcassette, and display all mounted in an 8 by 11 inch durable-looking plastic box. It weighs 5 pounds, has 64K RAM, and four interchangeable 32K ROMs. The ROMs contain CP/M, MBASIC, WordStar, CalcStar, a scheduler program, and a communications program.

Unfortunately, a modem isn't included. It is, however, available as an option, as are the 3½-inch disk drive and 60K or 120K memory expansion.

The PX-8 runs CP/M 2.2 on a low-power CMOS Z80 with a clock rate of 2.45 MHz. It's equipped with 64K of main memory which is always on, even when the PX-8 is turned off. A maximum of 24K (of the 64K) can be set up as a RAM disk.

CP/M resides in 32K of ROM. It's bank-switched over the lower 32K of RAM when the PX-8 is turned on. Other ROM based software like WordStar and BASIC get loaded (slowly) into lower RAM after CP/M is switched out.

Most simple CP/M utilities such as DDT and UNERA will run on the PX-8, and theoretically, so will any CP/M software compatible with a Soroc IQ-120 terminal (the PX-8's). (Borland and Epson have recently announced that they will jointly distribute the Turbo Series—Turbo Pascal, Toolbox, Tutor, and Graphix Toolbox.)

Epson says that larger and more complex CP/M programs will run, but with minor problems. (One of the problems is the limitation on program area when the RAM disk is being used.)

The liquid crystal display (LCD) features 5 by 7 dot matrix characters in an 8 line by 80 character format.

The PX-8 is powered by a nickel-cadmium battery which can run about 15 hours on a full charge. The battery recharges whenever the system is plugged into the wall transformer.

The system is also equipped with a real-time clock which can be used to turn the system on and run programs .

Two serial I/O ports are located on the back (one for modem, the other for printer).

WordStar

Portable WordStar is a subset of WordStar, and can edit about 10 pages at a time, which makes the PX-8 an adequate portable word processing machine.

Although add-on memory would permit a complete WordStar (some features like hyphen-help and paragraph tabs are omitted in Portable), Micropro chose to go with a subset to insure portability. Portable WordStar works on other lap computers as well as the PX-8.

This version of WordStar uses the same commands as regular WordStar. It offers an opening menu, and performs more or less like its big sister. Since Micropro correctly assumed that anyone using a lap computer would likely be transferring files to bigger machines, it added a transmit command to WS's opening menu.

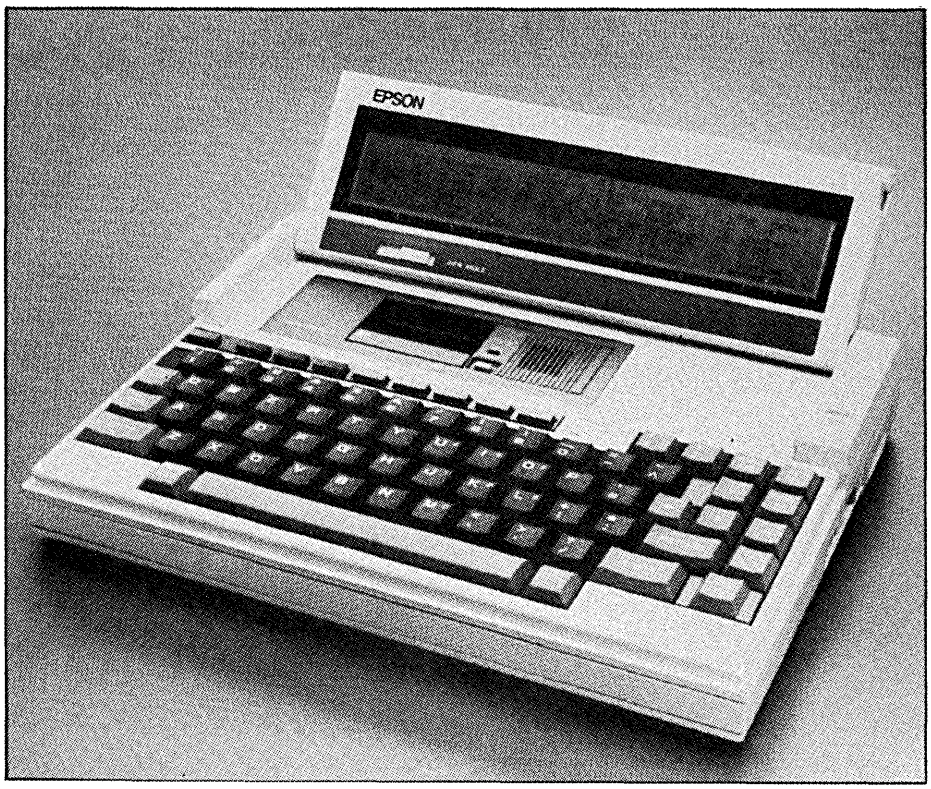
Unfortunately, there are a few inconveniences with Portable WordStar, but they can usually be circumvented.

Files can't be saved directly to microcassette, but can be copied to microcassette using WS's Copy command. A file stored on microcassette can't be opened by WS until it's copied into RAM.

CP/M

Only a subset of CP/M was included in the CP/M ROM (PIP, STAT, SUBMIT, and XSUB). Some familiar utilities like DDT and ASM are not included. However, they can be loaded in from disk. The optional disk drive package includes FORMAT, DISKCOPY, ED, DDT, ASM, LOAD, and DUMP.

In order to test CP/M system calls I wrote a little program to convert decimals into hexidecimals using 8080 mnemonics. Then, using a DDT-type utility called hexdump, I dumped the hex code to screen. I used the hex code as data, and poked it into a BASIC program on the PX-8 which called the machine lan-



Epson PX-8 Geneva: 8-line by 80-character display, built-in microcassette, WordStar, and CP/M.

guage subroutine. The program used four CP/M system calls (Console Input, Console Output, List Out Printer, and Read Console Buffer), and worked without a hitch. (Editor's note: There wouldn't have been room for a hitch if he'd needed one.)

Papers

The documentation is plentiful, but not well organized. For example, instructions indicate the RAM disk can be configured from 0 to 24K. However, if it is too large, some application programs, like BASIC, won't run. But in order to find out why BASIC suddenly doesn't work, you have to look in an entirely different part of the manual. In general, though, if you search long enough, you'll find the information you need; it just takes patience.

Final Details

The LCD is a minus. It can be adjusted, but the angle of the display is critical. If you want to read the characters on this screen you can't squirm.

Loading programs from ROM is slow, and writing to or reading from microcassette is slower. But once you get your programs into the RAM disk (assuming they fit), things pick up nicely. You might also purchase the optional disk drive so you could get around the problems of the RAM disk size and tape speed altogether.

Finally

I can recommend the PX-8 for someone needing truly portable word processing, BASIC, and CP/M utilities. The PX-8 Geneva would be handy for a writer (like myself) who has access to a larger computer, but needs to work while traveling, and can make do with 10 or so pages. Ten pages, after all, isn't that small a chunk.

The big selling point for this machine, though, is CP/M, which means that this little system can run much of the software that you are currently running on your full size CP/M system. While the rest of the herd strives for IBM PC compatibility, this lap computer has achieved both portability and CP/M compatibility. And though the basic system has some limitations, the tradeoffs seem reasonable.

■ ■ ■

BACK ISSUES

$3.00 each
US, CAN, MEX

$5.00 each
Other Foreign

ISSUE NO. 1 (8/81)
Power Supply
RAM Protection
Video Wiggle
½ PFM. PRN
16 pages

ISSUE NO. 2 (10/81)
Parallel Print Driver
Drive Motor Control
Shugart Jumpers
Program Storage Above PFM
½ PFM.PRN
16 pages

ISSUE NO. 3 (12/81)
4 MHz Mods
Configuring Modem 7
Safer Formatter
Reverse Video Cursor
FORTHwords begins
16 pages

ISSUE NO. 4 (2/82)
Keyboard Translation
More 4 MHz Mods
Modems, Lync, and SIOs
Undoing CP/M ERASE
Keyboard Encoder
20 pages

ISSUE NO. 5 (4/82)
Word Processing
Two Great Spells
Two Text Editors
Double Density Review
Scribble, A Formatter
20 pages

ISSUE NO. 6 (6/82)
BB I EPROM Programmer
Customize Your Chars
Double Density Update
Self-Loading ROM
Terminal in FORTH
24 pages

ISSUE NO. 7 (8/82)
6 Reviews of C
Adding 6K of RAM
Viewing 50 Hz
On Your Own begins
24 pages

ISSUE NO. 8 (10/82)
Drive Maintenance
Interfacing Drives
Installing a New BIOS
Flippy Floppies
C'ing Clearly begins
Xerox 820 begins
28 pages

ISSUE NO. 9 (12/82)
BB II EPROM Program
Relocating Your CP/M
Serial Print Driver
Big Board I Fixes
Bringing Up WordStar
Cheap RAM Disk
32 pages

ISSUE NO. 10 (2/83)
Saving a Flaky Disk
Hooking Wini to BB II
The Disk Inspector
JRT Fix
Serial Keyboard Interface
Pascal Procedures begins
36 pages

ISSUE NO. 11 (4/83)
BB I Expansions
BB II Details
Dyna, RAM Disk Review
Easier Reverse Video Cursor
PlannerCalc Review
KayPro Column begins
36 pages

ISSUE NO. 12 (6/83)
256K for BB I
Bringing Up BB II
dBase II
Look at WordStar
Double Sided Drives for BB I
Packet Radio
5MHz Mod for KayPro
40 pages

ISSUE NO. 13 (8/83)
CP/M Disk Directory
More 256K for BB I
Mini Front Panel
Cheap Fast Modem
Nevada Cobol Review
BB I Printer Interface
KayPro Reverse Video Mod
44 pages

ISSUE NO. 14 (10/83)
BB II Installation
The Perfect Terminal
Interface to Electronic Typewriter
BB I Video Size
Video Jitter Fix
Slicer Column starts
KayPro Color Graphics Review
48 pages

ISSUE NO. 15 (12/83)
Screen Dump Listing
Fixing Serial Ports
Playing Adventure
SBASIC Column Begins
Upgrading KayPro II to 4
Upgrading KayPro 4 to 8
48 pages

ISSUE NO. 16 (2/84)
Xerox 820 Column Restarts
BB I Double Density
BB II 5"/8" Interface Fix
KayPro ZCPR Patch
Adding Joystick to Color Graphics
Recovering Text from Memory
52 pages

ISSUE NO. 17 (4/84)
Voice Synthesizer
820 RAM Disk
Kaypro Morse Code Interface
68000-Based System Review
Inside CP/M 86
56 pages

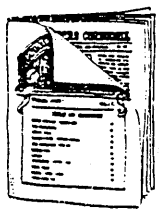
ISSUE NO. 18 (6/84)
Kaypro EPROM Programmer
I/O Byte: A Primer
Kaypro Joystick
Serial to Parallel Interface
Business COBOL
60 pages

ISSUE NO. 19 (8/84)
Adding Winchester to BBII
6MHz on the BBI
Bulletin Boards, Getting It Together
Track Buffering on the Slicer
4MHz for the 820-I
64 pages

ISSUE NO. 20 (10/84)
HSC 68000 Co-Processor
DynaDisk for the BBII
Serial Printer on BBI Sans SIO
Cheap & Dirty Talker for Kaypro
Extended 8" Single Density
72 pages

ISSUE NO. 21 (12/84)
Analog to Digital Interface
Communication Between High-Level and Assembly Language
Installing Turbo Pascal
Low Intensity BBI Video
Turbo Pascal, The Early Days
80 pages

ISSUE NO. 22 (2/85)
Converting a Xerox 820-II to a Kaypro-8
S-100 Expansion for Single Board Z80 Systems
Sound Generator for the STD BUS
Reviews of 256K RAM Expansion Boards
In the Public Domain
88 pages



ADVERTISER'S INDEX

ORDER FORM

Micro C works because it is a central information exchange for the doers in this crazy industry. So we encourage you to share your trials and tribulations. That way we can invent new wheels rather than redoing the old ones over and over.

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

What information would you like to see in Micro C?

Quantity	Description	Price Each			Total
		U.S.	Can & Mex	Other Foreign	
	SUBSCRIPTION (1 year—6 issues) ☐ New ☐ Renewal	☐ $16	☐ $22 (Air Mail)	☐ $30 (Air Mail)	
	USERS DISKS – 8" SSSD, CP/M #'s	$15	$15	$15	
	CP/M 86 DISKS – 8" SSSD #'s	$15	$15	$15	
	KAYPRO DISKS for the II, 4, & 10 – 5" SSDD #'s	$12	$12	$12	
	KAYPRO SCHEMATIC PACKAGES ☐ II & 4 (pre-84) ☐ 2, 4, 10 (84) ☐ 10 (No Modem)	$20	$20	$20	
	FREE KAYPRO CATALOG				
	BACK ISSUES #'s	$3	$3	$5	
	T-SHIRT (specify S, M, L, XL)	$6.95	$8.50	$8.50	
	OTHER ITEMS:				

Prices include media, package, 1st Class postage (Air Mail for Other Foreign)

☐ Check or money order enclosed
(US funds only, payable on a US bank)

Make checks payable to:
MICRO CORNUCOPIA

TOTAL ENCLOSED

Card No. _____ - _____ - _____ - _____ **Exp.** _____ _____
☐ Visa ☐ MasterCard Signature

Are you a current subscriber to Micro Cornucopia? ☐ Yes ☐ No

NAME _____ **PHONE (?)** _____

ADDRESS _____

CITY _____ **STATE** _____ **ZIP** _____

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709

Order Number (503) 382-5060 • Technical Calls 9 - Noon PST Only (503) 382-8048

The Last Page

By Gary Entsminger

Micro C Staff

Welcome to The Last Page, and hold on to your hats. This issue I'll introduce the format. Next, I hope you'll do most of the talking.

In response to the large number of letters we've gotten requesting more info on other systems, we've added a new section to Micro C: Future Tense, a cross between classic Micro C and Star Wars. In other words, we're going to talk about some non-Z80 stuff. Think of it as a vice, if it makes you more comfortable. Or think of it as being practical. Whatever works.

This issue, Future Tense introduces "Tidbits", and a review of Epson's new portable CP/M computer, the PX-8 Geneva. In the future, we'll feature your articles, so send us your best.

We'll also talk about where Z80 is headed. MSX looks really exciting; Dave's got one working, and we can't wait to see what he comes up with. (Although right now he says it feels like an old TRS 80 Model 1, which means it isn't going to threaten anyone in this form.)

MSX-DOS is file compatible with MS-DOS V1.X, but Microsoft claims it 'provides an environment' that permits 'nearly all' CP/M programs to run directly. Interesting.

I want to know what you think about other systems—the PC system bus, for example. Unix. CP/M 68K. Gem. I'm guessing there's a growing number of hobbyists out there just itching, or already itched, to get their hands on something that has some new bells and whistles.

16 Bits

There are definitely some interesting new products on the horizon. Jack Tramiel's 'Jackintosh' from Atari, a 68000-based Macintosh lookalike, is scheduled for sale **real soon**, retailing at $399 with 128K of memory. That price doesn't include monitor or disk drives, but even so, it looks like an inexpensive ride into something that's really powerful.

We've already started looking into the Kaypro 16 (See Issue #22), and we'll be letting you know what we find in that arena.

Public Domain

The public domain hobbyists are certainly out there and active. This month I received a newsletter from the East Carolina CP/M MS-DOS users group. They have two public domain libraries: one for CP/M, one for MS. We're going to see more of this combination, and we'll be looking for good MS-DOS public domain software here at Micro C, so send us your favorites.

Hardware

We've come a wild road from the 8008 processor, and the industry's changing very fast. What's next for the hardware guys?

What would it cost to put together a small computer system running a 68020 or a 32032? Has anyone started doing this? Or modifying? Are parts too expensive? Is it the cost of a development system? If there's a bottleneck, what is it? I'd like to hear from you. Novices, don't be shy.

Books On Assembly Language

If anyone's interested in 8088 Assembler, I recommend '8088 Assembler' by David C. Willen and Jeffrey I. Krantz. It's a Sams book.

Sams also did 'Soul of CP/M' by Waite and Lafore—an excellent introduction to 8080 assembler. If you're still having trouble figuring out things like how to use DDT and CP/M system calls, then this might be the one for you. Good cartoons, as well.

Return To The Sour Grapes Of Wrath

What's become of the Zilog family? Is there any hope of a Z800, or something that would be upward compatible with the Z80? I've always liked the idea of a family of chips (you know papa chip, mama chip and little bitty baby chip).

Applications

So now that you've got your system fine tuned, how are you using it?

I've been hearing a lot about neat laboratory applications—things like experiment controlling and data acquisition through the digital conversion of analog electrical signals.

Although a 16-bit bus transfers data twice as fast as an 8-bit bus, the 8-bit bus can still move data right along. In fact, the 8-bit bus is very popular in scientific applications. Software appears to be the deciding factor here.

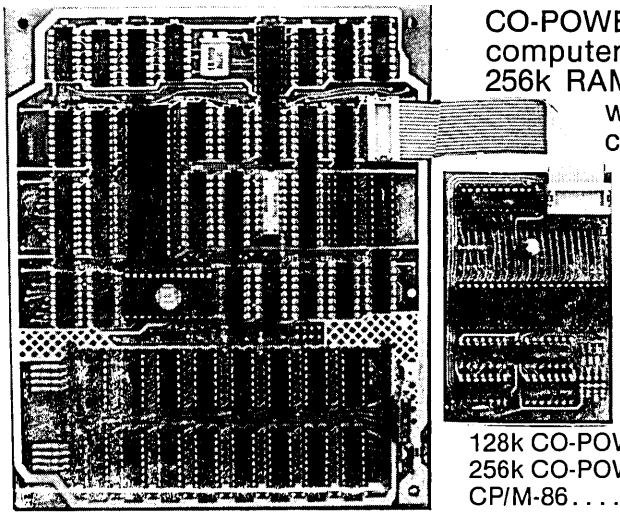
Wrap Up

And that's about it for the first 'Last Page.' I'll be here each issue, talking about systems, in its broadest definition. If enough of you are interested we'll add more pages to Micro C or expand through the back cover.

I won't even turn down a good (almost) sci-fi invention, or a good sci-fi novel recommendation. But for now I'd be happy with a good public domain CP/M MS-DOS file transfer program. Any offers?

■ ■ ■

SWP's CO-POWER-88 makes Z80, CP/M microcomputers IBM-PC compatible!



CO-POWER-88 is a 16-bit 8088 coprocessor for Z80 CP/M computers. Both versions of CO-POWER-88, 128k and 256k RAM, include both MSDOS, and RAM drive software, complete with MSDOS, IBM-PC compatibility.

Simple commands move system control between the Z80 and 8088 processor. CO-POWER-88's RAM can be used in CP/M as a RAM drive! Currently available for Kaypro, Bigboard, Zorba, Xerox 820-II, Actrix, Osborne, and ATR8000 computers.

128k CO-POWER-88 w/MSDOS & RAM Drive..... $400.00
256k CO-POWER-88 w/MSDOS & RAM Drive..... $500.00
CP/M-86..... $70.00

ATR8000: SWP's $499.95 CP/M Computer

SWP's ATR8000 is a 64k RAM,Z80A, 4 MHz computer that includes double density CP/M 2.2. The ATR-8000 runs up to four disk drives that are any mixture of size (5¼" and 8"), type (single-sided and double-sided), and density (single, double and quad). The ATR8000 has an RS-232 port for a modem or serial printer and includes software for both. There's also a parallel port with a parallel printer driver. The ATR8000 interfaces to an RS-232 terminal or to an ATARI home computer. Software includes a program that allows the ATR8000 to use CP/M disks from other computers. The ATR8000 can be upgraded to also run CP/M-86 and MSDOS by adding CO-POWER-88.

Bigboard Dual Density

We've just released a new version!

Hardware

- **A daughter board that plugs into the 1771 socket. With this board the system employs automatic density select.**
- **Instructions tell how to run 5¼" drives. A 50-34 pin disk drive adapter board is included with 5¼" disk orders.**

Software V#061983

- **One 8" version includes the code to make a 60k double density CP/M for:**
  8" SS 2.5 MHz 8" DS 2.5 MHz
  8" SS 4 MHz 8" DS 4 MHz
  5¼" SS 2.5 MHz
- **Printer drivers are built-in, selectable in the IOBYTE.**
- **Easy to change port parameters.**
- **8" SSDD disk storage is 674k; DS is twice as much! 5¼" is 183k.**
- **Includes DDINIT for SD and DD initializing and DDSYSGEN for DD sysgening.**
- **Special features have been added including a deluxe pause, screen print and clock.**
- **Source code is available for $25 after you sign a disclosure agreement.**

For Orders: We accept MasterCard, Visa, Money Orders or checks. Shipping charges and applicable taxes will be added. Call or write for delivery time. Prices and specifications subject to change without notice.

SWP
MICROCOMPUTER PRODUCTS, INC.

2500 E Randol Mill Rd. - 125
Arlington, Texas 76011
817/469-1181
817/861-0421

MICRO CORNUCOPIA
The Micro Technical Journal

P.O. BOX 223
BEND, OREGON 97709

B expires after issue 25
EDWARD ELIZONDO
6 CYPRESS COURT
EAST WINDSOR NJ 08520

ADDRESS CORRECTION REQUESTED

BULK RATE
U.S. POSTAGE
PAID
Forest Grove, OR
Permit No. 36