MICRO CORNUCOPIA
P.O. Box 223
Bend, Oregon 97709
503-382-8048

Editor & Publisher
David J. Thompson

Graphic Design
Sandra Thompson

Technical Staff
Dana Cotant

Advertising Director
Alice Holbrow

Staff Assistant
Dorcas Dsenis

Typography
Patti Morris & Martin White
Irish Setter

MICRO CORNUCOPIA is the single board systems journal supporting systems programming languages and single board systems—including the Big Board, Big Board II, Xerox 820, Kaypro, and Slicer.

MICRO CORNUCOPIA is published six times a year by Micro Cornucopia of Oregon, **P.O. Box 223, Bend, Oregon 97709.**

SUBSCRIPTION RATES:

1 yr.(6 issues)	$16.00
1 yr.(first class)	$20.00
1 yr.(Canada & Mexico)	$20.00
1 yr.(other foreign)	$26.00

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

ADVERTISING RATES: Available on request.

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

SOFTWARE, HARDWARE, AND BOOK VENDORS: We would very much like to review your CP/M & MSDOS compatible products. Send materials to the Review Department, Micro Cornucopia.

MICRO CORNUCOPIA

December 1983 The Single Board Systems Journal No. 15

Son of a Big Board!

Genealogy

The following is for all you computer genealogy buffs—that is, of course, those of you who are interested in the genealogy of your computer.

In the beginning there was the Big Board (BB I). The hardware was designed by Jim Ferguson, the software by Russell Smith (often referred to as the Ferguson and Smith gang). The BB I was announced in the summer of 1980 at a kit price of $649. Jim Tanner of Digital Research Computers of Texas—not to be confused with Digital Research of California—handled marketing, sales, and support.

The initial reaction by the S-100 buffs was that the BB I couldn't be a serious product. After all, what if you wanted to add another port or some more RAM? (And besides, S-100 systems even had their own magazine called S-100 Microsystems.)

Xerox 820

However, during the Fall of 1980 (as opposed to the rise of 1980) Xerox purchased rights to manufacture the Big Board and put it inside their 820. Xerox knew that IBM was working on a micro so Xerox wanted to get their own system onto the market VERY quickly.

The original 820 sold very well the first few months as independent software and hardware designers bought the new system and checked it out for potential new products. The 820 definitely needed some help. It didn't include much software, it had single-sided single-density 5" drives (unless you paid a gob extra for single-density 8"), and it ran at 2.5 MHz.

Information about the system was nonexistent. Xerox engineers had a very bad case of "not invented here" so they acted as though the 820 didn't exist. Because the 820 was a very close copy of the BB I, some of us knew where to get information (many Xerox dealers had to service the 820 using only the $5.00 BB I documentation package from DRC of Texas.)

I contacted Xerox marketing and engineering about supporting technical folks via Micro C. The answer from marketing was a resounding silence. Engineering wasn't so subtle. The engineering manager told me that I was not to speak to any of his engineers. (And then, what hurt most of all, he told his engineers that they weren't to speak to me.) Xerox was very concerned that someone might discover the genealogy of the 820.

Xerox had hired Jim Ferguson to help them with system configuration details but the company refused to listen when Jim suggested that they go to 4 MHz and double density. Much later, the 820-II came out with 4 MHz and double density. Surprise!

Xerox had high hopes for the 820 so they manufactured a whole scad of boards and shipped off thousands of systems to their official distributor, Hamilton Avnet. The rumor has it that two years later, Hamilton Avnet was still sitting on over 10,000 820-1s. (Not exactly the cushiest thing to sit on.)

Anyway, the independent software and hardware folks got tired of trying to pry information out of Xerox (and sales of the 820 were struggling, partially because of lack of independent software and hardware) so they looked around for another system to support.

Kaypro

Meanwhile, Non Linear Systems, a small manufacturer of digital meters, purchased two Big Boards. Lo and behold there soon appeared the KayComp computer. They used that name until the original owner of "KayComp" found out about Non Linear. So, the name became Kaypro II.

The Kaypro II is a Big Board with a few modifications (actually more modifications than Xerox made). Non Linear used eight 64K bit RAM chips instead of

(continued on page 42)

LETTERS

Dear Editor,

I am writing you to let you know that I highly recommend Cascade Electronics to Micro C readers. Dave and Kathy Garner provide excellent service.

A couple of us got six of their Shugart SA800-2 eight inch disk drives at about $100 each. They arrived promptly and were in excellent shape. In fact, they were immaculate. We have been running some of the drives 8 to 10 hours a day since April and they have been quite dependable.

Anyone needing disk drives, blank disks, power supplies, or enclosures should get a hold of them. Dave and Kathy sell quality merchandise and are really nice people to deal with.

Contact them at Cascade Electronics, Rt. 1 Box 8, Randolph, Mn. 55065, 507-645-7997.

Alan Gomes
14612 San Bruno Drive #3
La Mirada, CA 90638

Editor's note:

Amen! I got some of their two-drive cabinets with power supplies as well as their 8" drives. Everything I've received has been as good or better than advertised and service has been excellent.

Also, thanks, Alan for the letter. It's not very often that someone takes the time to write about good service, usually I hear from people after they've been royally (or unroyally) bushwhacked.

Dear Editor,

I would like to suggest an extension to William King's article on page 4 of issue #14. Readers should do two patches to the BB II monitor.

At address F324H, change the 60H to 68H. At address F32AH change the 40H to 48H.

These patches set the H bit in the STEPIN & STEPOUT commands, so that the heads remain loaded during seeks. Otherwise the head will chatter during some seeks.

Alex Cameron
14A Somers Ave
Malvern, Victoria 3144 Australia

Dear Editor,

I have had some success doing word processing for fellow students here in Bellingham, Washington. I charge $1.50 per page and give my customers a rough draft to edit. It seems expensive to many people but they can't beat the quality. My Kaypro and TP-1 printer really do a good job. I'm looking for a Gemini-10 printer for the rough drafts.

KKey is not all it's cracked up to be. The commands to alter CP/M don't always work. If you set it to load a program on a cold boot (but not on warm boots) it will load the program on all boots, so you can't get back to CP/M. Also, it sometimes sends my system out to Sunday lunch with no warning and it doesn't do keypad redefinitions. I have heard from another group that KKey is good but I've had nothing but problems with it.

Norman Bruland
1409.5 Harris Ave
Bellingham, WA 98225

Dear Editor,

One of your readers said that "everything else gets second priority when I find Micro C in my mailbox." I can understand that. I grabbed your pub as I walked in from work one evening, then sat down and started reading. My wife joked, "I can handle your ignoring me now but if you bring that magazine to bed . . . " I accepted the hint and removed the copies of Micro C from my nightstand.

After all, one must draw the line when it affects one's marriage. I put them on her side.

On a less significant note, I'm still having problems making my Shugart 851 double sided 8" drive look like two drives to the Big Board. You and DRC's folks make it sound very simple so I must be missing something.

By the way, the utility on User's disk #11 for configuring the Okidata 92 works great. It supports almost all the optional output formats and is simple to use. If folks are interested, I would be willing to write a utility to download graphics from a Heath terminal to the uL92.

Bernie Lannan
536 "E" St
Robins Air Force Base, GA

Editor's note:

I don't know whether to offer my condolences to your wife or to you. What if she too finds Micro C habit forming? Where will you keep the magazines then? If it's between you, let me know right away. It would be the first diagnosed case of multi-cornucopiitis (if the sound of that doesn't send chills down your doctor's spine, then he has no backbone). Because it is relatively new (no cases yet), there is no known cure (except, of course, expiration which is the ultimate cure, and too horrible to even consider).

Your Shugart should have a jumper option that lets it respond when either of two drives are selected. One side of the disk becomes one drive, the other is on the reverse side.

Note, that you must have a double-sided disk (the index hole is offset farther) to be able to access the back side (unless there is another option that lets the drive access the back side of a single-sided disk).

You will also have to format the back side—in a double-sided drive that looks like two drives—before you can access it. Formatting the disk in any other environment won't work.

Dear Editor,

After reading all the back issues of Micro C (and enjoying each one) I became somewhat overwhelmed by all the modifications your readers have submitted for the BB I.

In view of your background and experience, I would appreciate your suggestions on which modifications you would do if you were building a BB I and it were going to be your only system for a while.

Chris Paulsen
1138 E Electra Ln
Sandy, UT 84070

Editor's note:

Dana and I are building up a BB I system at this very moment (our third) and I'll tell you what we're doing to it (and in what order) in case you're doing a little at a time.

First, we will install the new character ROM (version 2.3). I refuse to look at the old characters.

Second, we will be doing the 4 MHz upgrade (we will use Otto Hiller's mod from Issue #2, page 3 and we will get the 16 MHz clock from the SWP dual density board). If I weren't planning to add the dual density board and I didn't want to spend much money, I'd go to issue #4 page 9 and do the 3.5 MHz mod (a Z80A processor might be the only part you'd need to purchase). If I were feeling rich, I'd do the 5 MHz mod with all Z80B parts and 150 ns RAM. (I'd be sure to read the Editor's note on that page, especially since I wrote it.)

Finally, of course, I will be adding the SWP dual density board. It halves the data access time and, of course, more than doubles the

amount of data on a disk. Great upgrade for the BB I.

I'd consider this a basic system (assuming two 8" Shugart or Siemens drives in a simple cabinet like the ones from Odd ball or Cascade Electronics). Don't scrimp on the keyboard unless you hunt and peck with a sledge. The keyboard and the monitor (and character set) can really make or break an otherwise good system. (I like my BMC green and USI PI-3 amber monitors.)

Other goodies that I'd add if there were money and time include—the EPROM programmer from Issue #6, a third drive, and the Dyna RAM disk package from LA Software or the Co-Power board from SWP (it has a RAM disk built in).

Dear Editor,

I am building a BB II from bare board and plan on using it in my MSCS research project (designing a graphics editor).

I would like to get into contact with other BB II owners so we can exchange hints and kinks. It would be great if they had access to a UNIX system so we could correspond by UNIX mail.

George Wilder
2 S 445 Cherice Dr
Warrenville, IL 60555
312-979-0817 work

Dear Editor,

I fixed the shift-lock problem in the PFM monitor. I put 18H (jump rel) at address F449H (was a 20H). This change forces the monitor to jump over the shift lock function.

The best change I made to my system, however, was the video wiggle fix by Darrell Collins in Issue #13 (page 36). This simple fix really put the video wiggle problem to bed for good. Darrell should give himself a pat on the back for coming up with this one!

Bryan Johnson
234 Ave Santa Barbara
San Clemente, CA 92672

Editor's note:

Your shift-lock fix is good for those who don't want the shift but do want to send nulls to programs etc. Those who would rather have their systems throw away nulls should, instead, change location F44BH from 34H to 00H.

The trick to translating the RAM address into the equivalent spot in ROM, is to subtract F000H and add 10H. So, the data in F44BH (in RAM) came from 045BH in the ROM.

Dear Editor,

I hate to be a spoil sport, but I think it's important so here goes. I got the schematics of the Kaypro II and they are very readable. However, the draftsman shows +5V as a signal ground symbol. The correct symbol should be a line, a circle, or an arrow.

Question: where do I get the rest of the schematics, such as power supply, CRT display, and drives? Back in the olden days, schematics for radios and TVs were hard to come by, but it wasn't long before you could get all the information from Sam's. Computers aren't there yet.

Harvey DeGering
1245 East Washington Blvd
Pasadena, CA 91104

Editor's note:

I agree with both your concerns regarding the lack of schematics and the lack of standardization of schematic symbols.

Actually, the up-arrow we used is probably the most standard way to signify +5V. The problem we faced trying to provide schematics for the other components on the Kaypro is that only the disk drives came from one source (and the circuit on the drives has changed). The power supplies and the CRT monitors came from innumerable sources and I'm not sure that even Kaypro could translate a serial number into a particular monitor or power supply.

Also, computer companies are very secretive about their circuits (especially when they are first released). Sam's simply reproduces the circuit diagrams and information provided by the TV and radio manufacturers. That, obviously won't work when many companies are trying to keep the general public from getting schematics.

Dear Editor,

I have two complaints with Micro C. One, it is late (and getting later) and two, the Xerox 820 column was discontinued. I can't believe there is nothing to write about. Some topics I'd like to see are: composite video interface and what bugs we will encounter when using the 820.

Michael Miller
(No address on note)

Editor's note:

I didn't stop the Xerox 820 column; John Marlin stopped writing it and no one else has taken it on. We are now paying our columnists $50 per column. I'd like to hear from anyone who would like to take on the Xerox 820.

Yes, there is a LOT of similarity between the two systems. 820 owners are using the PFM monitor listing in issues #1 and #2 as a guide when they disassemble the monitor in the 820. Disk formatters, serial print drivers, modem software, and the EPROM programming package run on the 820. All these programs are very machine specific so essentially any software should run identically on both systems. However, there is no reason that Xerox 820 owners (as well as BB I and Kaypro owners) wouldn't benefit from a regular 820 column.

For a composite video adaptor, check this issue.

```
0   GOTO 100
10  KILL "<PROGRAM_NAME>.BAK"
20  NAME "<PROGRAM_NAME>.BAS" AS
                    "<PROGRAM_NAME>.BAK"
30  SAVE "<PROGRAM_NAME>.BAS"
40  PRINT "<PROGRAM_NAME>.BAS SAVE AND
                              BACKED-UP"
50  END
100 REM BEGIN PROGRAM HERE
```

Figure 1 - Your BASIC Back-up Program

Dear Editor,

5 MHz is really great! I made the drive fix from issue 11 and speeded up my Kaypro and everything worked perfectly!

I compiled a little program in JRT Pascal. It took 55 seconds at 2.5 MHz and 40 seconds at 5 MHz (not half the time because of the disk accesses). Then I ran a test of prime numbers; lo and behold, the slow clock took 78 seconds, the fast took only 40. This is Great!

I found that I could plug a wire into U86's socket (where pin 4 used to reside) rather than soldering a wire to resistor R26. Radio Shack's ribbon cable has the right size conductor to plug into the socket. All in all, the instructions were very clear and the whole conversion (including disk timing, ROM monitor, and speed) took about 90 minutes.

Also, I discovered a little trick in BASIC. You see, MBASIC does not create a backup file when you save the program you are working on. So, I use the following code at the beginning of every program. (See Figure 1.)

To save your program the first time, enter RUN 30, then RUN 20 (avoids the "no file" error message). After that, simply enter RUN 10. If you simply enter RUN, then this program is bypassed and the main program is run.

Dave Truckenmiller
501 S Bennett
Palatine, IL 60067

(Letters continued on page 36)

Dumping the Screen on a BB I

By Glenn McEowen

5904 Waits Ave.
Fort Worth, TX 76133

Being your standard, card-carrying, low-budget tightwad, I was interested by Micro C's offer of a user disk for an article or a bit of software. Surely, I could write some dinky trifle that would qualify for a free disk!

Here, after many many hours of effort, is that "trifle." You may think it's a trifle too, but let me assure you, it turned into a major project!

It all started when I saw a nifty feature on an IBM small system computer (5120, system 23). This unit had a key that dumped the screen to the printer. So why couldn't I dump my Big Board screen too?

Introducing the Scroll Register

It should be easy: just get a byte from the screen memory and send it via BDOS to the LST: device. Not so simple, I soon found out. Mr. Ferguson, bless his heart, didn't want to bore us with simple screen memory addressing, so he threw in the scroll register to keep the screen guessing where its next byte was coming from. (That ridiculous little pun was truly an accident.) *(Editor's note: I'll bet!)*

Anyway, I had to uncalculate where the order of things in order to dump to the printer.

How It Works

Running SCRNDUMP.COM loads the program into an unused cranny in high memory and then warm boots the system. Now, whenever you want to copy the screen to the printer, you simply enter a cntl-W. Immediately, the LST: device starts receiving data. When it's finished, you are returned to your original program right where you left off.

When called by a cntl-W, SCRNDUMP first saves the registers, then proceeds to calculate the beginning screen RAM address after doing some fancy mathematical gymnastics with the scroll register. Finally, with the first address in hand, it bank switches to the screen RAM, reads the first byte, bank switches back, and sends the byte to CBIOS :LST routine.

One of my griefs with the program is that it could not simply send the byte to :LST through BDOS, but usually SCRNDUMP is called from BDOS and the reentrant problem was more than I could tackle. Anyway, the CBIOS jump vector

```
;*****************************************************
;*                                                   *
;*   SCRNDUMP.MAC: A screen dump to :LST device for  *
;*                 the Big Board                     *
;*                 by Glenn McEowen   7/25/83        *
;*                                                   *
;*****************************************************

                        Title   Scrndump.mac
                .Z80

                ;               Equates

DA00            DOSLC   EQU     0DA00H          ;Location 59.5K DOS
E80F            BIOSLST EQU     DOSLC+0E00H+0FH ;Jump vector in CBOIS to :LST
0000            WBOOT   EQU     0               ;C/PM warm boot
000D            CR      EQU     0DH             ;Carriage return
000A            LF      EQU     0AH             ;Line feed
0017            CHAR    EQU     017H            ;Dump character Cntl-W, activates
                                                ;"dump" to the :LST device
F009            CONIN   EQU     0F009H          ;Monitor location for console input
                                                ;routine which is over-written by
                                                ;SCRNDUMP
F009            CNOVL   EQU     CONIN           ;Overlay routine
                                                ;Checks for Cntl-W
3000            CRTMEM  EQU     03000H          ;Base of CRT ram
FF77            BASE    EQU     0FF77H          ;Current contents of scroll register
001C            BITDAT  EQU     01CH            ;System PIO port, for bank switching
                                                ;to CRT ram
FE00            SCRDMP  EQU     0FE00H          ;SCRNDUMP routine location
FECE            SCRSTK  EQU     SCRDMP+0CEH     ;SCRNDUMP private stack
F439            KBDIN   EQU     0F439H          ;Monitor's keyboard input routine

0000'                   ASEG
                        ORG     0100H

                ;       Relocate the SCRNDUMP routine to the high parts of RAM

0100    11 FE00 INIT:   LD      DE,SCRDMP       ;Load DE with pointer to bottom
                                                ;of SCRNDUMP routine
0103    21 011F         LD      HL,SCRL         ;Load HL with pointer to data in
                                                ;TPA
0106    01 00AA         LD      BC,CON-SCRL-1   ;Load BC with byte count
0109    ED B0           LDIR                    ;Move it up
010B    11 F009         LD      DE,CNOVL        ;DE points to the overwrite of
                                                ;CONIN routine
010E    21 01CA         LD      HL,CON          ;HL points to data to move
0111    01 0003         LD      BC,SCREND-CON   ;BC has byte count
0114    ED B0           LDIR                    ;Move it
0116    21 FEA8         LD      HL,DCHAR        ;HL points to "dump" character
                                                ;location
0119    3E 17           LD      A,CHAR          ;A gets the character
011B    77              LD      (HL),A
011C    C3 0000         JP      WBOOT           ;Go back to C/PM

011F            SCRL:   .PHASE  SCRDMP

FE00    CD F439 SCRDMP: CALL    KBDIN           ;Get the character from keybd
FE03    E5              PUSH    HL
FE04    21 FEA8         LD      HL,DCHAR        ;Check to see if it is the
FE07    BE              CP      (HL)            ;scrndump character
FE08    E1              POP     HL
FE09    C0              RET     NZ              ;character is not equal to
                                                ;"dump" character, so return
                                                ;to caller

FE0A    ED 73 FEA9 SCR: LD      (STSTOR),SP     ;Save the SP
FE0E    31 FECE         LD      SP,SCRSTK       ;Assign the new stack ptr
FE11    E5              PUSH    HL
FE12    D5              PUSH    DE              ;Save some things
FE13    C5              PUSH    BC
FE14    DD E5           PUSH    IX
FE16    21 3C00         LD      HL,CRTMEM+0C00H ;Get the top of screen ram
FE19    3A FF77         LD      A,(BASE)
FE1C    47              LD      B,A             ;Set B to (BASE) for later
FE1D    FE 17           CP      017H            ;Compare to max scroll count
FE1F    20 0F           JR      NZ,SCR0
FE21    21 3000         LD      HL,CRTMEM       ;HL is actual top of screen
FE24    3E 01           LD      A,01            ;Print 1 line before starting
FE26    32 FEA7         LD      (LCNT1),A       ;at the top of CRTRAM
```

```
FE29   3E 18              LD      A,018H
FE2B   32 FEA6            LD      (LCNT),A        ;LCNT gets total lines/screen
FE2E   18 13              JR      SCR2
FE30   04           SCR0: INC     B               ;Scroll is not at max, so we
FE31   3E 18              LD      A,018H          ;must set up for rollover.
FE33   32 FEA6            LD      (LCNT),A        ;LCNT gets total lines/screen
FE36   90                 SUB     B
FE37   47                 LD      B,A             ;B has the line counter
FE38   32 FEA7            LD      (LCNT1),A       ;LCNT1 gets line counter,too
FE3B   11 0080            LD      DE,080H         ;DE gets ram "line length"
FE3E   B7                 OR      A               ;Clear carry
FE3F   ED 52         SCR1: SBC     HL,DE           ;Decerase CRT memory line by
FE41   10 FC              DJNZ    SCR1            ;line
FE43   E5            SCR2: PUSH    HL              ;HL now contains the START
FE44   DD E1              POP     IX              ;Put HL into IX to save it
FE46   06 50              LD      B,050H          ;Character counter in B (80d)
FE48   F3            SCR3: DI                      ;Make sure nothing happens
FE49   DB 1C              IN      A,(BITDAT)      ;Switch memory bank select
FE4B   CB FF              SET     7,A
FE4D   D3 1C              OUT     (BITDAT),A
FE4F   4E                 LD      C,(HL)          ;Get the character from CRTRAM
FE50   DB 1C              IN      A,(BITDAT)      ;Switch memory bank back
FE52   CB BF              RES     7,A
FE54   D3 1C              OUT     (BITDAT),A
FE56   FB                 EI                      ;Interrupts back to normal
FE57   23                 INC     HL              ;Ready for next character
FE58   E5                 PUSH    HL              ;Save the CRTRAM pointer
FE59   C5                 PUSH    BC              ;Save the character count
FE5A   CD E80F            CALL    BIOSLST         ;Go print the character
FE5D   C1                 POP     BC              ;Get character counter back
FE5E   E1                 POP     HL              ;Get CRTRAM pointer back
FE5F   10 E7              DJNZ    SCR3            ;Repeat until line is done
FE61   CD FE9B            CALL    CRLF            ;Output a CR and LF

FE64   DD E5         NXTLN: PUSH    IX              ;Get START back into HL
FE66   E1                 POP     HL
FE67   3A FEA6            LD      A,(LCNT)        ;Let's see where we are
FE6A   3D                 DEC     A
FE6B   32 FEA6            LD      (LCNT),A
FE6E   28 1C              JR      Z,FIN           ;If 0 we are done
FE70   3A FEA7            LD      A,(LCNT1)
FE73   3D                 DEC     A
FE74   32 FEA7            LD      (LCNT1),A
FE77   28 07              JR      Z,NXT1          ;We're not to 3000h
FE79   11 0080            LD      DE,080H
FE7C   B7                 OR      A               ;Clear carry flag
FE7D   19                 ADD     HL,DE           ;HL points to next line
FE7E   18 C3              JR      SCR2            ;Go do another line

FE80   3A FF77       NXT1: LD      A,(BASE)        ;Scroll register in A
FE83   3C                 INC     A               ;So last line is correct
FE84   32 FEA7            LD      (LCNT1),A       ;Put it LCNT1 for dec's
FE87   21 3000            LD      HL,CRTMEM       ;HL has start of CRTRAM
FE8A   18 B7              JR      SCR2            ;Go do the last lines

FE8C   CD FE9B       FIN:  CALL    CRLF            ;Do another CR
FE8F   DD E1              POP     IX              ;Restore everything
FE91   C1                 POP     BC
FE92   D1                 POP     DE
FE93   E1                 POP     HL
FE94   ED 7B FEA9         LD      SP,(STSTOR)     ;Put the old stack back
FE98   C3 FE00            JP      SCRDMP          ;Done "dumping", get
                                                  ;another character

FE9B   0E 0D         CRLF: LD      C,CR            ;Print a CR and LF
FE9D   CD E80F            CALL    BIOSLST
FEA0   0E 0A              LD      C,LF
FEA2   CD E80F            CALL    BIOSLST
FEA5   C9                 RET


FEA6                 LCNT:   DEFS    1             ;Place to keep line counter
FEA7                 LCNT1:  DEFS    1             ;Place to keep sub-counter
FEA8                 DCHAR:  DEFS    1             ;Place to keep "dump" char
FEA9                 STSTOR: DEFS    2             ;Store old stack pionter

                             .dephase

01CA                 CON:    .phase  CNOVL

F009   C3 FE00       CNOVL:  JP      SCRDMP        ;This segment overlays
                                                   ;the old CONIN jump vector
                                                   ;in the PFM monitor

                             .dephase

01CD                 SCREND: END
```

END

entry proved a workable solution. So we send the byte to :LST then go back and get another.

When we reach the end of the line, we do a CRLF carriage return, line feed) and then go back to the calculations for the next line which may be in sequence or may not, depending on the scroll register.

When it finishes dumping the screen, SCRNDUMP outputs another CRLF for good measure, restores the registers, and returns to the caller, waiting patiently for another chance to perform.

More Boring Details

I chose to use cntl-W because it isn't used very often by other programs. (Unfortunately, Wordstar uses every control character.) Anyway, it's easy enough to change the character. (It's named CHAR and resides at FEA8H.)

The program lies in the unused (I hope) memory block between FE00 and FED0. It keeps its counters and its stack within the block.

SCRNDUMP also overlays the first 3 bytes of the monitor's CONIN routine (@F009) with a jump to SCRNDUMP at FE00. From here the buffer is read and tested for a cntl-W. If there is no match, the system continues as normal. If the character was a cntl-W, then the dump to LST: device starts. When the dump is complete, you are right back in your original program with no change to the screen.

I set the parameters to fit into the standard Big Board configuration with DRC's CBIOS and the 59.5K CP/M. You can alter it for different size systems by changing DOSLC, you can select a new control character by changing CHAR, and you can change the location of the program by changing SCRDMP. Then, of course you have to reassemble it. I used MicroSoft's M80 and L80 for the assembly.

SCRNDUMP has worked successfully from inside DDT, ZSID, a CB80 compiled program, CP/M's CCP, and even Wordstar (but I had to give up the upscroll which is cntl-W).

Editor's note: You Kaypro folks can modify this code for your system. The Kaypro Video is at the same location as the Big Board. The difference is that you don't have any free RAM at FE00 (You might try putting the program below the CCP) and you don't have a scroll register. So, the video RAM maps directly onto the screen, and you can remove the address calculations.

■ ■ ■

BB II Video Calculations

By Douglas H. Durland

663 Georgia Ave.
Palo Alto, CA 94306

The goal of this article is to help Big board II owners improve their video display.

Owners of the board who use the onboard video driver with the 9 by 11 character set soon discover that the 6845 controller programming which comes on the original monitor needs to be tuned up. Cal-Tex now supplies some suggestions for improvement, but they do not go far enough.

I'm going to discuss a few basics on how the 6845 works. Following these basics, there is a Pascal program that will calculate the values you need to program into the 6845 for your own monitor.

A Look at the 6845

The 6845 is basically a timing/synchronizing device which controls when the character data stored in the video RAM is presented to the 8002 output device, and it controls the timing of the horizontal and vertical sync pulses. It is this control of horizontal and vertical timing that we will look at here.

Registers

Register 0 controls the horizontal sync rate. Its contents are 1 less than the number of character times per horizontal line. Since a character time is 9 times the dot crystal period, the period between horizontal sync pulses can be calculated from the dot clock frequency and the contents of register 0.

Conversely, if you know the crystal frequency and the desired horizontal sweep rate (to match your monitor requirements) you can calculate the number you need in register 0.

Register 1 controls the number of characters to be displayed per line. It is usually set at 80.

Register 2 controls the timing of the H sync pulse with respect to the start of the line. This timing varies greatly for various monitors. This value does not affect the horizontal sweep rate.

Register 3 controls the width of the V and H sync pulses.

Registers 4 and 5 control the vertical timing.

Register 4 contains 1 less than the number of full lines per frame (a full line is 12 scan lines).

Register 5 lets you set the vertical sweep rate very precisely. Its contents are the number of scan lines to be added to the full lines to produce the correct interval between V sync pulses.

Register 6 contains the number of character lines to be displayed. It is normally fixed at 24.

Register 7 controls the timing of the V sync pulse with respect to the first character line. Different monitors have quite different timing requirements. The contents of this register do not affect the vertical sweep RATE.

So much for the background. Now let's see what we can do with this information.

Original BB II

As BB II arrives, it is programmed for an 18.3 KHz H sweep rate, and a 55.9 Hz V rate. You can prove this to yourself with the Pascal program.

The horizontal position (register 2) will generally place some of the data off the right hand side of the screen. This is easy to change, but other modifications aren't so easy.

Let's assume you have purchased a monitor with a nominal 18.6 KHz horizontal sweep. Unless it is awfully well shielded, you'll find you have a 4 Hz swim (the vertical scan rate is about 4 Hz less than 60 Hz).

So you try to get rid of the swim by tweaking registers 4,5, and 6. If you are lucky you get a vertical of 60.09 Hz, and don't see any swim.

Unfortunately, you can't count on being lucky. My first experience was with a Motorola 3003 (an excellent monitor!). My drives are right next to it, and the swim was unacceptable.

When I programmed the 6845 for a 60 Hz sweep, the swim disappeared but I discovered that the resulting 97 character horizontal line did not allow enough time for horizontal retrace, and I was losing characters. OK, so up the horizontal character number and I tweaked some more.

Video Crystal Frequency

Nuts, it looked like I had to choose between 80 characters and swim. The video crystal frequency was too low! (I should pause at this point and mention

Figure 2 - Routine to Reprogram the 6845

```
; THIS PROGRAM OUTPUTS A GROUP OF BYTES TO THE 6845 CRT
; CONTROLLER ON BIGBOARD II.  THIS PARTICULAR GROUP WORKS
; ON AN AMDEK 310 WITH A DOT CLOCK CRYSTAL OF 16 MZ.
; THE RESULTING H FREQ IS 18.33 KHZ.  THE V IS 60.09 HZ
;
; WRITTEN IN 8080 CODE SO IT CAN BE ASSEMBLED WITH ASM
;
; DOUG DURLAND 8-20-83
;
     ORG  100H
;
     LXI  H,TABLE        ; POINT TO 8 VALUES
     MVI  C,0       ; START AT REGISTER 0
     MVI  B,8       ; COUNT
OUTLP:                   ; HOW'S THAT FOR AN ORIGINAL NAME?
     MOV  A,C
     INR  C         ; DONT FORGET
     OUT  0DCH      ; SELECT 6845 REGISTER
     MOV  A,M
     INX  H         ; MAY AS WELL DO IT NOW
     OUT  0DDH      ; PUT THE VALUE INTO IT
     DCR  B
     JNZ  OUTLP
;
     JMP  0         ; BACK TO CP/M
;
TABLE:
     DB   96        ; 97 CHARACTERS TOTAL-1
     DB   80        ; DISPLAY 80
     DB   83        ; H SYNC POSITION
     DB   28H       ; SYNC WIDTHS
     DB   24        ; 25 LINES TOTAL-1
     DB   5         ; 5 EXTRA SCAN LINES
     DB   24        ; DISPLAY 24 LINES
     DB   24        ; V SYNC POSITION
;
     END
```

that further experience with other monitors—some Zeniths and an AMDEK 310—posed no problem at all. They worked just fine.)

I purchased an 18 MHz crystal (CalTex ships a 16 MHz) and I finished the programming. It's obvious that an 18 MHz crystal will be needed fairly often.

If this whole process sounds like a pain in the you-know-what, you're right, it was!

But when other folks around here found out I had successfully interfaced different monitors to the BB II, they wanted to know how to do it. So I wrote this article and the following software.

If you enter your dot-clock crystal frequency, your nominal H scan frequency, and your local line frequency, it will tell you what values to put in your 6845. It will tell you when the solution is marginal and when there is no solution.

For example, those who wish to use standard monitors (15.75 KHz sweep) are going to have to settle for a slow vertical sweep rate. If you want 24 lines at 12 scans per line, there just isn't enough time in 1/60th of a second.

Doing the Calculation

Enter the calcbyte program which follows and run it under JRT Pascal vrs 3.0 (Editor's note: This program should be relatively easy to translate into another version of Pascal or into another language, that's the beauty of a very readable language like Pascal.)

Now run the program. If you find that you lose horizontal characters when you program the 6845 with the new values, then install an 18 MHz crystal.

There is an easier way to program the 6845 than outputting bytes from the console. Make yourself a .COM file which outputs the group. You can play with it with DDT while you are experimenting with the values. I have included an assembly language routine which works for an AMDEK 310 with the original 16 MHz crystal, so you can try it out. It assembles under ASM. See Figure 2.

I hope this will help all of you get a rock solid and beautiful display on this great computer.

■ ■ ■

Figure 1 - JRT Pascal Program to Calculate 6845 Values

```
program calcbytes;

{ This program compiles and runs under JRT Pascal version 3.

Its purpose is to permit interactive development of an optimum
set of bytes to be programmed into the 6845 CRT controller which
is used on the BIGBOARD II.

  Doug Durland
  663 Georgia Ave.
  Palo Alto, CA 94306 }

type

  double = array[1..2] of char;

var
  answer : char;

  xtal_frequency, monitor_frequency, character_time,
  line_time, actual_h_freq, actual_v_freq : real;

  nmbr_of_characters, number_of_lines, vertic1_correction,
  vertical_total, power_line_frequency : integer;

function get_char : char;
var r : record
    flag,a,c,b,e,d,l,h : char;
    end;
begin
r.c := chr(1);
call (5,r,r);
get_char := r.a
end;

function hexint (x:integer) : double;
   var
     digit_value : array[1..2] of integer;
     i : integer;

     begin {converting to hex}
       digit_value[1] := x div 16;
       digit_value[2] := x mod 16;
         for i := 1 to 2 do
           if digit_value[i] < 10
             then
               hexint[i] := chr(digit_value[i] + ord('0'))
           else
             hexint[i] := chr((digit_value[i]-10) + ord('A'))
     end;

procedure do_the_work;

begin
  write ('enter your crystal frequency in MHz ');
  readln (xtal_frequency);
  write ('enter your monitor scan frequency in KHz ');
  readln (monitor_frequency);
  write ('what is your local power line frequency, in Hz? ');
  readln (power_line_frequency);
  nmbr_of_characters:=round(1000*xtal_frequency/(9*monitor_frequency));
  if nmbr_of_characters < 100
    then
      begin
    writeln;
    writeln ('Warning!  This combination results in less than');
    writeln ('100 characters total, and may give trouble with');
    writeln ('inadequate retrace time with some monitors.  Try');
    writeln ('to adjust your crystal/monitor scan combination to');
    writeln ('achieve at least 100 characters total.');
    writeln ('press <CR> to continue.');
    repeat
      answer := get_char
    until answer = chr(0dh);
     end;
```

(Program continued next page)

JRT Pascal Program to Calculate 6845 Values

(continued from page 7)

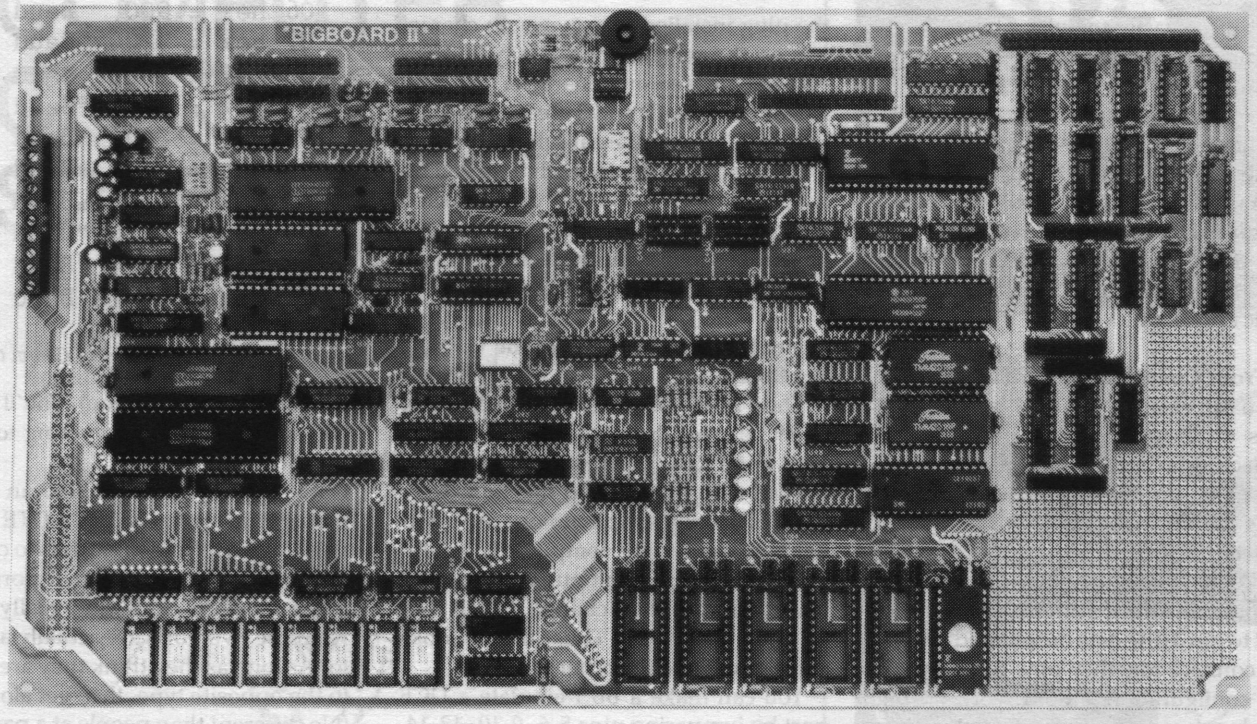
```
  character_time:=9.0E-06/xtal_frequency;
  line_time:=nmbr_of_characters*character_time;
  number_of_lines:= round((1/power_line_frequency)/line_time);
  vertic1_correction:= number_of_lines mod 12;
  vertical_total:= number_of_lines div 12;

  if vertical_total < 25
   then
     begin
       writeln;
       writeln ('Warning!  Cannot achieve desired vertical scan rate.');
       writeln ('Proceeding on 25 line total basis. (24 displayed)');
       writeln ('Press <CR> to continue.');
       repeat
         answer := get_char
       until answer = chr(0dh);
       vertical_total:=25;
       number_of_lines:=12*25;
       vertic1_correction:=0
     end;

  actual_h_freq:= 1/(1000*line_time);
  actual_v_freq:= 1/(number_of_lines*line_time);
  writeln;
  writeln ('crystal frequency=',xtal_frequency,' MHz');
  writeln('actual h freq.=',actual_h_freq:6:2,' KHz');
  writeln('actual v freq.=',actual_v_freq:6:2,' Hz');
  writeln('number of characters= ',nmbr_of_characters);
  writeln('vertical total= ',vertical_total);
  writeln('vertical correction= ',vertic1_correction);
  writeln;
  write ('R0=',nmbr_of_characters-1:4);
  write ('    ',hexint(nmbr_of_characters-1),'H');
  writeln ('    ; number of characters/line-1');
  write ('R1=  80');
  write ('    50H');
  writeln ('    ; number of displayed char/line');
  write ('R2= ',80+((nmbr_of_characters-80)div 2));
  write ('    ',hexint(80+((nmbr_of_characters-80)div 2)),'H');
  writeln ('    ; horizontal sync position (varies with monitor)');
  write ('R3=  28H   28H');
  writeln ('    ; V sync length (2)/H sync width (8)');
  write ('R4= ',vertical_total-1);
  write ('    ',hexint(vertical_total-1),'H');
  writeln ('    ; character lines total-1');
  write ('R5= ',vertic1_correction:3);
  write ('    ',hexint(vertic1_correction),'H');
  writeln ('    ; added scan lines to trim V frequency');
  write ('R6=  24    18H');
  writeln ('    ; number of character lines displayed');
  write ('R7=  24    18H');
  writeln ('    ; vertical sync position (varies with monitor)');
  write ('R8=   0     0');
  writeln ('     ; no interlace and no skew');
  write ('R9=  11    0BH');
  writeln ('    ; number of raster lines/char -1');
end;

begin {the repetetive part}

  repeat
    do_the_work;
    writeln;
    write ('Do you want to try other values? (y/n) ');
    readln (answer);
    writeln;
  until ((answer='n') or (answer='N'))

end.
```

END

NEW LOWER PRICES! ***NOW IN "UNKIT"* FORM TOO!***

"BIG BOARD II"

4 MHz Z80-A SINGLE BOARD COMPUTER WITH "SASI" HARD-DISK INTERFACE



$795 ASSEMBLED & TESTED $545 "UNKIT"* $245 PC BOARD WITH 16 PARTS

Jim Ferguson, the designer of the "Big Board" distributed by Digital Research Computers, has produced a stunning new computer that Cal-Tex Computers has been shipping for a year. Called "Big Board II", it has the following features:

■ 4 MHz Z80-A CPU and Peripheral Chips

The new Ferguson computer runs at 4 MHz. Its Monitor code is lean, uses Mode 2 interrupts, and makes good use of the Z80-A DMA chip.

■ 64K Dynamic RAM + 4K Static CRT RAM + 24K E(E)PROM or Static RAM

"Big Board II" has three memory banks. The first memory bank has eight 4164 DRAMs that provide 60K of user space and 4K of monitor space. The second memory bank has two 2Kx8 SRAMs for the memory-mapped CRT display and space for six 2732As, 2Kx8 static RAMs, or pin-compatible EEPROMS. The third memory bank is for RAM or ROM added to the board via the STD bus. Whether bought as a bare board, an "unkit"*, or assembled and tested, it comes with a 2732 EPROM containing Russell Smith's superb Monitor.

■ Multiple-Density Controller for SS/DS Floppy Disks

The new Cal-Tex single-board computer has a multiple-density disk controller. It can use 1793 or 8877 controller chips since it generates the side signal with TTL parts. The board has two connectors for disk signals, one with 34 pins for 5.25" drives, the other with 50 pins for 8" drives.

■ Vastly Improved CRT Display

The new Ferguson SBC uses a 6845 CRT controller and SMC 8002 video attributes controller to produce a display rivaling the display of quality terminals. There are three display modes: Character, block-graphics, and line-graphics. The board emulates an ADM-31 with 24 lines of 80 characters formed by a 7x9 dot matrix.

■ STD Bus

The new Ferguson computer has an STD Bus port for easy system expansion.

■ DMA

The new Ferguson computer has a Z80-A DMA chip that will allow byte-wise data transfers at 500 KBytes per second and bit-serial transfers via the Z80-A SIO at 880 Kbits per second with minimal processer overhead. When a hard-disc subsystem is added, the DMA chip makes impressive disk performance possible.

SIZE: 8.75" x 15.5"
POWER: +5V @ 3A, +-12V @ 0.1A

■ "SASI" Interface for Winchester Disks

Our "Big Board II" implements the Host portion of the "Shugart Associates Systems Interface." Adding a Winchester disk drive is no harder than attaching a floppy-disk drive. A user simply 1) runs a fifty-conductor ribbon cable from a header on the board to a Xebec controller that costs only $295 and implements the controller portion of the SASI interface, 2) cables the controller to a Seagate Technology ST-506 hard disk or one compatible with it, and 3) provides power for the controller-card and drive. Since our CBIOS contains code for communicating with hard-disks, that's all a user has to do to add a Winchester to a system!

■ Two Synchronous/Asynchronous Serial Ports

With a Z80-A SIO/O and a Z80-A CTC as a baud-rate generator, the new Ferguson computer has two full RS232-C ports. It autobauds on both.

■ A Parallel Keyboard Port + Four Other Parallel Ports for User I/O

The new Cal-Tex single-board computer has one parallel port for an ASCII keyboard and four others for user-defined I/O.

■ Two Z80-A CTCs = Eight Programmable Counters/Timers

The new Ferguson computer has two Z80-A CTCs. One is used to clock data into and out of the Z80-A SIO/O, while the other is for systems and applications use.

■ PROM Programming Circuitry

The new Cal-Tex SBC has circuitry for programming 2716s, 2732(A)s, or pin-combatible EEPROMs.

■ CP/M 2.2**

CP/M with Russell Smith's CBIOS for the new Cal-Tex computer is available for $150. The CBIOS is available separately for $25.

---

* *The "unkit" is a fully-socketed, wave-soldered "Big Board II". It requires NO soldering. All an "unkit" purchaser must do is carefully insert the prime ICs we supply in the proper sockets and systematically proceed to bring up and test the board.*

***CP/M is a registered trademark of Digital Research*

CAL-TEX COMPUTERS, INC.

780 E. TRIMBLE ROAD #504 • SAN JOSE. CA 95131 • (408) 942-1424

Terms: Orders paid for with a cashier's check or bank card will be shipped within three working days. Orders paid for with a personal check will be shipped within three weeks. Add $5 for packing & shipping in North America.

If Your Serial Port Isn't Feeling Its Oats

By John Allen

40 Rugg Road
Allston, MA 02134

It's often easier to build a board, connect up the power supply, the drives, the keyboard, and monitor than it is to connect up a simple peripheral via the "standard" RS-232 interface.

There are so many things that can go wrong that it's hard to tell whether a communications problem lies in the computer, the cable, or the peripheral. The following should make your diagnostic efforts a little easier.

Can Your Computer Speak RS-232?

You can easily check whether the system's serial port is working, without connecting a modem or printer to the port. You do this by connecting the port's output back to its input, so it talks to itself.

Take an RS-232 plug (usually a male will plug into your system) and connect pin 2 to pin 3 (the data output and input), and connect together pins 5, 6, and 8. This configuration should work on Big Boards set up with the DTE (T) jumpers and with Kaypros. DTE stands for Data Terminal Equipment or terminal for short.

If, on the other hand, your computer is set up to look like DCE (Data Communications Equipment—a modem), then you will still jumper pin 2 to pin 3, but the other three pins that you will connect together will be 4, 6, and 20.

Pin 6 supplies +12V so you are using it to pull up the handshaking pins 5 and 8 (DTE) or 4 and 20 (DCE). This high tells your system that it is OK to send data.

You can make up two plugs and check to see whether a port is set up as a DTE port or a DCE port. Whichever one works in the port tells you what kind of port you have (assuming the port works at all).

Running the Test

To test a port, install the new RS-232 plug, get into a modem program that has been configured for your system—Modem7, for instance. Enter "T" to get into terminal mode, then watch to see if the keys you type show up on the screen.

On some modem programs, there is an internal echo, on others there isn't unless you ask for it.

To find out which you have, temporarily disconnect pin 2 from pin 3, but leave the plug installed. If you see no characters on the screen when you type, then you should see one character for each keystroke when 2 and 3 are reconnected (you don't have internal echo).

If you see a character each time you hit a key while 2 and 3 are disconnected then you should see 2 characters for each keystroke when they are reconnected (if your serial port is talking, that is).

Easier Communications for the Big Board

The "Talking Serially" article in Micro C #9 tells you how to transpose signal lines in an RS-232 cable so two computers can talk. I have used this approach, but it is usually simpler to use the internal jumpers. If one computer is configured as DTE (most computers are configured this way) then change the BB (I or II) to emulate DCE. Thus, the BB looks like a modem to the other machine.

You can make a BB I emulate a DCE port by jumpering pins 5-6, 9-10, 13-14, 17-18, 21-22, and 25-26 on JB4 or JB5. On JB4 (port A) you will also need to jumper pins 31-32 and 35-36 (provide the clock to the SIO). On JB5 (port B) these extra clock jumpers aren't used.

BB II jumpers are similar; consult the BB II documentation.

Anyway, once you have reconfigured the BB as a DCE serial port, you can talk to just about any other system using a straight cable in which no signals are transposed.

One system you can't talk to, however, is the Osborne. The Osborne was told at birth that it was a modem (probably explains the small screen). You'll have to go back to the DTE jumpering to talk to an Osborne.

As you can probably see, an RS-232 port that is configured as a DTE port (it thinks it is a terminal) expects to communicate with an RS-232 DCE port (looks like a modem). It doesn't really matter what the two ports are actually attached to. As long as they think they are talking to the opposite type, they get along fine. (No different than people at a party.)

```
00 hex = 50     Baud
01 hex = 75     Baud
02 hex = 110    Baud
03 hex = 134.5  Baud
04 hex = 150    Baud
05 hex = 300    Baud
06 hex = 600    Baud
07 hex = 1200   Baud
08 hex = 1800   Baud
09 hex = 2000   Baud
0A hex = 2400   Baud
0B hex = 3600   Baud
0C hex = 4800   Baud
0D hex = 7200   Baud
0E hex = 9600   Baud
0F hex = 19.2K  Baud
```

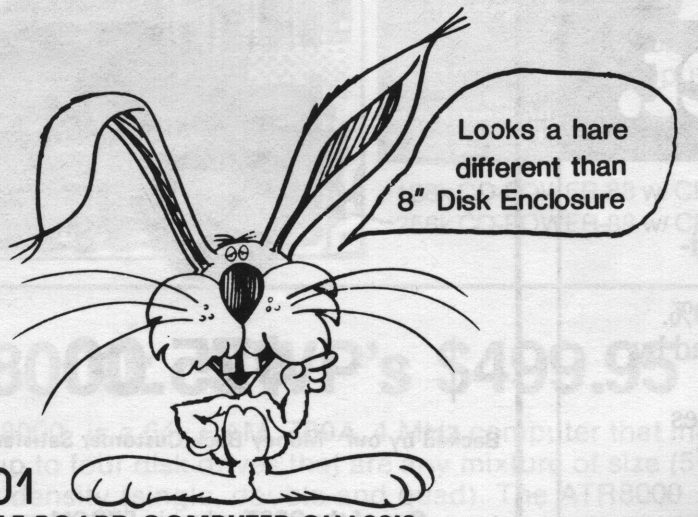
Figure 1 -
Baud Rate Generator Control Characters

Talking Fast

Once you have two systems talking (both using modem7 or the like which checks for errors), then try raising the baud rate until you begin to see errors. Then drop back a notch. To change the baud rate in Modem7, use PFM or DDT to change the value of location 14F9. See Figure 1 for the hex values to put in that location. These values should be correct for the BB I, the Xerox 820, and the Kaypro II and 4.

■ ■ ■

SINGLE BOARD COMPUTER CHASSIS $275



2901

SINGLE BOARD COMPUTER CHASSIS — NON S100 — 2 EIGHT INCH DRIVES — REGULATED PS

Cabinet size: 20"w × 22.4"d × 7.5"h. Cabinet painted dove grey; front, back and side accents black. Mounts 2 SHUGART SA851R EIGHT INCH FLOPPY DISK DRIVES or MECHANICAL EQUIVALENT*. Lighted reset switch on front panel. Lamp supplied, but not connected. 4 DB25, 1 Delta 50, 1 Centronics connector cutouts, two switched ac receptacles, 6' three wire line cord, EMI filter, 70CFM fan with washable dust filter, line fuse, power switch on rear panel. P2901 removable module power supply: +5@3A, +12@5A, −12@.5A for computer; +5@2.5A, +24@2.5A−4A peak, −5@.25A for disk drives. Includes power cable from power supply to computer (unterminated wires at computer end) and power cable to 2 disk drives (specify single or double sided drives when ordering).

INTEGRAND

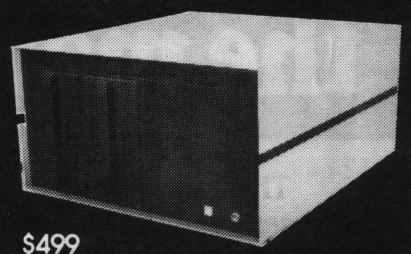
8620 Roosevelt Ave.
Visalia, CA 93291
209-651-1203

Main/Frames

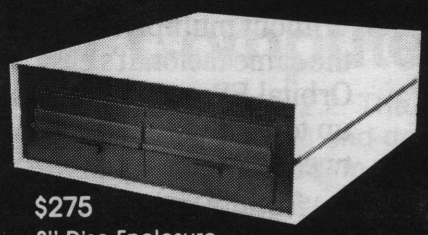
Main/Frames

from
$175

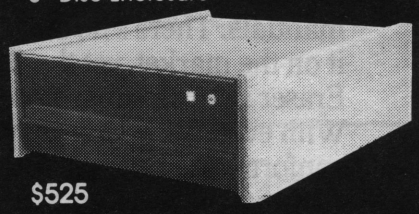
- 70 Models of Enclosures
- Assembled and tested
- Quasi-Coax Motherboards
- Power Supply
- Card cage and guides
- Fan, line, cord, fuse, power & reset switches



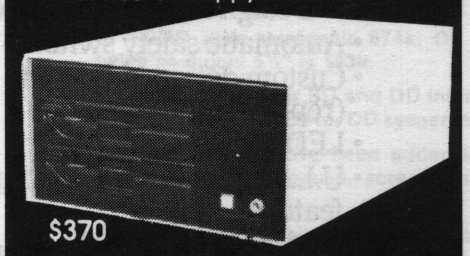
$499
8" Floppy Main/Frame



$275
8" Disc Enclosure



$525
Phase/80 8" Floppy Main/Frame



$370
Slim Line 8" Floppy Main/Frame

Write or call for our brochure which includes our application note:
"Making micros. better than any ol' box computer"

INTEGRAND

8620 Roosevelt Ave. • Visalia, CA 93291
209/651-1203

We accept BankAmericard/Visa and MasterCharge

Shedding some light on the best EProm Eraser on the market.

The unique Orbital EProm Eraser—multiple exposure multiplies productivity. Here's how to improve your productivity by 140%. Without multiplying your overhead by the same factor. It's easy, with the Orbital EProm Eraser, which erases up to 14 units at a time. All in just twenty minutes.

Totally unique, but conforms to all data sheet specifications erasing manuals. There's no other product like it on the market, yet the Orbital EProm Eraser performs like a proven winner. With even ultra-violet exposure that conforms to Federal Standards for UV radiation at 2537 angstroms.

Design features include:

- Automatic safety switch
- Custom designed long-life lamp (replacements available)
- LED indicator
- U.L. Approval— another unique feature

The most unique feature of all— the price: $49.95 including shipping.

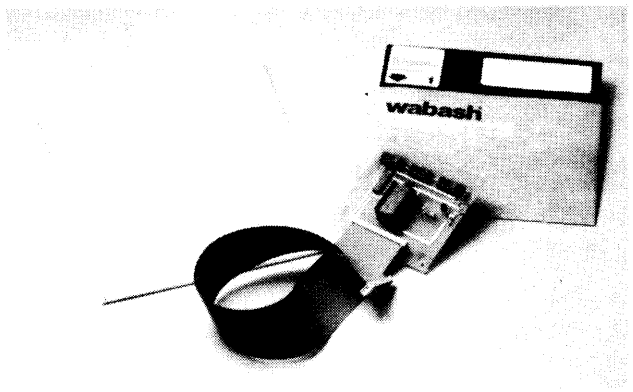
Shipped UPS: Master Charge or VISA Order 1-800-241-8003, ext. 2537.

Check or Money Order payable to:

EProm Eraser Co.

1753 Tully Circle, Atlanta, Ga. 30329

Big Board and Xerox 820 EPROM programmer kit



$75.00

Backed by our "Money Back Customer Satisfaction Policy"

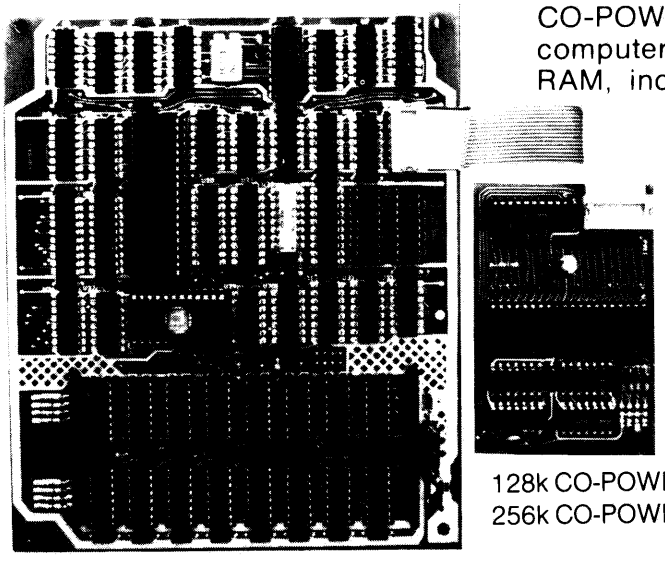
- One of the BEST values in EPROM programmers
- Plugs into parallel port
- Kit includes all parts, including ribbon cable and ZIF socket
- User friendly menu-driven software written in BDS C
- Includes all 1800 lines of source code
- Verifies erased EPROMs
- Programs
- Displays contents in Hex and ASCII
- Saves EPROM to disk
- Loads EPROM from disk
- Verifies programming
- Calculates EPROMs checksum
- Programs 2516, 2716 (5V), and 2532 EPROMs

Send for a complete documentation package or send your check or money order in the amount of $80 in U.S. funds (includes $5 shipping and handling) for immediate delivery. (N.Y. State residents add 7% sales tax.)

KCT

Kenmore Computer Technologies
P.O. Box 635
Kemmore, NY 14217
(716) 877-0617

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 CP/M-86 and MSDOS, complete with IBM-PC compatibility. A facility for transferring data files between CP/M and MSDOS is included.

Simple commands move system control between the Z80 and 8088 processors. CO-POWER-88's RAM can be used in CP/M as a RAM drive! Currently available for Kaypro, Bigboard, Zorba, Xerox, Access Matrix, Televideo, and ATR8000 computers.

128k CO-POWER-88 w/ CP/M-86 and MSDOS $550.00
256k CO-POWER-88 w/ CP/M-86 and MSDOS $699.95

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 inferfaces 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 comptuers. 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.**

Price: $149.95

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.

SWP Inc
MICROCOMPUTER PRODUCTS

2500 E. Randol Mill Rd – 125
Arlington, Texas 76011
817/469-1181

(We used to be Software Publishers, Inc.)

Great Eight Kaypro

By Dana Cotant

One of the most common requests we have received is for an article that describes how to upgrade a Kaypro II to a Kaypro 4. The first portion of this article covers that modification.

Turning your system into a 4 doesn't do much, however, for those of you who already have 4's. So, the last part of this article covers turning your 4 into an 8. It doesn't matter whether you started with with a II or a 4, once you have a 4 you can have an 8. (Then, of course, you might consider trading your new 8 for a 4 and two IIs so you can start over.)

Will the 4's Please Stand?

Note that even if your system is marked "Kaypro II," it may already be a 4 at heart. You may only need to change to double sided, 48 tpi Tandon TM 100-2s and get a copy of the 4 formatter in order to turn your system into a Kaypro 4.

Your Kaypro II has the heart of a 4 if it has a 2732 monitor ROM and the smaller ICs are soldered in rather than socketed. Look closely at U47 (it has paper stuck to the top) if it has 2716 printed on the ceramic you have a II board. If it has 2732 printed on the ceramic, then you probably have a 4 board.

Turning a II into a 4

First the Kaypro 2 board needs to be modified to use a 2732 for the monitor ROM. The extra code needed for drive side select will not fit into the nearly-full 2716 monitor ROM of the Kaypro 2.

To run a 2732 in a Kaypro 2 you first need to pick up an additional address line, A11. Pin 2 on U59 is address line A11.

You need to connect A11 to pin 21 on the 2716 socket. Pin 21 is normally tied to +5V. So, when you are ready to put a 2732 into the socket bend pin 21 out so that it doesn't go into the socket and then add a jumper between the bent out pin and U59 pin 2.

You have another option if you don't want to solder to the Monitor ROM (makes it easier to change ROMs later). Simply plug a second 24 pin dip socket on top of the original ROM socket. Bend out pin 21 on the new socket so that it doesn't go into the original socket. Now add the jumper to the socket's pin 21.

An additional select line from decoder U60 must be used to select the second half of the 2732.

To use the select line, remove U60, bend out pin 15, and re-insert U60 into the socket, making sure that pin 15 does not make contact with the socket. Now let's gate in this additional signal.

There are a number of unused gates on the Kaypro (see the Micro C schematic for a list). Each of these gates has one or more of its inputs grounded. On U80, pin 5 is a grounded input. Remove U80, bend out pin 5, and re-insert U80. Now connect U80 pin 5 to U60 pin 15.

Of Pins and Pads

When you are working on the underside of the board, you count IC pins clockwise. Pin 1 is just right of the notch on the IC. A solder pad is the small circular piece of copper circuit trace to which socket pins are soldered. When we say pad, we mean the pad on the board. When we say pin we mean the pin on the IC itself. So, after pulling out U60 pin 15 we can jumper to U60 pin 15 (the pin on the IC) and also jumper to U60 pad 15 (the pad on the circuit board that the pin used to be connected to).

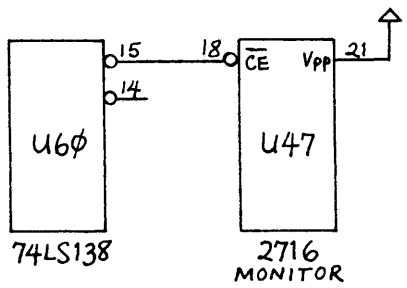
Working Under the Board

Jumper the following solder pads. Connect U80 pad 6 to U60 pad 15. This connects the output of U80 to the output enable (OE) input of the 2732. Also connect a jumper between U80 pad 4 and U60 pad 14. This brings the select for the upper 2K of ROM from the decoder U60 to a gate input in U80.

Floppy Drive Side Select

We have to find a spare inverter on the board and it turns out that U73, a 74LS04 hex inverter, has an unused gate. However, the inverter has to be fast, and have lots of output so we'll replace this 74LS04 with a 74S04. (They are identical except that the S part is faster and has more output than the LS part.)

Go ahead and remove U73. Before inserting the 74S04 in its place, bend out pin 5. Pin 5 is the grounded input of our unused inverter. Insert the 74S04 into socket U73 making sure pin 5 does not make contact with the socket. Connect this bent-out pin to the solder pad just right of the "E40" printed on top of the



Monitor Select Circuit (Before)

circuit board. The "E40" pad is just a convenient way to connect U73 pin 5 (the bent out pin) to U72 pad 13 (the pad underneath the board). This brings the otherwise unused bit 2 from the system PIO to the inverter.

Next we'll connect the output of the inverter to the disk drive cable.

On the solder side of the board, connect U73 pad 6 to J6 pad 32. J6 is the floppy drive connector. Note how the even numbers run up one side of the double row header and the odd pin numbers are on the other side (numbers are on the top of the board).

Now your Kaypro 2 will be able to run double-sided drives with the appropriate monitor ROM. All you need now is a Kaypro 4 monitor ROM (available from Micro C) and you have a more or less official Kaypro 4 board. Then, just replace one or both of your drives with double sided 48 tpi units (they plug right in) and you have 390K per disk.

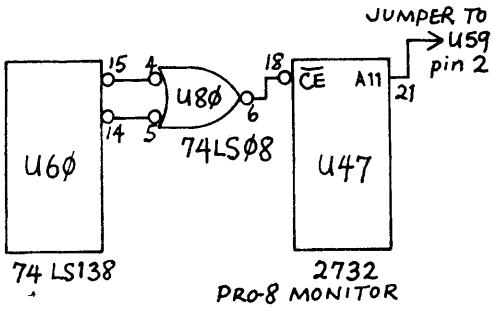
Turning a 4 into an 8

Since everyone now has a 4, it's trivial to go to 8. All you need to do is plug in a new monitor ROM and add some new disk formatters to your system disk.

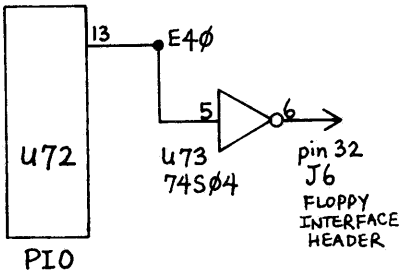
You get all these, plus the cursor setting program, in the Micro Cornucopia Pro 8 package. You get a special 2732 monitor ROM and the formatters you need to operate any drives you want. In other words, you can use any mix of single sided 48 tpi, double sided 48 tpi, and double sided 96 tpi drives as drives A and B. (What fun!)

You will also be able to boot up with any format disk that drive A will read. In other words, if drive A is a 96 tpi double sided drive, you'll be able to boot up with 191K, 390K, and 784K disks.

In fact, even if you are only going to



Monitor Select Circuit (After)



Floppy Side Select

use 390K drives, I think you'll find the Pro 8 package a valuable addition.

Define Your Own Cursor

With the Pro 8 package, you will be able to set your cursor to any character you like.

You could select a non-blinking block for text entry, a blinking underline for spreadsheets, and an invisible space for games (makes games look a lot better). You get to choose blink or non-blink no matter which character you select. In fact, you could even use one of the greek characters for the cursor. For now, the new Pro-8 package is the only way you can do this.

Details about Drives

Finally, let's take a closer look at disk drives.

The Kaypro 2 comes with 48 tpi (track per inch) single-sided drives. There is a little less than an inch of disk space so there are 40 usable tracks. The Kaypro 4 doubles the space by using double-sided 48 tpi drives.

The drives we are using on our new 8 are double-sided 96 tpi Tandon 100-4s. With these drives we get 80 tracks on each side of the disk. The tracks are narrower (the head writes a narrower data field) and the head only travels half as far each time the stepper motor steps.

Of course, when we made the change, we wanted to be able to continue to read and write disks for the 2 and the 4 which means when 48 tpi disks are used in 96 tpi drives the drive must step twice when it moves from one track to the next. We took care of this in the monitor.

In fact, our Pro 8 package will read, write, and format Kaypro II, 4, and 8 disks. Once the disk is formatted, the Pro-8 system looks at the disk (in either drive) and then automatically knows how to read or write that disk. The only thing you have to do is remember to warm boot when changing disk types. (You should warm boot when changing disks anyway.)

One Compatibility Problem

The limitation on this scheme is that there is a problem when 96 tpi drives write onto a disk that has been written on by a 48 tpi drive.

The data will be written correctly and will be usable for a while but soon there may be disk errors. This is because the 96 tpi heads leave narrower data track.

When the 96 tpi drive writes over a track written previously by a 48 tpi drive, a narrow data track is laid on top of the wide track left by the 48 tpi drives. The narrow track does not completely obliterate the wide track. Thus you can get errors, especially when reading the new data with one of the 48 tpi drives.

However, you can format and write an unformatted Kaypro II or 4 disk (48 tpi) with the 96 tpi drives and they will work forever. You can read them (and write to them) with the II and 4 with no problems. Once you've written on them with the II or 4, however, you will have created the wide data paths so don't plan to write on them with the 8 (reading's OK) until you have erased them with a magnet or recording tape eraser.

So there you have it, a very capable 784K per drive on your original Kaypro II, all without giving up compatibility with other Kaypros or ease of use. Now, with the speed-up and all this room, there's really no excuse to move up to something like a PC that's so much more limited. Is there?

■ ■ ■

Slicer Column

By David Thompson

It's not quite cricket for me to put just my name on this Slicer Column. You see, I've gotten so much great information from Earl Hinrichs, Dean Klein, Richard Harms, and Otto Baude that I'm tempted to add four more names to the author's line.

I haven't mentioned the others in previous issues for two very good reasons (as well as oversight).

First, Otto was the one who first called me about supporting the Slicer in Micro C and he has followed up by providing a lot of really good information.

Second, Otto came to the last SOG and brought Slicer tee shirts for my two little ones, Jennifer and Erin. They were men's size large which suited Erin just fine. She wears her bright purple Slicer shirt as a nightgown.

I understand that the whole crew will be coming to the next SOG. So, if you start hearing a lot more about Dean, Richard, and Earl after July 27, 1984, you'll know they showed up with oversize tee shirts.)

But, in the meantime, I can't ignore them totally; especially since they are doing some very interesting things. (Talk about product support, catch a load of the following.)

In early December I received a long handwritten note from Dean Klein (hardware designer of the Slicer). In the note he included details on the greatly anticipated expansion board. (The note was very legible until I started drooling on it, but here goes.)

Slicer Expansion Board

1. Same dimensions as the Slicer.
2. 256K Dynamic RAM (TMS4500 controller).
3. Two RS232 async ports (Signetics SC2681).
4. Two sync/async ports (Zilog 8530) capable of networking.
5. Battery backed-up real time clock (Motorola 6818).
6. One centronics printer port.

Dean says that you can add up to three of these boards to the Slicer, but the third board can only have 128K and cannot interrupt the processor.

They will be receiving the first prototypes on Jan 2 and plan to begin sales in February or March. The expansion boards will be available in kit form as well as assembled and tested.

In kit form, the expansion board will be available as a bare board, bare board with RAM, and complete package. They haven't set the prices for the new board yet (depends a lot on the prices of ICs) but the kits should be approximately the same price as the equivalent Slicer package.

Dean expects the expansion board to help turn the Slicer into a very powerful multi-user system.

PC Bus Board

Dean is also working on a board which will accept IBM PC plug-in cards. It will simulate the 8088 type bus signals at the same speed as the PC.

A Direct Quote

"It will not be a full PC implementation because it will not support direct memory addressing (DMA). Since the PC uses DMA to handle refresh for dynamic RAM, this bus will not support add-on memory (you have to add memory via the expansion board)."

"This board, however, will let you use PC video and I/O boards and it will have connectors for a PC compatible or parallel ASCII keyboard. Thus you can use the Slicer without a terminal and have some degree of PC compatability."

Dean began taping the PC board (what else you gunna' call it?) in mid December and expects to have it available in late Spring.

New Software

Meanwhile Earl Hinrichs hasn't been sitting on his keyboard. He's working on the PC emulation software. You will be able to run a program (in the CPM86 environment) which will set up the Slicer to read MSDOS disks and run MSDOS software. The Slicer won't, however, be running MSDOS. Pretty fancy, huh?

Also, Earl has been extending the CPM86 BIOS and the Slicer's monitor so that you can run a winchester. His setup program lets you specify floppy step rates, disk sizes and types. Plus, you can tell the system to boot off any disk, including a winchester.

More from Otto

Even though Dean does the hardware and Earl writes the software, Otto gets to do more than deliver tee shirts. Most manufacturers forget that someone has to figure out which terminals and disk drives and power supplies work with their single-board whiz bang.

It looks like Otto has taken on this arduous task which saves you and me a lot of fumbling around. Plus, Otto has a real nose for bargains.

80186s

Plus, Otto has kept me posted on the difficulties he has had getting 80186s. It appears that Intel is still having problems with yield. They tell Otto that they are getting very few 6 MHz parts and almost none that will run 8 MHz. The Slicer folks have just received a new shipment of 150 parts, all 6 MHz.

Meanwhile Intel's 80186 product manager has been telling a reporter from Buyer's News that everything is hunky-dory and that Intel is producing tens-of-thousands of 186s.

It turns out that Radio Shack is planning to come out with a 186 based system shortly.

So the question is, are the 186s really buggy or is Intel suddenly shipping great quantities of prime parts to Radio Shack and using quality as an excuse for putting off everyone else?

I've been doing a little checking around and I've learned from a very solid source that Intel is really having problems producing 186s. (He is so solid that I can't tell you who he is, they might not let him past the guards again.)

Meanwhile, Otto says that he has stopped counting how many Slicers they've shipped, (over 300 is a good guess) but they are limited now by Intel's production problems.

Cheap Terminals and other Notes

According to Otto, Jim Tanner's (Digital Research Computers of Texas) new terminal board works very well with the Slicer. The terminal has a reset problem which Otto thinks might be a problem with a 74LS04, but on the whole he thinks it is a tremendous value. You get

(continued on page 18)